UNIVERSITAT
JAUME•I

Erasmus Mundus Master in Geospatial Technologies
Master Thesis (2012/2013) (SIK013)

UJI's SmartCampus: Place finder App

Project presented by:

Diego Gargallo Tarín

Dissertation supervised by:

Oscar Belmonte Fernández
Roberto Henriques
Joaquín Huerta Guijarro

Castellon de la Plana, November 2012

Acknowledgements

First I would like to thank my supervisor Oscar Belmonte Fernández for his support and helpful guidance. My gratitude goes out as well to all the people involved in the Smart Campus project of Universitat Jaume I (UJI), the effort of which is enabling the development of so many interesting things. Lastly I wish to thank my most loved ones, who have been the target of many of the testing with real users, and whose support helped me to finish the thesis.

Abstract

This thesis consists in design a place finder for mobile devices, specifically locations within the campus of the Universitat Jaume I. The app will use web services for access to searched information and campus maps.

The smart cities and smart campus projects are born and growing around the world. These projects are based on using the newest information and communication technologies for the benefit of citizens.

The thesis is framed within the UJI Smart Campus project. Some of the UJI researchers are involved in this project in some way or another, and this app uses resources that have been produced by them. Thus some of the resources developed in this thesis will be able to be used for other purposes.

Planning was made following agile [17] development methods, which have allowed us to modify and improve the application requirements as was being implemented. Throughout the development has taken into careful consideration end users, and the details of the user interface have been care so that the final product be as usable and simple as possible.

The app is open for improvements, adding new functionalities and integrating with other parts of the UJI's Smart Campus project.

Key words

Smart campus, place finder, Android app, ArcGis Server, ArcGIS Runtime Software Development Kit (SDK) for Android.

List of Acronyms

ADT Android Development Tools

GIS Geographic Information System

GPS Global Positioning System

HTML HyperText Markup Language

HTTP Hypertext Transfer Protocol

IDE Integrated Development Environment

LDAP Lightweight Directory Access Protocol

OOP Object-Oriented Programming

OS Operative System

REST Representational State Transfer

SDK Software Development Kit

UI User Interface

UJI Universitat Jaume I

UML Unified Modeling Language

URL Uniform Resource Locator

Table of contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

Smart city projects are improving a lot of aspects of city management, using the newest information and communication technologies for the benefit of the citizen. The same principles of the Smart cities can be applied to university campus, producing similar results for the university community.

Nowadays smart phones allow the gathering of data from diverse sensors, easing the work of collecting information. Moreover, the last mobile devices can provide us the necessary tools to use all the valuable information generated by the SmartCity and SmartCampus projects.

In addition, UJI is creating some web services that serve many geolocated information, encouraging the creation of mobile applications that can take advantage of this. UJI is very interested in the possibility of developing a mobile application that can use any web service created in the smart campus project.

With all that the mobile app is a tool more attractive than others more generic. Being a mobile app, everyone will be able to install the app on their mobile devices and its functionalities may be used anytime.

1.2. Objectives

One of the most demanded functionalities is the creation of a mobile app to find on-campus facilities. Providing a base map with every room, and a service that locates and connects people and rooms, the application is created to locate people and rooms and display them on a map. Moreover, a mobile application has the advantage that the users can always carry it with them, and use it when they really need it.

The main objective is create this mobile place finder, specifically focused in the new UJI's Smart Campus project but modular enough to include new functionalities or adapt to other campuses or cities which have the same services.

Besides the application, some web services are created to provide it with different search criteria. These web services can be easily reused for other purposes.

1.3. Structure of the documentation

This doccument is organized in 6 chapters and 2 apendixes, and a section with the bibliography after the apendixes.

The chapterscontent is described below:

- **CHAPTER 1 - INTRODUCTION:** This chapter tries to review in a general way the undertaken work. It includes the motivation for the project, the pursued objectives and a description of the document organization.
- **CHAPTER 2 - PLANNING:** Here we can see the tasks that have to be complete to finalize the development of the app, also the expected time to perform them and the time that really is needed.
- **CHAPTER 3 - RESEARCH OF RESOURCES AND TECHNOLOGIES:** This chapter focuses to show the research that has been done to choose the resources and technologies to be used during development.
- **CHAPTER 4 - DEVELOPMENT:** This one describes the app and web services development.
- **CHAPTER 5 - TEST AND RESULTS:** To verify the correct functioning of the developed products and they fulfills their mission, they have been subjected to a test collection detailed in this chapter.
- **CHAPTER 6 - FUTURE WORK AND CONCLUSIONS:** To finish, the last chapter shows the improvement possibilities and the conclusions after the end of development.

And the appendixes' content:

- **APPENDIX A - UI EVOLUTION:** This appendix shows the evolution of the user interface throw the development.
- **APPENDIX B - USER GUIDE:** Tutorial to guide users through the most common actions to be taken on the application.

In addition to this document, a website is provided. Following this link, one can access more documentation that has not been included here because of its size and because exist other ways to offer better view.

The JavaDoc can be visualized using this link, and the complete source code is accessible in GitHub [6], with this other link.

2. Planning

This thesis was conceived like a mobile application development (both client-side and server-side). The methodology used for the development of the application is known as 'agile'. The development was made with iterations, obtaining a product at the end of each in order to add features and improve existing ones.

2.1. Planned timing

Schedule was planned after the research and before the app development. These tables containing the planned timing for this development, but only meant for guidance. The first one shows the first iteration planning, and the second one includes the rest of iterations to finish the implementation.

2.1.1. First iteration

The first iteration has aimed to obtain the structure of the application with some of its most basic functions. Its planning incorporates recurrent weekly revisions, which are used for collecting the generated documentation.

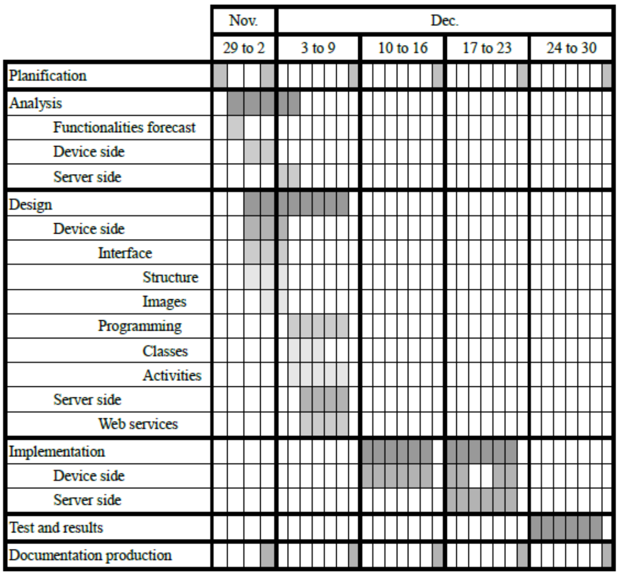


Figure 2.1.: Timing of the first iteration

2.1.2. Improving the app

After the first iteration the app development is distributed across different improvements over the initial structure. During each one the user interface and the internal structure are reviewed, adding relevant functionalities.

The latest iteration is focused on the implementation of web services to provide new search criteria.

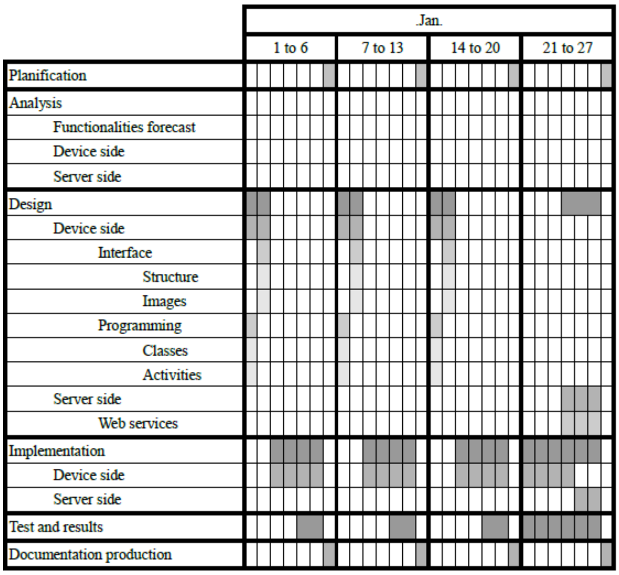


Figure 2.2.: Timing of the improving iterations

2.1.3. Writing the report

The report has been written throughout the entire process that led to the completion of the thesis, but the last week has been set aside to produce the last version of this, and include all the details of development.

2.2. Real timing

The application development has followed quite closely the planning that is specified in the preceding paragraph.

Eventually there have been some delays, caused mainly by the typical problems that arise with the use of technologies that are new for developers. These delays have been countered as soon as possible so they do not induce major changes in planning, and consequently in the final product.

3. Research of resources and technologies

This section tries to show the tools, resources and devices used during development, and decisions made while researching thereof.

3.1. Resources

The resources (hardware, data sources, mobile devices...) were chosen to simplify development, minimize costs and provide a suitable working environment.

3.1.1. Hardware

Only one laptop was used to develop the app, design all its aspects and write the documentation.

Testing has been made with some Android devices, as shown in figure 3.1: three smart phones to test the performance of the main features with different Android platform versions and Operative System (OS), and two smart phones and a tablet pc to test the user interface with other screen sizes.

3.1.2. Server and data

The used server is an ArcGIS server, accessible through its Representational State Transfer (REST) Services Directory [11]. This server provides very useful web services to use the UJI's maps and geographic information.

Table 3.1.: Android devices for testing

Device	**OS**	**Ver.**	**Screen**	**Features test**	**Interface test**
Galaxy S II	Cyanogenmod9[5]	4.0.4	4.3"	Used	Used
Galaxy S II	standard OS	2.3.3	4.3"	Used	Non used
Galaxy mini	standard OS	2.3.6	3.14"	Used	Used
Assus slider pad	standard OS	4.0.3	10.1"	Used	Used

Other web services have been used in our app utilize the UJI's institutional directory [10]. Where Lightweight Directory Access Protocol (LDAP) is used to reflect the organizational structure of the UJI.

These web services are enough to develop the main functionalities of our app. The number and quality of these web services is still growing, therefore presumably the new web services that will be implemented could be used in future improvements of our application.

