

# UCLCampus

Une application mobile Open Source pour étudiants UCL

Mémoire présenté par  
**Benjamin DAUBRY , Bruno MARCHESINI**

en vue de l'obtention du grade de Master  
**INFO2M & SINF2M**

Promoteur(s)  
**Yves DEVILLE**

Lecteur(s)  
**Louis DESTRÉE, Charles PECHEUR**

Année académique 2017-2018



# Abstract

Pour l'instant, l'accès aux différentes ressources des campus de l'Université catholique de Louvain se fait via divers moyens depuis différents supports tels que Moodle, ADE Expert, le portail UCL, l'application Resto4U, etc. Ceci n'est pas des plus rapide d'accès, ni des plus intuitif pour un étudiant qui arrive pour la première fois à l'université.

Dans le cadre de ce mémoire, le projet UCLCampus consiste au développement d'une application rassemblant et facilitant l'accès à de nombreuses informations, aides et services aux étudiants des infrastructures de l'Université Catholique de Louvain. L'application fournie découle de trois années de développement par différents binômes de mémorants et est une application hybride, c'est à dire qu'elle s'oriente vers un support pour plusieurs plateformes. Pour convenir à cette caractéristique, elle a été réalisée à l'aide du framework Ionic.

Il est également à noter que ce projet est Open Source afin de permettre à tout étudiant de s'impliquer dans le développement d'une application qui répond à ses besoins. Cette application se veut être développée par des étudiants pour des étudiants.

Ce mémoire nous a fait prendre conscience de la complexité à reprendre en charge un projet existant et de travailler dans un environnement impliquant un grand nombre de services décentralisés. Il n'est pas facile de s'y retrouver et d'harmoniser les ressources que l'on obtient.



# Table des matières

|            |                                                                   |           |
|------------|-------------------------------------------------------------------|-----------|
| <b>I</b>   | <b>Introduction</b>                                               | <b>13</b> |
| <b>II</b>  | <b>Background</b>                                                 | <b>16</b> |
| 1          | Qu'est ce qu'un projet Open Source .....                          | 17        |
| 1.1        | Histoire .....                                                    | 17        |
| 1.2        | Différence entre la notion de logiciel libre et Open Source ..... | 17        |
| 1.3        | Cadre théorique : The Open Source Definition .....                | 18        |
| 1.4        | Propriétés intellectuelles .....                                  | 19        |
| 1.5        | Les licences Open Source .....                                    | 20        |
| 1.6        | Composants d'un projet Open Source .....                          | 21        |
| 1.7        | Avantages d'un projet Open Source .....                           | 23        |
| 1.8        | Mise en parallèle avec l'application UCLCampus .....              | 24        |
| 2          | Frameworks utilisés .....                                         | 24        |
| 2.1        | Ionic .....                                                       | 25        |
| 2.2        | Cordova .....                                                     | 27        |
| 2.3        | Angular .....                                                     | 27        |
| 3          | Enterprise Service Bus .....                                      | 30        |
| 3.1        | Cadre théorique .....                                             | 31        |
| 3.2        | Utilisation à l'UCL avec WSO2 .....                               | 31        |
| 4          | Applications UCL .....                                            | 32        |
| 4.1        | Shibboleth .....                                                  | 34        |
| 4.2        | OSIS .....                                                        | 35        |
| 5          | Portail UCL, une source d'information .....                       | 36        |
| 6          | Application UCLCampus 2016-2017 .....                             | 37        |
| 6.1        | Contexte .....                                                    | 37        |
| 6.2        | Choix d'implémentation .....                                      | 37        |
| 6.3        | Apports à l'application 2016-2017 .....                           | 38        |
| <b>III</b> | <b>Cahier des charges</b>                                         | <b>39</b> |
| 1          | Contexte .....                                                    | 40        |
| 1.1        | Besoins des étudiants .....                                       | 40        |
| 1.2        | Ressources existantes .....                                       | 43        |
| 1.3        | Applications liées aux fonctionnalités .....                      | 44        |
| 1.4        | Ressources disponibles .....                                      | 45        |
| 2          | Spécifications fonctionnelles .....                               | 47        |
| 2.1        | Front-end .....                                                   | 47        |
| 2.2        | Back-end .....                                                    | 49        |
| 2.3        | User Stories .....                                                | 49        |
| 3          | Spécifications non fonctionnelles .....                           | 51        |

|           |                                                               |            |
|-----------|---------------------------------------------------------------|------------|
| <b>IV</b> | <b>Développement d’UCLCampus</b>                              | <b>53</b>  |
| 1         | Continuité dans le mouvement Open Source .....                | 54         |
| 1.1       | Licence .....                                                 | 54         |
| 1.2       | Forge .....                                                   | 55         |
| 1.3       | Création d’une communauté.....                                | 56         |
| 1.4       | Données sensibles.....                                        | 56         |
| 2         | Choix d’implémentation.....                                   | 56         |
| 2.1       | Ionic .....                                                   | 57         |
| 2.2       | L’objet Promise.....                                          | 57         |
| 2.3       | User Interface.....                                           | 57         |
| 3         | Sécurité.....                                                 | 58         |
| 3.1       | Enjeux et solutions pour l’authentification aux services..... | 58         |
| 3.2       | Stockage de l’information.....                                | 59         |
| 4         | Modules réalisés .....                                        | 61         |
| 4.1       | Accueil .....                                                 | 61         |
| 4.2       | Campus .....                                                  | 62         |
| 4.3       | Études.....                                                   | 68         |
| 4.4       | Outils.....                                                   | 75         |
| <b>V</b>  | <b>Prise de recul</b>                                         | <b>80</b>  |
| 1         | Open Source .....                                             | 81         |
| 2         | Framework Ionic .....                                         | 82         |
| 2.1       | Avantages et inconvénients.....                               | 82         |
| 2.2       | Mise à jour future .....                                      | 83         |
| 3         | Environnement UCL .....                                       | 83         |
| 4         | Sécurité.....                                                 | 84         |
| 5         | Réalisation .....                                             | 86         |
| 5.1       | Comparaison avec le cahier des charges .....                  | 86         |
| 5.2       | Améliorations proposées .....                                 | 90         |
| <b>VI</b> | <b>Conclusion</b>                                             | <b>93</b>  |
|           | <b>Bibliographie</b>                                          | <b>95</b>  |
| <b>A</b>  | <b>Enquête réalisée par les mémorants précédents .....</b>    | <b>99</b>  |
| <b>B</b>  | <b>Carnet des contacts liés au mémoire.....</b>               | <b>101</b> |
| <b>C</b>  | <b>Illustrations des requêtes et leurs réponses .....</b>     | <b>102</b> |
| <b>D</b>  | <b>Guide d’utilisateur .....</b>                              | <b>112</b> |



# Table des figures

|    |                                                                                                                                                                   |    |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1  | Évolution des parts de marché des systèmes d'exploitation des appareils mobiles ..                                                                                | 24 |
| 2  | Évolution des parts de marché des systèmes d'exploitation des tablettes .....                                                                                     | 25 |
| 3  | Représentation de l'architecture d'une application et des interactions entre les composants avec le framework Ionic 3 [1] .....                                   | 26 |
| 4  | Composition d'un module.....                                                                                                                                      | 28 |
| 5  | Les différentes formes de data binding .....                                                                                                                      | 29 |
| 6  | Infrastructure informatique de l'UCL .....                                                                                                                        | 32 |
| 7  | État initial des interactions entre les services de l'UCL .....                                                                                                   | 33 |
| 8  | État des interactions entre les services de l'UCL en 2017 .....                                                                                                   | 33 |
| 9  | État final des interactions entre les services de l'UCL .....                                                                                                     | 33 |
| 10 | Répartition du niveau d'étude des répondants et répartition des répondants par faculté - Graphique repris des mémorants de 2016-2017.....                         | 41 |
| 11 | Résultats de la question : "Quelles fonctionnalités/outils veux-tu avoir dans le cadre de tes études ?" - Tableau repris des mémorants de 2016-2017 .....         | 42 |
| 12 | Résultats de la question : Quelles fonctionnalités/outils veux-tu avoir dans le cadre de la vie sur le campus ?" - Tableau repris des mémorants de 2016-2017..... | 42 |
| 13 | Arborescence des pages .....                                                                                                                                      | 48 |
| 14 | Flux de la persistance des tokens synchrones.....                                                                                                                 | 59 |
| 15 | Flux de la persistance des tokens asynchrones .....                                                                                                               | 59 |
| 16 | Diagramme de séquence pour l'affichage de l'accueil .....                                                                                                         | 62 |
| 17 | Diagramme de séquence pour l'affichage des news .....                                                                                                             | 64 |
| 18 | Diagramme de séquence pour l'affichage des événements.....                                                                                                        | 65 |
| 19 | Diagramme de séquence pour l'affichage des sports.....                                                                                                            | 67 |
| 20 | Diagramme de séquence pour l'affichage des bibliothèques.....                                                                                                     | 69 |
| 21 | Diagramme de séquence pour l'affichage de la page études .....                                                                                                    | 72 |
| 22 | Diagramme de séquence pour l'ajout d'un cours et de l'affichage de l'horaire .....                                                                                | 72 |
| 23 | Diagramme de séquence pour l'affichage du programme de cours .....                                                                                                | 73 |
| 24 | Diagramme de séquence pour l'affichage du support et la recherche d'un membre de l'UCL.....                                                                       | 74 |
| 25 | Diagramme de séquence pour l'affichage de la carte .....                                                                                                          | 76 |
| 26 | Identification d'un utilisateur et récupération des données.....                                                                                                  | 85 |
| 27 | Récupération du token et requêtes aux API.....                                                                                                                    | 85 |



# Liste des tableaux

|     |                                                        |     |
|-----|--------------------------------------------------------|-----|
| 1   | Fonctionnalités réalisées propres à l'application..... | 86  |
| 2   | Fonctionnalités réalisées de la partie "Campus" .....  | 87  |
| 3   | Fonctionnalités réalisées de la partie "Etudes" .....  | 88  |
| 4   | Fonctionnalités réalisées de la partie "Outils" .....  | 88  |
| 5   | Fonctionnalités restantes du module Accueil .....      | 89  |
| 6   | Fonctionnalités restantes de la partie "Campus" .....  | 89  |
| 7   | Fonctionnalités restantes de la partie "Etudes" .....  | 89  |
| 8   | Fonctionnalités restantes de la partie "Outils" .....  | 89  |
| B.1 | Personnes contactée à propos du mémoire.....           | 101 |



# Glossaire

**AES** Advanced Encryption Standard. 60

**API** Application Programming Interface. 7, 27, 30, 32, 38, 46, 52, 56–61, 68, 69, 71, 73–77, 81, 84–86, 88

**BSD** Berkeley Software Distribution. 20, 21

**CLI** Commande Line Interface.. 26, 27

**CMS** Content Management System. 36, 37, 46

**CSS** Cascading Style Sheets. 25–27, 30, 82

**cURL** client URL Request Library. 85

**DFSG** Debian Free Software Guidelines. 17

**DOM** Document Object Model. 29, 30

**EAI** Enterprise Application Integration. 31

**ESB** Enterprise Service Bus. 30–32, 36, 45, 58, 68, 84, 88

**FSF** Free Software Foundation. 20

**GNU** GNU's not Unix. 17, 20

**GNU GPL** GNU General Public License. 19–21, 35, 36, 52, 54, 55, 90

**HTML** Hypertext Markup Language. 25–30, 35, 36, 52, 57, 60, 63, 77, 79, 82

**HTTP** Hypertext Transfer Protocol. 57, 58, 61

**JSON** JavaScript Object Notation. 45, 60, 62–66, 68–71, 74, 75, 77, 88, 91

**KUL** Katholieke Universiteit Leuven. 14

**LDAP** Lightweight Directory Access Protocol. 32

**MIT** Massachusetts Institute of Technology. 20, 21, 25, 27, 54, 55

**MOM** Message-Oriented Middleware. 31

**OS** Operating System. 27, 61

**OSI** Open Source Initiative,. 17, 20

**OSIS** Open Student Information System. 4, 32, 35, 36

**PHP** Hypertext Preprocessor. 36

**REST** Representational State Transfer. 32, 45, 58

**RSS** Rich Site Summary ou Really Simple Syndication. 44, 46, 47, 61, 63, 64, 66, 87

**SAML** Security Assertion Markup Language. 34

**Sass** Syntactically Awesome Style Sheets. 26, 57

**SDK** Software Development Kit). 25, 75

**SOA** Service Oriented Architecture. 31, 32

**SOAP** Simple Object Access Protocol. 32

**SQL** Structured Query Language. 34, 63

**SSO** SingleSign-On. 34, 35

**URL** Uniform Resource Locator. 26, 75

**W3C** World Wide Web Consortium. 30

**XML** Extensible Markup Language. 30, 63, 87

**YQL** Yahoo Query Language. 63, 65, 68



Première partie

Introduction

Étant des étudiants à l'Université catholique de Louvain, nous avons remarqué qu'il manquait une application rassemblant tous les services auxquels des étudiants peuvent avoir besoin. Les sources d'informations sont dispersées à travers différents sites et applications. Il n'est pas facile pour une nouvelle personne qui arrive sur le campus de savoir où trouver toutes les informations dont il a besoin et peut rapidement être perdu dans cette multitude de services proposés. C'est là dedans que réside notre motivation à créer une application appelée UCLCampus qui regrouperait les informations et intérêts académiques ainsi qu'extra-académiques des étudiants de différents campus de l'UCL.

Le début de ce projet a commencé durant l'année académique 2015-2016 par un premier groupe de mémorants. L'année suivante, nos plus proches prédécesseurs Jérôme Lemaire et Corentin Lamy ont repris en charge ce mémoire. Cependant, au vu de difficultés techniques rencontrées, ils ont décidé de repartir de zéro et de construire des bases solides à partir desquelles nous sommes repartis. Il faut également savoir que d'autres universités, telles que la KUL, disposent déjà d'une application atteignant l'objectif de rassembler leurs services en un seul point. C'est dans ce contexte que se situe ce mémoire où à l'heure numérique il est indispensable à notre université de s'adapter et d'utiliser les réseaux de communication pourvus par leurs étudiants. Il est aussi à noter que dû à l'accueil d'étudiants en échange international, il se veut que l'application soit bilingue, c'est à dire en anglais et en français.

L'approche abordée pour le développement de cette application a commencé par l'apprentissage de la prise en main du projet créé par nos prédécesseurs. Il n'a pas été des plus simples pour comprendre l'architecture et le fonctionnement de chaque module existant. Ensuite, nous avons repris les besoins des étudiants qui avaient été préalablement identifiés dans le mémoire de l'année académique 2016-2017. Une analyse de ceux-ci nous a permis de mettre en évidence les besoins auxquels l'application ne répondait pas encore et les améliorations à apporter aux modules existants pour qu'ils soient encore plus rapides et simples d'utilisation. Pour finir, c'est sur base de cela que nous avons développé les nouveaux modules et amélioré les anciens déjà présents. De nombreux rendez-vous avec les différents services et membres du personnel UCL se sont également déroulés pour trouver des solutions à des problèmes rencontrés mais aussi pour leur présenter l'état d'avancement au fur et à mesure de l'application afin d'avoir un feedback et d'autres pistes d'améliorations que nous avons pris en compte tout au long du développement.

Tout le travail réalisé durant ce mémoire est visible sur le GitHub public suivant : <https://github.com/bmarchesini/UCLCampus>

L'application est multiplateforme pour qu'elle soit disponible sur les appareils Apple et Android. Elle sera déployée sur les stores Apple et Android pour la rentrée académique 2018. Le projet quant à lui est Open Source, cette approche permet d'une part d'assurer la survie du projet dans le futur et d'autre part de laisser aux étudiants la possibilité de définir eux-même leurs besoins et de contribuer à la réalisation d'une application qui leur correspond. L'utilisation de multiples technologies Open Source nous a également poussé à suivre ce choix.

Dans ce document, nous commençons par établir le contexte et les environnements dans lesquels l'application a été pensée. Pour cela nous décrivons en détails le concept de l'Open Source qui a un lien direct dans la façon dont le projet a été créé. Nous décrivons également, dans des parties distinctes, les différents outils de développements et aspects techniques qui ont été utilisés et mis en oeuvre durant l'élaboration de l'application ainsi que l'environnement informatique de l'UCL dans lequel ce projet doit prendre place. Cette partie se termine sur une analyse de l'application fournie par les mémorants précédents.

Ensuite, la seconde partie se porte sur le cahier des charges de l'application qui fait part

des exigences qui lui sont relatives. Elle débute par la description du contexte dans lequel le développement d'UCLCampus a continué. Par la suite, les spécifications fonctionnelles de l'application, du point de vue front-end, back-end et user stories, sont listées et expliquées. Pour finir, une section reprend les spécifications non fonctionnelles.

La troisième partie, quant à elle, décrit plus précisément le développement même de l'application et est composée de quatre sections. La première se concentre sur l'environnement Open Source du projet. La seconde fait part des choix d'implémentation qui ont été réalisés. L'avant-dernière section se penche sur la sécurité, plus précisément en ce qui concerne le stockage de l'information et l'authentification aux services de l'UCL. Et pour finir cette partie, la dernière section présente en détail les modules réalisés dans l'application, en ce qui concerne le fonctionnement de ses fonctionnalités.

Mais encore, avant de terminer, l'avant dernière partie tend à prendre du recul sur le projet. Les différentes sections traversent les éléments principaux qui le composent avec les difficultés associées qui ont été rencontrées durant son développement et les solutions qui y ont été apportées lorsque cela était possible. Ces éléments sont principalement l'environnement, le processus de développement et les choix d'implémentation. Mais il y a également une comparaison qui est réalisée entre le projet final et le cahier des charges afin d'identifier ce qui a été réalisé et ce qui reste concrètement à faire ou à améliorer dans le futur.

Et pour conclure, nous faisons état de tout ce que nous avons appris durant la réalisation de ce mémoire ainsi que ce que notre solution apportera aux futurs étudiants de l'UCL.



Deuxième partie

Background

# 1 Qu'est ce qu'un projet Open Source

La notion d'Open Source est appliquée aux logiciels dont la licence doit respecter un nombre de critères établis par l'Open Source Initiative. Le but principal est de mettre à disposition le code source, provenant généralement d'une collaboration entre programmeurs, au grand public. La philosophie qui en résulte est l'atteinte de nouveaux objectifs par de nouvelles personnes à l'aide du travail réalisé et de l'expérience acquise par leurs prédécesseurs.

Plus précisément, dans le domaine de l'informatique, cela permet aux programmeurs de ne pas partir de zéro. Grâce au partage des codes sources de projets, la quantité d'expériences ne fait qu'augmenter et incite à se baser sur ces codes existants afin d'améliorer ou de réaliser de nouveaux projets. Cela permet également d'améliorer des projets existants par d'autres programmeurs.[2][3][4]

## 1.1 Histoire

Il faut remettre cette notion dans son contexte afin de comprendre comment elle a émergé. Cela a commencé par la différenciation entre logiciels propriétaire et logiciel libre, le concept de ce dernier étant plus ancien que celui de l'Open Source.

L'idée de logiciel libre est née d'un mouvement fondé par Richard Stallman, père fondateur du projet GNU. Il a fondé l'organisation américaine à but non lucratif Free Software Foundation le 4 octobre 1985 dont la mission principale est la promotion du logiciel libre et la défense des utilisateurs. Ce concept de logiciel libre s'opposait aux limitations de licences aux utilisateurs imposées par les logiciels propriétaires. L'objectif du libre est d'offrir un plus grand nombre de libertés aux utilisateurs individuels ou en collectivité quant à l'utilisation des logiciels.

Pour qu'un logiciel puisse être qualifié de libre, il doit respecter les 4 critères de liberté suivants :

- la liberté d'exécuter le programme comme vous le souhaitez, et à toute fin (liberté 0) ;
- la liberté d'étudier le fonctionnement du programme, et le modifier comme vous le souhaitez pour votre ordinateur (liberté 1) ;
- la liberté de redistribuer les copies que vous avez reçues (liberté 2) ;
- la liberté de distribuer des copies de vos versions modifiées pour donner à toute la communauté une chance de profiter de vos modifications (liberté 3).

Malheureusement, l'association entre logiciel libre et logiciel gratuit a constitué un obstacle quant à son adoption commerciale. Cette association est due à la confusion produite par leur traduction anglaise « *free software* » applicable aux deux termes français.

Un logiciel libre devrait être également disponible pour l'usage, le développement et la distribution de nature commerciale. C'est de cette confusion qu'a émergé la notion d'Open Source.

En 1998 en Californie, les partisans du logiciel libre se sont réunis afin d'apporter les idées et avantages du logiciel libre à l'industrie du logiciel commercial, c'est lors de cette session que le label « *Open Source* » a été créé. L'Open Source Initiative est dès lors fondé par Éric Steven Raymond et Bruce Perens. Sa première tâche était de rédiger l'Open Source Definition, qui était à l'origine dérivée des directives de logiciel libre Debian (DFSG), et de l'utiliser pour créer une liste exhaustive de licences approuvées par l'OSI.

[5][6]

## 1.2 Différence entre la notion de logiciel libre et Open Source

La différence formelle entre Open Source et logiciel libre n'a presque pas de conséquence dans l'évaluation des licences. Un contre exemple célèbre est le projet Darwin d'Apple la notion Open Source était applicable selon l'OSI, mais la notion de libre ne l'était pas au sens de la Free Software Foundation.

Ces deux désignations sont en réalité concurrentes pour un même type de licence de logiciel. À la différence que l'utilisation de la désignation logiciel libre met l'accent sur la finalité philosophique

et politique de la licence tandis que l'utilisation de la désignation Open Source met en avant la méthode de développement et de diffusion du logiciel.

Si on l'analyse sous un angle économique, la notion Open Source contribue plutôt à la création d'une nouvelle forme de marché et d'économie où il s'agit de fournir une approche plus pragmatique des avantages du logiciel libre, en omettant les connotations politiques et philosophiques, pour n'en garder que les points forts sur le plan de l'ingénierie.[2][7]

### 1.3 Cadre théorique : The Open Source Definition

Open Source ne signifie pas juste d'accéder au code source, l'Open Source Initiative a rédigé un ensemble de 10 critères devant être respectés afin de correspondre aux conditions de distribution des logiciels Open Source.

[5]

Ces 10 critères sont les suivants :

- **Redistribution libre** : la licence ne limitera quiconque de vendre ou donner le logiciel comme composant d'une distribution de logiciels constitués de programmes provenant de sources différentes. La licence n'exigera point de redevance ou d'autres commissions sur de telles ventes.
- **Code source** : le programme doit inclure le code source et permettre sa distribution sous sa forme compilée aussi bien que sous sa forme de code source. Quand un produit n'est pas distribué avec le code source, il doit y avoir des moyens rendus publics afin de l'obtenir sans autres frais qu'un coût de reproduction raisonnable, avec préférence via le téléchargement gratuit depuis Internet. Le code source doit être la forme privilégiée afin qu'un programmeur puisse modifier le programme aisément. Le code source délibérément rendu compliqué à comprendre est interdit ainsi que la soumission des formes intermédiaires (résultat d'un traducteur automatique ou d'un préprocesseur).
- **Oeuvres dérivées** : la licence doit permettre les modifications et les travaux dérivés, et doit également leurs permettre d'être distribués sous les mêmes termes que ceux de la licence du logiciel original.
- **Intégrité du code source de l'auteur** : la licence peut limiter la redistribution du code source sous forme modifiée seulement si la licence autorise la distribution de fichiers de correction (patch) avec le code source dans le but de modifier le programme au moment du développement. La licence doit explicitement permettre la distribution de logiciels développés à partir de codes sources modifiés. La licence peut exiger que des travaux dérivés portent un nom différent ou un numéro de version distinct du logiciel original.
- **Non-discrimination contre des personnes ou groupes** : la licence ne doit pas discriminer aucune personne ou aucun groupe de personnes.
- **Non-discrimination contre des champs d'application** : la licence ne doit pas limiter quiconque de se servir du programme dans un champ d'application spécifique. Par exemple, elle ne peut pas interdire l'utilisation du logiciel dans le cadre d'une entreprise ou pour la recherche génétique.
- **Distribution de licence** : les droits attachés au programme doivent s'appliquer à tous ceux à qui le programme est redistribué, sans le besoin d'obtention d'une licence supplémentaire par ces parties.
- **la licence ne doit pas être spécifique à un produit** : les droits attachés au programme ne doivent pas dépendre du fait qu'il fasse partie d'une distribution particulière de logiciels. Si le programme est extrait de cette distribution et utilisé ou distribué sous les termes de sa propre licence, toutes les parties auxquelles il est redistribué doivent avoir les mêmes droits que ceux que l'on accorde en accord à la distribution originelle de logiciels.
- **la licence ne doit pas restreindre d'autres logiciels** : la licence ne doit pas placer de restrictions d'autres logiciels distribués avec le logiciel licencié. Par exemple, la licence ne doit pas exiger que tous les autres programmes distribués sur le même support soient

des logiciels Open Source.

- **la licence doit être neutre sur le plan technologique** : aucune disposition de la licence ne peut aller à l'encontre d'une quelconque technologie ou style d'interface.

## 1.4 Propriétés intellectuelles

Comme énoncé précédemment, dans le domaine de l'informatique, la réutilisation de codes sources existants est un fait réel et récurrent. Il n'est pas habituel de tout recréer, cela irait à l'encontre de la productivité et l'efficacité de l'ensemble des développeurs à travers le globe. Il est très facile de trouver sur Internet, à l'aide de forums et sites de partage, un ensemble de solutions, sous forme de codes ou d'idées, à des problèmes posés.

Le questionnement à propos des propriétés intellectuelles des travaux publiés sur les sites de partage et des forums prend tout son sens.

Les notions de copyright, de copyleft et de licences seront explorées afin de nous éclairer sur cette question. [8][5]

### Copyright

Pour qu'un programme soit Open Source, il faut qu'une licence appropriée y soit associée. Mais avant cela c'est le copyright qui s'y applique. Vu que ce programme n'appartient pas au domaine public, un auteur ou propriétaire y est associé, c'est le développeur ou l'entreprise l'engageant qui détient les droits d'auteurs.

Le détenteur a la totale liberté de définir l'utilisation du logiciel. Cette liberté peut se traduire par l'interdiction de son utilisation, par la vente des droits à une personne physique ou morale, ou par la précision de conditions de son utilisation à l'aide des termes décrits dans la licence qui y sera associée.

### Les clauses communes des licences Open Source

L'existence d'une licence sur un programme ne peut se faire qu'à partir des droits d'auteur, sans ceux-ci une licence ne peut exister.

De plus, le simple fait de publier le code source d'un projet n'en fait pas un projet Open Source. Cette singularité est très importante, seuls les droits inscrits dans la licence rendent ce projet Open Source ou non.

Il est aussi important de souligner que seulement l'auteur du programme sera libre vis-à-vis de celui-ci, toute autre personne devra se soumettre aux termes de la licence stipulant les conditions d'utilisation du programme.

Les licences Open Source ont un ensemble de clauses communes, ces clauses sont les suivantes :

- L'identification claire du propriétaire du copyright du programme, mais également pour les copies ou travaux dérivés ;
- L'obligation de conserver la licence sur le programme ;
- La protection de l'auteur vis-à-vis des utilisateurs de son programme, ses éventuels défauts et les conséquences de ces défauts.

Pour la distribution du programme, il sera nécessaire de respecter les lois nationales ainsi que la licence qui ne pourra contredire en aucun cas le droit national.

### Copyleft

La notion « Copyleft » fait référence aux licences autorisant des oeuvres dérivées, mais sous la condition que ces oeuvres utilisent la même licence que l'oeuvre originale. Par exemple, si un programmeur écrit un logiciel et le libère sous une licence GNU GPL, qui est une licence copyleft très courante, puis que quelqu'un d'autre modifie ce logiciel et distribue cette version modifiée, celle-ci devra également être sous licence GNU GPL, ainsi que tout nouveau code dérivé de la

version modifiée. La licence copyleft garantit que la propriété soit perpétuée à toutes les dérivées en aval, le but principal est d'empêcher la distribution sous licence propriétaire d'un programme dérivé d'un autre programme sous licence Open Source.

Une majorité des licences copyleft sont Open Source mais toutes les licences Open Source ne sont pas copyleft. Un logiciel publié sous une licence Open Source non copyleft pourra être utilisé dans le cadre de programmes distribués sous d'autres licences, y compris des licences propriétaires.

La notion d'oeuvre dérivée est importante dans ce contexte, un programme bis est une oeuvre dérivée du programme original si ce programme bis ne peut pas fonctionner sans le programme original indépendamment des modalités techniques d'échange c'est-à-dire si le programme bis appelle des services du programme original au moyen d'un protocole d'échange standard.

## Cas juridique

La cohabitation de programmes Open Source et propriétaire au sein d'un même projet représente un cas juridiquement complexe. Si les deux types de programmes sont interfacés par des protocoles d'échanges standards, il n'y a pas de soucis. Si les deux programmes sont fusionnés en un seul programme, les deux licences devront être respectées et compatibles. Dans le cas contraire, les deux programmes ne pourront pas cohabiter en cas de diffusion du projet. Mais dans le cas où ces deux programmes sont interfacés par des protocoles d'échanges standards, aucun souci ne rentre en vigueur.

### 1.5 Les licences Open Source

Tout d'abord, pour qu'une licence soit reconnue comme licence Open Source elle doit être reconnue comme telle par l'OSI. Tout le monde peut créer sa propre licence s'il le désire, mais cela nécessite de s'intéresser à la compatibilité avec les autres licences sur le plan juridique et d'éviter les clauses mutuellement incompatibles qui poseront problème lors de l'intégration de plusieurs logiciels ou bibliothèques de programmation.

Le nombre de licences Open Source effectives est impressionnant, le site GNU en recense à lui seul plus d'une centaine. L'univers dans lequel elles gravitent n'est pas d'une moindre complexité, de nombreuses d'entre elles sont relativement similaires, certaines sont compatibles entre elles, d'autres non, il y en a aussi avec de véritables différences philosophiques.

A la vue de cette complexité, seul un nombre restreint d'entre elles sont étudiées dans cet écrit, celles qui sont le plus couramment utilisées. Les licences GNU et BSD seront regroupées dans leur famille respective afin d'expliquer leur définition générale qui en découle, les licences au sein de ces familles ne diffèrent généralement qu'au niveau des détails. Les licences MIT et Apache seront expliquées indépendamment.

La famille de licences GNU GPL est une famille de licences copyleft très restrictive. Elle est fournie par la FSF dont la philosophie de liberté défendue est respectée par le droit de modification du programme et de redistribution des modifications à la seule condition que ce soit sous la même licence. Cette condition empêche tout développeur travaillant sous une autre sorte de licence d'avoir accès aux programmes sous licence GNU GPL. Les licences les plus utilisées au sein de cette famille sont les versions 2 et 3. La famille de licences BSD (Berkeley Software Distribution License) est une famille de licences non copyleft, permissive du point de vue composition et dérivation. Quiconque est autorisé sans aucune restriction à réutiliser une partie du logiciel ou son entièreté, qu'il soit intégré dans un projet Open Source, libre ou propriétaire. La licence MIT, aussi appelée X11, est une licence académique non copyleft provenant du Massachusetts Institute of Technology. Elle donne le droit illimité à toute personne d'utiliser le logiciel, le modifier, le fusionner, le publier, le vendre, le distribuer et de changer sa licence. Très proche de la dernière licence BSD, elle ne diffère que par sa dernière clause qui fait état de l'obligation de mettre le nom des auteurs avec la notice de copyright. La licence Apache est une licence non copyleft gérée

par l'Apache Software Foundate, au même titre des licences BSD elle est permissive en matière de composition et de dérivation. Cependant, elle est plus étoffée, complète et explicite dans ce qu'elle autorise. Il y a obligation de maintenir le copyright lors de toute modification.[5][9][8][10]

## Compatibilité des licences

Au sein d'un même projet, les différences entre licences liées au degré de restriction ou à l'approche philosophique peuvent poser des problèmes de compatibilité entre elles. Un projet sous licence permissive ne pourra pas intégrer des programmes avec des licences plus restrictives, mais par contre un projet sous licence plus restrictive pourra intégrer des programmes sous licences plus permissives.

Une mise en contexte s'impose afin de mieux comprendre ce principe. La compatibilité d'une licence X avec une licence Y est correcte si l'intégration d'un programme sous licence X est possible avec un programme sous licence Y sans modifier les termes de sa licence. Si un programme tiers Z utilise deux programmes sous licences différentes X et Y, ce programme hérite des exigences des deux licences. Cependant, si une contradiction est en vigueur entre les licences héritées, un des deux programmes devra être ignoré.

Maintenant que ce principe est assimilé, ce paragraphe explique brièvement les compatibilités possibles des licences les plus communes expliquées précédemment. La famille de licences GNU GPL est la plus restrictive, celle-ci étant copyleft, ce qui oblige quiconque reprenant un programme sous cette licence de passer à cette même licence. L'avantage de cette famille est la possibilité d'intégration de nombreuses licences plus permissives aux projets sous cette licence. Pour ce qui en est de la famille de licences BSD, de la licence MIT et de la licence Apache, leur compatibilité supérieure, c'est-à-dire à pouvoir être intégrée, est possible potentiellement avec toute autre licence sous réserve qu'elle tienne en compte les obligations de ces licences respectives. La compatibilité inférieure, c'est-à-dire à pouvoir intégrer, est possible, mais est à examiner au cas par cas en fonction des licences rattachées au code externe. Cela est impossible pour des licences non permissives en dérivation et composition.[9][8]

## 1.6 Composants d'un projet Open Source

Qu'est-ce qui compose un projet Open Source ? C'est la question à laquelle nous allons essayer de répondre dans cette sous-section. Les éléments présentés sont nécessaires à tout projet Open Source pour qu'il puisse se développer sainement et survivre à la concurrence.[11]

### Existe-t-il un projet existant ?

Y aurait-il un projet existant auquel la contribution serait plus favorable que la création d'un nouveau projet ? C'est la première question à se poser lorsque l'on veut créer un nouveau projet. La création de nouveaux projets apporte une concurrence motivante et améliore l'offre globale. Mais si trop de projets existent, on remarque un manque d'activité dans ces projets, on peut voir cela comme de l'énergie perdue sur des travaux parallèles. L'esprit Open Source est celui de l'enrichissement collectif de biens communs, donc, si l'objectif n'est pas de construire une startup mais de construire le patrimoine logiciel, il est préférable de contribuer à des projets existants. En se faisant une réputation, petit à petit, il sera possible d'avoir son mot à dire dans la gouvernance du projet auquel on contribue.

## Communauté

La communauté est un élément clé dans un projet Open Source, elle représente l'ensemble des personnes qui participent à sa mise en oeuvre. Sans elle, le projet dépérirait au fur et à mesure du temps. Elle se compose principalement de développeurs mais pas seulement, il y a aussi les utilisateurs qui signalent les bugs rencontrés renforçant sa robustesse, ceux qui s'intéressent de

près au projet à travers les forums, ceux qui réalisent la documentation, ceux qui contribuent aux traductions et ceux qui font connaître le projet en parlant de celui-ci autour d'eux. La communauté a pour but de partager la charge de développement, de dynamiser le projet, de le promouvoir à plus grande échelle, de diffuser le produit fini, de faire évoluer la roadmap et de permettre au projet de survivre ce qui peut être vu comme condition nécessaire à son adoption par les utilisateurs exigeants. Cependant au sein d'une communauté il faut qu'il y ait un leader, un architecte ou un groupe de programmeurs qui peut prendre des décisions importantes quant à l'architecture du projet évitant ainsi la dispersion d'énergie et permettant aux programmeurs de travailler ensemble.

## **Modèle de développement communautaire**

Le modèle de développement communautaire est le modèle de développement à mettre en place lors d'un projet Open Source. Celui-ci s'oppose à l'un des plus connus, le modèle du cycle en V, dont le principe est l'enchaînement linéaire de phases (la phase de conception, de réalisation et de validation). Pour sa mise en place, il faudra procéder à une planification globale des tâches et à leur affectation centralisée. En prenant en compte la communauté, qui gravite autour d'un projet Open Source, il est impossible de mettre un tel modèle en place.

Le modèle de développement communautaire est quant à lui un modèle itératif, il prend tout son sens dans ce style de projet car à tout instant, toute personne peut contribuer au projet, améliorer des modules existants, en créer de nouveaux et apporter de nouvelles idées. Cependant, ce n'est pas toujours le cas, cela dépend de la stratégie mise en place. Certains projets Open Source assez fermés préféreront mettre en place un processus de validation très strict avant de pouvoir contribuer au développement du projet.

La modularité est par la même occasion très recommandée. C'est un principe très propice aux projets Open Source, il consiste à découper un projet en modules indépendants de taille gérable. Il sera dès lors plus simple aux développeurs de se répartir la charge de travail. Les modules posséderont également des interfaces bien définies afin de faciliter l'utilisation d'autres modules, les développeurs éviteront de devoir comprendre des parties de code qui ne leur sont pas nécessaires et pourront se concentrer sur le développement de leur propre module.<sup>1 2</sup>

## **Forge**

La forge est un système de gestion de développement collaboratif de logiciels, elle sera nécessaire pour rassembler la communauté autour du projet Open Source et d'accueillir les sources, les travaux et les échanges. Les forges les plus faciles à utiliser, les mieux intégrées et les plus connues sont GitHub, OverLeaf et Google Code. Elles offrent différents services dont, pour la majorité, un système de gestion des versions, un outil de suivi des erreurs, un outil de gestion des tâches et un outil de communication entre développeurs. En plus des outils proposés par les forges, elles contribuent à faire connaître les projets, si ceux-ci sont publics ils seront directement accessibles via une recherche.

## **Licence**

La licence choisie pour le projet n'est pas sans importance. S'il y a détention du droit d'auteur sur l'ensemble des sources, le changement de licence peut se faire ultérieurement, mais il est quand même préférable de faire le bon choix initialement. Le choix de la licence aura un impact sur la finalité du projet ainsi que sur les programmes qui pourront ou pas y être intégrés. De

---

1. [https://fr.wikipedia.org/wiki/Cycle\\_en\\_V](https://fr.wikipedia.org/wiki/Cycle_en_V)

2. [http://edutechwiki.unige.ch/fr/D%C3%A9veloppement\\_collaboratif](http://edutechwiki.unige.ch/fr/D%C3%A9veloppement_collaboratif)

plus, il faut noter, particulièrement en entreprise, que les utilisateurs n'apprécient pas les licences farfelues. Il est préférable d'utiliser une des licences les plus connues afin de convenir à tous.

## 1.7 Avantages d'un projet Open Source

Maintenant que les aspects théoriques et techniques de l'Open Source ont été couverts, il est temps de se pencher sur ses avantages. La sécurité, le coût, la pérennité, la qualité et la flexibilité sont les principaux avantages qui vont être expliqués dans cette section.[12][7][6]

### Sécurité

Cet avantage peut être analysé sous deux points de vues : le fait d'auditer ce qu'un programme fait en analysant le code source afin de s'assurer que ce qu'on utilise n'a pas de failles critiques et le peer reviewing pour s'assurer que notre propre code est sans danger. Le premier point de vue apparaît lorsque l'on utilise un programme, si celui-ci est Open Source il sera plus facile de l'auditer en analysant ses sources, cela permet d'être certain de ne pas utiliser un programme qui est rempli d'erreurs pouvant poser problème. Le second point de vue, et qui reste le plus intéressant, est la validation des pairs, ces pairs étant d'autres experts en sécurité. Le meilleur moyen d'être certain de la perfection de notre projet sur le plan sécurité est de l'exposer à l'audit de nombreux experts, cela peut se faire plus aisément en le rendant Open Source et accessible par tous. Les failles pourront alors être détectées et corrigées rendant le programme plus robuste et sécurisé.

### Coût

Le coût relativement faible est un attrait très important de l'Open Source. Même si Open Source ne signifie pas gratuit, les programmes sont généralement moins coûteux que les programmes propriétaires.

Une société développant des projets Open Source peut être viable en créant un modèle de service et de soutien informatiques aux sociétés demandeuses.

Un autre point de vue avantageux est la réduction des coûts grâce à la coopération dans un projet Open Source, cela se traduit par l'utilisation du savoir d'autres programmeurs qui ne sont pas employés par la société.

### Pérennité

La pérennité d'un logiciel Open Source réside dans son code qui est accessible par tous, celui-ci n'est pas lié à une firme ou organisation, mais est lié à une communauté qui le maintient, chaque personne pouvant l'améliorer, le modifier et le redistribuer.

Ces solutions Open Source n'ont pas une garantie d'éternité, elles peuvent mourir mais de mort lente. Cela se fera par une désaffection progressive de la part des communautés, pour dans une majorité des cas au profit d'une solution plus prometteuse.

De plus, si l'éditeur original du projet Open Source était un jour défaillant, il resterait possible pour la communauté de reprendre le projet en main ainsi que ses évolutions, c'est ici que réside le principe des licences Open Source. Les critères de pérennité comprenant, la notoriété, l'envergure de déploiements, la dynamique du développement et de la communauté, sont assez simples à évaluer. Une solution Open Source leader offre une meilleure garantie de pérennité qu'un grand nombre des solutions propriétaires.

### Qualité

La qualité d'une majorité des solutions Open Source est réellement supérieure. Étant donné le grand nombre de déploiements, du retour d'expérience, du modèle de développement utilisé



et de l'intégration de composants de haut niveau, cela permet à ces solutions de surclasser les produits propriétaires vieillissants.

Le modèle de développement Open Source oblige une certaine approche et rigueur aux développeurs pour que d'autres développeurs puissent entrer sans trop de difficultés dans le projet, ça résulte souvent en une amélioration de la qualité du code.

## Flexibilité

La flexibilité résulte dans le côté fort paramétrable des logiciels Open Source. Si l'utilisateur s'y connaît un peu en développement, il pourra personnaliser le logiciel selon ses propres besoins. De plus, dans des projets Open Source, si une partie de la communauté prend une vision différente quant à la direction du projet, cette partie peut décider de partir d'une ancienne version pour lui donner la direction désirée. Cette nouvelle direction peut alors devenir une nouvelle référence.

### 1.8 Mise en parallèle avec l'application UCLCampus

L'étude de la notion Open Source, dans sa globalité, à travers les sous sections précédentes a pour but de mettre un cadre théorique quant à notre décision de développer l'application UCLCampus sous une approche Open Source et a comprendre notre choix de licence.

## 2 Frameworks utilisés

De nos jours, les appareils mobiles (tablettes, smartphones, etc..) sont utilisés presque par tous et différents systèmes d'exploitation existent pour ces appareils. Comme on peut le constater sur la figure 1[13] et 2[14], le marché des systèmes d'exploitation mobile en Belgique est largement dominé par Android et iOS avec 61,97 % et 36,46 % des parts du marché respectivement. On voit également que pour les tablettes, iOS domine avec 74,49 % contre Android qui possède 25 % des parts. Il est donc préférable pour les développeurs d'applications mobiles de ne pas se contenter d'une plateforme comme Android, iOS ou un autre système d'exploitation avec un faible pourcentage d'utilisateurs, mais plutôt de viser un développement multiplateforme pour toucher un maximum d'utilisateurs.

