



Medical Center

DESIGN DOCUMENT

version 1.0



Diego ROTA, 841344

Simone MONTALTO, 841359

Politecnico di Milano, A.A. 2016 - 2017

Prof. Luciano BARESI

POLITECNICO MILANO 1863

Medical Center

Design Document

Medical Center application described below is thought, designed and implemented to meet the need of developing a solution for the progress of the public systems and in particular of the Italian healthcare system. In this document is provided a complete description of the design phase of the app, including all the architectural choices, use case models and sequence diagrams. The organization of an hospital and the needs of the patients have been taken into account to understand how to optimize the workflow of doctors and nurses and to improve the quality of patients life. The idea is to propose a mobile application that helps patients to handle the hospitalization phase. In particular, four areas of interest have been taken into account. Nowadays an hospitalized patient can't control his situation. He has to ask doctors or nurses in order to have information about his next examination or prescription. So it has been thought a service application which helps patient to have a granular control about its hospitalization. The purpose of *Medical Center* is to have an overview on the next examinations and prescriptions of each patient. Furthermore, the user can manage lunch and dinner menu or browse within the Points Of Interest of the structure, in order to easily found the right place and move to it through the integrated navigator. Thanks to the map, the patient can also know where he is exactly located while the news section provide useful informations. With these features, *Medical Center* aims to be a solution to the digital problem in a public structure like an hospital.

1.0.1 Stakeholders

Three categories of stakeholders have been identified:

- **Advisors of Politecnico di Milano:** people most interested in the design and implementation phase of the project. They had to analyse these two aspects, evaluating their realization.

- **Hospital Manager:** he is the person who has commissioned the job. He must evaluate the impact that this system has on the hospital's workflow.
- **Patients:** they are the final users of the application. *Medical Center* must be realized keeping in mind that it should be easy to use and intuitive, in order to help patients to retrieve easily useful informations and manage their hospitalization.

1.1 Introduction

1.1.1 Purpose and goals

This document represent the Requirement Analysis and Specification Document (RASD) and the Design Document (DD), with the objective to have an high-level description of the system that will be designed. In the following sections will be analysed functional and non functional requirements, the goals to reach, the real world (in order to model the constraints) and the real use cases which involve the actors. In the document are attached some diagrams that help to understand how all the part interact one with the others. Two of these diagrams are the UML and Use Cases. This document aims to show a clear description of the system that will be developed for the customer, and it can also help the future developers to understand the actual system in order to improve the functionality.

1.1.2 Overview and scope

The scope of the project is to develop a system which manage and optimize the hospitalization phase of patients. In particular, a patient can use a mobile application to have different kind of informations. *Medical Center* is divided in the following six sections:

- **Next Examination Section:** it provides a list of examinations divided into two further sections: *Today*, which provides all the examinations scheduled for the current day, and *Next Days*, which provides all the future examinations. If the user enables notifications, he will receive a reminder two hours before the beginning of the exam.
- **Prescriptions Section:** it provides all the drugs that the specific patient must take. Date and time are specified for each prescription. If the user

enables notifications, he will receive a reminder when he has to take the medicine.

- **Cafeteria Section:** it is a dynamic view which shows lunch and dinner menu, that are different for each day. Patients can select first and second dish and confirm their choice, sending informations directly to the cafeteria.
- **Places Section:** it provides the list of all the Points Of Interest of the hospital. Each of them is described by a list of informations like description, place, phone number, with the possibility to start a call or to start the navigation. The latter functionality lead the patient from a starting point to the destination, through textual and vocal commands, giving the best route to reach the final point and informations about obstacles presents on the path. User position is filtered using Kalman Filter in order to improve the precision of retrieved position.
- **Map Section:** it shows the map of the hospital and the position of the user inside it.
- **News Section:** it provides the latest news of the hospital, directly available in mobile version.

To use the system, the patient must be registered and logged. The registration phase is not managed by the final user, but username and password are provided by the reception of the structure.

1.1.3 Constraints

These are the constraints that the mobile application must satisfies:

- **Registration mandatory:** the registration is mandatory to use the service.
- **Unique username:** two different users cannot have the same username.
- **Hospitalized patient:** the application can be used only by hospitalized patients.
- **Parallelism:** the service must manage multiple requests from different customers at the same time.

- **External device for localization:** to be able to use localization and navigation functionalities, it is required an external device which sends informations about the user position to the mobile application. Nowadays it is a required condition, because mobile devices as smartphone or tablet don't include an UWB antenna, that is the technology used to estimate the position of an user.
- **Credentials at each access:** each time an user wants to use the service, he must insert username and password. If the device is equipped with touch ID, the patient can access to the app using it.

1.1.4 Non-Functional requirements

In the previous sections have been listed all the functional requirements of the application, that are the functionalities that the system provide to the final user. But the non-functional requirements are also important, because they permit to guarantee full functionalities for the system:

Design Constraints

For the development of the app, Swift language has been chosen. It means that, in the first release, it is compatible only with iOS devices (iPhone, iPad, iPod Touch). To guarantee that *Medical Center* works well on different devices, *auto-layout* is used, so the user interface can adapt itself automatically according to the device in use. This guarantees a good user experience. The architectural design pattern chosen is Model-View-Controller (MVC).

Performance

Informations about examinations, prescriptions and cafeteria must be retrieved from the hospital server. The system must answer at each request in the shortest possible time (a couple of seconds). So, to guarantee this requirement, data are exchanged using JSON files. JSON is a format to quickly transmit data objects through the web.