3.2. Chosen technologies

Some technologies used in this project are well-known in the software development and geographic information science. The following technologies are used during the project. These have been chosen from all the available possibilities.

3.2.1. Mobile platform

The application can be developed for one or more mobile platforms. Taking into account the number of potential users within the university community, the time available to develop and how easy it is for the developer to work with the chosed technology, the chosed mobile platform is Android.

3.2.2. Development toolset

The Android Development Tools (ADT) Bundle [1] provides the essential Android SDK components and a version of the Eclipse Integrated Development Environment (IDE) with built-in ADT to streamline the android app development.

3.2.3. Geographic information system

The ArcGIS Runtime SDK [4] for Android is the best choice to use in our application all the possibilities offered by ArcGIS Servers. It is well documented [7] to take advantage of all the capabilities of an ArcGIS Server.

3.2.4. Web services

RESTful web services follow the REST principles [13] [8]. This is the chosed tecnology to develop the web services what uses the UJI's institutional directory. Its acceptance is growing as an alternative to others with more users.

3.2.5. Project building

When a project development begins, the development team can choose to use a project builder or not. Maven [3] is the most popular project builder, and provide an easy and comfortable way to include modules, plug-ins and dependencies of our project.

On the one hand, at first thought to use it for building the app but ultimately the idea was rejected, mainly because it did not provide any important improvement (the app doesn't need to include a large number of modules, plug-ins and dependencies), and because nowadays the eclipse plugins for Maven and the eclipse plugins for ArcGis SDK could not be used together with the same project.

On the other hand, Maven was used to develop the web services for the UJI's institutional directory. It is because the project had already started, which has already used it. Besides this, the offered advantages of Maven in this project overcome the cost of use.

3.2.6. Documentation

Several tools have been used to create all the documentation about the work.

Javadoc is a documentation generator, and provides an easy and elegant way to create and show Java code documentation.

LaTeX is a widespread system for the text composition, oriented to the technical document creation. It was used with TeXworks, open source software (GPL license) for writing text documents. TeXworks [9] provides an editing and working environment with TeX, LaTeX, ConTeXt and XeTeX, plus a pdf viewer.

Html5 [12] is the fifth revision of the HyperText Markup Language (HTML) standard. It has been used to create a website that gives access to the application and other documentation.

3.3. Supporting different Android aspects

The big amount of different Android devices used around the world, and their intrinsic differences, make it imperative to develop applications that take into account these differences to build their user interfaces [2].

3.3.1. Platform version

Actual active devices are running a given version of the Android platform. The latest versions provide to us the newest functionalities and the best way to use the smart-

Table 3.2.: Statistics of android platform

Version	**Codename**	**API**	**Distribution**
1.5	Cupcake	3	0.1%
1.6	Donut	4	0.3%
2.1	Eclair	7	3.1%
2.2	Froyo	8	12%
2.3-2.3.2	Gingerbread	9	0.3%
2.3.3-2.3.7		10	53.9%
3.1	Honeycomb	12	0.4%
3.2		13	1.4%
4.0.3-4.0.4	Ice Cream Sandwich	15	25.8%
4.1	Jelly Bean	16	2.7%

phones possibilities. As shown in table 3.2, not all the Android active devices use the latest Android platform, because that is necessary select a version new enough to take advantage of it, but old enough to be used by most devices.

3.3.2. Screen

Android uses the size and the density to categorize the device screens. There are four generalized sizes (small, normal, large and xlarge), and four generalized densities (low, medium, high and extra high). The screen orientation (portrait and landscape) are important to design the user interface in terms of screen dimensions too.

Considering all this, and seeing that the future users of our app will use any type of device, it is necessary to use different resources (layouts and bitmaps) to cover all possible screens.

3.3.3. Languages

The potential users of our future app are the students and members of UJI comunity. The app must be developed to use in the most common languages of the university: Spanish and Valencian for the regular students and university comunity members, and English for the Erasmus students and exchange staff.

4. Development

The sections in this chapter describe the development process, distinguishing between the application and the server part. The focus is on the different types of Web Services used, but also on the analysis and application design and implementation.

4.1. Web services

Some of the used web services were developed for other purposes. Others, however, have been developed specifically for the "Mobile UJI Place Finder" (but trying to follow the same style of development so they can be reused as the remaining).

4.1.1. UJI's geotec ArcGis Server

UJI has some servers of various types to be used by the teaching and research team. The used Geographic Information System (GIS) server has installed ArcGis version 10.11, and provides a range of map services to be consumed by web sites and mobile applications.

Provided web services bring us the necessary map layers for the app, and the ArcGIS SDK [4] allow us to manipulate these layers in our benefit. The ArcGis server is accessible through a REST Services Directory [11].

Table 4.1.: ArcGisLayers

Layer	Web address	Description
viscaUJI_desaturado	/UJI/viscaUJI_desaturado/MapServer	Grayscale basemap of the UJI's campus.
viscaUJI	/UJI/viscaUJI/MapServer	Colour basemap of the UJI's campus.
Building Interior Spaces	/UJI/UJIBuildingInterior/MapServer/1	Feature layer with the interior of the buildings.

Table 4.2.: LDAPWebServices

Before thesis	WebService	Description
Developed	/Locations/offices/	Search offices by room number.
Developed	/Locations/people/	Search people by name.
Not developed	/Locations/telephones/	Search people by telephone number.
Not developed	/Locations/titles/	Search people by title.
Not developed	/Locations/mails/	Search people by mail.

4.1.2. UJI's institutional directory

The institutional directory has information of all the people with a relationship with the university, reflecting the organizational structure of the university.

Web services can be hosted on an Apache server, and these are in charge of the searches in the UJI's directory.

Before the thesis begins some web services was developed, the people search by name and the office search by room number. These web services are the start point of all other web services develop to use with the UJI's directory.

Web services development

The web service receives a request when someone tries to access to its address using Hypertext Transfer Protocol (HTTP) protocol. The web service uses the information passed to it in this way to make a search in the LDAP system.

4.2. Mobile App

The mobile app is the final product of the thesis, and therefore is the part where most efforts have been done to make it better.

4.2.1. Analysis and Design

The analysis and design of the app was made using UML [16], firstly drawing diagrams to capture the results. Then the designs will improve and gradually changing with each modification.

The followed UML diagrams are shown below. These are the latest version of the diagrams, as they have been modified as they were made in the implementation.

Some classes are included to facilitate the inclusion of new functionalities in the future.

These will be seen in greater detail in sections 4.2.2 and 6.2.

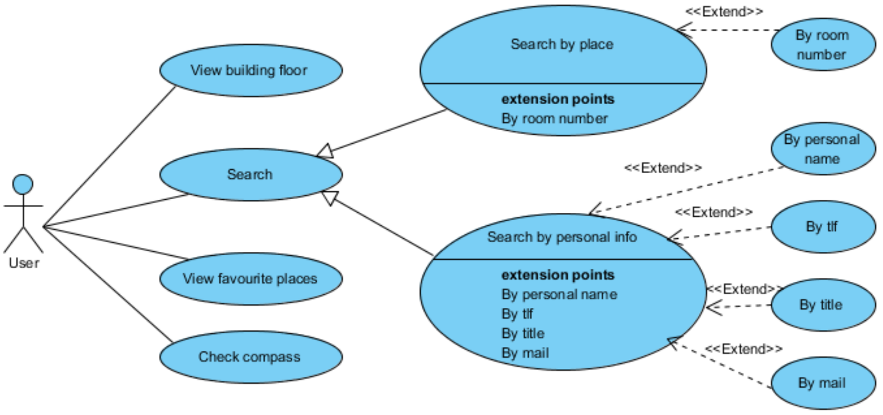
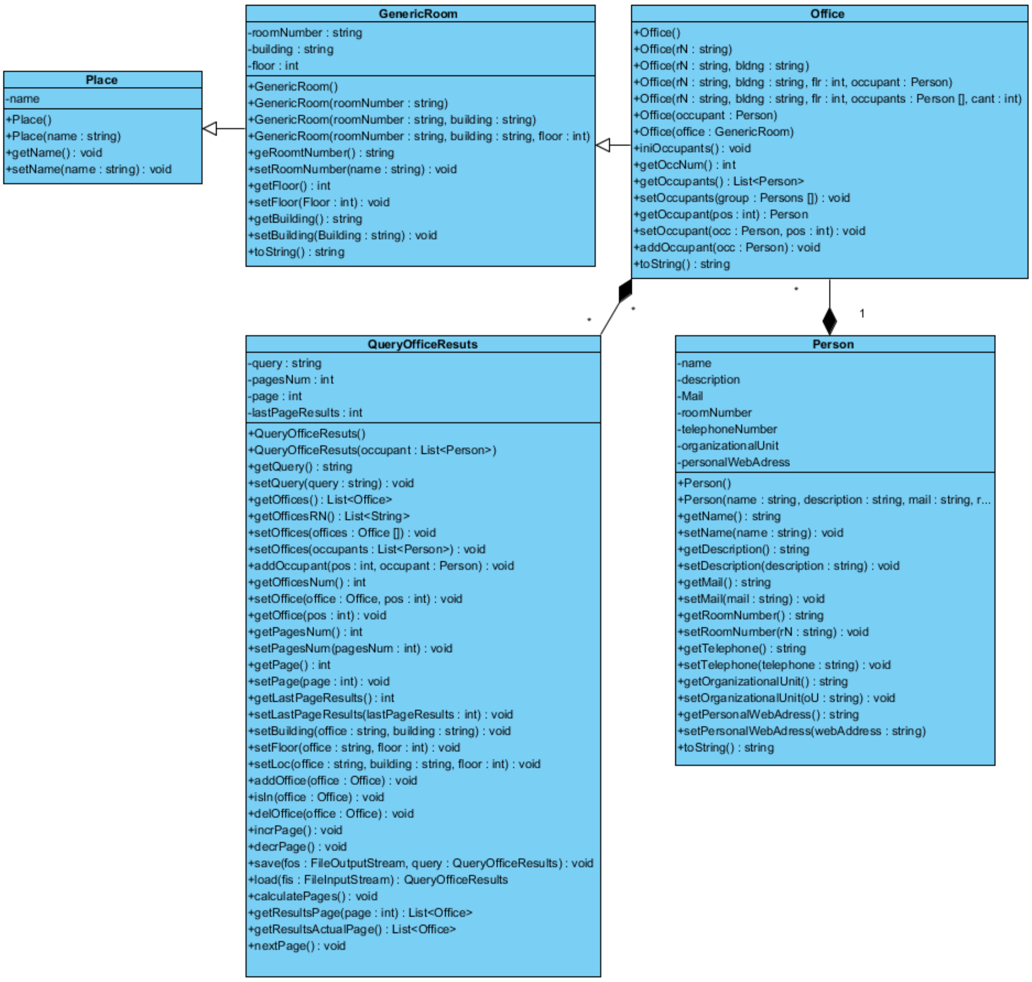