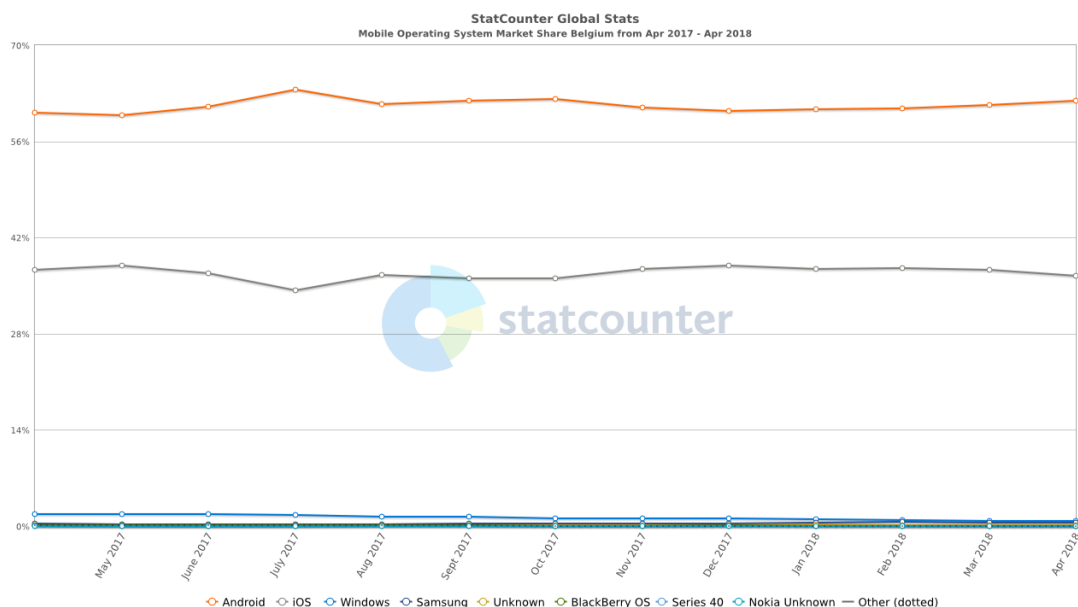


FIGURE 1 – Évolution des parts de marché des systèmes d'exploitation des appareils mobiles

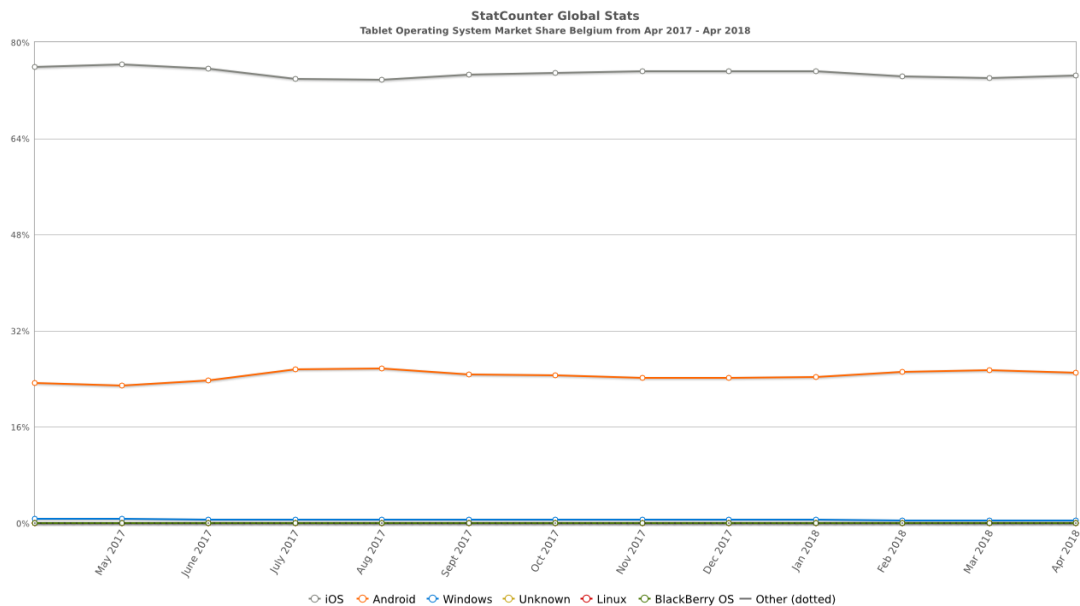


FIGURE 2 – Évolution des parts de marché des systèmes d’exploitation des tablettes

On qualifie de natif le développement d’une application pour une plateforme spécifique dans un langage spécifique. Par exemple, les applications Android sont développées en Java et celles sur iOS en Objective-C et plus récemment en Swift. Ces dernières années, le développement multiplateforme connaît une croissance avec de nombreux outils et frameworks qui ont été créés dans ce but. Il existe 3 possibilités majeures[15] :

- **Les webApps** : cela consiste à faire son application sous la forme d’une application accessible depuis un navigateur. Généralement, on utilise les technologies HTML5[16], CSS et JavaScript, mais il existe aussi des webApps qui ne sont qu’un site web responsive (qui s’adapte à la taille de l’écran utilisé).
- **Les applications hybrides** : ce sont la combinaison des webApps avec des éléments natifs aux différents systèmes tels que GPS, appareil photo, messages, etc.. Plusieurs outils existent afin de créer une application hybride, telle que Cordova[17] et PhoneGap[18] ou même Ionic[19][20]. Ceux-ci facilitent la création d’application basique dont on voudrait utiliser sur les différents supports mobiles. Il est à noter que les applications hybrides n’ont accès qu’à certains éléments natifs ce qui restreint la création d’applications nécessitant un développement plus poussé.
- **Le développement cross-plateforme** : pour ce type de développement, il faut utiliser l’outil Xamarin[21]. Avec celui-ci, l’application est développée entièrement en C# et sans utiliser de webApps. Les fonctionnalités natives de l’appareil sont elles aussi gérées grâce au code C#. Il permet un développement plus poussé qu’avec une application hybride ou une webApp et on disposera de plusieurs bibliothèques Microsoft (packages NuGET, etc..)

L’application UCLCampus est une application multiplateforme hybride. Le framework Ionic (dans sa version 3) et Cordova sont utilisés. Les points suivants portent sur une analyse séparée de leur fonctionnement et des différentes technologies telles que Angular, TypeScript, HTML5.

## 2.1 Ionic

Ionic est un SDK (Software Development Kit), complètement Open Source sous licence MIT, pour le développement d’application mobile hybride. C’est une surcouche d’Apache Cordova et AngularJS qui est remplacé par Angular[22][23] dans Ionic3. Ionic propose des outils et services

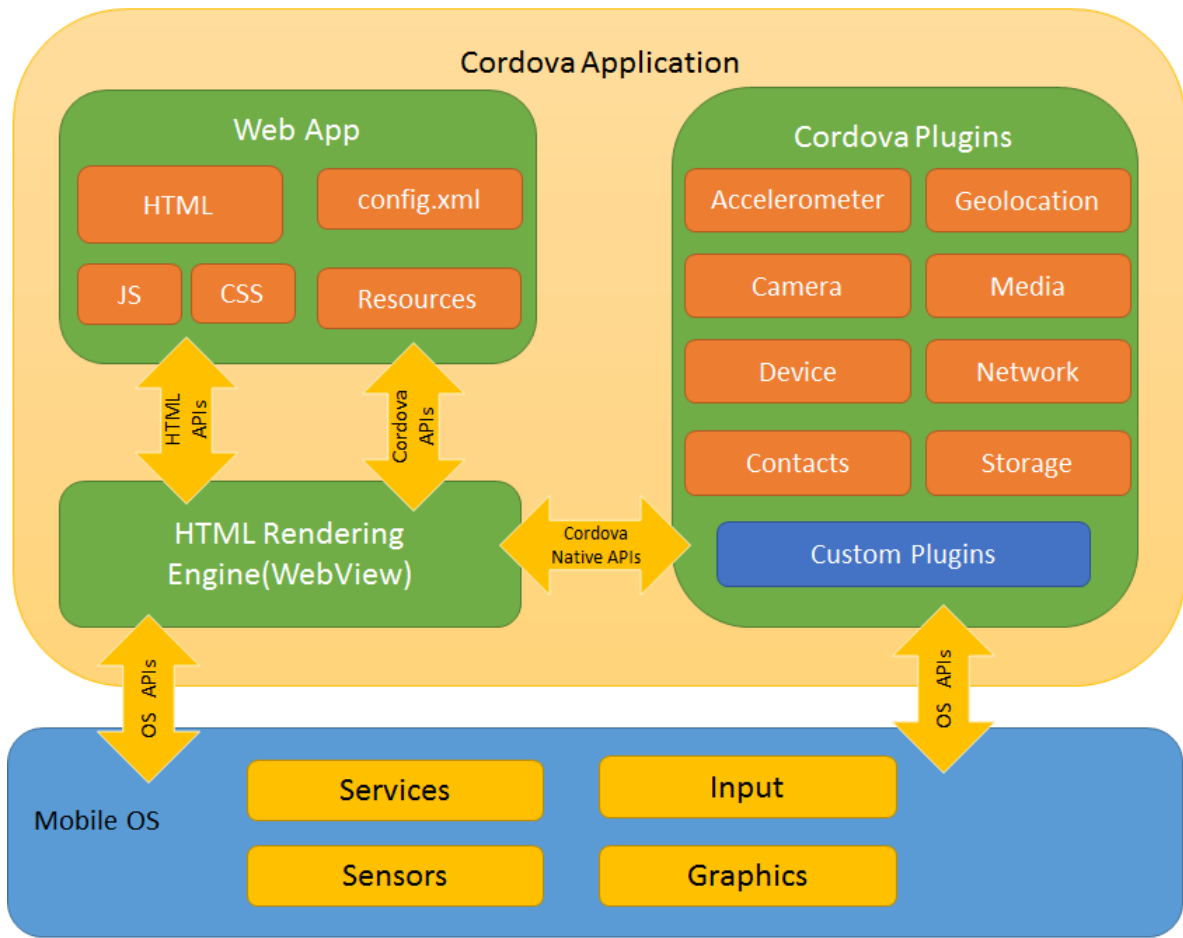


FIGURE 3 – Représentation de l'architecture d'une application et des interactions entre les composants avec le framework Ionic 3 [1]

utilisant des technologies web telles que CSS, Sass et HTML5. Les applications sont créées grâce à ces technologies et peuvent être déployées sur les stores afin d'être installées sur les appareils en tirant parti de Cordova. Le but d'Ionic[24] est de combler l'écart entre le développement d'applications natives et HTML5. Ionic n'est pas uniquement du balisage bien documenté et du CSS (Cascading Style Sheet), il est également un modèle de conception JavaScript pour faciliter la réalisation d'applications plus poussées avec des concepts similaires à Android et iOS. Ionic veut libérer les applications hybrides de leur barre d'URL et évoluer vers des concepts de View Controller plus génériques et plus puissants.

Les fonctionnalités natives sont exécutées grâce aux plugins de Cordova alors que la partie web et interface utilisateur est gérée par Angular. Sur la figure 3[1], on peut voir l'architecture et les interactions entre les différents composants : une instance de navigateur (la webView) affiche l'interface utilisateur grâce à Angular et utilise un wrapper avec le framework Cordova pour utiliser les fonctionnalités natives liées à l'appareil et à la plateforme.

Ionic propose également divers outils et fonctionnalités utiles au développement d'applications mobiles.

L'outil Ionic CLI (Command Line Interface) permet par exemple au développeur d'utiliser certaines commandes pour faciliter son travail :

- **ionic start nameProject** : crée un dossier de projet Ionic portant le nom nameProject avec toute l'architecture et la structure déjà prêtes ;
- **ionic serve** : lance l'application dans le navigateur et permet à l'application de se recharger automatiquement dès qu'un fichier est modifié. Ainsi, on évite de recompiler à

- chaque fois tout le projet et de revenir à la page voulue à chaque modification ;
- **ionic cordova run ios** ou **ionic cordova run android** : enveloppe le CLI de Cordova et permet de lancer un émulateur (iOS ou Android) pour tester l'application. Cela permet aussi de transférer l'application sur un mobile avec l'OS correspondant.

## 2.2 Cordova

Apache Cordova est un framework de développement mobile Open Source sous licence Apache développé par la fondation Apache. Il était anciennement appelé PhoneGap. Il permet d'utiliser les technologies Web standard telles que HTML5, CSS et JavaScript pour le développement multiplateforme au lieu de s'appuyer sur des API spécifiques à la plate-forme comme celles d'Android ou iOS. Les applications s'exécutent dans des « wrapper » ciblés pour chaque plate-forme et s'appuient sur des API pour accéder aux fonctionnalités natives de chaque périphérique tels que les capteurs (accéléromètre, GPS...), les données, l'état du réseau, etc. Concrètement, Cordova crée une application qui ouvre une WebView native sur l'appareil. Cette WebView est basée sur Chromium pour Android et sur Safari pour iOS. Elle est en fait une fenêtre de navigateur Web lancée dans l'application qui va interpréter et afficher le contenu de notre code. Cordova étend cette WebView avec certaines fonctionnalités natives à l'appareil.

Cordova maintient un ensemble de plugins appelés Core Plugins qui fournissent des accès aux capacités de l'appareil comme dit précédemment, mais il existe également plusieurs plugins tiers qui fournissent des accès supplémentaires qui ne sont pas nécessairement disponibles sur toutes les plates-formes.

## 2.3 Angular

Angular est une plateforme d'application web Open Source sous licence MIT basée sur TypeScript[25][26]. C'est une réécriture complète d'AngularJS qui lui était basé sur JavaScript. Ce projet est dirigé par l'équipe Angular de Google et par une communauté de particuliers et d'entreprises. L'architecture est basée sur les composants et les modules. Comme on peut le voir sur la figure 3[1], Angular utilise trois technologies de développement : HTML5, CSS et le langage TypeScript. Celles-ci sont présentées dans les points suivants.

### Fonctionnement

Angular[27] se base sur les modules appelés NgModules. Ils fournissent un contexte de compilation pour les composants.

Organiser son code en modules fonctionnels distincts aide à gérer le développement d'applications complexes et à tirer parti du « chargement paresseux » : le chargement des modules à la demande ce qui minimise la quantité de code à charger au démarrage.

Une application Angular a au moins un NgModule, le module racine qui s'appelle AppModule et réside dans un fichier nommé app.module.ts.

Un NgModule est défini comme une classe avec le décorateur @NgModule. Ce décorateur prend les métadonnées dont les propriétés décrivent le module. Les propriétés principales sont :

- **declarations** : les composants, directives et « pipes » appartenant à ce NgModule
- **exports** : sous-ensemble de déclarations qui doivent être visibles et utilisables dans d'autres modules
- **imports** : autres modules dont les classes exportées sont requises dans ce NgModule
- **providers** : créateur de services auxquels ce NgModule contribue
- **bootstrap** : vue principale (appelée composant racine). Elle héberge toutes les autres vues de l'application. Uniquement le module racine doit définir cette propriété.

Voici une définition simple de NgModule racine :

```
@NgModule( {
```

```

imports :      [ BrowserModule ],
providers :    [ Logger ],
declarations: [ AppComponent ],
exports:      [ AppComponent ],
bootstrap:    [ AppComponent ]
})
export class AppModule { }

```

Dans un module, plusieurs éléments interagissent entre eux comme nous pouvons le voir sur la figure 4[27]

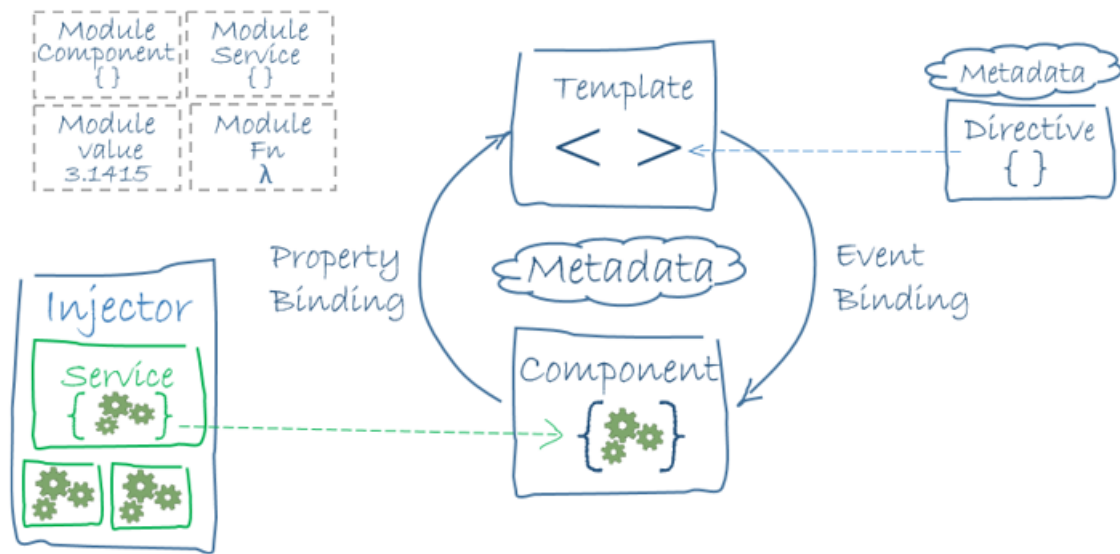


FIGURE 4 – Composition d'un module

- **Les composants** définissent et contrôlent des vues. Celles-ci sont des ensembles d'éléments visuels que Angular peut choisir et modifier en fonction de la logique et des données de l'application. Angular crée, met à jour et détruit des composants lorsque l'utilisateur se déplace dans l'application. Les composants d'une application définissent généralement plusieurs vues, organisées hiérarchiquement. Les composants utilisent également des services qui fournissent des fonctionnalités spécifiques qui ne sont pas directement liées aux vues. Les fournisseurs (ou providers) de services peuvent être injectés dans des composants tels des dépendances. Cela permet de rendre le code modulaire, réutilisable et efficace. Les composants utilisent le décorateur Component qui les identifie comme la classe Component et spécifie leurs métadonnées.
- **Les métadonnées** d'un composant indiquent à Angular où trouver les principaux blocs de construction dont il a besoin pour créer et présenter le composant et sa vue. Il associe un template au composant. Ensemble le composant et son template définissent une vue. Elles configurent également comment le composant peut être référencé en HTML et quels services il requiert.
- **Les décorateurs** sont des fonctions qui modifient les classes. Angular en définit un certain nombre, ils lient des types spécifiques de métadonnées aux classes pour savoir ce qu'elles signifient et comment elles devraient fonctionner.
- **Les templates** définissent la vue d'un composant. Il est écrit avec du HTML classique et il combine celui-ci avec les directives et le balisage d'Angular afin de modifier le HTML avant de produire le rendu pour l'affichage.
- **Le data binding** est responsable de « pousser » et afficher les valeurs des données dans les contrôleurs HTML ainsi que de transformer les réponses des utilisateurs en actions. La

mise à jour de valeurs est un travail fastidieux. C'est pour cela que le framework Angular s'en occupe. Angular supporte la liaison de données bidirectionnelles (ou Two-Way Data Binding) : c'est un mécanisme permettant de coordonner les parties d'un template avec les parties d'un composant. La figure 5[27] montre les quatre formes de data binding :

- L'interpolation d'une valeur du composant pour l'afficher.
- Le property binding : le passage d'une valeur du composant à une propriété d'un éventuel composant enfant.
- L'event binding : cela appellera la méthode handler lorsque l'événement se sera produit.
- Le two-way data binding : cela combine le property binding et l'event binding en un seul. Les actions et modifications faites par l'utilisateur ont des répercussions directes sur le composant. Les changements lors de la mise à jour du composant sont directement affichés sur le document HTML.

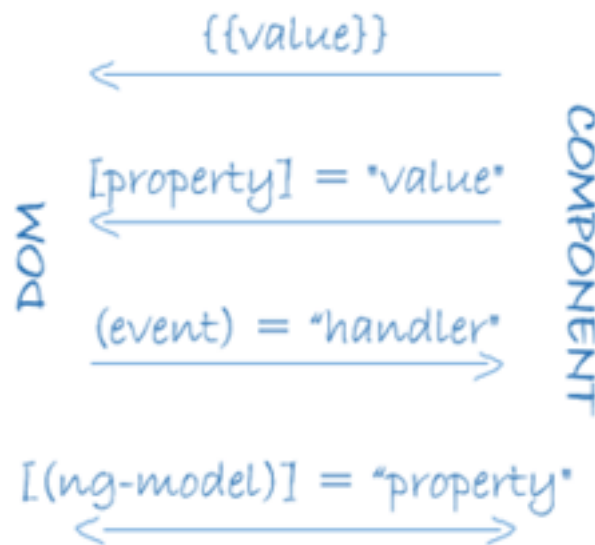


FIGURE 5 – Les différentes formes de data binding

- **Les directives** : lorsqu'Angular interprète les templates, qui sont d'ailleurs tous dynamiques, il transforme le DOM (qui sera défini plus bas) selon les instructions données par les directives. Une directive a le décorateur Directive. Il existe deux types de directives :
  - Les directives structurelles : elles modifient la mise en page en ajoutant, supprimant et remplaçant des éléments du DOM. Il y a par exemple `*ngIf` et `*ngFor`.
  - Les directives d'attribut : elles modifient l'apparence ou le comportement d'un élément existant. Dans les templates, elles ressemblent à des attributs HTML normaux. Il y a par exemple la directive `ngModel` qui implémente le two-way data binding.
- **Les services** sont une vaste catégorie englobant toute valeur, fonction ou fonctionnalité dont une application a besoin. Un service est généralement une classe avec un objectif bien défini. Il doit faire quelque chose de spécifique et le faire bien. Les services utilisent le décorateur `Injectable`.
- **L'injection de dépendances** Les composants consomment des services : il est possible d'injecter un service dans un composant lui donnant ainsi l'accès à cette classe du service. L'injecteur est le mécanisme principal. Angular crée cet injecteur à l'échelle de l'application pendant le processus de lancement de l'application. Il gère un conteneur d'instances de dépendance qu'il a déjà créé et les réutilise si possible. Lorsque Angular découvre qu'un composant dépend d'un service, il vérifie d'abord si l'injecteur a déjà des instances existantes de ce service. Si ce n'est pas le cas, l'injecteur en crée une en utilisant le provider enregistré et l'ajoute à l'injecteur avant de renvoyer le service à Angular. Lorsque tous les services requis ont été résolus et renvoyés, Angular peut appeler le constructeur du

composant avec ces services en arguments. Dans Angular, les providers sont des éléments qui gèrent et fournissent les services.

- **Le DOM**[28] (ou Document Object Model) est une API permettant d'accéder et de manipuler des documents (HTML et XML). Une page web est un ensemble de balises imbriquées les unes dans les autres qui peut se représenter sous forme d'arborescence. Il y a trois catégories principales du DOM :
  - Le document (balise `<html>`) contient les informations de l'ensemble du document HTML
  - Les éléments sont des balises internes au document. Ils ont des attributs et comportent des méthodes permettant d'accéder à leur élément parent, enfant et/ou frère de l'arbre du DOM
  - Le texte qui se trouve à l'intérieur d'une balise

## TypeScript

TypeScript est un langage de programmation Open Source développé et maintenu par Microsoft. C'est un surensemble syntaxique strict de JavaScript. TypeScript est conçu pour le développement de grandes applications et transpile vers JavaScript. Comme TypeScript est un surensemble de JavaScript, les programmes JavaScript existants sont également des programmes TypeScript valides. TypeScript peut être utilisé pour développer des applications JavaScript pour l'exécution côté client ainsi que le côté serveur (Node.js). TypeScript ajoute une programmation orientée objet classique, la notion de classes et d'interfaces à Javascript. Contrairement à Javascript qui utilise un typage dynamique, TypeScript utilise un typage statique. Ces ajouts permettent une meilleure lisibilité et structure du code ainsi qu'un meilleur retour sur les erreurs car le typage est vérifié lors de la compilation.

## HTML5

HTML5 (HyperText Markup Language 5) est la dernière révision majeure du HTML (format de données conçu pour représenter les pages web) qui a été finalisée en 2014. Ce langage comprend de nombreuses API (Application Programming Interface) et HTML5 en ajoute de nouvelles. Cette version ajoute également de nouvelles balises, éléments et attributs. De plus, il permet de traiter des documents avec une syntaxe non conforme grâce à un algorithme intégré. Le HTML est un langage de balisage et sert à représenter des pages web. Il permet de structurer et de mettre en forme le contenu de ces pages, d'y ajouter des formulaires, des images. On l'utilise souvent avec des feuilles de style CSS et du langage de programmation JavaScript.

## CSS

Les CSS (Cascading Style Sheets) ou les feuilles de style en cascade sont un langage qui définit la présentation visuelle des documents HTML et XML. Les standards définissant les CSS ont été publiés par le W3C (World Wide Web Consortium).

## 3 Enterprise Service Bus

Dans cette partie, nous commencerons par présenter le concept d'ESB, de sa composition ainsi que les idées fondamentales qui s'y attachent. Dans une seconde partie, il sera présenté un ESB en particulier : celui pour lequel l'Université Catholique de Louvain a opté : l'ESB WSO2. Il sera également développé l'état d'avancement de sa mise en place ESB à l'UCL ainsi que l'objectif final à atteindre.

### 3.1 Cadre théorique

L'Entreprise Service Bus (ESB)[29] est l'approche la plus prometteuse pour l'intégration d'applications d'entreprise (EAI) ces dernières années. C'est une architecture orientée services (SOA) qui intègre de manière itérative toutes sortes d'applications isolées dans une infrastructure décentralisée. Celle-ci combine les meilleures pratiques d'EAI, de middleware orienté message (MOM) et de services (web).

Roy Schulte, analyste de Gartner Inc., le définissait en 2003 : «Un Enterprise Service Bus (ESB) est une nouvelle architecture qui exploite les services web, les systèmes orientés messages, le routage intelligent et la transformation. L'ESB agit comme une colonne vertébrale légère et omniprésente de l'intégration à travers laquelle les services logiciels et les composants applicatifs dialoguent».

Les ESB ont des caractéristiques fondamentales[30] :

- **Connectivité et communication** : il doit permettre de communiquer avec de multiples protocoles et formats basés sur des standards reconnus par tous et de manière synchrone ou asynchrone. Pour cela, il doit être capable d'interagir avec un Middleware Orienté Message embarqué ou non au sein de l'ESB.
- **Routage** : il doit permettre de découpler fournisseurs et consommateurs de données et ainsi mettre à disposition un mécanisme permettant de distribuer les messages depuis un fournisseur vers un ou plusieurs consommateurs de manière transparente pour l'un et l'autre.
- **Transformation** : il doit mettre à disposition des moyens de transformation des messages car un fournisseur et un consommateur ne dialoguent pas forcément dans un même langage.
- **Orchestration** : il doit permettre aux données de circuler sur toutes les applications et services qui sont branchés sur le bus, qu'ils soient locaux ou distants.
- **Outil de surveillance** :
  1. Technique : les ESB doivent permettre de surveiller l'acheminement des messages et de fournir un ensemble métrique technique se rapportant aux flux et à la plateforme (Service Activity Monitoring : SAM).
  2. Métier<sup>3</sup> : certains ESB vont permettre de réaliser un contrôle fonctionnel et métier des messages qui transitent (Business Activity Monitoring : BAM)

Un ESB se compose de :

- **Un framework d'intégration** qui réalise les opérations de médiation sur les flux de messages.
- **Un framework de services** qui permet de concevoir les web services qui pourront être utilisés par les applications du SI.
- **Un système de gestion de composants** qui permet de gérer dynamiquement l'installation, le démarrage, l'arrêt et la désinstallation de composants techniques.
- **Un serveur d'application** qui est un conteneur de contexte d'exécution pour applications qui va gérer toutes les connexions.
- **Un Middleware Orienté Message** qui permet de gérer des files d'attente de messages dans le cadre d'échanges asynchrones.
- **Un registre** qui permet de stocker et gérer des informations contextuelles nécessaires aux composants pour fonctionner.

### 3.2 Utilisation à l'UCL avec WSO2

WSO2[29][31] est basé sur des composants 100% Open Source sous licence Apache2. Il se repose sur la plateforme Carbon qui possède de nombreux composants tels que l'interface d'administration, la gestion des logs ou la gestion du déploiement des configurations. Il utilise les

---

3. Un processus métier désigne un ensemble d'activités en interaction qui contribue aux finalités des affaires d'une organisation



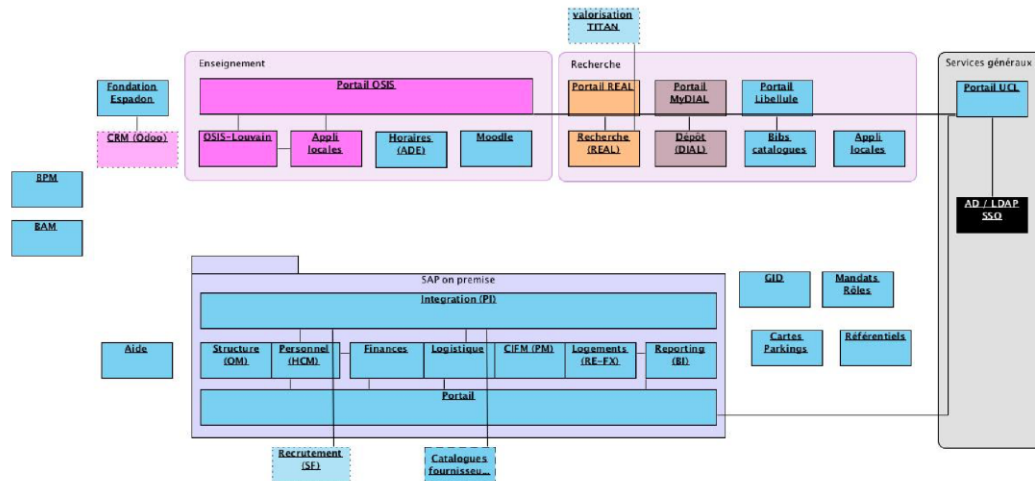


FIGURE 6 – Infrastructure informatique de l’UCL

projets de l’ESB Apache Synapse pour la composante de médiation et de routage ainsi qu’Apache Axis2 pour les webservices (dont l’éditeur est le créateur et le principal contributeur). L’ESB WSO2 est rapide, a une empreinte mémoire réduite, est hautement interopérable et supporte de nombreux transports, formats et protocoles. WSO2 ESB n’est qu’un produit parmi une plateforme middleware complète orientée SOA nommée WSO2. Une série de produits dédiés aux mobiles y ont été ajoutés. Grâce à Apache Axis2, l’ESB permet d’exposer des webservices en SOAP (Simple Object Access Protocol) et en REST (Representational State Transfer). WSO2 ESB offre la possibilité de sécuriser ses flux en utilisant Apache Rampart et il met à disposition des médiateurs Oauth2 et LDAP (Lightweight Directory Access Protocol) pour authentifier les messages dans un autre format.

L’Université Catholique de Louvain est une organisation qui a une infrastructure possédant de nombreux services informatiques afin d’automatiser un maximum de tâches utiles aux étudiants et aux membres du personnel. Cette infrastructure est visible sur la figure 6[32][33] qui met en évidence les différents services de l’infrastructure informatique qui permettent la gestion de l’UCL.

Afin d’éviter une paralysie du système informatique, la mise en place de l’ESB WSO2 à l’UCL se fait progressivement. La figure 7[32][33] montre l’état initial où l’on peut voir qu’il existe énormément d’interactions entre les différents services. Sur la figure 8[32][33], on peut observer l’avancement de cette mise en place en 2017. L’objectif étant de faire passer toutes les communications entre les différents services par l’ESB et d’atteindre l’état présenté à la figure 12[32][33]. Il existe des applications externes et internes au système pour ces services. Ils sont accessibles grâce à une autorisation d’accès à l’ESB ou via un API Manager qui propose une interface graphique afin de gérer et visualiser les différentes API. L’application UCLCampus utilise ces services via des API de l’API manager (Répertoire UCL, programme de cours d’un étudiant et les bibliothèques).

## 4 Applications UCL

Shibboleth et OSIS sont les noms de deux applications, impliquées dans le cadre informatique de l’UCL, jouant un rôle dans le développement de l’application UCLCampus. Shibboleth est une application qui assure l’authentification des utilisateurs voulant avoir accès aux services fournis par l’UCL. OSIS, sous le nom complet Open Student Information System, permet quant à elle aux étudiants de l’Université catholique de Louvain de s’inscrire aux cours, aux examens et

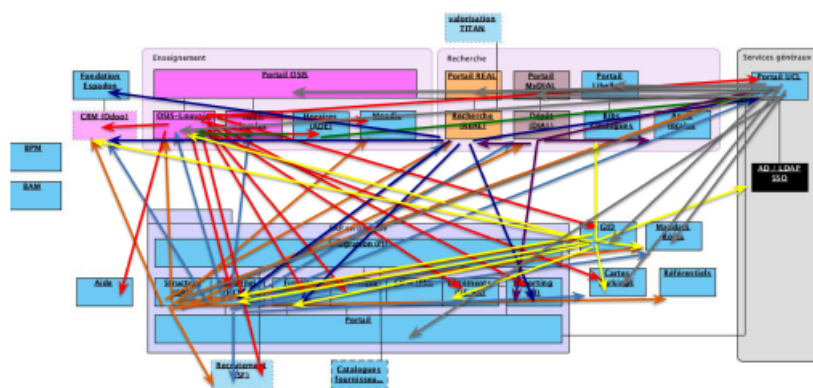


FIGURE 7 – État initial des interactions entre les services de l’UCL

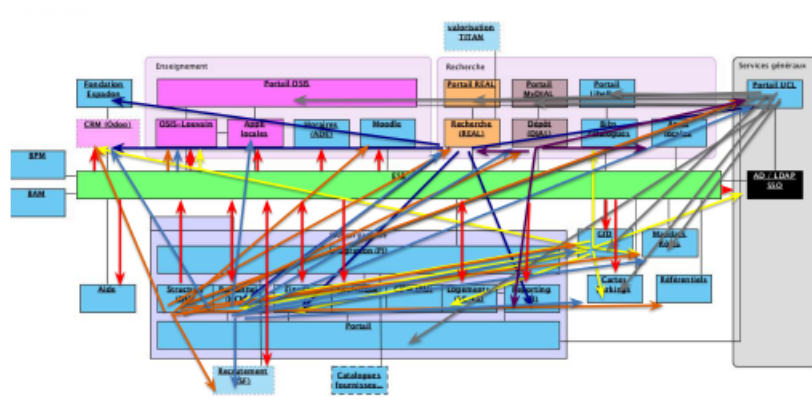


FIGURE 8 – État des interactions entre les services de l’UCL en 2017

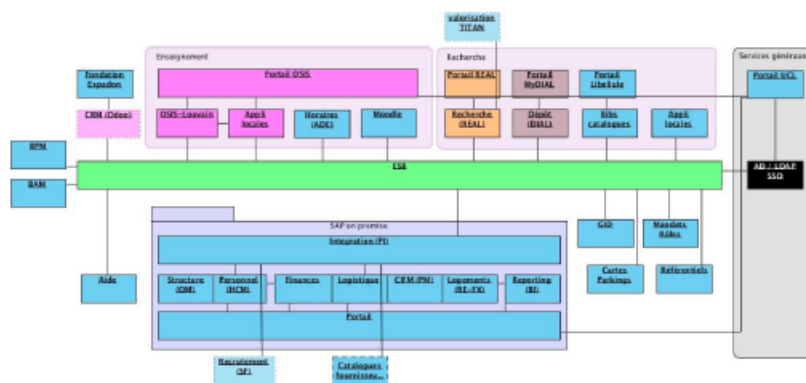


FIGURE 9 – État final des interactions entre les services de l’UCL

d'accéder à leurs points d'examens. Cette section a pour but d'étudier ces deux applications plus précisément indépendamment de l'application UCLCampus.

## 4.1 Shibboleth

Shibboleth est un mécanisme de fédération d'identités qui a été développé par le consortium Internet2, mais qui est aujourd'hui sous la responsabilité du consortium Shibboleth. Il est développé en tant que logiciel Open Source et est publié sous la licence Apache Software License. Ce sont principalement des établissements universitaires ou de recherche qui utilisent ce mécanisme. Le principe de la fédération d'identités est de déléguer l'authentification unique (SSO - Single Sign-On) sur le Web des utilisateurs à un service d'authentification dans l'organisme d'origine et de l'utilisateur.

Ce processus d'authentification unique va permettre aux utilisateurs de se connecter à un ensemble de services et d'applications en ayant besoin que d'une seule authentification à l'aide de leur identifiant et de leur mot de passe. Cela permet d'avoir un contrôle d'accès à des services et applications indépendantes entre elles, mais faisant partie d'un même système. L'utilisateur ne devra pas rentrer ses identifiants s'il désire accéder à un service différent appartenant au même système.

### Composants de Shibboleth

Les deux composants importants de Shibboleth sont le fournisseur de services et fournisseur d'identité.

Le fournisseur de services est un module d'authentification pour le serveur Web.

Il permet de :

- déléguer l'authentification des utilisateurs à un fournisseur d'identités ;
- transmettre le profil utilisateur ;
- gérer le contrôle d'accès de manière optionnelle.

Le fournisseur d'identité est une application répondant à la requête d'un fournisseur de services par la gestion de l'authentification des utilisateurs. L'authentification peut-être déléguée à un serveur, elle se fait à l'aide d'identifiant/mot de passe. Les attributs liés à l'utilisateur sont alors extraits d'un annuaire, d'une base SQL ou calculés pour ensuite être propagés au fournisseur de services.[34]

Pour fournir un cadre fédéré d'authentification unique et d'échange d'attributs, Shibboleth implémente des standards d'identités fédérées très répandus, tels que OASIS SAML. Un utilisateur s'authentifie avec ses identifiants et le fournisseur d'identité transmet les informations d'identité minimales nécessaires au fournisseur de services afin de permettre une décision d'autorisation. Shibboleth offre également une fonctionnalité de confidentialité étendue qui permet à l'utilisateur et à son site d'origine de contrôler les attributs libérés pour chaque application.[35]

### Fonctionnement

Afin d'expliquer le fonctionnement de Shibboleth, il faut connaître les éléments principaux d'un système de SSO basé sur le Web.

Ces éléments sont les suivants :

- **Navigateur Web** : représente l'utilisateur dans le processus SSO ;
- **Ressource** : contient du contenu à accès restreint que l'utilisateur désire ;
- **Fournisseur de services** : exécute le processus SSO pour la ressource ;
- **Fournisseur d'identité** : authentifie l'utilisateur.

Une fois ces éléments listés et expliqués, il est plus facile de comprendre comment fonctionne Shibboleth.

Cela se résume en 6 étapes : [35]

1. **L'utilisateur accède à la ressource :** L'utilisateur essaye d'accéder à la ressource protégée. Le moniteur de ressources détermine si l'utilisateur possède une session active et si ce n'est pas le cas le dirige vers le fournisseur de services afin de démarrer le processus de SSO.
2. **Le fournisseur de services émet une demande d'authentification :** L'utilisateur arrive chez le fournisseur de services. Celui-ci prépare une demande d'authentification et l'envoie avec l'utilisateur au fournisseur d'identité.
3. **Authentification de l'utilisateur chez le fournisseur d'identité :** Lorsque l'utilisateur arrive chez le fournisseur d'identité, il vérifie si l'utilisateur dispose d'une session existante avant de passer à l'étape suivante. Si ce n'est pas le cas, le fournisseur d'identité l'authentifie (par exemple en demandant et en vérifiant un nom d'utilisateur et un mot de passe) et l'utilisateur passe à l'étape suivante.
4. **Le fournisseur d'identité émet une réponse d'authentification :** Après avoir identifié l'utilisateur, le fournisseur d'identité prépare une réponse d'authentification et l'envoie avec l'utilisateur au fournisseur de services.
5. **Le fournisseur de services vérifie la réponse d'authentification :** Lorsque l'utilisateur arrive avec la réponse du fournisseur d'identité, le fournisseur de services valide la réponse, crée une session pour l'utilisateur et met à la disposition de la ressource protégée certains renseignements extraits de la réponse (par exemple l'identificateur de l'utilisateur). Ensuite, l'utilisateur est envoyé à la ressource.
6. **La ressource renvoie du contenu :** Comme à l'étape 1, l'utilisateur essaye d'accéder à la ressource protégée, mais cette fois avec une session identifiée. Avec ces informations, la ressource répondra à la requête de l'utilisateur et renverra les données demandées.

## 4.2 OSIS

OSIS est un système léger et évolutif visant à gérer au mieux le coeur du métier des établissements d'enseignement supérieur. Ce système résulte d'un projet financé par l'Université Catholique de Louvain dans le cadre de son objectif à long terme de la modernisation de ses systèmes d'information ciblant la future génération d'étudiants, de dispositifs, de pratiques et de technologie. Ce projet est libre et Open Source sous les termes de la licence GNU GPL v3. La philosophie attachée à ce projet est en adéquation avec l'esprit de partage des connaissances et de collaboration de la communauté que l'on peut retrouver dans les projets Open Source. Le framework Django 1.9 est utilisé, manipulant Python3.4 et HTML5, afin d'encourager un développement rapide avec un design propre et pragmatique du projet OSIS.

Les avantages d'offrir une solution Open Source pour ce système sont les suivants :

- sa réutilisation par d'autres établissements ;
- des opérations totalement transparentes pour l'ensemble de la communauté académique ;
- l'inspiration pour les entrepreneurs afin de créer des services intégrés qui pourraient être incorporés dans le système OSIS ;
- possibilité pour les étudiants de participer à un projet réel ;
- une réduction des coûts de maintenance pour l'Université grâce à la gamme de produits Open Source fiable requise par le système, tels que le serveur web Apache, la base de données PostgreSQL et le framework web Django.

Pour l'instant, OSIS est conçu sur base des besoins auxquels l'UCL essaye de répondre, mais a pour but futur de pouvoir être adapté afin de répondre aux besoins d'autres institutions attirées par le projet. L'utilisation d'OSIS a pour but de permettre à un établissement d'enseignement supérieur de maintenir, contrôler et suivre l'ensemble de la carrière de l'étudiant, de l'admission au diplôme et toutes les informations liées à ce processus fondamental tel que la planification de l'année académique, la gestion des partenariats avec d'autres institutions pour les programmes d'échange et de mobilité, l'affectation des étudiants aux cours et des cours aux étudiants, etc.[36]

Conceptuellement, le projet OSIS est scindé en deux entités distinctes et indépendantes, celles-ci sont nommées OSIS et OSIS portal. Ces deux entités peuvent fonctionner l'une sans l'autre sans aucun souci, elles possèdent leur propre base de données logique sur un serveur physique partagé. Le système OSIS dans son entièreté fait partie intégrante de la nouvelle infrastructure mise en place à l'UCL par l'interaction avec les autres services informatiques via l'ESB WSO2. Une troisième entité partagée par ces deux entités entre également en jeu, elle se nomme OSIS commun et fournit des fonctionnalités communes aux deux. OSIS portal est l'entité du système permettant aux étudiants d'interagir avec le système, un exemple est l'inscription aux cours et examens après l'authentification, alors que OSIS permettra, quant à elle, aux services administratifs de gérer la partie académique de l'établissement. La communication entre les bases de données des deux entités indépendantes est gérée par le mécanisme RabbitMQ qui implémente le protocole Advanced Message Queuing. Ce mécanisme est essentiel au découplage des modules OSIS et OSIS portal. Il permet aux modules de recevoir et manipuler des messages depuis différents canaux sans en connaître la source. Le canal par lequel le message est réceptionné influe sur l'action effectuée avec ce message. Une action spécifique indiquée par son nom est associée à chaque canal. La communication entre les deux modules est asynchrone. Un module envoie un message sur un canal à l'aide de RabbitMQ et peut ensuite passer à une autre activité. Le module destinataire quant à lui doit vérifier régulièrement chacun des canaux pour récupérer les messages reçus et agir en fonction. Les messages peuvent s'accumuler en attendant d'être réceptionnés. Cependant, la cohérence et la consistance de l'état des données sont assurées par la synchronisation des actions internes aux modules. Cette communication asynchrone se fait dans les deux sens pour OSIS et OSIS portal, le découplage lié à l'utilisation de RabbitMQ rend les modules indépendants, permettant une transition plus sûre des technologies. Vu que OSIS portal relève juste les messages en provenance d'un canal et qu'il ne nécessite pas de savoir d'où il vient, le changement de système lui envoyant les messages ne posera aucun problème.[32]

## 5 Portail UCL, une source d'information

Le portail géré par l'Université Catholique de Louvain contient une quantité d'informations prépondérantes qui lui sont relatives quant à sa vie et à ses informations brutes. Il est vu comme la façade de l'Université sur internet, c'est un outil de communication. Mais pas seulement, c'est aussi un outil permettant à ses membres, étudiants et personnels, d'accéder à leur compte UCL afin de pouvoir consulter et mettre à jour leurs informations personnelles ainsi que d'avoir accès à des informations ciblées.

Toutes les entités de l'Université, que cela soit ses facultés ou ses services, ont leurs pages personnelles qu'elles alimenteront à leur guise en contenu.

Le système de gestion de contenu Drupal est utilisé afin de gérer, organiser, structurer et mettre à jour l'ensemble des contenus et informations. Drupal est développé sous licence Open Source GNU GPL-v2 et écrit en langage PHP. Sa communauté est importante et active, elle développe de nouveaux modules qui ajoutent de nouvelles fonctionnalités utiles pour ses utilisateurs.

Les principales caractéristiques d'un tel système sont : [37][38]

- **Mise en place simplifiée** : la mise en place du site web peut se faire sans connaissance en programmation, tout se fait à travers des interfaces graphiques ;
- **Edition WYSIWYG des pages** : What You See Is What You Get, la mise en forme des pages webs se fait à l'aide d'une interface graphique, pas besoin d'écrire soi-même le HTML que l'on veut afficher aux utilisateurs, celui-ci est généré par le CMS ;
- **Séparation entre contenu et design** : sur un site web un bon nombre de pages ont un design semblable, le CMS intègre des outils pour appliquer un même design à toutes les pages, une fois le design choisi il ne sera pas nécessaire de le refaire à chaque fois. De plus, ça permet de modifier le design du site sans avoir à se soucier du contenu ;
- **Gestion des droits d'accès** : il peut exister des rôles différents selon les types d'utili-

sateurs au sein d'un site web. Un utilisateur peut avoir ou non le droit de faire certaines actions ou d'accéder à certaines pages. Cette possibilité de faire ou non une action est gérée par ce qu'on appelle des droits et la gestion des droits est simple à mettre en oeuvre dans le CMS. Pour cela, il faudra créer des groupes d'utilisateurs et de leur donner ou leur interdire l'accès aux différentes parties du site ;

- **Gestion des droits d'édition** : Les droits d'édition sur les pages du site sont régulés au travers de groupes dont la gestion est assurée par l'Administration des relations extérieures et de la communication. Seuls les éditeurs auront ces droits. Le droit d'éditer certaines pages dépendra du groupe dans lequel l'éditeur se trouve. Cette approche permet d'assigner les éditeurs par service et que chaque service gère de façon autonome ses propres pages du site ;
- **Extensions** : de base, le CMS contient les fonctionnalités de base indispensables, mais il est possible d'augmenter la flexibilité de l'outil en installant des plugins séparément selon les besoins ;
- **Politique de sécurité** : le principe d'authentification unique de ce système permet d'accéder à l'application sans répétition de la saisie de l'identifiant et du mot de passe.

## 6 Application UCLCampus 2016-2017

L'application UCLCampus est en développement depuis 2015, cette section décrit le contexte, le produit réalisé et les choix d'implémentation du dernier projet à partir duquel nous sommes repartis.

