

ReflectOn

Building a Mobile Application That Optimally Supports Employees' Learning and Reflection

Dissertation presented by

Mhd Ayad ALDAYEH

for obtaining the Master's degree in

Computer Science

Supervisor(s)

Kim MENS , Simon BEAUSAERT

Reader(s)

Kim MENS, Simon BEAUSAERT , Charles PECHEUR

Academic year 2015-2016

Abstract

Based on a research project conducted at the Faculty of Psychology and Educational Sciences (conducted by Julien Balasse, Isabel Raemdonck and Simon Beausaert) that looked into the reflection of employees and their learning, the idea of a website that gives an employee the ability to easily record his or her reflections on a learning experience was born. A web application was already built in the context of another master thesis [1]. It provides employees the opportunity to record (with video, audio, or simple notes) their reflections on learning experiences.

The goal of this master thesis is to enhance the existing web application by developing a mobile application extension that supports both iOS and Android mobile phones. This mobile application will give the employees the ability to reflect anyplace, at any time. Also it will give their coaches and supervisors the ability to view their reflections and give a feedback by adding comments on the reflections.

In order to achieve that I built two native mobile applications for each iOS and Android platforms, applying state-of-art technology and software engineering approaches to achieve our goals.

Acknowledgements

I would like to express my gratitude to several people that have been with me during this adventure. First of all, I would like to thank to my supervisor, Kim Mons, for his help and guidance during the whole year.

I would also like to thank to the Psychology's Research Team for their cooperation and help in different stages of this thesis.

Last but not least, I would like to thank to all the people that has supported me and helped me, without them this thesis would not have been possible.

Contents

List of Figures	5
1 Introduction	6
1.1 Context	7
1.2 Problem	7
1.3 Motivation	7
1.4 Objectives	7
1.4.1 Validation	7
1.4.2 Deliverables	8
1.4.3 Learning Objectives	8
1.5 Approach	8
1.6 Contributions	8
1.7 Roadmap	9
2 Background Materials	10
2.1 Psychology research	10
2.2 ReflectOn web application	10
2.3 SWIFT 2 (iOS App)	11
2.4 JAVA (Android App)	12
2.5 Database (Parse)	13
3 Problem Statement	15
4 Analysis	16
4.1 Software Development Approach	16
4.1.1 Scrum Agile method	16
4.1.2 Scrum choice	17
4.1.3 Scrum Advantages	17
4.2 Defining the Scope	18

4.2.1	Functional Requirements	18
4.2.2	Non-Functional Requirements	19
4.3	Analysis of development technologies	19
5	Development	24
5.1	Business Process Models	24
5.2	Mockup	27
5.2.1	Why did we choose Mockup	27
5.2.2	Mockup implementation	27
5.2.3	Mockup validation	29
5.3	Implementation	29
5.3.1	Coding	29
5.3.2	Database implementation	34
5.3.3	Main implementation issues	35
6	Validation	37
6.1	Tests	37
6.2	Internal validation	37
6.3	External validation	38
7	Future development	40
8	Conclusion	42
9	Bibliography	43
10	Annex	45
10.1	questions	45
10.2	Mockups	47
10.3	Mobile Applications Screenshots	49
10.4	Scenarios	51
10.5	Questionnaire	58

List of Figures

5.1	Business Process Model notations	24
5.2	"Login" process	25
5.3	"Adding a new reflection" process	26
5.4	"Coaching" process	26
5.5	Mockups	28
5.6	Model-View-Controller design pattern (MVC) for iOS	30
5.7	Screenshots from the final iOS mobile application	31
5.8	Screenshots from the final Android mobile application	33
5.9	Database Diagram	34
6.1	The average scores of the external evaluation	39

Chapter 1

Introduction

The goal of this thesis was to develop a mobile application that complements the “Reflect On” web application [1], which has been developed last year in the context of another master thesis. The purpose of this application is to create a tool to support an ongoing PhD research at the psychology faculty in UCL. So the application is not directly linked to computer science research, but nevertheless shows the growing importance of computer science, and how can we use it efficiently to help and support other scientific fields.

The psychology research project is about how the reflection of employees plays a core role in their learning process. It is assumed that the better the quality of their reflection, the better their learning results are.

“Reflect On” is a tool that helps employees record their reflections on different real life situations that they might face during their job. They can also use it to add some supportive media (photo, video or audio) to these reflections. These reflections will be monitored by coaches. Coaches are able to give their feedback on their coachees’ reflections.

As the web application had already been conceived, this made it easier to understand the concept and the problem that we are trying to solve. But deciding upon the exact functionalities that we wanted to include in our mobile application was harder, because we tried to make the mobile application as simple as possible for the end users, but at the same time it should include the core functionalities to be useful, as we will see later in section 4.2. Also, linking the mobile application to the web application via its database proved to be a challenge, as we will see in section 5.3.3.

This is a more practical and implementation-oriented thesis, but what we really care about is applying state-of-art technology and software engineering approaches to achieve our goals.

1.1 Context

As we mentioned before, the context of this thesis is to develop a mobile application as an extension to an existing web application. When we say mobile application we want to stress that we targeted the two major mobile platforms (iOS and Android).

1.2 Problem

As “Reflect On” was only a web application, users were not able to use it until they arrived back to their home or office. During that time, however, they might have forgotten some important details about the reflection that they want to write. Moreover, writing the reflection directly after the incident makes it purer and more honest, and allows it to reflect the real feelings of the user had at the time of the incident. So it was important for the doctoral research to provide the users with a tool that could be used anytime anywhere, in order not to forget any tiny details about the situation that they faced.

1.3 Motivation

Mobile technologies are evolving rapidly, and nowadays most people are depending on their mobile phones more than on their PCs. So the solution for the problem described above is to develop a mobile application that is available on the most common platforms, which provides the ability to use “ReflectOn” anytime anywhere even when they are offline.

1.4 Objectives

As this thesis is practical and implementation oriented, we can divide the objectives into three main categories.

1.4.1 Validation

The final product must meet the requirements specified by the co-promoter (psychology research team). All functional requirements should be validated on real scenarios. Moreover, we should validate the non functional requirements as well (easy of use, performance, ...).

1.4.2 Deliverables

The main two deliverables are the iOS mobile application and the Android mobile application, with the code documentation and the database documentation.

1.4.3 Learning Objectives

As I did not work on developing mobile applications before, this thesis was also a personal challenge for myself. I should choose the right technologies and frameworks for the implementation of this kind of applications, taking into consideration the time limit and the requirements. After that, I should learn and master those technologies in order to deliver an acceptable final product. Another important learning objective was to follow correctly the software engineering approaches.

1.5 Approach

In order to achieve the objectives of this master thesis I needed to decide what software engineering approach to follow. After reviewing the initial requirements, I decided to follow a “Scrum Agile Development Model”[2], because the project is not very complicated, there was only one developer (myself) and there is not a huge number of stakeholders. Also because we had a limited time frame that does not allow us to follow other approaches. Moreover, I had a weekly meeting with my supervisor which could be considered as a sprint meeting. Following this approach will allow us to implement the project incrementally starting with the core and the most important functionalities.

1.6 Contributions

Even though this thesis is a more practical and implementation-oriented thesis, it shows an overview about how to deal with real software projects, and what kind of obstacles and issues we might use during the development and how to deal with them in order to keep moving with development to achieve the most of the project goal. Moreover, it shows how can we measure and analyze the existing technologies and tools available to choose the most suitable one to use considering our application characteristics and our requirements and constraints.

1.7 Roadmap

The remainder of this document is structured as follows: the next chapter provides the necessary background material and report on related work. Then, the proposed solution of the problem is explained. The software architecture supporting the implementation of that solution is then explained. A validation and a case study is then conducted in order to validate the solution. Finally, a conclusion delivers the main contributions of this research and we present some avenues of future work.

Chapter 2

Background Materials

In this chapter, we will introduce the background material needed to understand the core chapters of this thesis. This includes all relevant material to make the thesis “self- contained” so that it can be understood without having to consult other sources. We will highlight the following topics: the psychology research project which study the relation between the employees’ reflections and their learning process, ReflectOn web application [1] which has been developed last year, SWIFT2 [3] which is the iOS applications’ programming language, JAVA [4] which is the Android applications’ programming language and finally Parse [5] which is a cloud backend we will use instead of developing a new backend from scratch.

2.1 Psychology research

A research project conducted at the Faculty of Psychology and Educational Sciences (conducted by Julien Balasse, Isabel Raemdonck and Simon Beausaert) that looked into the reflection of employees and their learning. From previous research it has been shown that reflection of employees plays a core role in their learning process. It is assumed that the better the quality of their reflection, the better their learning results are. However, it is not always easy to support employees’ reflection in an optimal way. To make employees reflect well, different tools are used. One way to support reflection is by using etools. Because of fastevolving technology, nowadays these tools can support employees’ reflection even better.

2.2 ReflectOn web application

It was developed last year by “Richard Bastien”. It includes many functionalities that were not in the scope of this master thesis, because the functionality of the mobile application was deliberately more limited in order not to overload the end-user with too many unneeded func-

tionalties.

The web application was developed in PHP [6] and MySQL database [7] using the Symfony2 web application framework [8]. Symfony 2 is a great framework that provides many advanced features like security and an ORM (Object Relational Mapping) [9]. An ORM, is a piece of software that makes a relationship between a relational database and the entities of an application. The link is automatically managed and allows the application to abstract over the database in order to correspond and deal directly with objects instead. This also helps to simplify the management of basic operations such as persistence, deleting, or searching for items that are hidden behind the abstract ORM. Moreover, it creates a less strong dependency between the application and the database, so the application does not communicate directly with the database, but through the ORM. But also ORM has some disadvantages, ORM suffers from an “impedance mismatching” between the object oriented model [10] and the relation model, so the object oriented model may get a relational feeling and not really profiting from object oriented characteristics such as inheritance. Or this could make the ORM mapping really complicated which leads to split the application design into two parts: the data access objects (DAO) that are mapped one on one to the database tables via the ORM, and the real objects which are following the traditional object oriented model. This solution complicates the design because we will have two types of objects that we need to connect to each other. But these kinds of problems typically start to arise only when the application scales up to a certain size (in terms of functionality and number of lines of code) but for more simple applications, simple ORM mappings often suffice.

2.3 SWIFT 2 (iOS App)

Swift [3] is a new powerful general- purpose programming language; it is managed by a collection of open source projects. The code written in Swift is safe, fast and expressive. It can be used to target all Apple platforms, also it could be used side by side with Objective C [11] or it can be used as a total replacement of Objective C. Binary builds of open-source Swift integrate with the Xcode developer tools [12], which is the main development platform on Mac to develop Apple-targeted platforms applications (iPhone, iPad, iMac, ...). Also it includes complete support for the Xcode build system, code completion in the editor, and integrated debugging. The structure of Swift code is very easy to learn, write, read and understand, because it is a C-like, object-oriented programming language. But also it includes many features that makes the code safer and smaller, such as a powerful control flow, extended pattern matching to work in if-clauses and for-loops and it is not exposing pointers and unsafe assessors.