Figure 4.1.: UML use case diagram



GenericRoom
-roomNumber : string
-building : string
-floor : int
+GenericRoom()
+GenericRoom(roomNumber : string)
+GenericRoom(roomNumber : string, building : string)
+GenericRoom(roomNumber : string, building : string, floor : int)
+geRoomtNumber() : string
+setRoomNumber(name : string) : void
+getFloor() : int
+setFloor(Floor : int) : void
+getBuilding() : string
+setBuilding(Building : string) : void
+toString() : string
Office
+Office()
+Office(rN : string)
+Office(rN : string, bldng : string)
+Office(rN : string, bldng : string, flr : int, occupant : Person)
+Office(rN : string, bldng : string, flr : int, occupants : Person [], cant : int)
+Office(occupant : Person)
+Office(office : GenericRoom)
+iniOccupants() : void
+getOccNum() : int
+getOccupants() : List<Person>
+setOccupants(group : Persons []) : void
+getOccupant(pos : int) : Person
+setOccupant(occ : Person, pos : int) : void
+addOccupant(occ : Person) : void
+toString() : string
Place
-name
+Place()
+Place(name : string)
+getName() : void
+setName(name : string) : void
*
*
*
1
QueryOfficeResuts
-query : string
-pagesNum : int
-page : int
-lastPageResults : int
+QueryOfficeResuts()
+QueryOfficeResuts(occupant : List<Person>)
+getQuery() : string
+setQuery(query : string) : void
+getOffices() : List<Office>
+getOfficesRN() : List<String>
+setOffices(offices : Office []) : void
+setOffices(occupants : List<Person>) : void
+addOccupant(pos : int, occupant : Person) : void
+getOfficesNum() : int
+setOffice(office : Office, pos : int) : void
+getOffice(pos : int) : void
+getPagesNum() : int
+setPagesNum(pagesNum : int) : void
+getPage() : int
+setPage(page : int) : void
+getLastPageResults() : int
+setLastPageResults(lastPageResults : int) : void
+setBuilding(office : string, building : string) : void
+setFloor(office : string, floor : int) : void
+setLoc(office : string, building : string, floor : int) : void
+addOffice(office : Office) : void
+isIn(office : Office) : void
+delOffice(office : Office) : void
+incrPage() : void
+decrPage() : void
+save(fos : FileOutputStream, query : QueryOfficeResults) : void
+load(fis : FileInputStream) : QueryOfficeResults
+calculatePages() : void
+getResultsPage(page : int) : List<Office>
+getResultsActualPage() : List<Office>
+nextPage() : void
Person
-name
-description
-Mail
-roomNumber
-telephoneNumber
-organizationalUnit
-personalWebAdress
+Person()
+Person(name : string, description : string, mail : string, r...
+getName() : string
+setName(name : string) : void
+getDescription() : string
+setDescription(description : string) : void
+getMail() : string
+setMail(mail : string) : void
+getRoomNumber() : string
+setRoomNumber(rN : string) : void
+getTelephone() : string
+setTelephone(telephone : string) : void
+getOrganizationalUnit() : string
+setOrganizationalUnit(oU : string) : void
+getPersonalWebAdress() : string
+setPersonalWebAdress(webAddress : string)
+toString() : string

Figure 4.2.: UML class diagram

Table 4.3.: APPOverview

Within	**Out**
Search places by person	Search history
Search places by code	Share places with other devices.
Free navigation	Show routes
Select building and floor to see the interior map	Space management(labs and classrooms booking, etc.).
Favorites list	Search in favorites list
Show the own location	
Show compass	
Show place attributes	

Functionalities

From the beginning certain functionalities were planned that the app must be present. During the development some of them were shifted by others more attractive, easier to implement or simply more user-friendly. Others disappeared from the list due to lack of development time, or because they still do not have the necessary data (or web services that can provide them).

Anyway, these are the features that finally presents the application. Some of the rejected functionalities are included in the section "Future work".

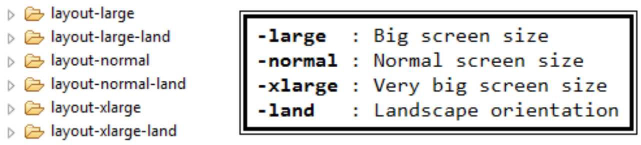
User interface

The UI is one of the most important aspects of the app. In the development of mobile applications certain facets of the user interface must be cared in detail, because the target users' experience is critical to the success of the application.

From the beginning the user interface has been attempted to design in a way that respect to the maximum how already being used similar applications. This was just to ensure that a large ratio of future users already know certain mechanisms of the interface, and its use will be as natural as possible.

After each feature implementation and test the UI was revised and improved. The app UI evolution is explained in the apendix A.

The design takes into account the different screen sizes and configurations presents in android devices. Every activity has different layouts for every combination of screen size and orientation. The figure 4.3 shows how the layouts are clasified.



- layout-large
- layout-large-land
- layout-normal
- layout-normal-land
- layout-xlarge
- layout-xlarge-land

```
-large  : Big screen size
-normal : Normal screen size
-xlarge : Very big screen size
-land   : Landscape orientation
```

Figure 4.3.: Layouts for every screen size and orientation

The final version of the user interface design can be seen schematized in the figures below. The schema 4.4 shows the layouts for the results visualization, used to display favorites too. The schema 4.5 shows the navigate interface. The other layouts are represented in the schema 4.6.

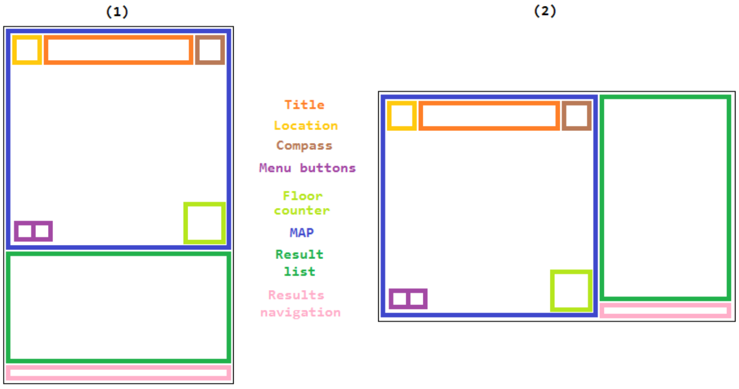


Figure 4.4.: Layouts of results and favorites, portrait (1) and landscape (2) orientation.

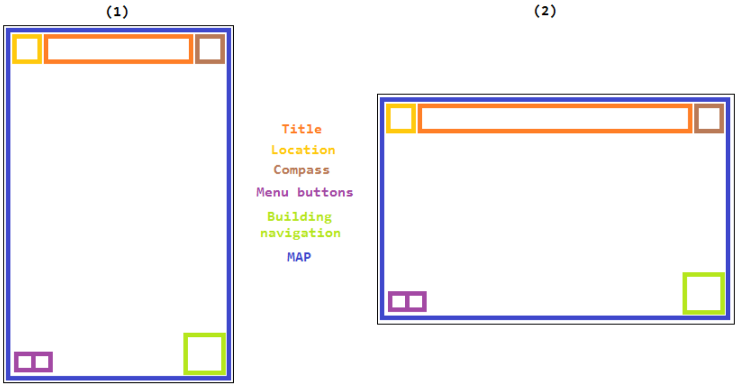


Figure 4.5.: Layouts of navigation, portrait (1) and landscape (2) orientation.

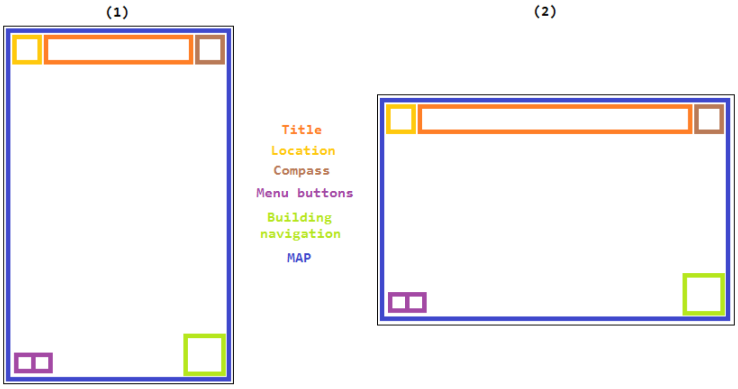


Figure 4.6.: Layouts of the activities.

The app uses different images to display significant information. The figure 4.7 is a representative collection of this imagery. The images (1) and (2) are the logos of the app, (3) is the compass and (4) is an array of images to represent the different floors of a building.

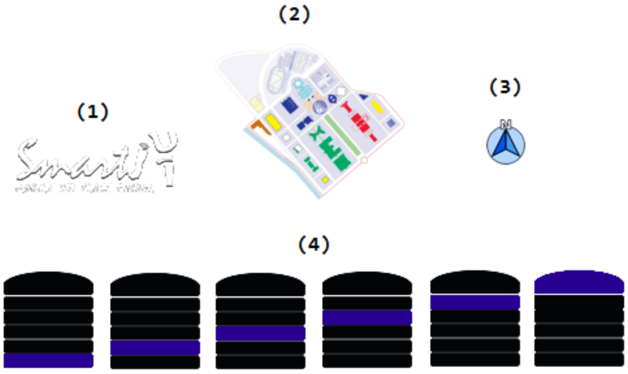


Figure 4.7.: Collection of images used in the app.

The figure 4.10 shows a collection of buttons, designed specifically for our app. The buttons of the first group are navigation buttons, the second group belongs to the menu buttons, and the third group are buttons that call functions.