### 6.1 Contexte

L'application UCLCampus est un projet étalé sur plusieurs années, il est développé dans la continuité du travail de fin d'études de deux étudiants en master en informatique à l'UCL de l'année 2016-2017. Ils avaient eux-mêmes repris le travail de deux précédents mémorants de 2015-2016. L'application se veut d'être utile pour un étudiant de l'UCL ou éventuellement certains membres du personnel, mais également à une personne tierce qui viendrait sur le campus et voudrait se renseigner sur les différentes activités proposées. Certaines fonctions de l'application sont disponibles publiquement tandis que d'autres sont spécifiques à un étudiant et nécessitent que celui-ci se connecte avec son nom d'utilisateur et son mot de passe.

On peut regrouper les différentes fonctionnalités de l'application en plusieurs catégories :

- **Études** : regroupe les fonctionnalités qui concernent les activités de l'étudiant telles que les horaires des bibliothèques, l'horaire des cours ou les différentes informations de support ainsi que le répertoire ;
- **Campus** : comprend les actualités de l'UCL, les différents événements proposés sur le campus ainsi que les horaires des sports ;
- **Outils** : propose une carte interactive pour situer les différents auditoriums, bibliothèques, locaux d'étude, parkings, etc. Il y a également possibilité de lancer l'application des Restaurants Universitaires et d'autres applications concernant la mobilité (SNCB, NextRide et covoiturage) ;
- **Paramètres** : permet de changer la langue (français ou anglais) et de choisir son campus.

### 6.2 Choix d'implémentation

L'application UCLCampus a été développée avec le framework Ionic en version 3, Angular 4 et Cordova. Le framework Ionic impose l'architecture et la structure de l'application. Le corps de celle-ci est élaboré en réalisant les différents composants et modules qui serviront aux différentes fonctionnalités. Les différents modules ont leur indépendance et ont chacun leur propre fonctionnalité. Il y a deux sortes de modules : les pages qui, via l'interface utilisateur,

gèrent les interactions de celui-ci et les événements. Deuxièmement, les services qui gèrent toute la partie logique et modèle de l'application. Ces derniers sont importés par les pages dont les fonctionnalités nécessitent ces services. Un module peut également utiliser les services d'un autre module en l'important. Un programmeur peut donc développer un module ou en améliorer un existant pour ensuite le faire interagir avec d'autres modules sans en connaître le code et le fonctionnement interne de ceux-ci. En 2016-2017, lors de la reprise du mémoire, les mémorants ont dû faire la transition entre Ionic version 1 qui utilisait AngularJS et Ionic version 3 qui utilise Angular depuis la version 2. Cette transition demandait beaucoup trop d'effort et de trop nombreux changements ce qui les a poussés à repartir de zéro en utilisant cette nouvelle version d'Ionic.

### 6.3 Apports à l'application 2016-2017

Lors de la reprise du projet réalisé, par les précédents mémorants, en 2016-2017, un certain nombre de fonctionnalités désirées n'étaient pas encore implémentées.

Les fonctionnalités manquantes de l'époque sont terminées dans ce mémoire. Les manques de fonctionnalités ont été listés :

- L'inexistence de la partie "Sports", comprenant l'horaire des équipes universitaires et l'horaire des sports communs. Elle n'avait pas été implémentée dû à un manque de contact avec le service des sports ;
- L'impossibilité de choisir un campus autre que celui de Louvain-la-Neuve ;
- La disponibilité de l'application uniquement en français ;
- L'impossibilité pour un étudiant de l'UCL de se connecter avec ses identifiants pour récupérer son programme annuel de cours ;
- L'inexistence du répertoire UCL pour pouvoir rechercher un membre du personnel, l'API n'étant pas disponible à l'époque.

L'application n'a donc pas été déployée sur les stores car certaines de ces fonctionnalités étaient requises pour la rendre déployable et atteindre un maximum d'étudiants, le simple fait de se contenter d'un seul campus n'était pas recevable. De plus, la fonctionnalité "Études" est un élément clé de l'application, sans qu'elle soit implémentée il était inconcevable de la mettre à disposition du grand public, son but premier n'étant pas rempli. Pour être certain que l'application soit utilisée par les étudiants, l'application ne peut être rendue disponible sur les stores seulement si toutes ses fonctionnalités sont implémentées et fonctionnelles.

Troisième partie

Cahier des charges



Ce chapitre est consacré au cahier des charges de l'application actuelle UCLCampus, les exigences relatives à cette application y seront développées. La première section a pour but de décrire le contexte dans lequel le développement de l'application a continué, c'est à dire de l'attente des étudiants d'une application rassemblant les services des campus, de la continuation d'un projet existant et des défis à relever. Ensuite les spécifications fonctionnelles de l'application, du point de vue front-end, back-end et user stories, seront listées et expliquées. Pour finir, une section reprendra les spécifications non fonctionnelles.

## **1 Contexte**

Les étudiants des différents campus de l'Université Catholique de Louvain sont dans l'attente constante d'une application rassemblant les informations utiles et services offerts par différentes entités sur les campus, ceux-ci étant pour l'instant dispersés sur différents sites et applications. L'Université souhaite fournir l'application mobile qui aidera l'étudiant dans sa vie académique ainsi que dans sa vie sur le campus en lui fournissant les informations attendues et services dont il aura besoin. Le développement de cette application a commencé durant l'année 2015, mais n'est pas finalisé, le produit doit être complété avant de pouvoir être délivré à l'ensemble des étudiants. Une étude par les précédents mémorants, au travers d'une enquête, a mis en avant les besoins des étudiants, c'est par ceci que nous allons commencer.

### **1.1 Besoins des étudiants**

Le public cible de l'application est l'ensemble des étudiants se situant sur tous les campus de l'UCL, ceux-ci sont dans une tranche d'âge entre 18 ans et 25 ans. La version développée durant l'année 2016-2017 se cantonnait au campus de Louvain-la-Neuve, les mémorants en chargeant de ce travail voulaient commencer par le gros campus de l'université en terme d'étudiants et d'infrastructures. La version développée pour ce mémoire a élargi la prise en compte des campus à ceux de Woluwé et de Mons.

Pour connaître les besoins des étudiants, les mémorants de l'année académique 2016-2017 ont créé une enquête qu'ils ont distribuée sur le campus de Louvain-la-Neuve. Cette enquête prenait la forme de questions avec un ensemble de réponses proposées, l'étudiant y répondait en cochant les réponses qu'il voulait tout en pouvant développer celles-ci. L'enquête était divisée en trois parties : l'une pour connaître la faculté et niveau d'étude de l'étudiant qui y répondait, une autre interrogeait l'étudiant sur les besoins rencontrés dans le cadre des études universitaires et la dernière concernait la vie sur le campus universitaire. Le questionnaire qu'ils ont distribué se trouve en annexe.

## Résultats de l'enquête

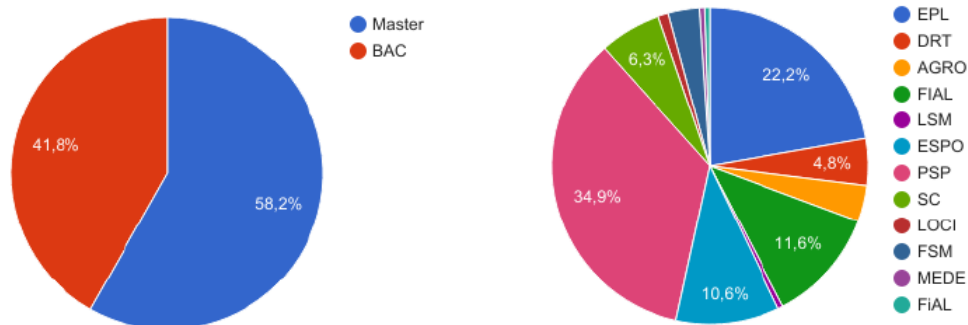


FIGURE 10 – Répartition du niveau d'étude des répondants et répartition des répondants par faculté - Graphique repris des mémorants de 2016-2017

Les résultats de la première partie mettent en avant que les mémorants ont en majorité fait répondre leur entourage étant donné que le public qui a répondu est composé à 58,2% d'étudiants de Master. Cela est intéressant vu leur expérience au sein de la vie sur le campus, de la connaissance du système universitaire et de son fonctionnement, mais malheureusement cette population n'utilisera probablement jamais l'application, le public cible prioritaire représente 41,8%. Pour ce qui en est des facultés, l'EPL (École Polytechnique de Louvain) et la PSP (Faculté de Psychologie et Sciences de l'éducation) représentent plus de 55% des sondés. La prise en compte des besoins différents des étudiants en fonction de leurs études. Certaines facultés sont moins représentées dans ce sondage, cela représente un inconvénient. Selon les facultés, les étudiants utilisent différemment les ressources informatiques disponibles. Par exemple, les valves informatiques ont une importance différente selon les facultés, pour certaines c'est un canal privilégié de communication d'informations vers les étudiants alors que pour d'autres facultés les valves électroniques ne sont pas consultées car aucune information n'y est affichée, l'information circule par mail.

| Fonctionnalités                                                                                           | Taux de demande (%) |
|-----------------------------------------------------------------------------------------------------------|---------------------|
| Agenda reprenant mon horaire de cours et mes auditoires                                                   | 93,6                |
| Notifications en cas de changement d'horaire ou d'auditoire                                               | 88,3                |
| Accès à mes points                                                                                        | 74,5                |
| Accès à moodle                                                                                            | 69,1                |
| Horaire d'ouverture et de fermeture des bibliothèques                                                     | 62,8                |
| Accès à la messagerie universitaire                                                                       | 60,6                |
| Localisation des salles d'étude et des salles informatique                                                | 55,9                |
| Outils de gestion d'achat et de revente de livres en seconde main                                         | 53,7                |
| Connaitre le nombre de places disponibles dans les bibliothèques                                          | 47,9                |
| Forum étudiants permettant de centraliser les questions sur un cours                                      | 38,8                |
| Informations sur la vie universitaire                                                                     | 37,2                |
| Canaux d'interaction en temps réel pour poser des questions aux professeurs lors d'un cours magistral UCL | 29,8                |
| Accès aux informations sur les services offerts par l'UCL                                                 | 26,6                |

FIGURE 11 – Résultats de la question : "Quelles fonctionnalités/outils veux-tu avoir dans le cadre de tes études?" - Tableau repris des mémorants de 2016-2017

| Fonctionnalités                                                                                     | Taux de demande (%) |
|-----------------------------------------------------------------------------------------------------|---------------------|
| Agenda de tous les événements, sorties, soirées, festivités                                         | 70,9                |
| Menu, horaire et prix des restaurants universitaires                                                | 70,9                |
| Partage de bons plans                                                                               | 61                  |
| Horaire du sport CSE + réservation de terrains                                                      | 58,8                |
| Réseau social spécifique à l'UCL (Communauté d'étudiants par centre d'intérêt, sport, musique, ...) | 36,8                |
| Se géolocaliser sur une carte                                                                       | 36,3                |
| Description et géolocalisation des différents KAP's                                                 | 33,5                |
| Description et géolocalisation des différents cercles                                               | 27,5                |
| Inviter des personnes à discuter, partager sa table lors d'un temps de midi                         | 25,3                |
| Création personnelle d'événements pour une communauté, un groupe d'amis                             | 19,2                |

FIGURE 12 – Résultats de la question : Quelles fonctionnalités/outils veux-tu avoir dans le cadre de la vie sur le campus?" - Tableau repris des mémorants de 2016-2017

Les deux tableaux qui précèdent répondent aux parties faisant références aux études universitaires et à la vie sur le campus. Les deux questions posées étaient “Quelles fonctionnalités/outils veux-tu avoir dans le cadre de tes études?” et “Quelles fonctionnalités/outils veux-tu avoir dans le cadre de la vie sur le campus?”, les résultats permettent de mettre en avant dans l'ordre les fonctionnalités que les étudiants désiraient au sein de l'application.

C'est à partir de cette enquête que les étudiants de l'année académique 2016-2017 ont établi les objectifs à atteindre et ont réalisé le prototype de l'application à partir duquel ce mémoire se base. En nous basant sur les objectifs de l'an passé déduits de l'enquête et des nouvelles demandes qui

ont été faites par des entités de l'université nous avons défini les objectifs de l'année académique 2017-2018. L'application mobile UCLCampus doit permettre aux étudiants de gérer et organiser efficacement ainsi qu'intuitivement leur vie au sein des campus universitaires de l'UCL. Pour cela, l'application fait l'interface entre l'agenda du smartphone et les événements auxquels l'étudiant veut participer.[32]

L'application est séparée en trois groupes de services :

1. **Le groupe "Etudes" :** contient tout ce qui permet à l'étudiant de consulter les informations au sujet des bibliothèques (horaires d'ouvertures, numéro de téléphone, adresse, etc.), leur horaire de cours avec possibilité de l'ajouter au calendrier (cours magistraux et TP) et une série d'informations utiles avec également un accès au répertoire du personnel ;
2. **Le groupe "Campus :"** contient l'actualité liée à l'université, la liste des événements se déroulant sur le campus, l'horaire des sports des équipes universitaires et l'horaire des sports communs à tous. Les événements et sports peuvent être ajoutés au calendrier ;
3. **Le groupe "Outils" :** contient un accès à une carte interactive permettant à l'étudiant de se situer ainsi que de situer des lieux sélectionnés, un accès à l'application des restaurants universitaires, un accès aux applications liées à la mobilité (transports en commun, covoiturage) et la campagne Guindaille2.0 d'Univers Santé.

## 1.2 Ressources existantes

Des descriptions succinctes d'applications intéressantes à analyser sont dressées dans cette partie ainsi que la mise en évidence de ressources disponibles utiles à l'application UCLCampus. Elles auront pour but de mettre en avant des fonctionnalités populaires de l'enquête et de donner une idée de leur mise en place.

### Application d'autres universités

Il est intéressant de se baser sur les applications d'autres universités pour connaître ce qui est fait ailleurs. Les applications Quivre et UCL Go! sont utilisées à cette fin.

**Quivr**[39] est l'application mobile de l'université de Leuven pour les étudiants et membres du personnel.

Elle contient un nombre de fonctionnalités intéressantes :

- **Horaire :** c'est la partie la plus utilisée de Quivr, on peut y retrouver son horaire de cours et les locaux dans lesquels les cours se passent. Cet horaire est également disponible hors-ligne. Il peut même être vu comme un agenda étant donné la possibilité d'ajouter ses propres événements et rendez-vous ;
- **Amis :** cette fonctionnalité permet d'ajouter un ami et de voir son calendrier, cela rend les réunions plus simples ;
- **Bibmonitor :** cette fonctionnalité intéressante fournit à l'étudiant le taux d'occupation des différentes bibliothèques, l'étudiant sait immédiatement où il y a de la place pour étudier ;
- **Restaurant :** cette fonctionnalité affiche les heures d'ouverture ainsi que le menu du restaurant ;
- **Actualité :** c'est la partie qui va garder l'étudiant à jour au niveau de ce qui se passe sur le campus, toutes les nouvelles s'y trouvent.

**UCL Go!**[40] est l'application mobile de l'université de Londres pour les étudiants. Elle contient un nombre de fonctionnalités intéressantes :

- **Calendrier :** c'est la partie qui donne accès aux événements de l'horaire en ligne ;
- **Services de bibliothèque :** cette fonctionnalité permet la recherche de livres dans le catalogue, la réservation de livre et son renouvellement ;

- **Cartes du campus** : cette fonctionnalité offre des cartes en haute résolution des principaux campus de Bloomsbury, Archway et Hampstead ;
- **Disponibilité du PC** : cette fonctionnalité donne en temps réel la disponibilité des ordinateurs et des salles de travail sur le campus ;
- **Actualité** : cette fonctionnalité est alimentée par des flux RSS sélectionnables de l'université ;
- **Information transport** : cette fonctionnalité informe en temps réel sur les heures d'arrivée des bus ;
- **Annuaire du personnel** : cette fonctionnalité fournit l'annuaire téléphonique de l'UCL ;
- **Key Info** : cette partie rassemble toutes les informations importantes et utiles de l'université ;

### 1.3 Applications liées aux fonctionnalités

Cette partie reprend une liste d'applications qui implémentent des fonctionnalités désirées pour UCLCampus.

**Resto 4U**[41] est une application développée par Apptree pour les besoins des restaurants universitaires. Elle permet à chacun de :

- Découvrir la carte restauration ;
- Commander depuis l'application ;
- Recevoir les informations pratiques et utiles ;
- Donner son avis grâce aux enquêtes de satisfaction ;
- Epingler ses mets et événements favoris.

**Guindaille2.0**[42] est l'application de l'ASBL Univers Santé. Elle a pour but d'informer les étudiants des dangers de l'alcool et de les rendre responsables. Ceci se fait au travers de pictogrammes de la campagne Guindaille 2.0 et d'accès aux informations d'urgence. Mais ce n'est pas tout, cette application offre également toutes les informations sur les activités extra académiques, ces activités sont des alternatives aux soirées étudiantes classiques et proviennent de Louvainfo.

**LLNMap**[43] est une application développée par Mathieu Zen, assistant à l'UCL et qui a pour fonctionnalité d'offrir la carte de Louvain-la-Neuve à ses utilisateurs. Elle contient un nombre de fonctionnalités intéressantes :

- Carte hors ligne de Louvain-la-Neuve ;
- Marqueur indiquant les différents bâtiments ;
- Recherche par rues ;
- Localisation de l'utilisateur pointée sur la carte ;
- Parcours et temps pour atteindre la destination ;
- Mode boussole et itinéraire.

**Commuty**[44] est une plateforme pour promouvoir une mobilité durable et responsable, elle incite au covoiturage. L'UCL en est cliente. Cette plateforme apporte les fonctionnalités suivantes :

- Recherche de co-voyageur pour effectuer des trajets ;
- Contact simplifié avec ces co-voyageurs à l'aide d'un chat ;
- Remplissage du journal de trajet pour récupérer les avantages fiscaux ;
- Réservation et partage des places de parking de l'entreprise ;
- Organisation pour l'utilisation des vélos en partage.

**NextRide**[45] est l'application donnant accès aux horaires STIB et TEC. Elle contient les fonctionnalités suivantes :

- Info en temps réel des perturbations sur le réseau ;
- Accès à l'horaire sans connexion ;
- Plannificateur d'itinéraire pour comparer visuellement les trajets ;
- Rappels pour ne pas louper son bus.

**SNCB**[46] est l'application officielle de Belgianrail qui permet de faciliter les voyages en train à ses utilisateurs. Elle contient un nombre de fonctionnalités qui aideront les voyageurs :

- Recommandations de voyage sur base de la localisation GSM ;
- Info train en temps réel reprenant les départs/arrivées par gare et par train ;
- Info voie en temps réel ;
- Achat de billet ;
- Notification des perturbations sur le réseau ;
- Numéros utiles.

**Affluences**[47] est une application qui fait connaître à ses utilisateurs l'affluence des lieux dans lesquels ils veulent aller. Elle contient plusieurs fonctionnalités :

- **Taux d'occupation** : procure en temps réel l'occupation actuelle et les prévisions pour la journée ;
- **Temps d'attente** : donne en temps réel le temps d'attente et prévisions pour la journée ;
- **Réservation** : permet de réserver une salle, une visite, un atelier très simplement ;
- **Informations pratiques** : fournit une page par établissement avec les équipements, les coordonnées, les horaires, etc. ;
- **Carte** : permet de visualiser tous les établissements sur la carte et propose un itinéraire pour les rejoindre ;
- **Ouvert ou fermé ?** : indique les horaires d'ouvertures et prévient en cas de fermeture exceptionnelle.

#### 1.4 Ressources disponibles

Au sein même de l'université, il y a des entités qui offrent des services, aux étudiants et membres du personnel, en lien avec les besoins mis en évidence lors du sondage. Ces entités ont été contactées afin de partager les ressources numériques et informatiques dont elles disposent et qui sont nécessaires à la réalisation de l'application UCLCampus.

##### Les bibliothèques

Les bibliothèques de l'université catholique de Louvain sont coordonnées par le service central des bibliothèques, il gère leurs ressources et les représente à l'extérieur. Pour chaque bibliothèque une page est accessible sur le portail de l'UCL, elle y reprend les informations spécifiques telles que les contacts, les horaires, les formations, les services proposés, les aides, les ressources documentaires, un flux d'actualités, un agenda ainsi que des informations sur l'espace et l'équipement.

Les mémorants de l'année académique 2016-2017 responsable d'UCLCampus avaient rencontré le bibliothécaire en chef accompagné d'un informaticien dans l'équipe Bibsys et du directeur de la bibliothèque des sciences économiques, sociales et politiques. Grâce à cette réunion, ils ont eu l'obtention d'un accès aux informations spécifiques de chaque bibliothèque présente sur le portail UCL et plus particulièrement leurs horaires d'ouvertures. Ces horaires comprennent ceux en période d'examen, en période de cours et en période de vacances. L'application a le droit d'envoyer des requêtes sur deux services REST, de l'ESB, qui renvoient des réponses sous forme de JSON. Le premier service renvoie la liste des bibliothèques alors que le second renvoie en fonction de la bibliothèque une note qui concerne les horaires, les jours de fermeture et les informations de contact.

##### Le service des sports

Le service des sports s'occupe de la mise en oeuvre des activités sportives sur les campus de l'université. C'est également eux qui sont en charge du maintien de l'horaire des sports que ce soit pour les équipes universitaires ou pour tout étudiant et membre du personnel.

Une rencontre a eu lieu au sein de leurs locaux en présence de Messieurs Kevin Weber et Nicolas Gilson. Le projet UCLCampus leur a été présenté, ils y étaient intéressés et ont posé de nombreuses questions. Le service des sports a pour prochain objectif de développer une application permettant d'interagir plus facilement avec les étudiants, afin de promouvoir leurs événements et diffuser leurs actualités plus aisément. Cependant, ils ont pris contact avec Monsieur Jean-Luc Fortemaison pour donner l'accès aux flux RSS de l'horaire hebdomadaire des sports à l'application UCLCampus. Deux flux sont disponibles, le flux de l'horaire des sports des équipes universitaires et celui de l'horaire des sports accessibles à tous. Dans l'horaire, les sports sont nommés et accompagnés du bâtiment ainsi que de la salle dans laquelle il est pratiqué, de la date, de l'heure de début, de l'heure de fin et d'une note s'il en existe une. Cette note permet de tenir au courant les utilisateurs de quelque remarque telle qu'un changement de salle, d'un niveau nécessaire pour pouvoir pratiquer ce sport, d'une annulation, etc.

## **Resto U**

Le service des restaurants universitaires gère six établissements que l'on peut retrouver sur les campus universitaires de Louvain-la-Neuve, Woluwe et Mons. Cette entité est extérieure à l'UCL, elle fait partie de l'unité technique d'exploitation. Tous ces établissements proposent des menus et soupes du jour ainsi qu'une carte fixe. Chaque semaine, le menu hebdomadaire est encodé manuellement dans le CMS Druppal. Il n'y a pas de serveur ou d'API pour récupérer les informations sur les restaurants universitaires et leur menu. Cependant, les mémorands de l'année académique 2016-2017 ont rencontré la direction de cette entité pour leur présenter le projet, ces derniers en ont été enjoués et leur ont présenté leur projet d'application Resto4u qui reprend toutes les informations importantes sur les services offerts aux clients. Cette application est disponible sur les stores permettant son intégration à UCLCampus via un lien qui y accède.

## **Univers Santé**

L'UCL a créé l'ASBL Univers Santé en 2000 à la suite d'une étude sur les besoins de santé des étudiants. Univers santé promouvait la santé en milieu étudiant à l'aide d'actions d'information, de sensibilisation, d'accompagnement, de prévention et de réduction des risques qu'elle développe et travaille également en partenariat avec ses publics cibles, des professionnels de la santé, des acteurs et des associations de terrain, etc. De nombreux sujets sont abordés tels que l'alimentation, la vie affective et sexuelle, la santé mentale, le stress, la santé sociale, les assuétudes, etc.[48] Une campagne leur tient particulièrement à cœur sur le campus de l'Université Catholique de Louvain, c'est la campagne Guindaille2.0, elle est à l'initiative du développement de l'application mobile portant le même nom. Celle-ci a pour objectif la réduction des risques et des nuisances liés à la "guindaille", elle a été élaborée en étroite collaboration avec des représentants des organisations étudiantes.[49] Une rencontre a eu lieu au sein de leur bureau à Louvain-la-Neuve avec le responsable Monsieur Martin de Duve durant laquelle il a présenté l'application "Guindaille2.0" et a lui aussi découvert l'application "UCLCampus". Leur application n'ayant pas escompté le succès voulu, il recherche un autre moyen d'atteindre les étudiants. L'intégration du contenu de leur application à l'application UCLCampus l'a fortement intéressé. Il a mis à disposition l'ensemble des pictogrammes de leur campagne "Guindaille2.0", l'effectomètre et les explicatifs qui y sont liés. De plus, le flux RSS du CarpeStudentem, contenant toutes les informations sur les événements disponibles sur le campus, a également été fourni.

## **Le site internet de l'UCL**

Le portail web de l'UCL est la dernière source d'information qui sera traitée. C'est l'outil central de diffusion d'information et d'offre de services numériques fournis par l'université, il est l'outil principal de communication. De nombreux flux RSS contenant les actualités peuvent être exploités par l'application UCLCampus. Un flux d'actualité est associé à chaque faculté,

mais également un flux d'actualité général à l'université, un flux pour les études, un flux pour la recherche et un flux pour le service à la société. Mais les flux ne sont pas les seules ressources disponibles, une quantité d'informations statiques peuvent être reprises et intégrées à l'application telles que les informations sur la sécurité, les informations sur le WiFi, etc.

## 2 Spécifications fonctionnelles

Cette partie va se concentrer sur les spécifications fonctionnelles de l'application, c'est à dire du comportement attendu de celle-ci. Ces spécifications découlent des besoins identifiés lors du sondage des étudiants, de l'analyse des ressources existantes ainsi que des outils disponibles.

### 2.1 Front-end

Cette section décrit uniquement les fonctionnalités concernant le côté client : les interactions possibles entre l'utilisateur et l'interface.

#### Description des fonctionnalités par page

- **Accueil** est la première page affichée à l'ouverture de l'application. Elle comporte des boutons pour accéder aux fonctionnalités principales de l'application et un bouton paramètres afin de modifier ceux-ci.
- **News** Affiche les différentes nouvelles de l'UCL sous 3 catégories : les études, la recherche et le service à la société. Cette fonctionnalité prendra également en compte les actualités liées à la faculté d'un étudiant une fois que les flux RSS seront disponibles.
- **Détails News** affiche les détails pour une nouvelle en particulier : image, texte explicatif complet et lien vers la page internet de la news.
- **Événements** Affiche la liste des événements à venir ou en cours. Ceux-ci sont séparés en semaines et sont catégorisés visuellement via des pictogrammes. Il est possible de filtrer ces événements selon la catégorie et l'écart de temps. L'utilisateur a également la possibilité d'ajouter un événement dans l'agenda de son appareil ou dans une section "favoris".
- **Détails Événements** affiche les détails d'un événement en particulier : image, description complète et lien vers la page de l'événement.
- **Sports** Affiche la liste des sports communs accessibles à tous et des entraînements des équipes universitaires de l'UCL. Cette liste varie en fonction du campus sélectionné dans l'application. L'horaire affiché s'étale sur 7 jours et est rafraîchi à chaque ouverture de la page. Il est également possible d'ajouter un sport en favoris ou de l'ajouter dans l'agenda de l'appareil. Un système de filtre est applicable et permet de sélectionner les sports que l'on désire afficher, cela réduit le nombre de sports de la liste et facilite l'expérience de l'utilisateur.
- **Bibliothèques** affiche la liste des bibliothèques de l'UCL.
- **Détails Bibliothèque** affiche les détails d'une bibliothèque en particulier : numéro de téléphone, site web, horaires (période de cours, période d'examens, vacances et dates de fermetures).
- **Études** présente deux onglets : un onglet "Horaire" qui affiche tous les cours y étant ajoutés, il est possible d'en ajouter manuellement via ce même onglet en fonction de leur sigle. Tous les cours affichés peuvent être ajoutés entièrement à l'agenda de l'appareil, tous les slots restants de cours magistraux et des TP seront dès lors ajoutés. L'onglet "Programme de cours" permet à l'étudiant de se connecter grâce à ses identifiants UCL et d'avoir accès à son programme de cours annuel. Chacun des cours affichés peut alors être ajouté à la partie "Horaire".
- **Horaire d'un cours** affiche l'ensemble des cours magistraux et des TP restants du cours sélectionné dans l'onglet "Horaire" de la partie "Étude". Les cours magistraux et les



TP sont séparés en 2 onglets. Il y a la possibilité d'ajouter les séances que l'on choisit à l'agenda du smartphone. Pour chaque séance des détails sont affichés : local, heure, professeur, étudiants admis.

- **Support** affiche les informations utiles sur le Service Desk, la carte d'accès, la connexion au réseau Eduoram ainsi qu'un accès au répertoire UCL qui permet de rechercher un membre du personnel.
- **Détails Employés** affiche les informations sur un membre du personnel UCL : adresse, e-mail, numéro de téléphone, numéro de secrétariat ainsi que tous les organismes auxquels il appartient.
- **Carte** affiche une carte positionnée sur l'emplacement de l'utilisateur. En dessous, il y a une liste des différents auditoriums, restaurants, parkings et bibliothèques du campus sélectionné. Chaque élément peut être ajouté à la carte afin de le situer et de s'y rendre. La liste des salles d'études, salles informatiques, salles de sports doit être ajoutée.
- **Restaurants Universitaires** lance l'application Resto4U si celle-ci est déjà installée sur l'appareil. Si ce n'est pas le cas, le store de l'appareil (PlayStore ou AppleStore) se lancera sur la page de l'application pour proposer de la télécharger.
- **Mobilité** propose de lancer différentes applications concernant la mobilité (SNCB, Co-voiturage : Commuty, Bus : NextRide). Si l'une de ces applications n'est pas installée, il sera également proposé de l'installer via le store.
- **Guindaille2.0** affiche différents éléments de la campagne Guindaille2.0 créée par Univers Santé : leur pictogramme, leur effectomètre ainsi que les informations de secours.
- **Paramètres** permet de choisir son campus (Louvain-la-Neuve, Mons, Woluwe, ...) ainsi que la langue de l'application (français ou anglais).

## Structure

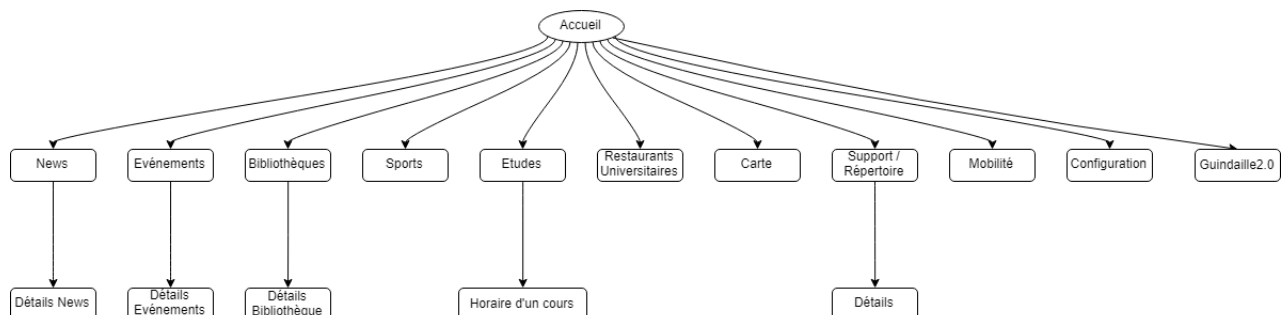


FIGURE 13 – Arborescence des pages

Comme nous pouvons le voir sur la figure 13, la page racine de notre application est la page “Accueil”, on peut aussi remarquer que l'arborescence globale n'est pas profonde. L'utilisateur n'aura pas de difficulté pour trouver l'information désirée, il lui sera compliqué de se perdre dans la structure des pages. La profondeur de pages de l'application se limite à deux pages. Depuis la page “Accueil” il est possible d'accéder à toutes les pages du second étage de l'arbre soit à l'aide du menu se trouvant sur la page même ou à l'aide du menu de navigation. Ce dernier menu permet également à quelconque page racine du deuxième niveau d'accéder à une autre page racine du même niveau ou de retourner à la page “Accueil”. Quant à lui, le bouton retour en arrière ramène les pages enfants à la page parent qui les précède. Pour ne pas rendre cette structure horizontale trop compliquée, il faut veiller à minimiser le nombre de pages racines différentes du second niveau par une catégorisation minutieuse des fonctionnalités.

## 2.2 Back-end

Dans cette section, il sera présenté les fonctionnalités du côté “serveur” : la partie non visible à l'utilisateur ainsi que les contraintes dues à la nécessité de sécuriser certaines informations.

### Sécurité

L'application comporte deux types d'échange d'information : les informations publiques telles que les news, les événements, l'horaire des sports, etc. Il y a également des informations qui sont plus personnelles et confidentielles telles que le programme de cours d'un étudiant et les identifiants de ce dernier. Pour les informations publiques, l'aspect sécurité n'est pas important étant donné qu'elles sont disponibles sur le site de l'UCL et d'autres sites publics. Pour les autres, la sécurité est primordiale. Un étudiant qui utilise l'application et veut charger son programme de cours à l'intérieur doit d'abord se connecter en utilisant son identifiant et mot de passe. Ces derniers se doivent d'être cryptés lors de l'échange d'information sur le réseau et ne pas être stockés dans l'appareil. Le programme de cours de l'étudiant n'est pas une information disponible publiquement donc il faut également garantir un transfert sécurisé et ne doit pas être stocké dans le smartphone.

## 2.3 User Stories

Dans la méthode de développement agile, les récits utilisateurs ou user stories sont des phrases simples qui décrivent avec précision les fonctionnalités attendues à développer. Cette phrase essaie de répondre aux trois questions : Qui ? Quoi ? Pourquoi ?

### Accueil

- En tant qu'utilisateur, je veux choisir dans quelle section de l'application je veux me diriger.
- En tant qu'utilisateur, je veux avoir accès aux paramètres facilement.

### News

- En tant qu'utilisateur, je veux afficher les actualités de l'UCL afin de me tenir informé.
- En tant qu'utilisateur, je veux accéder à une actualité spécifique grâce à une recherche par mot-clé.
- En tant qu'utilisateur, je veux afficher toutes les informations disponibles à propos d'une actualité.
- En tant qu'utilisateur, je veux aller sur site web où l'actualité a été postée.

### Évènements

- En tant qu'utilisateur, je veux afficher la liste des événements et diverses activités en cours et à venir.
- En tant qu'utilisateur, je veux afficher les informations détaillées d'un événement.
- En tant qu'utilisateur, je veux ajouter un événement à mes favoris afin de le retrouver plus rapidement à l'avenir.
- En tant qu'utilisateur, je veux ajouter un événement à l'agenda de mon appareil afin d'avoir un rappel pour cet événement.
- En tant qu'utilisateur, je veux pouvoir aller sur le site web de l'événement.
- En tant qu'utilisateur, je veux accéder à un événement spécifique grâce à une recherche par mot-clé.
- En tant qu'utilisateur, je veux filtrer les événements pour les X prochains mois ou en fonction de leurs catégories.

## Sports

- En tant qu'utilisateur, je veux afficher la liste des sports à venir dans la semaine.
- En tant qu'utilisateur, je veux afficher la liste des entraînements universitaires à venir dans la semaine.
- En tant qu'utilisateur, je veux voir l'horaire et la salle d'un sport (ou d'un entraînement) en particulier.
- En tant qu'utilisateur, je veux être prévenu en cas de changement de salle, annulation ou s'il y a besoin d'une réservation.
- En tant qu'utilisateur, je veux accéder à un sport (ou un entraînement) spécifique grâce à une recherche par mot-clé.
- En tant qu'utilisateur, je veux filtrer les sports (ou les entraînements) en fonction du type de sport.
- En tant qu'utilisateur, je veux ajouter un sport (ou un entraînement) à mes favoris afin de le retrouver plus rapidement à l'avenir.
- En tant qu'utilisateur, je veux ajouter un sport (ou un entraînement) à l'agenda de mon appareil.

## Bibliothèques

- En tant qu'utilisateur, je veux afficher la liste des bibliothèques de l'UCL.
- En tant qu'utilisateur, je veux afficher les informations détaillées (numéro de téléphone, heures d'ouverture, etc.) d'une bibliothèque en particulier.
- En tant qu'utilisateur, je veux pouvoir aller sur le site web de la bibliothèque.

## Études

- En tant qu'utilisateur, je veux ajouter un cours à l'application afin de pouvoir en voir l'horaire.
- En tant qu'utilisateur, je veux voir l'horaire des cours magistraux et des travaux pratiques d'un cours en particulier.
- En tant qu'utilisateur, je veux voir la salle où se déroule un cours magistral ou une séance de travaux pratiques.
- En tant qu'utilisateur, je veux pouvoir accéder au site web de l'UCL.
- En tant qu'utilisateur, je veux pouvoir accéder au site web de Moodle.
- En tant qu'utilisateur, je veux ajouter un cours magistral ou une séance de travaux pratiques à l'agenda de mon appareil.
- En tant qu'utilisateur, je veux ajouter en une fois toutes les séances de cours magistraux et de travaux pratiques d'un cours en particulier dans l'agenda de mon appareil.
- En tant qu'utilisateur, je veux choisir la séance de T.P ou de cours magistral à laquelle je dois aller.
- En tant qu'utilisateur, je veux pouvoir me connecter avec mes identifiants UCL.
- En tant qu'utilisateur connecté, je veux voir mon programme de cours de l'année.
- En tant qu'utilisateur connecté, je veux ajouter un cours de mon programme à ma liste de cours.
- En tant qu'utilisateur connecté, je veux voir mes notes d'examen.

## Support

- En tant qu'utilisateur, je veux afficher les informations à propos du Service Desk afin de le contacter en cas de besoin.
- En tant qu'utilisateur, je veux afficher les informations liées à la carte d'accès.
- En tant qu'utilisateur, je veux afficher les informations liées au réseau WiFi eduoram.

- En tant qu'utilisateur, je veux rechercher un membre du personnel de l'UCL par son nom ou son prénom.
- En tant qu'utilisateur, je veux afficher les informations détaillées d'un membre du personnel.

## Carte

- En tant qu'utilisateur, je veux connaître ma géolocalisation actuelle sur la carte.
- En tant qu'utilisateur, je veux afficher la liste des points d'intérêts par catégorie.
- En tant qu'utilisateur, je veux connaître la position d'un point d'intérêt sur la carte.
- En tant qu'utilisateur, je veux connaître l'itinéraire vers un point d'intérêt afin de m'y rendre.
- En tant qu'utilisateur, je veux accéder à un point d'intérêt spécifique grâce à une recherche par mot-clé.

## Restaurants Universitaires

- En tant qu'utilisateur, je veux lancer l'application des restaurants universitaires (Resto4U) afin d'avoir toutes les informations disponibles (horaires, menus, etc.).
- En tant qu'utilisateur, je veux pouvoir installer l'application Resto4U.

## Mobilité

- En tant qu'utilisateur, je veux lancer l'application SNCB afin de connaître les horaires de train.
- En tant qu'utilisateur, je veux pouvoir installer l'application SNCB.
- En tant qu'utilisateur, je veux lancer l'application NextRide afin de connaître les horaires de bus.
- En tant qu'utilisateur, je veux pouvoir installer l'application NextRide.
- En tant qu'utilisateur, je veux lancer l'application Commuty afin d'avoir accès aux services de covoiturage.
- En tant qu'utilisateur, je veux pouvoir installer l'application Commuty.

## Guindaille2.0

- En tant qu'utilisateur, je veux avoir accès aux différents éléments (pictogrammes, effectomètre, préventions, etc.) de la campagne Guindaille2.0.
- En tant qu'utilisateur, je veux afficher les informations de secours telles que le numéro de téléphone des urgences générales et du gardiennage UCL.

## Paramètres

- En tant qu'utilisateur, je veux pouvoir choisir la langue utilisée par l'application.
- En tant qu'utilisateur, je veux pouvoir choisir un des campus de l'UCL.

# 3 Spécifications non fonctionnelles

Les spécifications non fonctionnelles rassemblent les exigences qui ne sont pas liées au comportement de l'application en terme d'activités mais qui devront être satisfaites.

Du point de vue de l'**utilisation**, il faut que l'utilisateur trouve ça le plus simple et intuitif possible. Un guide utilisateur ainsi qu'un tutoriel de prise en main doivent être à sa disposition pour qu'il puisse s'y référer en cas de nécessité. L'application doit également rester fonctionnelle

lorsque des services UCL sont en maintenances ou que l'utilisateur ne peut pas se connecter à internet, pour cela il faut veiller à ce que les informations récupérées lors de la précédente utilisation soient mises en cache. Pour finir le niveau utilisation, l'aspect Open Source implique la présence d'un fichier readme et d'un guide d'installation pour les futurs développeurs. Cela leur sera nécessaire pour se joindre au projet plus facilement. Le code quant à lui se devra d'être compréhensible, les commentaires ainsi que le choix des noms des variables et des fonctions seront appropriés.

Du point de vue **programmation**, afin d'aller dans la continuité du projet UCLCampus de l'année académique 2016-2017, le framework a utilisé est Ionic version 3 qui est basé sur Angular 4.0, TypeScript, JavaScript et HTML5.

Du point de vue de la **fiabilité**, les problèmes techniques évidents ne doivent pas se produire, l'application doit être la plus fiable possible. Si des problèmes récurrents surviennent, l'utilisateur s'en lassera très rapidement et la désinstallera.

Du point de vue de la **modularité**, il faut que les services développés dans l'application soient indépendants entre eux. De plus, l'application doit être en adéquation avec le concept d'architecture trois tiers. C'est-à-dire qu'elle sera divisée en 3 couches, la couche interface, la couche logique et la couche donnée.

- La couche interface correspond à la partie visible et interactive de l'application pour les utilisateurs ;
- La couche logique correspond à la partie fonctionnelle de l'application ;
- La couche donnée correspond à la partie gérant l'accès aux données de l'application.

Du point de vue de la **compatibilité**, l'application doit pouvoir tourner sur tous les smartphones avec une version Android 4.4 ou plus récente ainsi que sur les smartphones sous iOS version 8 ou plus récent. Elle devra par la même occasion être disponible sur le Play Store et l'App Store. Il faut également respecter la convention de semantic versionning, c'est-à-dire que pour une version MAJOR.MINOR.PATCH, une mise à jour impliquera l'incrément de :

- **MAJOR** lorsque l'on effectue des modifications API incompatibles ;
- **MINOR** lorsque l'on ajoute des fonctionnalités de manière rétrocompatible ;
- **PATCH** lorsque l'on fait des corrections de bugs rétrocompatibles.

Des labels supplémentaires pour les métadonnées de pré-version et de compilation sont disponibles sous forme d'extensions au format MAJOR.MINOR.MINOR.PATCH.

Du point de vue de la **performance**, l'utilisateur requiert un temps de chargement minime d'une page, la limite est d'une seconde, une fois franchie un logo de chargement doit être affiché. Ajouté à cela, l'application a pour objectif d'être téléchargée par un maximum d'étudiants, approximativement 20 000, elle doit être capable de pouvoir supporter une telle charge. Pour la performance niveau utilisation des ressources, l'application devra d'une part ne pas exploiter trop de mémoire cache et d'autre part limiter sa consommation en énergie pour préserver la batterie du smartphone.

Et enfin pour l'**aspect légal**, l'application sera sous licence GNU GPL v3 et devra respecter les règles de la charte graphique de l'Université catholique de Louvain. Une seule version du logo de l'UCL est en vigueur, seule celle-ci peut-être utilisée.

**Quatrième partie**

**Développement d'UCLCampus**

Ce chapitre décrit plus précisément le développement de l'application. Il se compose de 4 sections. La première se concentre principalement sur l'environnement Open Source du projet. La seconde quant à elle fait part des choix d'implémentation qui ont été réalisés. L'avant-dernière section se penche sur la sécurité, un point à ne pas négliger, plus précisément en ce qui concerne le stockage de l'information et l'authentification aux services de l'UCL. Pour terminer, la dernière section présente en détail les modules réalisés, en ce qui concerne le fonctionnement des fonctionnalités de l'application.

## 1 Continuité dans le mouvement Open Source

L'application UCLCampus est un projet Open Source, à cette fin, elle doit être développée dans un environnement en adéquation avec ce concept. Cette section explique le choix de la licence à appliquer, les termes à respecter liés à la licence, le choix de la forge la plus pertinente pour le développement, la mise en place durant des canaux de communication pour informer la communauté de l'état d'avancement du projet et pour finir la solution apportée pour le stockage des données sensibles.

### 1.1 Licence

Le choix de licence pour le projet UCLCampus s'est tourné vers une licence "restrictive". Ce choix a été fait durant l'année académique 2016-2017 afin de protéger le projet des détournements ne s'alignant pas à la philosophie et aux valeurs du projet ainsi que de l'Université catholique de Louvain. UCLCampus est à ce jour un produit fini, avec toutes les fonctionnalités opérationnelles, qui s'adresse directement aux étudiants et non à d'autres développeurs. C'est pour cela que ce type de licence a été choisi, principalement pour se protéger de toute entité voulant l'utiliser à des fins commerciales. Plus spécifiquement, pour se protéger d'une concurrence néfaste par des modifications plus attrayantes du projet initial sans en faire profiter celui-ci. Les licences restrictives permettent de contrer ce problème, toute entité qui reprend le projet doit rendre son code Open Source avec la même licence, ce qui réduira ses possibilités et permettra à l'UCL de garder un oeil sur le contenu de l'application concurrente. Si des fonctionnalités supplémentaires sont développées par la concurrence se basant sur le projet UCLCampus, l'UCL pourra également reprendre ses fonctionnalités afin d'améliorer leur propre application.

