

---

**UCL**

---

**Université  
catholique  
de Louvain**

---

UNIVERSITE CATHOLIQUE DE LOUVAIN

ECOLE POLYTECHNIQUE DE LOUVAIN



# UCLCampus: a mobile application for UCL students

Supervisor: Yves DEVILLE

Readers: Kim MENS

hildeberto MENDONÇA

Mathieu ZEN

Jorge PEREZ MEDINA

Thesis submitted for the Master's degree

in computer science and engineering

options:

by : Arnold MOYAUX

Baptiste LACASSE

Louvain-la-Neuve

Academic year 2015-2016

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	Cross-platform mobile development tools . . . . .	5
2.1.1	The native approach . . . . .	5
2.1.2	The web approach . . . . .	6
2.1.3	The hybrid approach . . . . .	6
2.1.4	Our choice . . . . .	6
2.2	Open-source project and code sharing . . . . .	6
2.3	Project Management Methodologies . . . . .	7
<b>3</b>	<b>Functionalities of UCLCampus</b>	<b>8</b>
3.1	Choice of functionalities and sections . . . . .	8
3.2	User interface . . . . .	8
<b>4</b>	<b>Implementation</b>	<b>9</b>
4.1	Architecture . . . . .	9
4.1.1	Folder . . . . .	9
4.1.2	State provider . . . . .	10
4.2	Coding standards . . . . .	10
4.3	Security . . . . .	10
4.4	Information retrieval . . . . .	10
<b>5</b>	<b>The application</b>	<b>11</b>
5.1	The application UCLCampus . . . . .	11
5.1.1	Studies . . . . .	11
5.1.2	Campus . . . . .	11
5.1.3	City . . . . .	11
5.1.4	Tools . . . . .	11
5.1.5	Others . . . . .	11
5.2	Modularity and how to add a new functionality . . . . .	11
5.3	Future functionalities and possible improvements . . . . .	11

<b>6</b>	<b>Analysis</b>	<b>12</b>
6.1	Ionic framework . . . . .	12
6.2	GitHub . . . . .	12
6.3	Project Management . . . . .	12
<b>7</b>	<b>Conclusion</b>	<b>13</b>
<b>8</b>	<b>Bibliography</b>	<b>14</b>

# Chapter 1

## Introduction

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

## Chapter 2

# Background

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 mobile 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 2.1: 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 2.2: 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 2.3: Pros and cons of the hybrid 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.

## **Chapter 3**

# **Functionalities of UCLCampus**

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**



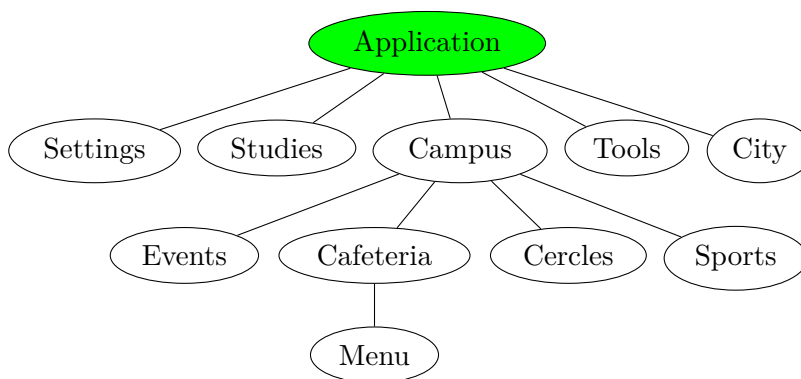
## Chapter 4

# 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

The purpose of the application is to be extensible and easily maintainable. We wanted a programmer can add its functionalities at each level to the application. For this we thought our implementation as a tree. At the top level we have the application with the settings menu and base configuration after it we have four branches pointing to global sections that we decided to create, themselves pointing to their functionalities and so on. Here is a part of the tree in order to give you the idea.



Each node having his own javascript,html and css (if needed) code.

#### 4.1.1 Folder

We wanted to keep the same state of mind for the file organisation. The ionic base architecture is to put all html file in a folder named templates, all js in a JS folder, ... The problem is that become messy once we have a lot of functionalities(maybe having more than one js and html). We modify it to respect the tree architecture we want. With our folder system a programmer can add his own subtree to the main tree. And if you want to modify a specific functionality,

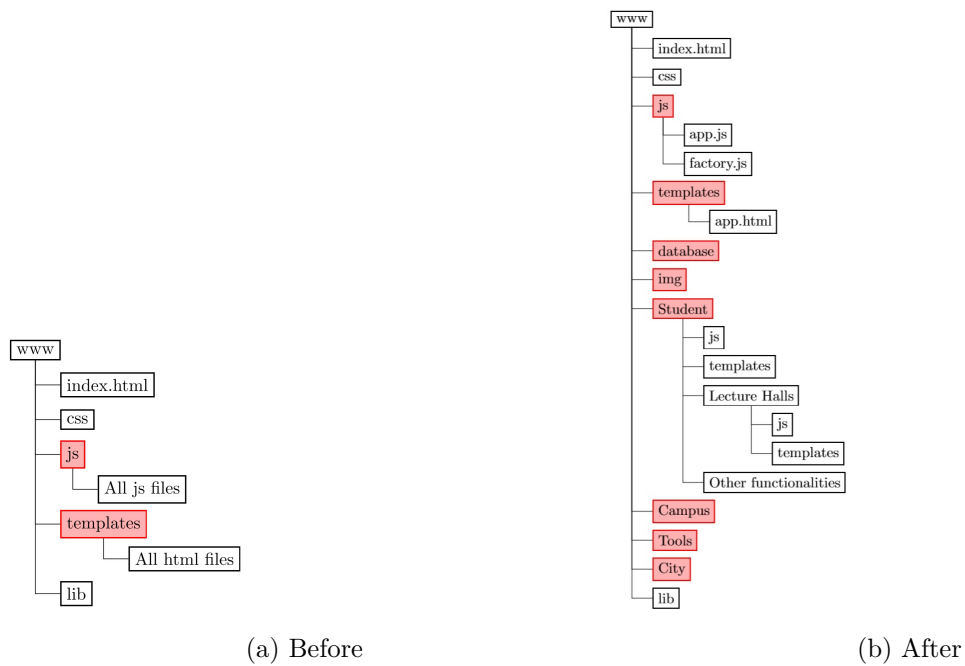


Figure 4.1: Folder evolution

you have directly access to the related files. Figure 4 1 is a summary of the change, red folder are those we modify.

#### 4.1.2 State provider

### 4.2 Coding standards

### 4.3 Security

### 4.4 Information retrieval

# Chapter 5

## The application

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

### 5.1 The application UCLCampus

#### 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

## Chapter 6

# 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

## Chapter 7

## Conclusion

## Chapter 8

# Bibliography