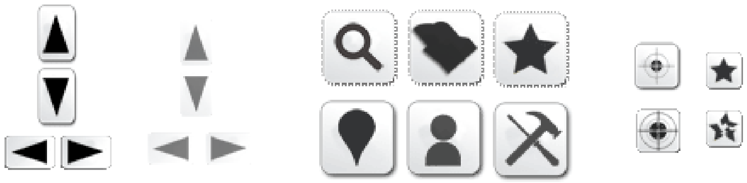


Figure 4.8.: Collection of buttons used in the app.

Finally the figure 4.9 shows the buttons to display the results of a search or the favorite list. The first one is the button when it is disabled, the second one is the look when it is enabled and the third one is when it is selected.

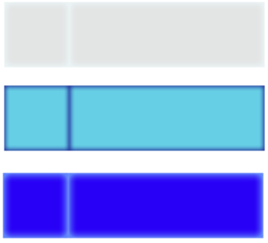


Figure 4.9.: Buttons for the results list.

4.2.2. Implementation

Using the tools described in 3.2, and taking into account the UML generated in first instance, the app's prototype development was initiated.

The implementation follows the same order in each iteration. Firstly to prepare the activities to obtain the general structure of the app. After this to create or modify the classes to be used by the activities, to give them functionalities. The third step is to include asynchronous tasks to perform actions that require asynchronous responses. The last step of every iteration is to adapt the user interface to the written code.

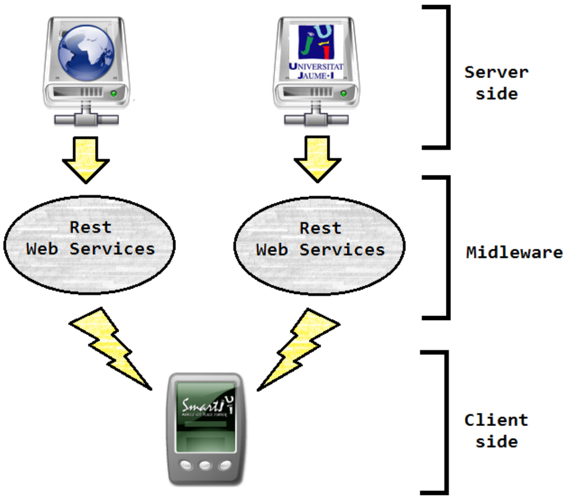


Figure 4.10.: Collection of buttons used in the app.

The sections below show the most important aspects of the implementation during these steps.

Table 4.4.: Used activities description

Activity	Intent extras	Description
TitleActivity	none	The first screen displayed. It has the name of the app and the logos. As the text indication say, the user must touch the screen to continue.
NavigateActivity	none	It is the second screen to display. In this one the user can navigate over the university map, and using the controls situated at the right part of the bottom of the screen the user can choose the building and the floor to be displayed.
MenuActivity	none	This one is used when a search will be started. It shows a menu to choice one between 3 search options, and allows to access to the appropriate form.
PFormActivity	use(String) and criteria(String)	Depending on the extras received from previous activities, this one shows a different forms. The user can search in the UJI's directory and show the search results in the next activity.
ResultsActivity	use(String)	The received extra indicates which function perform the activity. If it has the value "res" the activity shows the result of a search, but if the value is "fav" the activity shows the favorite list.
InfoActivity	origin(String) and Office(Office)	Shows full information of a location.

Activities

The activities are one of the most important points of the Android app development. These kinds of objects represent the things that the users can do.

Normally one activity is presented as full-screen windows. The selection of our activities has been given by the list of features to implement. The table 4.4 shows our activities, what additional information receive from other activities and a brief description of their purpose. The figure 4.11 shows a diagram with the main workflow of the app.

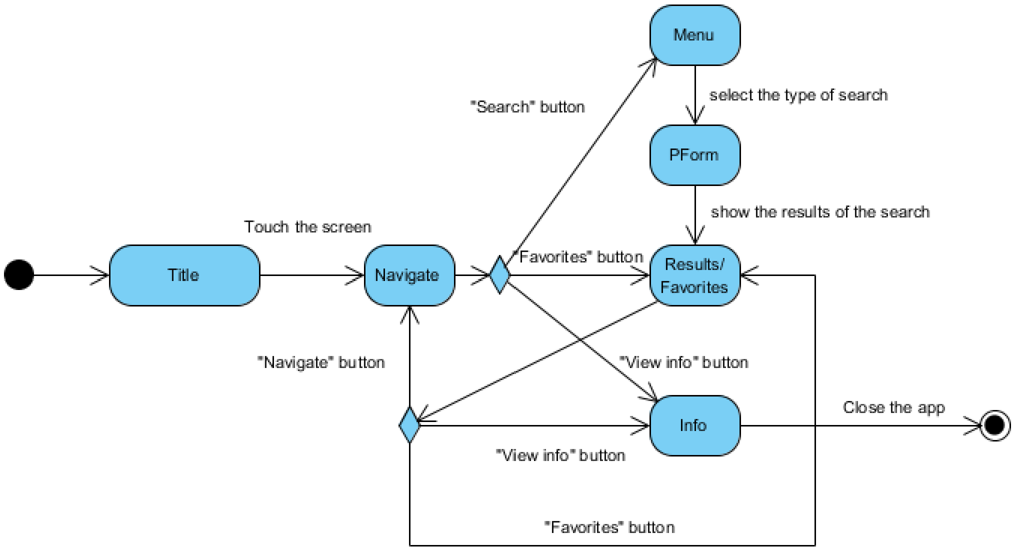


Figure 4.11.: Activities workflow.

Classes

The development of our app tries to take full advantage of the possibilities offered by the used technologies. The Object-Oriented Programming (OOP) provides a lot of benefits, such as encapsulation and inheritance.

The included classes were selected using as a guide the UML diagrams in section 4.2.1 (specifically the Class diagram of the figure 4.2). In addition to the classes shown in the diagram, other classes have been added to the application from the different functions that have been generated during development.

Tasks

The queries to different servers using the web services need a certain time of response. Asynchronous tasks run on different threads, and when the information requested is served the execution of these threads ends. A good example of the inner workings of the application for a search can be described in figure 4.12.

The tasks to be done asynchronously are those that serve data to be used by other functions of our application. Those tasks are described in the table 4.6.

Table 4.5.: Used classes description

Classes	Description
GenericRoom	Defines the attributes of a generic room. It is the parent class of "Office", and it is included to be part of future implementations could be other classes as "Classroom", "Lab", "canteen", etc..
Office	Defines the attributes of an office. This class is the centeron which rest the most searches that our application performs.
Person	The occupants of an office are deffined by this class. This is an important part of the class Office, since the user may want the data of the persons occupping an office.
Place	This is the parent class of Generic room. The class is another one to take into account in future developements, when other places different of rooms will be included in the searchs.
QueryOfficeResults	This class defines the results of a search.
SMLParser	This class provides the functions to parse.

Table 4.6.: Tasks description

Classes	Belongs to activity...	Description
FloorTask	NavigateActivity	Receives a building and a floor, and return a graphic layer with all the features with the received building and floor.
NavigateTask	NavigateActivity	Receives an Uniform Resource Locator (URL) (to search by room number), and search its information in the UJI's directory. This information is used to fill the corresponding callout and to show it.
OfficesTask	PFormActivity	Receive an URL to search in te UJI's directory by one of the criteria of the form. Return a list of persons who meet the requirements.
RoomTask	ResultActivity	Receives a building, a floor and a room id, and returns two graphic layers: one with all the features with the received building and floor, and the other with the selected room.

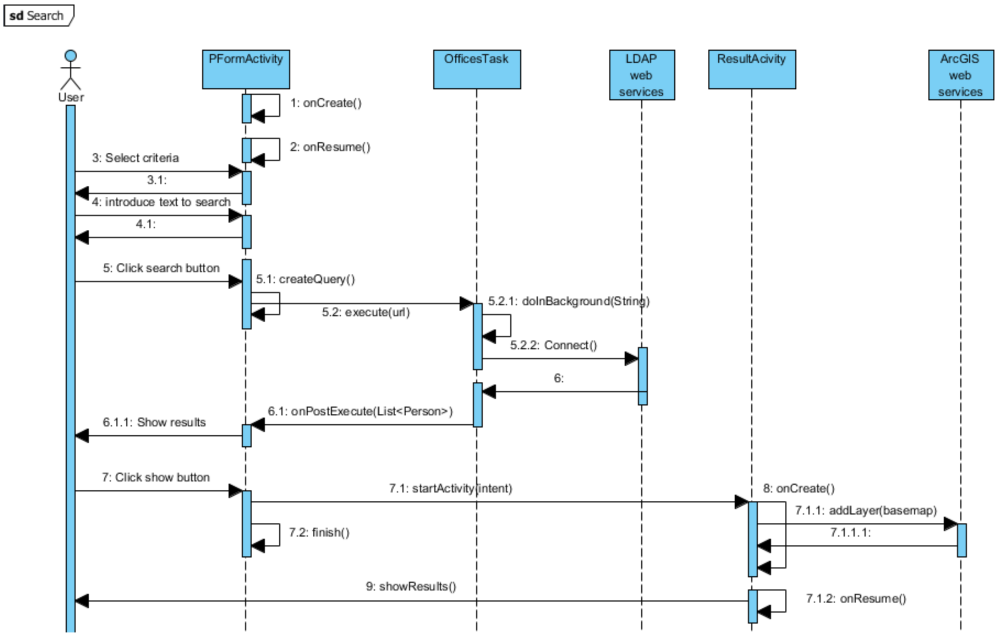


Figure 4.12.: Sequence diagram of a search.

UI

The UI in Android is defined by the app resources. The images to use, the layouts for the activities, the values to show information to the user and so on. All of these are encompassed in what we call resources. The list below shows the resources used in our app.

- Drawables:
  - **Logos:** The app logos to reinforce the brand image.
  - **Buttons:** The imagery for the appearance of the buttons.
  - **Compass:** The compass image.
- Layouts:
  - **activity_info.xml:** The layout shows the info of a place.
  - **activity_menu.xml:** The layout shows the search menu.
  - **activity_navigate.xml:** The layout shows the free navigation screen.
  - **activity_results.xml:** The layout shows the results of a search or the favorite list.

  - **activity_search.xml:** The layout shows the form of search.

  - **activity_title.xml:** The layout shows the title page.

  - **callout.xml:** The layout for the ArcGIS map callout.

- Values:

  - **colours.xml** The values of the colours used in the app.