A small representative code fragment in Swift is shown below which represent the "Login" function:


```

@IBAction func loginBtnAction(sender: AnyObject) {
    // Check the Internet connection
    if GlobalVariables.isOffline(self, isFranch: languageSwich.on){
        return
    }

    if (usernameTxtField.text == "" || passwordTxtField.text == ""){
        let alert = UIAlertController(title: alertTitle, message: alertEnterMessage, preferredStyle:
            UIAlertControllerStyle.Alert)
        alert.addAction(UIAlertAction(title: alertActionTitle, style: UIAlertActionStyle.Default,
            handler: nil))
        self.presentViewController(alert, animated: true, completion: nil)
        return
    }
    PFUser.logInWithUsernameInBackground(usernameTxtField.text!, password:passwordTxtField.text!) {
        (user: PFUser?, error: NSError?) -> Void in
        if user != nil {
            // Do stuff after successful login.
            GlobalVariables.currentUser = PFUser.currentUser()!
            self.getUserLanguage()
            self.performSegueWithIdentifier("loginSegue", sender: nil)
        } else {
            // The login failed. Check error to see why.
            let alert = UIAlertController(title: self.alertTitle, message: self.alertcheckMessage,
                preferredStyle: UIAlertControllerStyle.Alert)
            alert.addAction(UIAlertAction(title: self.alertActionTitle, style:
                UIAlertActionStyle.Default, handler: nil))
            self.presentViewController(alert, animated: true, completion: nil)
            return
        }
    }
}

```

2.4 JAVA (Android App)

The most recommended way for developers to write native Android applications is by using JAVA [4] and the Android SDK [13]. JAVA is a general purpose, object-oriented programming language, and it is one of the most popular programming languages in use. Moreover, JAVA has been designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

JAVA code syntax is akin to C syntax but it has fewer low-level facilities (such as pointer manipulation), which in fact make it really easy to write, read and understand the code.

Android development relies on these Java fundamentals. The Android SDK includes many standard Java libraries as well as specialized Android libraries that help in developing Android applications.

In order to develop our Android application, we used Android Studio [14]. It is the official Integrated Development Environment (IDE) advised for Android application development, it has a powerful code editor and developer tools such as debugging and performance tools.

A small representative code fragment in JAVA is shown below which also represent the "Login" function:

```

// Login Button Click Listener
loginbutton.setOnClickListener(new OnClickListener() {

    public void onClick(View arg0) {
        // Retrieve the text entered from the EditText
        usernametxt = username.getText().toString();
        passwordtxt = password.getText().toString();

        //Check if the username or the password is null
        if (usernametxt == null || passwordtxt == null )
        {
            Toast.makeText(
                getApplicationContext(),
                "Oops, sorry, please check your username and password",
                Toast.LENGTH_LONG).show();
        }

        else {
            // Send data to Parse.com for verification
            ParseUser.logInInBackground(usernametxt, passwordtxt,
                new LogInCallback() {
                    public void done(ParseUser user, ParseException e) {
                        if (user != null) {
                            // If user exist and authenticated, send user to Reflections.class
                            Intent intent = new Intent(
                                LoginSignupActivity.this,
                                Reflections.class);
                            startActivity(intent);
                            finish();
                        } else {
                            Toast.makeText(
                                getApplicationContext(),
                                "Oops, sorry, please check your username and password",
                                Toast.LENGTH_LONG).show();
                        }
                    }
                });
        }
    }
});
}
}

```

2.5 Database (Parse)

Parse [5] is an open source mobile cloud backend offered as a service owned by Facebook. It gives mobile application developers a way to link their applications to backend cloud storage and offers APIs exposed by back end applications while also providing features such as user management, push notifications, and integration with social networking services. These services are provided via the use of custom software development kits (SDKs) and application programming interfaces (APIs).

We decided to use Parse, instead of developing a complete backend for our applications ourselves, because Parse already provides a strong and reliable services, and also due to our own time constraints.

One of the great advantages that we gain by using Parse is the ability to query the data using NoSQL queries [15], which are easier to write and understand than regular SQL queries if our application and our database is simple (which is true in our project) but for more complex

applications and databases, SQL is a better choice.

Unfortunately, after starting using Parse, Facebook decided to shut it down so it become obsolete on January 2017, but we can still migrate the database to another cloud server or we can create our own server and migrate our full backend into it. For that reason, Parse provides automated migration tools to help the users in the migration process.

Chapter 3

Problem Statement

The main goal of this master thesis is to develop a mobile application to enhance the existing ReflectOn web application. Which is an e-tool to support employees' reflection by giving them the ability to easily record their reflections. This mobile application will give the employees the opportunity to reflect anywhere, at any time, whenever they think a reflection might be valuable. Also it provides employees the opportunity to add supportive attachments (photo, video, or audio) to their reflections.

In order to have a reflection of an even higher quality, the mobile application can provide the employees with additional questions, depending on what the reflection is about. At the same time, the mobile application can be regarded as a folder that keeps track of the employees' previous reflections and which can easily be discussed with a coach or supervisor afterwards, by giving them the ability to add comments on each reflection to help the employees go even one step further in their reflections.

Chapter 4

Analysis

In this chapter we will introduce the analysis procedures that had been done at the beginning of the project. What software development approach did we choose to develop our application, defining the scope of it and analyzing the development technologies that are exist in order to choose the best and suitable tools to use for the development.

4.1 Software Development Approach

In order to achieve the objectives of this master thesis, I decided to follow the “Scrum Agile Development Model” [2], where my supervisor and I define the tasks for each sprint and alter the basic application in order to have these tasks done incrementally.

4.1.1 Scrum Agile method

Scrum is an agile development model which is used for completing software development projects. Agile is a response to the failure of the traditional waterfall model and of the dominant software development project management paradigms. An agile Scrum process has the following benefits on software development projects:

- Increase the quality of the deliverables
- Cope better with requirement changes, so for example if an optional functionality is blocking the development of the software, the decision of skipping it and keep following the plan will be easier.
- Need less time to deliver the final product, because we do not waste time in exhaustive planning, modeling and documentation.
- Provide a better control of the project planning and state.

4.1.2 Scrum choice

I chose to work with Scrum since this Agile methodology fits more the nature of this thesis in terms of constraints, such as time, and supervision. I had a limited time frame to develop two mobile applications from scratch. Also I had a weekly meeting with my supervisor in addition to three meetings with my co-promoter who represents my customer (psychology research team). Scrum is therefore the most adapted agile method to my project.

Scrum methodology defines a number of roles in the team, expanded to the customer.

- The "Product Owner" which is the "owner" of the product, as a client representative he has the right to remove or add requirements. In this project my co-promoter is the product owner. He was involved during the planning meetings and validation.
- The "Scrum Master" is the manager of the project, responsible for assisting the team to apply the rules and requirements of Scrum, also he helps in resolving problems. In this project my supervisor is the scrum master.
- The team is all the trades necessary to implement a functionality, which in this project is composed of one person, myself.
- Sprint Planning or iteration: before each sprint, a meeting is organized with my supervisor to discuss the requirements, the amount of work achieved so far, and to go through and test the developed functionalities. In this project each sprint iteration is one or two weeks depending on the meetings schedule. During the weekly meetings with my supervisor, we discussed the functionalities and concentrate on the main ones, so that the planning of the project is not disturbed by optional functionalities.
- Daily scrum: a daily scrum with myself, which takes up to 15 minutes to respond to three questions: "What concretely has been done since the last scrum? What will be completed by the next scrum? And what obstacles could I face? "

4.1.3 Scrum Advantages

Scrum is different from other development methodologies; we can get many benefits by following it, such as:

- Maximize the cooperation: the weekly meeting with the supervisor and exchanging the solution ideas gives us more corporation and a mutual agreement on what to do next.
- Dealing faster with changes: Scrum sprints allows us to deal and respond faster and more efficiently with requirements change or obstacles. For example, during the development of this project I faced some critical problems such as the database APIs and the offline mode.

During the weekly meeting with my supervisor (sprint iteration meeting) we discussed those issues and agreed to solutions that do not block the development process of the project and its core functionality.

- Increased productivity: by eliminating certain "constraints" in conventional methods such as documentation, Scrum allows to increase the productivity in the project.

4.2 Defining the Scope

Our mobile application is a complement of the existing “ReflectOn” web application. So we took the functionalities of the web application as a starting point to define our scope, taking into consideration the psychology research team’s requirements. They wanted the application to be as simple as possible, and they did not want to include all functionalities that are exist in the web application. After discussing the functionalities with them we agreed on the following requirements:

4.2.1 Functional Requirements

User Side

Anyone who is registered and has access to our system is considered as “User”.

- The user can choose either French or English as his preferred language.
- The user can upload and update his profile picture.
- The user can update his profile information (email address and mobile number).
- The user can view his list of reflections.
- The user can add new reflection. This includes the following:
 - The user should choose the reflection type from the dynamic type list.
 - The user should answer some questions according to the type he chooses (the questions are dynamically fetched from the database according to the type) the full list of questions can be found in appendix 10.1.
 - The user can save his answers.
 - The user can attach supportive media (photo, video or audio) to his reflection.
 - The user should validate his reflection after answering all questions.
 - After validation, the user should not be able to modify his reflection nor modify the supportive attachments list.

- The user can communicate and discuss each reflection with his coach by writing comments on each reflection.

Coach Side

Any user who has a coach role is considered as “Coach”. Each user has only one coach, and each coach is responsible for one or more users. The coach can do every thing the user could do, in addition to the following:

- The coach can view a list of the users those he is responsible for, along with their photo, last reflection and last comment.
- The coach can view a list of the validated reflections for each user he is responsible for.
- The coach can view a reflection from the previous list along with its attachments.
- The coach can communicate and discuss each reflection with the user by writing comments on each reflection.

4.2.2 Non-Functional Requirements

- The mobile application should be bilingual (French and English).
- The mobile application should use and interact with the web application database.
- The mobile application should be used in either online mode or offline mode.
- The mobile application should be user friendly.
- The mobile application should be easy to use without any previous training.
- The mobile application should be fast; it should load each screen within 4 seconds (considering that the user has a good internet connection).
- The iOS mobile application should work on iPhones with iOS version 8.0 and higher.
- The Android mobile application should work on Android version 4.3 (Jelly Bean) and higher.

4.3 Analysis of development technologies

The mobile applications industry has rapidly grown during the past few years, which is why we now have many development tools and frameworks that target the many available mobile platforms (iOS, Android, Windows Phone, Blackberry ... etc.). It is thus very important to

choose the right tool in order to develop our application, taking into consideration the unique project characteristics and constraints.

In this chapter we will compare some potential tools and frameworks in order to choose the one we decided to use to develop our project, and justify our choice.

Given that the most common platforms used today are iOS and Android, it was important for our project to run on both platforms. In order to achieve this, we had to choose between “native development” and “cross-platform development”.

Native development: is to develop the mobile application using its native platform tools and framework (SWIFT for iOS and JAVA for Android). This implies that if we are targeting two platforms, we have to develop two different applications from scratch.

Cross-platform development [16]: is to develop the application using a cross platform environment from which native mobile applications for many platforms can be generated. At first sight this approach seems to be more promising and less time and effort consuming, but the problem is that it is not possible to have 100% shared code for all mobile platforms. Although the platform will generate the core logic of the application, we still have to do more work using each platform’s native language, in order to get our final applications. Because the automatically generated code might not support all features due to the fact that each native platform has its own APIs and restrictions, we also might lose the flexibility of each platform because the cross-platform solution uses only the features common among all native platforms. Also, we should not forget that the UI components (widgets, transitions, user interaction, ...) are different between each platform, so even if we get a 100% working backend we have to develop the user interfaces separately for each targeted native platform.

Our project is unique because we have many constraints:

- **Time:** the two applications should be created within the limited timeframe of this master’s thesis, so it is very important to choose the right tool to get our applications ready in time.
- **Cost of entry:** in order to achieve the project goals, we have to learn and master the chosen technologies (remember that this was our first experience with building mobile applications), so we have to consider the time needed for this as well in order to be ready to start developing our project using those technologies.
- **Testing and debugging:** choosing between native development and cross-platform development may also have an effect on the testing and debugging overhead. So either we will debug one common code if we use cross-platform development or we will debug two different codes if we used native development.
- **Future development:** we also need to take into consideration possible future development plans of this project. Do we have to deal with one code (cross platform) or several

codes (native)? How hard will it be to find a team that knows the used tools and frameworks? How hard is it to understand the existing code? How hard to modify and add new functionalities to the existing code? How hard to test and debug the new applications?

