Bachelor’s thesis

Developing an HTML5 mobile application for Barcelona School of Informatics Intranet

Cristian Ortega Gómez

February, 2014

Directed by Maria José Casany Guerrero and Marc Alier Forment

Co-directed by Jaume Moral Ros
Thank you.

Maria José Casany and Marc Alier,
for accepting this project and helping me succeed with it.

Jaume Moral and Albert Obiols,
for accepting my suggestions to make improvements on the public API of the Barcelona School of Informatics Intranet in order to make a better and more useful application, as well as for their unconditional support.

Roger Sala and Marcel Arbó, 
for providing me information about their past projects and experiences when developing mobile applications of the Intranet as their final thesis.

Ideaknow (Inverline Networks, S.L.),
for loaning me Android and iOS devices to test the application as well as to some teammates for their support and for helping me with some issues, with special mention to:

    Oscar Salas    Ricardo Ramos

My family and friends,
for helping and supporting me to arrive here, with special mention to:

    Gracia Gómez    Laura Ortega    Jaime San Félix    Sergi Purcet
    Vicente Ortega    Pedro Mesa    Adrián Kleque    Estefanía Salinas

Table of contents

1. Introduction.......................................................................................................................... 11

2. State of art.......................................................................................................................... 12
   2.1. Android version............................................................................................................. 12
   2.2. iOS version.................................................................................................................. 13
   2.3. Applications from other Catalan universities............................................................... 15
   2.4. Satisfaction survey carried out .................................................................................... 16
   2.5. Existing Intranet API.................................................................................................... 17
      2.5.1. Public data available ............................................................................................. 17
      2.5.2. Private data available ......................................................................................... 17
      2.5.3. Other data available ............................................................................................ 18

3. Project definition................................................................................................................. 19
   3.1. Scope............................................................................................................................. 19
   3.2. Objectives.................................................................................................................... 19
   3.3. Requirements .............................................................................................................. 19
      3.3.1. Functional requirements ...................................................................................... 19
      3.3.2. Non-functional requirements ............................................................................. 20
   3.4. Risks............................................................................................................................ 20
      3.4.1. Direct risks ........................................................................................................... 20
      3.4.2. Indirect risks ....................................................................................................... 20
   3.5. Use cases ..................................................................................................................... 21
      3.5.1. Notes ................................................................................................................... 22
      3.5.2. Authentication .................................................................................................... 24
      3.5.3. Settings ............................................................................................................... 25
# Table of illustrations

Figure 1: Current Android application interface .................................................................12  
Figure 2: Current Android application review on Google Play ........................................13  
Figure 3: Current iOS application version ........................................................................13  
Figure 4: Current iOS application version adaption for tablets .......................................14  
Figure 5: Current iOS application review on App Store ....................................................14  
Figure 6: Application from Universitat Politècnica de Catalunya (UPC) for Android ..........15  
Figure 7: iPhone app of Universitat Pompeu Fabra (UPF) ................................................15  
Figure 8: Platforms used by Barcelona School of Informatics' students and staff ..........16  
Figure 9: Use cases diagram .............................................................................................21  
Figure 10: View latest notes use case diagram ..................................................................22  
Figure 11: View note details use case diagram ..................................................................22  
Figure 12: Download attachment use case diagram .........................................................23  
Figure 13: Mark note as read use case diagram ................................................................23  
Figure 14: Authenticate use case diagram ........................................................................24  
Figure 15: Sign out use case diagram ..............................................................................24  
Figure 16: Change settings use case diagram ...................................................................25  
Figure 17: View timetable use case diagram ....................................................................25  
Figure 18: View subjects list use case diagram ................................................................26  
Figure 19: View subjects details use case diagram ..........................................................26  
Figure 20: View subject note use case diagram ..............................................................27  
Figure 21: View buildings use case diagram .....................................................................27  
Figure 22: View room details use case diagram ..............................................................28  
Figure 23: View news list use case diagram .....................................................................28  
Figure 24: View new details use case diagram ...............................................................29  
Figure 25: Agile waterfall model process ..........................................................................30  

*Figure 26: Gantt diagram of the project scheduling* ..........................................................34  
Figure 27: Actual Gantt diagram .....................................................................................34  
Figure 28: Conceptual schema .......................................................................................39  
Figure 29: Packages diagram ..........................................................................................41  
Figure 30: Login, news and notes packages diagram ......................................................42  
Figure 31: Notifications, rooms and settings packages diagram .......................................43
Figure 65: iPad rooms list
Figure 66: iPad timetable view
Figure 67: Android phone lateral menu and latest notes view
Figure 68: Android phone note details and subjects list view
Figure 69: Android phone subject details and news list views
Figure 70: Android phone new details and rooms list views
Figure 71: Android phone timetable views
Figure 72: Android phone push notifications overview
Figure 73: Android tablet latest notes view
Figure 74: Android tablet note details view
Figure 75: Android tablet subjects list view
Figure 76: Android tablet subject details view
Figure 77: Android tablet news view
Figure 78: Android tablet new details view
Figure 79: Android tablet rooms list view
Figure 80: Android tablet timetable view
Figure 81: Windows Phone lateral menu and latest notes view
Figure 82: Windows Phone note details and subjects list views
Figure 83: Windows Phone news views
Figure 84: Windows Phone rooms list and timetable views
Figure 85: Windows Phone timetable view with toggle arrows and push notifications overview
Figure 86: Windows 8 latest notes view full-size
Figure 87: Windows 8 note details view full-size
Figure 88: Windows 8 subject list view full-size
Figure 89: Windows 8 subject details view full-size
Figure 90: Windows 8 news list view full-size
Figure 91: Windows 8 new details view full-size
Figure 92: Windows 8 new details view half-size
Figure 93: Windows 8 room list view full-size
Figure 94: Windows 8 timetable view full-size
Figure 95: Windows 8 latest notes view snapped
Figure 96: Windows 8 push notifications overview
Figure 97: Total surveyed users
1. Introduction

Mobile phones have evolved quite a lot since their boom in the 90’s. At the beginning they were used only for calling family or friends, even colleagues or business partners. However, today’s devices are called smartphones because they have a very important difference with their predecessors: they are now smart.

Current mobile usage is incredibly increasing in part due to increasing number of devices being launched. This important revolution started with the Apple’s device known as iPhone in 2007 and has been followed by Android, Blackberry and Windows Phone devices. The most important feature of these devices is that they allow users to download and install applications from an integrated store, which can contain both free and paid ones.

According to some studies[1], the statistics show people could not live without a smartphone nowadays:

- **84%** of mobile users cannot go a single day without their device.
- Smartphone usage grew in 2012 by **81%**.
- **44%** of smartphone owners have slept with their phone nearby because they did not want to miss a notification.
- Mobile Internet traffic in 2012 was more than **12 times** the entire Internet in 2000.
- **2014** is likely to be the year when mobile-connected devices will exceed world’s population.

Furthermore, today we do not just use smartphones but we use a whole set of different devices connected to each other and to the cloud, such as tablets, laptops, consoles, desktops, phablets\(^1\) and so on. When developing an application, one must be aware of all of these available choices for customers. The more platforms your application is available for, the more users will be able to use it.

Moreover, regardless of the platform the user wants to choose, an application must run as fast and fluid as possible on each of them, and we must be aware of each platform specifications and recommendations. Ensuring this will make our application usable and available for all users, on any platform and on any device.

\(^1\) Phablet is an informal term used by touch-devices users to denominate those devices with a screen sized between five and seven inches, such as Samsung Galaxy Note I or II.
2. State of art

Nowadays, Barcelona School of Informatics Intranet has native applications for Android[2] and iOS[3] to access its contents and functionalities. These applications were developed as final bachelor’s thesis in 2011. They have been available to all students and faculty staff for two years and their maintenance has been supported by inLabFIB.

Both applications have support for subject notes (or news) and syllabus, lecturers contact information, school and university news, personal calendar schedule and so on. Moreover, their interfaces have been adapted to each platform taking into account platform specific recommendations and guidelines. Nevertheless, those guidelines have evolved quite a lot since these applications were developed and launched.

2.1. Android version

The Android version of the application was developed by Roger Sala Angordans as an Intern at inLabFIB on 2011. It allows users to sign in using their school’s account and they can see their subjects, calendar and so on. Figure 1 shows its interface.

It supports a wide variety of features but it has not been adapted to tablets. Hence, users with an Android tablet cannot use this application. However, as shown in Figure 2, 60 users have rated it on average of 4.1 about 5 and has been downloaded between 1,000 and 5,000 times in two years as of June 25th, 2013.

Figure 1: Current Android application interface
2.2. iOS version

The iOS version was developed by Marcel Arbó Lack also as an Intern at inLabFIB as his Bachelor’s thesis on 2011. It supports the same features than the Android version plus it automatically configures Eduroam² wireless network on the device. Figure 3 shows its interface.

In contrast with the Android version, the iOS version was adapted to tablets by inLabFIB on 2012. However, the interface has poorly been adapted as shown in Figure 4.

---

² Eduroam is a standard for wireless networks in educational environments such as universities and high schools. More information about Eduroam is available at [http://www.eduroam.org/](http://www.eduroam.org/).
Figure 4: Current iOS application version adaption for tablets

Information about downloads is not available but we can see it has been rated 4.5 about 5 on average by 30 users as of June 25th, 2013, as shown in Figure 5.

Figure 5: Current iOS application review on App Store
2.3. Applications from other Catalan universities

More and more universities and schools are publishing their own applications for smartphones[4]. The majority of them have public information about the institution and then they provide a link to the school’s intranet, but this is opened on an external browser and it is not part of the application but a normal webpage. Figure 6 shows an example of an application, which opens an external link for the Intranet.

However, some universities provide more powerful features, such as viewing qualifications in real-time and other subject information relatives. Figure 7 shows an example of these types of applications.
2.4. Satisfaction survey carried out

In order to know how useful these applications are for users, a survey has been carried out during the second semester of the course 2012-2013, through the Intranet's forum. A total of 80 students have participated in it. The entire survey and results are available at the end of this document as an annex.