Security

All the informations provided by the application are confidential. The patient can access only to his own data, so he must login to the system at each access either through username and password or using Touch ID (if available on the device).

1.2 Architecture

In this section are listed the hardware components used in *Medical Center* and it is explained how they interact each other. Some diagrams shown static and dynamic behaviour of the components. At the end of this section is shown the architectural design pattern used, with a schema to better understand how it works.

1.2.1 Architectural structure

In *Medical Center* system have been identified different hardware components in order to obtain the goal:

- **Database Server:** this component stores all the data used by the system. Here are stored all the informations about each patient (examinations, prescriptions, selected menu), which could be accessible through an HTTPS call that returns a JSON file with all the requested informations.
- **Client (mobile application):** this component retrieves JSON files from the server, display them in the application view and sends data to the server with all the informations about the chosen menu of the patient. It also manages all the notifications that must be sent to the user device when there is a new examination or prescription.
- **Tag Device:** this component sends packets to the user about its position in the hospital through a bluetooth connection.
- **News API:** this component is an API that returns JSON metadata for the headlines currently published on a range of news sources and blogs. This API is used to retrieve the informations about the medical news to show in **News** section.

1.2.2 Architectural patterns

MVC is the chosen architectural pattern. It provides a separation between the user interface of the application (the view, which is shown on the client), the controller (the logic of the software, also present on the client) and all the data (model), that can be retrieved by different sources. In particular, all the informations of each patient can be retrieved from a database through an HTTPS call, the news are retrieved through an HTTPS to *newsAPI service*

while the position can be retrieved through bluetooth from an external device (Tag device), that is an object assigned to the patient when hospitalized.

1.3 Data model

In this section will be mentioned all the relationship between the various entity of the system. These will define the database, which data can be exchanged with the application through JSON files.

1.3.1 ER Diagram

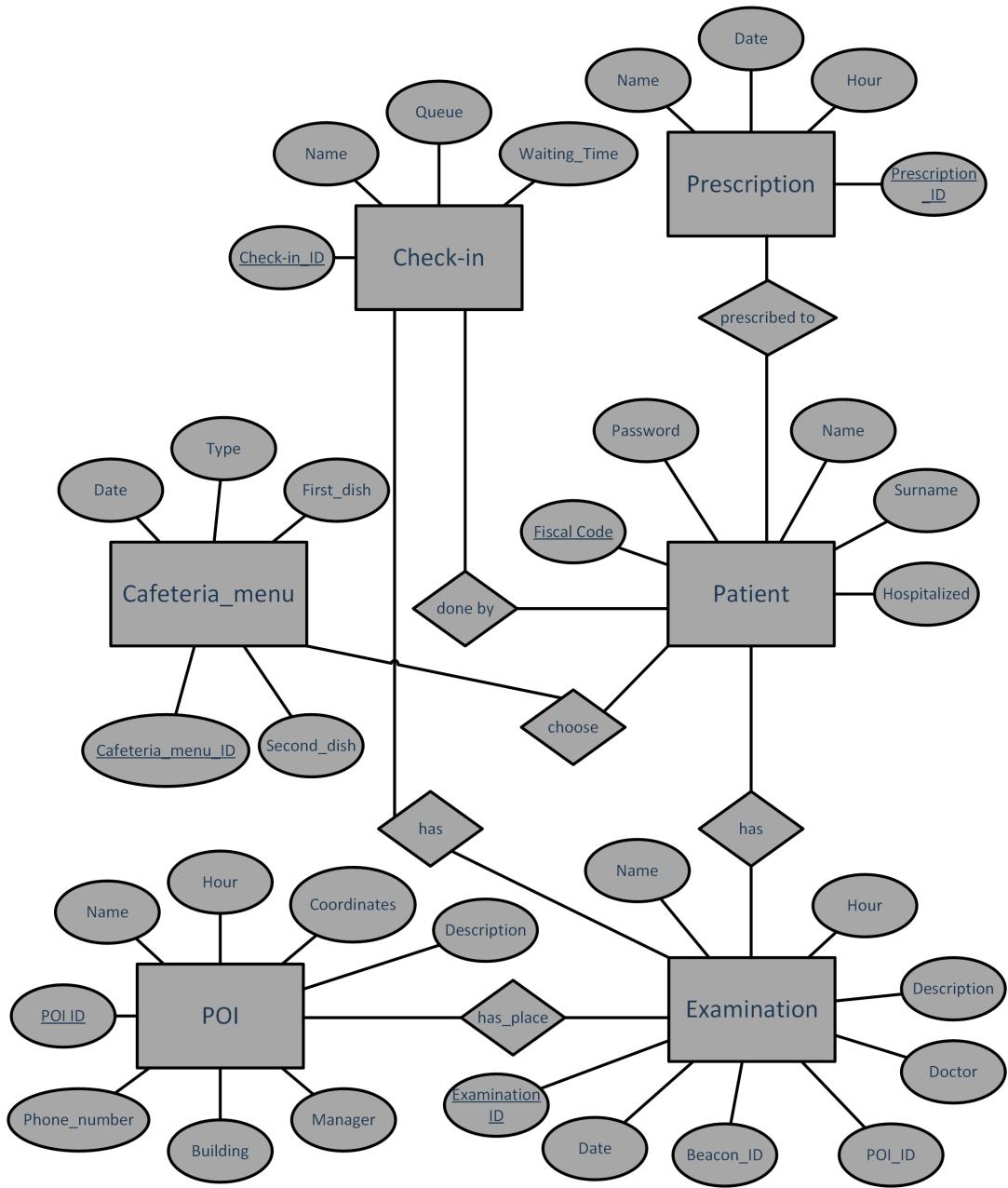


FIGURE 1.1: Entity-Relationship Diagram

1.3.2 Logical-Conceptual design

In the following sections are listed all the tables of the database. For each table, it is present key value which identify each instance of an entity and is useful to connect two different tables.

