Mobile application for connected objects

Alice ABADIE, Celia CHAUVALON, Jérome DUBAN, Jérémie FOUCAULT, Guillaume SCHAHL and Sandrine SOUB

Supervisors: Alexis CARREL-BILLIARD, Laurent REVEILLERE,

aabadie@enseirb.fr, cchauvalon@enseirb.fr, jduban@enseirb.fr, jfoucault@enseirb.fr, gschahl@enseirb.fr, ssoubnguemtchueng@enseirb.fr

Abstract: Many events such as Scientific Conferences, Sports or Cultural events take place in unknown environment for most users. We will use our GPS to get there but when we will be, how to not get lost in the building? Beacons could be used as an indoor GPS and provide to users additional information according to their current location. The application we're offering on the two most famous mobile platforms do not restrict themselves to basically display some data; our application is handy, useful and smart.

1 Introduction

Explosive growth of smartphones, tablets and PCs brought the number of connected devices to 12.5 billion in 2010 and is estimated to rise to 25 billion in 2015[1]. Indeed, nearly everything around us has now the ability to be brought online to interact with other devices and technology companies around the world are continuously trying to find new ways to link the Internet with physical objects in a wide variety of use such as home, business, hospitals, cars and entire cities. In this context, Apple introduced in 2013 a new technology called iBeacon. An iBeacon or a beacon, as the name suggests, allows mobile Applications to listen for signals from beacons in the physical world and react accordingly. iBeacon's main purpose is to provide, on a micro-local scale, the estimated distance between the user and the beacon. As a result, assuming that several beacons are detected nearby, the location of the user according to these devices can be determined. Then it is possible to trigger specific actions and/or events in certain situations. Plenty of innovative use cases already exist for this technology and its low consumption property contributes greatly to its recent massive spreading. For example, a lot of museums are using beacons to provide additional information to visitors, a new kind of advertising has appeared with applications to guide clients in stores and direct them to relevant articles and iBeacon key rings have been made to easily find lost belongings. More and more companies are developing some smart and really useful products thanks to this technology and beacons will obviously make the concept of the Internet of Things an everyday habit in the next few years. The aim of our project was to find an innovative and useful use case for this technology.

2 Use Case

The application had to use efficiently and ingeniously most of beacon's features. Most of the time, these devices are used to trigger an action or to estimate the distance to a given object. But beacons can also be an alternative to GPS inside a building. As a result it has been decided that the application had to offer a handy way to locate oneself in a building. The context of a conference at the ENSEIRB-MATMECA seemed to be a good starting point to begin our thought.

2.1 Scenarios

First of all, a lot of conferences take place in foreign countries and in unknown buildings so it can become really difficult for attendees to find their way. This is why the application will provide a map of the building where the

[1] According to CISCO company *http://www.livescience.com/38562-internet-of-things.html*

conference is hosted in such a way that the user will be able to identify its location and potentially its destination.
Furthermore as an attendee walks through the conference, he may want to know which talks are attended around him, thereby beacons could be used to send relevant notifications about talks that take place or are going to occur nearby.
Finally, to organize its conference's schedule, the user may need a way to consult the conference planning and information about sessions and talks. Thus our application will provide such schedule and detailed information.

In order to illustrate this use case, we are going to take the example of John. John is an English developer and is going to attend a conference at the ENSEIRB-MATMECA. When John arrives at the conference, the building and the conference are automatically recognized by the application which will download all the data needed in a transparent way. It is important to say that John did not need to download a new app, he only had to use the same application he employed for his previous conferences.
John is so absent-minded that he forgot his conference schedule at the hotel. However he does not worry because he knows that his application will provide him everything he needs. When the developer reaches the building, the application schedule displays a list of the conference's talks. If John is interested in a talk's name, he simply has to select the event to get more detailed information. Unfortunately he does not know how to reach the room where the talk takes place. This is not a problem anymore because a button enables him to open the map with the selected room and his location displayed. He just has to go through the building and follow the dot on the map to reach the talk. The developer can obviously interact with the map (zoom, drag...) if the display is not convenient enough.
John can also look for a room, a talk or a session from the map view in order to know where he has to go. Sometimes, a talk title may not be explicit enough. This is why a notification system offers more information about the talks nearby. If John stands close to a room for a little while, *ie* 5 seconds, he will receive a notification to warn him about the talk that takes, or is going to take, place. If John misses the notification, he can consult it later within the application.