As a conclusion, the following statements can be extracted from survey results:

- **70%** of surveyed users have a device with Android, **21%** with iOS and **3%** with Windows Phone. These proportions are quite similar than mobile OS current usage in Spanish market.
- **56%** of surveyed users said existing apps are less than “acceptable” in relation to performance.
- **35%** of surveyed users said existing apps are less than “acceptable” in relation to aesthetics and usability.
- **90%** of Android users complained about speed while loading new information.
- **65%** of iOS users complained about the inexistency of push-notifications in that platform version of the app.

*Figure 8: Platforms used by Barcelona School of Informatics’ students and staff*
2.5. Existing Intranet API

There is currently an Intranet API which provides access to both public and private data from users.

2.5.1. Public data available

**Subject’s timetable**: /api/horaris/horari-assignatures.txt
Returns the timetable of the given subjects.

**Subjects of a studies plan**: /api/horaris/assignatures-titulacio.txt
Returns the subjects of a studies plan which have timetable.

**Availability of a computer**: /api/aules/disponibilitat-pc.txt
Returns the availability of a computer, taking into account both classes and reservations.

**Laptop reservations**: /api/aules/reserves-portatils.txt
Returns the list of current laptop reservations.

**Available spaces in computer rooms**: /api/aules/places-lliures.json
Returns a list of available places in each computer room.

**Today’s computer rooms’ timetable**: /api/aules/horari-avui.ics
Returns today’s computer rooms’ timetable.

**Available subjects in EHEA studies plan**: /api/assignatures/llista.json
Returns a list of available places in each computer room.

**Subject information**: /api/assignatures/info.json
Returns the details of the given subject.

2.5.2. Private data available

**Personal timetable**: /api-v1/horari-setmanal.json
Returns the timetable of the current user in JSON format.

**Personal information**: /api-v1/info-personal.json
Returns the personal information of the current user.

**Calendar**: /api-v1/calendari-portada.ics
Returns the calendar for the current user.

**Enrolled subjects**: /api-v1/assignatures.json
Returns the list of enrolled subjects for the current user.

**Personal timetable:** /api-v1/horari.ics
Returns the timetable of the current user in ICS format.

**Latest notes:** /api-v1/avisos.json
Returns the latest notes for the current user in JSON format.

**Latest notes of a subject:** /api-v1/avisos-assignatura.json
Returns the latest notes for the current user and the given subject in JSON format.

**Download an attachment:** /api-v1/attachment
Returns the latest notes for the current user and the given subject in JSON format.

**Profile picture:** /api-v1/foto-personal.jpg
Returns the profile picture of the current user in JPG format.

**Mark a note as read:** /api-v1/marcar-llegit
Mark the given note as read to avoid duplicate notifications across all devices.

**Subscribe notification system:** /api-v1/subscribe-notification-system
Subscribes a device to a notification system (for push notifications).

**Unsubscribe notification system:** /api-v1/unsubscribe-notification-system
Unsubscribes a device to a notification system (for push notifications).

2.5.3. Other data available

**FIB news:** http://www.fib.upc.edu/en/rss.rss
Provides the latest news from FIB.

**UPC news:** http://www.upc.edu/saladepremsa/actualitat-upc/RSS?set_language=en
 Provides the latest news from UPC.

**A5 room availability map:** http://www.fib.upc.edu/poa/mapa.php?mod=a5
Provides a map of the A5 building to check available rooms.

**B5 room availability map:** http://www.fib.upc.edu/poa/mapa.php?mod=b5
Provides a map of the B5 building to check available rooms.

**C6 room availability map:** http://www.fib.upc.edu/poa/mapa.php?mod=c6
Provides a map of the C6 building to check available rooms.
3. Project definition

3.1. Scope

This project is focused in developing an HTML5 application compatible with Windows Phone 8, Windows 8, iOS 7 and Android 4.0+, all of them including their respective packaging for their stores.

3.2. Objectives

This project has one clear SMART objective:

“To design, develop and test an HTML5 application for Barcelona School of Informatics Intranet that works on any device before February, 2014.”

However, it can be divided into a few sub objectives, related to the scope explained above, all of them before February, 2014:

- Provide existing users with a more useful and usable app for iOS and Android.
- Provide an app for Windows 8 and Windows Phone 8 to students and faculty staff.
- Provide a more maintainable and extensible solution to inLabFIB.

3.3. Requirements

3.3.1. Functional requirements

- Users must be able to log in and log out.
- Users must be able to subscribe and unsubscribe from push notifications.
- Users must be able to view their latest notes, if any.
- Users must be able to directly download attached files.
- Users must be able to view their enrolled subjects and its details, as well as its notes and teachers.

---

3 Windows Phone 8, Windows 8, iOS 7 and Android 4.0+.
- Users must be able to view their timetable, if existing.
- Users must be able to check for an available room on any building related to Barcelona School of Informatics.
- Users must be able to view the latest notes from FIB and UPC.
- Users must be able to personalize the initial page to be shown on startup.

3.3.2. Non-functional requirements

- The application must run in Windows Phone 8 and Windows 8, as well as being compatible with iOS 7 and Android 4.0+.
- The application must adapt its interface to each platform accordingly its guidelines.
- The application has to be usable.
- The application must load quickly and be fast.

3.4. Risks

During the development of the project the following risks may take place.

3.4.1. Direct risks

- Important delays on project scheduling: it will be controlled once a month to ensure it is correct or to change it properly.
- Incompatibility of any framework or feature with any supported platform: in this case, we must ensure that a framework or feature will be compatible with all the supported platforms before implementing it.

3.4.2. Indirect risks

- Incompatibility with Intranet’s public API (Application Programming Interface): in this case, an alternate API will be provided to develop the application and missing information or failure will be added or fixed, respectively.
- Failure of any testing device or computer hardware/software: in this case, alternate software or hardware will be considered in the budget for these situations.

---

4 The Barcelona School of Informatics’ Intranet has a public API that allows developers to create applications with access to personal information from users.
3.5. Use cases

Figure 9: Use cases diagram
3.5.1. Notes

View latest notes

![View latest notes diagram]

Figure 10: View latest notes use case diagram

**Actor:** User.

**Trigger:** Current user wants to see its latest notes.

**Description:**

1. User selects “Latest notes” on the menu.
2. The application shows the latest notes for the current user currently stored on the device.
3. If synchronization time is applicable, the application refreshes the latest notes for the current user and shows them to the user.

View note details

![View note details diagram]

Figure 11: View note details use case diagram

**Actor:** User.

**Trigger:** Current user wants to see the details of a note.

**Description:**

1. User selects a note from the latest notes list or the subject notes list.
2. The application shows the details for the selected note.
Download attachment

![Download attachment use case diagram](image)

**Actor:** User.

**Trigger:** Current user wants to download an attachment from a note.

**Description:**
1. User selects an attachment from a note (which can be a latest note or a subject note).
2. The application downloads the attachment and shows it to the user.

Mark note as read

![Mark note as read use case diagram](image)

**Actor:** System.

**Trigger:** Current user is currently viewing the details of an unseen note.

**Description:**
1. The application communicates with the server to mark the current note as read.
3.5.2. Authentication

Authenticate

Figure 14: Authenticate use case diagram

Actor: User.

Trigger: User wants to log in.

Description:
1. User clicks on “Connect with my account”.
2. The application shows the log in form from the Intranet.
3. Once the user authenticates and authorizes the application, it receives an authorization token.
4. The application exchanges this token for an access token, in order to access private data from the actually logged in user.

Sign out

Figure 15: Sign out use case diagram

Actor: User.

Trigger: User wants to log out.

Description:
1. User selects “Sign out” option from the menu.
2. The application delete the access token and deletes personal information of the user from the device.
3.5.3. Settings

**Change settings**

*Figure 16: Change settings use case diagram*

**Actor:** User.

**Trigger:** User wants to change application settings.

**Description:**

1. User selects “Settings” option from the menu.
2. The application shows its settings and allows the user to freely change them.

3.5.4. Timetable

**View timetable**

*Figure 17: View timetable use case diagram*

**Actor:** User.

**Trigger:** Current user wants to see his/her timetable.

**Description:**

1. User selects “Timetable” option from the menu.
2. The application shows the timetable of the current user.
3.5.5. Subjects

View subjects list

*Figure 18: View subjects list use case diagram*

**Actor:** User.

**Trigger:** Current user wants to see his/her subjects he/she is currently enrolled in.

**Description:**

1. User selects “Subjects” option from the menu.
2. The application shows subjects list for the current user.
3. If synchronization time is applicable, the application refreshes the subjects for the current user and shows them to the user.

View subjects details

*Figure 19: View subjects details use case diagram*

**Actor:** User.

**Trigger:** Current user wants to see the details of a subject he/she is currently enrolled in.

**Description:**

1. User selects a subject from the subjects list.
2. The application shows the details for the selected subject, such as its notes and teachers.
3. If synchronization time is applicable, the application refreshes the subject information for the current user and shows them to the user.
View subject note

Figure 20: View subject note use case diagram

**Actor:** User.

**Trigger:** Current user wants to see the details of a note from a subject he/she is currently enrolled in.

**Description:**
1. User selects a note from the subject notes list.
2. The application shows the details of the selected note.

3.5.6. Rooms

View buildings

Figure 21: View buildings use case diagram

**Actor:** User.

**Trigger:** Current user wants to see the availability of each building.

**Description:**
1. User selects "Rooms" option from the menu.
2. The application shows the list of available rooms with the available spaces in each of them, as well as a map for each building.
3. If synchronization time is applicable, the application refreshes the buildings information and shows them to the user.
View room details

Figure 22: View room details use case diagram

**Actor:** User.

**Trigger:** Current user wants to see the daily schedule for a concrete room.

**Description:**

1. Current user selects a room from the buildings list.
2. The application shows the daily schedule for the selected class, as well as the current available spaces on it.