Patient table

TABLE 1.1

Field	Type	Description
<u>Fiscal Code</u>	String	It is the key of the table and represents the fiscal code of each user.
Password	String	It is the password to access to the application.
Name	String	It is the name of the patient.
Surname	String	It is the surname of the patient.
Hospitalized	Boolean	It is a boolean that is true if the patient is hospitalized, false otherwise.

Examination table

TABLE 1.2

Field	Type	Description
<u>Examination ID</u>	String	It is the key of the table.
Name	String	It is the name of the examination.
Date	Date	It is the date when the examination will be done.

Hour	String	It is the hour when the examination will be performed.
Description	Boolean	It is a description of the examination.
Doctor	String	It is the name of the doctor who will perform the examination.
POI ID	String	It is an identifier of the place where the examination will be taken.
Fiscal code	String	It is the identifier of the patient interested in the visit.
Beacon ID	String	It is the ID of the beacon used to perform the check-in.
Check-In ID	Boolean	It is the ID of the Check-in table.

Prescription table

TABLE 1.3

Field	Type	Description
Prescription ID	String	It is the key of the table.
Name	String	It is the name of the prescription.
Date	Date	It is the date when the prescription must be taken.
Hour	String	It is the hour when the prescription must be taken.

Fiscal code	String	It is the identifier of the patient interested in the prescription.
-------------	--------	---

Point of Interest table

TABLE 1.4

Field	Type	Description
<u>POI ID</u>	String	It is the key of the table.
Name	String	It is the name of the place.
Hour	String	It is the hour when the place is open.
Coordinates	String	It represents the coordinates of the place.
Description	String	It is the description of the place.
Manager	String	It is the name of the person who manages the place.
Building	String	It is the name of the building where the Point of Interest is.
Phone number	String	It is the phone number to contact the Point Of Interest.

Check-in table

TABLE 1.5

Field	Type	Description
-------	------	-------------

<u>Check-in ID</u>	String	It is the key of the table.
Ticket	Integer	It is the number of the ticket.
Queue	Integer	It is the number of people in queue before the user.
Waiting time	Integer	It is the waiting time for the user in minutes.

Cafeteria table

TABLE 1.6

Field	Type	Description
<u>Cafeteria menu ID</u>	String	It is the key of the table.
Date	Date	It is the date of the menu.
Type	String	It identifies if the menu is for lunch or dinner.
First Dish	String	It is a list of first dishes.
Second Dish	String	It is a list of second dishes.
Fiscal Code	String	It is the ID of the user who chose the menu.

1.4 External services and libraries

Some external services and libraries have been used to implement the functionalities of *Medical Center*: Some of them are needed to perform logical

operations, others instead are used to retrieve informations from external sources.

1.4.1 Library: SwiftyJSON

All the informations retrieved from the database or external services are formatted in a JSON format. So, there is the necessity to parse this kind of data structure. SwiftyJSON is a simplified JSON parsing library that gives a clearer syntax respect to the built-in iOS libraries. It doesn't handle HTTPS request to retrieve JSON (this operation is performed using internal Swift library) but it manages only the parsing of the files.

1.4.2 Library: Upsurge

Upsurge is a math utilities library. It provides support for linear operations on vectors and matrices, and slicing of higher-dimensional tensors. It relies on Accelerate, which is a framework that provides high-performance functions for matrix math, digital signal processing, and image manipulation. This library is useful to perform some mathematical operations to apply Kalman Filter on user position retrieved from tag device.

1.4.3 Service: News API

Medical Center provides news about the hospital. Actually, it is not possible to have real news, so an external system has been used to provide this kind of informations. Medical news are retrieved in JSON format from a news website and then are parsed and formatted on the view of the application. *News API* is a service which provides API and returns JSON metadata for the headlines and the links currently published on a range of news sources and blogs. In particular, *new-scientist.com* has been chosen as source.

1.5 UML diagrams

In the following sections are attached some diagrams that helps to understand how all the part interact one with the others. In particular, there are Class Diagram and Use Case.

1.5.1 Class diagram

To have a general idea of the implementation of the app, in Figure 1.2 is presented the class diagram of *Medical Center*. It represents the classes of the application, with all the relevant connections.