  - **menus.xml:** The arrays of strings to show in the menus.

  - **strings.xml:** The strings used to show to the user.

The drawables designs were made with Adobe Photoshop CS3, taking into account the image file sizes, the properties of the designs and the purpose of each one.

The layouts are defined for normal, large and extra large screens, and for both orientations (landscape and portrait). These ensure correct viewing on most mobile devices.

The values are defined for English, Spanish and Catalan. This means that when a mobile is configured in any of these languages the app shows the information in the right language (English is the default option for other languages).

5. Tests and results

This chapter shows the key aspects of the tests undertaken and the results of the development.

The test stage is very important in the development of applications. It debugs the functionalities, the performance and the user interface, making the app evolves to achieve the desired results.

In our app development the test phase has been distributed through the whole process, doing user testing after each iteration mentioned in the section 2.1 .

5.1. Test description

The test phase consists of a battery of processes to check on the performance of the functionalities implemented. All the interim apps produced by iterations are tested by the developer, and some of them are used in guided exercises with real users.

Guided exercises have consisted in perform different actions with the application, noting all the reactions, comments and suggestions from the user. Notes taken during the guided exercises with real users help us to see more clearly the flaults in the design of the user interface, allowing changes in the following iterations.

The developer testing is essential to check if all the functions of the program are well-done, and make changes and improvements to the code immediately.

5.2. Test results

The results of the testing have helped to improve the app in key areas such as performance, proper functioning and user friendly environment. Some improvements in the UI that can be found in Appendix A have been triggered by the test results. Also capabilities have been found to implement in future improvements.

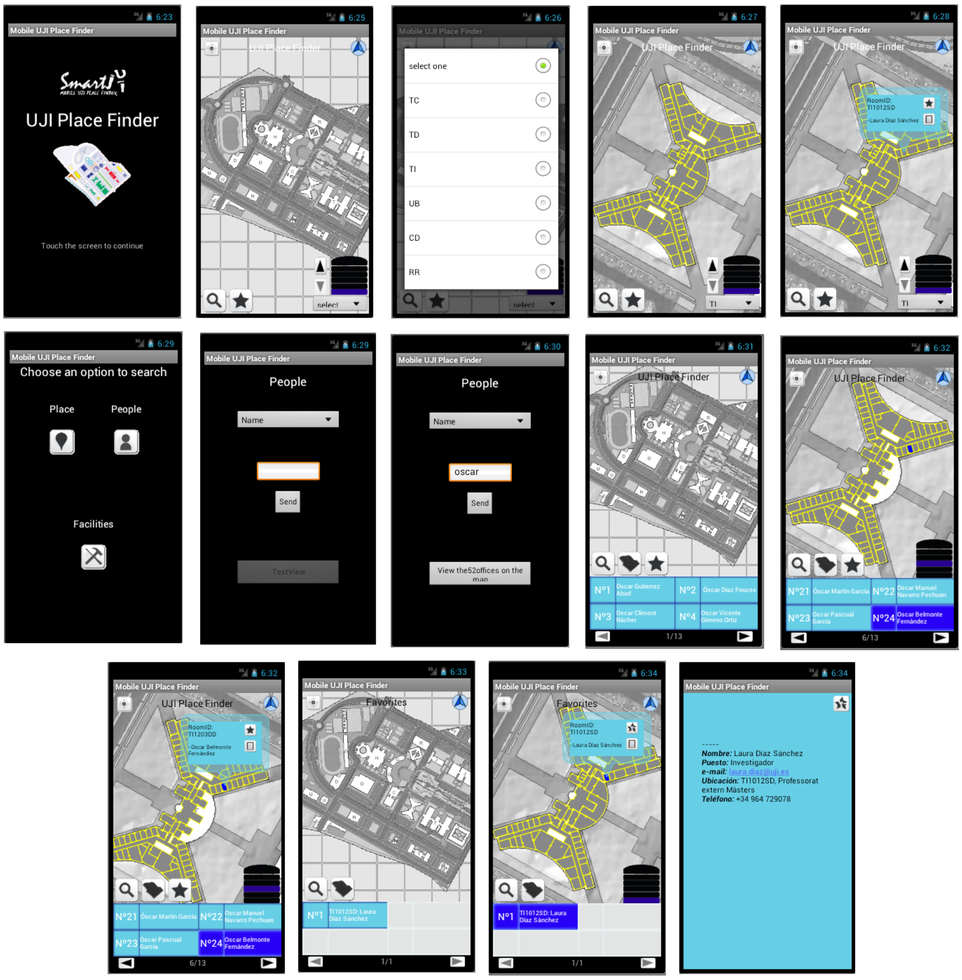
During testing the importance of completed and updated layers has revealed. Issues were found when the UJI's directory provided information of offices inside buildings that have not been included in the used layers. Fortunately there are people working to complete digitalization of UJI maps and inclusion of missing data in the layers we use, and our app will work perfectly when this occurs.

5.3. Resulting app

With the last iteration of the development has produced an application to find places inside the UJI campus. The app include more functionalities than a place finder: the user can store his favorite places, can freely explore the campus map and search the contact information of everyone in UJI's directory.

The application has been made to be easily improved. The section 6.2 deepens widely on possible improvements that could be implemented in the future.

The figure 5.1 shows a summary of the app with screenshots. Development outcomes of the application can be reviewed on the website accessible from here.



Nombre: Laura Diaz Sánchez
Puesto: Investigador
e-mail: laura.diaz@uji.es
Ubicación: TI1012SD, Professorat extern Màsters
Teléfono: +34 964 729078

Figure 5.1.: Screenshots of the app

6. Conclusions and Future work

6.1. Conclusions

The result of this work is a very useful Android application that can search the UJI's directory and show the results on a map. It can be used to search and browse the UJI map and to guide users within the campus. As byproducts of app development we find web services that allow us to search in UJI's directory.

The use of technologies such as ADT Bundle to streamline the Android app development, the ArcGIS SDK for Android with ArcGIS servers and RESTful web services to access information stored on different servers, have greatly facilitated the achievement of this work.

There are a lot of possibilities for improvement, and some of them can be done using resources that are already available. Future improvements will be able to use the resources offered by the smart campus project, but they will also be able to use the sensors available on most mobile devices. The sensors added to future mobile devices can be used without problems and easily on future improvements.

Besides all of this, the mobile place finder can also be exported to other campuses or cities with the same services. This would be possible by simply changing a few lines of code to use similar resources located on other servers.

6.2. Future work

6.2.1. Show more info

The information about tutoring schedules if the finded place is an office, or occupation schedules if is a classroom or a laboratory. A good improvement would be to include a system for sending messages to people found in UJI's directory.

6.2.2. Search other places

Now the app can search rooms by their numbers, and the offices of people who are part of the UJI directory.

The app can include more feature layers to show other kind of places: toilets, canteens, UJI services headquarters, etc. And, of course, functionalities to search those places can be implemented.

6.2.3. Indoor positioning

The app can get the position of the device that is running, but only uses Global Positioning System (GPS) and does not work well indoors. A good improvement would be to include the indoor positioning, there are many technological options for doing so.

6.2.4. Search routes

The app can show a searched place in the map, and can show the own position. The next step is show the best route to connect both using a geoprocess web service for calculate routes developed for the UJI campus.

The routing possibilities don't finish here: search the nearest place of a determinate kind, calculate the time to spend going between two places or organize the class schedule taking into account the classes location and the time to go from one class to another.

6.2.5. Occupancy

The saturation of different public areas can be interesting for the users of our app. Areas as canteens, study rooms, free access computer rooms or parking areas are sensitive to being targeted by this service.

This information can be obtained by diverse methods [18] [14] as computer vision [19] [15], infrared beams or thermal imaging, and can be served by a web service for use by mobile and web applications.

The app map can show the searched places with different colours depicting its occupation and the value that this represents.

6.2.6. Other improvements

Other languages can be included, and even new screen sizes if necessary. UI, code clarity and efficiency of the functions are also subject to refinements.

Mixing several future improvements can be achieved more concrete things: given the location of the bars, the time needed to get from one classroom to them and saturation that occurs in a given time or the statistics of their occupation at a certain time, you can decide in which of all you have time enough for lunch.

A. Appendix - UI evolution

The app UI has undergone many changes since its first design until the development of the app has completed. The app structure has been modified, and the activities interfaces have been significantly enhanced (especially those including maps). That evolution will be explained in the next sections.

A.1. App structure

The first version of the app (structure showed in the figure A.1) uses a menu after the title page to allow the access to the different actions. The navigation between screens after the first action is not highly regarded during user testing.

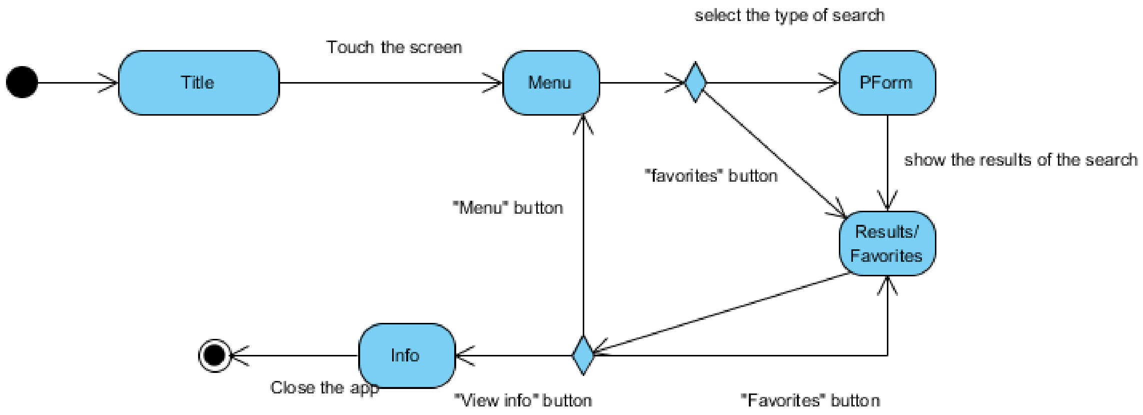


Figure A.1.: First app structure aproach

The second version includes an activity to explore freely the UJI campus. Managing the application gets complicated. The figure A.2 shows it.