2.2 Beacon experimentations

The main risk in the development of applications was the use of beacons. Beacons are using Bluetooth Low Energy in order to locate the user in the building, so the environment may have a huge impact on the performance of our applications. To prevent and predict this impact, several tests in different situations have been done. First, the study focuses on the attenuation of the BLE signal strength, the Received Signal Strength Indication (RSSI), with distance on iOS and Android without any obstacle. This experiment has been done several times in other contexts especially with barriers such as brick or floor. The crowd also has a meaningful impact on the signal strength received.
Furthermore, beacon's signal stability has been observed during these tests. These measures lead to the fact that the power of each beacon was really unstable. Indeed measures taken at a same place fluctuated quickly both on the iOS and Android platforms. The power variation is visible on the figure below:

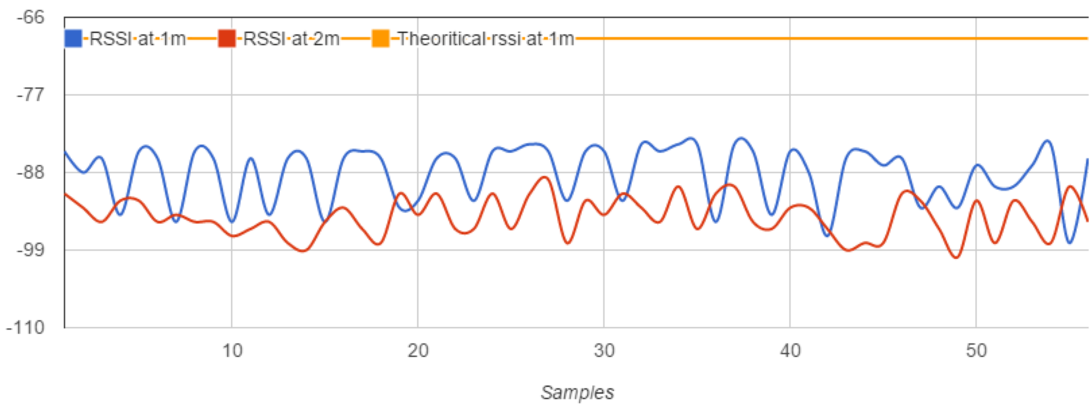


Figure 1: Fluctuation of beacon's RSSI on Android phone

All these studies allow us to cope with the difficulties inherent of the use of beacons. Finally, the tests permit to define more precisely the characteristics of the used beacons. Beacon detection is possible thanks to the smartphone for a distance lower than 25 meters (see figure 2) without any barrier and 3 levels of detection have been defined: immediate (0-1m), near (1m-3m) and far (3m-25m). Furthermore, an important diminution of the perceived intensity is noticed on the device when the signal comes across an obstacle. It represents around 25% of loss for each obstacle.

To conclude, these tests show the necessity to use an extensive amount of beacons to have an efficient localization. The beacons have to be placed close to each other (around 10meter) in order to provide an accurate localization.