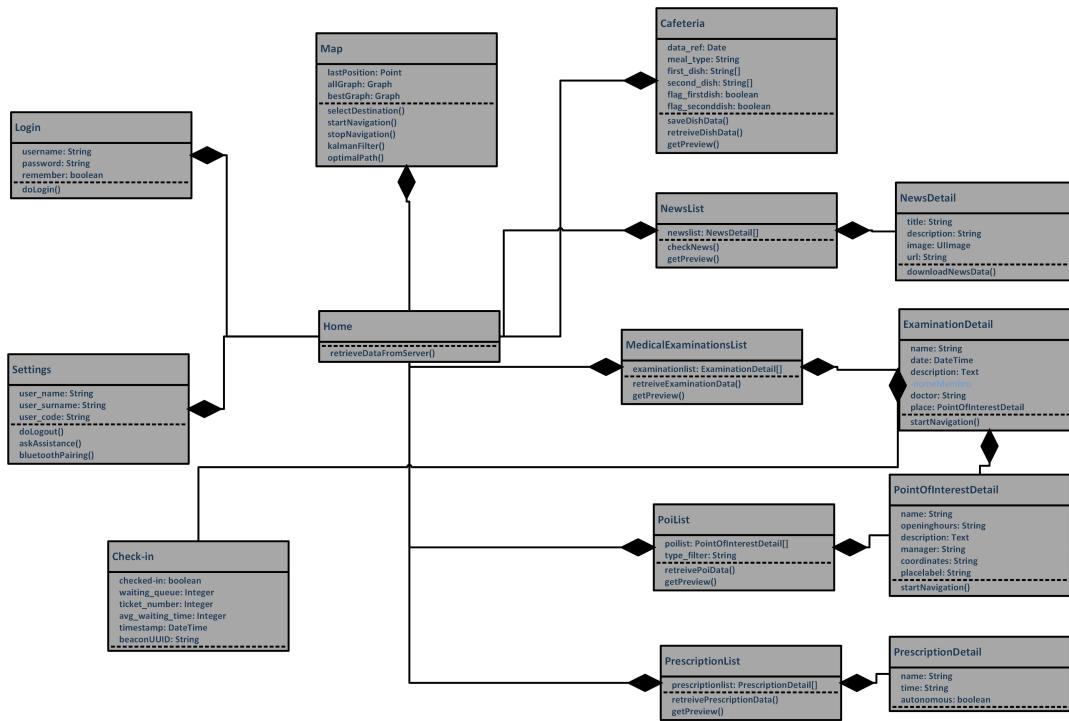


FIGURE 1.2: Medical Center Class Diagram

1.5.2 Use case diagrams

In this section will be shown the most relevant use cases of the application. They are an overview of the usage requirements for the system.