3.5.7. News

**View news list**

Figure 23: View news list use case diagram

**Actor:** User.

**Trigger:** Current user wants to see the news from FIB and/or UPC.

**Description:**

1. Current user selects “News” option from the menu.
2. The application shows the latest news from FIB and/or UPC.
3. If synchronization time is applicable, the application refreshes the latest news and shows them to the user.
View new details

Figure 24: View new details use case diagram

**Actor:** User.

**Trigger:** Current user wants to see the details of a new from FIB or UPC.

**Description:**

1. Current user selects a new from the news list.
2. The application shows the summary of that new and a link to continue reading it.
4. Project scheduling and methodology

The methodology used has been the agile waterfall model as shown in Figure 25.

Upon the project initialization, the analysis stage started and we collected all necessary documents, data and requirements of those features of the system to be developed in a period of time. Then, the design stage took place and in the implementation stage, those designed operations were implemented and tested.

Finally, when those developed features were accepted they were deployed or corrected, if necessary. This methodology will be done recursively for each considered feature sets in the project scheduling.

4.1. Considerations

The project began on June 13th, 2013 and finished on January 31st, 2014. As previously stated, the methodology used has been the agile waterfall model; this implies each stage is executed sequentially for each subset of tasks defined. Following we define these subsets, named sprints, and both the initial and real action plans.
4.2. Sprints and stages

In this project we could clearly difference between the following subsets of tasks:

4.2.1. Sprint 1: Project initialization and browser compatibility testing

This sprint was aimed to create the HTML5 application structure and store packages, in order to deploy them to targeted devices and ensure compatibility. The methodology stages were as follows:

- **Analysis**: to check out available frameworks and technologies for each platform and HTML5.
- **Design**: to think the HTML5 application structure and their possible problems.
- **Implementation**: to implement the designed HTML5 application structure, as well as to store packages of targeted platforms, and test it on devices.
- **Acceptance and deployment**: no further action required in this case.

4.2.2. Sprint 2: Login logic, first feature and application interface basis

The second sprint was aimed to create the application interface basis and to implement the first and most used feature, subject notes (or news), as well as login logic. In this sprint, the methodology stages were as follows:

- **Analysis**: to check out what information is currently offered by the API about subject notes, how they work in current apps and what users requested at the survey for this specific feature. Moreover, we had to check how to implement OAuth authentication in HTML5.
- **Design**: to think how login logic and first feature would be implemented.
- **Implementation**: to implement that designed architecture both for login logic and first feature, as well as test it on devices.
- **Acceptance and deployment**: to show the app to inLabFIB in this stage to get feedback.

4.2.3. Sprints 3 to 7: Implementation of all other supported features

The sprints from third to seventh were aimed to implement all other features supported, which are subjects information (including lecturers contact information and syllabus), school and university news, personal timetable and settings. In this case, the methodology stages were the same as in the sprint 2, except for the login implementation which would have already been implemented and tested.
4.2.4. Sprint 8: Push notifications

The eighth and last sprint was aimed to make store packages compatible with push notifications. In this case, a test API was developed to test push notifications without interacting directly with the production API. This test API will simulate how push notifications should work, because they are not currently supported by Intranet on all platforms. The methodology stages were as follows:

- **Analysis**: to check out how push notifications are supported and implemented by each platform and how school’s Intranet could support it.
- **Design**: to propose a test API as similar as possible to the production API and the school’s Intranet normal way of work, and to propose a solution to implement in the store packages created.
- **Implementation**: to implement the test API and push notifications in all targeted platforms.
- **Acceptance and deployment**: to have a reunion with inLabFIB to ensure the test API could be easily reproduced on the school’s Intranet API and internal functionalities in the future.

4.2.5. Initial hours estimation

<table>
<thead>
<tr>
<th>Task, sprint or stage</th>
<th>Expected hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis and feasibility study (GEP course)</td>
<td>30 hours</td>
</tr>
<tr>
<td>Sprint 1: Project initialization and browser compatibility testing</td>
<td>100 hours</td>
</tr>
<tr>
<td>Sprint 2: Login logic, first feature and application interface basis</td>
<td>250 hours</td>
</tr>
<tr>
<td>Sprint 3: Subjects information</td>
<td>20 hours</td>
</tr>
<tr>
<td>Sprint 4: School and university news</td>
<td>20 hours</td>
</tr>
<tr>
<td>Sprint 5: Personal timetable</td>
<td>20 hours</td>
</tr>
<tr>
<td>Sprint 6: Configuration settings</td>
<td>20 hours</td>
</tr>
<tr>
<td>Sprint 7: Room availability</td>
<td>40 hours</td>
</tr>
<tr>
<td>Sprint 8: Push notifications</td>
<td>40 hours</td>
</tr>
<tr>
<td>Abstract and project ending</td>
<td>100 hours</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>640 hours</strong></td>
</tr>
</tbody>
</table>

*Table 1: Expected hours of the scheduled sprints*

---

5 Push notifications are those alerts received on a smartphone, like receiving an alert when you receive a new message in WhatsApp™.
4.2.6. Actual hours accounting

<table>
<thead>
<tr>
<th>Task, sprint or stage</th>
<th>Expected hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysis and feasibility study (GEP course)</td>
<td>40 hours</td>
</tr>
<tr>
<td>Sprint 1: Project initialization and browser compatibility testing</td>
<td>250 hours</td>
</tr>
<tr>
<td>Sprint 2: Login logic, first feature and application interface basis</td>
<td>180 hours</td>
</tr>
<tr>
<td>Sprint 3: Subjects information</td>
<td>10 hours</td>
</tr>
<tr>
<td>Sprint 4: School and university news</td>
<td>10 hours</td>
</tr>
<tr>
<td>Sprint 5: Personal timetable</td>
<td>10 hours</td>
</tr>
<tr>
<td>Sprint 6: Settings</td>
<td>10 hours</td>
</tr>
<tr>
<td>Sprint 7: Room availability</td>
<td>20 hours</td>
</tr>
<tr>
<td>Sprint 8: Push notifications</td>
<td>70 hours</td>
</tr>
<tr>
<td>Abstract and project ending</td>
<td>150 hours</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>750 hours</strong></td>
</tr>
</tbody>
</table>

Table 2: Actual hours accounting

4.3. Resources

In order to carry out this project the following resources have been needed:

**Hardware**
- Laptop or desktop with Windows 8.
- MacBook Pro.
- HTC Trophy 8X or Nokia Lumia 920 and tablet with Windows 8.
- Apple iPhone 5 and Apple iPad.
- Google Nexus 4 and tablet with Android.

**Software**
- Sublime Text.
- Windows Phone SDK, Android SDK and iPhone SDK.
- Adobe Photoshop and Adobe Illustrator.
- Microsoft Office.
- Windows 8 and Mac.

**Other licenses**
- Developer license for Windows Phone and Mac.
4.4. Alternative and action plans

This project was initially focused in developing an HTML5 mobile application compatible with Windows Phone 8 and Windows 8. However, compatibility with iPhone and Android was finally considered. The scheduling has been continuously reviewed and modified.

4.5. Gantt diagram

4.5.1. Initial schedule estimation

![Gantt diagram](image1.png)

*Figure 26: Gantt diagram of the project scheduling*

4.5.2. Actual schedule

![Gantt diagram](image2.png)

*Figure 27: Actual Gantt diagram*
5. Project budget

5.1. Considerations

In order to carry out this project the resources stated above have been needed, but all of them have a price. In this document, an estimation of the cost of this project is presented taking into account respective hardware and software resources amortizations.

5.2. Budget monitoring

In order to control the budget, at the end of each sprint the budget was updated with the effective total amount of hours. Hence, the final budget is a completely real budget based on real times. Specially, at the end of the second and eighth sprints the total amount of hours widely varied.

5.3. Human resources budget

This project has been developed only by one person. Hence, this person has needed to be both a project manager and a software developer engineer, as well as a software developer engineer in test. Thus, we have to difference between each role in the total of 750 hours. In Table 4, the cost accounting is provided.

5.3.1. Initial estimation

<table>
<thead>
<tr>
<th>Role</th>
<th>Estimated hours</th>
<th>Estimated price per hour</th>
<th>Total estimated cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Manager</td>
<td>50 hours</td>
<td>50 €/hour</td>
<td>2.500,00 €</td>
</tr>
<tr>
<td>Software Developer Engineer</td>
<td>450 hours</td>
<td>35 €/hour</td>
<td>15.750,00 €</td>
</tr>
<tr>
<td>Software Developer Engineer in Test</td>
<td>140 hours</td>
<td>30 €/hour</td>
<td>4.200,00 €</td>
</tr>
<tr>
<td><strong>Total estimated</strong></td>
<td><strong>640 hours</strong></td>
<td></td>
<td><strong>22.450,00 €</strong></td>
</tr>
</tbody>
</table>

*Table 3: Initial estimation of human resources budget*

5.3.2. Actual cost

<table>
<thead>
<tr>
<th>Role</th>
<th>Hours</th>
<th>Price per hour</th>
<th>Total cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Manager</td>
<td>60 hours</td>
<td>50 €/hour</td>
<td>3.000,00 €</td>
</tr>
<tr>
<td>Software Developer Engineer</td>
<td>480 hours</td>
<td>35 €/hour</td>
<td>16.800,00 €</td>
</tr>
<tr>
<td>Software Developer Engineer in Test</td>
<td>210 hours</td>
<td>30 €/hour</td>
<td>6.300,00 €</td>
</tr>
<tr>
<td><strong>Total cost</strong></td>
<td><strong>750 hours</strong></td>
<td></td>
<td><strong>26.100,00 €</strong></td>
</tr>
</tbody>
</table>

*Table 4: Human resources budget*
5.4. Hardware budget

In order to be able to design, implement and test all applications functionalities, a set of hardware have been needed for different purposes. In Table 5, an accounting of the cost of that hardware is provided taking into account their useful life, as well as their amortizations.