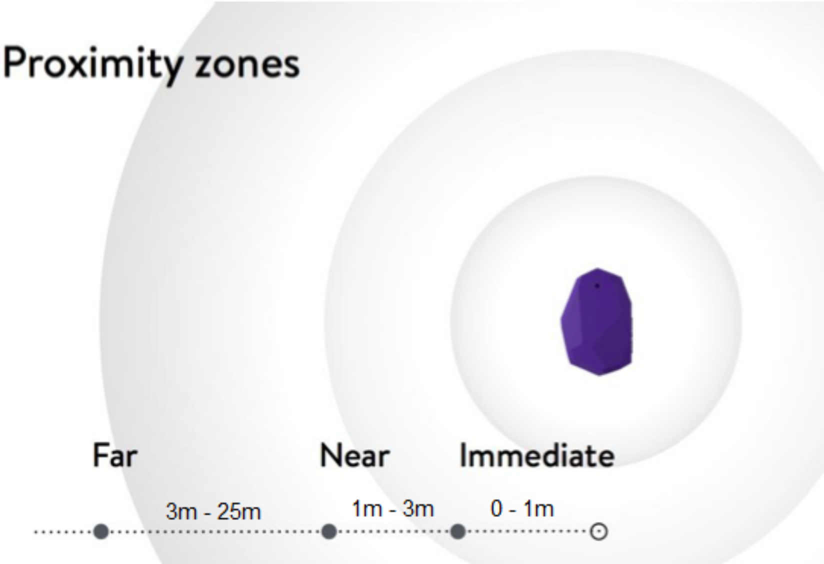


Figure 2: Scope of beacons detection

3 Application Building

3.1 Bluetooth Low Energy

One of the major features of the application will reside in its indoor localization functionality. The location of the user is determined by its distance from the detected beacons at a given time.

Beacon technology uses the Bluetooth Low Energy (BLE), also known as Bluetooth Smart. Bluetooth Low Energy is a light-weight subset of classic Bluetooth. Designed and marketed by the *Bluetooth Special Interest Group*, it was introduced as part of the Bluetooth 4.0 core specification. In comparison with classic Bluetooth, Bluetooth Smart is intended to provide considerably reduced power consumption and cost while maintaining a similar communication range. Many computers and mobile operating systems including iOS, Android natively support Bluetooth Low Energy.

Beacon uses Bluetooth Low Energy proximity sensing to transmit periodically broadcast advertising data. The information can be picked up by central devices such as mobile phones, tablets or computers. Each BLE peripheral device sends a unique identifier (UUID) and some data payload. Apple specified that the iBeacon model outlines the payload by introducing two other identifiers: the major and the minor. The specification indicates amongst others that the theoretical signal power at 1 meter has to be sent within the advertisement data.

3.2 iBeacon specification and mobile implementation

As the iBeacon technology is developed by Apple, iOS supports natively the iBeacon specification. Conversely, even though the Bluetooth Low Energy is well handled on Android, beacon management can't be done properly without a library. As a result, some algorithm implementations differed somewhat according to the platform (iOS, Android).

When Apple introduced iBeacon specification, they also introduced many APIs to manage them. These APIs are part of the Location library which contains tools to get user location. iOS and Android beacon libraries, both handle beacons quite the same way. Two basic notions are introduced to manage and detect beacons. The first one is "monitoring". Monitoring is a way to detect user's entry and exit into specific regions. A beacon region is an area defined by the device's proximity to Bluetooth low-energy beacons. With such region and monitoring, applications can be notified when a user crosses geographic boundaries or when a user enters or exits the vicinity of a beacon. The second notion introduced is "Ranging", it allows the application to retrieve more information about beacons in range. This process helps the iOS device to estimate how far it is from an iBeacon. Yet, the measured distance being imprecise, three proximity levels have been defined: Immediate, Near and Far that can be used to refine how far we are.