Mais ce n'est pas seulement pour ça que l'application est sous licence Open Source, c'est aussi pour permettre à d'autres étudiants ou universités d'utiliser le code de base afin de l'adapter pour répondre à ses propres besoins. La licence doit être la plus compatible avec d'autres licences pour que l'utilisation de plugins et librairies externes soit possible.

Pour répondre au mieux à ces critères, le choix le plus pertinent est la licence GNU GPL v3. Le caractère copyleft de cette licence protège le projet des dérivées commerciales et oblige toute copie de ce projet souhaitant être distribuée d'être également sous licence GNU GPL v3. De plus, étant donné son côté très restrictif, l'intégration de code est autorisée s'il provient d'autres licences plus permissives et non contradictoires telles que MIT, Apache 2.0.

*Copyright (c) Université catholique Louvain. All rights reserved*

*This file is part of UCLCampus*

*Licensed under the GPL 3.0 license. See LICENSE file in the project root for full license information.*

*UCLCampus is free software : you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.*

*UCLCampus is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.*

*You should have received a copy of the GNU General Public License along with UCLCampus. If not, see <<http://www.gnu.org/licenses/>>.*

Le texte ci-dessus est à ajouter à chaque fichier pour que le projet UCLCampus soit reconnu sous cette licence. Ce texte contient l’avis de copyright et la déclaration de la licence. Le copyright appartient à l’UCL pour lui permettre une plus grande flexibilité dans la gestion du code source. Si cela était aux noms des développeurs de l’application, étant donné la licence choisie, l’UCL devrait les contacter pour leur demander leur accord si elle désire changer de licence. Il est donc préférable que les contributeurs acceptent de céder leurs droits d’auteur sur leur participation au code source du projet à l’université, évitant ainsi tout problème. En plus de ce texte, le projet contient un fichier LICENCE qui détaille le contenu de la licence GNU GPL v3 appliquée.

Du point de vue des licences des frameworks, Cordova est sous licence Apache 2.0 alors qu’Angular et Ionic sont sous licence MIT, elles ne posent aucun problème avec celles choisies pour notre projet. Pour les plugins utilisés dans l’application, c’est principalement la même chose, ils sont soit sous licence Apache 2.0 ou sous licence MIT. Aucun problème de compatibilité n’est à déplorer.

## 1.2 Forge

Comme expliqué dans la partie Open Source du second chapitre, une forge est un système de gestion de développement collaboratif de logiciel. Elle est nécessaire pour accueillir le projet UCLCampus. GitHub a été choisi pour prendre ce rôle dû à son interface conviviale et intuitive. C’est une plateforme propriétaire de projet libre qui utilise le logiciel de gestion de version Git, elle propose un service web d’hébergement et de gestion de développement de logiciel. Selon leurs propres chiffres sortis en 2016, cette forge est utilisée par plus de 14 millions d’utilisateurs et héberge plus de 35 millions de projets, elle est considérée comme forge “sociale” grâce aux nombreuses fonctionnalités proposées se rapprochant de celles des réseaux sociaux. Chaque utilisateur a son propre compte, il peut communiquer avec d’autres au sujet d’un projet, il peut suivre d’autres utilisateurs ou projets ainsi que voir leurs activités telles que les commit, fork, etc., augmentant ainsi la visibilité des projets.

GitHub offre également la possibilité de réaliser un Wiki spécifique à chaque projet. Le Wiki définit la feuille de route du projet, montre son état d’avancement actuel et sa documentation. L’avantage d’un Wiki est que tous les membres de la communauté peuvent le modifier et l’enrichir librement.

Mais ce n’est pas tout, GitHub propose aussi d’intégrer des services externes à ses projets afin de rendre plus simple le développement et l’interaction avec les contributeurs et les utilisateurs du projet.

Gitter et ZenHub sont les deux services externes à être utilisés pour le projet UCLCampus, ils ont été mis en place afin de faire participer la communauté Open Source.

## Gitter

Ce service externe est une plateforme de chat et de networking qui aide à gérer, développer et connecter les communautés par le biais de la messagerie, du contenu et de la découverte. Elle est conçue pour la collaboration communautaire, son objectif est que les messages de la communauté, la collaboration et la découverte soient aussi fluides et simples que possible.[50] Une salle de discussion publique a été créée pour le projet UCLCampus et est accessible facilement grâce à



un badge graphique ajouté dans le fichier README du projet GitHub. Cela permettra d'attirer rapidement l'attention des utilisateurs et des développeurs.

## **ZenHub**

ZenHub quant à lui est un outil de gestion de projet, pour les équipes de développeurs, s'intégrant à l'interface utilisateur de GitHub à l'aide d'un plugin à installer à votre navigateur Chrome ou Firefox. C'est une solution pour améliorer le flux de travail, il utilise des données en live et donne un aperçu plus approfondi ainsi des mesures plus précises dans le processus de développement du projet. La fonctionnalité principalement utilisée pour le projet UCLCampus est le board. Il permet aux utilisateurs d'y ajouter des issues, c'est-à-dire des sujets ou problèmes importants à régler. Les issues sont transférées par les développeurs d'une colonne à une autre en fonction de leur état d'avancement. La communauté est alors capable d'interagir et de donner un retour sur le projet et les problèmes rencontrés. D'autres fonctionnalités existent mais n'ont pas été utilisées telles que TODO et EPIC.

### **1.3 Création d'une communauté**

La communauté rassemble un panel de personnes participant à la vie du projet allant du développeur à l'utilisateur qui fait du bouche-à-oreille afin de faire connaître le projet. Cette communauté est nécessaire pour l'évolution et la survie du projet UCLCampus. Dans le contexte de cette application, elle sera composée principalement d'étudiants de l'Université catholique de Louvain mais elle pourrait aussi intéresser des membres d'autres universités.

La création de la communauté Open Source demande un investissement considérable, la réalisation d'un manuel utilisateur, d'un tutoriel vidéo et d'un manuel de développement est une base pour faciliter l'intégration au projet mais ce n'est pas suffisant. Des outils tels que GitHub, ZenHub et Gitter sont primordiaux pour permettre un développement continu de l'application grâce à leurs canaux de communication servant à garder un contact avec chacun durant la vie du projet.

### **1.4 Données sensibles**

Même si le projet UCLCampus est développé en Open Source, certaines informations peuvent être confidentielles et ne doivent pas apparaître au grand public. C'est le cas pour les clés API. L'utilisation de ses clés se fait notamment pour l'accès aux services Google Maps qui nécessite une identification de l'application à l'aide de celles-ci.

Ces clés uniques sont fournies par Google pour chaque application, elles permettent l'accès aux services voulus mais sont payantes après un certain quota de connexion, c'est une des façons pour Google de gagner de l'argent. Ces clés ne doivent donc sous aucun prétexte apparaître dans le code source sous peine d'être volées et détournées. Les développeurs qui voudraient participer au projet doivent acquérir par eux-mêmes auprès de Google leur propre clé API afin de faire fonctionner l'application.

Un fichier de configuration qui permet de contenir les clés API a été créé sur le GitHub du projet. Il a été ajouté à la liste des fichiers à ignorer en cas de soumission de code sur le GitHub de façon à protéger les clés API des développeurs qui participent au projet. Les fichiers utilisant les clés API font appel aux valeurs et données dans le fichier de configuration où les clés sont stockées. Si le fichier de configuration est correctement écrit, les données s'y trouvant sont invisibles sur le Github du projet et l'application fonctionne comme il se doit.[51]

## **2 Choix d'implémentation**

Cette section explique les choix d'implémentation lors de la réalisation de l'application UCLCampus. Pour cette réalisation, le framework Ionic d'Angular4 est utilisé. Premièrement, il

sera présenté la structure de l'application et les choix résultant de l'utilisation de ce framework. Il sera également présenté un objet que TypeScript propose : la Promise. Ensuite, le rôle des différents modules implémentés pour l'interface utilisateur sera développé.

## 2.1 Ionic

L'utilisation du framework Ionic impose de respecter une certaine architecture, structure générale pour l'application. Le développeur étend les fonctionnalités de l'application en créant différents modules et composants qui serviront à une fonctionnalité précise. Il existe deux types de modules :

- **Les pages** : elles s'occupent de tout ce qui est interface graphique, gestion des événements et interaction avec l'utilisateur.
- **Les services ou “providers”** : est utilisé pour s'occuper de la partie logique (calcul, récupération de données via une API).

Un service est importé dans une page selon la nécessité des fonctionnalités qui interagissent avec la partie logique. Un module a une fonctionnalité précise et est indépendant des autres. Cependant, un module peut importer des services implémentés dans d'autres modules ce qui rend plus facile le développement Open Source car il n'est pas nécessaire de connaître la composition et le fonctionnement d'un module pour qu'un développeur le fasse interagir avec le module qu'il aurait lui-même créé. Ça facilite également la création d'un environnement de tests spécifiques à un module et ses interactions avec d'autres modules ou services. Il est donc possible d'améliorer rapidement un module en gardant ses fonctionnalités et de le remplacer.

## 2.2 L'objet Promise

L'application UCLCampus utilise de nombreuses requêtes HTTP GET afin de récupérer diverses informations pour les différents modules. Ces requêtes étant asynchrones, TypeScript propose l'utilisation de l'objet Promise pour traiter les réponses à ces requêtes. Une promise[52] est la transformation ou la réification d'une fonction asynchrone en un objet pour faciliter la manipulation de cette fonction. Le constructeur de cette Promise prend en paramètre une fonction qui prend elle-même deux fonctions : `resolve()` qui est appelée en cas de succès de la fonction asynchrone et `reject()` qui est appelée en cas d'échec. La Promise ne peut être résolue et rejetée simultanément.

## 2.3 User Interface

Ce sont les pages qui composent le côté interface utilisateur. Celles-ci sont composées de plusieurs éléments :

- **page.ts** qui contiendra le code TypeScript de la page. C'est dans ce fichier que sera géré le data-binding expliqué précédemment, l'appel de méthode suite à une interaction avec l'utilisateur ou même l'affichage d'une valeur.
- **page.html** qui contiendra le code HTML. Celui-ci est responsable de la structure et de l'affichage de la page. Angular permet d'ajouter des balises HTML personnalisées qui ont leur propre module avec leur fonctionnalité spécifique.
- **page.scss** qui contient les styles Sass (Syntactically Awesome Stylesheets) qui est un langage de génération de feuille de style. Ce fichier s'occupera du rendu visuel de la page.
- **page.module.ts** qui servira au chargement paresseux des modules et pages utilisées. Il s'occupe donc d'importer tous les services, modules et pages enfants nécessaires au fonctionnement du module.

On peut imaginer le système de navigation entre les pages comme une pile : lorsqu'on lance l'application, le module “app” est le premier à être appelé. C'est dans celui-ci que le menu déroulant est implémenté ce qui le rend accessible dans toutes les pages racines de l'application.

On peut voir ce module app comme la base de cette pile. “app.module.ts” importe tous les services qui serviront dans l’application ainsi que toutes les pages racines. Le premier élément de la pile sera la page Accueil (qui sera donc le premier étage de la pile). Lorsqu’on appuie sur une icône de cette page, la nouvelle page est ajoutée au sommet de la pile (et sera le deuxième étage de la pile) et lorsqu’on appuie sur le bouton retour, cette page est retirée de la pile. Les pages ayant une page enfant fonctionnent de la même manière : la page enfant s’ajoute à la pile et sera donc le troisième étage et peut ensuite être retirée via le bouton retour. Ici, nous considérerons les pages racines comme les pages du premier étage (et donc uniquement la page Accueil) de la figure 13 page 48. L’utilisation du menu déroulant est encore à modifier : celui-ci considère toutes les pages du deuxième étage comme étant des pages racines également. Lorsqu’on l’utilise, il supprime la page Accueil de la pile et ajoute une de ces pages au sommet de la pile à sa place.

### 3 Sécurité

L’application utilise et a besoin de plusieurs services de l’Université Catholique de Louvain afin de récupérer les informations utiles et de les présenter à l’utilisateur. Pour certaines informations qui sont privées, il est nécessaire à un utilisateur étant un étudiant à l’UCL de pouvoir s’identifier au sein même de l’application avec son identifiant et mot de passe personnel UCL. Ces données sont le programme de cours de l’étudiant pour le moment mais il est possible que plus tard, l’accès aux notes d’examens soit ajouté. Dans la première section, il sera décrit la problématique de cette authentification vers les services UCL qui se doit d’être sécurisée. Ensuite, il sera présenté la manière de stocker ces informations sensibles de manière sécurisée dans l’appareil et lors des échanges d’informations.

#### 3.1 Enjeux et solutions pour l’authentification aux services

L’Université Catholique de Louvain utilise Shibboleth, qui a été expliqué précédemment, qui est un mécanisme de fédération d’identité pour s’authentifier. L’application UCLCampus n’ayant aucun accès direct à un Service Provider ou un Identity Provider, ne peut pas utiliser un tel système d’authentification sécurisé. La mise en place d’une API sur l’API manager de l’ESB WSO2 a été nécessaire afin de permettre à un étudiant de l’UCL de récupérer son programme de cours après s’être identifié avec son identifiant et mot de passe. Il est d’abord nécessaire d’enregistrer l’application UCLCampus dans l’API manager[53] ce qui générera une clé et un secret de consommateur qui représentent les informations d’identification de l’application enregistrée. Cette clé de consommateur devient en fait l’identifiant unique de l’application, similaire au nom d’utilisateur d’un utilisateur, et est utilisée pour authentifier l’utilisateur ou l’application. Lorsqu’un Token d’accès est généré pour l’application, il est émis contre la dernière clé de consommateur connue. Les utilisateurs d’API génèrent des Tokens d’accès et les intègrent dans les requêtes vers les API : la clé API (le token d’accès généré) est une simple chaîne de caractère qu’il suffit de passer dans le header de la requête HTTP. Ceci permet d’éviter certains types d’attaque denial-of-service (DoS) : si le token passé dans une requête est invalide, le WSO2 API Manager rejette lui-même cette demande dans la première étape du traitement.

Le WSO2 API Manager fournit deux types de tokens d’accès pour l’authentification :

- **Tokens d’accès aux applications** : ils permettent d’identifier et d’authentifier une application entière. Une application est une collection de nombreuses API. Avec un seul token d’accès à l’application, il est possible d’appeler toutes ces API. Ces tokens exploitent OAuth2 pour fournir une gestion simple des clés.
- **Tokens d’accès utilisateur** : ils servent à identifier l’utilisateur final d’une application. Par exemple, ici, un étudiant de l’UCL. Ces tokens sont générés ou renouvelés en appelant l’API Login à travers un client REST.

## Persistance du token[54]

Le composant OAuth2 de l'API Manager de WSO2 (WSO2-API-M) possède deux implémentations utilisables pour gérer la persistance des tokens dans la base de données, à savoir la persistance des tokens synchrones et asynchrones :

### — Tokens synchrones

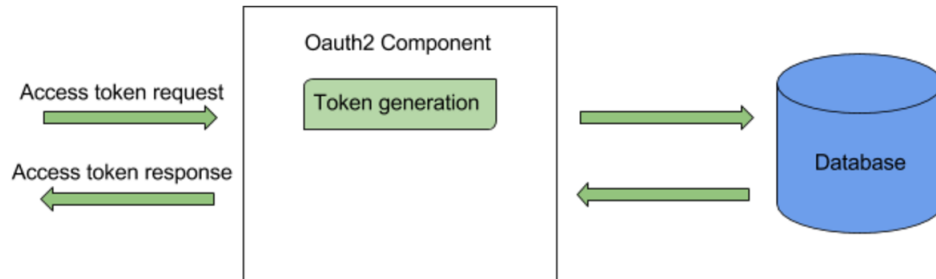


FIGURE 14 – Flux de la persistance des tokens synchrones

1. Le client envoie une demande de token d'accès.
2. OAuth2 recherche un token d'accès actif existant pour le client/utilisateur. Il vérifie d'abord le cache et si un token actif n'est pas trouvé, il vérifie ensuite la base de données.
3. Si un token d'accès actif est trouvé, le token est renvoyé au client.
4. Sinon, OAuth2 en crée un nouveau et le conserve dans la base de données en utilisant le même thread. Une fois qu'il est conservé, le nouveau token est renvoyé au client.

### — Tokens asynchrones

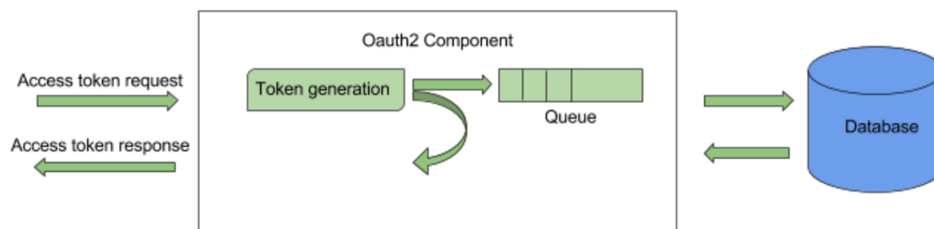


FIGURE 15 – Flux de la persistance des tokens asynchrones

1. Le client envoie une demande de token d'accès.
2. OAuth2 recherche un token d'accès actif existant pour le client/utilisateur. Il vérifie d'abord le cache et si un token actif n'est pas trouvé, il vérifie ensuite la base de données.
3. Si un token d'accès actif est trouvé, le token est renvoyé au client.
4. Sinon, OAuth2 en crée un nouveau et l'ajoute à une file d'attente persistante.
5. Une fois ajouté à la file, le token est renvoyé au client.
6. Par la suite, il y a des threads en arrière-plan qui consomment la file d'attente et ils conservent ces jetons dans la base de données.

## 3.2 Stockage de l'information

Afin de fonctionner correctement, il est nécessaire que l'application stocke certaines informations. Celles-ci sont de plusieurs types :

- **Données statiques** : ce sont les données qui sont écrites directement dans les fichiers (HTML, JSON, etc.) Par exemple, il y a tous les textes de l'application ainsi que leurs traductions en anglais et il y a aussi les coordonnées des différents locaux et salles.
- **Données publiques/non sensibles** : ce sont les données résultant d'une requête mais étant accessible sans s'identifier. Elles sont donc disponibles publiquement et ne nécessitent pas de sécurité particulière. Par exemple, il y a les horaires des différentes bibliothèques, l'horaire des sports ainsi que les news et événements.
- **Données privées/sensibles** : ce sont les données nécessitant l'authentification de l'étudiant et qui sont donc propre à l'utilisateur. Ces données se doivent d'être stockées de manière sécurisée. Par exemple, il y a le programme de cours d'un étudiant et à l'avenir les notes d'examens.

Évidemment, selon le type de données, il faut les garder en mémoire de manière différente. Heureusement, il existe des techniques de stockage ayant chacun leurs avantages, inconvénients et surtout un niveau de sécurité différent :

- **Stockage simple** : cela consiste simplement à stocker l'information sans chiffrement/-cryptage. Il suffit d'accéder à l'endroit où sont les informations pour les récupérer ou les lire.
- **Stockage par chiffrement** :
  1. **Symétrique** : cette méthode utilise un algorithme qui utilise la même clé de chiffrement pour le codage et le décodage. Cette clé doit être tenue secrète. L'avantage du chiffrement symétrique est qu'il est très rapide.
  2. **Asymétrique** : l'algorithme utilise une paire de clés : une publique et une privée. La clé publique est connue par tous et sert à chiffrer ; la privée est tenue secrète et sert à déchiffrer. Pour déchiffrer un message, il faut donc la clé privée associée à la clé publique ayant servi au chiffrement du message. L'inconvénient par rapport au chiffrement symétrique est qu'il est beaucoup plus lent.

L'utilisation d'Ionic et Cordova permet d'utiliser deux de leurs plugins dans l'application afin de garder en mémoire les données :

- **Storage[55]** est un module de stockage simple utilisant une paire clé-valeur avec un support prêt à l'emploi pour SQLite. Il facilite l'utilisation du meilleur moteur de stockage disponible, en fonction de la plateforme, sans avoir à interagir directement avec celui-ci. Il permet également d'utiliser des objets JSON. Ce mode de stockage étant non sécurisé, il ne servira qu'à garder en mémoire les données non sensibles.
- **SecureStorage[56]** permet à l'application de stocker en toute sécurité des informations sensibles. Il a été implémenté dans le plugin cordova-plugin-secure-storage et est supporté par plusieurs plateformes. Selon la plateforme, le plugin fonctionne différemment :
  - **Android** : L'API SecureStorage est implémentée comme suit :
    - Une clé AES de 256bits est générée.
    - Les données sont chiffrées avec la clé.
    - La clé AES est chiffrée avec un algorithme RSA (RSA / ECB / PKCS1Padding) proposé par Android.
    - La combinaison de la clé AES et des données chiffrées est stockée dans SharedPreferences.
    - Lorsqu'il faut récupérer les données, le procédé inverse est suivi.
 Afin d'utiliser cette méthode, il est nécessaire que l'utilisateur ait un écran de verrouillage sécurisé car c'est ce code qui servira de clé AES. Il serait nécessaire que l'application refuse la requête aux données sensibles si l'utilisateur n'a pas paramétré d'écran de verrouillage et de le renvoyer vers les paramètres correspondants.
  - **iOS** : Les données sont stockées directement dans le stockage KeyChain d'iOS via la librairie SAMKeychain. Une keychain est un conteneur crypté où seront enregistrées les données. Chaque application a son propre keychain auquel elle seule a accès mais il

est possible de partager les données sensibles entre plusieurs applications. Il y a deux clés de chiffrement : le code de l'écran de verrouillage de l'appareil et une clé unique associée à l'appareil

## 4 Modules réalisés

Dans cette section l'application UCLCampus sera décortiquée de manière approfondie, partie par partie, module par module. Chaque partie contient plusieurs modules qui lui sont propres selon son domaine de prédilection. Tout d'abord, il sera présenté le module Accueil qui permet à l'utilisateur de naviguer dans les différentes parties de l'application et d'accéder aux paramètres. Les trois parties distinctes de l'application sont "Campus", "Études" et "Sports". Le fonctionnement de chaque module sera détaillé au sein de cette section avec par la même occasion une description des pages qui le composent. De plus, les modules utilisent des informations récupérées de différentes façons telles que la lecture de flux RSS, des requêtes HTTP GET, des interactions avec les API de WSO2 ainsi que celles de services externes et internes à l'UCL. Ces systèmes de récupérations de l'information seront également décrits pour mieux comprendre le fonctionnement des modules et d'où ils récupèrent leurs informations.

### 4.1 Accueil

Lors du premier lancement de l'application, ce module propose à l'utilisateur de sélectionner un campus. Une fois choisi, le menu qui permet d'accéder à toutes les fonctionnalités/modules est affiché pour que l'utilisateur puisse naviguer à travers l'application.

Ce module est composé de plusieurs éléments :

- **HomePage** : est une view qui affiche soit le choix du campus lors du premier lancement, soit le menu permettant de naviguer à travers l'application ;
- **UserService** : est un service qui permet de récupérer des données enregistrées dans le stockage de l'appareil ;
- **TranslateService** : est un service qui provient de @ngx-translate et permet d'avoir une application multilingue.

Le front-end est implémenté avec la page HomePage tandis que le back-end est formé avec le service UserService. Lors du chargement du module Accueil, un appel à UserService est fait pour savoir si l'utilisateur a déjà sélectionné un campus lors d'un précédent lancement de l'application. UserService va dès lors vérifier si un campus est enregistré dans le stockage grâce au plugin @ionic/storage. Si UserService renvoie false, HomePage affiche une view permettant à l'utilisateur de choisir son campus entre Louvain-la-Neuve, Woluwe et Mons. Une fois le choix réalisé, un nouvel appel à UserService est fait pour enregistrer ce choix dans le stockage via le même plugin @ionic/storage. Ensuite, HomePage affiche le menu qui permet de naviguer à travers l'application. Si UserService renvoie true, ce menu est directement affiché.

Lorsque l'utilisateur sélectionne une icône du menu, deux scénarios sont possibles :

1. Le module est implémenté dans l'application : un appel à NavController de ionic-angular est réalisé afin de push la page/le module et de l'afficher ;
2. Le module est une application externe (par exemple Resto4U) : un appel à Device de @ionic-native/device est réalisé afin de vérifier sur quel OS l'application est utilisée. Ensuite un appel à App-Availability de @ionic-native/app-availability est fait pour vérifier si l'application externe est déjà installée sur l'appareil. Si c'est le cas, InAppBrowser de @ionic-native/in-app-browser est appelé pour lancer cette application. C'est Market de @ionic-native/market qui est appelé pour lancer le store et proposer à l'utilisateur de l'installer.

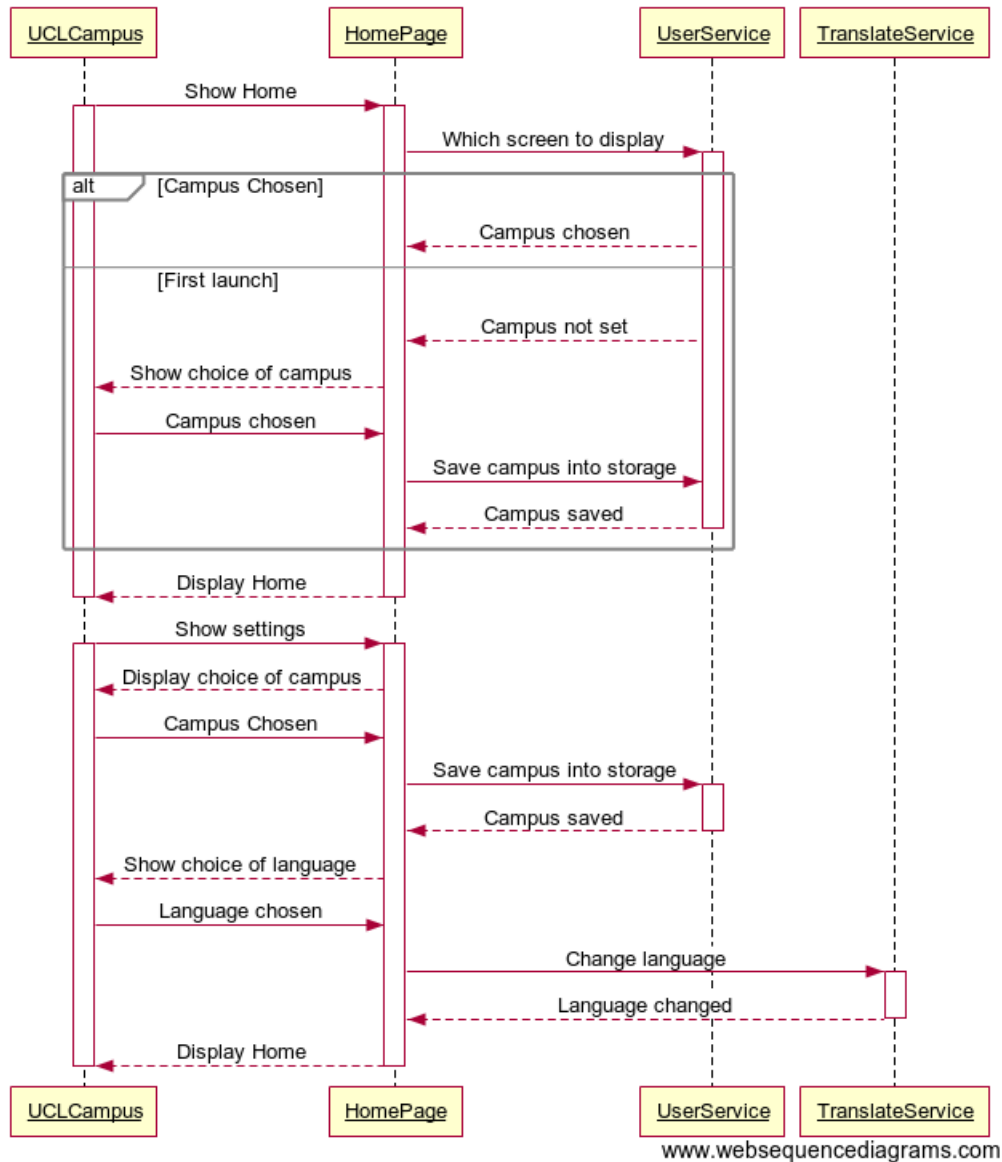


FIGURE 16 – Diagramme de séquence pour l’affichage de l’accueil

## Paramètres

Le module "Paramètres" n'en est pas un à proprement parlé, c'est plutôt une fenêtre de dialogue lancée via Alert-Controller de ionic-angular. La première fenêtre propose à l'utilisateur de changer de campus parmi ceux de Louvain-la-Neuve, Woluwe et Mons. Une fois validé, un appel à UserService est réalisé pour enregistrer ce choix dans le stockage. Un nouvel appel à Alert-Controller est alors fait afin d'afficher une fenêtre de dialogue qui propose à l'utilisateur de choisir sa langue entre l'anglais ou le français. Lorsque l'utilisateur valide son choix, TranslateService est appelé pour changer la langue de l'application. Ce service utilise des fichiers JSON qui contiennent tous les textes de l'application. En fonction de la langue sélectionnée, il va chercher le fichier correspondant pour en afficher les différents textes.

## 4.2 Campus

Cette section est consacrée aux fonctionnalités/modules concernant la vie sur le campus UCL. Afin de maintenir informé l'étudiant, elle comprend l'actualité de l'université, les événements

se déroulant sur le campus ainsi que l'horaire et locaux des différents sports et entraînements universitaires organisés par les différents campus de l'UCL. Ces trois modules tirent leurs informations de différents flux RSS qui seront expliqués dans les parties leur étant consacrées. Un flux RSS (Rich Site Summary ou Really Simple Syndication) n'est rien d'autre qu'un fichier XML qui est composé de balises encadrant du contenu. C'est avec ces balises qu'il est possible de savoir que tel élément est un titre, un lien, une image, etc. Afin de lire ce flux, il faut un script convertissant ce XML en HTML. Pour cela, il y a des agrégateurs de news ou des lecteurs RSS. Une personne voulant suivre un tel flux n'a qu'à s'abonner à ce flux via ce genre de service. L'application UCLCampus utilise le service gratuit de Yahoo : Yahoo Query Language (YQL) qui permet d'interroger, filtrer et combiner différents flux RSS. Il fonctionne avec des requêtes semblables aux requêtes SQL, mis à part qu'au lieu de filtrer une base de données, il filtre les données de services web (HTML, RSS ou XML). Lorsqu'il traite une requête, il accède aux sources de données sur Internet et transforme les données avant de renvoyer le résultat au format XML ou JSON. La requête à YQL et la récupération des informations renvoyées par celui-ci sont gérées par le service générique RSSService. Seule la partie Sports n'utilise pas YQL et donc pas UCLService car le format du flux RSS n'est pas encore au point et n'est donc pas accepté.

## News

Ce module regroupe l'ensemble de l'actualité de l'Université Catholique de Louvain. Il y a plusieurs éléments qui le compose :

- **NewsPage** est une view qui affiche la liste des news du campus ;
- **NewsDetailsPage** est une view qui affiche les détails d'une news sélectionnée par l'utilisateur ;
- **NewsItem** est un objet qui représente une news ;
- **NewsService** est un service qui transforme les données récupérées via un flux RSS en objets NewsItem ;
- **RSSService** est un service générique qui s'occupe de récupérer les informations du flux RSS via YQL.

Le front-end est implémenté avec les pages NewsPage et NewsDetailsPage. Le back-end est quant à lui formé par NewsService et RSSService. Le front-end récupère et traite des informations sous forme d'objets NewsItem prov-enant du backend qui quant à lui récupère et traite des informations au format JSON. Lors du chargement du module News, le premier élément à être appelé est le composant NewsPage. Cette page est divisée en 3 onglets : les études, la recherche et le service à la société (chacun venant d'un flux RSS différent). Chaque onglet contient une liste d'actualités. Au moment du chargement, NewsPage fait appel à ConnectivityService afin de vérifier que l'utilisateur possède une connexion internet avant d'appeler NewsService pour récupérer les informations correspondant à l'onglet sélectionné. Tout d'abord, NewsService fait lui-même appel à RSSService qui va s'occuper de faire la requête à YQL et de lui renvoyer les informations récupérées en format JSON. Comme dit précédemment, selon l'onglet, le flux est différent :

- Les études : <https://uclouvain.be/actualites/p1/rss>
- La recherche : <https://uclouvain.be/actualites/p2/rss>
- Le service à la société : <https://uclouvain.be/actualites/p3/rss>

Le lien de ces flux est transmis à RSSService par NewsService. Par la suite, NewsService s'occupe de transformer le résultat renvoyé en objet NewsItem pour chacune des news. Ce dernier est un objet qui se compose de plusieurs attributs :

- Description
- Lien vers la source de la news
- Titre
- Image



- “Hidden” qui sert à n’afficher que les news correspondantes lorsque l’utilisateur utilise la barre de recherche affichée en haut de NewsPage
- Date de publication

Finalement, une liste de ces NewsItem est renvoyée à NewsPage qui s’occupe de l’afficher. En haut de cette page se trouve une barre de recherche qui permet à l’utilisateur de filtrer la liste des évènements sur base de la chaîne de caractères entrée par l’utilisateur.

Lorsque l’utilisateur appuie sur une news, la page NewsDetailsPage est appelée. Cette page est appelée avec l’objet NewsItem correspondant, créé précédemment, en paramètre afin d’en afficher les détails. Il contient dans ses attributs l’ensemble des informations dont un utilisateur a besoin à propos d’une actualité, il n’est donc plus nécessaire de faire appel à NewsService.

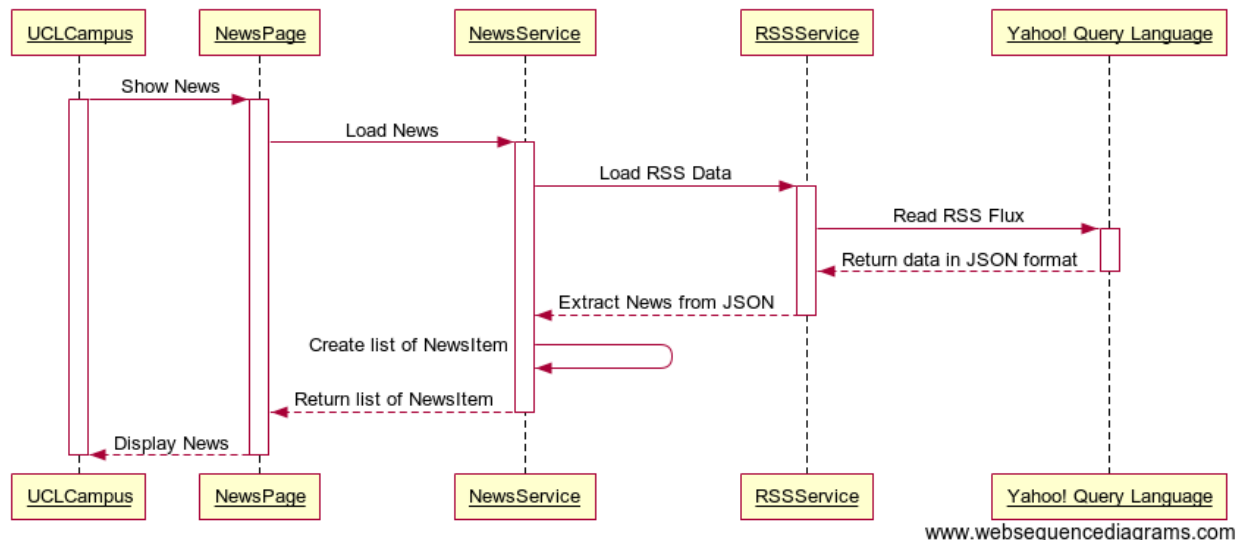


FIGURE 17 – Diagramme de séquence pour l’affichage des news

## Event

Ce module regroupe l’ensemble des évènements, regroupés par le kot à projet CarpeStudentem, qui prennent place sur le campus.

Les éléments qui le compose sont les suivants :

- **EventsPage** est une view qui affiche la liste des évènements récupérés ;
- **EventsFilterPage** est une view qui affiche les options de filtrage des évènements telles que la durée sur laquelle les évènements sont visibles et les catégories à afficher ;
- **EventItem** est un objet qui représente un évènement ;
- **EventsDetailsPage** est une view qui affiche les détails d’un évènement ;
- **EventsService** est un service lié à EventsPage qui opère le traitement des informations provenant du flux RSS ;
- **RSSService** est un service générique qui s’occupe de récupérer des informations venant d’un flux RSS ;
- **UserService** est un service externe au module qui est utilisé pour la gestion du stockage des évènements mis en favoris par un utilisateur.

La partie interface utilisateur de ce module est composée des views EventsPage et EventsDetailsPage alors que la partie logique est composée des services EventsService, RSSService et UserService. La première partie va récupérer et traiter des informations sous forme d’objets EventItem provenant de la seconde partie qui quant à elle récupère et traite des informations au format JSON.

Les éléments qui composent l’objet EventItem sont les suivants :

- catégorie ;
- titre ;
- date de début et date de fin ;
- localisation ;
- description ;
- image ;
- lien vers la source de l'article.

Pour la gestion de l'affichage de la liste des événements s'ajoute deux attributs, l'un pour permettre le masquage d'un objet de la liste et l'autre pour le marquage d'un objet de la liste comme favori de l'utilisateur.

Lors du chargement du module Event, le premier élément à être appelé est le composant **EventsPage**. Cette page est divisée en 2 onglets, la liste des événements et les favoris de l'utilisateur. Dans ces deux onglets, les utilisateurs ont accès à des actions en glissant les événements vers la gauche. Pour chaque événement, l'utilisateur pourra l'ajouter à son calendrier personnel présent sur le smartphone ainsi qu'ajouter ou supprimer l'événement de ses favoris stockés dans l'application. L'ajout d'un événement au calendrier est réalisé grâce au module Calendar, cette fonctionnalité provient du plugin cordova-plugin-calendar<sup>1</sup>. En cliquant sur le bouton calendrier, l'utilisateur voit apparaître une view permettant d'ajouter un événement au calendrier avec des options préremplies avec les détails de l'événement telles que le titre, la date, l'heure et le lieu. L'utilisateur peut enregistrer l'événement dans le calendrier ou d'abord y apporter des modifications. Une fois enregistré l'utilisateur revient à la view qui affiche la liste des événements.

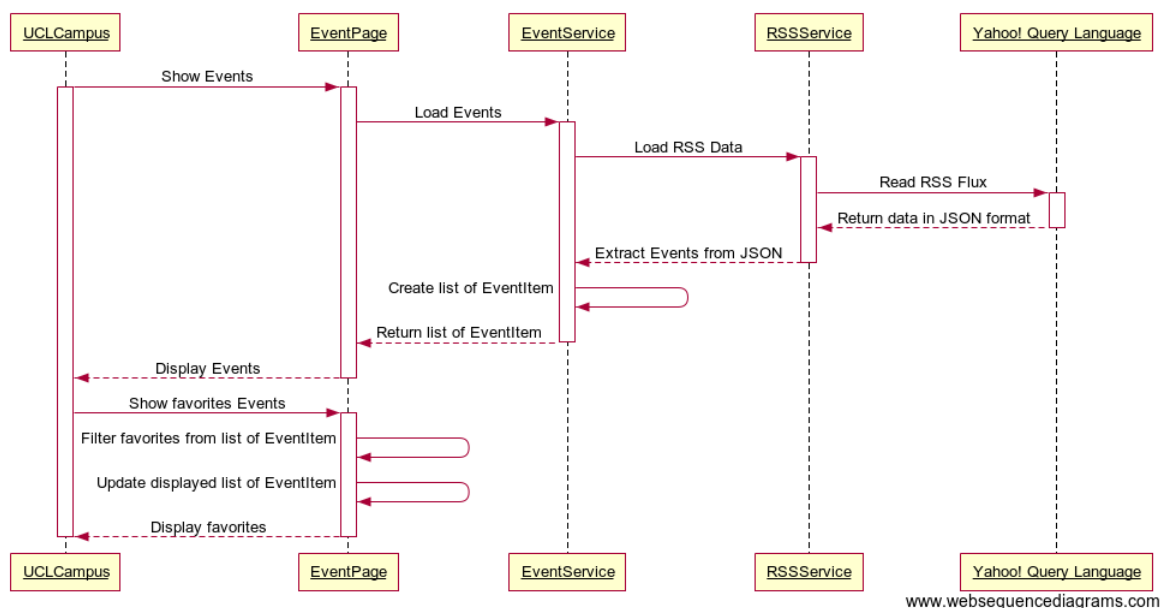


FIGURE 18 – Diagramme de séquence pour l'affichage des événements

Au moment du chargement, EventsPage fait appel à EventsService pour récupérer la liste des événements à afficher. La source du flux qui fournit les informations nécessaires à ce module vient du site <http://louvainfo.be/evenements/feed/calendar/> alimenté par le CarpeStudentem. EventsService fait lui-même appel à RSSService qui va s'occuper de faire la requête à YQL et de lui renvoyer les informations récupérées en format JSON. Le lien du flux est transmis à RSSService par EventsService. Une fois que les informations envoyées par RSSService sont reçues

1. <https://github.com/EddyVerbruggen/Calendar-PhoneGap-Plugin>

par `EventsService`, il va s'occuper de transformer le résultat en objet `EventItem` pour chaque event et de l'ajouter à la liste des événements à afficher. Une fois terminée, la liste de ces `EventItem` est renvoyée à `EventsPage` qui s'occupe dès lors de l'afficher. L'onglet Favoris ne fait pas appel aux services `EventsService` et `RSSService`, sa finalité est simplement d'afficher les événements marqués comme favoris par l'utilisateur dans la liste de départ. Pour comprendre le fonctionnement, il faut rappeler que la liste d'événements est une liste d'objets `EventItem` où chaque objet contient un attribut booléen "favori". La liste des favoris est alors tout simplement créée par le service `UserService`, externe au module, qui va récupérer et enregistrer des données propres à l'utilisateur. Les événements favoris font partie de ces données enregistrées pour l'utilisateur. `UserService` utilise le plugin `Ionic Storage` et non `SecureStorage`, celui-ci est suffisant pour gérer la liste de favoris ne représentant pas des données sensibles et qui ne nécessite pas d'une sécurité renforcée.

Lorsque l'utilisateur appuie sur un événement, la page **EventsDetailsPage** est appelée. Cette page est appelée avec l'objet `EventItem` correspondant, créé précédemment, en paramètre afin d'en afficher les détails. Il contient dans ses attributs l'ensemble des informations qu'un utilisateur a besoin à propos d'un événement, il n'est donc plus nécessaire de faire appel à `EventsService`.

**EventsFilerPage** est appelée quand l'utilisateur appuie sur le bouton de filtrage en haut à droite de la page `EventsPage`, il s'ouvre alors un volet de contenu par dessus. L'utilisateur aura droit à deux styles de filtres à appliquer à la liste initiale d'événements, un filtre ajustant la durée sur laquelle l'utilisateur désire connaître les événements disponibles (X mois) et un filtre qui permet de sélectionner les catégories d'événements que l'utilisateur désire voir dans la liste. Une fois le filtre appliqué, seuls les événements respectant les conditions seront affichés.

Comme autre filtre, il reste une barre de recherche en haut de la page `EventsPage` qui permet à l'utilisateur de filtrer la liste des événements sur base de la chaîne de caractères entrée par l'utilisateur. La décision par les mémorants de l'année académique 2016-2017 était garder cette barre sur la page `EventsPage` pour la facilité d'accès à l'utilisateur bien que cela soit un paramètre de filtrage.

## Sports

Ce module regroupe l'ensemble des sports disponibles sur le campus de l'utilisateur.

Les éléments qui le compose sont les suivants :

- **SportsPage** est une view qui affiche les sports disponibles sur 7 jours et séparés par jour ;
- **SportsFilterPage** est une view qui affiche le filtrage des sports où l'utilisateur sélectionnera les sports qu'il désire voir disponibles ;
- **SportItem** est un objet qui représente un sport ;
- **SportsService** est un service lié à `SportsPage` qui opère le traitement des informations provenant du flux RSS ;
- **RSSService** est un service générique qui s'occupe de récupérer des informations venant d'un flux RSS ;
- **UserService** est un service externe au module qui est utilisé pour la gestion du stockages des sports mis en favoris par un utilisateur.

Le front-end est implémenté avec la page `SportsPage`. Le b-ackend est quant à lui formé par `SportsService` et `RSSService`. Par le même principe que les deux modules précédents, le frontend récupère et traite des informations sous forme d'objets `SportItem` provenant du backend qui quant à lui récupère et traite des informations au format JSON.

Les éléments qui composent l'objet `EventItem` sont les suivants :

- nom du sport ;
- sexe autorisé à pratiquer le sport ;
- type de sport ;
- bâtiment-salle où a lieu le sport ;

- jour du sport ;
- heure de début et heure de fin du sport ;
- remarque associée au sport.

Pour la gestion de l’affichage de la liste des sports s’ajoute deux attributs, l’un pour permettre le masquage d’un objet de la liste et l’autre pour le marquage d’un objet de la liste comme favori de l’utilisateur.