<table>
<thead>
<tr>
<th>Product</th>
<th>Price</th>
<th>Units</th>
<th>Useful life</th>
<th>Total amortization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asus VivoBook S400CA</td>
<td>685,00 €</td>
<td>1</td>
<td>5 years</td>
<td>68,50 €</td>
</tr>
<tr>
<td>MacBook Pro</td>
<td>1,229,00 €</td>
<td>1</td>
<td>5 years</td>
<td>122,90 €</td>
</tr>
<tr>
<td>Nokia Lumia 925</td>
<td>599,00 €</td>
<td>1</td>
<td>3 years</td>
<td>99,83 €</td>
</tr>
<tr>
<td>Microsoft Surface</td>
<td>489,00 €</td>
<td>1</td>
<td>4 years</td>
<td>61,13 €</td>
</tr>
<tr>
<td>Apple iPhone 5</td>
<td>669,00 €</td>
<td>1</td>
<td>3 years</td>
<td>111,50 €</td>
</tr>
<tr>
<td>Apple iPad 3</td>
<td>499,00 €</td>
<td>1</td>
<td>4 years</td>
<td>62,38 €</td>
</tr>
<tr>
<td>Google Nexus 4</td>
<td>299,00 €</td>
<td>1</td>
<td>3 years</td>
<td>49,83 €</td>
</tr>
<tr>
<td>Google Nexus 10</td>
<td>399,00 €</td>
<td>1</td>
<td>4 years</td>
<td>49,88 €</td>
</tr>
<tr>
<td><strong>Total cost</strong></td>
<td><strong>4,868,00 €</strong></td>
<td><strong>1</strong></td>
<td><strong>625,94 €</strong></td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Hardware budget

5.5. Software budget

Additionally, some software products have been needed to carry out the project. Although some of them are available for free as this is an academic project, the real cost is considered. As in the hardware budget, their amortizations have been taken into account. In Table 6 the software budget is shown.

<table>
<thead>
<tr>
<th>Product</th>
<th>Price</th>
<th>Units</th>
<th>Useful life</th>
<th>Total amortization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visual Studio 2012 Professional</td>
<td>615,00 €</td>
<td>1</td>
<td>4 years</td>
<td>76,88 €</td>
</tr>
<tr>
<td>XCode</td>
<td>0,00 €</td>
<td>1</td>
<td>N/A</td>
<td>0,00 €</td>
</tr>
<tr>
<td>ADT</td>
<td>0,00 €</td>
<td>1</td>
<td>N/A</td>
<td>0,00 €</td>
</tr>
<tr>
<td>Sublime Text</td>
<td>53,45 € (70$)</td>
<td>1</td>
<td>4 years</td>
<td>6,68 €</td>
</tr>
<tr>
<td>Windows Phone SDK</td>
<td>0,00 €</td>
<td>1</td>
<td>N/A</td>
<td>0,00 €</td>
</tr>
<tr>
<td>iPhone SDK</td>
<td>0,00 €</td>
<td>1</td>
<td>N/A</td>
<td>0,00 €</td>
</tr>
<tr>
<td>Android SDK</td>
<td>0,00 €</td>
<td>1</td>
<td>N/A</td>
<td>0,00 €</td>
</tr>
<tr>
<td>Adobe Photoshop</td>
<td>24,59 €/month</td>
<td>1</td>
<td>6 months</td>
<td>147,54 €</td>
</tr>
<tr>
<td>Adobe Illustrator</td>
<td>24,59 €/month</td>
<td>1</td>
<td>6 months</td>
<td>147,54 €</td>
</tr>
<tr>
<td>Microsoft Office 365 Home Premium</td>
<td>9,99 €/month</td>
<td>1</td>
<td>6 months</td>
<td>59,94 €</td>
</tr>
<tr>
<td>Windows 8 Professional</td>
<td>69,99 €</td>
<td>1</td>
<td>3 years</td>
<td>Included on Asus VivoBook</td>
</tr>
<tr>
<td><strong>Total cost</strong></td>
<td><strong>1,023,47 €</strong></td>
<td><strong>438,58 €</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Software budget
5.6. Other licenses budget

Finally, other licenses have been needed in order to carry out the project. For instance, a repository was necessary for saving the code securely (and saving different versions), as well as a test server for the test API. Furthermore, in order to develop for some platforms a developer license was needed, as shown in Table 7.

<table>
<thead>
<tr>
<th>Product</th>
<th>Price</th>
<th>Units</th>
<th>Useful life</th>
<th>Total amortization</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitHub premium service (private repositories)</td>
<td>6,10 € (7,99$)/month</td>
<td>1</td>
<td>6 months</td>
<td>36,60 €</td>
</tr>
<tr>
<td>Windows Phone Developer License</td>
<td>75,68 € (99$)/year</td>
<td>1</td>
<td>1 year</td>
<td>37,84 €</td>
</tr>
<tr>
<td>Apple Developer License</td>
<td>75,68 € (99$)/year</td>
<td>1</td>
<td>1 year</td>
<td>37,84 €</td>
</tr>
<tr>
<td><strong>Total cost</strong></td>
<td><strong>224,56 €/year</strong></td>
<td></td>
<td></td>
<td><strong>112,28 €</strong></td>
</tr>
</tbody>
</table>

*Table 7: Other licenses budget*

5.7. Total budget

By adding all the budgets provided above, we get the total estimated budget for this project, as shown in Table 9.

5.7.1. Initial estimation

<table>
<thead>
<tr>
<th>Concept</th>
<th>Estimated cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>625,94 €</td>
</tr>
<tr>
<td>Software</td>
<td>438,58 €</td>
</tr>
<tr>
<td>Other licenses</td>
<td>112,28 €</td>
</tr>
<tr>
<td>Human resources</td>
<td>22,450,00 €</td>
</tr>
<tr>
<td><strong>Total estimated cost</strong></td>
<td><strong>23,626,80 €</strong></td>
</tr>
</tbody>
</table>

*Table 8: Initial estimation of total budget*

5.7.2. Actual cost

<table>
<thead>
<tr>
<th>Concept</th>
<th>Total cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>625,94 €</td>
</tr>
<tr>
<td>Software</td>
<td>438,58 €</td>
</tr>
<tr>
<td>Other licenses</td>
<td>112,28 €</td>
</tr>
<tr>
<td>Human resources</td>
<td>26,100,00 €</td>
</tr>
<tr>
<td><strong>Total cost</strong></td>
<td><strong>27,276,80 €</strong></td>
</tr>
</tbody>
</table>

*Table 9: Total budget*
6. Social and environmental concerns

6.1. Social concerns

Being analyzed the social impact of this project at the beginning of the definition phase, it is clear now that this application will have a major impact on how students check their marks or other notes from their subjects, as well as they find an available room or check their timetable.

inLabFIB has shown its interest in the application providing new features within the public API, for two reasons:

- This application will provide a better end-user experience than the existing ones, providing more useful features (like downloading the attachments) and being compatible with more platforms (Windows Phone, Windows 8 and any other one that can be added in the future).

- It will also provide a better solution for them: one app, any device. They will be able to maintain just one app and any change will be applicable to every supported platform. Just writing it once.

Hence, this project will make a major impact in two different kind of people related with the school.

6.2. Environmental concerns

This project does not have important environmental concerns, because it is not bringing new information or features to Intranet users that can affect it.

However, if we consider it is also a way to access personal information for students and teachers, this project contributes to the paperless objective of the Intranet, bringing information to final users from the distance without the need to use paper. Hence, it contributes to preserve the environment.
7. Software architecture

7.1. Conceptual schema

In order to design the application we must first understand the key concepts of the Intranet. In the Barcelona School of Informatics, students or teachers (users) are enrolled or teach subjects, respectively. Those subjects have notes, and a timetable. Hence, a user might have a timetable assigned.

Also, there are different buildings within the campus and each of them has its own rooms. Both the school and the university publish news regularly.

The domain stated above is simplify represented in the following conceptual schema:

![Conceptual schema](image)

Figure 28: Conceptual schema

These facts— and others not stated here— are part of the definition of the school’s Intranet and it is out of scope of this project to discuss about them. However, we must take them into account in order to design the architecture of our application appropriately.
7.2. Frameworks used

To carry out this project we needed to use and combine many existing Javascript frameworks, all of them free for commercial use:

- Backbone.js: used to architecture the application following the Model-View-Controller pattern[12].
- Underscore.js: used to deal with data conversion and treatment in Javascript[13].
- jQuery: used to simplify DOM selection operations.
- Require.js: used to load modules asynchronously in order to save as much memory as possible[14].
- Apache Cordova (commercially known as PhoneGap): used to package the web application within a native one for each targeted platform.
- WinJS: used to access WinRT APIs on Windows 8.

7.3. Design schema

7.3.1. Packages

This application has been designed following the Model-View-Controller (MVC) pattern. Hence, the main packages are Controllers, Models and Views. However, other ones such as Templates, Helpers and Collections have been added.