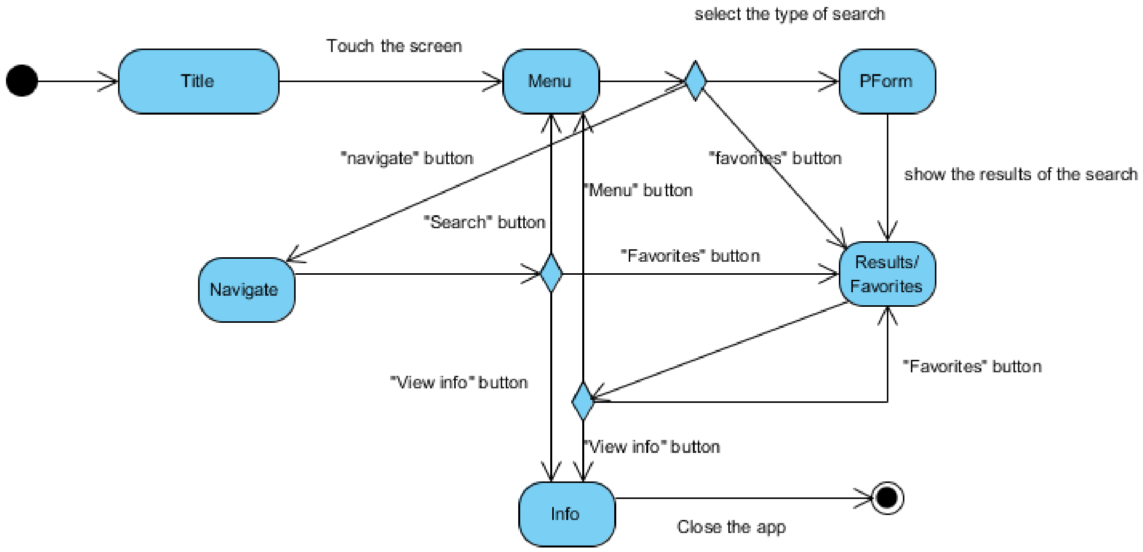


Figure A.2.: Second app structure aproach

Successive improvements of the internal structure produce an app usage more natural and intuitive, requiring a lower number of user actions to perform the most common tasks. The final app structure is showed in the figure A.3.

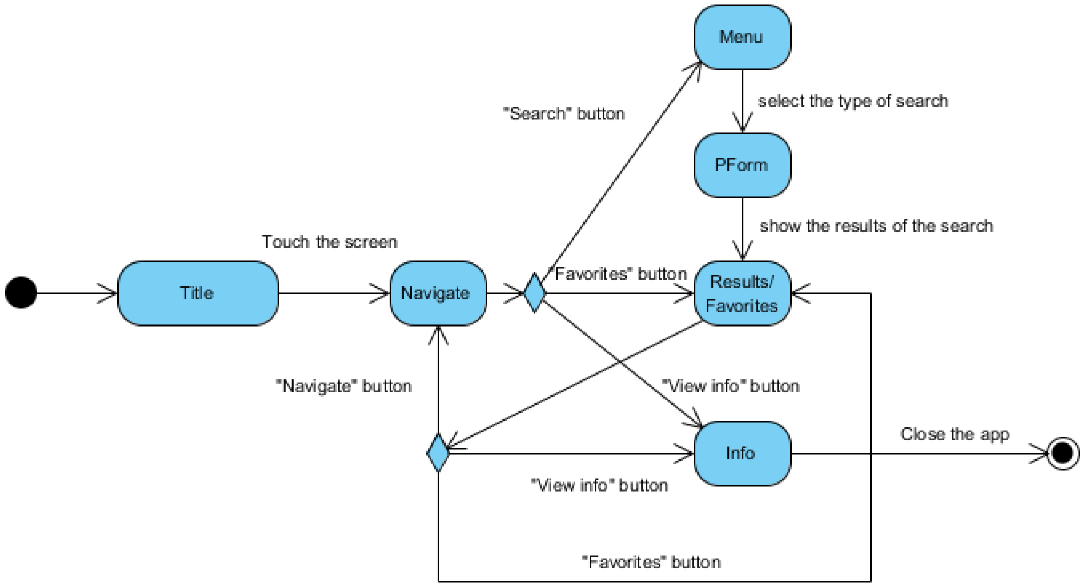


Figure A.3.: Final app structure aproach

A.2. Map screens

Screens with maps of our app was also designed to host search forms so that it was not necessary to change of activity to enter the search criteria, as show in figure A.4.

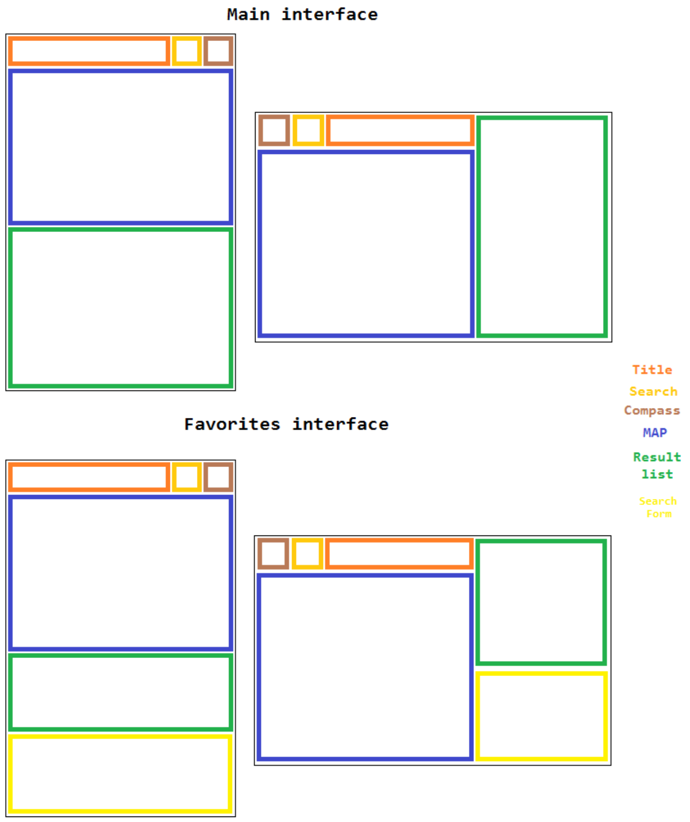


Figure A.4.: First UI aproach

This has proven to be counterproductive because the space occupied by the form directly took it off the map display, which is unacceptable in an application that has as main objective to locate on the map the results of a search. In addition once adopted this improvement the app can use the same interface to display the results of a search and to list the favorite places, thus saving resources. The figure A.5 shows this design.

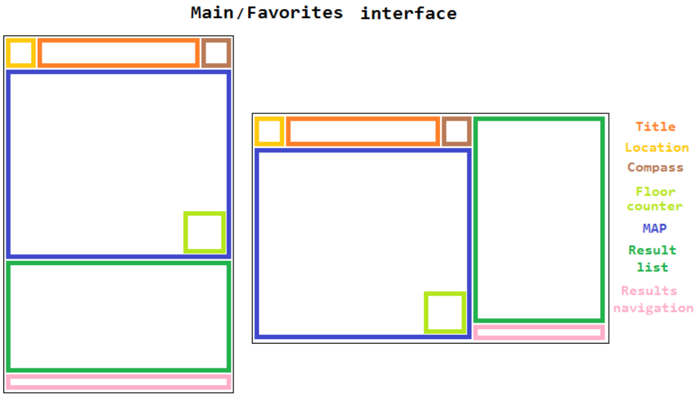


Figure A.5.: Second UI aproach

After the removal of the form from the map screens, the next improvement is to add buttons that will serve as menu navigation among the options offered by the program. These buttons have been added over the map, so its inclusion does not give the feeling of losing display space. The final UI design is represented in the figure A.6