Lors du chargement du module Sports, le premier élément à être appelé est le composant SportsPage. Cette page est divisée en 3 onglets, la liste des sports communs à tous, la liste des sports des équipes universitaires et les favoris de l’utilisateur. Dans ces trois onglets, les utilisateurs ont accès aux mêmes actions que pour les évènements, c’est à dire l’ajout à l’agenda personnel et l’ajout ou suppression d’un sport aux favoris. L’ajout d’un sport au calendrier est réalisé grâce au module Calendar suivant le même principe qu’expliqué dans le module Évènement.

Au moment du chargement, SportsPage fait appel à SportsService pour récupérer la liste

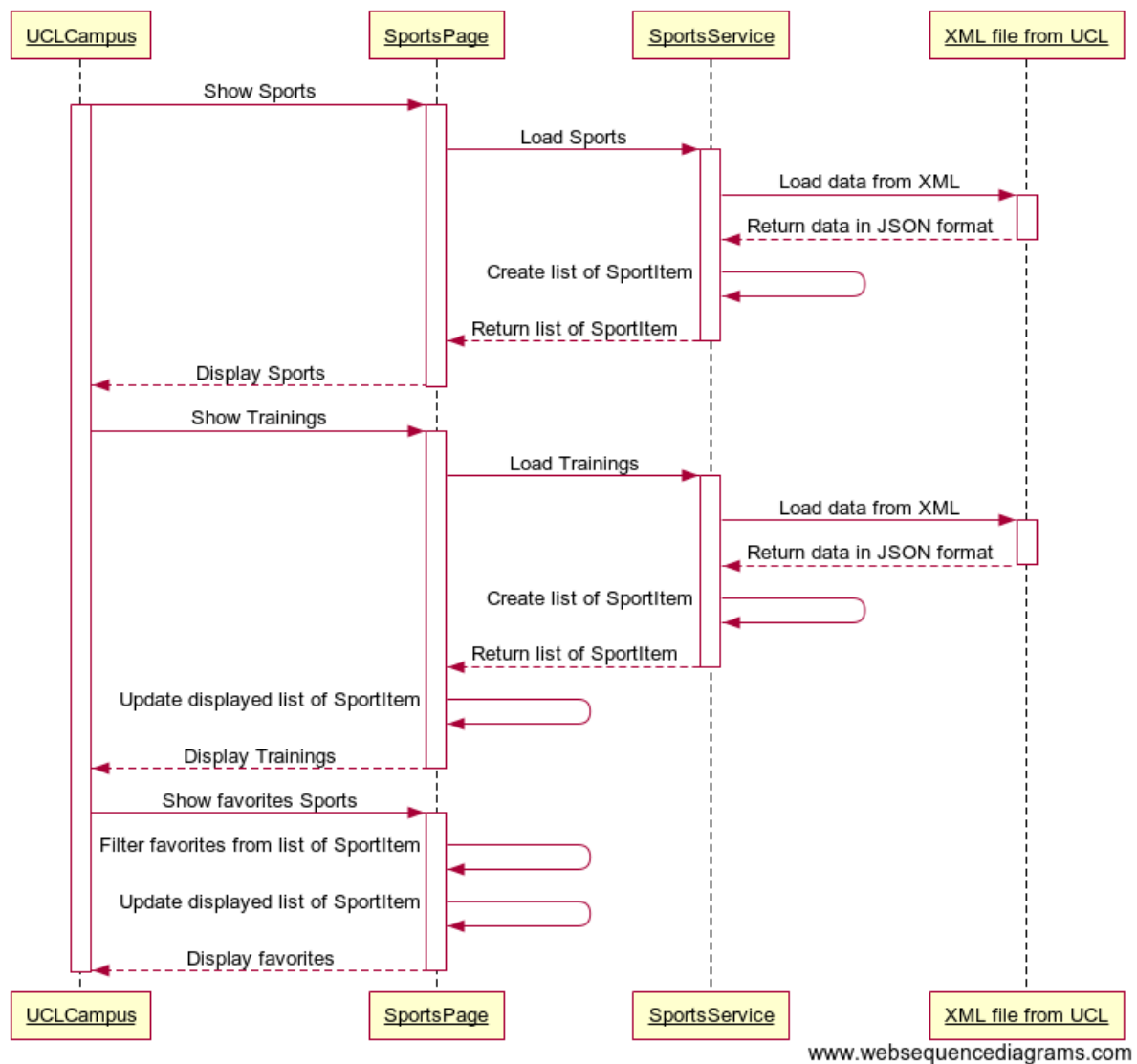


FIGURE 19 – Diagramme de séquence pour l’affichage des sports

des sports communs à tous à afficher. Les sources des flux qui fournissent les informations nécessaires à ce module viennent de quatre sites, celui sélectionné pour la liste des sports communs

dépendra de la variable campus de l'utilisateur initialisée lors du choix initial à la première ouverture de l'application ou changée à l'aide des paramètres. Le flux des sports des équipes universitaires quant à lui est unique.

Ces quatre sites sont les suivants :

- Louvain-la-Neuve : `http://ucl-fms01.sipr.ucl.ac.be:82/ucl_sport/rsssport.php?action=t1` ;
- Woluwe : `http://ucl-fms01.sipr.ucl.ac.be:82/ucl_sport/rsssport.php?action=t3` ;
- Mons : `http://ucl-fms01.sipr.ucl.ac.be:82/ucl_sport/rsssport.php?action=t5` ;
- Equipe universitaire : `http://ucl-fms01.sipr.ucl.ac.be:82/ucl_sport/rsssport.php?action=t7` ;

SportsService fait lui-même appel à RSSService qui va s'occuper de faire la requête à YQL et de lui renvoyer les informations récupérées en format JSON. Le lien du flux est transmis à RSSService par SportsService. Une fois que les informations envoyées par RSSService sont reçues par SportsService, il va s'occuper de transformer le résultat en objet SportItem pour chaque sport et de l'ajouter à la liste des sports à afficher. Une fois terminée, la liste de ces SportItem est renvoyée à EventsPage qui s'occupe dès lors de l'afficher. Contrairement au module Event, toutes les informations au sujet des sports sont affichées au sein même de la liste des sports, aucune page de détails ne peut être ouverte.

L'onglet Équipes universitaires fait appel aux services SportsService pour récupérer la liste des sports des équipes universitaires. Le fonctionnement est exactement le même que pour les sports communs à tous.

L'onglet Favoris ne fait pas appel aux services SportsService et RSSService, sa finalité est simplement d'afficher les sports marqués comme favoris par l'utilisateur dans l'une des deux listes des sports. Le fonctionnement pour arriver à cela est le même qu'expliqué dans le module Event, les sports favoris font partis des données enregistrées pour l'utilisateur.

**SportsFilterPage** est appelée quand l'utilisateur appuie sur le bouton de filtrage en haut à droite de la page SportsPage, il s'ouvre alors un volet de contenu par dessus. L'utilisateur pourra dès lors sélectionner les sports qu'il désire voir dans la liste. Une fois le filtre appliqué, seuls ceux respectant ces conditions seront affichés dans la liste des sports. Comme autre filtre, il reste une barre de recherche en haut de la page SportsPage qui permet à l'utilisateur de filtrer les listes des sports sur base de la chaîne de caractères entrée par l'utilisateur. Elle reste sur cette page car elle est plus facile d'accès pour l'utilisateur.

### 4.3 Études

Cette section est consacrée aux fonctionnalités/modules fournissant les diverses informations utiles à un étudiant de l'UCL. L'étudiant peut ainsi avoir une liste de toutes les bibliothèques de l'UCL et en avoir les heures d'ouverture et informations de contacts. Il y a également un module permettant de récupérer son programme de cours et d'avoir l'horaire des cours sélectionnés. Pour finir, il y a une partie regroupant les informations propres à l'UCL telles que le Service Desk, la documentation sur la carte d'accès et sur le réseau WiFi eduoram ainsi qu'un accès au répertoire UCL afin de rechercher les informations à propos d'un membre du personnel de l'UCL. La plupart de ces informations (bibliothèques, répertoire UCL et programme de cours) sont récupérées grâce à des requêtes à l'API Manager de l'ESB WSO2 qui seront expliquées plus en détail dans la suite. Le service s'occupant des requêtes vers l'API manager, lors de son premier appel, fait une requête afin d'avoir un token qu'il utilisera par la suite pour s'identifier à chacune des requêtes nécessaires.

#### Bibliothèques

Ce module regroupe les différentes bibliothèques de l'Université Catholique de Louvain avec l'ensemble des informations dont l'utilisateur doit disposer.

Plusieurs éléments le composent :

- **LibrariesPage** est une view qui affiche la liste des différentes bibliothèques récupérées ;
- **LibraryDetailsPage** est une view qui affiche les détails d'une bibliothèque sélectionnée par l'utilisateur ;
- **LibraryItem** est un objet qui représente une bibliothèque ;
- **LibrariesService** est un service qui transforme les données récupérées en objet LibraryItem ;
- **Wso2Service** est un service générique qui s'occupe de récupérer des informations via une requête à l'API Manager de WSO2.

LibrariesPage et LibraryDetailsPage font partie de l'interface utilisateur tandis que la partie logique est composée des services LibrariesService et Wso2Service. La première partie va récupérer et traiter les informations sous forme d'objets LibraryItem provenant de la partie logique avant de les afficher. Tandis que la partie logique récupère et traite les informations au format JSON.

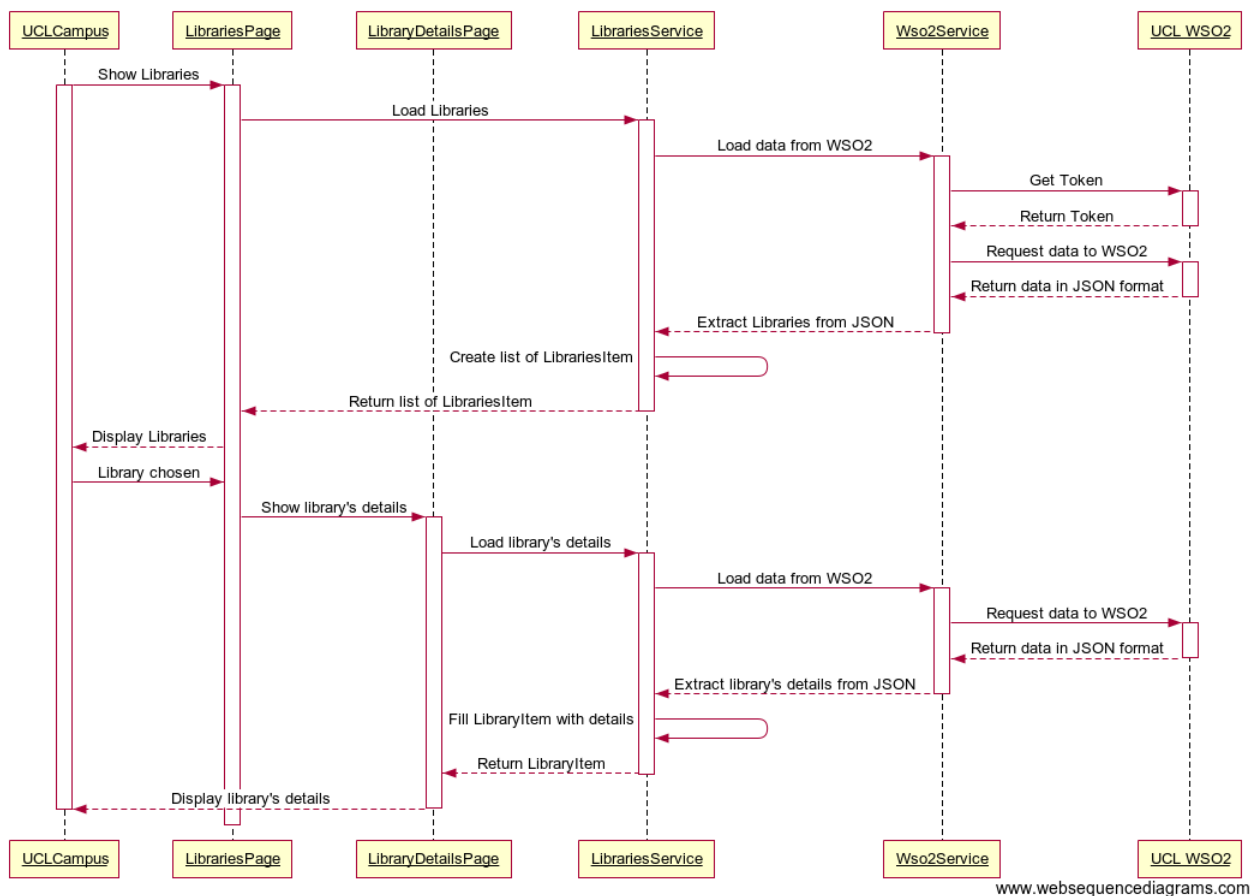


FIGURE 20 – Diagramme de séquence pour l'affichage des bibliothèques

Lors du chargement du module Bibliothèques, le premier composant à être appelé est le composant LibrariesPage. Cette page affiche une liste qui reprend toutes les bibliothèques de l'UCL. Lors du chargement, LibrariesPage fait appel à LibrariesService pour récupérer cette liste. Tout d'abord, LibrariesService fait lui même appel à Wso2Service en lui passant les différents éléments nécessaires à la requête qu'il fera à l'API Manager. Cette requête sera faite à l'API Libraries qui nous renverra une liste de bibliothèques au format JSON. Ensuite, lorsque Wso2Service transmet les informations à LibrariesService, celui-ci va transformer ce résultat en objet LibraryItem pour chaque bibliothèque.

Les éléments qui composent l'objet LibraryItem sont les suivants :

- Id de la bibliothèque ;

- Nom de la bibliothèque ;
- LocationId ;
- MapLocation ;
- Numéro de téléphone et Email ;
- Site web ;
- Heures d'ouverture en période de cours, en période d'examen et en période de vacances ;
- Note à propos des heures d'ouverture ;
- Dates de fermeture.

L'attribut MapLocation est lui même un objet comprenant les éléments suivants :

- Titre ;
- Code ;
- Latitude ;
- Longitude ;
- Adresse.

Les différents attributs heures d'ouverture sont quant à eux des objets TimeSlot, leurs attributs sont les suivants :

- Jour ;
- Heure de début ;
- Heure de fin.

À cette phase de chargement, l'objet instancié par LibrariesService n'est pas complet, le fichier JSON reçu ne contenant que la paire nom/id pour chaque bibliothèque. Les autres éléments sont mis à jour à l'aide d'autres requêtes lorsque l'utilisateur sélectionne une bibliothèque particulière. Cette approche permet de ne pas charger les informations de toutes les bibliothèques en une seule fois.

Finalement, une liste de ces LibraryItem est renvoyée à LibrariesPage qui s'occupe d'afficher la liste de toutes les bibliothèques.

Lorsque l'utilisateur sélectionne une bibliothèque particulière en cliquant dessus, la page LibraryDetailsPage est appelée. Son appel se fait avec comme paramètre l'objet LibraryItem correspondant, créé précédemment. Pour rappel, à ce stade l'objet n'est pas complet, seuls son nom et son id sont corrects. La page LibraryDetailsPage appelle alors LibrariesService qui agit comme expliqué précédemment mais la requête sera différente, celle-ci demande les détails d'une bibliothèque en fonction de son Id récupéré dans l'objet LibraryItem passé en paramètre. Pour terminer, LibrariesService remplit l'objet LibraryItem avec les données restantes avant de le transmettre à LibrariesDetailsPage qui s'occupe d'afficher toutes les informations à propos de cette bibliothèque.

Une fois les détails d'une bibliothèque affichés, l'objet LibraryItem reste complété tant que le module est ouvert. Si l'utilisateur décide dès lors de revoir ces détails durant cette fenêtre temporelle, la requête pour compléter les informations de cette bibliothèque ne sera pas exécutée une deuxième fois.

Il est à noter qu'au lancement de LibrariesPage et de LibraryDetailsPage, un appel à ConnectivityService est fait afin de vérifier que l'utilisateur possède une connexion internet afin de faire les diverses requêtes.

## Mes études

Ce module implémente deux fonctionnalités : l'affichage de l'horaire des différents cours sélectionnés et ajoutés à la liste ainsi que la récupération du programme de cours d'un étudiant après son identification avec ses identifiants UCL.

Ce module est composé de plusieurs éléments :

- **StudiesPage** est une view qui affiche la liste des cours ajoutés, le formulaire d'identification de l'étudiant ainsi que le programme de cours récupéré après identification ;
- **ModalProjectPage** est une view qui affiche la liste des projets ADE disponible ;

- **CoursePage** est une view qui affiche l'horaire des différents travaux pratiques et cours magistraux d'un cours sélectionné par l'utilisateur ;
- **Course** est un objet qui représente un cours ;
- **AdeProject** est un objet qui représente un projet ADE ;
- **Activity** est un objet qui représente un cours magistral ou le T.P. d'un cours ;
- **CourseService** est un service qui transforme les données récupérées en des objets Activity ;
- **AdeService** est un service qui récupère l'horaire des cours magistraux et travaux pratiques d'un cours ;
- **StudiesService** est un service qui transforme les données récupérées en des objets AdeProject ;
- **StudentService** est un service qui transmet à Wso2Service les données nécessaires à la requête ;
- **Wso2Service** est un service générique qui s'occupe de récupérer des informations via une requête à l'API Manager de WSO2.

Parmi ces éléments, StudiesPage, ModalProjectPage et CoursePage font partie du front-end. StudiesPage va récupérer, afficher et traiter les informations sous forme d'objet Course, ModalProjectPage va faire pareil mais avec des objets AdeProject et CoursePage avec des objets Activity. Ces objets proviennent de la partie logique/back-end. Les éléments CourseService, AdeService, StudiesService, StudentService et Wso2Service font partie du back-end. Ce sont ces services qui vont s'occuper de faire la requête vers l'API Manager (Wso2Service et StudentService) pour identifier l'étudiant et récupérer les informations au format JSON ou de faire une requête à ADE (AdeService) avant de transformer les données en objets Activity (CourseService) et en objets AdeProject (StudiesService) . Lorsque le module Études est lancé, le premier composant à être appelé est la view StudiesPage qui, avant d'apparaître, va appeler la view ModalProjectPage afin que l'utilisateur choisisse un projet ADE en cours. Cette dernière view fait appel au service StudiesService afin de récupérer la liste des projets disponibles. Pour cela, ce service fait lui-même appel à AdeService qui fait une requête à ADE Expert qui va lui répondre au format JSON une liste des projets. AdeService renvoie cette liste à StudiesService qui s'occupe d'instancier des objets AdeProject avec les données récupérées avant de les renvoyer à ModalProjectPage.

Cet objet AdeProject est composé des attributs suivants :

- Id du projet ;
- Nom du projet.

Une fois le projet sélectionné par l'utilisateur, la view StudiesPage s'affiche et le projet sélectionné est enregistré grâce au plugin @ionic/storage. La view StudiesPage est composée de deux onglets. Le premier onglet est l'onglet Horaire où il est possible pour l'utilisateur d'ajouter manuellement des cours afin d'en avoir l'horaire. Pour cela, l'utilisateur doit rentrer le sigle du cours et lui donner un nom au choix. Un objet Course est alors créé et le plugin @ionic/storage est utilisé pour enregistrer cet objet dans la mémoire.

Cet objet est composé de ces attributs :

- Acronyme du cours ;
- Nom du cours ;
- Activités qui est un tableau d'objets Activity rempli par la suite.

Lorsque l'utilisateur sélectionne un cours parmi la liste de ceux ajoutés, la view CoursePage est appelée. Celle-ci fait appel au service CourseService qui lui même appelle AdeService avec l'Id du cours. Ce dernier fait une requête à ADE qui lui renvoie un JSON avec la liste des activités (cours magistraux et travaux pratiques). AdeService renvoie ensuite les données à CourseService qui s'occupe de créer un tableau d'objets Activity avant de les renvoyer à la view qui divisera le résultat en deux onglets, cours magistraux et travaux pratiques.

L'objet Activity est composé de plusieurs attributs également :

- Type d'activité (cours magistral ou T.P.) ;



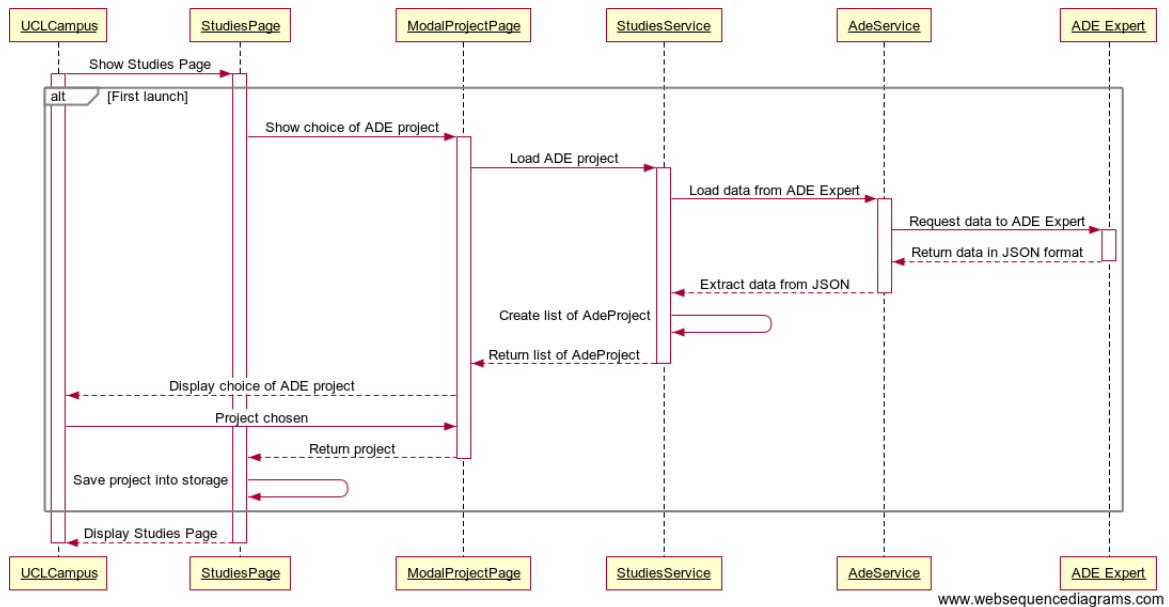


FIGURE 21 – Diagramme de séquence pour l’affichage de la page études

- Professeurs ;
- Etudiants admis ;
- Heure de début ;
- Heure de fin ;
- Auditoire.

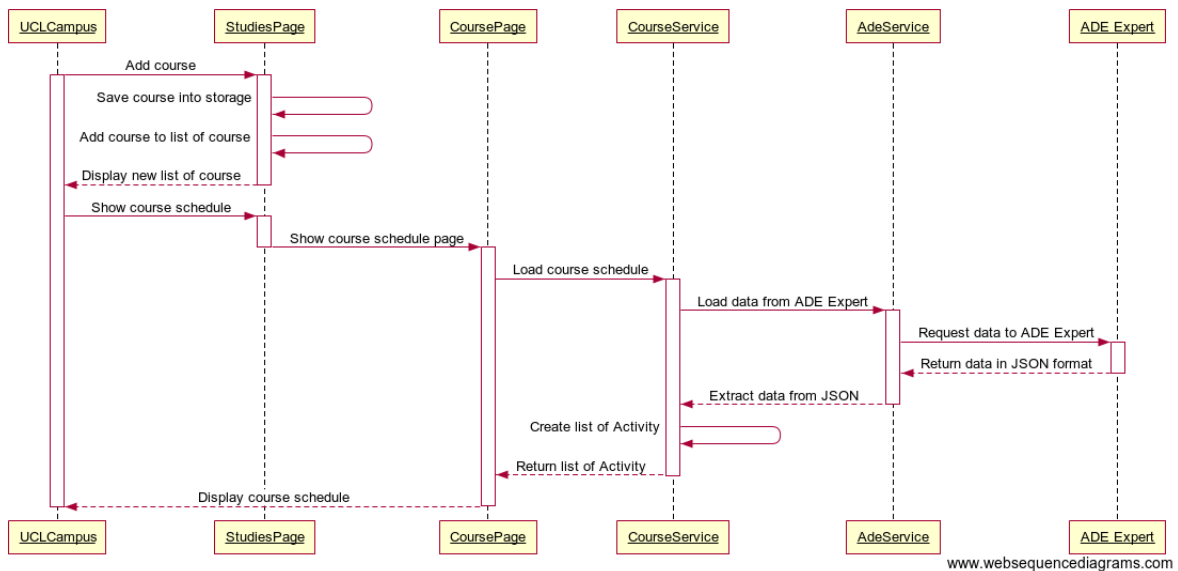


FIGURE 22 – Diagramme de séquence pour l’ajout d’un cours et de l’affichage de l’horaire

Sur cette view, il est possible pour l’utilisateur de faire glisser vers la gauche une session afin de l’ajouter au calendrier natif via le plugin @ionic-native/calendar. Il est également possible sur la view StudiesPage de faire la même chose afin d’ajouter toutes les sections d’un cours à ce calendrier. Il y a ensuite le second onglet de la view StudiesPage : l’onglet Programme de cours. Sur cet onglet, il est proposé à l’utilisateur d’utiliser ses identifiants globaux afin de récupérer son programme de cours. Une fois le formulaire validé, la view appelle le service Wso2Service qui

va faire une requête à l'API manager pour avoir un token spécifique à l'étudiant. Une fois cela fait, la view appellera StudentService qui lui-même rappellera Wso2Service afin de récupérer le programme de cours de l'étudiant enregistré, via l'API My, avant de les renvoyer vers la view qui s'occupera de l'afficher. Une fois le programme affiché, l'utilisateur peut ajouter un cours à l'autre onglet afin d'en avoir l'horaire, en faisant glisser ce cours vers la gauche.

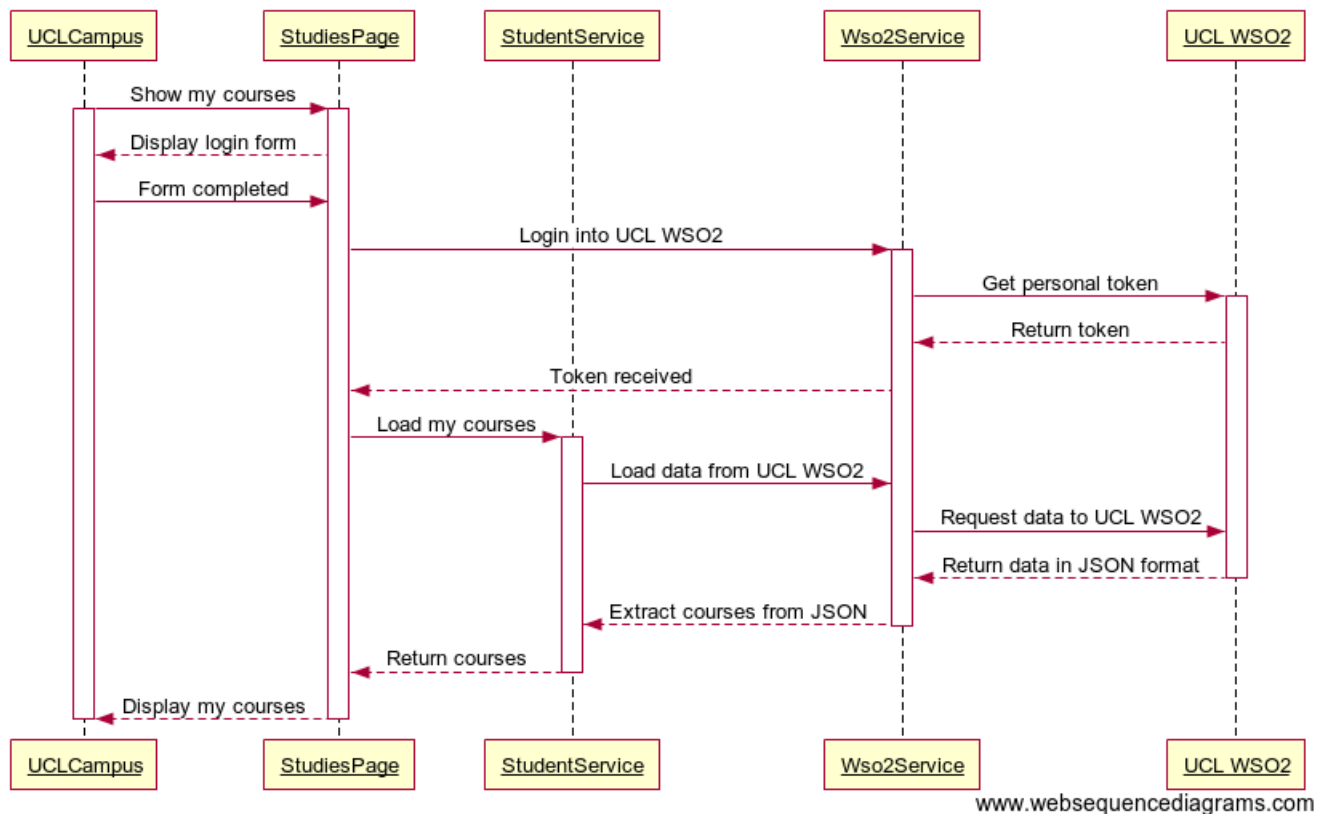


FIGURE 23 – Diagramme de séquence pour l'affichage du programme de cours

## Support

Ce module est divisé en quatre parties distinctes :

1. **Service Desk** qui définit brièvement ce qu'est le Service Desk et fournit les informations de contact de celui-ci ;
2. **SmartCard** explique à quoi sert la carte d'accès à l'UCL, la composition de celle-ci aux recto et verso. Il y a également différent lien à suivre en cas de problème avec cette carte et une explication sur son activation ;
3. **WiFi** qui présente les différents réseaux WiFi de l'UCL et leurs utilités. Il est expliqué quel réseau sert à quoi, pour qui il est destiné et comment s'y connecter. Il y a également des redirections vers des PDF qui fournissent plus de détails. Il est par la même occasion expliqué la couverture du service dans les différents campus ;
4. **Répertoire** qui fournit un accès au répertoire UCL afin de rechercher un membre du personnel et d'en avoir les informations utiles.

Les trois premières parties sont codées en dur. Toutes les informations s'y trouvant proviennent du portail UCL. Il en a été jugé qu'il n'était pas nécessaire de les récupérer d'une autre manière plus automatique, la rapidité et la simplicité de mise en place de ces 3 premières parties étaient des avantages de choix. De plus, ces informations sont peu sujettes à un changement régulier

et si un changement est observé il suffira à la communauté du projet de changer directement l'information dans la partie codée en dur pour qu'elle soit à nouveau pertinente avec le portail. Tandis que le répertoire nécessite plus d'explications. Le répertoire est composé de plusieurs éléments :

- **HelpDeskPage** est une view qui affiche les quatre parties mentionnées précédemment ;
- **EmployeeDetailsPage** est un view qui affiche les informations utiles à propos d'un membre du personnel recherché et sélectionné par l'utilisateur ;
- **EmployeeItem** est un objet qui représente un membre du personnel de l'UCL ;
- **RepertoireService** est un service qui transforme les données récupérées en objets EmployeeItem ;
- **Wso2Service** est un service générique qui s'occupe de récupérer des informations via des requêtes à l'API manager de WSO2.

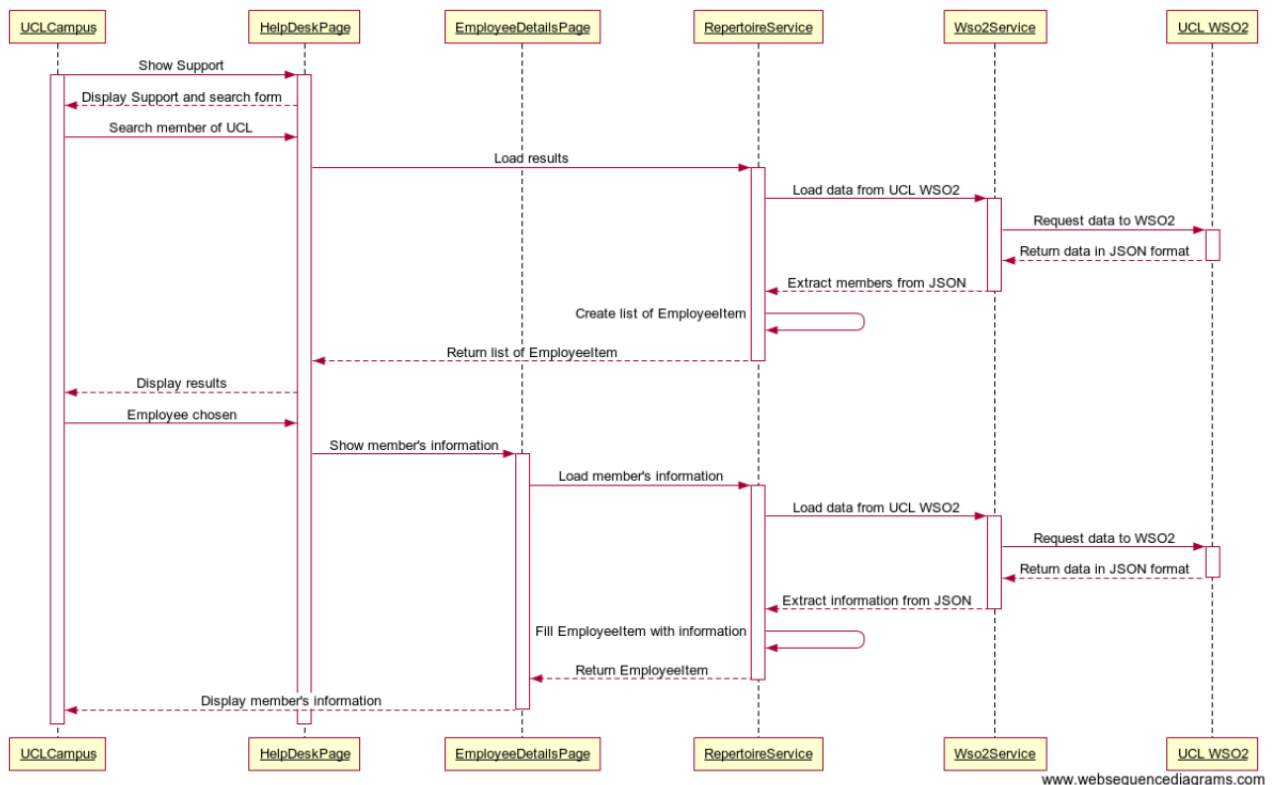


FIGURE 24 – Diagramme de séquence pour l’affichage du support et la recherche d’un membre de l’UCL

HelpDeskPage et EmployeeDetailsPage font partie du front-end tandis que RepertoireService et Wso2Service font partie du back-end. Le front-end s’occupe de récupérer et traiter les données sous forme d’objet EmployeeItem provenant de la partie logique ainsi que d’afficher les diverses informations écrites directement dans le code. Le back-end, quant à lui s’occupe de faire les requêtes à l’API manager avant de récupérer et traiter les données au format JSON pour ensuite les transformer en EmployeeItem. Lors du chargement du module Support, le premier composant à être chargé est la view HelpDeskPage. Cette page affiche les parties mentionnées précédemment, c’est-à-dire le Service Desk, la SmartCard, les réseaux WiFi et le répertoire UCL. Ce dernier permet de rechercher un membre du personnel UCL via son nom ou son prénom. Une fois le bouton “rechercher” appuyé, le service ConnectivityService est appelé afin de vérifier que l’utilisateur est bien connecté à internet avant que la view appelle RepertoireService avec en argument le nom et/ou le prénom recherché. Ce service va lui même appeler Wso2Service qui s’occupera de faire la requête à l’API manager, via l’API Directories, avant de lui renvoyer le

résultat au format JSON. Lorsque RepertoireService récupère les données, il va les transformer en objet EmployeeItem.

Cet objet a les attributs suivants :

- Matric\_fgs;
- Prénom et Nom du membre du personnel;
- Email;
- Départements;
- Adresse;
- Contacts professionnels sous forme de tableau;
- Contrats sous forme de tableau;
- Genre;
- URL d'une photo.

Lorsque RepertoireService renvoie l'objet à la view, celle-ci affiche la liste des résultats. Lorsque l'utilisateur appuie sur un membre du personnel, la view EmployeeDetailsPage est appelée afin d'afficher le résultat à l'utilisateur. Avant de l'afficher, une nouvelle vérification de la connexion est faite via le service ConnectivityService. Si l'utilisateur est bel et bien connecté à internet, RepertoireService est appelé avec le fgs de la personne sélectionnée afin de faire à nouveau une requête à l'API manager pour avoir les détails de cet employé. Lorsque EmployeeDetailsPage reçoit les détails, elle s'occupe de les afficher.

## 4.4 Outils

Cette section est consacrée aux fonctionnalités/modules concernant les outils qui peuvent servir à l'étudiant. Nous y retrouvons notamment une carte interactive grâce à laquelle toute personne peut se situer sur le campus ainsi que situer des points d'intérêts tels que les parkings, les bibliothèques, les auditoriums et les restaurants universitaires. Une autre fonctionnalité est un accès direct à plusieurs applications externes, d'une part l'étudiant peut ouvrir l'application des restaurants universitaires et d'une autre part il a le choix d'ouvrir une des trois applications de transports en commun soit SNCB, soit NextRide ou soit Commuty. Le dernier module de cette section est Guindaille2.0, comme demandé par Univers Santé ce module reprend leur campagne, c'est à dire l'ensemble de leurs pictogrammes avec une description associée, leur effectomètre et les informations secours.

### Carte

À l'aide de ce module, si l'utilisateur dispose d'une connexion à un réseau internet, il a accès à une carte centrée sur sa position. Sa position est également représentée par marqueur sur la carte. Selon la variable campus, une liste avec de multiples points d'intérêts est disponible, l'utilisateur peut sélectionner ceux qu'il désire afficher sur la carte, une fois sélectionnés ceux-ci seront visibles à l'aide d'un marqueur. Cependant si aucune connexion n'est disponible, l'application prévient l'utilisateur qu'aucune connexion n'est détectée et la carte est gelée jusqu'au retour de la connexion.

L'implémentation de cette fonctionnalité utilise les services Google Maps, une connexion à un réseau internet est obligatoire pour leur fonctionnement. Les API Javascript et mobiles de Google Maps, plus précisément Google Maps Android API et Google Maps SDK, sont les API nécessaires pour ce module. L'API Google Maps pour mobiles permet d'avoir un temps de réponse plus rapide et une utilisation du réseau plus faible que l'API Google Maps Javascript.[57] Ceci est dû au fait que l'API Javascript charge des images pour créer la carte alors que les API mobiles chargent des vecteurs de données pour dessiner les tuiles de la carte. La charge sur le réseau est donc réduite grâce à l'utilisation d'un vecteur au lieu d'images. À titre indicatif, une carte composée de neuf tuiles représente 113KB de données téléchargées alors que l'utilisation d'un

vecteur de données diminue cette consommation à moins de 12KB. Le temps de réponse est directement en corrélation avec cette charge réseau, il est simple de déduire que le temps d'afficher la carte sera plus important si le poids des données est élevé.

Un second avantage à souligner est l'absence de quotas pour les API mobiles de Google Maps. L'API Javascript est quant à elle soumise à un quota de 25 000 requêtes par tranche de 24h, une fois passé une facturation s'impose.[58] Le plugin cordova-plugin-googlemaps est utilisé pour interagir avec les services Google Maps pour les plateformes Android et iOS.

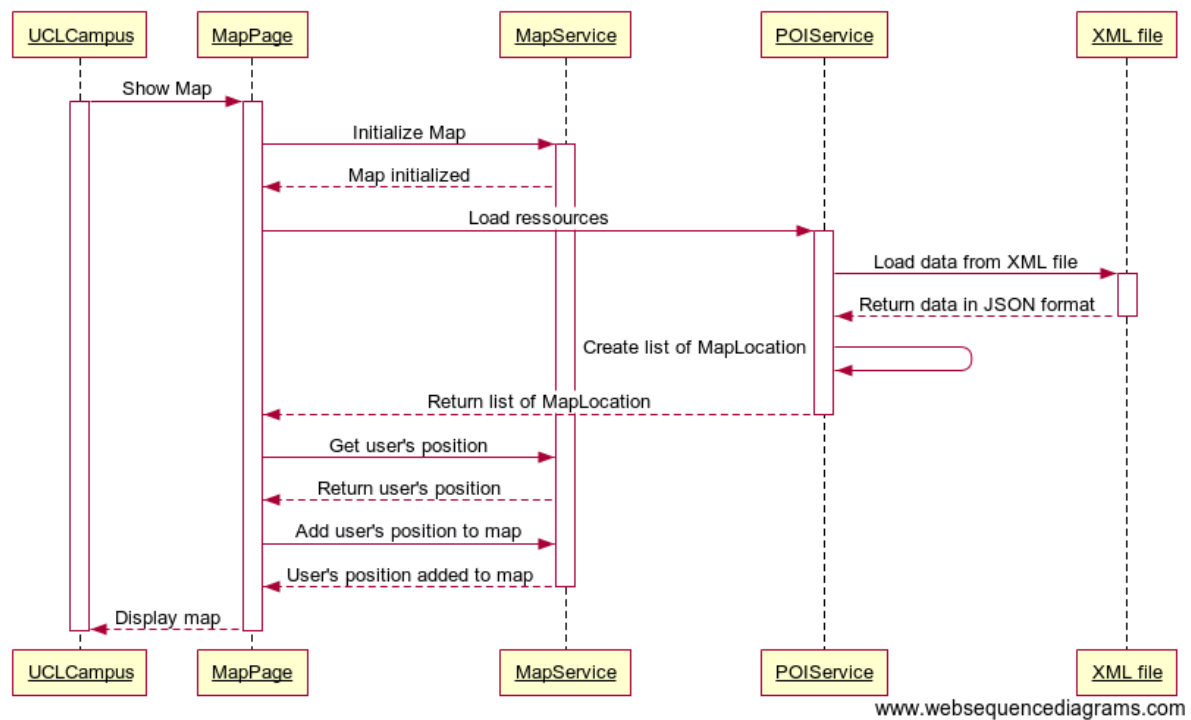


FIGURE 25 – Diagramme de séquence pour l'affichage de la carte

Pour ce module MapPage fait partie de l'interface utilisateur tandis que la partie logique est composée des services MapService et PoiService. Lors du chargement du module Carte, le premier composant à être appelé est le composant MapPage, cette page est en charge de l'affichage de la carte et de tous les éléments qui la composent, c'est-à-dire le marqueur de la position de l'utilisateur, les marqueurs des points d'intérêts sélectionnés et la liste des points d'intérêts du campus en vigueur. Lors du chargement, MapPage fait appel à MapService pour initialiser la carte à afficher ainsi qu'à POIService pour le chargement de la liste de ses points d'intérêt. Ces deux requêtes sont gérées par des Promises dues à leur caractère asynchrone. Une fois la réponse reçue de la part des deux requêtes, MapPage crée un objet MapLocationItem propre à la position de l'utilisateur pour l'ajouter à la liste des localisations à afficher.

Les éléments qui composent l'objet MapLocationItem sont les suivants :

- nom associé au point d'intérêt ;
- adresse de la position ;
- latitude et longitude de la position ;
- sigle.

Chaque localisation que l'on peut afficher sur la carte est représentée par un objet MapLocationItem. Une fois l'objet de la position de l'utilisateur créé, la page MapPage envoie une nouvelle requête au service MapService pour ajouter un marqueur le représentant sur la carte.

Le service MapService permet de créer et de maintenir l'objet map à afficher peut importe la plateforme de l'utilisateur. Son fonctionnement est le suivant. Pour vérifier que l'utilisateur est bien connecté à un réseau internet, ce service appelle un service externe au module quand il est appelé. Le service externe se nomme ConnectivityService.

Une fois cette vérification terminée, il entame une seconde vérification au sujet de la plateforme utilisée afin d'adapter l'origine du chargement de la carte. L'origine du chargement sera différente si la plateforme est mobile ou ne l'est pas. Si elle est mobile, le plugin cordova est appelé tandis que c'est l'API Javascript qui est appelée dans le cas contraire.

Dès que la deuxième vérification est terminée et que le service est connecté, ce dernier appelle le plugin Ionic Geolocation pour récupérer la position de l'utilisateur grâce à sa latitude et sa longitude. Le service instancie alors l'objet map avec deux paramètres, MapOptions et MapElement. Le premier donne les options de la carte telles que le zoom initial, la position de l'utilisateur sur laquelle la carte sera centrée et le type de carte qui dans le cas d'UCLCampus est une roadmap. Le second objet est une référence directe vers le conteneur HTML dans lequel est affichée la carte. À partir de ce moment, la carte étant initialisée et affichée, le rôle de MapService se réduit à la gestion des marqueurs et le contrôle de la connexion.

La gestion des marqueurs prend place quand l'utilisateur veut activer un point d'intérêt sur la carte. Un objet Marker est alors créé et ajouté à la liste des marqueurs à afficher. Le service référence alors l'objet Map à ce Marker pour qu'il soit affiché sur la carte. Pour éviter la création et la destruction successive d'un même Marker, sa suppression est gérée en rendant sa visibilité "false". L'objet reste présent dans la liste des marqueurs même s'il n'est plus affiché sur la carte. Le contrôle de la connexion par le service MapService est assuré durant l'utilisation du module grâce à deux EventListeners ajouté au module par l'appel à la méthode document.addEventListener(). Ils sont nommés respectivement "online" et "offline". MapService instancie, si nécessaire, et présente la carte quand l'évènement "online" se produit alors qu'il prévient l'utilisateur qu'il n'est pas connecté et désactive l'utilisation de la carte quand l'évènement "offline" se produit.