- Controllers: contains the logic controllers of the application.
- Models: contains the models that represents entities within the application.
- Views: contains the views of the application,
- Templates: contains the templates rendered by the views.
- Helpers: contains the helpers used within the application, such as server communication and/or offline synchronization.
- Collections: contains the collections (a set of models) represented within the application.
Figure 29: Packages diagram
7.3.2. Controllers

Login

```java
<<singleton>>
LoginController

currentUser: User
initialize(): void
isLoggedIn(): boolean
login(): void
logout(): ... (): void

OAuthController

static config: OAuthConfiguration
oAuthService: OAuthService
accessToken: OAuthAccessToken
initialize(): void
checkSavedAccessToken(): void
isAuthenticated(): boolean
connect(): void
logout(): void
```

News

```java
<<singleton>>
NewsController

upcNews: NewsList
fibNews: NewsList
upcNewsLatestSync: DateTime
fibNewsLatestSync: DateTime
initialize(): void
fetchNewsAsync(): void
getNewsAsync(): void
getFIBNewsAsync(): void
getUPCNewsAsync(): void
openExternal(url): void
```

Notes

```java
<<singleton>>
NotesController

latestNotes: NoteList
latestSync: DateTime
initialize(): void
fetchLatestNotesAsync(): void
getLatestNotesAsync(): void
openAttachment(id, subject, name): void
markNoteAsRead(id): void
```

Figure 30: Login, news and notes packages diagram
Figure 31: Notifications, rooms and settings packages diagram
Subjects

```
<<singleton>>
SubjectsController
subjects: SubjectList
latestSync: DateTime
initialize(): void
fetchSubjectsAsync(): void
getSubjectsAsync(): void
```

Timetable

```
<<singleton>>
TimetableController

timetable: Timetable
subjects: SubjectList
latestSync: null
initialize(): void
fetchTimetableAsync(): void
getTimetableAsync(): void
```

Figure 32: Subjects and timetable packages diagram

7.3.3. Models

All models extends Backbone.Model to get its data binding properties and methods. In order to simplify the diagrams of this document we avoid hierarchy indicators.

Figure 33: Backbone model package diagram
Developing an HTML5 mobile application for Barcelona School of Informatics’ Intranet

Figure 34: Application models diagram
7.3.4. Views/Templates

Each view has its own template written in HTML. Every time the application renders a view, it compiles the template and shows it to the user. While the view contains the logic itself, the template contains just its visual aspect.

All views extend Backbone.View to get its render properties and methods. However, in order to provide a common functionality to some views, we have also created a BaseView class. At the end, all views follow the same structure.

Figure 35: Application and login view packages diagram
Developing an HTML5 mobile application for Barcelona School of Informatics’ Intranet

Figure 36: News and notes views packages diagram
Figure 37. Rooms, settings and subjects view packages diagram
7.3.5. Helpers

**Dispatcher**

- beginInvoke(delegate): void
- beginInvoke(delegate, delay): void

**MobileDetector**

- getMobileOS(): string
- isNativeApp(): boolean
- isWindowsPhone(): boolean
- isWindows(): boolean
- isAndroid(): boolean
- isiOS(): boolean

**Application**

- setMenuActiveElement(element): void
- setActiveElements(): void

**Data**

- stringToXml(string): XMLDocument
- xmlToJson(xml): JSONObject

**HttpClient**

- loadingLayer: HTMLElement
- cancelLoadingHide: boolean
- loadingIsGoingToHide: boolean
- signUrl(url): Uri
- getSignedAsync(url, silent): Object
- postAsync(url): Object
- getAsync(url): Object
- readStreamAsync(url): Object
- putSignedAsync(url): Object
- showLoading(): void
- hideLoading(): void

**Environment**

- isNativeApp(): boolean
- log(message): void
- showView(content, element): void
- showConfirmationDialogAsync(message, title): void
- getApplicationLanguage(): Language

Figure 38: Timetable view package diagram

Figure 39: Helpers diagram
7.3.6. Collections

All our collections extend from Backbone.Collection to inherit its data binding properties and methods. In order to avoid understandable diagrams, we omit hierarchy symbols here.

![Backbone Collections Diagram]

**Figure 40: Application collections diagram**

7.4. Understanding object-oriented programming in Javascript

Javascript is an object-based language based on prototypes, rather than being class-based[11]. Because of this different basis, it can be less apparent how Javascript allows you to create hierarchies of objects and to have inheritance of properties and their values.

On the one hand, class-based object-oriented languages, such as Java and C#, are founded on the concept of two distinct entities: classes and instances.

- A *class* defines all of the properties (considering methods and fields in Java, or members in C++, to be properties) that characterize a certain set of objects. A class is an abstract thing, rather than any particular member of the set of objects it describes. For example, the Employee class could represent the set of all employees.
- An instance, on the other hand, is the instantiation of a class; that is, one of its members. For example, Victoria could be an instance of the Employee class, representing a particular individual as an employee. An instance has exactly the properties of its parent class (no more, no less).

On the other hand, a prototype-based language, such as Javascript, does not make this distinction: it simply has objects. A prototype-based language has the notion of a prototypical object, an object used as a template from which to get the initial properties for a new object. Any object can specify its own properties, either when you create it or at run time. In addition, any object can be associated as the prototype for another object, allowing the second object to share the first object's properties.

### 7.4.1. Javascript closures

Languages such as Java provide the ability to declare methods private, meaning that they can only be called by other methods in the same class.

Javascript does not provide a native way of doing this, but it is possible to emulate private methods using closures[10]. Private methods are not just useful for restricting access to code: they also provide a powerful way of managing your global namespace, keeping non-essential methods from cluttering up the public interface to your code.

Using closures in this way provides a number of benefits that are normally associated with object oriented programming, in particular data hiding and encapsulation. A closure in Javascript is a function which contains a set of methods and attributes that automatically executes itself to create an object completely separated from the global scope.

### 7.4.2. Deferred objects and promises

A callback is a piece of executable code that is passed as an argument to other code, which is expected to execute the argument at some convenient time. In other words, a function can be passed as an argument to another function to be executed when it is called.

There is nothing inherently wrong with callbacks, but depending on which environment we are programming in there are a number of options available for managing asynchronous events. One of the keys to effectively working with asynchronous events in Javascript is understanding that the program continues execution even when it doesn’t have the value it needs for work that is in progress. Dealing
with as yet unknown values from unfinished work can make working with asynchronous events in Javascript challenging.

Considered from a high level, promises in Javascript give us the ability to write asynchronous code in a parallel manner to synchronous code[15]. A promise is a placeholder for a result which is initially unknown while a deferred represents the computation that results in the value. Every deferred has a promise which functions as a proxy for the future result. While a promise is a value returned by an asynchronous function, a deferred can be resolved or rejected by its caller which separates the promise from the resolver. The promise itself can be given to any number of consumers and each will observe the resolution independently meanwhile the resolver / deferred can be given to any number of producers and the promise will be resolved by the one that first resolves it. From a semantic perspective this means that instead of calling a function (callback), we are able to return a value (promise).

7.5. Data binding

Backbone.js gives structure to web applications by providing models with key-value binding and custom events, collections with a rich API of enumerable functions, views with declarative event handling, and connects it all to your existing API over a RESTful JSON interface. Backbone does this by providing just four primary concepts: Models, Views, Collections, and the Router.

Backbone Views are extremely simple: they only provide a bare minimum framework. Backbone asks a lot of the developer to fill in the holes (how to render a view, how to update the data in the view, how to handle events, etc.). While the simple Models and Collections are usable out of the box, Views have to be adapted to work in your environment.

Data binding is the process by which application data (usually coming from the application’s REST API and stored in a Model) is bound to the view of the application. This makes write-only data binding (that is, writing to the DOM) very easy, just by subscribing to change or reset events on Backbone Collections. In this project, these actions are always done within the `bindEvents` function on each View.

7.6. Hash-driven navigation

Hash-driven navigation is a web programming technique to create in-page navigations without the need of the browser to reload the page. This technique is commonly used in web applications which changes
its contents depending on the value of the hash (the part following the hash symbol in a URL). Backbone.js routed navigation uses this method to detect which page needs to be rendered every time.

Our application router contains the definition of all functions required to render all possible hash coming from the menu, such as:

- `#!/latestNotes`: will render the latest notes view.
- `#!/latestNotes/1234`: will render the latest note view with the model (Note) identified by 1234.
- `#!/timetable`: will render the timetable view.
- `#!/rooms`: will render the room availability view.
- `#!/settings`: will render the settings view.
- And so on.

### 7.7. Call sequence

According to the architecture of the application, all views, calls and methods are named exactly the same regardless of which they are. Hence, once we see how a full interaction is handled we can immediately extrapolate this information to any other interaction within the application. For this reason, we will only focus on explaining what happens since the user selects the ‘Latest notes’ option from the menu until they become visible for him/her.

![Interaction and location change diagrams](image-url)
7.8. Data persistence

With HTML5, web pages can store data locally within the user’s browser. Earlier, this was done with cookies. However, Web Storage is more secure and faster. The data is not included with every server request, but used only when asked for. It is also possible to store large amounts of data, without affecting the website’s performance. The data is stored in key/value pairs, and a web page can only access data stored by itself.

Web storage is supported in Internet Explorer 8+, Firefox, Opera, Chrome, and Safari. There are two new objects for storing data on the client:

- Local storage: stores data with no expiration date.
- Session storage: stores data for one session.

This application uses the local storage in order to persist user data such as latest notes, subjects and so on. Hence, once the user opens the application for the second time and further, we can show him/her previously retrieved data while syncing.

### 7.9. OAuth 1.0

OAuth is an open standard for authorization. It provides a method for clients to access server resources on behalf of a resource owner (such as a different client or an end-user). It also provides a process for end-users to authorize third-party access to their server resources without sharing their credentials (typically, a username and password pair), using user-agent redirections.

In the traditional client-server authentication model, the client uses its credentials to access its resources hosted by the server. OAuth introduces a third role to this model: the resource owner. In the OAuth model, the client (which is not the resource owner, but is acting on its behalf) requests access to resources controlled by the resource owner, but hosted by the server.

In order for the client to access resources, it first has to obtain permission from the resource owner. This permission is expressed in the form of a token and matching shared-secret. The purpose of the token is to make it unnecessary for the resource owner to share its credentials with the client. Unlike the resource owner credentials, tokens can be issued with a restricted scope and limited lifetime, and revoked independently.

OAuth is used in this project to authenticate the user with the Barcelona School of Informatics' Intranet. Hence, user credentials are never shared with the application and there is no risk for the user to have its data stolen. Unless it is not the purpose of this document to explain how the OAuth 1.0a protocol works, more information can be found in the project references.