3.3 Centre of mass Algorithm

When a beacon is detected nearby, the only way to estimate the position of the user is to use the received signal strength (RSSI). This datum enables to define a region on iOS and to estimate a distance on Android thanks to the Android Beacon Library as said before. As a result, it has been necessary to find a way to implement the localization algorithm so that it could work on each platform. This problematic is similar to the GPS one which needs to find the coordinates of a given device according to its distance from other peripheral devices. First, the trilateration algorithm used by the GPS seemed to be a good way to localize the user within a building. However the permanent fluctuation of the number of beacons detected did not enabled us to use this algorithm. A simpler algorithm was eventually used: the centre of mass. Even if this concept is supposed to be used with masses, it can also be implemented with distances. Indeed, the heavier a point is, the closer the centre of mass will be to this point. Conversely the location of the user is determined by its distance from the beacons nearby so if the user is close to a beacon he has to appear next to this one.

Concretely, given a set of beacons, their minor field will be used to retrieve their coordinates from the data previously download and locally stored. For each of these beacons, a weight is also computed using the device estimated distance to the beacon. The used formula differs a little between iOS and Android:

$$weight_{iOS} = \frac{iOS\ region}{distance} \qquad\qquad weight_{Android} = \frac{1}{distance}$$

Where "iOS region" corresponds to the region resolved by iOS. If the device is in Immediate proximity of a beacon, the value 3 will be used for "iOS region", if it is Near value 2 will be used and value 1 if it's Far.

Even if the same algorithm and the same technologies are used in our applications, the distance is not computed the same way on iOS and Android, thus the algorithm implementation differs between the two platforms. Indeed, on iOS, Apple explicitly tells developers to not use the beacon's accuracy field to identify a precise location for the beacon as it may fluctuate due to RF (Radio Frequency) interference. Because of these, the current centre of mass is the average of the last one and the new one. Furthermore to reduce these fluctuations, it is assumed that the user's walk speed is approximately 1.3 m/s thus he cannot walk 10m in 1s. With this hypothesis in mind we can remove points that seem wrong. Therefore if the newly computed centre of mass is located too far from the last one to be realistic, this centre of mass will be ignored.
On Android such calculations are not necessary as the library handling Beacons already provides a quite precise

distance. The counterpart is that the computation is quite slow, 1s to 5s, but this is not an issue for the application as it is not affecting the user experience.

3.4 Localization leaps

With the centre of mass as it stands, the user is likely to experience some localization leap. It means that its location can quickly move from one place to another. This could be caused by the fluctuation and the lack of exactness of the signal strength estimation. However, leaps can be quite important and have to come from something else. Indeed, sometimes some beacons are not detected during a scan. The figure 3 below illustrates the consequences of such a thing.

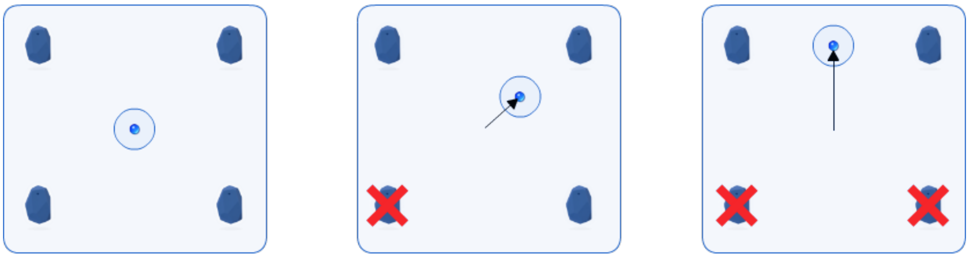


Figure 3: Localization leap example

The first picture shows the case where the user detects four beacons nearby at equal distance. As a result, the user is located in the middle of the beacons. Nevertheless, if a beacon is not detected anymore then the user will be placed at the centre of the triangle formed by the three remaining beacons. The location of the user will consequently vary. If less beacons are detected, the user will be localized even further. Most of the time, this is due to the Bluetooth Low Energy properties of the beacon. The low energy consumption is due to the fact that the beacon alternates between active and sleep periods. The broadcast signal is actually only sent during a short amount of time then the beacon stays passive before sending the advertisement again. The figure 4 presents an overview of a Bluetooth Low Energy device activity over time.