Le service POIService permet quant à lui de charger les points d'intérêt à partir d'un fichier JSON qui se situe dans les assets de l'application UCLCampus. Le fichier JSON chargé est celui propre au campus de l'utilisateur défini grâce à sa variable "campus". Les points d'intérêt sont ceux présentés à l'utilisateur. Il existe 3 fichiers JSON différents, un par campus. Pour l'instant, les campus de Louvain-la-Neuve, Mons et Woluwe sont représentés. L'ajout de points d'intérêt pour un campus est simple pour les développeurs du projet UCLCampus, il suffit de les ajouter au fichier JSON correspondant en veillant à respecter sa structure. Il est également possible de créer un nouvel ensemble de points d'intérêt, par exemple les "Infrastructures sportives" en respectant également la structure du fichier JSON. L'ajout d'un campus quant à lui se réalisera par la création d'un nouveau fichier JSON.

## Resto U

Cette fonctionnalité a pour but principal d'informer l'utilisateur des heures d'ouverture et du menu hebdomadaire du restaurant. Les restaurants universitaires, partenaire de l'UCL, ont déjà déployé une application fournissant tout ce qui est nécessaire et est nommée Resto4U. Une solution aurait pu être d'intégrer le contenu qui y est disponible au sein même de l'application UCLCampus mais la solution choisie est de permettre le lancement de Resto4U sur l'application UCLCampus. Il y a des avantages et des inconvénients au lancement d'applications externes.

D'une part, cela permet :

- de réutiliser des fonctionnalités spécifiques existantes sans devoir les recréer, cela représente un gain de temps important, la solution est directement apportée au problème ;
- d'éviter de concurrencer inutilement une application n'apportant une solution qu'à un problème spécifique ;
- d'éviter la duplication des fonctionnalités pour les applications de l'environnement UCL

- et par la même occasion de faire connaître ces applications à un plus large public ;
- de laisser la responsabilité totale aux entités externes quant aux contenus et aux informations des applications.

Mais d’une autre part, l’utilisateur est obligé de télécharger différentes applications pour pouvoir accéder à tous les services qui lui sont proposés. Ajouté à cela il y a un contraste important entre les interfaces et le design des différentes applications, il n’y a aucune convention entre les applications proposées, chacune a été développée par des sociétés différentes. L’utilisateur sera obligé de se faire à ce changement radical étant donné qu’une fois une application externe lancée, l’utilisateur quitte momentanément l’application UCLCampus pour l’utiliser.

Tout ce qui permet de lancer une application externe se situe dans la page où le bouton correspondant à cette application est affiché.

Pour l’application Resto4U, la méthode de lancement de celle-ci intervient soit lors de l’appel à la méthode d’ouverture `openRootPage()` quand l’utilisateur utilise le menu latéral, soit lors de l’appel à la méthode `changePage()` quand l’utilisateur utilise le bouton situé sur la page d’accueil. Les méthodes `openRootPage()` et `changePage()` vérifient si la page ouverte implique le lancement d’une application externe et appelle dans ce cas la méthode `launchExternalApp()`. Le fonctionnement de `launchExternalApp()` est relativement simple.

Tout d’abord, la plateforme du mobile est reconnue grâce au service Ionic Device <sup>2</sup>.

Ensuite une phase de vérification permet de savoir si l’application est présente ou non sur le mobile, cela se fait à l’aide du service Ionic AppAvailability <sup>3</sup>. Si l’application est bien installée sur le mobile de l’utilisateur, le service in-app-browser <sup>4</sup> la démarre avec comme argument son Uniform Resource Identifier associé. Si ce n’est pas le cas, le service market <sup>5</sup> se charge d’ouvrir le store relatif au mobile utilisé permettant à l’utilisateur de télécharger directement cette application. La page de l’application est directement ouverte sur le store.

Une fois que l’utilisateur a fini de naviguer sur l’application externe, la WebView la contenant est fermée et dans le cas de Resto4U l’utilisateur est redirigé sur la page contenant des informations sur ce partenaire. Cette page est codée en dur et les informations s’y trouvant proviennent du portail UCL.

## Mobilité

Dans ce module, trois applications externes sont accessibles : SNCB, NextRide et Commuty. Ces applications mettent en avant les moyens de locomotions tels que respectivement les trains, les bus et les voitures avec le concept de covoiturage. Toutes les informations dont l’utilisateur pourrait avoir besoin s’y trouvent. Le fonctionnement change légèrement du point de vue du lancement et de la fermeture par rapport à la partie “Restaurants universitaires”, contrairement à cette partie les logos boutons se trouvent directement sur une page racine à part entière. La méthode `launchExternalApp()` sera dès lors directement appelée lorsque l’utilisateur appuiera sur une des applications présentes. La logique d’implémentation de `launchExternalApp()` reste quant à elle la même. Pour ce qui en est de la fermeture, une fois la navigation terminée sur l’application externe, l’utilisateur est directement redirigé sur la page où il a lancé l’application. Aucune page contenant des informations sur ces applications n’a été créée au sein de l’application UCLCampus.

## Guindaille2.0

L’ensemble des informations de ce module sont codées en dur, aucun service, ni interne ni externe, n’est appelé. Aucune autre solution n’a été envisagée, toutes les sources d’informations

---

2. <https://github.com/apache/cordova-plugin-device>

3. <https://github.com/ohh2ahh/AppAvailability>

4. <https://github.com/apache/cordova-plugin-inappbrowser>

5. <https://github.com/xmartlabs/cordova-plugin-market>

reçues étaient sous format papier et resteront valables à long terme, ce ne sont pas des informations sujettes à des changements fréquents. Il en a été jugé en accord avec le responsable Guindaille2.0 que ces informations pouvaient rester statiques au sein de l'application UCLCampus. L'avantage principal est la rapidité et la simplicité de mise en place d'un tel module, la création d'une page statique aux couleurs de la campagne était la seule chose à faire. Un autre avantage de cette méthode est la facilité de modification, si des sources génériques sont mises en place pour récupérer les informations, le développeur doit juste récupérer les données à l'aide de services et les transformer en objet pour les intégrer dans la structure HTML existante de la page.

Sur la page actuelle Guindaille2.0 sont présents les pictogrammes liés à la campagne visant la réduction des risques et des nuisances liés à la "guindaille", à chacun de ces pictogrammes est lié une petite description qui s'ouvre dans un pop-up lorsque l'on appuie dessus. Un autre élément constituant cette page est l'effectomètre, c'est une image horizontale que l'utilisateur peut faire défiler, elle a été créée par Univers Santé. Son but est de situer l'utilisateur sur son état d'ébriété et de lui fournir des informations pour chacune de ces étapes. Lorsqu'il appuie sur une étape un pop-up s'ouvre également avec un texte descriptif comprenant des conseils. Le dernier élément de ce module est la partie "secours" qui contient les numéros d'urgences et de gardiennage à appeler en cas de problème.



Cinquième partie

Prise de recul

Un tel projet n'est pas des plus simples à réaliser, ce chapitre tend à prendre du recul sur le développement de l'application UCLCampus. Les différentes sections traversent les éléments principaux qui le composent avec les difficultés associées qui ont été rencontrées durant le développement de ce projet et les solutions qui y ont été apportées lorsque cela était possible. Cela commence par la section Open Source faisant état des difficultés et des impacts liés à son cadre général apporté dans le développement du projet. La seconde section établit les avantages et inconvénients de la technologie utilisée par les développeurs pour créer l'application en elle-même. La section suivante quant à elle se concentre sur l'environnement UCL par sa diversité de services proposés, la disparité de ses services et la difficulté de rassembler les besoins de chacun. La dernière section fournit une analyse précise de la réalisation du projet dans sa globalité, elle commence par une comparaison avec le cahier des charges pour ensuite lister toutes les tâches accomplies et finir par décrire ce qu'il reste à faire de concret pour avoir une application entièrement fonctionnelle respectant les attentes des utilisateurs.

## 1 Open Source

Pour commencer, le projet UCLCampus a migré du compte GitHub des étudiants responsable de ce projet durant l'année académique 2016-2017 vers notre propre compte GitHub. Le compte GitHub sur lequel le projet est développé est actuellement un compte public. Une migration finale sera prévue vers le compte GitHub officiel de l'UCL pour que les futures personnes en charge du projet puissent le maintenir et le gérer les prochaines années. Pour pérenniser le projet, il semble essentiel qu'une personne s'investisse dans le projet de façon permanente afin de permettre la transmission de la philosophie qui y est liée, de l'information, du cahier des charges et de la justification des choix d'implémentation. Cette personne serait le contact de référence du projet et donnerait la ligne directrice qu'il doit prendre au fur et à mesure de son évolution.

De nombreux outils ont été mis en place par les anciens mémorants pour créer un écosystème propice pour attirer de nouveaux développeurs.

Ces outils comportent un Readme décrivant les étapes à suivre pour installer le projet sur une machine, un Gitter qui offre un espace de discussions pour la communauté, un ZenHub pour établir une liste des tâches à réaliser sur un board et un Wiki qui sert à tenir informé la communauté de l'avancement du projet et de la direction qui lui est donnée.

Ces outils sont toujours disponibles mais une mise à jour s'impose. Les efforts fournis ne se sont pas concentrés sur cette partie du projet. Il a d'abord fallu un certain temps à comprendre le projet, à le prendre en main petit à petit et à reprendre contact avec les différents services impliqués. Ensuite, le développement a été la partie à laquelle la majorité du temps restant a été consacrée. Et pour terminer, la rédaction de ce mémoire représente la dernière partie du temps consacré à ce projet.

L'investissement nécessaire à créer une communauté et à attirer de nouveaux développeurs n'a malheureusement pas été suffisant pour toucher le milieu étudiant. Cependant, même si GitHub et les outils créés ne sont pas utilisés, un certain nombre de personnes au sein des entités de l'UCL sont investies dans le projet. Ces personnes fournissent une aide considérable au développement de l'application à l'aide de feedback, d'API développées et fournies, de ressources rendues accessibles, etc.

Le manque d'attrait par les développeurs extérieurs peut être traduit par la trop forte dépendance des technologies et de la structure de l'information présente à l'UCL. Ce projet est certainement trop propre à l'UCL et axé uniquement sur les besoins de ses propres étudiants. Les projets attirants de nombreux développeurs visent plutôt le développement de nouvelles technologies, de frameworks, de plugins qui sont souvent plus génériques et répondent à un besoin plus large.

Pour ce qui en est des étudiants de l'UCL, ceux-ci n'ont malheureusement pas encore le réflexe de s'intéresser à des projets Open Source et encore moins en ceux développés au sein de l'université. Des événements tels que l'Open Week, permettant de faire découvrir le concept de développement

de projets Open Source, sont malgré tout organisés pour augmenter cet attrait mais une trop faible quantité d'étudiants y participent.

Le déploiement de l'application sur les stores Android et Apple marquera un tournant décisif pour la communauté du projet. Il se peut si le nombre de téléchargements atteint l'estimation calculée, approximativement 20 000, que l'application crée un engouement auprès des étudiants de l'UCL et que certains d'entre eux s'impliquent dans le projet pour le maintenir à flot et le faire évoluer. Si c'est le cas, la communauté du projet Open Source trouvera alors un équilibre et une utilité.

## 2 Framework Ionic

L'utilisation du framework Ionic en version 3 a été une décision mûrement réfléchie par les deux précédents mémorants. En effet, ceux-ci ont eux-mêmes repris le travail de leurs prédécesseurs qui utilisaient la version 1 avec l'architecture d'AngularJS. À cette période, le projet Ionic étant en pleine évolution afin de devenir plus stable, ils ont pris la décision de migrer vers la version 2 d'Ionic et plus tard à la version 3 sur laquelle l'application UCLCampus a été développée car la première version commençait à être délaissée et à devenir obsolète. Cette migration de la version 1 à la version 2 força nos prédécesseurs à repartir à zéro car il y avait un changement total d'architecture et de structure avec le passage à Angular2. Le passage à la version 3 a été beaucoup plus simple, Ionic3 utilise l'architecture Angular4 et supporte TypeScript 2.2.

Lors de la reprise du projet, nous avons commencé par lire entièrement le précédent mémoire car celui-ci était riche en information à propos d'Ionic, Cordova, Angular et TypeScript. Nous avons également pris rendez-vous avec nos prédécesseurs afin de leur poser des questions sur les éventuels problèmes de compréhension que l'on pouvait avoir au commencement. Ceci nous a facilité la compréhension de l'architecture de l'application ainsi que le fonctionnement des différents frameworks utilisés. La seule difficulté a été d'installer et de faire fonctionner le projet existant sur nos machines personnelles. Une fois cela fait, lorsque la phase de développement a commencé, la prise en main du projet s'est faite assez rapidement.

### 2.1 Avantages et inconvénients

Le framework permet un développement d'applications mobiles multiplateformes ou hybrides. Son principal atout est donc qu'il est possible d'écrire le code une seule fois pour plusieurs plateformes à la fois telles que iOS et Android contrairement au développement natif où il aurait fallu écrire entièrement un code en Java pour Android et en Objective-C pour iOS. Le développement avec Ionic repose sur des technologies comme HTML, CSS et TypeScript qui sont assez simples à prendre en main. L'application sera alors lancée à l'intérieur d'une "webview" dans une application native : une instance d'un explorateur internet. En effet, le développement multiplateforme permet d'accéder aux fonctionnalités natives de l'appareil (GPS, caméra, système SMS et appel, accès à la carte SD, etc.) via Ionic-Native qui est une surcouche de Cordova. Ce type de développement permet également de fonctionner sans connexion internet contrairement aux applications webs. L'avantage de l'utilisation de TypeScript plutôt que Javascript est d'avoir un code plus structuré et plus propre. Ionic offre également beaucoup de fonctionnalités graphiques telles que des listes, des éléments "glissants", des boutons, etc. qui nous sont très utiles pour designer l'application.

Son utilisation offre encore d'autres avantages, Ionic est entièrement gratuit, Open Source et donne accès à des plugins grâce à Cordova ce qui permet d'ajouter un nombre considérables de fonctionnalités déjà implémentées. Ionic est très bien maintenu et possède une large communauté très réactive et dynamique. Mais le développement multiplateforme n'est pas que bénéfique : l'exécution de l'application à l'intérieur d'une webview a pour conséquence une perte de perfor-

mance et de fluidité. En effet, dans le cadre de notre projet, l’affichage de liste avec beaucoup d’éléments prend du temps à se charger et à s’afficher. Il est également à noter que le temps de lancement de l’application est assez long mais il sera aisément possible d’accélérer ce processus grâce à une des nouvelles fonctionnalités d’Ionic 3 : le lazy loading[59]. Lors du lancement d’une application Ionic, toutes les pages et services sont chargés. Avec le lazy loading, il est possible d’uniquement charger ceux-ci lorsqu’on en a besoin. Cela rend également le code plus modulable et maintenable car avec cette technique, un module Angular autonome est créé pour chaque page et au lieu d’envoyer une instance d’une page, on envoie une chaîne de caractère ce qui supprime beaucoup de dépendances dans le code. Il est à noter qu’à l’heure actuelle, les systèmes d’exploitation mobile et les navigateurs internet sont en constante évolution ce qui implique que les performances seront améliorées avec le temps.

## 2.2 Mise à jour future

Une version 4 d’Ionic[60] est en cours de développement avec quelques changements par rapport aux anciennes versions. En voici quelques-uns :

- Tous les composants d’Ionic pourront être des composants webs et non plus forcément des composants Angular ;
- La création d’un compilateur de composants webs d’Ionic : Stencil ;
- De nouvelles syntaxes pour les éléments Ionic comme par exemple `<button ion-button>` deviendra simplement `<ion-button>` ;
- Un changement architectural au vu de la prise d’indépendance d’Ionic par rapport à Angular ;
- Au lieu d’utiliser le système push/pop de navigation, il sera possible d’utiliser le routage d’Angular.

Une fois cette version terminée, il serait intéressant d’en analyser les différences et de réfléchir à une possible migration vers Ionic 4 si la version est stable et si ce changement n’exige pas de repartir de zéro.

## 3 Environnement UCL

L’environnement de l’UCL est celui dans lequel a pris naissance le projet UCLCampus. Un carnet de contacts pouvant nous apporter de l’aide et des ressources pour le développement de l’application avait été créé par les étudiants responsables du projet UCLCampus durant l’année académique 2016-2017. Celui-ci a été mis à jour et ajouté dans les annexes. L’environnement et l’infrastructure informatique présente à l’UCL sont relativement complexes à prendre en main, ces différents contacts ont été nécessaires afin de mieux situer les services et les interactions entre eux.

Pour ce qui en est du développement, une bonne partie du temps est passée à contacter des membres du personnel de l’UCL, à envoyer des mails, à aller à la rencontre de contacts au sein de leur bureau, à présenter l’application aux personnes qui se sont impliquées dans le projet, à récupérer des feedbacks, etc. Ce temps passé était du temps perdu à coder l’application attendue. Il a fallu trouver un équilibre pour pouvoir avancer convenablement sur l’application tout en continuant les prises de contact pour répondre à des besoins auxquels des solutions étaient à chercher en dehors de nos compétences. L’avancement de certains modules dépendait de ressources qui devaient d’abord être créées par d’autres programmeurs. Il fallait donc jongler avec ces contre-temps et se pencher sur d’autres tâches en attendant d’être débloqué.

Un des problèmes soulignés au sein de l’environnement est la disparité et le manque d’interaction du personnel informatique des différents services. Cela peut être démontré très rapidement à l’aide de trois services de l’UCL, c’est-à-dire le service des sports, Univers Santé et les restaurants

universitaires. Les deux derniers services ont chacun créé leur propre application et le premier est en train de chercher des pistes pour en développer une. Cependant, aucun des trois n'a fait appel aux mêmes développeurs. Même si les besoins sont différents, une cohérence entre les différents projets pourrait être trouvée. Un design similaire ainsi que le développement et la maintenance en interne pour toutes les applications UCL sont des pistes à envisager.

Pour arriver à cela, les différents services devraient améliorer la communication entre eux sur leurs projets informatiques. Un organe de réflexion et d'harmonisation devrait être créé afin de permettre à chaque service d'étaler ses besoins et de réfléchir à des solutions communes. Cet organe répartirait également les ressources informatiques au sein de ces services afin qu'une certaine égalité soit respectée en fonction des besoins.

Il ne faut pas omettre que des efforts sont réalisés par l'université par la mise en place de l'ESB WSO2 qui vise à l'harmonisation technique des différents services informatiques de l'université. Les interactions entre ces services devraient être simplifiées grâce à une standardisation de communication. L'intégration des différents services et la création d'API se font petit à petit.

## 4 Sécurité

L'aspect sécurité de l'application UCLCampus concerne deux fonctionnalités importantes : l'identification d'un étudiant de l'UCL et la récupération de son programme de cours. Il est évident pour l'identification que l'identifiant et le mot de passe de l'utilisateur sont des données sensibles et qu'elles se doivent d'être protégées. Le programme de cours d'un étudiant est une donnée moins sensible mais qui n'est malgré tout pas accessible publiquement. Ce dernier n'est enregistré nulle part dans l'application. Lorsque l'utilisateur récupère son programme, il a le choix d'ajouter les cours s'y trouvant à la liste de l'onglet "Horaire". Dès que l'utilisateur quitte la page, le programme de cours est tout simplement effacé et il lui est demandé de se reconnecter s'il retourne sur cette page et désire revoir son programme. Ainsi, lorsque l'appareil d'un étudiant est utilisé par un autre utilisateur, ce dernier n'a pas accès au programme de cours et aux identifiants de l'étudiant à qui appartient l'appareil.

Depuis le travail réalisé par nos prédécesseurs, un réel système d'authentification a pu être mis en place afin de récupérer ce programme dans une application mobile externe. Pour cela, un protocole OAuth a été élaboré entre l'application UCLCampus et l'API Manager de WSO2. Son fonctionnement est assez simple et est visible sur la figure 26[61].

Il y a trois acteurs dans ce processus :

- L'utilisateur : un étudiant de l'UCL ;
- Le client : l'application UCLCampus ;
- Le serveur : le WSO2 de l'UCL.

Les étapes du processus sont les suivantes :

1. L'application UCLCampus est enregistrée auprès de l'API Manager de WSO2 et des services UCL en créant un client et en listant les droits dont l'application aura besoin (les API's à utiliser) ;
2. L'UCL WSO2 attribue un consumer-key (ID) et un consumer-secret à l'application ;
3. Lorsque l'étudiant veut se connecter en utilisant UCLCampus, il est redirigé vers une adresse spécifique de UCL WSO2 où il lui est demandé s'il accepte de donner ses informations à l'application ;
4. Si l'étudiant accepte, UCL WSO2 lui attribue un code d'autorisation ;
5. Grâce à ce code, l'application demande un token d'accès ;
6. UCL WSO2 attribue un token d'accès pour l'étudiant utilisant l'application ;
7. Ce token peut être utilisé par l'application afin de faire des requêtes aux diverses API pour demander des données ;

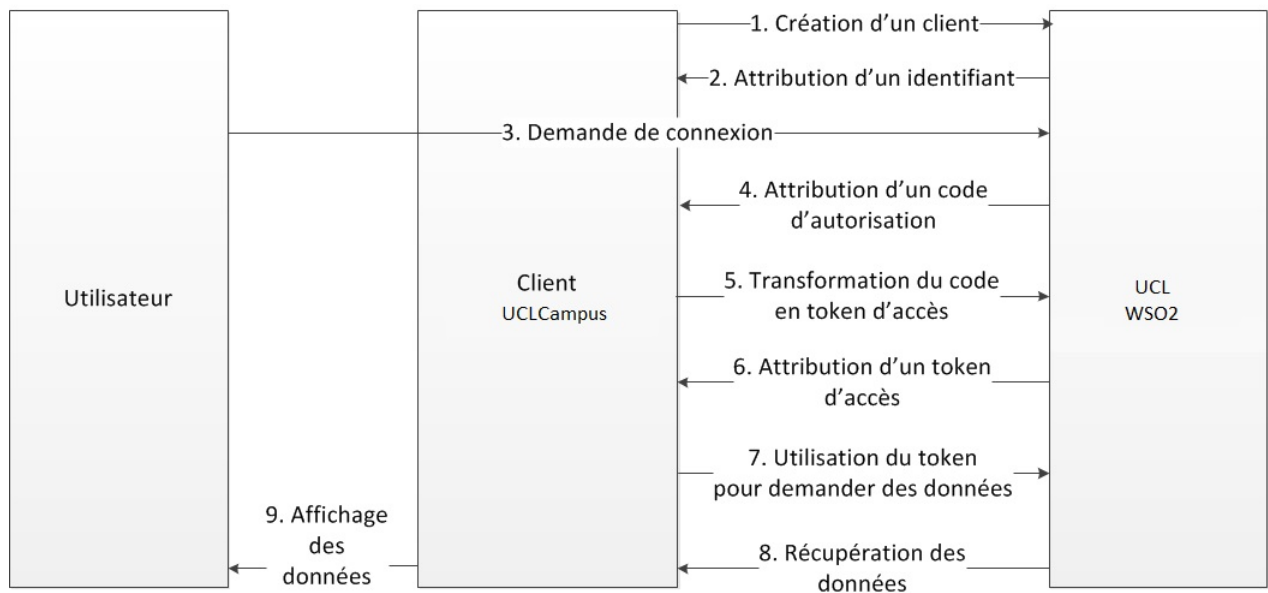


FIGURE 26 – Identification d'un utilisateur et récupération des données

8. UCL WSO2 renvoie les données à l'application si le token d'accès est valide ;
9. L'application peut afficher les données à l'utilisateur.

Les points 5 à 7 sont représentés plus en détail sur la figure 27[62].

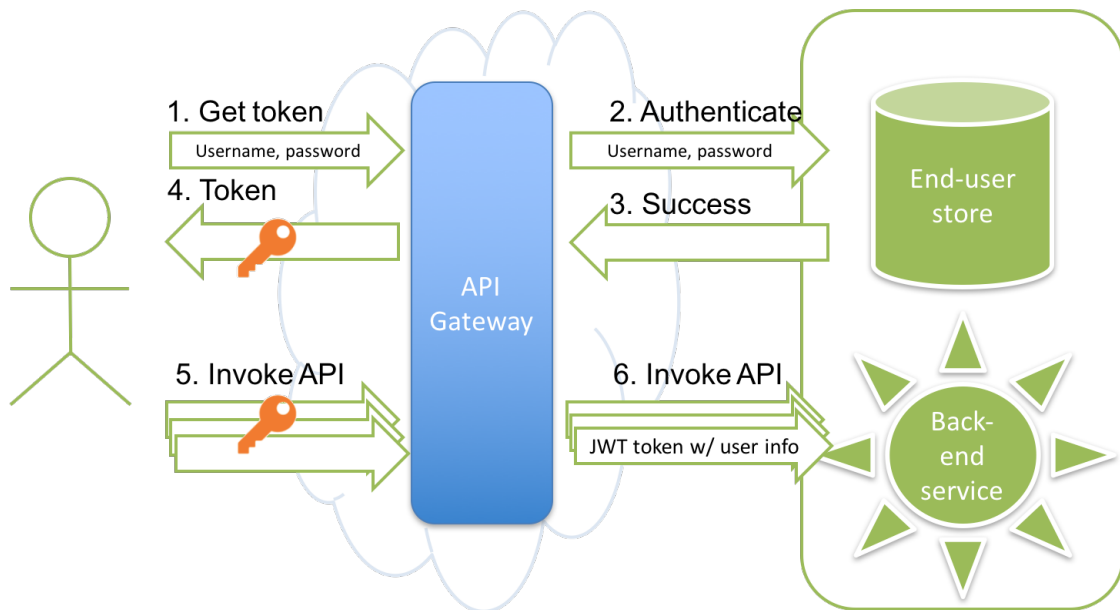


FIGURE 27 – Récupération du token et requêtes aux API

1. Concrètement, lorsque l'utilisateur s'identifie sur l'application, une requête est faite à l'API Manager avec ses identifiants pour récupérer un token d'accès via l'API Token. Cette requête est une requête cURL contenant un en-tête "Authorization Header" avec le consumer-key et consumer-secret chiffré en base64 (Base64(consumer-key :consumer-secret)) ;
2. L'API Gateway vérifie si l'utilisateur existe dans l'userstore (une base de données contenant les utilisateurs). Si c'est le cas, il vérifie que les identifiants sont valides ;

3. L'utilisateur s'occupe de cette vérification et signale si c'est un succès ou un échec ;
4. Le gateway génère le token OAuth et le renvoie à l'application ;
5. A chaque requête vers une API, UCLCampus ajoute ce token dans l'"Authorization Header" ;
6. Le gateway vérifie ce token avant d'autoriser l'appel et de transmettre les informations de l'utilisateur au back-end.

## 5 Réalisation

Quand est-il de la réalisation de l'application ? Cette section a pour but de répondre à cette question en deux parties distinctes. D'une part, la comparaison avec le cahier des charges est établie afin de mettre en avant les tâches accomplies ainsi que les tâches restantes. Alors que la seconde partie a comme objectif de proposer de nouvelles améliorations de l'application non prises en compte dans le cahier initial des charges et qui ont été jugées intéressantes à mettre en oeuvre durant le développement de l'application.

### 5.1 Comparaison avec le cahier des charges

La majorité des fonctionnalités présentées au début de ce mémoire ont été implémentées et sont opérationnelles au sein de l'application. Le déploiement de l'application dans l'état actuel est envisageable, les modules présents répondent correctement aux besoins des utilisateurs mais de légères modifications rendraient l'expérience de son utilisation encore plus agréable et intuitive. Cette partie reprend les éléments du cahier initial des charges de l'application. Les spécifications fonctionnelles sont divisées en deux groupes, les tâches accomplies et les tâches restantes. Les tâches qui se trouvent dans ces groupes sont présentées et rassemblées en tableaux, chaque tableau est propre à une partie de l'application. Les différentes parties sont "Campus", "Etudes", "Outils" et "Accueil". Des remarques sont ajoutées à certaines tâches afin de donner des détails sur celles-ci. Pour les spécifications non-fonctionnelles, elles sont revues en détails afin de faire le point sur le respect de leur réalisation.

#### Tâches accomplies

| Fonctionnalités de la page Accueil        | Remarques                                                                |
|-------------------------------------------|--------------------------------------------------------------------------|
| Accéder à chaque section de l'application |                                                                          |
| Accéder aux paramètres                    | Bouton ouvrant une fenêtre de dialogue                                   |
| Fonctionnalités de la page Paramètres     | Remarques                                                                |
| Changer la langue                         | Avec le plugin @ngx-translate mais beaucoup de textes restent à traduire |
| Choisir un campus de l'UCL                |                                                                          |

TABLE 1 – Fonctionnalités réalisées propres à l'application

|                                                         |                                       |
|---------------------------------------------------------|---------------------------------------|
| <b>Fonctionnalités de la page News</b>                  | <b>Remarques</b>                      |
| Afficher une liste de news                              | Flux RSS du portail UCL               |
| Rechercher les news par mot-clé                         |                                       |
| Afficher les détails d'une news                         |                                       |
| Rediriger vers le site web de la news                   |                                       |
| Séparer les différentes news                            | En catégories                         |
| <b>Fonctionnalités de la page Évènements</b>            | <b>Remarques</b>                      |
| Afficher une liste d'événements                         | Flux RSS de Louvaininfo               |
| Afficher les détails d'un événement                     |                                       |
| Ajouter un événement en favoris                         |                                       |
| Ajouter un événement à l'agenda natif                   |                                       |
| Rediriger vers le site web de l'événement               |                                       |
| Rechercher un événement par mot-clé                     |                                       |
| Filtrer des événements                                  | Filtre par catégorie et durée désirée |
| <b>Fonctionnalités de la page Sports</b>                | <b>Remarques</b>                      |
| Afficher une liste de sports                            | Fichier XML en ligne                  |
| Afficher une liste d'entraînements universitaires       | Fichier XML en ligne                  |
| Afficher l'horaire et salle d'un sport                  |                                       |
| Afficher une note importante lorsqu'elle est disponible |                                       |
| Rechercher un sport par mot-clé                         |                                       |
| Filtrer des sports                                      | Filtre par sport                      |
| Ajouter un sport aux favoris                            |                                       |
| Ajouter un sport à l'agenda natif                       |                                       |

TABLE 2 – Fonctionnalités réalisées de la partie "Campus"



| <b>Fonctionnalités de la page Bibliothèques</b>                | <b>Remarques</b>                             |
|----------------------------------------------------------------|----------------------------------------------|
| Afficher une liste des bibliothèques                           | Via l'ESB WSO2 de l'UCL avec l'API Libraries |
| Afficher les détails d'une bibliothèque y compris ses horaires | Via l'ESB WSO2 de l'UCL avec l'API Libraries |
| Rediriger vers le site web de la bibliothèque                  |                                              |
| <b>Fonctionnalités de la page Etudes</b>                       | <b>Remarques</b>                             |
| Ajouter un cours manuellement à ma liste                       |                                              |
| Voir l'horaire d'un cours                                      | Via ADE Expert                               |
| Voir la salle où se déroule un cours                           | Via ADE Expert                               |
| Accéder au site web de l'UCL                                   |                                              |
| Accéder au site web Moodle                                     |                                              |
| Ajouter une séance de cours à l'agenda natif                   |                                              |
| Ajouter tout un cours à l'agenda natif                         |                                              |
| S'identifier à l'UCL                                           | Via l'ESB WSO2 avec l'API My                 |
| Voir le programme de cours                                     | Via l'ESB WSO2 avec l'API My                 |
| Ajouter un cours du programme à ma liste                       |                                              |
| <b>Fonctionnalités de la page Support</b>                      | <b>Remarques</b>                             |
| Voir les informations du Service Desk                          | Informations codées en dur                   |
| Voir les informations liés à la carte d'accès                  | Informations codées en dur                   |
| Voir les informations sur le WiFi                              | Informations codées en dur                   |
| Rechercher un membre du personnel UCL                          | Via l'ESB WSO2 avec l'API Directories        |
| Afficher les informations d'un membre du personnel UCL         |                                              |

TABLE 3 – Fonctionnalités réalisées de la partie "Etudes"

| <b>Fonctionnalités de la page Carte</b>                      | <b>Remarques</b>                                          |
|--------------------------------------------------------------|-----------------------------------------------------------|
| Afficher sa position sur la carte                            |                                                           |
| Afficher une liste de points d'intérêts                      | Via un fichier JSON                                       |
| Séparer les points d'intérêts par catégories                 | Catégories : Parkings, Auditoires, Bibliothèques, Resto U |
| Afficher la position d'un point d'intérêt                    |                                                           |
| <b>Fonctionnalités de la page Restaurants Universitaires</b> | <b>Remarques</b>                                          |
| Lancer l'application Resto4U                                 |                                                           |
| Proposer le téléchargement de l'application Resto4U          |                                                           |
| <b>Fonctionnalités de la page Mobilité</b>                   | <b>Remarques</b>                                          |
| Lancer l'application SNCB                                    |                                                           |
| Proposer l'installation de l'application SNCB                |                                                           |
| Lancer l'application NextRide                                |                                                           |
| Proposer l'installation de l'application NextRide            |                                                           |
| Lancer l'application Commuty                                 |                                                           |
| Proposer l'installation de l'application Commuty             |                                                           |
| <b>Fonctionnalités de la page Guindaille2.0</b>              | <b>Remarques</b>                                          |
| Accéder aux différents éléments de la campagne               | Pictogrammes et effectometre                              |
| Afficher les informations de secours                         | Numéro d'urgence et de gardiennage                        |

TABLE 4 – Fonctionnalités réalisées de la partie "Outils"

## Tâches restantes

Dans cette section, il sera brièvement présenté les différentes fonctionnalités non réalisées mais qui seraient malgré tout importantes ou du moins utiles aux utilisateurs de l'application. Celles-ci seront détaillées dans la section concernant les améliorations. Le tableau 5 présente les différentes fonctionnalités à réaliser sur le module Accueil, le tableau 6 présente celles à réaliser sur le module News, le tableau 7 sur le module Etudes et le tableau 8 sur le module Map. Comme précisé précédemment, ces tâches seront expliquées plus en détails dans la partie améliorations. Cela pourrait simplement être de l'implémentation d'un module ou son amélioration comme cela pourrait être la rencontre officielle avec des membres de l'UCL avec qui nous discuterions de la manière d'envisager les éventuels besoins de l'application qui ne sont pas encore possibles avec l'architecture de l'UCL.

| <b>Fonctionnalités à réaliser de la page Accueil</b>       |
|------------------------------------------------------------|
| Echanger la ligne du module "Campus" et du module "Etudes" |
| Ajouter un bouton Guindaille2.0                            |

TABLE 5 – Fonctionnalités restantes du module Accueil

| <b>Fonctionnalités à réaliser de la page News</b> |
|---------------------------------------------------|
| Ajouter un onglet d'actualité de faculté          |
| Changer le flux des actualités universitaires     |

TABLE 6 – Fonctionnalités restantes de la partie "Campus"

| <b>Fonctionnalités à réaliser de la page Etudes</b> |
|-----------------------------------------------------|
| Choisir la séance de TP ou CM                       |
| Voir les notes d'examen                             |

TABLE 7 – Fonctionnalités restantes de la partie "Etudes"

| <b>Fonctionnalités à réaliser de la page Carte</b>                                                          |
|-------------------------------------------------------------------------------------------------------------|
| Afficher un itinéraire vers un point d'intérêt sélectionné                                                  |
| Rechercher un point d'intérêt par mot-clé                                                                   |
| Ajouter "Salles d'étude", "Salles informatiques", "Salles de sport" et "Salles d'activité" comme catégories |
| Ajouter les points d'intérêts manquants                                                                     |
| <b>Fonctionnalités à réaliser de la page Guindaille2.0</b>                                                  |
| Déplacer les informations secours dans un module à part entière                                             |

TABLE 8 – Fonctionnalités restantes de la partie "Outils"

## Spécifications non fonctionnelles

Les différentes exigences non fonctionnelles de l'application, identifiées dans le cahier des charges, sont parcourues une à une dans cette partie afin de vérifier si elles ont été respectées durant la réalisation du projet et si ce n'est pas le cas, expliquer pourquoi. Contrairement aux deux parties précédentes, le choix d'un texte continu a été préféré à un tableau dû aux remarques plus longues et à la forme qui s'y prête mieux.

Du point de vue **utilisation**, l'architecture horizontale de l'application et la présence d'un menu latéral lui permettent d'être intuitive et simple à utiliser pour l'utilisateur. Un guide

utilisateur est également fourni à l'utilisateur pour l'aider dans la prise en main de l'application. Pour ce qui en est des développeurs, un readme est présent sur le GitHub du projet facilitant son installation et sa prise en main. Cependant, plusieurs points manquent à l'appel. D'abord, il faut créer un tutoriel vidéo pour aider l'utilisateur dans ses débuts mais cela est prévu dans la dernière phase avant le déploiement de l'application sur les stores. Ensuite, le code doit être rendu plus compréhensible par l'ajout de commentaires pour que toute personne s'y intéressant puisse le comprendre sans difficulté. Pour finir, la mise en cache d'informations doit être effective afin que l'application reste fonctionnelle lorsque les services UCL sont en maintenance ou que l'utilisateur ne peut pas se connecter à internet.

Du point de vue **programmation**, l'utilisation du framework Ionic version 3 a été respectée pour rester dans la continuité du projet UCLCampus de l'année académique 2016-2017.

Du point de vue **fiabilité**, l'application n'est pas sujette à des problèmes techniques récurrents mais il se peut que celle-ci rencontre des problèmes mineurs lors de chargements de données inaccessibles. Ces problèmes devront également être réglés avant son déploiement sur les stores.

Du point de vue **modularité**, le projet respecte ce critère d'une part grâce à l'architecture du framework Ionic et d'autre part par la scission entre les différentes couches, à savoir les couches interface, logique et donnée. Les services sont indépendants entre eux.

Du point de vue **compatibilité**, l'application est multiplateforme, elle est supportée par les mobiles qui tournent sous iOS version 8 et ultérieur ou sous Android version 4.4 et ultérieur. Le respect de la convention de sementic versionning n'est quant à lui pas encore effectif étant donné qu'aucune mise à jour de l'application n'a été déployée.

Du point de vue **performance**, l'application est fluide et affiche comme il se doit un logo de chargement quand cela est nécessaire, c'est à dire lorsque le temps de chargement d'une page excède 1 seconde. Quant à l'utilisation de l'espace mémoire du mobile par l'application, celle-ci est inférieure à 15 Mo, ce qui est négligeable et ne porte pas à y faire attention.

Du point de vue **légal**, comme prévu, l'application est sous licence GNU GPL v3 et les plugins utilisés sont compatibles avec cette licence. Cependant, pour le respect des règles de la charte graphique quelques changements seront à faire. Un nouveau logo a été créé, il faut remplacer l'ancien dans l'application par le nouveau une fois qu'il sera en vigueur.

## 5.2 Améliorations proposées

Dans cette partie, un ensemble d'améliorations sont proposées. Celles-ci seront réparties en plusieurs catégories : les différents éléments à ajouter à l'application en elle-même, les fonctionnalités à améliorer ou à ajouter aux différents modules ainsi que la nécessité de redesigner l'application pour la rendre plus attractive pour les utilisateurs.

### Application

Comme précisé dans un chapitre précédent, une nouvelle version du framework Ionic est en cours de développement, la version 4. Cela pourrait être bénéfique pour l'application UCLCampus de migrer sur celle-ci si elle ne demande pas de repartir de zéro et que les améliorations par rapport à la version actuelle apportent vraiment un plus au niveau des performances.

Il est également possible qu'un changement de nom pour l'application soit appliqué.

Quant à lui, le module Guindaille2.0 contient les numéros d'urgence à contacter, ceux-ci étant évidemment important, il faudrait les rendre plus accessibles à l'utilisateur et que celui-ci n'ait pas à naviguer à travers les modules pour les trouver.

Pour ce qui concerne la communication de l'application après son déploiement, ajouter un lien vers les différents réseaux sociaux de l'UCL tels que LinkedIn, Twitter, Facebook, Youtube, Instagram pourrait s'avérer très utile afin de faire connaître l'application par les utilisateurs qui voudraient partager UCLCampus sur leurs réseaux sociaux.

Il est également à noter que dans la section Support, plusieurs organismes pourraient désirer

avoir une place au sein de l'application. Un lien vers ces différents organismes tels que le service d'aide, financier, de logements ou psychologique pourrait être ajouté à ce module.

Pour terminer, il n'est pas intuitif à l'utilisateur de trouver la possibilité de glisser vers la gauche un élément afin de l'ajouter aux favoris ou de l'ajouter au calendrier natif. Il serait plus qu'important de le mentionner visuellement dans l'application sur les pages concernées.

## Fonctionnalités

Connaissant le nombre de services d'aides disponibles sur les différents campus de l'UCL, il serait intéressant d'ajouter aux ressources (sous forme de fichier JSON pour chaque campus) différents points d'intérêts tels que le service d'aide financier, le service d'aide psychologique, le service logements UCL, etc. Peu d'étudiants sont au courant de l'existence de ceux-ci. L'un des modules de l'application UCLCampus pourrait aider les utilisateurs à s'informer à leur propos et de les localiser afin de s'y rendre. D'autres points d'intérêts utiles devraient aussi être ajoutés tels que les différentes salles d'études, les salles de sports, les salles informatiques et la localisation d'UniverSanté.

Pour le module Libraries, une amélioration intéressante serait de fournir le nombre de places restantes ou la fréquentation de chaque bibliothèque. Plusieurs institutions ont déjà implémenté cette fonctionnalité en fonction du nombre d'appareils connectés au WiFi. Cette idée pourrait être une piste à visiter pour l'UCL.

Afin d'augmenter les performances de l'application et la fluidité pour l'utilisateur, il y aurait un grand intérêt à mettre certaines données en cache tant que l'application est ouverte. Un bon exemple est la liste de bibliothèques : celle-ci change rarement de jour en jour mis à part si une remarque importante est ajoutée. L'application pourrait mettre en cache cette liste afin que l'appareil n'ait pas à la charger à chaque ouverture du module. Il serait également intéressant de penser à cette possibilité pour la liste d'actualités, d'événements et de sports.

De plus, du côté pratique et utile à un étudiant, plusieurs fonctionnalités seraient intéressantes à ajouter :

- L'horaire disponible sur ADE Expert n'étant pas toujours clair vu le nombre de différentes sessions possibles pour un même T.P., il serait pratique de permettre à l'utilisateur de sélectionner la session de T.P. à laquelle il doit se rendre afin qu'uniquement celle-ci s'affiche sur son horaire ;
- Avoir la liste de toutes les séances d'un seul cours s'avère utile mais un horaire hebdomadaire le serait encore plus pour l'étudiant. Il faudrait que celui-ci soit créé, il rassemblerait les sessions de tous les cours que l'utilisateur a ajouté à son projet ADE en un unique horaire pour la semaine effective, l'utilisateur pourrait dès lors voir chaque jour quel cours est prévu à son horaire à la place de devoir parcourir cours par cours ceux dans la liste ;
- L'application propose déjà d'ajouter un événement, une séance de sport, une séance de cours ou directement toutes les séances d'un cours à l'agenda natif de l'appareil de l'utilisateur. Ce dernier étant de plus en plus immergé dans le numérique et utilisant probablement d'autres systèmes d'agenda numérique, l'ajout à un calendrier tel que Google Scholar ou Outlook serait bénéfique.

## Design

Le projet UCLCampus, étant un projet académique visant des étudiants d'une université renommée, se doit de ne pas être "extravagant" et de rester dans la sobriété. Cependant, l'application se doit d'être un minimum attractive pour les utilisateurs et que ceux-ci aient envie de passer du temps dessus. Il faut surtout qu'ils aient une première bonne impression et ne pas la désinstaller directement.

Pour cela il faudrait que chaque module respecte les mêmes "conventions". Dans le cas d'UCLCampus, l'application est articulée sous forme d'onglets alors que certains modules utilisent

des listes déroulantes. Afin d'uniformiser ceci, il serait nécessaire de remplacer ces listes par un système d'onglets pour garder une cohérence dans l'application.

Pour l'attractivité de l'application, une idée serait la coloration des pictogrammes des événements ainsi qu'un ajout de pictogrammes pour les différents sports. Il est à noter que les pictogrammes des événements proviennent de Louvaininfo et CarpeDiem, cela porte donc à discussion.

Lors du développement de l'application, le design ne fut pas la préoccupation principale. Aucune mention du nom de l'application ou de l'UCL ne se trouvent dans les pages d'UCLCampus mis à part celles qui parlent de l'UCL. Il faudrait au moins ajouter ces éléments sur le menu principal et peut-être sur les en-têtes des différentes pages.

Sixième partie

Conclusion

Nous avons travaillé sept mois sur ce projet et fournissons une solution complète qui rassemble tous les besoins principaux attendus par les étudiants des différents campus de l'Université catholique de Louvain. Pour arriver à cela, nous avons travaillé sur une application mobile multiplateforme à destination de ces étudiants. Elle est nommée UCLCampus et comporte un ensemble de modules répondant aux besoins identifiés dans le cahier des charges. Le projet en lui-même a été développé à l'aide de la technologie Ionic et placé sous licence Open Source. Mais l'apport du travail effectué ne se limite pas à l'application délivrée, nous avons également rédigé le cahier des charges pour identifier quoi développer, mis à jour le GitHub qui est le point clé pour la future communauté Open Source, fournit un guide utilisateur afin d'informer les étudiants du contenu disponible sur l'application et créé un carnet de contacts pour s'y retrouver dans l'environnement UCL plus aisément.