Taking into account these constraints we will compare several cross-platform frameworks with native frameworks according to the following criteria with their relative weights of importance:

Code	Criteria	Weight	Note
C1	Development language	High	Using more common languages will development easier and faster
C2	Development time	High	We need to be able to complete the two applications in the time available
C3	Performance (user experience)	High	We have to get the best user experience possible
C4	Shared code	High	More share code means less development time
C5	Access to native APIs	Medium	To achieve the projects goals
C6	Cost of entry	Medium	Will affect the development time and the future development
C7	Native look and feel (user interface)	Medium	We have to get the best user experience
C8	Platform updates	Medium	Will the current app remain compatible with new releases of targeted platform?
C9	Future development	Medium	The cost of future enhancements and the probability to find someone able to do it
C10	Testing	Low	In any case we have to test the application separately on each platform
C11	Security	Low	Low relevance for our particular application because we are not really dealing with critical of highly sensitive data

Code	Cross-Platform Development			Native Development
	Xamarin	PhoneGap	Titanium	
C1	C# for the backend Native interfaces	JavaScript, HTML5 and CSS	JavaScript backend Native interfaces	Swift for iOS JAVA for Android
C2	High	Moderate	High	High
C3	Might be slow and have some delays because of the shared middle layer.			Best performance
C4	60% to 100%	100%	60% to 100%	No shared code
C5	Access to limited APIs and not supporting new features			Full access
C6	Moderate	Moderate	High	High
C7	High	Limited	Limited to high	High
C8	May require updates to be compatible with the new platform, this require also to wait until getting the updated framework			Mostly compatible
C9	Need to know how to use the frameworks APIs and the sometimes the native languages			Required to know only the native language
C10	Needs more testing and debugging for the shared code and for the individual generated native apps			Testing and debugging both apps
C11	Less secure			More secure

According to the previous comparison and according to our project's requirements, in the end we decided to develop the mobile applications using native development for the following reasons [17]:

- Using a cross-platform development framework will not guarantee 100% shared code, so in any case in both cases we have to deal with native code either to develop the UI or to test and debug the generated native applications.
- The cross-platform development platform generates less performant and less user-friendly final mobile applications.
- Native development will give us access to all native APIs, which will make it easier to deal with the device recourses such as the memory, the camera, the microphone ... etc. and will give a more native look-and-feel to the application.
- Native development will give us the ability to get the updated tools as soon as a new platform update is releases, where in cross-platform development we often have to wait longer time to get the framework update.
- The generated native applications from the shared code might have compatibility issues with the new releases of the targeted platforms. Applications that have been developed natively, however, will be more likely compatible with new releases of the targeted platforms.

- Considering future development, finding a team who knows how to develop native applications is easier than finding team who knows how to use the cross platform framework's APIs. Because in the real world developing native applications is more common in companies than developing cross-platform applications, so the developers are trying to master one platform instead of struggling with cross-platform development environments and their limitations.
- Even though it will take more time to learn and develop two applications with two different native languages, we will get much more advantages and it will make it easier to continue developing the project in the future, and it will give the end-users better experience.

Chapter 5

Development

5.1 Business Process Models

Business Process Model (BPM) [18] in systems engineering is the diagram that represent the processes of an enterprise (in our case the software). We founded that this kind of model can describe easily the main processes of our mobile application. The main notations of BPM are shown in figure 5.1 :

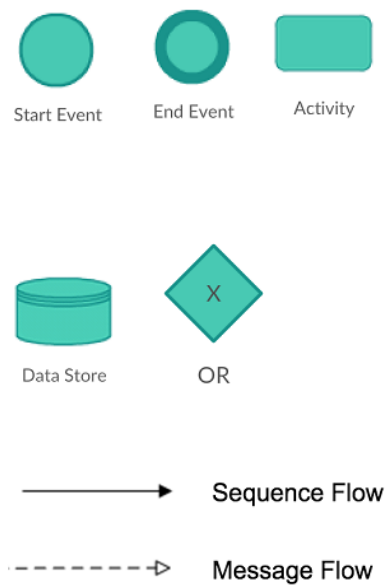


Figure 5.1: Business Process Model notations

ReflectOn mobile application has three main processes, figure 5.2 shows the "login" process, figure 5.3 shows "adding a new reflection" process and figure 5.4 shows "coaching" process.

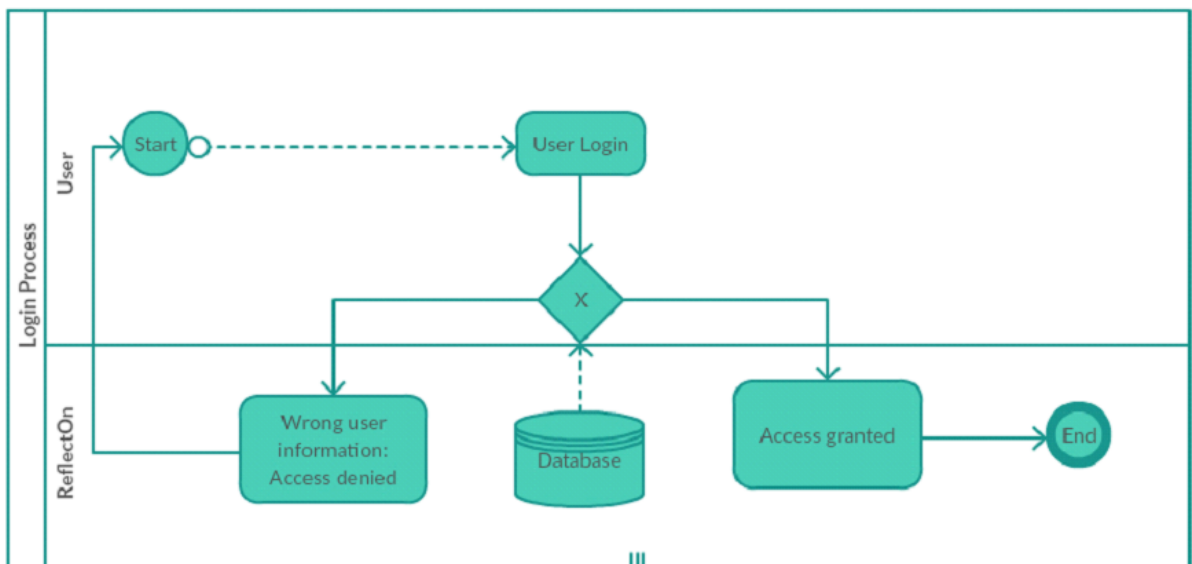


Figure 5.2: "Login" process

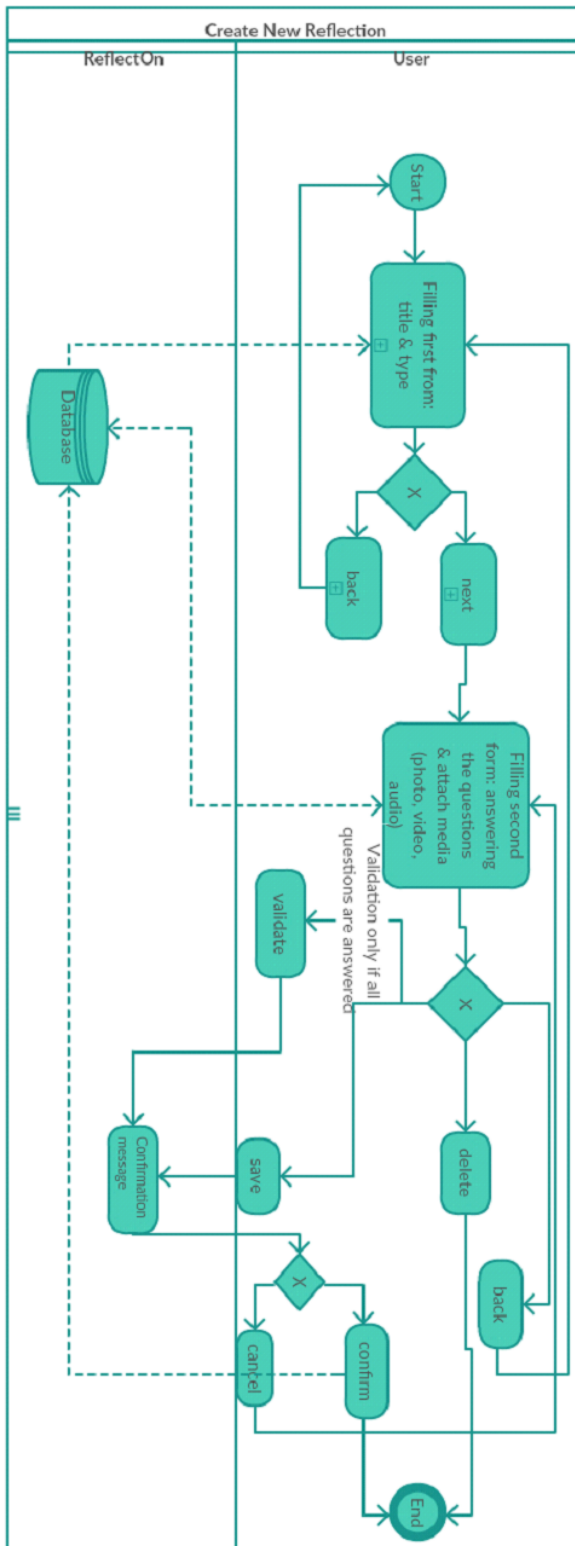


Figure 5.3: "Adding a new reflection" process

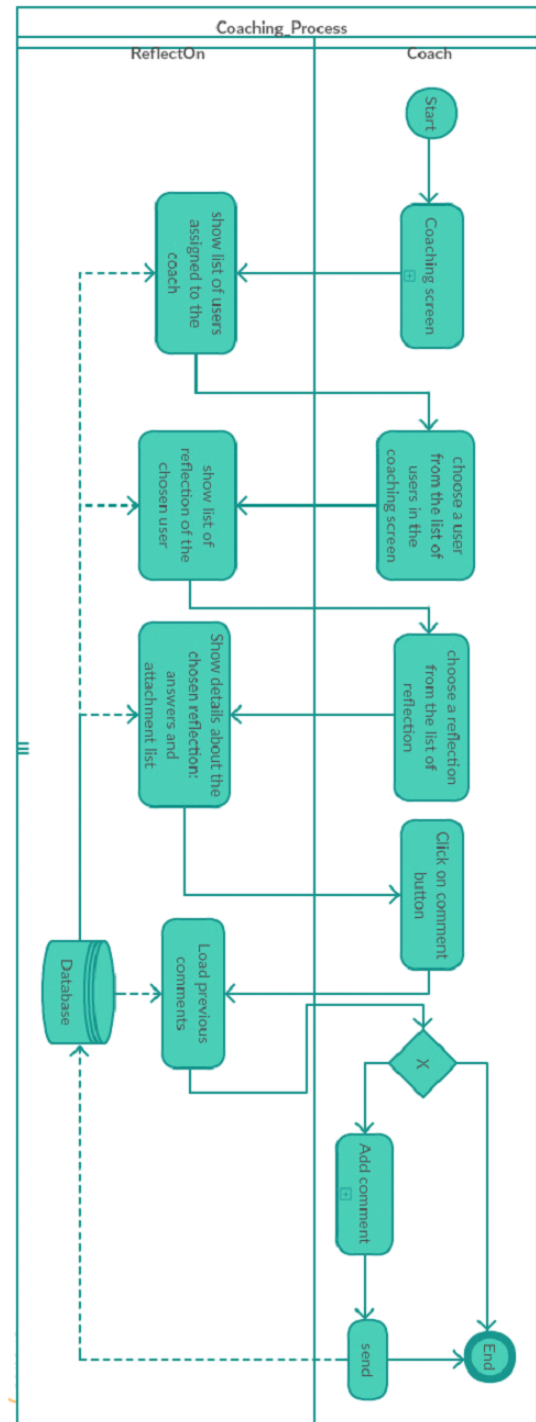


Figure 5.4: "Coaching" process

5.2 Mockup

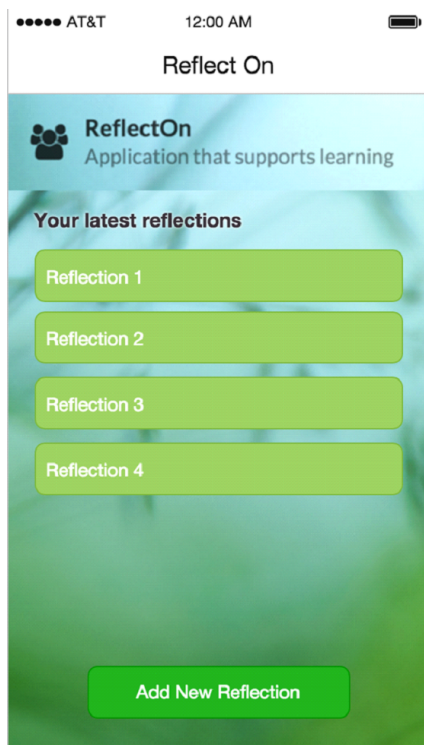
Mockup [19] is a scaled or a full size model of the system used for demonstration and evaluation. In software engineering we create mockups to design the interfaces and show the users how the application will look like without having to build the software or the underlying functionality.