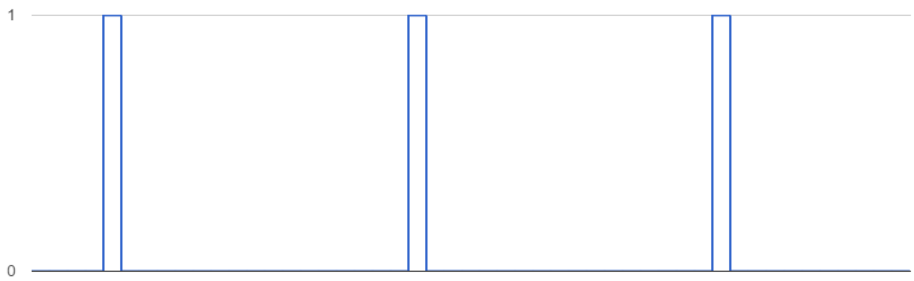


Figure 4: BLE advertisement over time

As a result, if the central device (phone or tablet) does the scan during a sleep period, the beacon will not be detected. This is why some beacons within range are not detected, causing these leaps on the localization algorithm.

3.5 Data Model

To organize data elements and standardize how the data elements relate to one another, a data model has been set up. Three different types of JSON files are used.

- **Conference** contains all the information about a conference.
  The following vocabulary has been established:
  - The object ***conference*** is the congress event. It is composed of several attributes such as a title, an address, a starting and an ending day and an id.
  - The conference gathers several ***tracks***. A track is a thematic and is defined by a title and an id.
  - The track owns ***sessions***. A session is a group of ***talks*** taking place in a specific room.

- **Building** contains the information about the floors and the rooms which are used during a conference.
- **Beacon** contains the position of the beacons.

The application conference data are written in JSON files and stored on a remote server. It is this file that the audience will load when arriving to the conference and which will enable to display conference's information on the application.

4 Assessment

4.1 State of the project

To get all the information displayed smoothly, the user needs to update his application whenever he attends a new conference, this is done automatically when the application is launched.
On the application's homepage, conference's main information is displayed. The conference's name, address and dates are displayed.
To have access to all the other information offered by the application the sliding menu on the left edge of the screen can be accessed at any time. The building map is accessible through this menu as well as the conference's schedule and the list of the received notifications which is only available on Android phones at the moment.
The "Map" view enables the user to access the map of the building where the conference is hosted and his location. Furthermore, the search bar allows the user to seek a room, a talk or a session. The room of the talk will then be displayed in red on the map to show him where he needs to go.
Thanks to the schedule view the user will find all the information about the timetable of each talk and the room where it takes place. He can pick a talk and see more details about it such as a summary or the speaker, and then directly go to the map to see where the talk is going to occur.
On Android phone, the notification page lists all the notifications received on the user's phone during its day.

4.2 Possible improvement

At this point, the application is fully functional, but can still be improved, especially concerning the server side.

Indeed, there is no intelligence at all on the server side. The server side only hosts some static JSON files and no API enables to easily edit them. Every information has to be edited manually whether it concerns the building, a beacon or a conference. A major improvement could be the creation of an API to provide the necessary tools to develop a website or a mobile configuration application. These interfaces will provide a way to configure the physical data such as the coordinates of the beacons or the building parameters but also all the information related to a conference. Enable speakers to edit their own talks could be considered.
This kind of interface would make the data management much easier and would make the system more adjustable. The installation of the system would obviously be more straightforward thanks to a favourable administrator interface.