Pour atteindre nos ambitions et fournir la solution finale, nous avons jonglé entre différentes étapes de travail et de réflexion dûes à des contraintes rencontrées en cours de route auxquelles nous devons trouver des solutions pour pouvoir continuer. Le chemin parcouru n'a pas été un long fleuve tranquille, l'enchaînement des tâches était par moment ralenti en attendant des solutions réalisées en externes par des services que nous ne pouvions pas superviser. Les différentes phases parcourues sont expliquées en quelques paragraphes afin de se faire une idée du cheminement de nos efforts.

Tout d'abord, nous avons commencé par la prise en connaissance du travail fourni par nos prédécesseurs, une analyse de leur mémoire nous a permis de savoir ce qui avait été réalisé et les challenges à relever. La prise en main du code déjà développé a quant à elle été une tâche fastidieuse, se lancer dans projet à plus de 80 fichiers prend un temps considérable pour comprendre leurs utilités et l'architecture qui a été élaborée. Différentes prises de contact avec le personnel des multiples services de l'UCL ont été réalisées en parallèle avec ces deux premières tâches afin d'identifier plus aisément ce qu'il restait à faire et si les ressources manquantes pouvaient nous être fournies. Ajouté à cela, nous avons eu la chance que nos prédécesseurs aient déjà réalisé un sondage auprès des étudiants pour connaître leurs souhaits et les fonctionnalités que l'application devait contenir. C'est sur base de tous ces éléments que nous avons terminé cette première phase préparatoire par la création du cahier des charges qui a guidé tout au long le reste de notre travail.

Suite à ce début, nous avons pu nous lancer dans le développement des différents modules qui restaient à implémenter. Les réunions avec divers services continuaient à s'enchaîner afin de trouver des solutions aux problèmes rencontrés. Cette phase représente une majorité du temps passé sur le projet.

Une fois que tous les modules se trouvant dans le cahier des charges étaient présents dans l'application, nous avons décidé de reprendre contact avec une majorité des services et personnes qui s'étaient impliqués ou intéressés à notre application UCLCampus. Cette phase nous a permis d'obtenir un feedback sur notre première version et d'identifier ce que nous devons améliorer. Suite à ces retours, une deuxième phase de développement a pris place pour implémenter les améliorations essentielles que nous n'avions pas prises en compte dans notre première version de l'application.

La dernière phase se limite à la présentation de la dernière version à plusieurs membres du personnel de l'UCL, de la prise en compte de leur feedback et de la rédaction de ce mémoire qui fait état de tout ce qui a été réalisé en définissant en détail tous les composants de ce projet.

Il faut aussi souligner, dans cette conclusion, les deux principales difficultés auxquelles nous avons été confrontés, à savoir l'environnement UCL et le contexte Open Source du projet. Com-

mençons par l'environnement UCL, celui-ci est très étendu et la dispersion des différents services représente une réelle contrainte. Juste au niveau de la prise de rendez-vous, il n'existe aucune personne étant disposée à contacter chacun des services afin d'harmoniser cela. Il a fallu trouver par nos propres moyens les différentes personnes à contacter pour répondre à nos besoins et avancer au rythme des solutions que nous récupérons au fur et à mesure par les différents services. En lien avec cet environnement UCL dispersé, on a également dû prendre en compte la diversité des standards utilisés par les différents services, cela représente une réelle complexité du point de vue développement. Les décisions stratégiques prises par chacun des services ont eu un impact considérable dans les choix que nous devons prendre pour développer l'application UCLCampus. Les factorisations de code représentent l'élément clé pour conserver un code modulaire. Terminons ce paragraphe par les implications liées au choix de travailler en Open Source. Il nous a fallu suivre une certaine rigueur dans l'écriture du code, celle-ci étant obligatoire afin que le code soit compréhensible et rapide à prendre en main par les futurs développeurs qui s'intéresseront au projet. L'attrait d'une communauté d'étudiants fait partie de la difficulté liée au caractère Open Source du projet et c'est ce que nous n'avons malheureusement pas su surmonter. Malgré un ensemble d'outils tels que ZenHub, Gitter et un Wiki, nous n'avons pas réussi à rassembler d'autres étudiants ou développeurs sur notre GitHub.

Pour finir, il reste un ensemble de challenges à relever avant le déploiement de l'application sur les stores pour la rentrée académique 2018, qui reste le prochain objectif. Ces challenges ont été rassemblés en cette liste :

- Traduction de l'ensemble des modules (français et anglais) ;
- Ajout des points d'intérêt manquants pour les différentes catégories déjà présentes dans le module Carte ;
- Ajout des locaux d'étude à la liste d'intérêts de la carte ;
- Ajout des locaux des sports à la liste d'intérêts de la carte ;
- Ajout des locaux d'activité à la liste d'intérêts de la carte ;
- Ajout des locaux d'études à la liste d'intérêts de la carte ;
- Ajout des locaux informatiques à la liste d'intérêts de la carte ;
- Déplacement des numéros d'urgence du module Guindaille2.0 vers un module à part entière ;
- Ajout des liens vers les différents réseaux sociaux de l'UCL ;
- Création d'un horaire hebdomadaire rassemblant les cours de l'étudiant ;
- Changement des flux d'actualités du module News.

Une fois ces challenges relevés et l'application déployée, la mission des personnes qui reprendront ce projet sera d'attirer de nouveaux membres, principalement des étudiants, dans leur communauté Open Source. C'est un point indispensable pour la survie d'UCLCampus et de sa pérennité.



# Bibliographie

- [1] Apache Cordova. Cordova overview, architecture. <https://cordova.apache.org/docs/en/dev/guide/overview/index.html>.
- [2] Michael Guilloux. Logiciel libre et open source : les deux concepts sont parfois utilisés de manière interchangeable, 2015. <https://www.developpez.com/actu/87401/Logiciel-libre-et-open-source-les-deux-concepts-sont-parfois-utilises-de-maniere-interchangeable-mais-quelle-est-la-difference/>.
- [3] Gabriel Dabi-Schwebel. Open source (logiciel), 2014. <https://www.1min30.com/dictionnaire-du-web/open-source-logiciel>.
- [4] Margaret Rouse. Open source, 2016. <https://www.lemagit.fr/definition/Open-Source>.
- [5] OSI. Open source initiative. <https://opensource.org>.
- [6] Pulsar Informatique. Avantages de l'open source, 2015. <https://www.pulsar-informatique.com/creation-site-internet/comment-creer-un-site-internet/quelle-technologie-choisir/logiciels-open-source/avantages-de-l-open-source>.
- [7] Jean-Marc Amon. Libre et open source, avantages et inconvénients, 2013. <https://marclabs.com/libre-et-open-source-avantages-et-inconvenients>.
- [8] Diatem. Les différentes licences open source, 2015. <https://www.diatem.net/les-licences-open-source/>.
- [9] Alain Clapaud. Comment se repérer dans la jungle des licences open source, 2014. <https://www.journaldunet.com/solutions/dsi/comparatif-des-licences-open-source.shtml>.
- [10] BlackDuck. Top open source licenses. <https://www.blackducksoftware.com/top-open-source-licenses>.
- [11] Smile Open Source Solutions. Point de vue sur l'open source, analyse et décryptage, 2013.
- [12] Patrice Lamarque. 5 raisons pour choisir l'open source, 2017. <https://www.exoplatform.com/blog/fr/2017/06/14/les-5-avantages-de-lopen-source-apprecies-par-nos-clients>.
- [13] statcounter GlobalStats. Mobile operating system market share in belgium - april 2018. <http://gs.statcounter.com/os-market-share/mobile/belgium>.
- [14] statcounter GlobalStats. Tablet operating system market share in belgium - april 2018. <http://gs.statcounter.com/os-market-share/tablet/belgium>.
- [15] Jérôme MEYER. Le développement mobile multiplateforme, Janvier 2017. <https://www.supinfo.com/articles/single/3967-developpement-mobile-multiplateforme>.
- [16] W3C. Html5, 2014. <https://www.w3.org/TR/html5>.
- [17] Apache Cordova. <https://cordova.apache.org>.
- [18] Adobe Systems Inc. Phonegap, 2016. <https://phonegap.com>.
- [19] Drifty Co. Ionic, 2018. <https://ionicframework.com>.
- [20] ionic team. Ionic github, 2018. <https://github.com/ionic-team/ionic>.
- [21] Xamarin Inc. Xamarin, 2018. <https://www.xamarin.com>.

- [22] Google. Angular, 2018. <https://angular.io>.
- [23] Angular. Angular github, 2014. <https://github.com/angular/angular>.
- [24] Adam Bradley. Ionic github, Octobre 2013. [https://blog.ionicframework.com/where-does-the-ionic-framework-fit-in/?\\_\\_ga=2.168979684.414987698.1499183802-470528657.1497369702](https://blog.ionicframework.com/where-does-the-ionic-framework-fit-in/?__ga=2.168979684.414987698.1499183802-470528657.1497369702).
- [25] Microsoft. Typescript javascript that scales, 2018. <http://www.typescriptlang.org>.
- [26] Microsoft. Typescript is a superset of javascript that compiles to clean javascript output., 2018. <https://github.com/Microsoft/TypeScript>.
- [27] Angular. Architecture overview, 2018. <https://angular.io/guide/architecture>.
- [28] W3C. W3c dom4, Novembre 2015. <https://www.w3.org/TR/2015/REC-dom-20151119>.
- [29] Smile Open Source Solutions. Esb lesmeilleures solutions open source, 2014. <http://www.asprom.com/technologie/ESB.pdf>.
- [30] Martin Breest. An introduction to the enterprise service bus. <https://pdfs.semanticscholar.org/23f0/d133df8e4d7ff132e4bdef3c4b65d57592a2.pdf>.
- [31] WSO2. Wso2 enterprise service bus, 2018. <https://wso2.com/products/enterprise-service-bus/>.
- [32] Lamy Corentin et Lemaire Jérôme. Uclcampus : une application mobile open source pour étudiants ucl, September 2017.
- [33] Université catholique de Louvain Service générale du système d'information (SGSI). Présentation infrastructure d'intégration, entreprise service bus, wso2 dss/esb/am, 2017.
- [34] Plume. Shibboleth. <https://www.projet-plume.org/fiche/shibboleth>.
- [35] Shibboleth Consortium. What is shibboleth? <https://www.shibboleth.net/>.
- [36] Université Catholique de Louvain. About osis. <http://uclouvain.github.io/osis/about/>.
- [37] Jean-Claude Torrel. Les caractéristiques d'un cms, November 2016. <https://openclassrooms.com/courses/decouvrez-les-solutions-cms/les-caracteristiques-d-un-cms>.
- [38] Drupal. Drupal build something amazing. <https://www.drupal.org/8>.
- [39] QUIVR. Caractéristiques de quivr. <https://quivr.be/en/features/>.
- [40] University College London. Ucl go! a mobile app for student tasks. <https://www.ucl.ac.uk/isd/services/websites-apps/ucl-go>.
- [41] Apptree. Resto4u. <https://play.google.com/store/apps/details?id=com.apptree.resto4u>.
- [42] Univers Santé. « guindaille 2.0 », l'appli qui te donne tout l'agenda de l'animation étudiante dans ta poche. <http://www.guindaille2-0.be/application/>.
- [43] MisterZen. Lln map. <https://play.google.com/store/apps/details?id=be.zen.llnmapshl=fr>.
- [44] Commuty. Commuty. <https://www.commuty.net/>.
- [45] NextRide. Nextride, les horaires stib et tec dans votre poche. <https://nextride.be/about>.
- [46] SNCB. L'app sncb : si facile de voyager! <http://www.belgianrail.be/fr/service-clientele/outils-voyage/itineraires-mobile/apps.aspx>.
- [47] Affluence. Est-ce qu'il y a du monde? <https://www.affluences.com/>.
- [48] ASBL Univers Santé. Qui sommes-nous? <http://www.univers-sante.be/qui-sommes-nous/>.
- [49] ASBL Univers Santé. Guindaille 2.0, mets-toi à jour! <http://www.guindaille2-0.be/>.
- [50] Gitlab. Where communities thrive. <https://gitter.im/>.
- [51] Amy Conchie. Hide api keys, 2016. <https://gist.github.com/derzorngottes/3b57edc1f996dddcab25>.

- [52] yahiko. Comprendre les promises en javascript / typescript, Juin 2017. <https://javascript.developpez.com/actu/146280/Comprendre-les-Promises-en-JavaScript-TypeScript-article-de-yahiko/>.
- [53] WSO2 Inc. Working with access tokens, 2018. <https://docs.wso2.com/display/AM220/Working+with+Access+Tokens>.
- [54] WSO2 Inc. Token api, 2018. <https://docs.wso2.com/display/AM220/Token+API>.
- [55] ionic team. Ionic storage, 2017. <https://github.com/ionic-team/ionic-storage>.
- [56] Crypho. Securestorage plugin for apache cordova, 2017. <https://github.com/Crypho/cordova-plugin-secure-storage>.
- [57] Masashi Katsumate. How different are the js apis and the mobile sdks?, 2017. <https://github.com/mapsplugin/cordova-plugin-googlemaps-doc/blob/master/v1.4.0/How-different-are-the-JS-APIs-and-the-mobile-SDKs/README.md>.
- [58] Google. Google maps javascript api limites d'utilisation, 2017. <https://developers.google.com/maps/documentation/javascript/usage?hl=fr>.
- [59] Mike Hartington. Ionic and lazy loading pt 1, Avril 2017. <https://blog.ionicframework.com/ionic-and-lazy-loading-pt-1/>.
- [60] Josh Morony. What to expect when ionic 4 is released, Mai 2018. <https://www.joshmorony.com/what-to-expect-when-ionic-4-is-released/>.
- [61] UTT Net Group. Oauth : qu'est ce que c'est, 2014. <http://e-tuutt.readthedocs.io/fr/latest/api/introduction/oauth.html>.
- [62] Dmitry Sotnikov. End-user authentication via api gateway, Août 2016. <https://wso2.com/blogs/cloud/end-user-authentication-via-api-gateway/>.

## Annexe A

# Enquête réalisée par les mémorants précédents

### UCLCampus

Formulaire destiné à découvrir vos envies, besoins et souhaits pour réaliser une application utile dans votre quotidien d'étudiants à l'UCL. Cette application sera réalisée par des étudiants en informatique dans le cadre de leur mémoire.  
Merci pour votre participation

1. Dans quelle faculté études-tu ?

---

2. Année d'étude

*Une seule réponse possible.*

☐ BAC

☐ Master

### UCLCampus Etude

3. Quelles fonctionnalités/outils veux-tu avoir dans le cadre de tes études ?

*Plusieurs réponses possibles.*

☐ Accès à mes points

☐ Un agenda reprenant mon horaire de cours et les auditoires

☐ Des notifications en cas de changements d'horaire de cours ou d'auditoires

☐ Accès à moodle

☐ Accès à la messagerie universitaire

☐ Horaire d'ouverture et de fermeture des bibliothèques

☐ Horaire d'ouverture et de fermeture des secrariats

☐ Localisation des salles d'études et des salles informatiques

☐ Savoir le nombre de places disponibles dans les bibliothèques

☐ Canaux d'interaction en temps réel pour poser ces questions aux professeurs lors d'un cours magistral

☐ Forum étudiants permettant de centraliser les questions sur un cours

☐ Accès aux informations sur les services offerts par l'UCL

☐ Accès au répertoire UCL

☐ Outil de gestion d'achat et de revente de livres en seconde main

☐ Informations sur la vie universitaire

4. As-tu d'autres idées/ remarques ?

---

---

---

---

---

## UCLCampus Vie sur le campus

5. Quelles fonctionnalités/outils veux-tu avoir dans le cadre de la vie sur le campus ?

*Plusieurs réponses possibles.*

- ☐ Agenda de tous les événements, sorties, soirées, festivités
- ☐ Horaire du sport CSE + réservation de terrains
- ☐ Réseau social spécifique à l'UCL (Communauté d'étudiants par centre d'intérêt, sport, musique, ...)
- ☐ Création personnelle d'événements pour une communauté, un groupe d'amis
- ☐ Savoir où il y a des places de parking dispo
- ☐ Partage de bons plans
- ☐ Description et géolocalisation des différents KAP's
- ☐ Description et géolocalisation des différents cercles
- ☐ Se géolocaliser sur une map
- ☐ Menu, horaire et prix des resto U
- ☐ Inviter des personnes à discuter, partager sa table lors d'un temps de midi

6. As-tu d'autres idées/remarques ?

---

---

---

---

---

## Annexe B

# Carnet des contacts liés au mémoire

Dans cette annexe, il sera regroupé sous forme de tableau l'ensemble des personnes contactées à propos du mémoire durant l'année. Pour chaque personne, il y aura au dessus le nom de son service tandis qu'à côté il y aura son adresse mail et la raison de la prise de contact.

| Nom                                                                    | Adresse email                      | Contacté à propos de                                                                      |
|------------------------------------------------------------------------|------------------------------------|-------------------------------------------------------------------------------------------|
| <b>Administration des relations extérieures et de la communication</b> |                                    |                                                                                           |
| Daniel Rahier                                                          | daniel.rahier@uclouvain.be         | Flux RSS des actualités de l'UCL                                                          |
| Pierre Escoyez                                                         | pierre.escoyez@uclouvain.be        | Flux RSS des actualités de l'UCL                                                          |
| Alain Nassogne                                                         | alain.nassogne@uclouvain.be        | Flux RSS des actualités de l'UCL                                                          |
| <b>Administration de la Vie étudiante</b>                              |                                    |                                                                                           |
| Jean Jacques Deheneffe                                                 | jeanjacques.deheneffe@uclouvain.be |                                                                                           |
| <b>Service général du système d'information</b>                        |                                    |                                                                                           |
| Jean-Luc Fortemaison                                                   | jean-luc.fortemaison@uclouvain.be  | Création d'un flux RSS pour l'horaire des sports                                          |
| Louis Destrée                                                          | louis.destree@uclouvain.be         | ESB WSO2 et authentification                                                              |
| Francis Somers                                                         | francis.somers@uclouvain.be        | ESB WSO2 (API Directories, Libraries et My)                                               |
| <b>Univers Santé</b>                                                   |                                    |                                                                                           |
| Martin de Duve                                                         | martin.deduve@uclouvain.be         | Application Guindaille2.0 pour les différents éléments de leur campagne et les événements |
| <b>Service des sports</b>                                              |                                    |                                                                                           |
| Nicolas Gilson                                                         | nicolas.gilson@uclouvain.be        | Information sur le service des sports                                                     |
| <b>Application NextRide</b>                                            |                                    |                                                                                           |
| Thomas Hermine                                                         | hello@thomashermine.be             | Ajout de l'application NextRide                                                           |

TABLE B.1 – Personnes contactée à propos du mémoire

# Annexe C

## Illustrations des requêtes et leurs réponses

### 1 Requêtes des Flux RSS

#### 1.1 Récupération du flux RSS des actualités de l'UCL

##### Requête

*`https://query.yahooapis.com/v1/public/yql?q=select%20*%20from%20rss%20where%20url%3D%27https://uclouvain.be/actualites/p1/rss%27&format=json`*

##### Réponse (limitée à deux news)

```
{
  "query": {
    "count": 1, "created": "2018-06-07T08:37:00Z", "lang": "fr-BE", "meta": {
      "url": {
        "id": "https://uclouvain.be/fr/actualites/p1/rss", "status": "200", "headers": {
          "header": [
            { "name": "Server", "value": "ATS" },
            { "name": "Content-Type", "value": "application/rss+xml" },
            { "name": "X-Drupal-Cache", "value": "MISS" },
            { "name": "Expires", "value": "Sun, 19 Nov 1978 05:00:00 GMT" },
            { "name": "Cache-Control", "value": "no-cache, must-revalidate" },
            { "name": "X-Content-Type-Options", "value": "nosniff" },
            { "name": "X-DRUPAL-HOSTNAME", "value": "drupal4" },
            { "name": "charset", "value": "UTF-8" },
            { "name": "Content-Language", "value": "fr" },
            { "name": "X-Frame-Options", "value": "SAMEORIGIN" },
            { "name": "Vary", "value": "Accept-Encoding" },
            { "name": "Content-Encoding", "value": "gzip" },
            { "name": "Access-Control-Allow-Origin", "value": "*" },
            { "name": "Strict-Transport-Security", "value": "max-age=15768000" },
            { "name": "Date", "value": "Thu, 07 Jun 2018 08:37:00 GMT" },
            { "name": "Age", "value": "3" },
            { "name": "Transfer-Encoding", "value": "chunked" },
            { "name": "Connection", "value": "keep-alive" },
            { "name": "Via", "value": "https/1.1 ecq2.ycs.bf1.yahoo.com (ApacheTrafficServer [cMsSfW])" }
          ]
        }
      }
    }
  }, "results": {
    "rss": {
      "version": "2.0", "atom": "http://www.w3.org/2005/Atom", "channel": {
        "link": [ { "href": "https://uclouvain.be/fr/actualites/p1/rss", "rel": "self", "type": "application/rss+xml" },
          { "href": "https://uclouvain.be/fr/actualites/p1" } ],
        "title": "Actualités du site p1", "description": "List of news of p1 site", "language": "fr-be", "copyright": "Copyright (C) 2018 https://uclouvain.be",
        "item": [
          {
            "title": "Cours préparatoires d'été",
            "description": "AOUT - SEPTEMBRE 2018\nLes cours préparatoires d'été s'adressent à vous qui comptez entamer des études universitaires ou supérieures et qui souhaitez prendre une longueur d'avance.\n\nProgramme complet de l'activité\n\n",
            "link": "https://uclouvain.be/fr/etudier/actualites/cours-preparatoires-d-ete.html",
            "pubDate": "Thu, 24 May 2018 16:59:38 +0200",
            "guid": "https://uclouvain.be/fr/etudier/actualites/cours-preparatoires-d-ete.html",
            "enclosure":
          }
        ]
      }
    }
  }
}
```

```

        {"length":"595900","type":"image/jpeg","url":"https://cdn.uclouvain.be/
        styles/rectangle_vertical/groups/cms-editors-rdvinfos/0329
        _pub_FB_prepa_ete_1200x628.jpg?itok=ARd9bR01"}},
        {"title":"75 ans de sciences de la motricité à l'UCL",
        "description":"\n\nL'UCL fete 75 ans d'enseignement de sciences de la motricité
        et de nombreuses évolutions en la matière : nouvelles technologies,
        recherches de pointe, notamment avec son laboratoire de l'effort, nouveaux
        cours... et étudiants plus nombreux que jamais ! Ce sont aujourd'hui 2
        000 étudiants en kiné ou éducation physique qui sont inscrits à l'UCL,
        soit 530 \% de plus qu'en 1977.\n\nEn réalité, la Faculté des sciences de
        la motricité de l'UCL(FSM) fête en 2018 un double anniversaire : les 75
        ans de l'Institut d'éducation physique et de réadaptation, créé ",
        "link":"https://uclouvain.be/fr/etudier/actualites/le-prof-de-gym-d-hier-
        utilisait-un-sifflet-aujourd'hui-une-tablette.html",
        "pubDate":"Tue, 22 May 2018 08:00:01 +0200",
        "guid":"https://uclouvain.be/fr/etudier/actualites/le-prof-de-gym-d-hier-
        utilisait-un-sifflet-aujourd'hui-une-tablette.html",
        "enclosure":
        {"length":"5725585","type":"image/gif","url":"https://cdn.uclouvain.be/
        styles/rectangle_vertical/groups/cms-editors-pl/fsm-actu.gif?itok=
        HNIYXI62"}},
        ...]}
    }
}
}

```

## 1.2 Récupération du flux RSS des événements

### Requête

*[https://query.yahooapis.com/v1/public/yql?q=select%20\\*%20from%20rss%20where%20url%3D%27http://louvainfo.be/calendrier/feed/calendar%27format=json](https://query.yahooapis.com/v1/public/yql?q=select%20*%20from%20rss%20where%20url%3D%27http://louvainfo.be/calendrier/feed/calendar%27format=json)*

### Réponse (limitée à deux événements)

```

{
  "query":{
    "count":82,"created":"2018-06-07T09:45:57Z","lang":"fr-BE",
    "results":{
      "item":[
        {"title":"RHIZOMATICS - Une balade numérique et littéraire",
        "link":"http://louvainfo.be/calendrier/2330/",
        "description":"<p>Au gr&eacute; des sons, des vibrations, des mots et des images,
        des exp&eacute;riences tactiles et sensorielles, red&eacute;couvrez les rues
        et places de Louvain-la-Neuve avec votre smartphone&hellip;&nbsp;</p>\n\n<p>En
        Aout dernier, les participants de l&rsquo;atelier d&rsquo;écriture &
        laquo; Ecrire pour le num&eacute;rique &raquo;, ont invent&eacute; des mots,
        des bribes de textes et de phrases, inspir&eacute;s de leurs d&eacute;
        ambulations dans la ville. Ce contenu a ensuite &eacute;,
        "guid":"event-6072",
        "location":"Louvain-la-Neuve - Espace public (départ: Inforville)",
        "date_begin":"07/06/2018 00:00",
        "date_end":"07/06/2018 23:59",
        "photo":"http://louvainfo.be/media/events/rhizomatics-940x616.jpg",
        "category":"Culturel et artistique",
        "organisateur":"autre",
        "prix":"0"},
        {"title":"Expo temporaire Musée L: Une passion pour l'art belge!",
        "link":"http://louvainfo.be/calendrier/2716/",
        "description":"<p>&nbsp;</p>\n\n<p>Docteur en histoire de l&rsquo;art, galeriste
        passionn&eacute; et instigateur de la Fondation pour l&rsquo;art belge
        contemporain, Serge Goyens de Heusch s&rsquo;est enti&egrave;rement d&eacute;
        di&eacute; &agrave; l&rsquo;art, en particulier aux artistes belges, avec
        lesquels il a entretenu des liens forts d&rsquo;amiti&eacute;. Cette
        exposition pr&eacute;sente une nouvelle donation de tableaux d&rsquo;artistes
        modernes belges (Delahaut, Bertrand, Lismonde) et de dessins et e",
        "guid":"event-7642",
        "location":"Musée L, place des sciences, Louvain-la-Neuve",
        "date_begin":"08/06/2018 08:32",
        "date_end":"08/06/2018 17:32",
        "photo":"http://louvainfo.be/media/events/AM1601_-_copie.jpg",
        "category":"Culturel et artistique",
        "organisateur":"autre",
        "prix":"6"}],
      ...}
    }
}

```

## 1.3 Récupération du flux RSS de l'horaire des sports

### Requête

*[http://ucl-fms01.sipr.ucl.ac.be:82/ucl\\_sport/rsssport.php?action=t1](http://ucl-fms01.sipr.ucl.ac.be:82/ucl_sport/rsssport.php?action=t1)*

### Réponse (fichier XML, limité à deux sports)



```

<?xml version="1.0" encoding="utf-8" ?><rss version="2.0">
<channel>
<title>UCL – Service des Sports</title>
<description>Horaire : Etudiant Louvain–La–Neuve</description>
<lastBuildDate>07/06/2018</lastBuildDate>
<link>http://ucl-fms01.sipr.ucl.ac.be:82/ucl_sport/recordlist.php?action=t1</link>
<item>
<title>
<![CDATA[NATATION]]>
</title>
<sport>
<![CDATA[NATATION]]>
</sport>
<sexe>
<![CDATA[ ]]>
</sexe>
<lieu>
<![CDATA[ PISC BAS]]>
</lieu>
<salle>
<![CDATA[ Selon dispo]]>
</salle>
<jour>
<![CDATA[ Jeudi]]>
</jour>
<date>
<![CDATA[ 07/06/2018]]>
</date>
<hdebut>
<![CDATA[ 16:00 ]]>
</hdebut>
<hfin>
<![CDATA[ 17:00 ]]>
</hfin>
<type>
<![CDATA[ Accès libre]]>
</type>
<online>
<![CDATA[ ]]>
</online>
<remarque>
<![CDATA[ ]]>
</remarque>
<active>
<![CDATA[ Oui ]]>
</active>
</item>
<item>
<title>
<![CDATA[FITNESS CARDIOTRAINING]]>
</title>
<sport>
<![CDATA[FITNESS CARDIOTRAINING]]>
</sport>
<sexe>
<![CDATA[ ]]>
</sexe>
<lieu>
<![CDATA[ CSB]]>
</lieu>
<salle>
<![CDATA[ LIFE STYLE]]>
</salle>
<jour>
<![CDATA[ Jeudi]]>
</jour>
<date>
<![CDATA[ 07/06/2018]]>
</date>
<hdebut>
<![CDATA[ 17:00 ]]>
</hdebut>
<hfin>
<![CDATA[ 18:00 ]]>
</hfin>
<type>
<![CDATA[ Accès libre]]>
</type>
<online>
<![CDATA[ Oui]]>
</online>
<remarque>
<![CDATA[ ]]>
</remarque>
<active>
<![CDATA[ Oui ]]>
</active>
</item>
...
</channel>
</rss>

```

## 2 Requêtes ADE

### 2.1 Ouverture de session

#### Requête

*http://horaire.uclouvain.be/jsp/webapi?function=connect&login=etudiant&password=student*

#### Réponse

```
<?xml version="1.0" encoding="UTF-8"?>
<session id="163d929f1f8"/>
```

### 2.2 Récupération des projets

#### Requête

*http://horaire.uclouvain.be/jsp/webapi?sessionId=163d929f1f8&function=getProjects&detail=2*

#### Réponse

```
<?xml version="1.0" encoding="UTF-8"?>
<projects>
  <project id="1" name="2017-2018" uid="1520345904235"/>
</projects>
```

### 2.3 Sélection d'un projet

#### Requête

*http://horaire.uclouvain.be/jsp/webapi?sessionId=163d929f1f8&function=setProject&projectId=1*

#### Réponse

```
<?xml version="1.0" encoding="UTF-8"?>
<setProject sessionId="163d929f1f8" projectId="1"/>
```

### 2.4 Récupération de l'id ADE du cours LINGI1123

#### Requête

*http://horaire.uclouvain.be/jsp/webapi?sessionId=163d929f1f8&function=getResources&code=LINGI1123*

#### Réponse

```
<?xml version="1.0" encoding="UTF-8"?>
<resources>
  <resource id="13148"/>
</resources>
```

### 2.5 Récupération des activités du cours LINGI1123

#### Requête

*http://horaire.uclouvain.be/jsp/webapi?sessionId=163d929f1f8&function=getActivities&resources=13148&detail=17*

## Réponse

```
<?xml version="1.0" encoding="UTF-8"?>
<activities>
  <activity id="30010" name="LINGI1123=E_MAJ" type="Examen_ecrit" folderId="11888" url="" size="100" repetition="1" lastUpdate="06/07/2018_09:01" creation="01/15/2015_10:17" lastSlot="14" lastDay="4" lastWeek="38" firstSlot="2" firstDay="4" firstWeek="38" additionalResources="0" durationInMinutes="180" nbEventsPlaced="1" nbEvents="1" duration="180" email="" weight="0" seatsLeft="" maxSeats="129" info="" codeZ="" codeY="2016" codeX="80001" timezone="Q2:30.0+30.0" code="CALCULABILITE" color="255,255,255" isActive="true" isNotSameDay="false" isGrouped="false" isAligned="false" isSuccessiveDays="false" ownerId="92" owner="verheydenj">
    <events>
      <event id="104249" activityId="30010" session="0" repetition="0" name="LINGI1123=E_MAJ" endHour="11:30" startHour="08:30" date="08/06/2018" absoluteSlot="17282" slot="2" day="4" week="38" additionalResources="0" duration="12" info="" note="Les_étudiants_du_master_en_cybersecurite_passent_à_l'examen_le_8_juin_2018.&#10;&#10;Repartition_par_ordre_alphabetique_stricte:&#10;&#10;SUD03(20)_de_Aid,_Ayman_Cabrera_Alba,_Laritza&#10;&#10;BARB94(55):_de_Calbert,_Simon_Mawait,_Maxime&#10;&#10;BARB92(51):_Mbonyincungu,_Jean-Pacifique_Xenakis,_Alexandros" color="255,255,255" isLockPosition="false" lastUpdate="02/28/2018_11:23" creation="01/22/2018_16:12" isLockResources="false" isSoftKeepResources="false">
        <eventParticipants>
          <eventParticipant fromWorkflow="false" nodeId="158821" nodeOrId="-1" quantity="1" category="trainee" name="MAP2M_C._au_Choix_C." id="8022"/>
          <eventParticipant fromWorkflow="false" nodeId="158824" nodeOrId="-1" quantity="1" category="trainee" name="SINF1PM" id="3754"/>
          <eventParticipant fromWorkflow="false" nodeId="207785" nodeOrId="-1" quantity="1" category="trainee" name="Majinfo13" id="7970"/>
          <eventParticipant fromWorkflow="false" nodeId="207786" nodeOrId="-1" quantity="1" category="trainee" name="sinf13ba" id="7982"/>
          <eventParticipant fromWorkflow="false" nodeId="207787" nodeOrId="-1" quantity="1" category="trainee" name="sinf1PM" id="7983"/>
          <eventParticipant fromWorkflow="false" nodeId="207831" nodeOrId="-1" quantity="1" category="category5" name="LINGI1123" id="13148"/>
          <eventParticipant fromWorkflow="true" nodeId="193809" nodeOrId="221721" quantity="1" category="classroom" name="SUD_03" id="2306"/>
          <eventParticipant fromWorkflow="true" nodeId="193786" nodeOrId="221721" quantity="1" category="classroom" name="BARB_94" id="32"/>
          <eventParticipant fromWorkflow="true" nodeId="121339" nodeOrId="221721" quantity="1" category="classroom" name="BARB_92" id="31"/>
          <eventParticipant fromWorkflow="false" nodeId="207833" nodeOrId="207832" quantity="1" category="instructor" name="Deville_Yves" id="278"/>
        </eventParticipants>
        <additional/>
      </event>
    </events>
    <resources activityId="30010">
      <and load="1.0" quantity="1" nodeId="158821" id="8022" name="MAP2M_C._au_Choix_C._d_interet" path="EPL.EPL_Cycle_2.EPL_Masters_120.MAP2M.MAP2M_Options.MAP2M_C._au_Choix." category="trainee" isGroup="false"/>
      <and load="1.0" quantity="1" nodeId="158824" id="3754" name="SINF1PM" path="EPL.EPL_Cycle_2.EPL_Masters_120.SINF2M1." category="trainee" isGroup="false"/>
      <and load="1.0" quantity="1" nodeId="207785" id="7970" name="Majinfo13" path="EPL.EPL_Cycle_1.EPL-niveau_13.fsa13BA-minmaj." category="trainee" isGroup="false"/>
      <and load="1.0" quantity="1" nodeId="207786" id="7982" name="sinf13ba" path="EPL.EPL_Cycle_1.SINF1BA." category="trainee" isGroup="false"/>
      <and load="1.0" quantity="1" nodeId="207787" id="7983" name="sinf1PM" path="EPL.EPL_Cycle_1.SINF1BA." category="trainee" isGroup="false"/>
      <and load="1.0" quantity="1" nodeId="207831" id="13148" name="LINGI1123" path="SST/_EPL.DEP_INGI." category="category5" isGroup="false"/>
      <or isContinuous="false" category="instructor" quantity="1" id="207832">
        <and load="1.0" quantity="1" nodeId="207833" id="278" name="Deville_Yves" path="Personnel_UCL.D." category="instructor" isGroup="false"/>
      </or>
      <orRequest isContinuous="false" category="classroom" quantity="3" id="221721" senderId="98" copyByMail="false" message="" subject="Requete">
        <recipients managerUsers="true"/>
        <and load="1.0" quantity="1" nodeId="121339" id="31" name="BARB_92" path="Zone_Sciences_et_technologies.Auditoires_Sainte_Barbe_(BARB)." category="classroom" isGroup="false"/>
        <and load="1.0" quantity="1" nodeId="193786" id="32" name="BARB_94" path="Zone_Sciences_et_technologies.Auditoires_Sainte_Barbe_(BARB)." category="classroom" isGroup="false"/>
        <and load="1.0" quantity="1" nodeId="193809" id="2306" name="SUD_03" path="Zone_Sciences_et_technologies.Auditoires_Croix_du_Sud_(SUD).Croix_du_Sud_Ac_(02-03-04-12-13-14)." category="classroom" isGroup="false"/>
      </orRequest>
    </resources>
    <rights othersRights="read" groupRights="rw" userRights="none" group="root" user="root" profileName="W_ADE/_U_Appariteurs/_U_R_OTHER" profileId="83"/>
  </activity>
  ...
</activities>
```

## 3 Requêtes à WSO2

### 3.1 Récupération du token pour l'application

#### Requête

```
curl -k -d "grant_type=client_credentials" -H "Authorization : Basic Base64(consumer-key :consumer-secret)" https://api.sgsi.ucl.ac.be :8243/token
```

#### Réponse

```
{
  "access_token": "TOKEN",
  "scope": "am_application_scope default",
  "token_type": "Bearer",
  "expires_in": 3600
}
```

### 3.2 Récupération de la liste des bibliothèques

#### Requête

```
https://api.sgsi.ucl.ac.be :8243/libraries/v1/list
```

#### Réponse

```
{
  "return": {
    "library": [
      {
        "id": 33,
        "name": "UCL - BDRT: Bibliothèque de droit"
      },
      {
        "id": 43,
        "name": "UCL - BFLT: Bibliothèque des arts et des lettres"
      },
      {
        "id": 5633,
        "name": "UCL - BSPO: Bibliothèque de sciences économiques, sociales et politiques"
      },
      ...
    ],
    "count": 13
  }
}
```

### 3.3 Récupération des informations à propos de la BSPO

#### Requête

```
https://api.sgsi.ucl.ac.be :8243/libraries/v1/list/5633
```

#### Réponse

```
{
  "return": {
    "library": {
      "id": 5633,
      "name": "UCL - BSPO: Bibliothèque de sciences économiques, sociales et politiques",
      "locationId": 700000,
      "address": {
        "street": "Place Montesquieu 1, bte L2.08.02",
        "place": "Place Montesquieu 5 (entrée)",
        "locality": "Louvain-la-Neuve",
        "postalCode": "B-1348",
        "country": "BE"
      },
      "phone": "+32 (0) 10 47 41 66",
      "email": "aide-bspo@uclouvain.be",
      "website": "https://uclouvain.be/fr/bibliotheques/bspo",
      "openingHours": [
        {
          "day": "Lundi",
          "startHour": "08:30",

```

```

        "endHour": "19:30"
    },
    {
        "day": "Mardi",
        "startHour": "08:30",
        "endHour": "19:30"
    },
    {
        "day": "Mercredi",
        "startHour": "08:30",
        "endHour": "19:30"
    },
    {
        "day": "Jeudi",
        "startHour": "08:30",
        "endHour": "19:30"
    },
    {
        "day": "Vendredi",
        "startHour": "08:30",
        "endHour": "18:00"
    },
    {
        "day": "Samedi",
        "startHour": "09:00",
        "endHour": "17:00"
    }
],
"openingHoursNote": "La bibliothèque sera ouverte les samedis 13, 20, 27 mai et les samedis 3, 10 et 17 juin de 9h à 17h.",
"closedDates": [
    {
        "from": "Jeudi, 25 mai 2017",
        "description": "Ascension"
    },
    {
        "from": "Lundi, 5 juin 2017",
        "description": "lundi de Pentecote"
    },
    {
        "from": "Vendredi, 21 juillet 2017",
        "description": "fete nationale"
    },
    {
        "from": "Mardi, 15 aout 2017",
        "description": "Assomption"
    },
    {
        "from": "Mercredi, 27 septembre 2017",
        "description": "Fête de la Communauté française"
    },
    {
        "from": "Mercredi, 1 novembre 2017",
        "description": "Toussaint"
    },
    {
        "from": "Mardi, 16 mai 2017",
        "description": "fermé"
    }
]
}
}
}
}

```

### 3.4 Recherche du nom Deville dans le répertoire UCL

#### Requête

*<https://api.sgsi.ucl.ac.be:8243/directories/v1/employees/search?lastname=deville>*

#### Réponse

```

{
  "persons": {
    "person": [
      {
        "matric_fgs": "00030908",
        "directory": "E",
        "lastname": "De Villé",
        "firstname": "Philippe",
        "lastnameDisplay": "De Villé",
        "firstnameDisplay": "Philippe",
        "email": "philippe.deville@uclouvain.be",
        "departments": {
          "department": [
            {
              "linkType": "Affectation",
              "entity": {
                "entity_id": "01000062",
                "acronym": "ESPO",
                "acronyms": "SSH/ESPO",
                "departmentType": "F",
                "name_fr": "Faculté des sciences économiques, sociales, politiques et de communication",

```

```

        "name_en": "Faculty of Economic, Social and Political Sciences and Communication"
    }
  ]
},
"rownum": 1
},
{
  "matric_fgs": "00030910",
  "directory": "E",
  "lastname": "Deville",
  "firstname": "Yves",
  "lastnameDisplay": "Deville",
  "firstnameDisplay": "Yves",
  "email": "yves.deville@uclouvain.be",
  "departments": {
    "department": [
      {
        "linkType": "Affectation",
        "entity": {
          "entity_id": "01000363",
          "acronym": "EPL",
          "acronyms": "SST/EPL",
          "departmentType": "F",
          "name_fr": "Ecole Polytechnique de Louvain",
          "name_en": "Louvain School of Engineering"
        }
      },
      {
        "linkType": "Affectation",
        "entity": {
          "entity_id": 50024117,
          "acronym": "ICTM",
          "acronyms": "SST/ICTM",
          "fullAcronym": "ICTEAM",
          "departmentType": "I",
          "name_fr": "Institute of Information and Communication Technologies, Electronics and Applied Mathematics"
        }
      },
      {
        "linkType": "Appartenance",
        "entity": {
          "entity_id": "01000396",
          "acronym": "INGI",
          "acronyms": "SST/ICTM/INGI",
          "fullAcronym": "INGI",
          "departmentType": "P",
          "name_fr": "Pole en ingénierie informatique"
        }
      }
    ]
  },
  "rownum": 2
}
]
}
}
}

```

### 3.5 Récupération des informations à propos de M. Yves Deville

#### Requête

*<https://api.sgsi.ucl.ac.be:8243/directories/v1/employees/00030910/detail>*

#### Réponse

```

{
  "businessInformation": {
    "matric_fgs": "00030910",
    "photo_url": "http://photos.carte.uclouvain.be/photos_cp/photos/image09/00030910.jpg",
    "matric_sap": "00209560",
    "gender": "M",
    "directory": "E",
    "lastname": "Deville",
    "firstname": "Yves",
    "lastnameDisplay": "Deville",
    "firstnameDisplay": "Yves",
    "email": "yves.deville@uclouvain.be",
    "contracts": {
      "contract": [
        {
          "idm": 1,
          "grade": "professeur ordinaire"
        }
      ]
    }
  },
  "departments": {
    "department": [
      {
        "linkType": "Affectation",
        "entity": {
          "entity_id": "01000363",
          "acronym": "EPL",

```



## Réponse

```
{
  "activities": {
    "activity": [
      "SIGLE1",
      "SIGLE2",
      ...
      "SIGLEN"
    ]
  }
}
```



## Annexe D

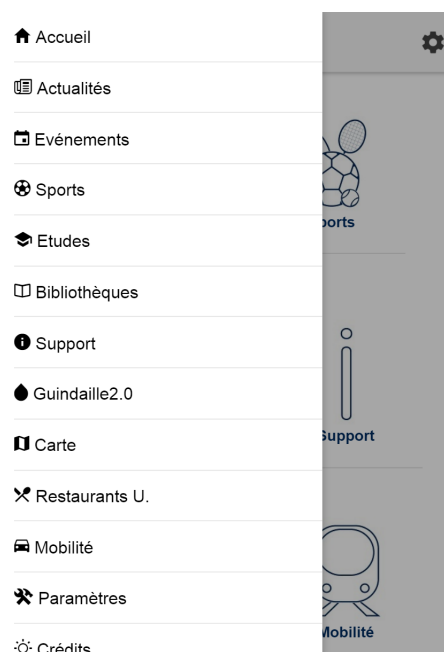
# Guide d'utilisateur

Ce guide a pour objectif de faire découvrir les différentes fonctionnalités proposées par l'application UCLCampus à l'utilisateur. L'application UCLCampus est une application mobile multiplateforme supportée par tous les appareils tournant sur Android version 4.4 et plus ainsi que sur iOS version 8 et plus.

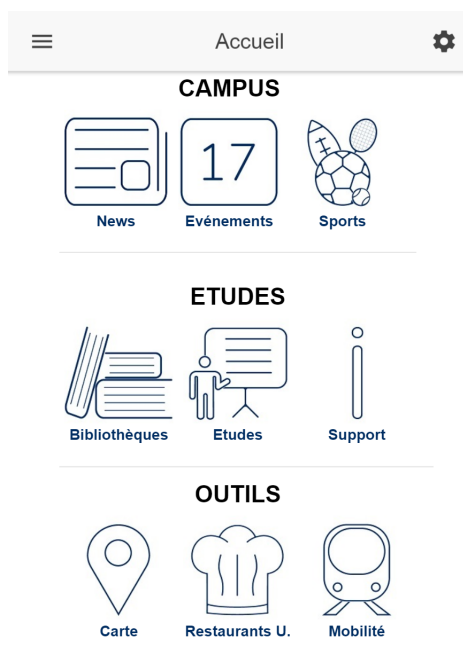
Elle propose de regrouper les multiples services et outils qui peuvent être utiles à un utilisateur sur l'un des campus de l'Université catholique de Louvain.