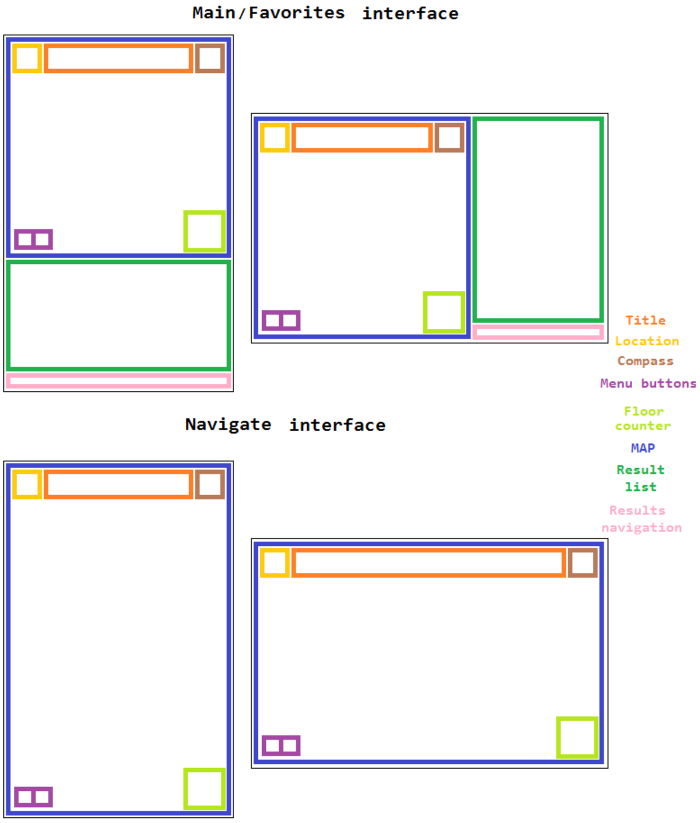


Figure A.6.: Final UI design

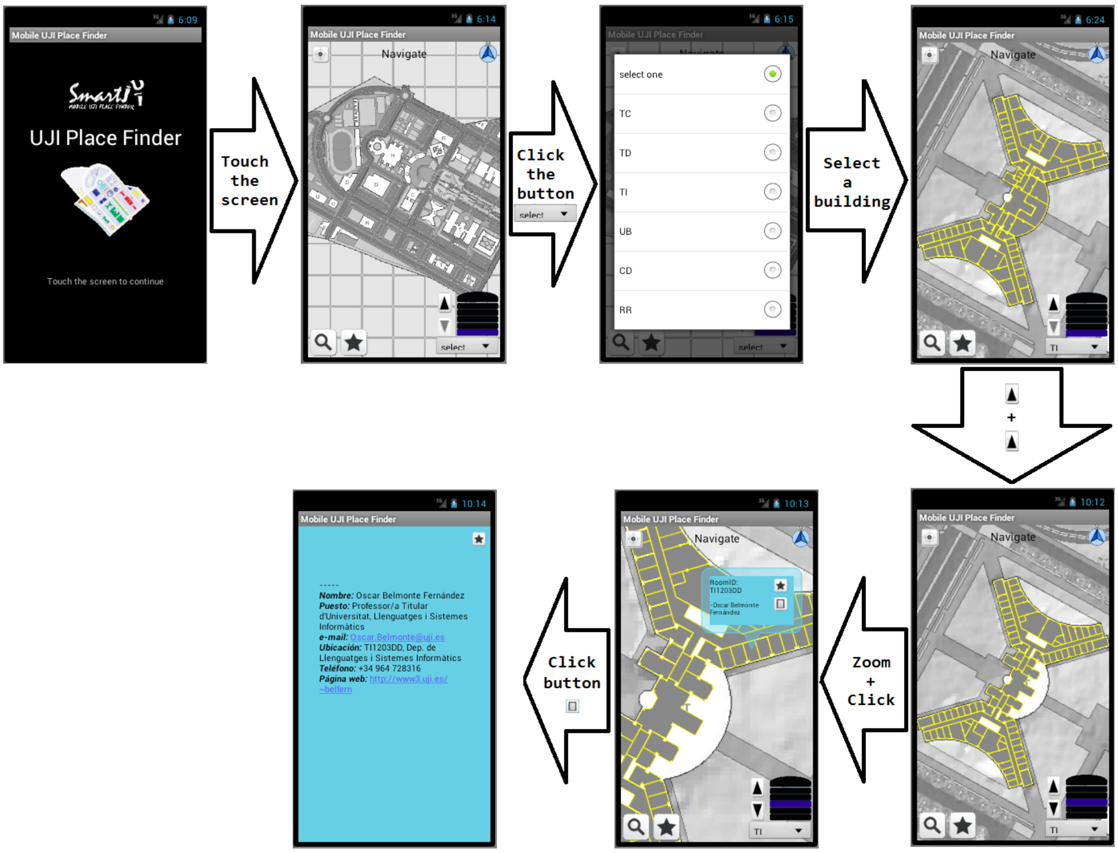
A.3. Images and buttons

Images and buttons have also been improved. The users were consulted to assign icons to buttons depending on the action to perform. During the tests these icons have changed as directed by users.

B. Appendix - User guide

B.1. Free navigation

The figure B.1 shows the sequence of actions to be carried out to see information for a specific room using free navigation.

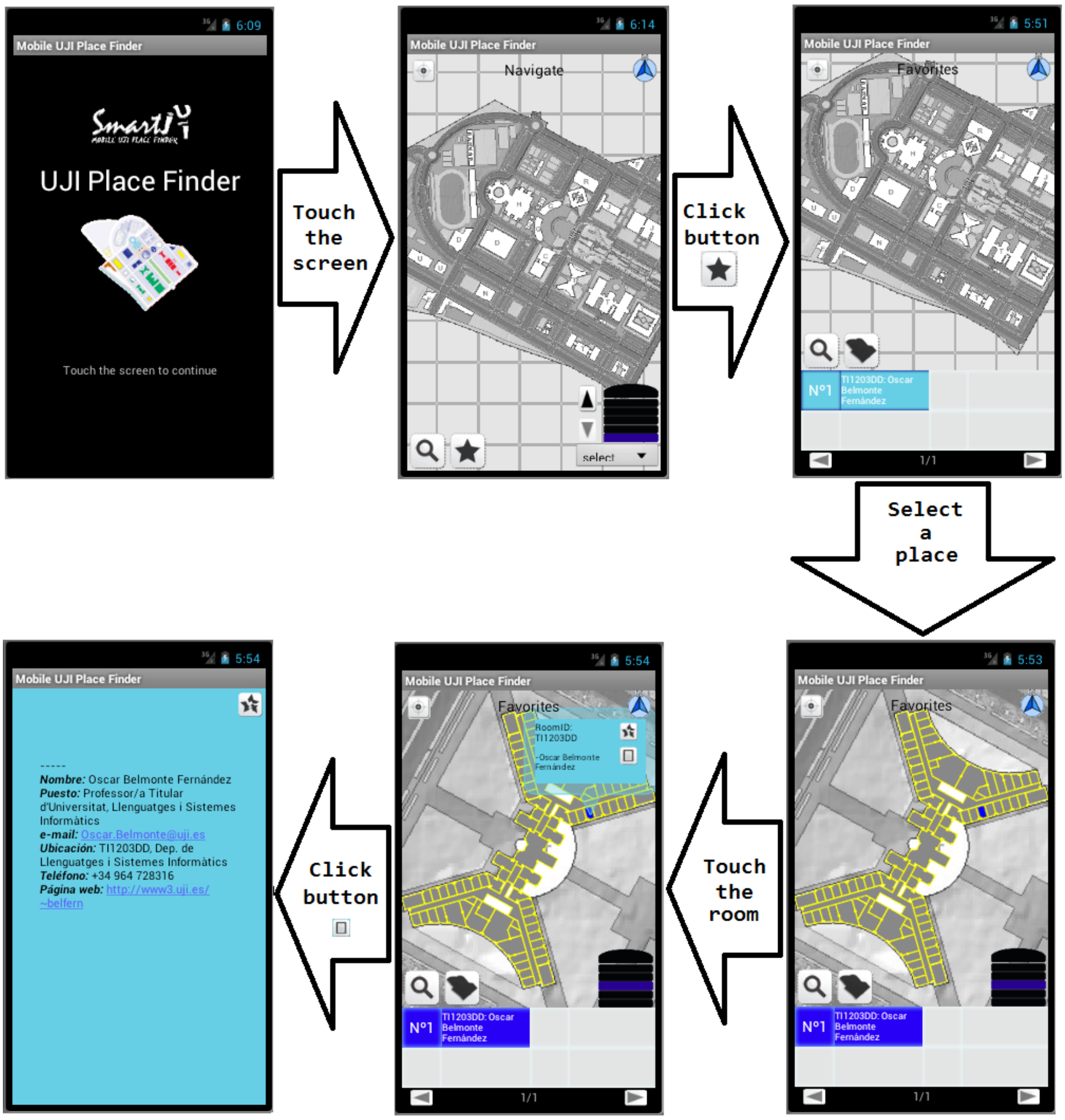


Nombre: Oscar Belmonte Fernández
Puesto: Professor/a Titular d'Universitat, Llenguatges i Sistemes Informàtics
e-mail: Oscar.Belmonte@uji.es
Ubicación: TI1203DD, Dep. de Llenguatges i Sistemes Informàtics
Teléfono: +34 964 728316
Página web: http://www3.uji.es/~belfern

Figure B.1.: Sequence of a free navigation action.

B.2. Favorites

The figure B.2 shows the sequence of actions to be carried out to see information for a specific room of the favorites list.

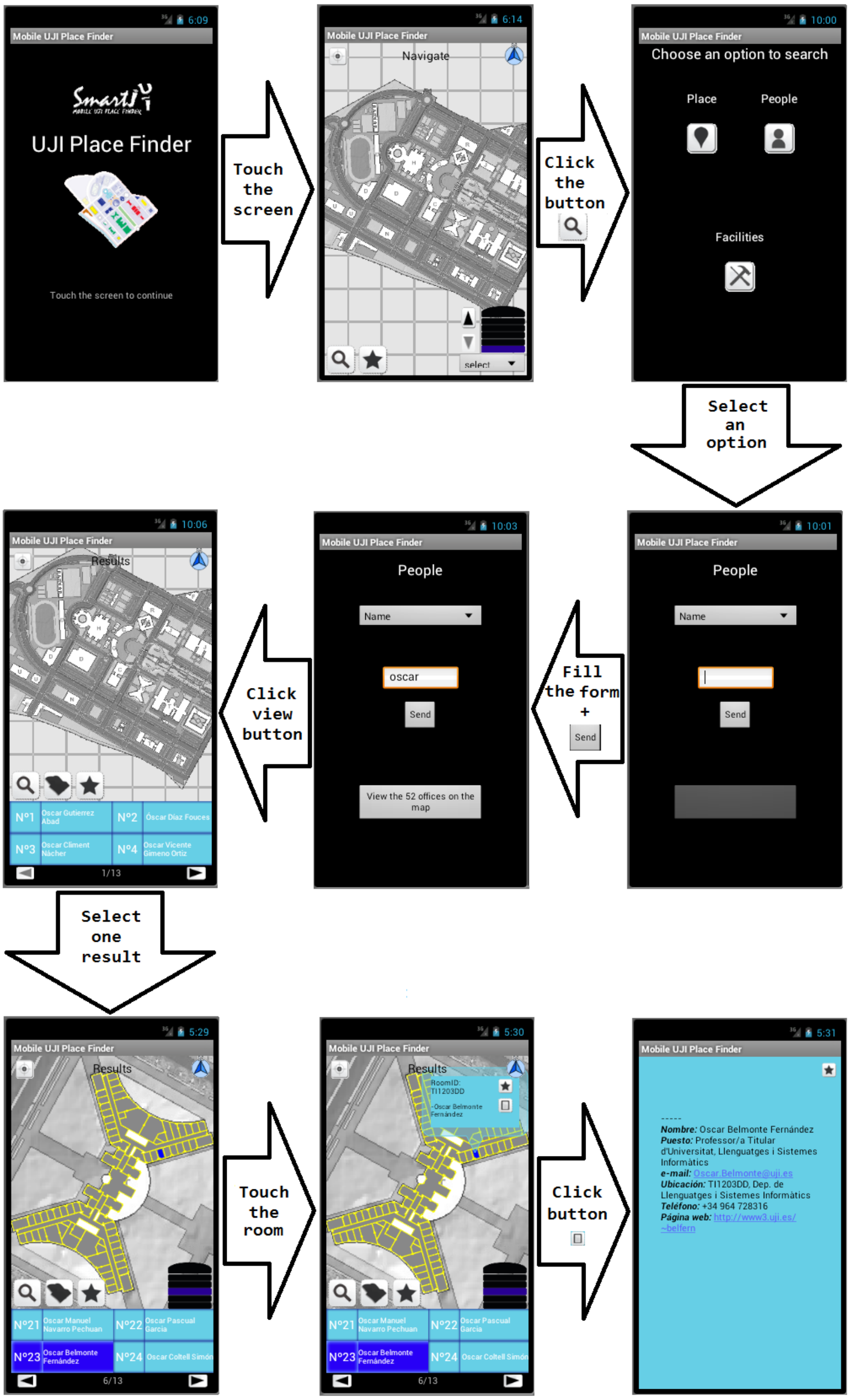


Nombre: Oscar Belmonte Fernández
Puesto: Professor/a Titular d'Universitat, Llenguatges i Sistemes Informàtics
e-mail: Oscar.Belmonte@uji.es
Ubicación: TI1203DD, Dep. de Llenguatges i Sistemes Informàtics
Teléfono: +34 964 728316
Página web: http://www3.uji.es/~belfern

Figure B.2.: Sequence of a favorites navigation action.