### 7.10. Web encapsulation

This application is purely and entirely a web application. It can be accessed from any browser and you can authenticate, as well as navigate through it. However, the real purpose of it is to be a mobile application that users can install on their devices and use it without data connection if needed.
For this reason, one of the most important parts of this project is the web encapsulation. This term describes how a web application is packaged within a native application for each platform, providing a transparent user experience for the end-user.

Web encapsulation consists on a native project for each platform, which contains a unique main view with a Browser element on it. This browser elements is the one that shows the web application as if the user were actually navigating from the phone’s browser native application. However, this way the application can access native APIs provided by Apache Cordova (commercially known as PhoneGap).

![Figure 44: Packaged web application architecture](image)

**7.11. Push notifications**

Push notifications are one of the key factors of this project. As from now they are only supported in the Android application, but this new one introduces them to all targeted platforms. Thanks to this feature, users can get notifications initiated by the server when a new note has been published, providing a new and exciting way for university to communicate with their students.

In this section, push notifications workflow is explained for each supported platform.
7.11.1. Windows Phone: Microsoft Push Notification Service

The Microsoft Push Notification Service in Windows Phone is an asynchronous, best-effort service that offers third-party developers a channel to send data to a Windows Phone app from a cloud service in a power-efficient manner[6].

The following diagram shows how a push notification is sent:

1. The app requests a push notification URI from the Push client service.
2. The Push client service negotiates with the Microsoft Push Notification Service (MPNS), and MPNS returns a notification URI to the Push client service.
3. The Push client service returns the notification URI to the app.
4. The app can then send the notification URI to our cloud service.
5. When our cloud service has info to send to the app, it uses the notification URI to send a push notification to MPNS.
6. MPNS routes the push notification to the app.

Depending on the format of the push notification and the payload attached to it, the info is delivered as raw data to the app, the app’s Tile is visually updated, or a toast notification is displayed. MPNS returns a response code to your cloud service after a push notification is sent indicating that the notification has been received and will be delivered to the device at the next possible opportunity. Although MPNS doesn’t provide an end-to-end confirmation that your push notification was delivered from your cloud service to the phone, it is possible for MPNS to return a response or error code to the cloud service which indicates that the notification will not be delivered to the device.
7.11.2. Windows 8: Windows Push Notification Services

The Windows Push Notification Services (WNS) enables third-party developers to send toast, tile, badge, and raw updates from their own cloud service[7]. This provides a mechanism to deliver new updates to your users in a power-efficient and dependable way.

The following diagram shows the complete data flow involved in sending a push notification. It involves these steps:

1. The app sends a request for a push notification channel to the Notification Client Platform.
2. The Notification Client Platform asks WNS to create a notification channel. This channel is returned to the calling device in the form of a Uniform Resource Identifier (URI).
3. The notification channel URI is returned by Windows to the app.
4. The app sends the URI to your own cloud service. This callback mechanism is an interface between our own app and our own service. It is your responsibility to implement this callback with safe and secure web standards.
5. When our cloud service has an update to send, it notifies WNS using the channel URI. This is done by issuing an HTTP POST request, including the notification payload, over Secure Sockets Layer (SSL). This step requires authentication.
6. WNS receives the request and routes the notification to the appropriate device.

Figure 46: WNS workflow
7.11.3. Android: Google Cloud Messaging

Google Cloud Messaging for Android (GCM) is a service that helps developers send data from servers to their Android applications on Android devices[8]. Using this service one can send data to his/her application whenever new data is available instead of making requests to server in timely fashion. Integrating GCM in an android application enhances user experience and saves lot of battery power.

This is how it works:

1. The Android device sends its sender and application identifiers to GCM server for registration.
2. Upon successful registration, GCM server issues a registration identifier and sends it to the device.
3. After receiving the registration identifier, the device will send it to our server.
4. Our server will store the registration identifier in the database for later usage.
   
a. Whenever a push notification is needed, our server sends a message to GCM server along with the device registration identifier (which was stored earlier in the database).
   
b. GCM server will deliver this message to the corresponding device using the given registration identifier.

Figure 47: GCM for Android workflow
7.11.4. iOS: Apple Push Notification Service

Apple Push Notification service (APNs for short) is the centerpiece of the push notifications feature. It is a robust and highly efficient service for propagating information to iOS and OS X devices[9]. Each device establishes an accredited and encrypted IP connection with the service and receives notifications over this persistent connection. If a notification for an application arrives when that application is not running, the device alerts the user that the application has data waiting for it.

Software developers originate the notifications in their server software. The provider connects with APNs through a persistent and secure channel while monitoring incoming data intended for their client applications. When new data for an application arrives, the provider prepares and sends a notification through the channel to APNs, which pushes the notification to the target device.

In addition to being a simple but efficient and high-capacity transport service, APNs includes a default quality-of-service component that provides store-and-forward capabilities.

The following diagram shows how it works:

Figure 48: APNs workflow
8. Testing

One of the most complicated parts of this project has been testing all features on all supported platforms. Unfortunately, there is currently no way to write automation for web applications packaged within a native one on all targeted platforms.

In order to manually test this project, a checklist was created to review after completing each sprint:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Expected result</th>
<th>Actual result</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Latest notes</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select ‘Latest notes’ in the menu</td>
<td>The latest notes are shown to the user. If there are no notes available, a message is shown.</td>
<td>□ Success □ Fail □ N/A</td>
</tr>
<tr>
<td>Select an already read note from the latest notes list.</td>
<td>The details of that note are shown.</td>
<td>□ Success □ Fail □ N/A</td>
</tr>
<tr>
<td>Select an attachment from a note.</td>
<td>The attachment is downloaded and shown to the user without the need to authenticate again.</td>
<td>□ Success □ Fail □ N/A</td>
</tr>
<tr>
<td>Select an unread note from the latest notes list.</td>
<td>The note has been marked as read and if we go back we do not see it highlighted anymore.</td>
<td>□ Success □ Fail □ N/A</td>
</tr>
<tr>
<td><strong>Subjects</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select ‘Subjects’ in the menu</td>
<td>The subjects list is shown to the user. If there are no subjects available, a message is shown.</td>
<td>□ Success □ Fail □ N/A</td>
</tr>
<tr>
<td>Select a subject from the list</td>
<td>The details of the subject are shown, including its teachers and notes. If there are no teachers of notes, a message is shown.</td>
<td>□ Success □ Fail □ N/A</td>
</tr>
<tr>
<td>Select a note from the subject notes list.</td>
<td>The details of the subject note are shown.</td>
<td>□ Success □ Fail □ N/A</td>
</tr>
<tr>
<td><strong>Timetable</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select ‘Timetable’ in the menu</td>
<td>The timetable for the user is shown. If no timetable is available at the current time, a message is shown. If the current device is a phone, the view is the current day with two arrows, one to go back and another one to go forward in the week. If the current device is a tablet and its current orientation is landscape, the full week is shown.</td>
<td>□ Success □ Fail □ N/A</td>
</tr>
</tbody>
</table>
## Rooms

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Success</th>
<th>Fail</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select 'Rooms' in the menu</td>
<td>The room availability information for all buildings is shown. If this information is not available at the current time, a message is shown.</td>
<td>☐ Success ☐ Fail ☐ N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select a room from a building.</td>
<td>The details for that room are shown, including the daily schedule. If details are not available at that moment, a message is shown.</td>
<td>☐ Success ☐ Fail ☐ N/A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## News

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Success</th>
<th>Fail</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select 'News' in the menu</td>
<td>The news of FIB and UPC are shown. If there are no news available at the current time, a message is shown.</td>
<td>☐ Success ☐ Fail ☐ N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select a new from the list.</td>
<td>The details of the new are shown.</td>
<td>☐ Success ☐ Fail ☐ N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select 'continue reading' option</td>
<td>The application opens the link externally for the user to continue reading the new.</td>
<td>☐ Success ☐ Fail ☐ N/A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Settings

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Success</th>
<th>Fail</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select 'Settings' from the list</td>
<td>Settings view is shown.</td>
<td>☐ Success ☐ Fail ☐ N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select an initial page</td>
<td>Next time the application is opened the initial page is the one selected.</td>
<td>☐ Success ☐ Fail ☐ N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Toggle push notifications</td>
<td>If push notifications have been enabled, a channel or device identifier is shown in the debug console. Otherwise, if they have been disabled, a confirmation message is shown in the debug console.</td>
<td>☐ Success ☐ Fail ☐ N/A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Authentication

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Success</th>
<th>Fail</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select 'Connect with your account'</td>
<td>The login dialog is displayed. Once the user authorizes the application, the latest notes view is shown. If user does not allow the application, an error message is displayed.</td>
<td>☐ Success ☐ Fail ☐ N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select 'Sign out' in the menu</td>
<td>A confirmation dialog appears. If the user selects ‘Yes’, the current session is closed and the initial page is shown. If user selects ‘No’, there is no further action.</td>
<td>☐ Success ☐ Fail ☐ N/A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Language

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Success</th>
<th>Fail</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change the device language</td>
<td>The application has also changed its language.</td>
<td>☐ Success ☐ Fail ☐ N/A</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
9. Usability and user experience

The same mobile applications is different on each platform, because each of them has its own design and interaction guidelines. It is important to follow them in order to help the user understand how the application works without the need to provide additional instructions.

One of the most important part of this project has been to provide a great user experience and a usable user interface. Even though this application has been developed in HTML5, the Cascade Style Sheets (CSS) have been adapted to each platform, in order to make it feel a native application as much as possible.

9.1. iOS adaptations

iOS devices does not have a hardware back button so all applications must have its own at the top. For this reason, I have added a menu button that changes to a back one when you navigate to a secondary or tertiary depth.

Moreover, it is also recommended by Apple that applications with a lot of sections uses a side menu visible only by the action of the user.
Finally, Apple also recommends that all applications use the system font, in order to provide uniformity across all applications installed by the user.
9.2. Android adaptations

Android has less uniformity than other platforms because there are a lot of manufacturers that are able to customize the systems as they want. Hence, it is more difficult to provide uniformity across all applications.