5.2.1 Why did we choose Mockup

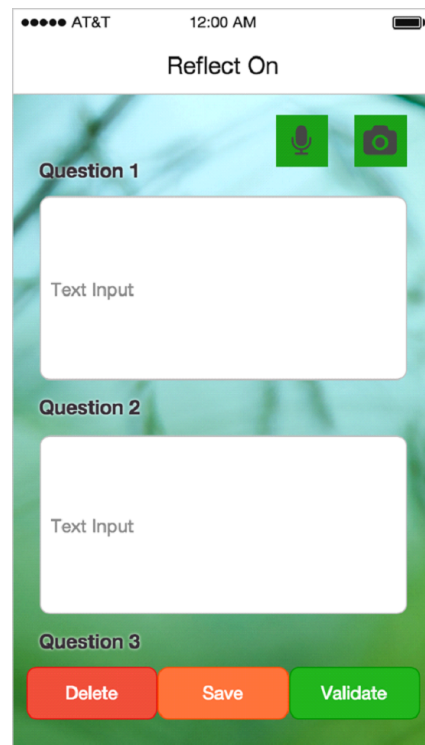
We decided to do mockups instead of wireframes because mockups look more real and realistic, and they help us to think behind the functionality. Also it will be easier for the users to understand and judge a mockup because it looks very close to the final product. By using a good mockup tool we can create our mockups in a short time and we can have an idea about what components does the mobile device supports and what it does not.

5.2.2 Mockup implementation

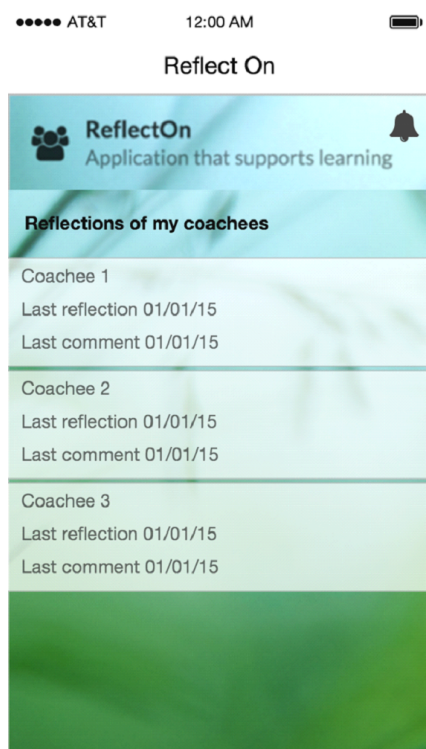
To implement our mockups, we used “Mockingbot.com” [20], which is a free online tool to create mockups. It is easy to use and it has the mobile interface components ready to use, so it can represent how the application will look like after development. Moreover, we did not have to design any component our selves. Also it provides a navigation functionality between the implemented mockup screens, which allows us to have a mixture between mockup and prototype to demonstrate it to the co-promoter and to make sure that we covered all functionalities needed. To use the tool, you have to specify the targeted platform, in our case we decided to do the mockups just for the iOS platform because the application on Android platform will almost be the same as the iOS application. Then you have to choose the background and the color theme. After that you have to add the components you need to use from the platform components list. At the end you can define a transition behavior for each component, this will allow you to navigate through the mockup screens and will give you the feeling that you are using a real prototype. Figure 5.5 shows some of the mockups that have been done. The remaining mockups can be found in appendix 10.2



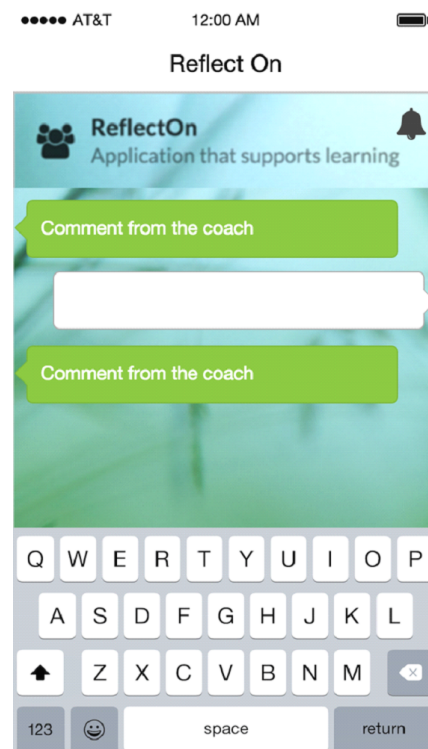
(a) List of reflection screen



(b) Adding a new reflection screen



(c) Coaching



(d) Comments screen

Figure 5.5: Mockups

5.2.3 Mockup validation

In order to make sure that I understood all requirements gathered from my promoters, I had to validate the mockups with them, showing them how the application will look like after development and what functionalities it will include. First, I did the validation with my supervisor. I got few comments such as show non-validated reflections in different color than the validated ones within the reflection list, adding a red point on the notification center button to show that the user has new reflection, and show the user photo within the list of users in the coaching screen. My promoter accepted the mockups and gave me the permission to move forward to the implementation phase. Second, I did the validation again with my co-promoter and his research team. He was happy with the mockups and accepted the design and the functionalities which will be provided by the application.

5.3 Implementation

5.3.1 Coding

In order to efficiently manage and keep track of my code I decided to use Bitbucket [21] as an online source management tool. Bitbucket is a very powerful tool to organize and keep track of my code versions. While I was coding, I tried to keep my code as clean and representative as possible, I also added the necessary comments to make the code clear and understandable for the future developers.

IOS Application

I used Swift2 as a programming language to implement the iOS application using Xcode developer tool which is the official Apple development platform. Writing Swift code is not hard because its syntax is akin to C syntax and it is an object oriented programming language. The hard task was to understand the components and deal with the limitations of Apple devices. Because for Apple devices not everything is possible and they have many restrictions in interacting with the device hardware. That is why I decided to start with implementing the iOS version to have the time to find a solution for the problems that I might face. Developing the iOS application took longer time than what I have planned. But at the end I managed to complete the implementation and start with implementing of the Android app. iOS application development follows the Model-View-Controller design pattern (MVC) [22] figure 5.6, which classifies objects according to the general roles they play in an application. (MVC) design pattern assigns objects in an application one of three roles: model, view, or controller. The pattern defines not only the roles objects play in the application, it defines the way objects communicate with each other. Each of the three types of objects is separated from the others by abstract boundaries and communicates with

objects of the other types across those boundaries.

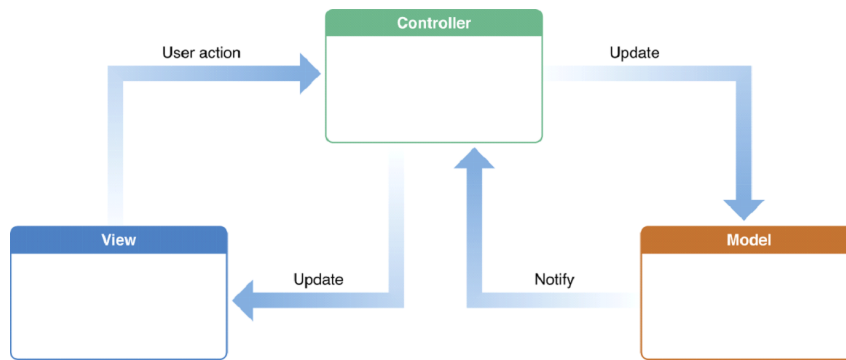


Figure 5.6: Model-View-Controller design pattern (MVC) for iOS

- Model objects: encapsulate the data and define the logic and computation that manipulate and process that data.
- View objects: present information to the user (what the user can see). A major purpose of view objects is to display data from the application’s model objects and to enable the editing of that data.
- Controller objects: act as an intermediary between the application’s view objects and its model objects. Controller objects can also perform setup and coordinating tasks for an application and manage the life cycles of other objects.

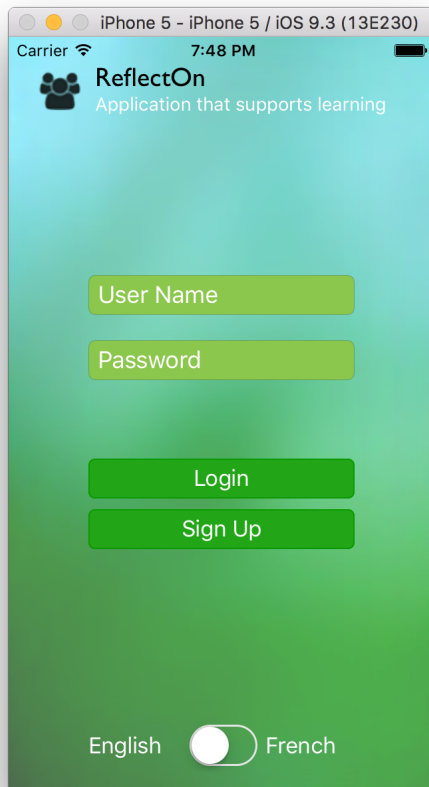
In order to complete the implementation with the expected results, I used some external libraries, because some functionalities were not existing as native components, such as the drop down menu.

I wanted to include a drop down menu to list the reflection types, but there is no such a component supported by Apple. So I used a library called “DropDownCell” [23] to have a drop down menu in my application.

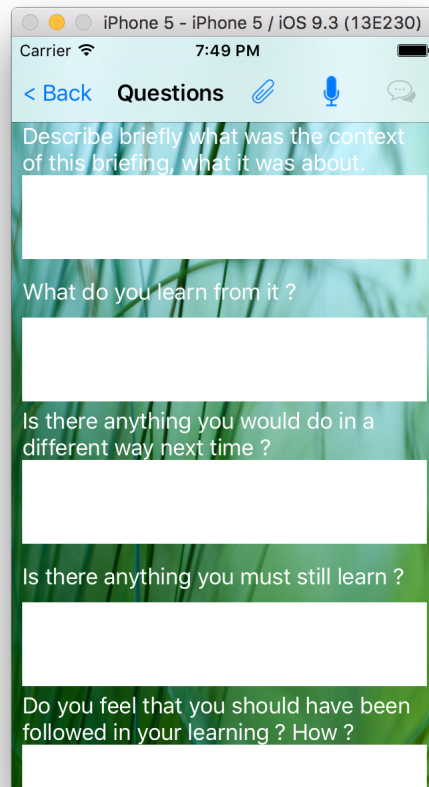
Moreover, Apple supports only images as an object, but it does not support voice memos nor videos. So in order to be able to attach voice memos I had to include a library called “VoiceMemos”[24].

During the tests, we faced some strange behavior with the native keyboard, there is no way to make it disappear after we write a text, so we decided to use another keyboard from “IQKeyboardManagerSwift” library [25]. This keyboard allows us to navigate through the screens cells and hide the keyboard when it is not needed.