Furthermore, the Beacon technology is still evolving and may know important changes in the coming years. The project will be impacted by any of these changes and the algorithm used for indoor localization could be updated to be more accurate and efficient.
The application battery consumption has not been studied during the development process and a lot of improvements could be done to reduce battery usage. Indeed, a BLE scan requires a lot of energy. A solution could be to increase the period between scans if no beacons are detected nearby

5 Related Work

5.1 State of the art: technology

Indoor localization permits to know more or less precisely the position of someone in places where GPS access is not available or inaccurate. Several solutions are currently explored to answer this issue. They are mainly based on wireless technologies such as Wi-Fi, Bluetooth, UWD (Ultra Wide Band) or RFID.
RFID has been used for years mainly in the industry for logistics tracking. But unfortunately this is not well suited for indoor localization as most smartphones are not able to receive RF anymore. Another solution is a mix between Wi-Fi and Bluetooth which use the smartphone ability to automatically connect to unsecured Wi-Fi network to easily increase the number of beacons able to provide data about user‘s location. Indeed as the BLE beacons, Wi-Fi signal strength can be measured to estimate distance to the emitter.

There is also many start-up working on new indoor localization systems using for instance exotic technologies such as ultrasound or infra-red light.

5.2 State of the art: mobile development

There are many ways to build smartphone apps. Some seem to be really easy to use and powerful, while the standard way to build it usually discourage many developers.
Each operating system has a specific way to create mobile application. In order to build it, a specific SDK (Software development kit) is needed as well as a particular language. This type of development allows to create a native application on each operating system. Thus the resulting application will be able to support all the phone’s features and will be more ergonomic. However it can represent an important hourly rate if the application has to be developed on several platforms.

Another solution is to create a web application or web app. A web application is a website developed with usual web languages such as HTML, CSS or JavaScript but, it is created specifically for smartphones. Such application can be used on any device but requires an internet connection and cannot use the phone features such as its GPS or camera.
A hybrid solution is to create application thanks to specific framework using the web technologies. Several frameworks such as PhoneGap, Sencha or AppAccelerator allow to do this. PhoneGap which is one of the most used today was at first glance the perfect solution for this project. PhoneGap allows the creation of mobile application on any mobile platform (iOs, Android, Windows Phone) using standardized web language. Moreover access to the specific phone functionalities is possible thanks to PhoneGap plugins. This kind of framework permits the easy creation of application for a large variety of platforms.
However the application has been developed natively because of difficulties met on PhoneGap, since the framework does not support well the BLE technology at the moment. Indeed, the BLE plugins available on Phonegap were really limited and did not offer all the functionalities needed for each platform to create the desired application.

6 Management

As it was essential to share the work amount among the team, different work packages were created and assigned to the members of the team. The Agile method was chosen at the beginning of the project because it encourages collaboration and communication between students. Furthermore, it has been proved to be really efficient in computer programming. Moreover, the nature of the project involved small cycles and a fast integration in order to solve issues that could have occurred following a previous integration. This method worked well with our weekly meeting. Thus, the Agile method fitted perfectly.

Figure 5: Main stages in Agile method

User stories were described at the beginning in order to define what a user should be able to do. Sprints were created to gather one or many user stories which defined a schedule (cf. appendices). One task was assigned to each student at the beginning of each sprint and he had to achieve it within the end of the sprint in order to be able to merge it with the main project. Furthermore, a project manager header was chosen. His objective was to organize the project and to get a global vision of it. Additionally, as integration of all the work packages was essential, an integrator manager was chosen. His purpose was to deal with integration conflicts. In order to get a different point of view in the integration part, the integrator manager was different from the project manager. Each meeting was the occasion to progress and to integrate the work. Thus, at the end of each meeting, a report with the state of the sprint and its user story was made to gauge the project progress week after week and know the difficulties encountered. These reports and every detail about the work advancements were shared on Google Drive to permit to keep track of the project and to share it with all the team. The source code was also shared on GitHub in order to work collaboratively. This method also allowed the project manager and integrator manager to check the progress and the difficulties encountered to be able to deal with it rapidly.

