UNIVERSITE CATHOLIQUE DE LOUVAIN

ECOLE POLYTECHNIQUE DE LOUVAIN

UCLCampus: a moblie 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

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 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 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

In order to define what kind of functionalities we wanted to be part of our application, we needed to know what the students needed. The first step was to define a number of user stories that we would then translate into functionalities.
We split our user stories into several categories:

- Studies: anything directly related to a user's studies, for instance his classes, the lecture halls or the libraries.
- Campus: anything related to student life in the campus but not related to the user's studies. For instance "Kot à Projets" or "Cercles".
- City: anything related to the city the user is in but not related to the university. For instance a cinema or restaurants.
- Tools: the tools offered by the application that might relate to several other categories. For example the map.
- Settings: the settings of the application. For example the language or the currently selected campus.

We also define two types of users:

- Students: students can access all the functionalities of the application using their UCL login information. Indeed, some functionalities are student specific. For instance, it wouldn't make sense for a person who isn't a student to try to access his or her schedule.

- Users: users are people who aren't students but might still be interested in some functionalities the application has to offer.

In our user stories, any story starting by 'as a student' cannot be used by users while any story starting by 'as a user' can be used by both users and students.

We will now give a list of the different user stories we thought of for each of our categories.

Studies

- Schedule
  - As a student, I can access my schedule in order to know when my courses are given.
  - As a student, I want to know where a course is given.
  - As a student, I want to know the name of a teacher giving a certain course of my schedule.
  - As a student, I can export my schedule to my phone's agenda so that I don't need Internet access to see it.
- Libraries
  - As a user, I can see whether a library is open or closed.
  - As a user, I can display the address of any library.
  - As a user, I can have a GPS guide to access libraries from my location.
- Lecture halls
  - As a user, I can check the address of any lecture hall.
  - As a user, I can have a GPS guide to access lecture halls from my location.
- Websites
  - As a user, I can quickly access the moodle website through the application.
  - As a user, I can quickly access the UCL website through the application.

Campus

- Events
  - As a user, I can see a list of events taking place in my campus.
  - As a user, I can sort the events by category.
- Kots à Projet
  - As a user, I can check Kots à Projet to know their address and projects.

  - As a user, I can have a GPS guide to access Kots à Projet from my location.

- Cercles
  - As a user, I can check "Cercles" to know their address.
  - As a user, I can have a GPS guide to access "Cercles" from my location.

- Restaurants Universitaires
  - As a user, I can see the different "Restaurants Universitaires" in my campus.
  - As a user, I can check the menu of the "Restaurants Universitaires".
  - As a user, I can have a GPS guide to access "Restaurant Universitaires" from my location.

- Sports
  - As a user, I can see a list of sports organized in my campus.
  - As a user, I can sort the sports by day or by sport.

City

- Tourism
  - As a user, I can see the address of the city's information center.
  - As a user, I can see a list of the museums of the city I'm in.
  - As a user, I can see whether a museum is opened or closed.

- Activities
  - As a user, I can see the address of the city's cinema in order to access it with the help of a GPS guide.
  - As a user, I can see several activities I can do in the city I'm in.

- Restaurants and bars
  - As a user, I can see a list of the restaurants of the city I'm in.
  - As a user, I can see a list of the bars of the city I'm in.

Tools

- As a user, I can access a map of the city I'm in in order to check points of interests.
- As a user, I can receive help from a GPS guide in order to access a location of my choice on the map.

Settings

- As a user, I can change the application's language to French, English or Dutch.
- As a user, I can select my campus.

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

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