Login

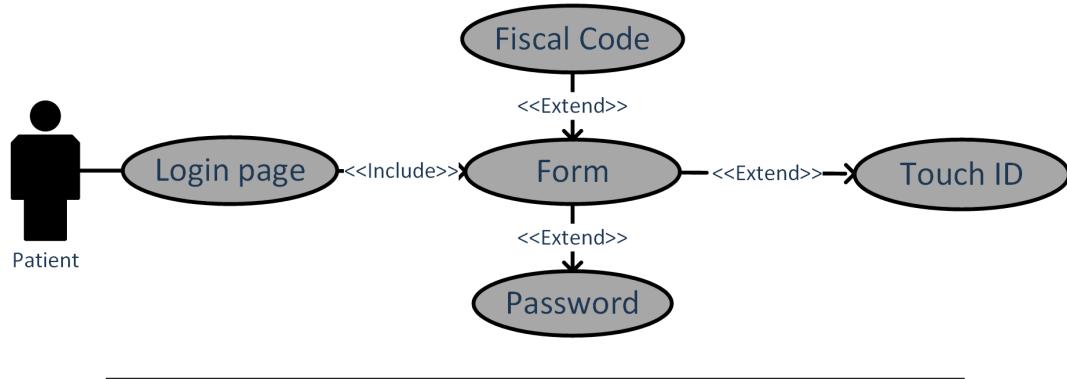


FIGURE 1.3: Login Use case

Settings

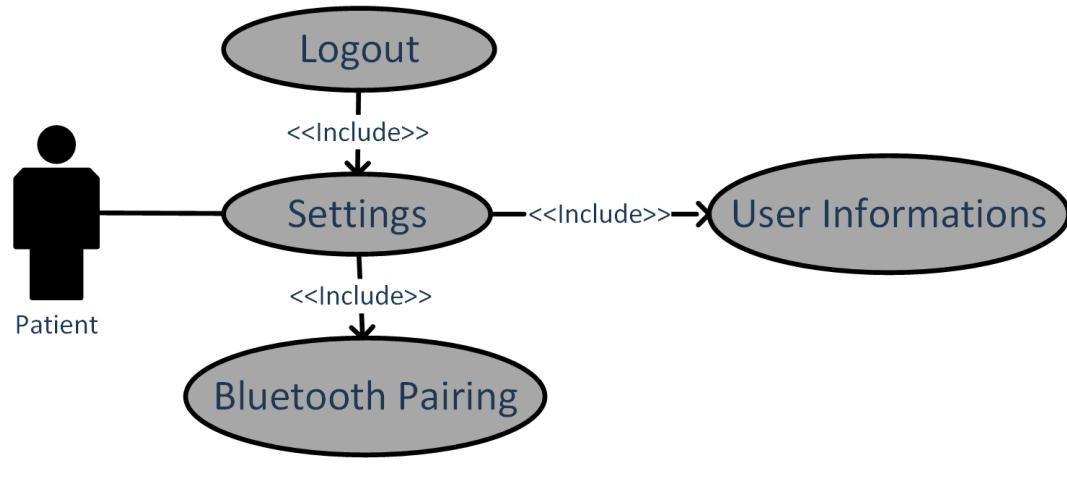


FIGURE 1.4: Settings Use case

Homepage

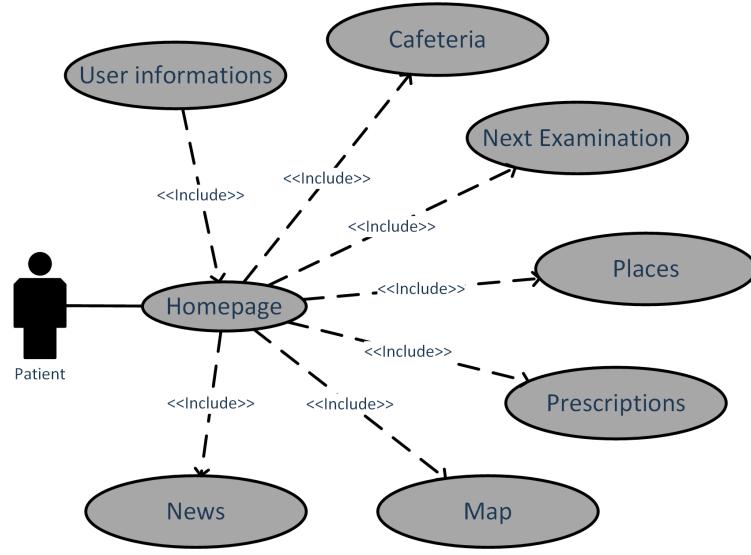


FIGURE 1.5: Homepage Use case

Cafeteria

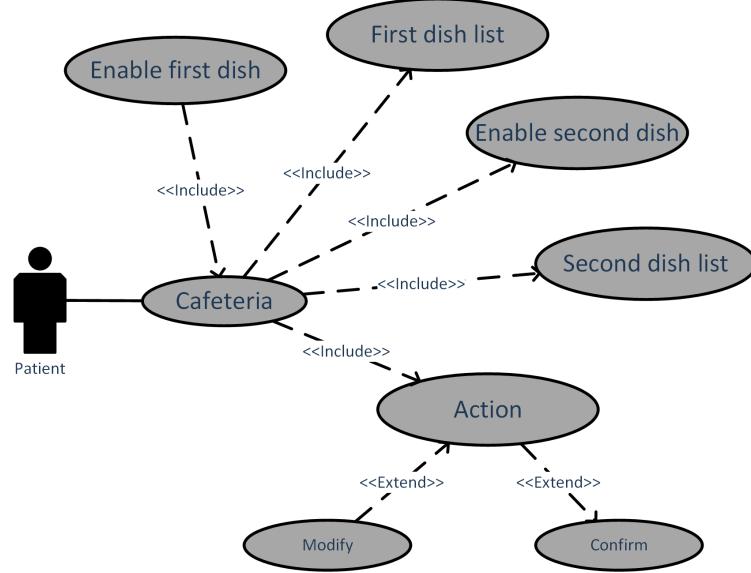


FIGURE 1.6: Cafeteria Use case

Medical Examination

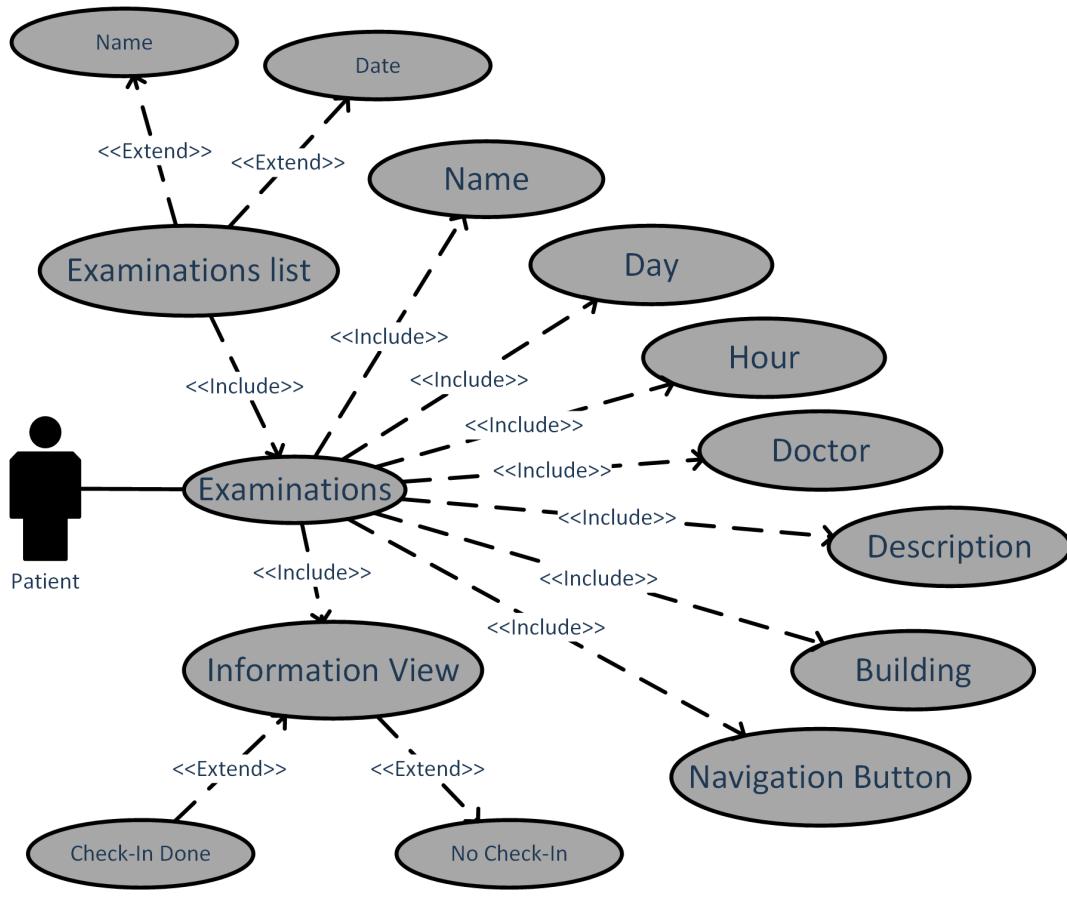


FIGURE 1.7: Medical Examination Use case

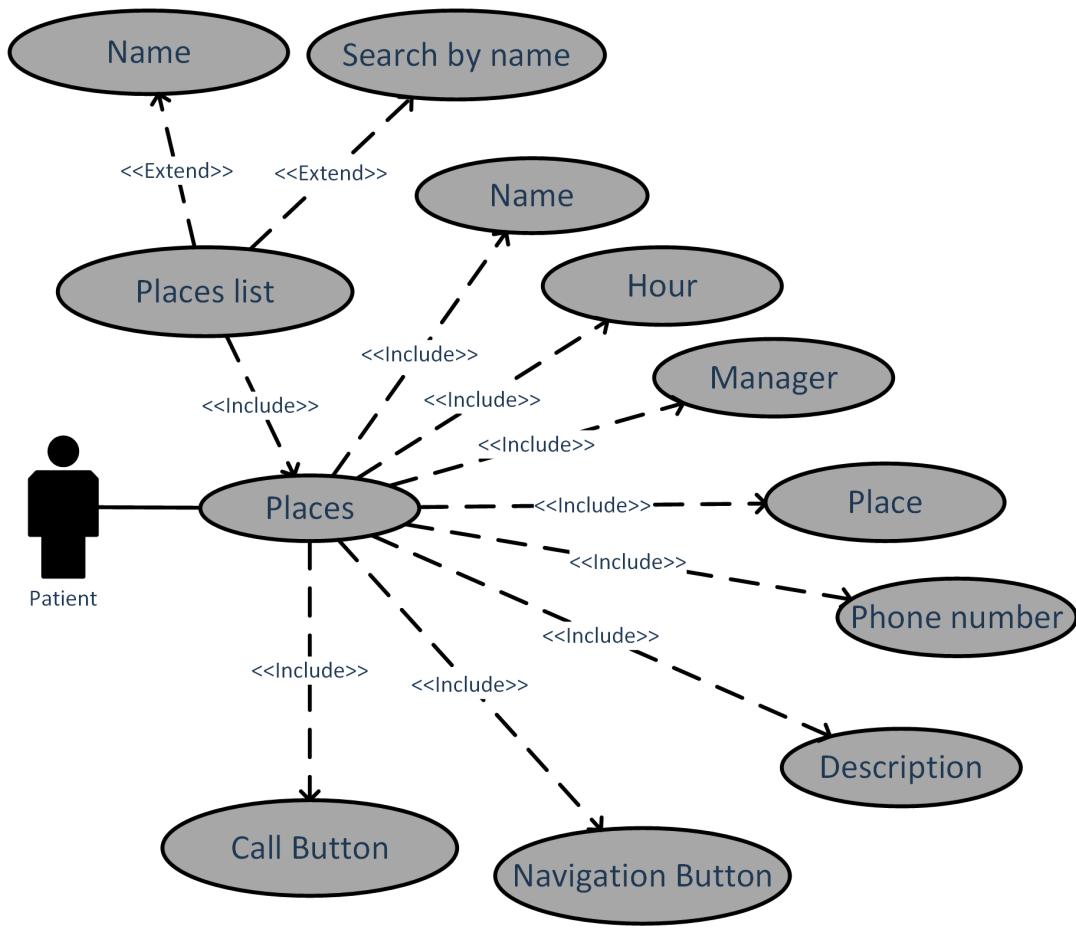
Places

FIGURE 1.8: Places Use case

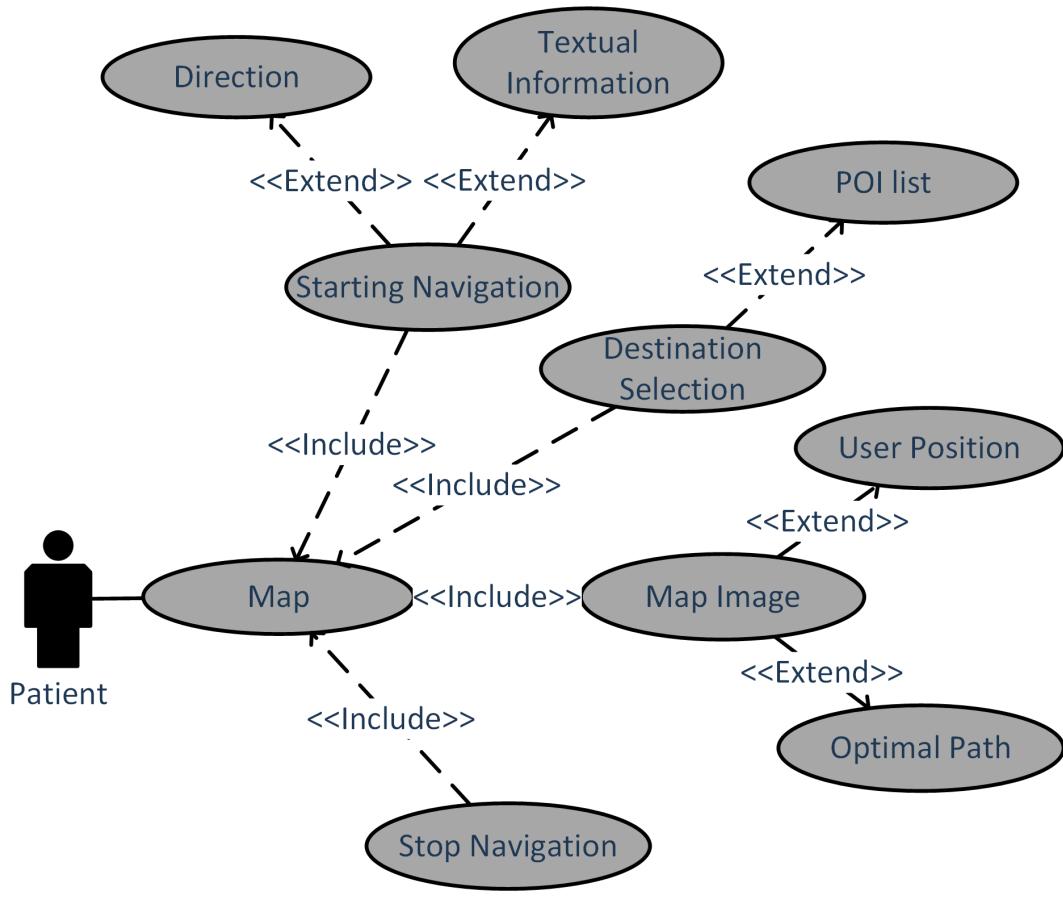