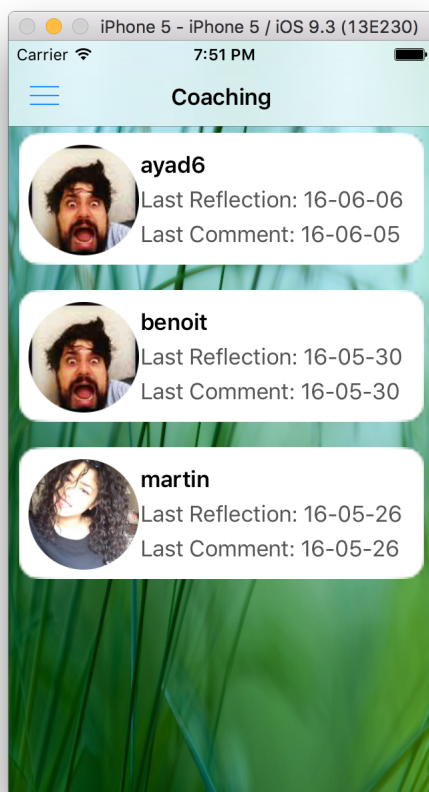
The figure 5.7 bellow shows some screenshots of the final iOS mobile application. The complete screenshots of the final application can be found in appendix 10.3.



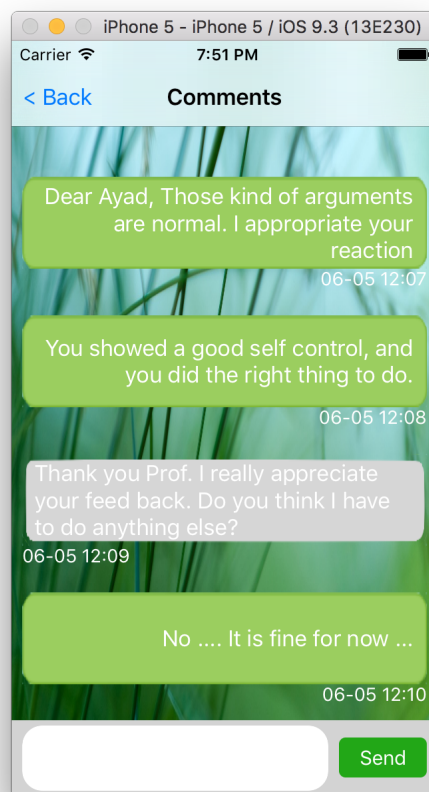
(a) Login screen



(b) Adding a new reflection screen



(c) Coaching screen



(d) Comments screen

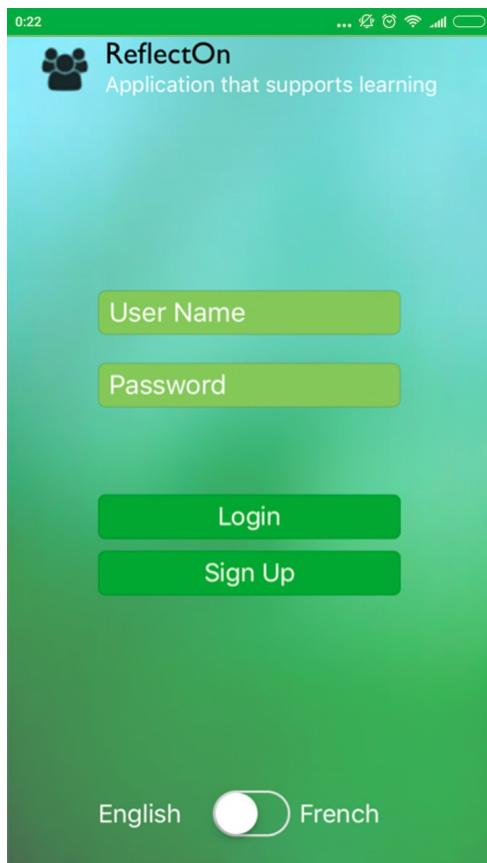
Android Application

Developing Android applications using Java [4] and Android Studio [14] follow different approach than iOS where it depends on activities and extending the existing classes in Android SDK [13]. It also offers many advantages through separating between the design of the user interfaces, which can be defined as XML, and the core business components (like what we saw in iOS MVC). Where each component does a specific task, for example, having activities responsible for presenting the data to the user and services responsible for processing these data behind. This concept gave an extra benefit into isolating the presentation layer of the application from the business layer.

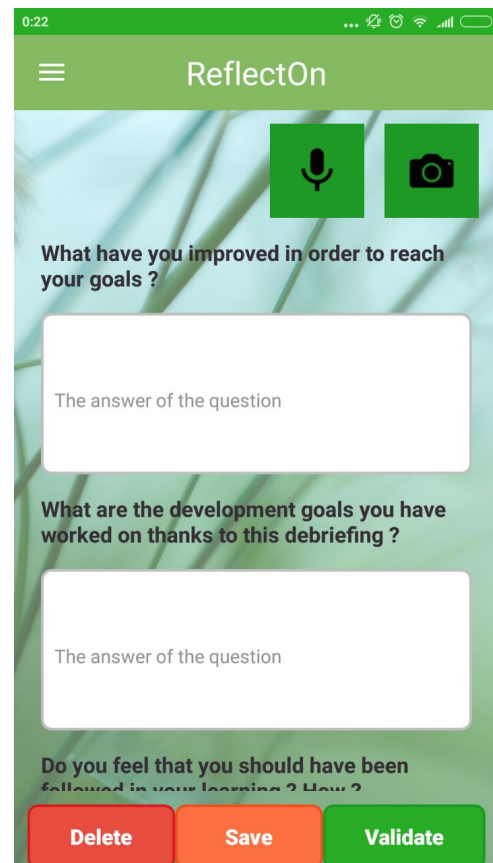
On the contrary of iOS development, I could overcome the problems I faced during the development, thanks to the extensibility features in the framework (which is open source), where I could dig into the source of some bugs and exceptions and adapt them according to the functionality in need to perform. This feature might be an advantage and disadvantage at the same time, for example, if I wanted to use an easy-to-configure grid I could find many implementations but each one suffers from the lack of efficiency or slowness at rendering, so reusing existing code was not always feasible.

But still there is some known problems with Android were impossible to overcome, mentioning for example, having various screen sizes take a major effort in resizing the interface and all of its components so it can fit properly on different screen sizes. Besides that, the simulator of Android takes major time to lunch and run which consumed major time in testing and fixing bugs progressively.

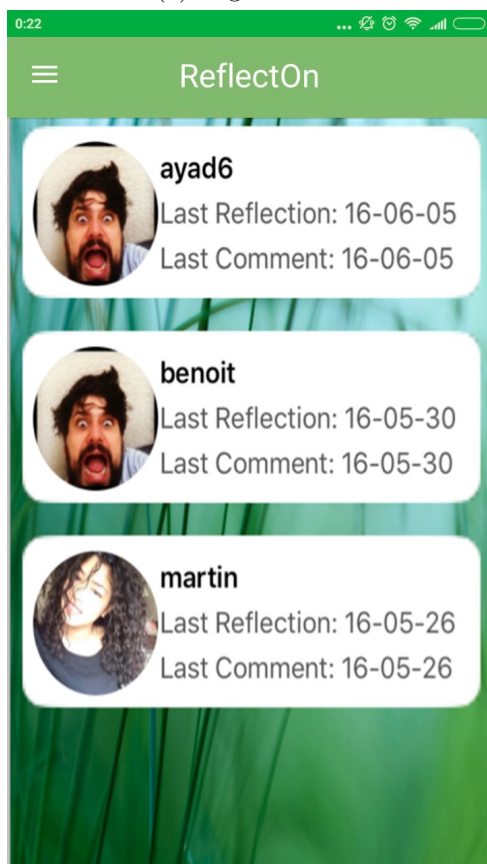
The figure 5.8 bellow shows some screenshots of the final Anroid mobile application.



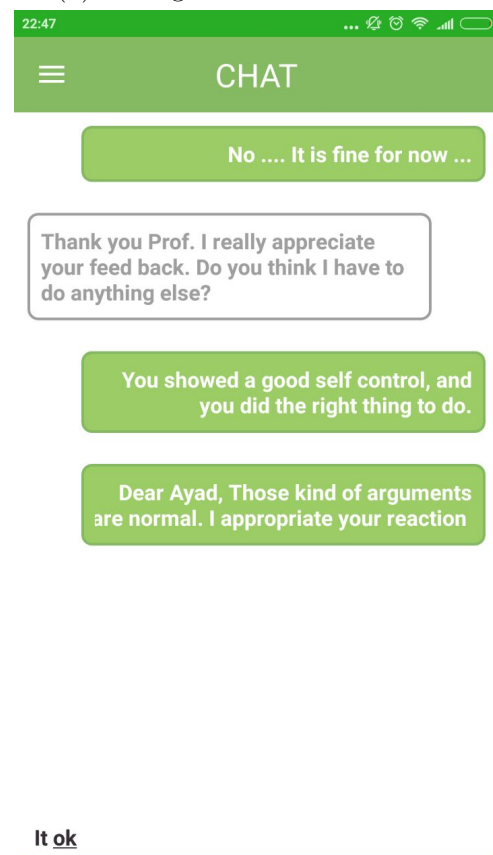
(a) Login screen



(b) Adding a new reflection screen



(c) Coaching screen



(d) Comments screen

Figure 5.8: Screenshots from the final Android mobile application

5.3.2 Database implementation

As we described in section 2.5 we used Parse as our cloud backend database. Using the tools available on Parse we designed and created our mobile application database. Figure 5.9 shows the database diagram that has been implemented in Parse.

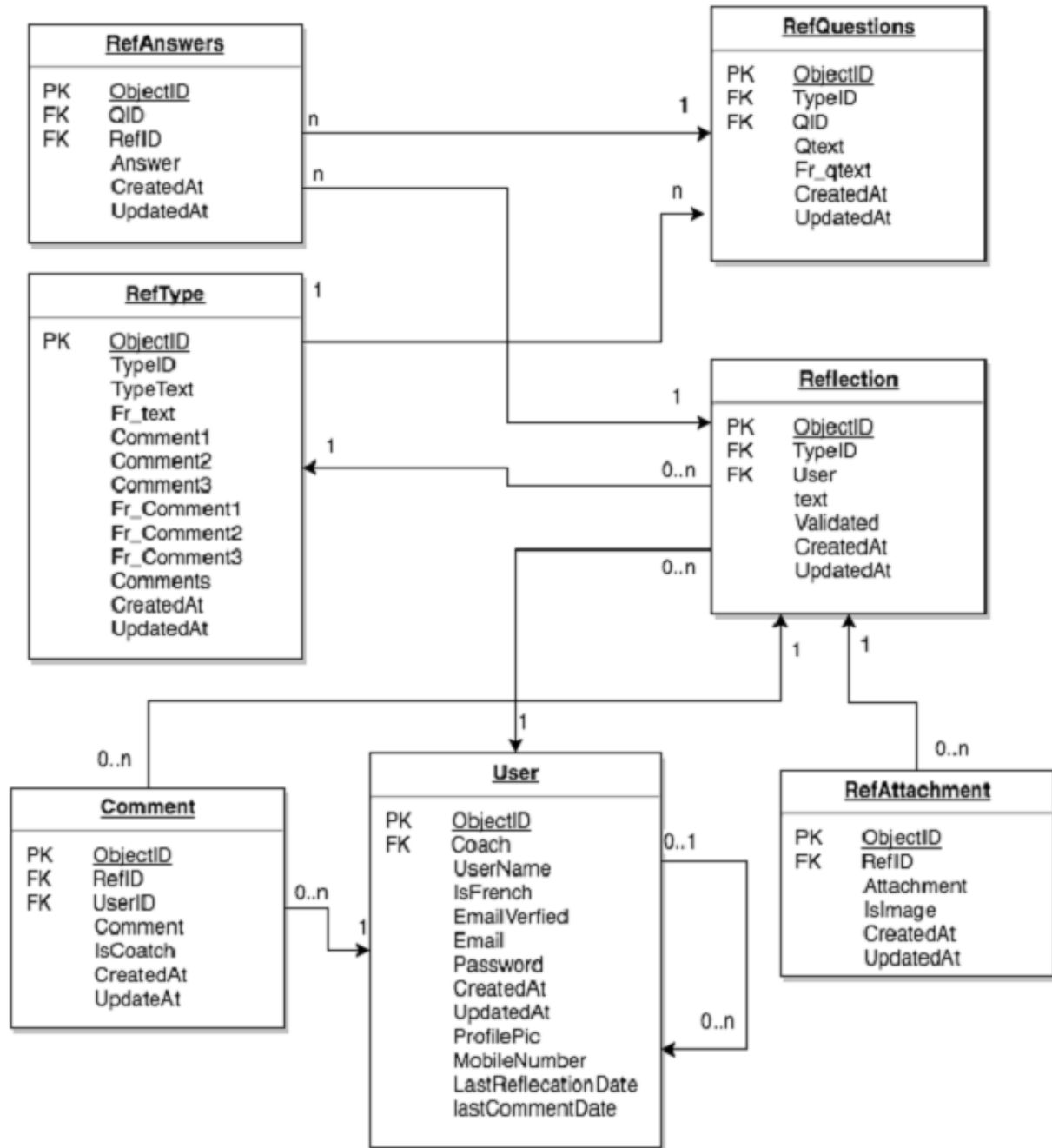


Figure 5.9: Database Diagram

5.3.3 Main implementation issues

During the implementation process, I faced some critical issues that prevented me from moving forward. So I had to find a quick solution for each of them. Thanks to the scrum agile methodology that I was following, which allows me to take quick actions regarding these issues. The main issues and difficulties I had were:

Database APIs (webservices)

The first problem I had is to access the web application database. Because the web application was developed using Symfony, which is a very advanced frameworks and it provides many security constraints, I could not access the database nor developing new APIs to communicate with our mobile application. Moreover, modifying the existing web application was out of the scope of the project. I tried also to access the database directly and bypass the Symfony framework, but those tries were useless because the database contains special fields and values generated by the framework it self such as the reflection types and the reflection questions which are codes representing a string that uses a translation code in Symfony. So it was impossible to modify or add any record without passing through the framework.

To bypass this issue, I decide with the arrangement with my supervisor to use a new database specially for our mobile application. Then link the two databases together or develop the required webservices using Symfony. We choose to use Parse as a backend cloud database because of the capabilities and the APIs that it offers as we described earlier in section 2.5.

Offline Mode

I chose Parse to be our backend, this decision was made because Parse can provide tools and libraries to be able to use the application offline, but unfortunately after trying to use them I faced many issues. Those tools and libraries were not completed and they do not provide automatic solutions as they claimed. Also if we wanted to go through this feature and enable a local data store we might face many problems and side effects such as[27] [28]:

- Parse APIs allows us to write NoSQL queries, while the local database in iOS is SQLite which does not support NoSQL.
- Having a local data store allow us to have only one instance of any given PFObjct (a local representation of data persisted to the Parse cloud), This will be useless in our application because we have more than one instant of each PFObjcts (reflections, questions and their answers, and the coachees list for a given coach).
- Synchronizing the data. An algorithm should be developed to do the synchronization between the local data store and Parse database, taking into consideration that both

databases could have new or modified data. So at the end of the algorithm execution both databases should be consistence.

- Making the user aware about what information have been synced to the server and what information are only exist in his device.

To keep moving in the implementation process, I decided to ignore the ability to use the application offline for now, and come back to it later if I had more time after finalizing the application, or leave it for the future development.

Migration

After I started to use Parse as a backend for our mobile application, Facebook decided to shut it down in January 2017, which gave me another challenge to deal with. So at the end of the development we have migrate our backend to another server or service provider that is compatible with what we did earlier during this project. To do that we have three solutions:

- Parse is an open source project, so we can install Parse server on any private server and change the connection attributes in our mobile application. To be able to do that we should have our own private sever, rent a cloud server, or get the permission to install the backend on one of the university servers.
- Parse was very commonly used, so when they announced that they wanted to shut it down, many competitors tried to attract Parse users to migrate their backend to their servers. They provide free migration tools and attractive packages. We could find a good, stabile and free service provider to migrate our back end to it using the migration tools provided by them along with the migration tools provided by Parse.
- Develop the webservises using Symfony to link our mobile application directly to the web application database. This is the best solution we could do, but it will be time consuming because we have to learn about Symfony and to dig into the old web application code to make sure that we are not violating any constrain. C

Chapter 6

Validation

6.1 Tests

Our application was not very complicated nor containing many functionalities, which is why we decided to test it manually during the implementation. Each screen on both iOS and Android applications has been tested intensively considering all possible scenarios and cases. But we will do unit tests later, in order to help future developers to test the code automatically.

Performing tests using the integrated simulator during the development process allowed us to discover some bugs and errors that we corrected directly. At the end, when we finished the implementation, we did an intensive round of tests for the whole application, trying to cover all possible cases.

Then to do more realistic tests, we used real mobile devices (iPhone 5 and Samsung Galaxy S6 edge) to perform the tests again. Those tests allowed us to discover some issues that did not appear during the simulator tests.

6.2 Internal validation

Internal validation concerns the validation of the application by its stakeholders. These validations have been started from the beginning of the development during the weekly meetings with our supervisor. They enabled us to always keep the evolution of the application under control.

Once the development was completed, a first more formal validation was made using a pre-defined scenario that can be founded in the appendix 10.4. This scenario has a specific situation sketched, which the testers had to follow. The validation took about one our to be completed. During this scenario my supervisor played a new professor role, then a more experienced professor (coach). We followed the scenario description, which starts by describing an overview of the role

the tester is playing, then provides specific steps to be followed and specific actions to be done in order to complete a given task. Those steps cover the functionalities available in our application according to the tester role. That is why we did the validation twice. During the first my supervisor played a user role, and during the second he played a coach role.

Because of the fact that my supervisor was involved during the implementation process, he was already quite familiar with the application and its features, but it was also an opportunity for him to judge the real use of the application. So as a result of this validation test we got some performance and quality comments, also some future improvement comments.

6.3 External validation

External validation concerned the validation of the application by its future users. After the success of the internal validation and various improvements, we carried out the validation at the fire department in Dinant. Julien Balasse and I went there to validate the application with new users who did not know anything about the application before. We started the validation with an overview about the research and the main purpose of the application, we did not teach them how to use the application nor show them the application before starting using it, because we wanted to see their reactions during their first use of the application, and to measure the non-functional requirements such as user-friendliness and the ease-of-use and we wanted them to navigate through the various functions. We started the validation with four firefighters and Julien Balasse using their own mobile phones. The validation took around two hours to be completed, following the same methodology used during the internal validation. We did the scenario number 1 and also did it two times in order to let them play both the role of a user and of a coach. At the end of the validation we asked them to fill an evaluation questionnaire that can be found in appendix 10.5. This evaluation questionnaire was divided into four main criteria, each one of them further subdivided into sub-criteria that could be evaluated with a value from 1 to 5 representing, respectively, not satisfied, partially satisfied, natural, satisfied and very satisfied. Those main criteria were:

- C10 – Design: which evaluates the design of the application, the choices of the colors, the layout of the application, and if the application is attractive to be used.
- C20 – Functionality: which evaluates the features of the application, do they meet their expectations, and do the required features exist and function as expected.
- C30 – Efficiency: which evaluates the ease to use of each functionality and the speed of the application.
- C40 – Application: which evaluates the application overall.

Figure 6.1 shows the average score of each criterion and its sub-criteria that have been obtained from the testers after doing the validation scenario number 1.

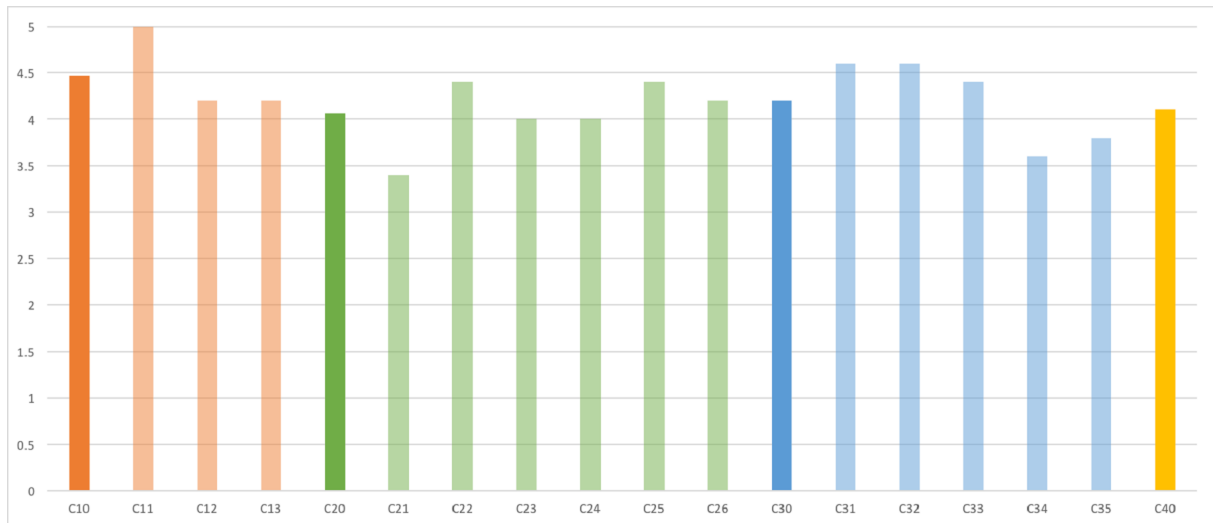


Figure 6.1: The average scores of the external evaluation

As can be seen from the figure, the validation tests were very successful. The firefighters were convinced by the application and they find it user friendly and easy to use. The validation went very smooth and I did not have to give them any hint on how to use any functionality during the test, even though they were using this application for the first time. The validation test has not identified any critical error. However, the firefighters did make some comments and suggested some future improvements such as adding the ability to have several coaches for one user and asking to improve the speed of the application which was a bit slow for the tester who was using the 3G network instead of the Wi-Fi. This issue was due to the weak mobile network coverage in the city of Dinant.

Chapter 7

Future development

The mobile applications were completed as stand alone applications. In order to make them more functional and useful there are some future improvements that could be done. As we described in section 5.3.3, we faced some difficulties and issues during the implementation so we decided to have a quick solution and move the actual problem to deal with it during future development. Also we had several improvement comments during the internal and external validation phases. This chapter highlights the main improvements that could be done in future.

Database APIs (Symfony webservices): As we mentioned in section 5.3.2.1 our mobile applications use a cloud backed instead of interaction directly with the web application database. This is the first important issue to deal with in future development, to give the user the ability to use both web application and mobile application at the same time. In order to develop those webservices, the future developer should know how to work with Symfony framework, and he/she should understand the web application code in order to develop a reliable webservices that do not violate the constraints and security measures of the application.

Offline mode: We can technically refer to it as "local database". As we mentioned in section 5.3.2.2 in order to run our mobile applications when the device is not connected to the internet, the future developer should develop a synchronizing algorithm to deal with the modified and the new records in both databases. In our case we cannot consider one side is always up to date, because while the user is offline he can add new reflection while in the same time his coach can write a comment. So the algorithm should be able to identify those changes and update both databases. Moreover, finding a solution to bypass the issues that could be arise from providing such functionality as described in section 5.3.3.

Include more of the web application's functionalities: As we discussed in the section 4.2 when we gathered the requirements of our mobile application. The psychology team did not want to include in the mobile application all functionalities existed in the web application. But after developing and validating the mobile application, we figured out some necessary functions that makes the use of the application easier such as considering the organization of the users including

all functionalities related to it, granting the coach role directly from the mobile application, managing and controlling the goals assigned to each user, and adding the ability to share a reflection with other users.