## 1 Menu

Un menu, donnant accès à toutes les fonctionnalités de l'application, est disponible en glissant le bord gauche de l'écran vers la droite ou grâce au bouton en haut à gauche de la page d'accueil.



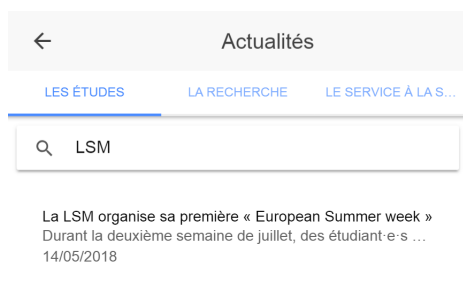
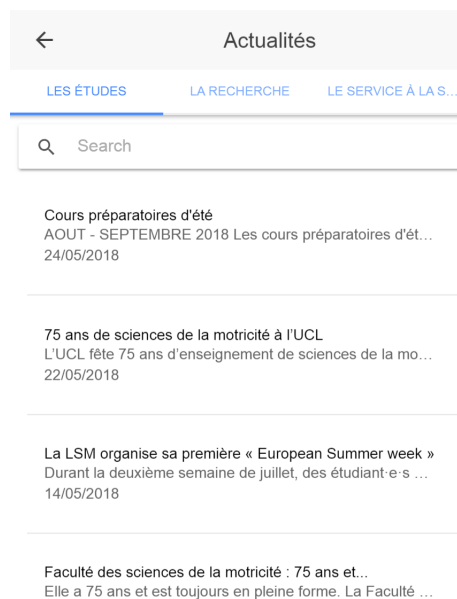
## 2 Accueil



La page "Accueil" est la première page à s'afficher lors de l'ouverture de l'application. Durant le premier lancement, il est également demandé à l'utilisateur dans quel campus il se situe. La page "Accueil" permet de naviguer entre les différentes fonctionnalités de l'application. Il suffit simplement d'appuyer sur une icône pour s'y rendre. Il est également possible d'accéder au menu des fonctionnalités en appuyant en haut à gauche ainsi que de changer de langue et de campus à l'aide du bouton paramètres en haut à droite.

## 3 Actualités

La page "Actualités" donne accès à trois onglets affichant chacun un flux d'actualités provenant du portail UCL : les études, la recherche ainsi que le service à la société. Les news composant les listes de chaque onglet comprennent le titre de la nouvelle, sa date et le début d'une description détaillée. Il suffit d'appuyer sur une nouvelle spécifique pour en avoir tous les détails. Il est possible de retourner à l'accueil à l'aide de la flèche en haut à gauche.

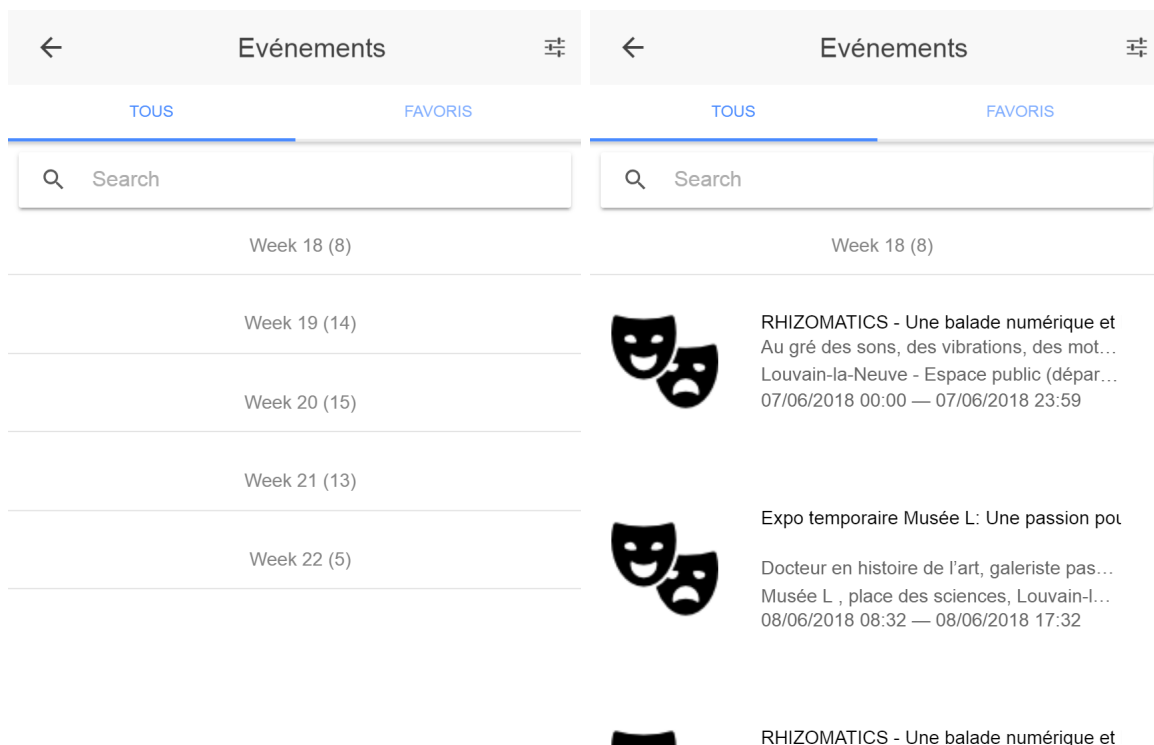


Une barre de recherche est également disponible en haut de la page. Celle-ci permet de rechercher une nouvelle avec des mots-clés. La recherche se fait automatiquement et les actualités sont filtrées instantanément.

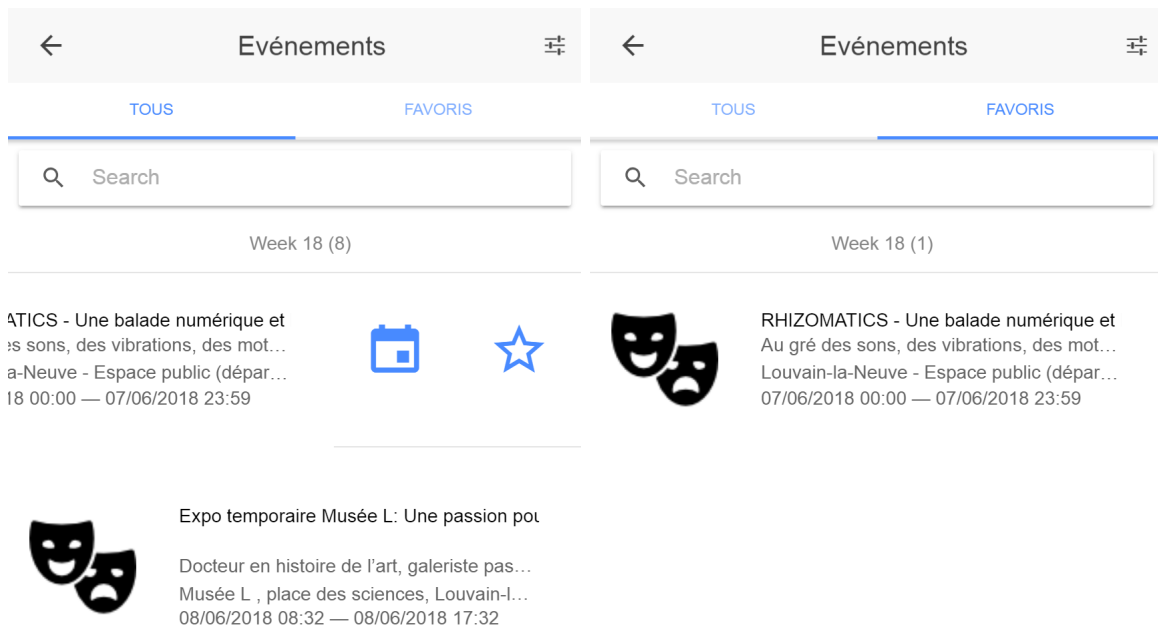
Lorsqu'on appuie sur une nouvelle, une autre page s'ouvre avec les informations détaillées de celle-ci. Cette page est composée du titre de la nouvelle, d'une éventuelle image la représentant, d'une description détaillée ainsi que d'un bouton "VOIR SUR LE SITE" afin de s'y rendre directement via un navigateur. La flèche en haut à gauche permet de retourner sur la liste des actualités. Il est également possible d'y retourner avec le bouton retour de l'appareil.



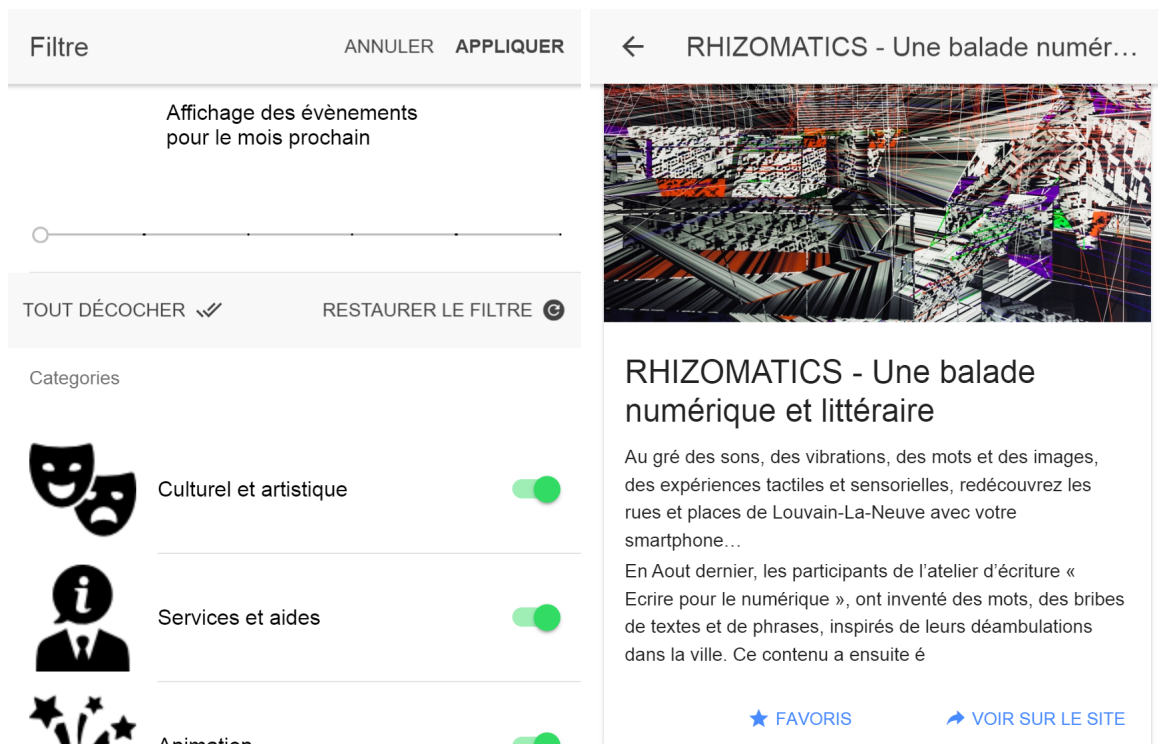
## 4 Évènements



La page "Évènements" donne accès à deux onglets. Le premier, "TOUS", affiche la liste d'évènements en cours ou à venir sur le campus. Cette liste est divisée en semaines. Lorsqu'on appuie sur une semaine, la liste des évènements de la semaine sélectionnée se déroule. Ces évènements sont directement tirés du calendrier du site [louvainfo.be](http://louvainfo.be) maintenu par le kot à projet CarpeStudentem. Il y a également une barre de recherche en haut de la page qui fonctionne de la même manière que sur la page Actualités. Chaque évènement est présenté avec son titre, le début de sa description détaillée, l'endroit où se déroule l'activité et sa date. Il y a un pictogramme à sa droite qui correspond à la catégorie à laquelle il appartient. Le bouton en haut à droite permet d'ouvrir une fenêtre permettant de filtrer la liste des évènements. Il est possible de revenir à l'accueil grâce à la flèche en haut à gauche ou le bouton retour de l'appareil.



Glisser un évènement vers la gauche fait apparaître deux nouveaux boutons. Le premier, en forme de calendrier, permet d'ajouter l'évènement à l'agenda natif de l'appareil. Le second, en forme d'étoile, permet d'ajouter l'évènement en question à une liste de favoris accessible via le deuxième onglet de la page. Lors de l'ouverture de cet onglet, une page similaire à la première s'affiche et il est également possible d'y faire glisser les évènements afin que deux boutons apparaissent. Le premier ajoute l'évènement à l'agenda natif et le deuxième, en forme de croix, permet de supprimer cet évènement de la liste de favoris.



Comme mentionné précédemment, un bouton permettant de filtrer les évènements est disponible en haut à droite. Lorsqu'on appuie dessus, une nouvelle page s'ouvre et il est possible de

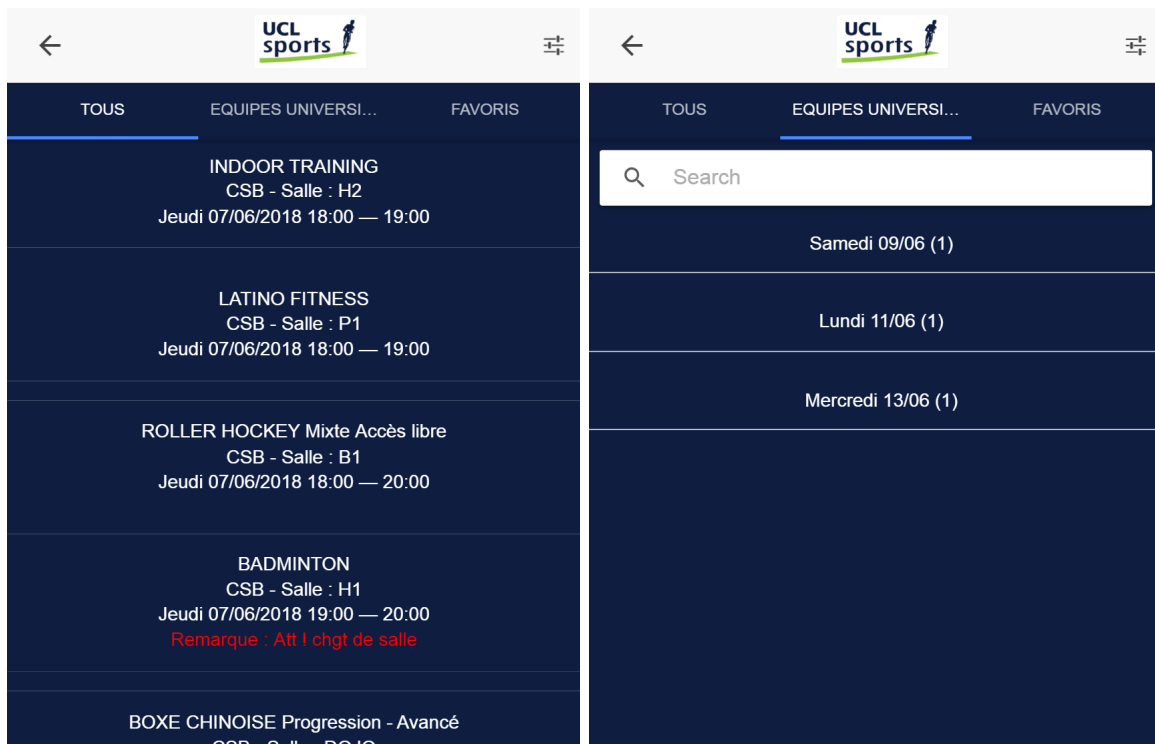
définir la durée pour laquelle vous voulez afficher les activités. Il y a aussi toutes les catégories avec leur pictogramme, il suffit d'appuyer sur le petit rond vert à leur droite pour les désactiver. Il est possible de tout décocher d'une seule fois à l'aide du bouton "Tout décocher" ou faire l'inverse avec le bouton "Restaurer le filtre". Une fois les filtres sélectionnés, il suffit d'appuyer sur "Appliquer" en haut de la page afin de voir la nouvelle liste d'activités respectant ces critères.

Lorsqu'on appuie sur un événement, une nouvelle page s'affiche avec les informations détaillées le concernant : le titre de l'évènement, une description détaillée, une image représentant l'activité, un bouton permettant de l'ajouter aux favoris ainsi qu'un bouton "VOIR SUR LE SITE" qui permet d'ouvrir un navigateur et de se rendre sur le site ayant publié l'évènement.

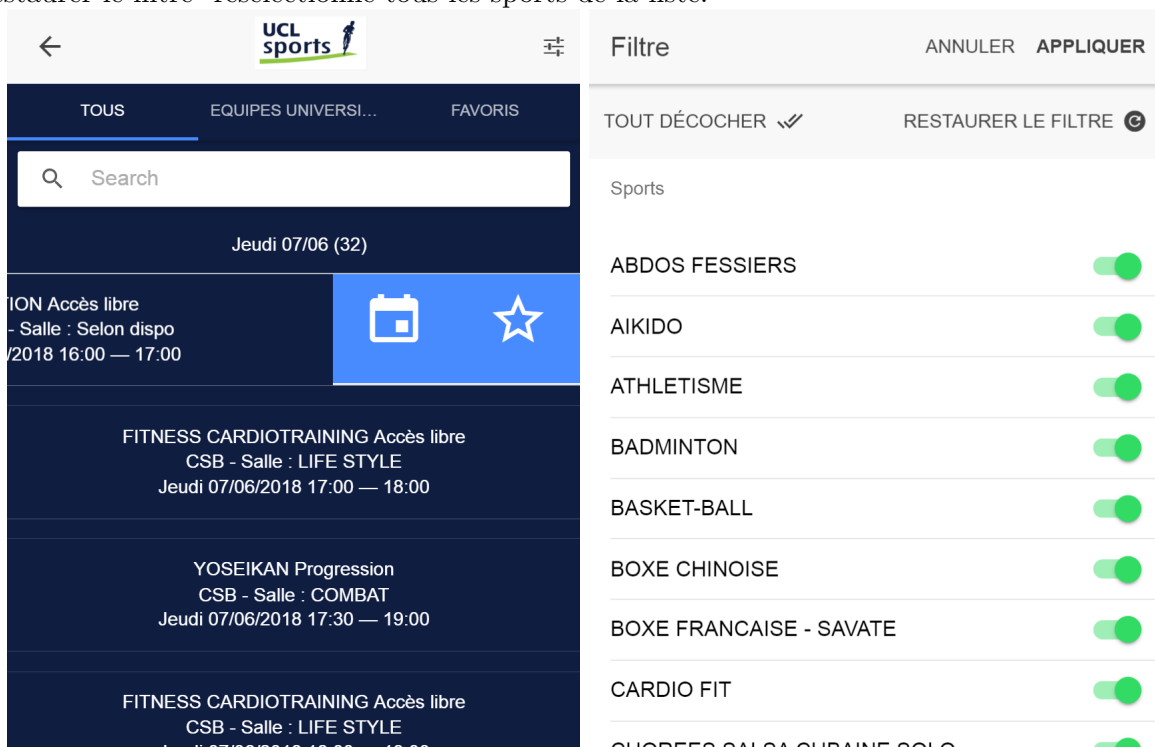
## 5 Sports



La page "Sports" s'articule à peu près de la même manière que la page "Évènements". Elle est composée de trois onglets. L'onglet "Tous" regroupe une liste de sports sur une durée d'une semaine. Cette liste est séparée en jour avec le nombre de sports entre parenthèses pour chacun d'eux. Lorsqu'on appuie sur une des journées, une liste de tous les sports disponibles se déroule. Chaque sport est présenté avec son nom, le sexe ciblé si ce n'est pas en accès libre, le bâtiment, la salle, la date, l'heure à laquelle il se déroule ainsi qu'une note en rouge (visible sur la troisième image) si une remarque est à afficher. Une barre de recherche est également disponible en haut et s'utilise de la même manière qu'expliqué précédemment. Il y a un bouton en haut à droite qui sert à filtrer la liste.



L'onglet "Équipes Universitaires" fonctionne exactement de la même manière que l'onglet précédent mais affiche une liste propre aux entraînements des équipes universitaires. Dans chacun de ces deux onglets, il est possible de faire glisser de la droite vers la gauche un sport avec son doigt. Cela permet d'afficher deux nouveaux boutons. Le premier sert à ajouter le sport au calendrier natif de l'appareil et le second permet d'ajouter un sport ou un entraînement au troisième onglet qui est une liste des favoris ajoutés. Lorsqu'on appuie sur le bouton servant à filtrer les sports, une nouvelle page s'ouvre avec la liste de tous les sports proposés. Il suffit d'appuyer sur "Tout décocher" et de cocher que les sports désirés pour afficher que ceux-ci. Une fois la sélection faite, il suffit d'appuyer sur "appliquer" afin de filtrer la liste. Le bouton "Restaurer le filtre" resélectionne tous les sports de la liste.

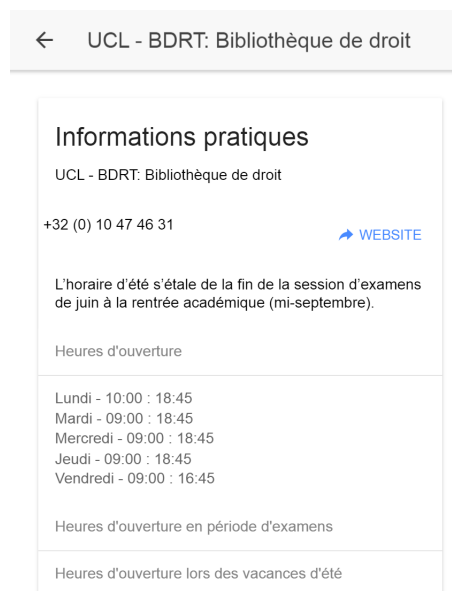


## 6 Bibliothèques



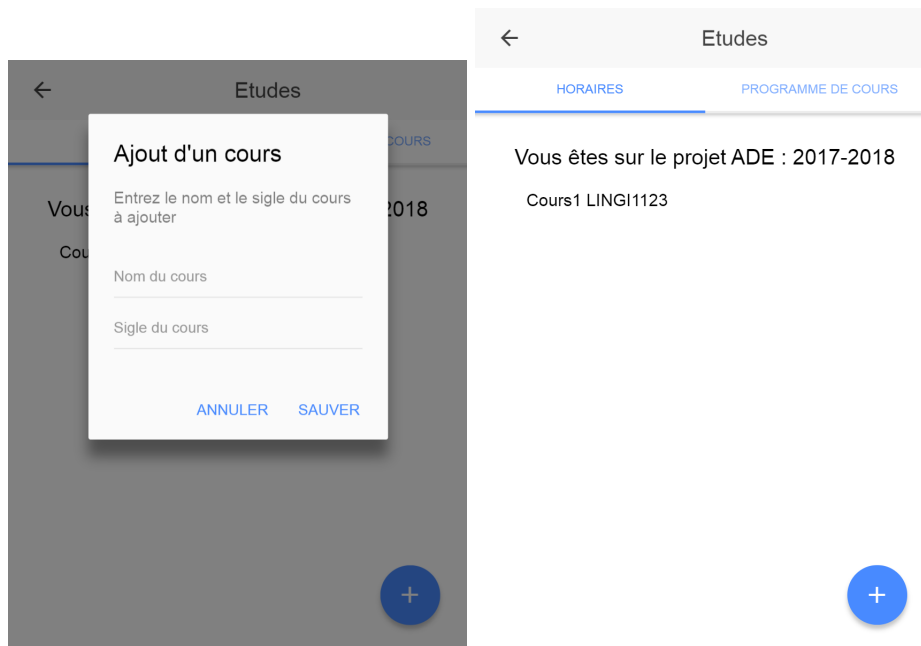
Lorsque l'on a appuyé sur l'une des bibliothèques, une page s'affiche avec toutes les informations importantes à son propos. Tout d'abord, il y a son nom complet avec son acronyme. Ensuite, il y a le numéro de téléphone suivi, parfois, de diverses informations pratiques. Il y a également un lien permettant de se rendre directement sur le site web de la bibliothèque. Les différents horaires de celle-ci sont ensuite présentés. Ceux-ci se divisent en trois périodes : l'horaire normal, l'horaire en période d'examens et l'horaire en période de vacances d'été. Il est possible de consulter les jours de fermetures de la bibliothèque.

La page "Bibliothèques" présente une liste des différentes bibliothèques de l'Université catholique de Louvain. Pour obtenir de plus amples informations sur l'une d'elle, il suffit d'appuyer dessus.

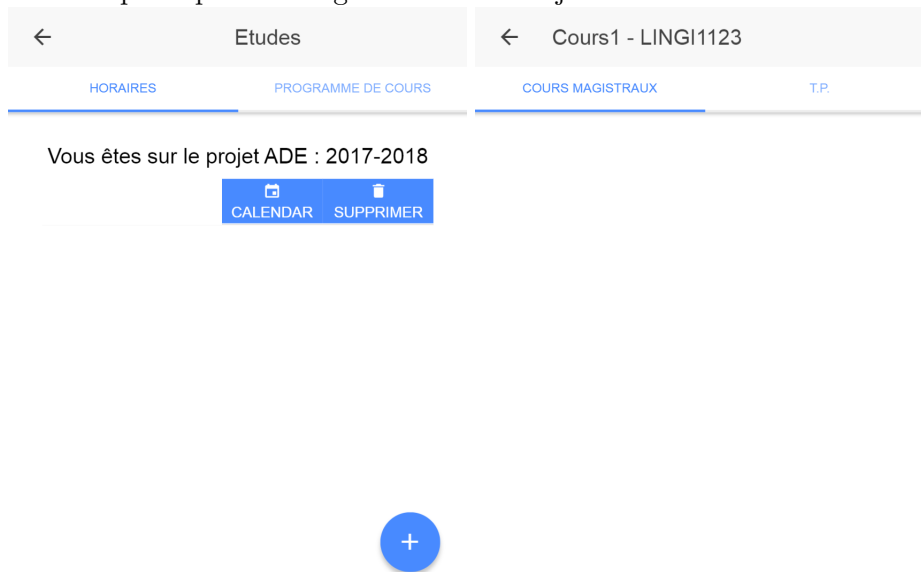


## 7 Études

La page "Études" permet de directement gérer son horaire de cours. Celui-ci est lié au service ADE Expert de l'université. Lors du premier lancement de la page, il est demandé de sélectionner l'un des projets ADE disponibles durant l'année en cours. Une fois sélectionné, une autre page s'affiche avec deux onglets. Le premier, l'onglet "Horaire", reprend la liste des différents cours ajoutés. Afin d'ajouter un cours, il suffit d'appuyer sur le bouton rond avec une croix dedans et une fenêtre de dialogue apparaît. Dans celle-ci, il est nécessaire de rentrer le sigle exact du cours désiré, de lui donner un nom au choix et de terminer en appuyant sur "Sauver". Le cours s'ajoute immédiatement à la liste.



Sur cette liste, il est possible de faire glisser vers la gauche afin de faire apparaître deux boutons. Le premier permet d'ajouter toutes les séances du cours à l'agenda natif de l'appareil tandis que le second permet de supprimer le cours de la liste. Lorsqu'on appuie sur un cours, une autre page s'ouvre. Celle-ci présente toutes les séances de ce cours. Ces séances sont séparées via deux onglets : "T.P." et "Cours magistraux". Sur cette page, il est également possible de glisser une séance spécifique vers la gauche afin de l'ajouter au calendrier natif de l'appareil.



Le second onglet, l'onglet "Programme de cours", affiche un formulaire permettant à un étudiant de l'UCL de se connecter grâce à ses identifiants. Une fois le formulaire complété et le bouton "Login" appuyé, une liste s'affiche reprenant tous les cours de l'étudiant pour l'année en cours. Seuls les sigles sont visibles, les noms ne sont pas affichés.



| ← Etudes |                    | ← Etudes  |                    |
|----------|--------------------|-----------|--------------------|
| HORAIRES | PROGRAMME DE COURS | HORAIRES  | PROGRAMME DE COURS |
| Username |                    | LINGI2369 |                    |
| Password |                    | LFSA2995  |                    |
| LOGIN    |                    | LINGI2146 |                    |
|          |                    | LINGI2990 |                    |
|          |                    | LINGI2266 |                    |

Néanmoins, il est possible de faire glisser chaque sigle vers la gauche afin de l'ajouter à la liste présente dans l'autre onglet et en même temps de lui attribuer un nom. Quel que soit l'onglet sur lequel on se situe, un glissement du doigt partant du bord droit de l'écran vers la gauche permet d'ouvrir un menu latéral permettant de se rendre sur le site du portail UCL, le site Moodle ainsi que de modifier le projet ADE sélectionné préalablement.

| ← Etudes  |                    | ←                  |                    |
|-----------|--------------------|--------------------|--------------------|
| HORAIRES  | PROGRAMME DE COURS | HORAIRES           | PROGRAMME DE COURS |
| LINGI2369 |                    | uclouvain.be       |                    |
| LFSA2995  |                    | Moodle ucl         |                    |
| LINGI2146 |                    | Changer projet ADE |                    |
| LINGI2990 |                    |                    |                    |
| LINGI2266 |                    |                    |                    |

## 8 Support

[←](#) Support

Service Desk

Le Service Desk est un **point de contact unique** pour toutes les demandes de nature informatique à l'UCL. Un *call center* orientera l'utilisateur vers le bon interlocuteur. Il aide tout nouveau membre de l'UCL à résoudre les problèmes liés à son identité : carte d'accès, identifiant global, mot de passe. Il reçoit aussi les demandes liées aux services IT (courrier électronique, agenda, WiFi, portail, etc.)

Le Service Desk est accessible :

- par téléphone, de 8h30 à 17h30 tous les jours ouvrables sans interruption au n° d'appel : 010/47.8282
- via l'adresse mail 8282@UCLouvain.be

SmartCard

Wifi

Répertoire

La page "Support" reprend l'ensemble des informations disponibles sur le portail de l'UCL. Ces informations sont divisées en trois catégories. Les informations à connaître sur le Service Desk et les divers moyens de le contacter. Celles à connaître à propos des SmartCard ou carte d'accès. Et pour finir, les explications pour les différentes démarches à réaliser afin de se connecter avec ses identifiants au réseau Eduo-ram.

[←](#) Support

Service Desk

SmartCard

Wifi

Répertoire

Nom nom

Prénom prénom

RECHERCHER

[←](#) Support

SmartCard

Wifi

Répertoire

Nom deville

Prénom prénom

RECHERCHER

De Villé Philippe

Deville Yves

[←](#)

**Yves Deville**

(PROFESSEUR ORDINAIRE)

SST/EPL – Ecole Polytechnique de Louvain (EPL)

SST/CTM – Institute of Information and Communication Technologies, Electronics and Applied Mathematics (ICTM)

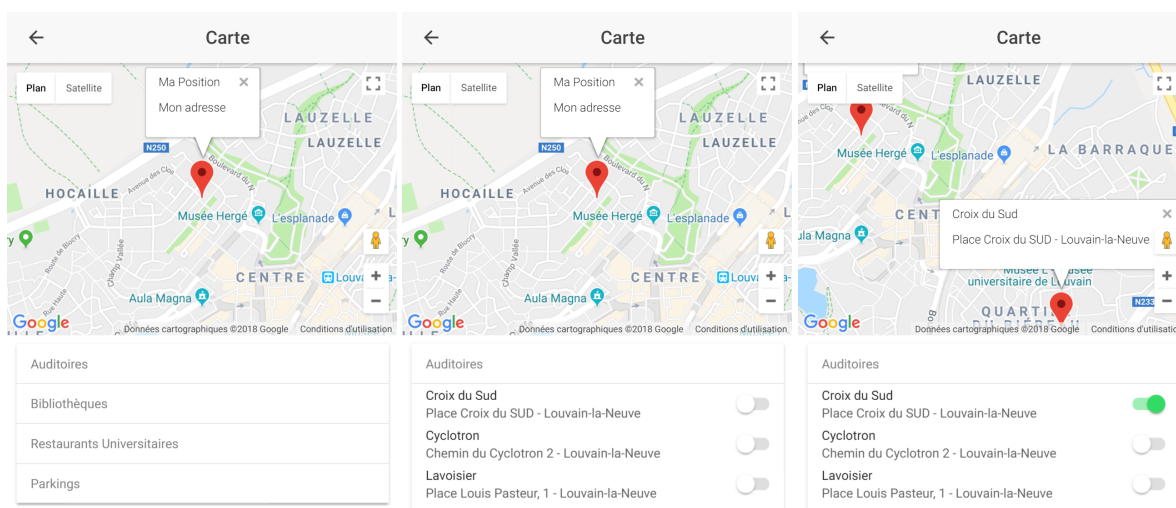
→ SST/CTM/INGI – Pôle en ingénierie informatique (INGI)

**Contact**

Adresse postale INGI  
Place Sainte Barbe  
2/L5.02.01  
1348 Louvain-la-Neuve

Une quatrième partie est également disponible : le répertoire UCL. Celui-ci permet de rechercher un membre du personnel UCL grâce à son nom et/ou son prénom. Pour cela, il suffit d'écrire dans les champs correspondants et ensuite d'appuyer sur "Rechercher". Une fois la recherche effectuée, une liste des résultats va apparaître en dessous du bouton. Lorsqu'on appuie sur l'un des résultats, une autre page s'affiche avec les diverses informations concernant la personne sélectionnée. Ces informations comprennent son nom, son prénom, son statut à l'UCL, l'acronyme et le nom complet des différents organismes UCL auxquels il appartient. Tout ceci est suivi des différentes informations permettant de contacter la personne sélectionnée : l'adresse postale de son bureau, son adresse mail ainsi que ses différents numéros (bureau et secrétariat). Il est possible de retourner sur la page "Support" grâce à la flèche présente en haut à gauche de la page.

## 9 Carte



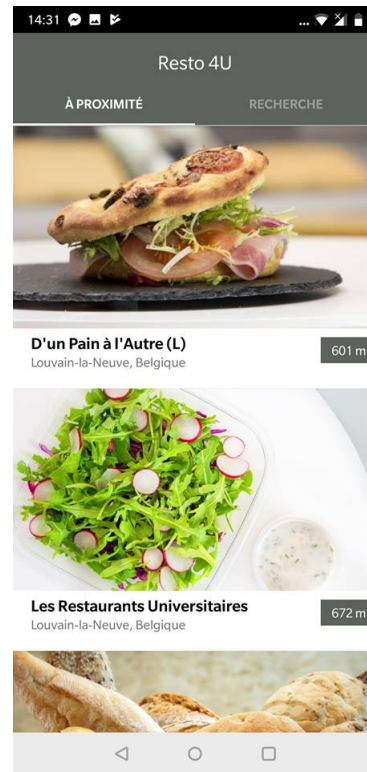
La page "Carte" présente une carte Google Map centrée sur votre localisation si la géolocalisation de l'appareil est activée. Sous cette carte, différentes listes déroulantes sont présentes : les auditoires, les bibliothèques, les restaurants universitaires ainsi que les parkings. Lorsqu'on appuie sur l'une de ces catégories, la liste des points d'intérêts qui la compose se déroule. Ceux-ci sont présentés avec leur nom et leur adresse exacte. Dès que vous cochez l'un des points, celui-ci s'affichera instantanément sur la carte afin de le situer.

## 10 Restaurants Universitaires



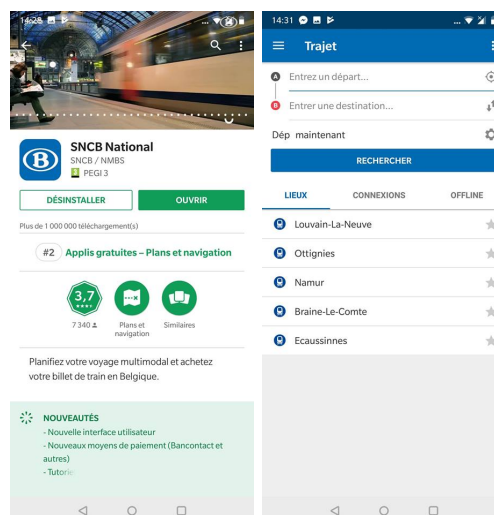
Lorsque, depuis la page d'accueil ou le menu, vous appuyez sur l'icône des restaurants universitaires Resto4U et que vous ne possédez pas l'application sur votre appareil, vous serez automatiquement redirigé vers la page permettant de télécharger l'application sur le store. Afin de profiter de cette fonctionnalité, il est nécessaire d'installer l'application en question.

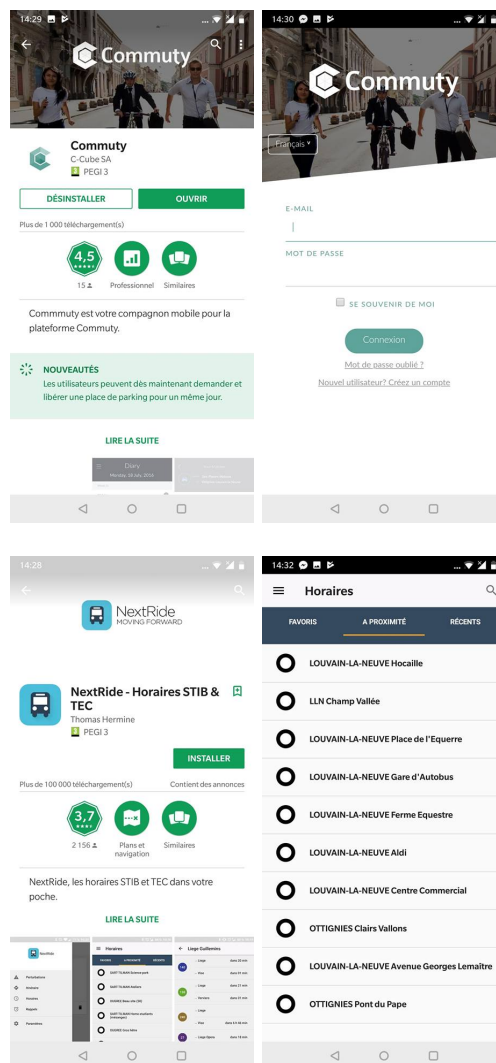
Une fois l'application téléchargée et installée, lorsque vous retournerez sur la page "Restaurants Universitaires", l'application Resto4U se lance automatiquement. Afin de retourner sur UCLCampus, il suffit d'appuyer sur le bouton retour de l'appareil ou de quitter Resto4U. De retour sur l'application, une page présente diverses informations concernant les restaurants universitaires.



## 11 Mobilité

La page "Mobilité" propose trois outils : l'horaire des trains SNCB, l'horaire des différents bus grâce à l'application NextRide ainsi qu'une application permettant d'organiser et de participer aux covoiturages Commuty. Chacun de ces outils fonctionne de la même manière que la page "Restaurants Universitaires". Dans le cas où l'application n'est pas installée sur l'appareil, une redirection vers le store est faite afin de la télécharger. Tandis que si elle est installée, l'application se lance automatiquement.

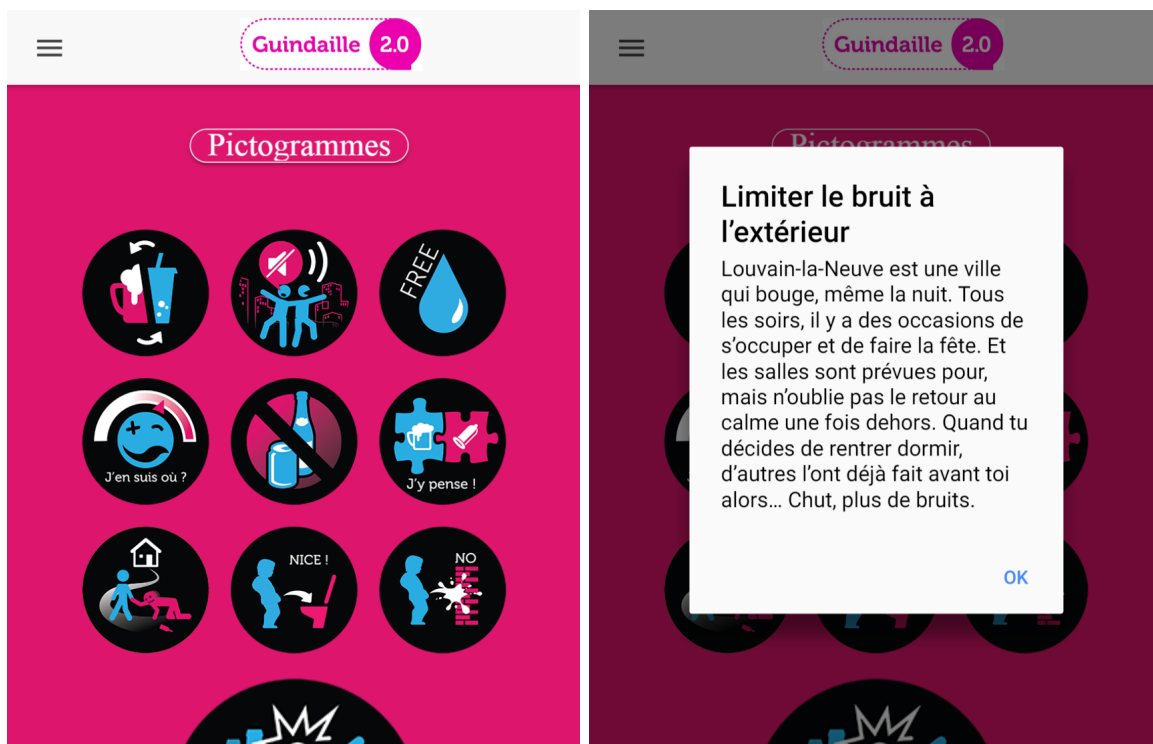




## 12 Guindaille2.0



La page "Guindaille2.0" présente les différents éléments de la campagne Guindaille2.0 organisée par l'ASBL Univers Santé. Elle est divisée en trois parties : les différents pictogrammes de la campagne, l'effectomètre ainsi que les informations en cas d'urgence.

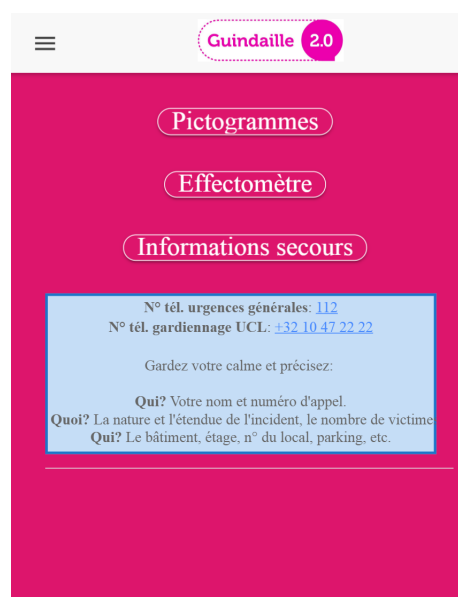


Les différents pictogrammes de la campagne sont affichés une fois que le bouton "Pictogrammes" est appuyé. Il est possible d'appuyer sur chacun d'entre eux afin d'en obtenir plus d'informations.



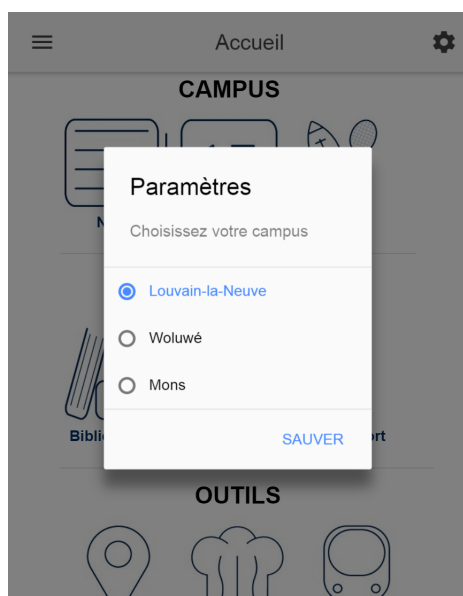
L'effectomètre est une brochure ou un outil visant à sensibiliser et responsabiliser les jeunes à propos de leur consommation d'alcool. Cet outil rappelle également la législation en vigueur en matière de vente d'alcool.

La dernière section présente diverses informations telles que les différents numéros à appeler en cas d'urgence : numéro de téléphone pour les urgences générales ainsi que le numéro de gardiennage UCL. Quelques conseils sont également présentés sous ces numéros.



## 13 Paramètres

Lorsqu'on appuie sur le bouton des paramètres qui se situe en haut à droite de la page d'accueil, deux fenêtres de dialogue vont consécutivement s'ouvrir :



La première permet de sélectionner un campus de l'UCL parmi ceux de Louvain-la-Neuve, Woluwé et Mons. Lors du changement de campus, la liste des sports et les différents points d'intérêts de la carte changeront en fonction du campus sélectionné.

La seconde permet de changer la langue de l'application (anglais ou français). Une fois le choix validé, l'entièreté de l'application sera traduite dans la langue sélectionnée.

