Université Catholique de Louvain

# UCLCampus: a moblie application for UCL students

Master's Thesis

*Authors:*

Arnold Moyaux

Baptiste Lacasse

*Supervisor:*

Pr. Yves Devilles

April 2, 2016

# Contents

# 1 Introduction

Brief introduction of the project, the goals and the contents of the rest of the thesis.

# 2 Preliminary analysis

In this section, we will look at the different existing technologies relating to the different aspects of our project and will explain the choices we made. // TODO

## 2.1 Cross-platform mobile development tools

In each of these sections, we will detail the different approaches one could choose to develop a cross-platform moblie applications. We will also present several frameworks using these approaches. We will then compare them and choose one of those approaches for the rest of the project. //TODO

### 2.1.1 The native approach

The first approach we considered for our project was what we call a native approach. The native approach consists in using the native technology and language for each platform, for instance Java for Android and Objective-C for iOS.

| Pros | Cons |
|---|---|
| Best achievable performance | Low maintainability |
| Always up-to-date with the latest API | Harder to find contributors fluent in all technologies |
| Can use any platform | Can lead to different versions of the application |

Table 1: Pros and cons of the native approach

### 2.1.2 The web approach

A second approach we considered was the web approach. This approach consists in using HTML5 to develop an application that will be usable on any platform.

| Pros | Cons |
|---|---|
| Can be used on any mobile platform | Doesn't have access to native platform features |
| Easy to find contributors fluent in HTML5 | Harder to implement local storage/security (//TODO NEED BETTER SOURCE) |
| Easy to maintain | Not as performant as native |

Table 2: Pros and cons of the web approach

### 2.1.3 The hybrid approach

The last approach to develop a mobile application is called the hybrid approach. An hybrid app is mostly built using HTML5 and JavaScript and is then wrapped inside a thin native container, giving it access to native features.

| Pros | Cons |
|---|---|
| Can be used on any mobile platform | Not as performant as native |
| Easy to find contributors fluent in HTML5 and JavaScript | |
| Easy to maintain | |

Table 3: Pros and cons of the web approach

#### 2.1.4 Our choice

### 2.2 Open-source project and code sharing

In this section, we will explain the choices we made concerning the code sharing platforms we used as well as the licenses we used to protect our work.

### 2.3 Project Management Methodologies

Here we detail the choices we made as to how we were going to manage de different parts of the project.

## 3 Project definition

In this part, we will show how we defined the relevant functionalities of our application as well as the user interface.

### 3.1 Choice of functionalities and sections

### 3.2 User interface

## 4 Project implementation

Here we will explain the overall architecture of the application. We will also explain some aspects we considered when implementing the application.

### 4.1 Architecture

### 4.2 Coding standards

### 4.3 Security

### 4.4 Information retrieval

# 5 The application

In this section we will present the application as we implemented it.

## 5.1 Functionalities

### 5.1.1 Studies

### 5.1.2 Campus

### 5.1.3 City

### 5.1.4 Tools

### 5.1.5 Others

## 5.2 Modularity and how to add a new functionality

## 5.3 Future functionalities and possible improvements

# 6 Post-implementation analysis

Here we will reflect about the many choices we made and try to decide wether they were the right ones or not.

## 6.1 Ionic framework

## 6.2 GitHub

## 6.3 Project Management

# 7 Conclusion

# 8 Bibliography