Enhancing the notifications: Our mobile application contains a basic notification system, to notify the users when the coach comments on their reflections, also to notify the coach when any of his coachees added new reflection or commented on an existing one. The future developer could enhance those notifications by adding a notification symbol on the reflection record itself within the reflection list to indicate that this reflection is new or contains new comments. Moreover, it could be enhanced more by adding the number of notifications related to each reflection.

iPad support: Our iOS application was designed to run on iPhone with iOS version 8.0 or higher. The future developer could optimize it to make it compatible with iPads. To be able to do that only the design and the layout should be optimized.

Apple's App Store: If the psychology team decided to make their iOS mobile application available on Apple's App Store, the future developer should follow App Store review guidelines [26], to make sure that the application meets their strict requirements. Moreover, he/she should enroll in Apple's iOS Developer Program then get a Distribution Certificate, because the university's developer account does not allow to distribute the application in Apple's App Store.

Chapter 8

Conclusion

The aim of this master thesis was to develop ReflectOn mobile application as an extension to the existing ReflectOn web application. At the end we have observed almost all core functionalities that should be available according to our requirements. I developed two native mobile applications for both iOS and Android platforms. Both mobile applications can perform all functionalities needed, which stated earlier in section 4.2, but with two limitations. The backend of the mobile applications is different than the backend of the web application as described in section 5.3.3, and the applications need an internet connection to be used as described in section 5.3.3.

The results of this thesis have met its objectives specified earlier in section 1.4. The validation results were promising, the firefighters liked the mobile application and they were able to use it without any previous training, also they said it meets their expectations and covers all functionalities.

In terms of learning objectives, this thesis was an opportunity for me to get some experience in developing mobile applications. Moreover, facing some obstacles and problems during the implementation, made me think out of the box and try to find solutions that do not block the development of the application and in the same time do not reduce or deduct many functionalities. Also analyzing and comparing the available tools to develop such application enhanced my experience to professionally and reasonably choose the most suitable tools to use, even though at the end we noticed that it was a wrong decision to go for two native applications instead of a cross-platform. Because I was very optimistic in evaluating the time needed to learn and develop the application in two native languages. This will make me think more about my decisions in future and try not to be very optimistic but more reasonable.

Chapter 9

Bibliography

- [1] ReflectOn web application. <http://reflecton.be/>
- [2] Scrum. <https://en.wikipedia.org/wiki/Scrum>: Scrum (software development)
- [3] Swift. <https://swift.org/about/>
- [4] JAVA. <https://en.wikipedia.org/wiki/Java>)
- [5] Parse. <https://parse.com/>
- [6] PHP. <http://php.net/>
- [7] MySQL. <https://www.mysql.com/>
- [8] Symfony2. <https://symfony.com/>
- [9] ORM. <https://en.wikipedia.org/wiki/Object-relational-mapping>
- [10] Object-oriented programming. <https://en.wikipedia.org/wiki/Object-oriented-programming>
- [11] Objective-C. <https://en.wikipedia.org/wiki/Objective-C>
- [12] Xcode. <https://developer.apple.com/xcode/>
- [13] Android SDK. <https://developer.android.com/studio/index.html>
- [14] Android Studio. <https://developer.android.com/studio/intro/index.html>
- [15] NoSQL. <https://en.wikipedia.org/wiki/NoSQL>
- [16] Cross Platform Development. <http://www.appfutura.com/forum/multiplatform-development/topic421>
- [17] Cross platform vs Native development. <http://www.developereconomics.com/pros-cons-top-5-cross-platform-tools/>

- [18] Business process management. <https://en.wikipedia.org/wiki/Business-process-management>
- [19] Mockup. <https://en.wikipedia.org/wiki/Mockup>
- [20] Mockingbot. <https://mockingbot.com/>
- [21] BitBucket. <https://bitbucket.org/>
- [22] MVC. <https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>
- [23] Drop Down Cell library. <https://github.com/AssistoLab/DropDown>
- [24] Voicememo library. <https://github.com/MoZhouqi/VoiceMemos>:
- [25] Alternative iOS Keyboard. <https://github.com/hackiftekhar/IQKeyboardManager>
- [26] Apple Review Procedures. <https://developer.apple.com/app-store/review/>
- [27] Offline Application issues. www.bricklin.com/offline.htm
- [28] Parse local datastore limitations. <https://parse.com/docs/ios/guide>

Chapter 10

Annex

10.1 questions

Critical incident

1. What was the incident ?
2. Why the incident was unexpected/special ?
3. What was the problem linked to this incident ?
4. What competencies were necessary to handle this incident ?
5. What are the development goals you have worked on thanks to this incident ?
6. What have you improved in order to reach your goals ?

Individual reflection

1. Describe briefly what is the context of this individual reflection, what it is about.
2. What have made easier the learning ?
3. What have made harder the learning ?
4. What are the development goals you have worked on thanks to this activity ?
5. What have you improved in order to reach your goals ?
6. Do you have developed a competency thanks to this activity ?
7. How can you learn better ?

Debriefing

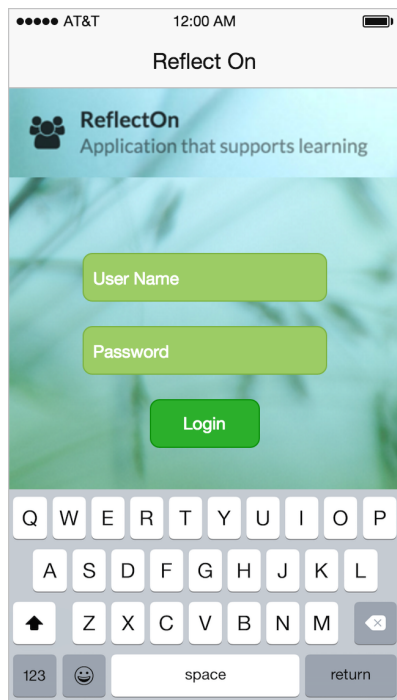
1. Describe briefly what was the context of this briefing, what it was about.
2. What do you learn from it ?

3. Is there anything you would do in a different way next time ?
4. Is there anything you must still learn ?
5. Do you feel that you should have been followed in your learning ? How ?
6. What are the development goals you have worked on thanks to this debriefing ?
7. What have you improved in order to reach your goals ?

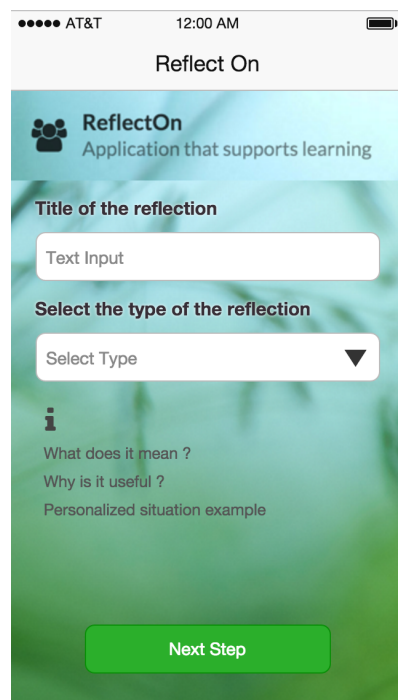
Informal feedback

1. Describe briefly what was the context of the feedback, what it was about and when.
2. Who give you the feedback ? A colleague, a supervisor ?
3. Was it useful ? Yes/No then explain.
4. Does this feedback give you a chance to improve yourself ? Yes/No then explain.
5. What are the development goals you have worked on thanks to this feedback ?
6. What have you improved in order to reach your goals ?

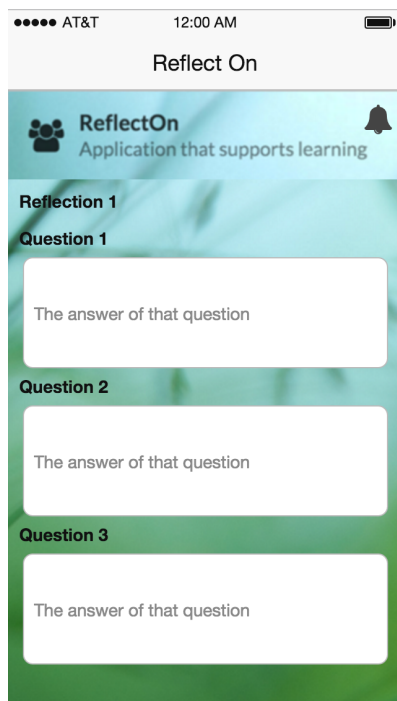
10.2 Mockups



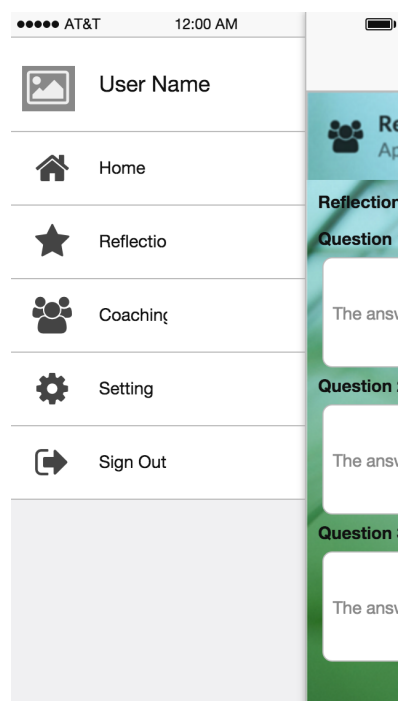
(a) Login screen



(b) Selecting the type of the new reflection screen

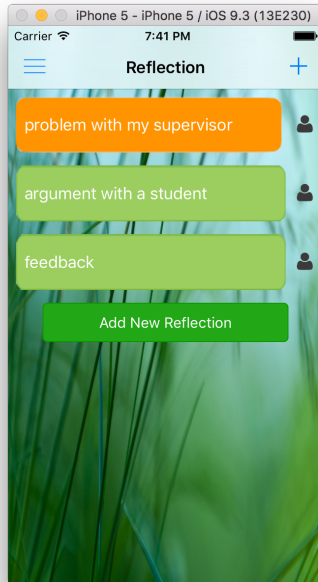


(c) Adding new reflection screen

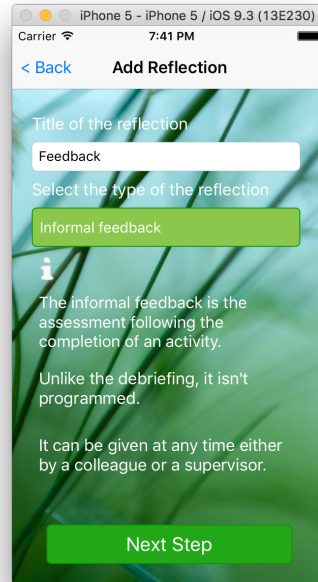


(d) Left side menu

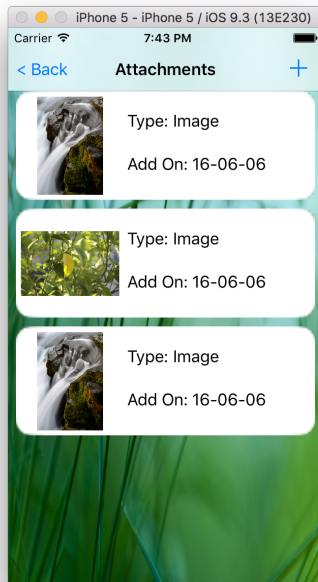
10.3 Mobile Applications Screenshots



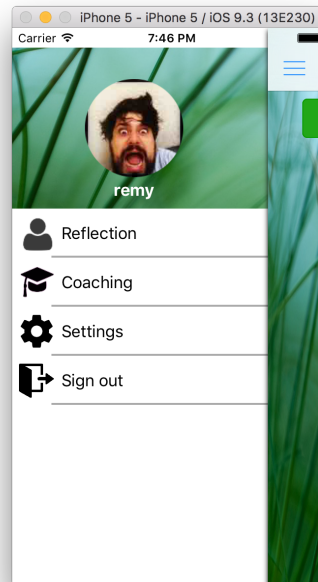
(a) Reflections list screen



(b) Selecting the type of the new reflection screen



(c) Attachments screen



(d) Left side menu

10.4 Scenarios

Scenario 1 sheet 1

Role: Firefighter, coachee

Scenario:

Your name is **Martin Trottier**.