Map

FIGURE 1.9: Map Use case

Prescriptions

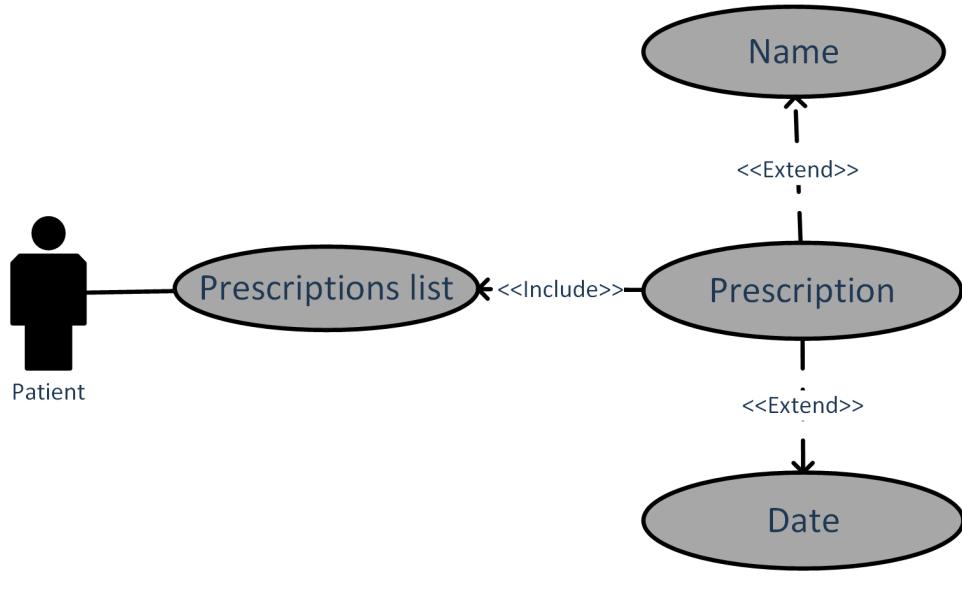


FIGURE 1.10: Prescriptions Use case

News

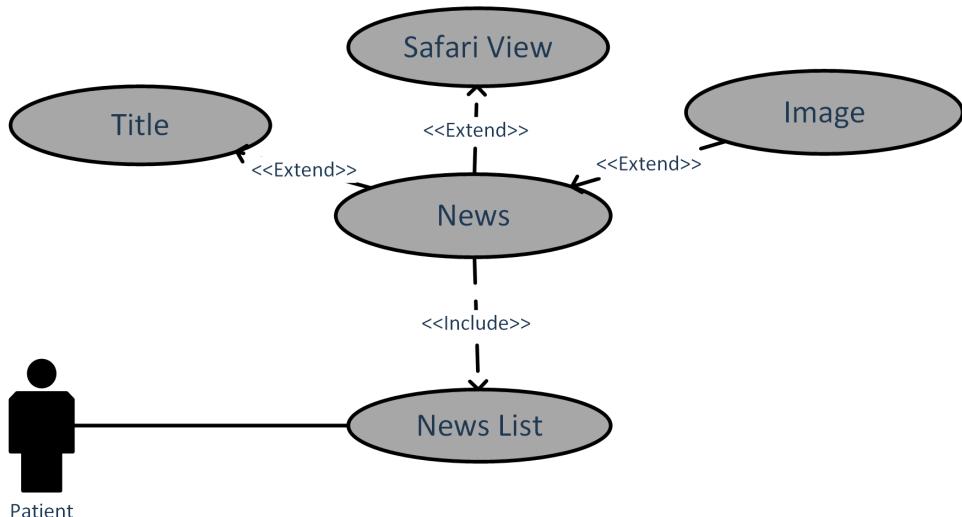


FIGURE 1.11: News Use case

1.6 User interface

Nowadays, the user interface is essential for a mobile application. There are many apps on the market, so what characterises an application respect to another is the user-friendliness. For this reason, a lot of effort has been made on the usability and the graphics of the application. No external graphics library have been used to develop the views. The intention was to avoid confusion for the end-user, using the same graphic elements as the operating system. So, iOS Human Interface Guidelines¹ have been used to design and implement an iOS application that works both on iPhone and iPad (Portrait and Landscape orientations are supported). In this section will be shown and explained the relevant views of *Medical Center* app. To navigate between each view it has been implemented a tab bar controller, which shows on the bottom all the views implemented in the application.