7 Conclusion

At the moment the application is fully functional, but can still be improved, especially concerning the server side. Indeed currently static JSON files are used to retrieve data from server. The features offered by iBeacon allow us to create a new type of application gathering indoor localization, schedule of conferences and the use of notifications in order to enhance the user experience during conferences. Thus the application focus on what information will be relevant for the audience, how to access it seamlessly and in a visual way. Our application is ready to be used in real conditions as part as the event organised by M.Reveillere in May 2015. However, the increase of the number of beacons in the building is essential to use our application on a wider scale. With such change, our application can be expanded for any events.

References

[1] http://developer.android.com/index.html

[2] https://developer.apple.com/

[3] https://altbeacon.github.io/android-beacon-library/index.html

Appendices

In order to carry out well our project, two teams were created. Four people worked in the Android development and two people on the iOS. The configuration was made so because of hardware complexities to develop on Apple environment.
Initially, the project was divided into six sprints. After a month, the initial draft was modified due to the abandonment of PhoneGap framework.

Second version of the sprints:

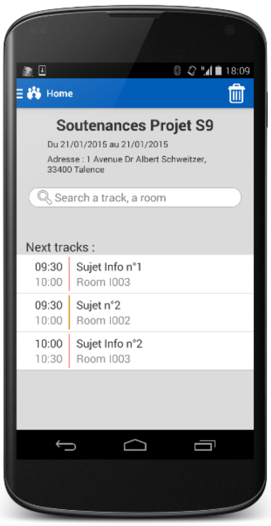
Sprint	Functionalities	Complexity
Sprint 1	10 October - 24 October	3
	• Definition of the subject • State of the art	
Sprint 2	25 October - 14 November	3
	• Study of the framework PhoneGap • Research Bluetooth Low Energy • Define specifications • Tests of beacons	
Sprint 3	15 November - 24 November	3
	• Data model • Study of the development with iOS and Android	
Sprint 4	25 November - 2 December	5
	• Architecture of the application • JSON file	
Sprint 5	3 December - 10 December	12
	• Create the map of the building • Locate the user position on the map	
Sprint 6	11 December - 18 December	9
	• Choose the destination • Locate the destination on the map	
Sprint 7	19 December - 29 December	9
	• Notification history • Display the schedule	
Sprint 8	30 December - 9 January	3
	• Update data • Acceptance tests • Report	
Sprint 9	10 January - 20 January	3
	• Report • Acceptance tests	

Budget

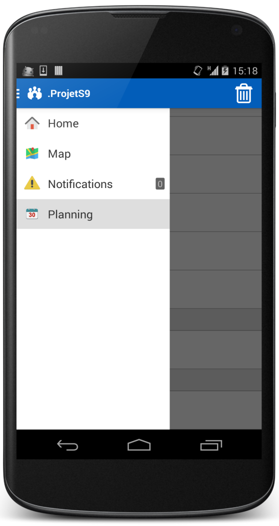
This second table represents the time spent for each platform. The remaining time correspond of the time needed at the end of the sprint. It is not the current time because at the moment all the user stories were functional.

US	Name	Spent Android (Hour)	Remaining Android (Hour)	Spent iOS (Hour)	Remaining iOS (Hour)
US0	Define the subject, BLE test and PhoneGap	70	25	70	25
US1	Display the main Conference information	29	17	27	17
US2	See the map	30	5	20	30
US3	Choose a destination	5	8	7	4
US4	Locate myself and my destination on the map	20	5	5	20
US5	Notification when I stay near a door (~15s) or when I attend a talk	0	4	0	5
US6	Get conference's schedule	3	8	9	2
US7	Update data	1	1.5	1	1.5
US8	Notifications history	4	4	0	0

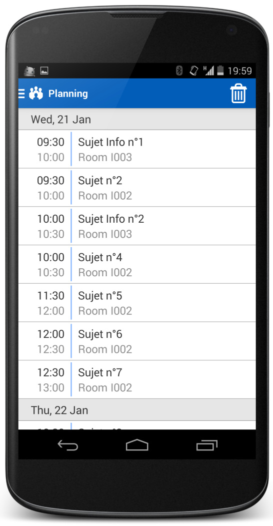
Application screens:
Android views:



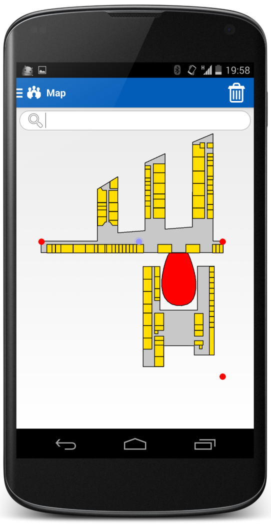
Home Page



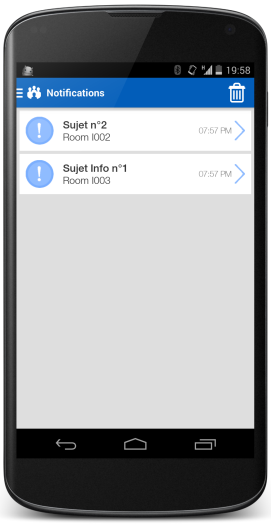
Menu



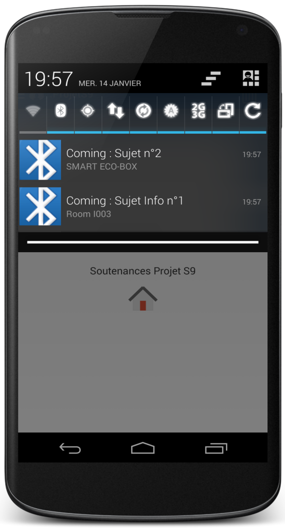
Schedule



Map

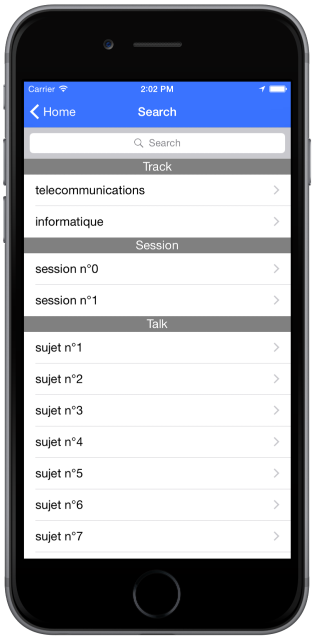


Notification List

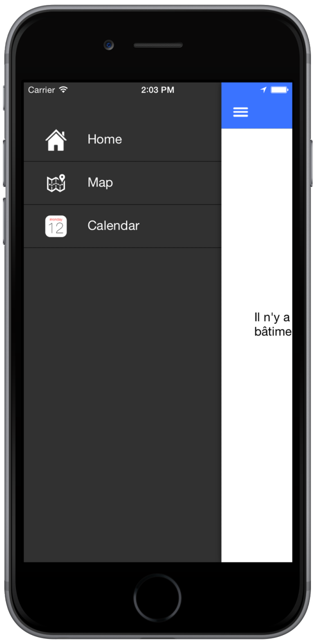


Notifications

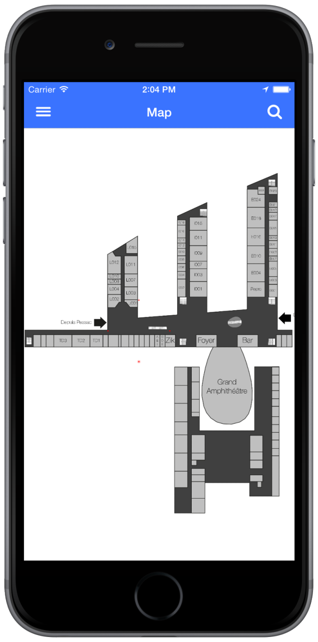
iOS views:



Search Bar



Menu



Map