B.3. Search a room

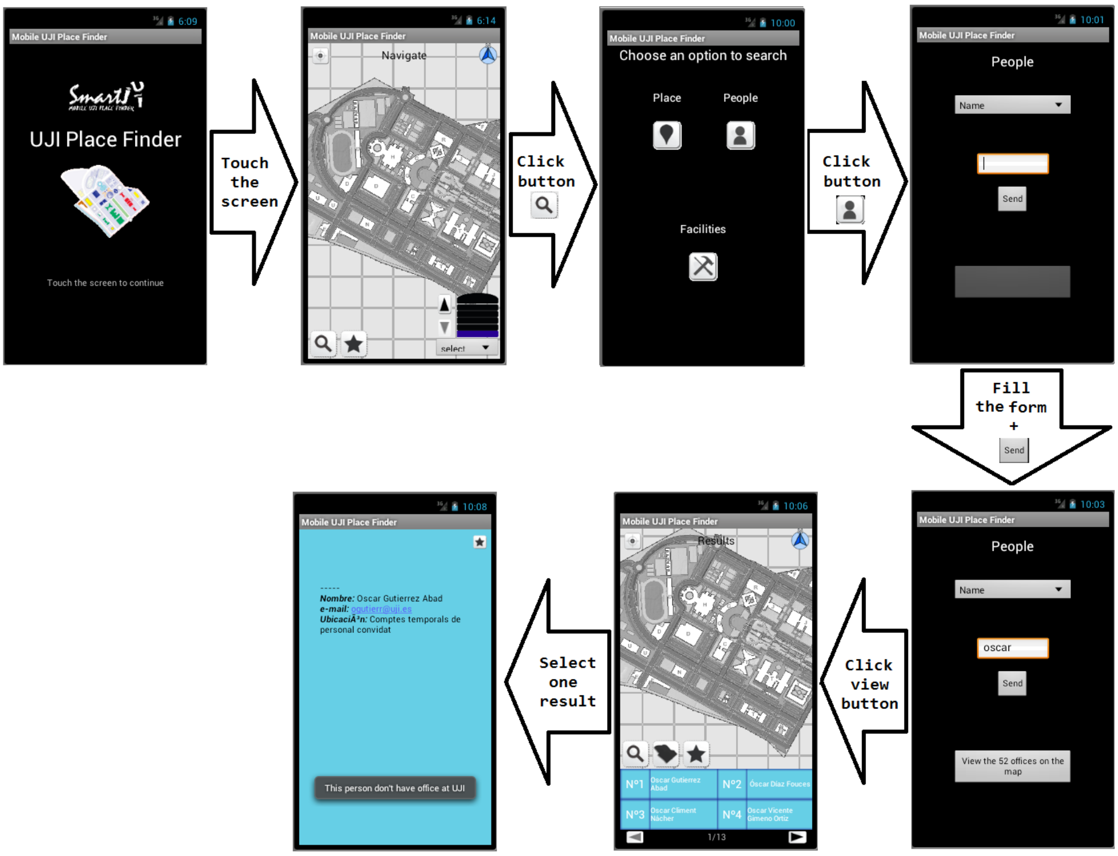
The figure B.3 shows the sequence of actions to be carried out to see information for a specific room using the search form.



Nombre: Oscar Belmonte Fernández
Puesto: Professor/a Titular d'Universitat, Llenguatges i Sistemes Informàtics
e-mail: Oscar.Belmonte@uji.es
Ubicación: TI1203DD, Dep. de Llenguatges i Sistemes Informàtics
Teléfono: +34 964 728316
Página web: http://www3.uji.es/~belfern

Figure B.3.: Sequence of a search action.

If you select a person who does not have an office on campus will display the information of this person, instead of showing the location of an office on the map. This is shown in figure B.4.



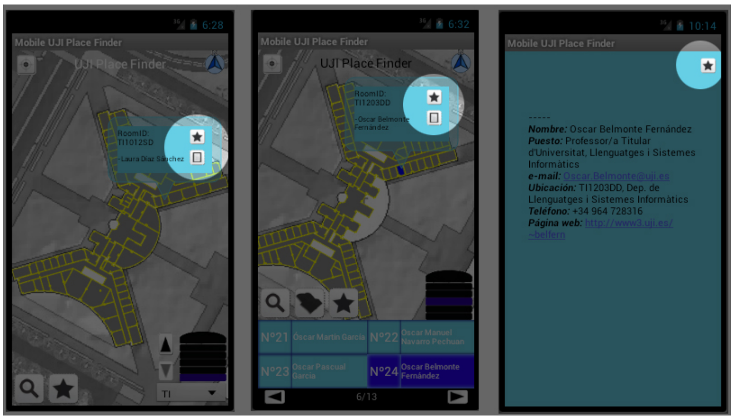
Nombre: Oscar Gutierrez Abad
e-mail: ogutierr@uji.es
UbicaciÃ³n: Comptes temporals de personal convidat

This person don't have office at UJI

Figure B.4.: Sequence of a search action.

B.3.1. Add a place

The user can add a place to the favorite list using the "add to favorites" button, present both in the bubble of information displayed on the map when one select a room in navigation mode or when one select one of the results of a search, and in the display of information about a place. The figure B.5 shows where the user can find the "add to favorites" button.

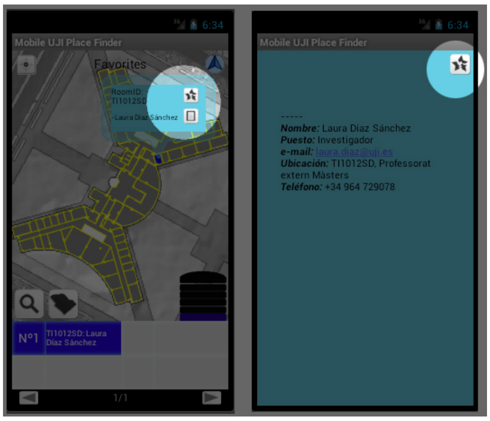


Nombre: Oscar Belmonte Fernández
Puesto: Professor/a Titular d'Universitat, Llenguatges i Sistemes Informàtics
e-mail: Oscar.Belmonte@uji.es
Ubicación: TI1203DD, Dep. de Llenguatges i Sistemes Informàtics
Teléfono: +34 964 728316
Página web: http://www3.uji.es/~belfern

Figure B.5.: "Add to favorites" button location.

B.3.2. Delete a place

The user can delete a place from the favorite list using the "delete from favorites" button, present in the bubble of information displayed on the map when one select a room from the favorite list and in the display of information about a place. The figure B.5 shows where the user can find the "delete from favorites" button.



Nombre: Laura Diaz Sánchez
Puesto: Investigador
e-mail: laura.diaz@uji.es
Ubicación: TI1012SD, Professorat extern Màsters
Teléfono: +34 964 729078

Figure B.6.: "Delete from favorites" button location.

Bibliography

[1] Adt bundle. `http://developer.android.com/sdk/installing/bundle.html`. [Internet; downloaded 12-February-2013].

[2] Android developers - supporting different devices. `http://developer.android.com/training/basics/supporting-devices/index.html`. [Internet; downloaded 19-November-2012].

[3] Apache maven. `http://maven.apache.org/`. [Internet; downloaded 12-February-2013].

[4] Arcgis runtime sdk for android. `http://resources.arcgis.com/content/arcgis-android/sdk`. [Internet; downloaded 25-November-2012].

[5] Cyanogenmod webpage. `http://www.cyanogenmod.org/`. [Internet; downloaded 19-January-2013].

[6] Github - online project hosting using git. `https://github.com/`. [Internet; downloaded 12-February-2013].

[7] Help for arcgis runtime sdk for android. `http://help.arcgis.com/en/arcgismobile/10.0/apis/android/help/`. [Internet; downloaded 25-November-2012].

[8] Restfull web services: The basics. `http://www.ibm.com/developerworks/webservices/library/ws-restful/`. [Internet; downloaded 18-November-2012].

[9] Texworks project page. `http://www.tug.org/texworks/`. [Internet; downloaded 12-February-2013].

[10] Uji web site - servei de directori ldap. `http://www.uji.es/CA/serveis/si/ldap/infoldap.thtml`. [Internet; downloaded 15-January-2013].

[11] Uji's geotec arcgis server - arcgis rest services directory. `http://geotec.dlsi.uji.es:6080/arcgis/rest/services`. [Internet; downloaded 15-January-2013].

[12] Wikipedia - html5. `http://en.wikipedia.org/wiki/HTML5`. [Internet; downloaded 12-February-2013].

[13] Wikipedia - representational state transfer. `http://en.wikipedia.org/wiki/Representational_state_transfer`. [Internet; downloaded 18-November-2012].

[14] Wikipedia - representational state transfer. `http://en.wikipedia.org/wiki/People_counter`. [Internet; downloaded 26-November-2012].

[15] Jorge Badenas and Filiberto Pla. Applying computer vision techniques to traffic monitoring tasks. *Dept. Informatica, Universitat Jaume I, Castellon*, 1998.

[16] Martin Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley, 2003.

[17] Craig Larman. *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley, 2004.

[18] Jim Sinopoli. Count me in: People counters, a simple system for esential building data. *AutomatedBuildings.com*, 2009.

[19] Thiago Teixeira and Andreas Savvides. Lightweight people counting and localizing in indoor spaces using camera sensor nodes. *Electrical Engineering Department, Yale Univerisity*, 2007.