However, Google provides official guidelines for Android developers and I have adapted some details of the application to make it feel as native as possible.

For instance, the side menu has a different slide effect than other platforms, super-positioning to other elements instead of displacing them.

*Figure 52: Android KitKat lateral menu used in the application*
Moreover, the action buttons are located on the right side of the top bar, leaving the left one for the page title and back button if applicable, depending on the page or section the user is currently located.

![Android application header](image)

**Figure 53: Android application header**

### 9.3. Windows Phone 8 and Windows 8 adaptations

Windows Phone 8 and Windows 8 follow a design language called Modern UI (formerly known as Metro), created by Microsoft Corporation to highlight information instead of details across all types of devices. It is characterized by squares and rectangles, containing as less information as possible.

![Windows 8 and Windows Phone 8 application bars with commands](image)

**Figure 54: Windows 8 and Windows Phone 8 application bars with commands**

As we can see, action buttons are placed on the application bar, while other platforms have them at the top bar. This is an exclusive characteristic of the Modern UI style.
10. Justification of decisions

10.1. Removing the calendar feature

At the beginning of the project, the calendar feature was considered to be added to the application in order to allow users to see their exams and other events from their subjects and/or general school events. However, since this feature is not widely used by users and it does not provide too many information, we have agreed with inLabFIB to remove it because users can already add it to their existing calendar application on their phone from the Intranet’s website. This change has a minor impact in the usefulness of the application.

10.2. Adding iOS and Android support

This project was initially aimed to develop an HTML5 application compatible with Windows Phone 8 and Windows 8, because there are existing applications for Android and iOS. However, since these two platforms are WebKit based and they have a high compatibility with the newest HTML5 features, native packages have also been provided for these platforms.

10.3. Push notifications support on all platforms

Although push notifications are not supported on all platforms by the Intranet’s API service, I considered all native packages created in this project must have support to receive them. For this reason, I created a test API which can send push notifications to any supported platform, simulating a new note has been published on a subject, for example.

Once this project is delivered to inLabFIB, it is their decision to implement push notifications on all platforms by adding support on the Intranet’s API service. This issue is out of the scope of this project.

10.4. Inability to change application language at run time

Even though it is possible for the user to select the display language on the Intranet’s website, I considered this feature should not be applicable to any of the supported platforms. Both Google and Apple, as well as Microsoft, recommends that the application must use the phone language, or the default one accordingly (if the current language is not supported). Hence, this application supports Spanish, Catalan and English and will show one or another depending on the currently selected system language.
11. Conclusions

Being finished the project, we can extract the following conclusions:

- An HTML5 application has been developed, which is compatible with Windows Phone 8, Windows 8, Android 4.0+ and iOS 7 - the most used mobile platforms as of January, 2014.

- It meets all the requirements stated at the beginning of the project (both the functional and non-functional ones).

- It supports the most common features of the Intranet, such as latest notes, subjects, room availability, timetable and news.

- Push notifications are now supported by all targeted platforms.

- The new application adapts itself to the platform it is currently being executed at, and this offers a better user experience for students and teachers.

- The design of the application is simple and less-cluttered. It brings the real information to the front.

- The new design supports the color schema of the Intranet's website.

- It is available in Catalan, Spanish and English.

- With this application, inLabFIB can add new features just developing them once and deploying them everywhere.
12. Improvements for the future

Although the developed application improves the user experience and usefulness of the application, there is still a lot of work to do to make it better. Following there are some examples:

- The Android version must implement the new side menu introduced by the newest version known as KitKat, which has the button to open the menu near the application title.

- Both iOS and Android versions should implement “pull to refresh” method to reload information, instead of adding a refresh button.

- Windows Phone version should use a Hub to present information when this is supported in Windows Phone 8.1.

- Windows 8 version should also add desktop features such as subject deliveries.

- This application could also be ported to Linux and Mac OS X (desktop versions).

- Android design should be updated to use the system colors and elements.

- New push notifications should be added (final thesis status changes, general advertisements and so on).

- New features could be added, such as:
  - Ability for teachers to publish a new note directly from the application.
  - Ability to sync the calendar with the phone directly from the app.
  - Ability to find a classmate within the campus.
  - Ability to reserve a computer or a laptop.
  - Ability to check the remaining quotes.

- inLabFIB might publish the application.
References

Retrieved on June 25th, 2013 from EdTech Magazine:
http://www.edtechmagazine.com/higher/article/2013/03/13-impressive-statistics-about-mobile-device-use

Retrieved on June 25th, 2013 from UPC Commons:
http://upcommons.upc.edu/pfc/bitstream/2099.1/13942/1/77788.pdf

Retrieved on June 25th, 2013 from UPC Commons:
http://upcommons.upc.edu/pfc/bitstream/2099.1/13957/1/77954.pdf

Retrieved on June 25th, 2013 from EmergencyEdTech:
http://www.emergingedtech.com/2013/04/6-higher-education-institutions-leading-the-way-with-mobile-learning-apps/

[5] Ortega, C. (March, 2013). A survey carried out to Barcelona School of Informatics’ students and staff has been added to this document as an annex below.

Retrieved on January 8th, 2014 from Windows Phone Dev Center:

Retrieved on January 8th, 2014 from Windows Dev Center:

Developing an HTML5 mobile application for Barcelona School of Informatics’ Intranet

Retrieved on January 9th, 2014 from Android Developers page:


Retrieved on January 10th, 2014 from iOS Developer Library:


Retrieved on October 10th, 2013 from Mozilla Developer Network:


Retrieved on January 20th, 2014 from Mozilla Developer Network:


Retrieved on January 22nd, 2014 from Backbone.js Official Docs:
http://backbonejs.org/


Retrieved on January 23rd, 2014 from Underscore.js Official Docs:
http://underscorejs.org/


Retrieved on January 20th, 2014 from Require.js Official Docs:
http://requirejs.org/


Retrieved on January 18th, 2014 from Chris Webb:
Appendix I. Application screenshots

Appendix I.I. iOS

Appendix I.I.I. Phone

Figure 55: iPhone lateral menu and latest notes view

Figure 56: iPhone note detail and subjects list view
Developing an HTML5 mobile application for Barcelona School of Informatics’ Intranet

Figure 57: iPhone news views

Google Europe Scholarship for Students with Disabilities
a day ago

The Google Anita Borg Memorial Scholarship
a day ago

Application period to FIB Master’s programmes
a day ago

A4ve 2014 IT

Figure 58: iPhone rooms list and timetable views

Google Europe Scholarship for Students with Disabilities
a day ago

Open the period for submission to the Google Europe Scholarship for Students with Disabilities for the academic year 2014-2015.

Continue reading »
Appendix I.I.II. Tablet

Figure 59: iPad latest notes view

Figure 60: iPad note details view
Developing an HTML5 mobile application for Barcelona School of Informatics’ Intranet

```
Figure 61: iPad subjects list view

Figure 62: iPad subject details view
```
Figure 63: iPod news view

Figure 64: iPod new details view
Building A5

Figure 65: iPad rooms list

Figure 66: iPad timetable view
Appendix I.II. Android

Appendix I.II.I. Phone

Figure 67: Android phone lateral menu and latest notes view

Figure 68: Android phone note details and subjects list view
Developing an HTML5 mobile application for Barcelona School of Informatics’ Intranet

Figure 69: Android phone subject details and news list views

Figure 70: Android phone new details and rooms list views
Figure 71: Android phone timetable views

Figure 72: Android phone push notifications overview
Appendix I.II. Tablet

Figure 73: Android tablet latest notes view

Figure 74: Android tablet note details view
Figure 75: Android tablet subjects list view

Figure 76: Android tablet subject details view
Developing an HTML5 mobile application for Barcelona School of Informatics’ Intranet

Figure 77: Android tablet news view

Google Europe Scholarship for Students with Disabilities
17 hours ago

The Google Anita Borg Memorial Scholarship
17 hours ago

Application period to FIB Master’s programmes
17 hours ago

Atos 2014 IT Challenge
17 hours ago

UPC nanoSat Lab develops nanosatellite for launch in 2014

Figure 78: Android tablet new details view

Google Europe Scholarship for Students with Disabilities
17 hours ago

Open the period for submission to the Google Europe Scholarship for Students with Disabilities for the academic year 2014-2015.

Continue reading

Google
Figure 79: Android tablet rooms list view

Figure 80: Android tablet timetable view
Appendix I.III. Windows

Appendix I.III.I. Phone

Figure 81: Windows Phone lateral menu and latest notes view

Figure 82: Windows Phone note details and subjects list views
Figure 83: Windows Phone news views

Figure 84: Windows Phone rooms list and timetable views
Developing an HTML5 mobile application for Barcelona School of Informatics’ Intranet

Figure 85: Windows Phone timetable view with toggle arrows and push notifications overview

Appendix I.III.II. Tablet

últimos avisos

Figure 86: Windows 8 latest notes view full-size
últimos avisos

Módulo 3
Hace 11 horas en GRAS-125

ARCHIVOS ADJUNTOS (0)
1. Gestores de referencias (Módulo 3.2.2).pdf
2. Habilidades informacionales (Módulo 3.3.3).pdf
3. Módulo 3.1 - Dirección de personas y equipos.pdf
4. Módulo 3.3 - Habilidades interculturales.pdf
5. Módulo 3.3 - Técnicas de comunicación eficientes.pdf
6. Módulo 3 - Temario.pdf
7. Taller 4 - Presentación preliminar.pdf
8. Taller 5 - Contexto y referencias.pdf

Figure 87: Windows 8 note details view full-size

asignaturas

AsPECTOS SOCIALS - mediambientals de la informàtica
UBAC-155

CONCEPTEES PER A BASES DE DADES ESPECIALITZADAS
UBAC-135

PROIECTES D'EMPRESA DE SOFTWARE
UBAC-136

SIMULACIÓ
UBAC-137

VALORATGE DE PROIECTES EMPRENDITORS
UBAC-138

GESTIÓ DE PROIECTES
UBAC-139

Figure 88: Windows 8 subject list view full-size
Developing an HTML5 mobile application for Barcelona School of Informatics’ Intranet

Asignaturas

Aspectos Sociales y Medioambientales de la Informática

Últimos avisos

PROFESORES
Esta asignatura no tiene ningún profesor asignado.