You are a young firefighter who just joined Namur fire department. To assist you in learning your new job, your colleague "**Remy Trudeau**" proposed to be your coach using "ReflectOn" mobile application.

Usage:

- 1) To start using the application, launch the "ReflectOn" application on your mobile phone. Log in using the previously approved username "**martin**" using the password **123**.
- 2) Using the left side menu, click on the profile photo to access your profile information. Change your profile picture, your email and your phone number.
- 3) Now, try to remember a positive feedback that your supervisor gave you yesterday during a meeting. The feedback is about a fire that was easily contained because of the good collaboration between the team members. You want to encode this reflection in the application.

First, create a new reflection, check the 4 different types and choose the most appropriate one to the situation. Upon completing this process, you start answering a few questions. Suddenly, you have to leave urgently, you decide to save the reflection and continue later. A few hours later, you come back to continue encoding the reflection.
- 4) You have two photos in your gallery which you need to attach to your reflection.
- 5) At the end validate your reflection.
- 6) After some time, you go back to your reflection because your coach has commented on it. You check his comment and then you reply to it.

Scenario 1 sheet 2

Role: Firefighter, coachee

Scenario:

Your name is **Nicolas Collin**.

You are a young firefighter who just joined Namur fire department. To assist you in learning your new job, your colleague "**Olivia Jennings**" proposed to be your coach using "ReflectOn" mobile application.

Usage:

- 1) To start using the application, launch the "ReflectOn" application on your mobile phone. Log in using the previously approved username "**nicolas**" using the password **123**.
- 2) Using the left side menu, click on the profile photo to access your profile information. Change your profile picture, your email and your phone number.
- 3) Now, try to remember a negative feedback that your supervisor gave you yesterday during a meeting. The feedback is about a fire that was not easily contained because of your slow performance and because you were not collaborating efficiently with the team members. You want to encode this reflection in the application.

First, create a new reflection, check the 4 different types and choose the most appropriate one to the situation. Upon completing this process, you start answering a few questions. Suddenly, you have to leave urgently, you decide to save the reflection and continue later. A few hours later, you come back to continue encoding the reflection.
- 4) You have two photos in your gallery you need to attach them to your reflection.
- 5) At the end validate your reflection.
- 6) After some time, you go back to your reflection because your coach has commented on it. You check his comment then you reply to it.

Scenario 1 sheet 3

Role: Firefighter, coach

Scenario:

Your name is **Remy Trudeau**.

You are a senior firefighter in the Namur fire department and you have an excellent experience.

To help **Martin Trottier**, a young firefighter who just joined your department, in learning his new job, your manager offered you to be his coach using the "ReflectOn" mobile application.

Usage:

- 1) To start using the application, launch the "ReflectOn" application on your mobile phone. Log in using the previously approved username "**remy**" using the password **123**.
- 2) Using the left side menu, click on the profile photo to access your profile information. Change your profile picture, your email and your phone number.
- 3) Using the left side menu too, navigate to the "Coaching" screen. View the users that you are coaching.
- 4) Choose your new coachee "Martin" from the list and review his new reflection and its attachments. After reading it, you realize that this reflection is very well explained and positive. Moreover, it shows that your coachee has been very effective within the team.
- 5) Write a comment on his reflection showing your support.
- 6) After some time, you go back to the same reflection because Martin has commented on it.

Scenario 1 sheet 4

Role: Firefighter, coach

Scenario:

Your name is **Olivia Jennings**.

You are a senior firefighter in the Namur fire department and you have an excellent experience.

To help **Nicolas Collin**, a young firefighter who just joined your department, in learning his new job, your manager offered you to be his coach using the "ReflectOn" mobile application.

Usage:

- 1) To start using the application, launch the "ReflectOn" application on your mobile phone. Log in using the previously approved username "**olivia**" using the password **123**.
- 2) Using the left side menu, click on the profile photo to access your profile information. Change your profile picture, your email and your phone number.
- 3) Using the left side menu too, navigate to the "Coaching" screen. View the users that you are coaching.
- 4) Choose your new coachee "Nicolas" from the list and review his new reflection and its attachments. After reading it, you realize that this reflection is very pessimistic.
- 5) Write a comment on his reflection showing your support and to encourage him to work more on his skills, and with time he will get the experience he needs to avoid such situations.
- 6) After some time, you go back to the same reflection because Martin has commented on it.

Scenario 2 sheet 1

Role: Professor, coachee

Scenario:

Your name is **Benoît Hutan**.

After a brilliant research career, UCL wanted to hire you as a professor. To assist you in learning your new teaching profession, your colleague "**Emilio Vargas**" was proposed to be your coach using the "ReflectOn" mobile application.

Usage:

- 1) To start using the application, launch the "ReflectOn" application on your mobile phone. Log in using the previously approved username "**benoit**" using the password **123**.
- 2) Using the left side menu, click on the profile photo to access your profile information. Change your profile picture, your email and your phone number.
- 3) During one of your courses, an argument started between you and one of the students regarding an algorithm you just explained, He was sticking to his idea and you were not sure if it worked or not. After a long argument you decided to tell him that you will check it and inform him. You want to encode this reflection in the application.

First, create a new reflection, check the 4 different types and choose the most appropriate one to the situation. Upon completion this process, you start answering a few questions. Suddenly, you have to leave urgently, you decide save the reflection and continue later. A few hours later, you come back to continue encoding the reflection.
- 4) You have two photos in your gallery you need to attach them to your reflection.
- 5) At the end validate your reflection.
- 6) After some time, you go back to your reflection because your coach has commented on it. You check his comment then you reply on it.

Scenario 2 sheet 2

Role: Professor, coach

Scenario:

Your name is **Emilio Vargas**.

You are a very experienced professor at UCL. To help **Benoît Hutan**, a young professor, in learning his new teaching profession, your management has offered you to be his coach using the "ReflectOn" mobile application.

Usage:

- 1) To start using the application, launch the "ReflectOn" application on your mobile phone. Log in using the previously approved username "**emilio**" using the password **123**.
- 2) Using the left side menu, click on the profile photo to access your profile information. Change your profile picture, your email and your phone number.
- 3) Using the left side menu too, navigate to the "Coaching" screen. View the users that you are coaching.
- 4) Choose your new coachee "**Benoît**" from the list and review his new reflection and its attachments. After reading it, you realize that this he acted very well during that incident.
- 5) Write a comment on his reflection showing your support and to encourage him to work more on his skills to control anger, and with time he will get the experience he needs to avoid such situation.
- 6) After some time, you go back to the same reflection because Martin has commented on it.

10.5 Questionnaire

<u>Design</u>		1	2	3	4	5
The choice of the colors and the harmony between them						X
The layout of the application					X	
The application is attractive to be used					X	
Comments						

<u>Functionality</u>		1	2	3	4	5
properly fulfill their roles (logic of expected actions with the application)				X		
The application meets the objectives and expectations (general operation)					X	
partial reflection (ability to leave and return later)					X	
Adding new reflection functionality				X		
The coaching functionality					X	
Comments functionality						X
Comments						

1- not satisfied 2- partially satisfied 3- natural 4- satisfied 5- very satisfied

<u>Efficiency</u>		1	2	3	4	5
Easy to understand the application features					X	
Easy to navigate and access all functionalities						X
Easy to use (adding reflection, coaching and commenting)						X
Error and success messages				X		
Response speed					X	
Comments						

<u>Application</u>		1	2	3	4	5
Over all score of the application					X	

<u>Suggested Improvements</u>						

1- not satisfied 2- partially satisfied 3- natural 4- satisfied 5- very satisfied

<u>Design</u>		1	2	3	4	5
The choice of the colors and the harmony between them						✓
The layout of the application					✓	
The application is attractive to be used					✓	
Comments						

<u>Functionality</u>		1	2	3	4	5
properly fulfill their roles (logic of expected actions with the application)				✓		
The application meets the objectives and expectations (general operation)					✓	
partial reflection (ability to leave and return later)					✓	
Adding new reflection functionality					✓	
The coaching functionality					✓	
Comments functionality				✓		
Comments						

<u>Efficiency</u>		1	2	3	4	5
Easy to understand the application features					✓	
Easy to navigate and access all functionalities				✓		
Easy to use (adding reflection, coaching and commenting)					✓	
Error and success messages					✓	
Response speed					✓	
Comments						

<u>Application</u>		1	2	3	4	5
Over all score of the application					✓	

<u>Suggested Improvements</u>						

1- not satisfied 2- partially satisfied 3- natural 4- satisfied 5- very satisfied

<u>Design</u>		1	2	3	4	5
The choice of the colors and the harmony between them						✓
The layout of the application					✓	
The application is attractive to be used					✓	
Comments						

<u>Functionality</u>		1	2	3	4	5
properly fulfill their roles (logic of expected actions with the application)				✓		
The application meets the objectives and expectations (general operation)						✓
partial reflection (ability to leave and return later)					✓	
Adding new reflection functionality					✓	
The coaching functionality						✓
Comments functionality					✓	
Comments						

1 - not satisfied 2 - partially satisfied 3 - natural 4 - satisfied 5 - very satisfied

<u>Efficiency</u>		1	2	3	4	5
Easy to understand the application features						✓
Easy to navigate and access all functionalities						✓
Easy to use (adding reflection, coaching and commenting)					✓	
Error and success messages					✓	
Response speed				✓		
Comments						

<u>Application</u>	1	2	3	4	5
Over all score of the application				✓	

<u>Suggested Improvements</u>					

1 - not satisfied 2 - partially satisfied 3 - natural 4 - satisfied 5 - very satisfied

<u>Design</u>					
	1	2	3	4	5
The choice of the colors and the harmony between them					✓
The layout of the application				✓	
The application is attractive to be used				✓	
Comments					

		<u>Functionality</u>				
		1	2	3	4	5
properly fulfill their roles (logic of expected actions with the application)				✓		
The application meets the objectives and expectations (general operation)					✓	
partial reflection (ability to leave and return later)					✓	
Adding new reflection functionality					✓	
The coaching functionality					✓	
Comments functionality					✓	
Comments						

<u>Efficiency</u>		1	2	3	4	5
Easy to understand the application features						✓
Easy to navigate and access all functionalities						✓
Easy to use (adding reflection, coaching and commenting)					✓	
Error and success messages				✓		
Response speed				✓		
Comments						

<u>Application</u>	1	2	3	4	5
Over all score of the application				✓	

<u>Suggested Improvements</u>

1- not satisfied 2- partially satisfied 3- natural 4- satisfied 5- very satisfied

<u>Design</u>		1	2	3	4	5
The choice of the colors and the harmony between them						✓
The layout of the application						✓
The application is attractive to be used						✓
Comments						

<u>Functionality</u>		1	2	3	4	5
properly fulfill their roles (logic of expected actions with the application)						✓
The application meets the objectives and expectations (general operation)						✓
partial reflection (ability to leave and return later)					✓	
Adding new reflection functionality						✓
The coaching functionality						✓
Comments functionality						✓
Comments						

		<u>Efficiency</u>				
		1	2	3	4	5
Easy to understand the application features						X
Easy to navigate and access all functionalities						X
Easy to use (adding reflection, coaching and commenting)						X
Error and success messages					X	
Response speed						X
Comments	I have one more error but that can be fixed. minor					

<u>Application</u>					
	1	2	3	4	5
Over all score of the application				X	

<u>Suggested Improvements</u>				
The challenge for the application is to be used regularly by the users. My global feedback is that the application comes to the goal and could be a bit improved on the dimensions. Some modifications should be added when you have answers. more clear				

1- not satisfied 2- partially satisfied 3- natural 4- satisfied 5- very satisfied