Módulo 1
Horario

Módulo 2
Horario

Información general del módulo transversal (1, 2, 3)

Módulo 3
Horario

Notas

1.1.2.1.1

Figure 89: Windows 8 subject details view full-size

noticias

Noticias FIB

Seminar de empresa 2014
Horario

Horario de los Servicios Administrativos durante el periodo de Fiestas
Horario

Microsoft te invita al cine
Horario

Pruebas CLIC enero 2014
Horario

Noticias UPC

Publicación de la primera resolución parcial de los becas DSA para el curso 2013-2014
Horario

Estudiantes de la UPC, mentores de alumnos de primaria y secundaria a través del
Horario

Se gradúa la quinta promoción del CIES en el
Horario

Hasta el 3 de enero se puede apoyar a los
Horario

Figure 90: Windows 8 news list view full-size

89
Microsoft te invita al cine


Figure 91: Windows 8 new details view full-size

Figure 92: Windows 8 new details view half-size
Developing an HTML5 mobile application for Barcelona School of Informatics’ Intranet

**aulas**

**Edificio A5**

![Room List View](image1)

*Figure 93: Windows 8 room list view full-size*

**horario**

<table>
<thead>
<tr>
<th></th>
<th>Lunes</th>
<th>Martes</th>
<th>Miércoles</th>
<th>Jueves</th>
<th>Viernes</th>
</tr>
</thead>
<tbody>
<tr>
<td>08:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>09:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18:00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Timetable View](image2)

*Figure 94: Windows 8 timetable view full-size*
Figure 95: Windows 8 latest notes view snapped

Figure 96: Windows 8 push notifications overview
Appendix II. Survey carried out

Please note this appendix is not available in English because this survey was done to students of Barcelona School of Informatics’ in Spanish.

Appendix I.A. Questionnaire

¡Hola!

Nos gustaría conocer tu opinión sobre las aplicaciones móviles del Racó. Te agradeceríamos que contestaras a las preguntas que encontrarás a continuación; creemos que no te llevará más de cinco minutos.

El propósito de esta encuesta es meramente informativo con el fin de llevar a cabo un Trabajo Final de Grado (TFG). La Facultat d'Informàtica de Barcelona (FIB) no tiene ninguna relación con la misma.

Pregunta 1 *
¿Qué sistema operativo tiene tu teléfono móvil?
Por favor, selecciona la opción correspondiente del desplegable.

- iOS (iPhone)
- Android
- Windows Phone
- Ubuntu
- BlackBerry OS
- Symbian
- Bada
- Otro
- No sabe/No contesta

Pregunta 2 *
¿Has instalado alguna vez la aplicación del Racó en tu teléfono móvil?
Por favor, selecciona la opción correspondiente del desplegable.

- Sí
- No, porque desconozco si hay una aplicación disponible.
- No, porque no hay aplicación disponible para mi plataforma.
- No sabe / No contesta.

Si tu respuesta es 'No', por favor salta a la pregunta 12.
Pregunta 3 (Opcional)
En cuanto a velocidad, ¿cómo valorarías la aplicación?
Por favor, selecciona la opción correspondiente del desplegable.
- Increíblemente rápida
- Muy rápida
- Rápida
- Aceptable
- Un poco lenta
- Demasiado lenta
- Excesivamente lenta
- No sabe / No contesta

Pregunta 4 (Opcional)
En cuanto a utilidad, ¿cómo valorarías la aplicación?
Por favor, selecciona la opción correspondiente del desplegable.
- Extremadamente útil
- Muy útil
- Útil
- Poco útil
- Muy poco útil
- Debe mejorar muchísimo
- No sabe / No contesta

Pregunta 5 (Opcional)
En cuanto a facilidad de uso, ¿cómo valorarías la aplicación?
Por favor, selecciona la opción correspondiente del desplegable.
- Extremadamente simple
- Muy fácil de utilizar
- Fácil de utilizar
- Un poco complicada
- Muy complicada
- Extremadamente complicada
- No sabe / No contesta

Pregunta 6 (Opcional)
En cuanto al aspecto de la interfaz, ¿cómo valorarías la aplicación?
Por favor selecciona la opción correspondiente del desplegable.
- Me encanta
- Me gusta
- Aceptable
- No me gusta mucho
- Muy inadecuada
- No sabe / No contesta
Pregunta 7 (Opcional)
En tu opinión, ¿cuál es la funcionalidad más útil de la aplicación?
Por favor, selecciona la opción correspondiente del desplegable.

- Últimos avisos
- Calendario de prácticas
- Mi horario
- Correo electrónico
- Mapa y puntos de interés
- Lista de asignaturas y detalle de las mismas
- Consultar la guía docente de las asignaturas
- Noticias de la FIB/UPC
- No sabe / No contesta

¿Y cuál es la que consideras menos útil para ti?
Por favor, selecciona la opción correspondiente del desplegable.
Las respuestas son las mismas que en la pregunta anterior.

Pregunta 8 (Opcional)
¿En qué momento del semestre la utilizas más?
Por favor selecciona la opción correspondiente del desplegable.

- Al inicio
- A mediodías
- En períodos de exámenes
- La utilizo siempre igual
- No la utilito
- No sabe / No contesta

Pregunta 9 (Opcional)
¿Utilizas la aplicación sin conexión?
Por favor, selecciona la opción correspondiente del desplegable.

- Muy a menudo
- A menudo
- Habitualmente
- Raramente
- Muy excepcionalmente
- No, siempre la utilizo con conexión a Internet.
- No sabe / No contesta.

Pregunta 10 (Opcional)
¿Qué cambiarías de la aplicación actual?
Por favor, sé breve y conciso/a.
La respuesta a esta pregunta es abierta.
Pregunta 11 (Opcional)
¿Qué funcionalidades echas en falta en la aplicación?
Por favor sé breve y conciso/a (p.ej. 'Podr estar descargar los ficheros adjuntos de un aviso')
La respuesta a esta pregunta es abierta.

Pregunta 12*
¿Tienes una tableta?
Por favor selecciona la opción correspondiente del desplegable.
- Sí
- No
- No sabe / No contesta

En caso de que tu respuesta sea afirmativa, ¿podrías decírnos cuál es?
Por favor selecciona la opción correspondiente del desplegable.
- iPad y/o iPad Mini
- Tablet Android
- Tablet con Windows 8/Windows RT
- Kindle Fire
- Tablet con BlackBerry OS
- Tablet con Ubuntu
- Otro dispositivo
- No sabe / No contesta

Pregunta 13 (Opcional)
Otros comentarios:
¿Tienes más comentarios? ¡Nos interesan mucho!
La respuesta a esta pregunta es abierta.

Pregunta 14 (Opcional)
¿Podrías decírnos qué titulación estás estudiando?
Por favor, selecciona la opción correspondiente del desplegable.
- Grado en Ingeniería Informática (GEI).
- Máster en Ingeniería Informática (MEI).
- Máster TI (MTI).
- Ingeniería Informática (EI).
- Ingeniería Técnica en Informática de Gestión (ETIG).
- Ingeniería Técnica en Informática de Sistemas (ETIS).
- Otras titulaciones.
- No sabe / No contesta.
Pregunta 15 (Opcional)
¿En qué curso estás?
La respuesta a esta pregunta es abierta.

Appendix I.B. Aggregated results

**Figure 97: Total surveyed users**

**Quin sistema operatiu mòbil té el teu telèfon intel·ligent?**

**Figure 98: Operating system use**
Has instal·lat alguna vegada l’aplicació al teu mòbil?

**Figure 99: Application installation measurement**

En quant a velocitat, com valoraries l’aplicació?

**Figure 100: Application responsiveness results**
En quant a utilitat, com valoraries l’aplicació?

Figure 101: Application usefulness results

En quant a facilitat d’ús, com valoraries l’aplicació?

Figure 102: Application ease of use results
En quant a l’aspecte de la interfície, com valoraries l’aplicació?

Figure 103: Application aesthetics results

Quina creus que és la funcionalitat més útil de l’aplicació?

Figure 104: Most used application features measurement
I la funcionalitat menys útil?

![Bar chart showing the least used features of the application.](image)

*Figure 105: Application less used features measurement*

Quines funcionalitats trobes a faltar?

- Poder veure i descarregar els fitxers adjunts als avisos.
- Notificacions push a iPhone.
- Poder definir intervals de sincronització en segon pla.
- La descripció completa de les assignatures.
- Poder llegir els correus directament des de l’aplicació.
- Entregar pràctiques.
- Accés al fòrum.
- Que es marqui un avís com a llegit al Racó quan el llegeixes des del mòbil.
- No haver de subscriure’s per correu per rebre notificacions push.
- Compatibilitat amb iOS 6.

*Figure 106: Missing features results*
Què canviaries de l’aplicació actual?

- Millorar molt la velocitat, cal agilitzar l’aplicació.
- Les notificacions push no van bé, arriba abans el correu.
- Les finestres s’obren en un navegador apart.
- Una interfície més atractiva.
- Falla quan actualitzes els avisos per la xarxa mòbil si tens una connexió lenta.
- Integració amb el calendari natiu del sistema.
- Configuració de la xarxa eduroam automàticament a Android.
- Fer-la més intuitiva i simple.
- Reduir dràsticament els temps de càrrega de la informació.
- Millorar molt la visibilitat dels avisos i de l’horari.
- L’aplicació d’Android provoca molts tancaments forçats.

*Figure 107: Current application change suggestions results*

Utilitze l’aplicació sense connexió?

*Figure 108: Offline use of the application*