¹<https://developer.apple.com/ios/human-interface-guidelines/overview/design-principles/>

1.6.1 Login View

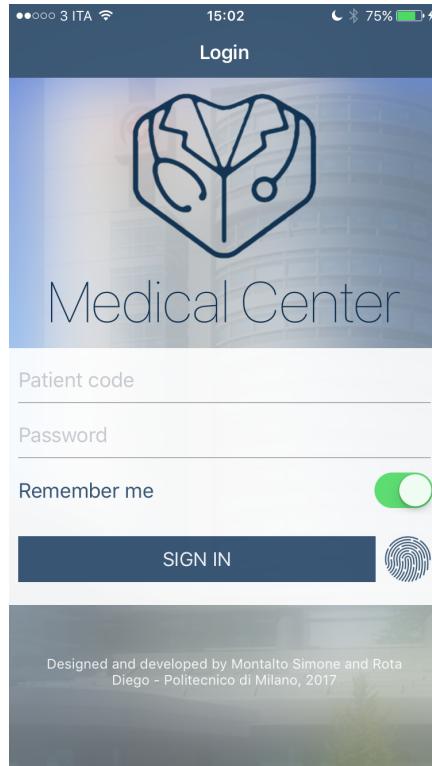


FIGURE 1.12: Login view of Medical Center

This is the first view that is shown when a user tap on the app icon. It permits to do the login with patient credentials. There isn't a *register* button because username and password are provided by the reception when a patient is hospitalized. Two labels are needed to insert the patient code and the password (password field is obscured in order to avoid that external people could see the password during the typing). A switch can be enabled to memorize the password. If the iOS device has Touch ID sensor, enabling Remember me field it is possible to access to the app using the fingerprint. Finally, a button associated to the Touch ID is present to show up the view for login with the fingerprint.

1.6.2 Home View

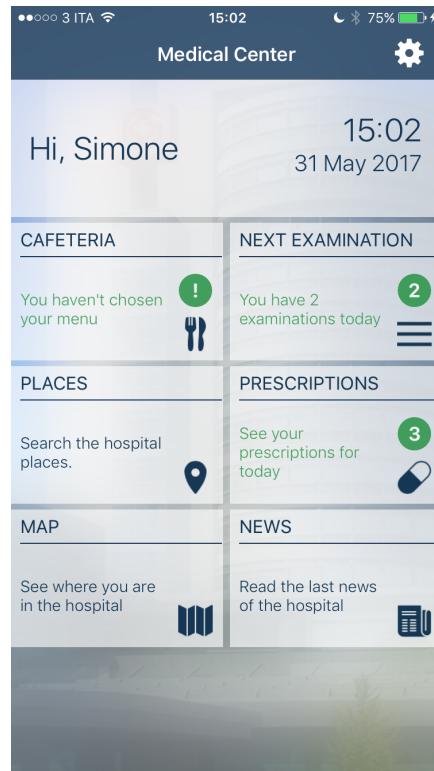


FIGURE 1.13: Home view of Medical Center

The homepage is a dynamic view. It is composed by seven tiles that show different information. The first tile is dedicated to the user's name, with date and hour which update itself automatically. Then, a tile has been dedicated for each section of the app, which is tapped to open the relative view. Below the title of each tile there's a brief description of its content or a quick view of relevant information for the user. For example, the *Next Examination* tile inform the patient if there are new examination for the day or not. A coloured badge is shown if a tile must attract the attention of the user.

1.6.3 Cafeteria View

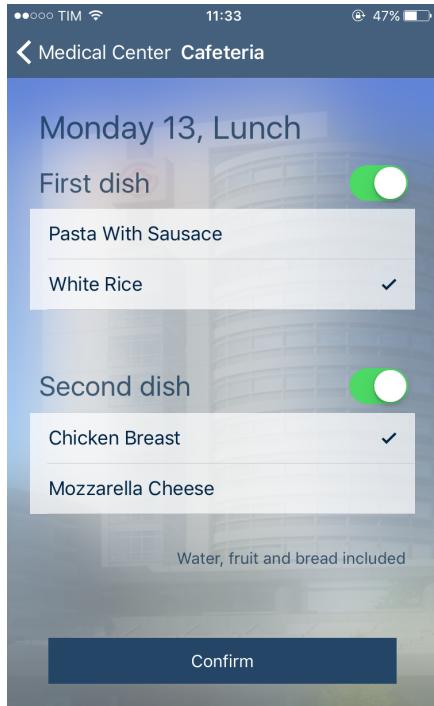


FIGURE 1.14: Cafeteria view of Medical Center

Cafeteria View shows the menu of the day. Each patient can select a personalized meal, with the possibility to deselect first or second dish if he want only one course. At the end of the selection, he can tap on *Confirm* button to confirm its selection. Until a new menu isn't inserted, the user can modify his choice tapping on *Modify* button.

1.6.4 Next Examination View

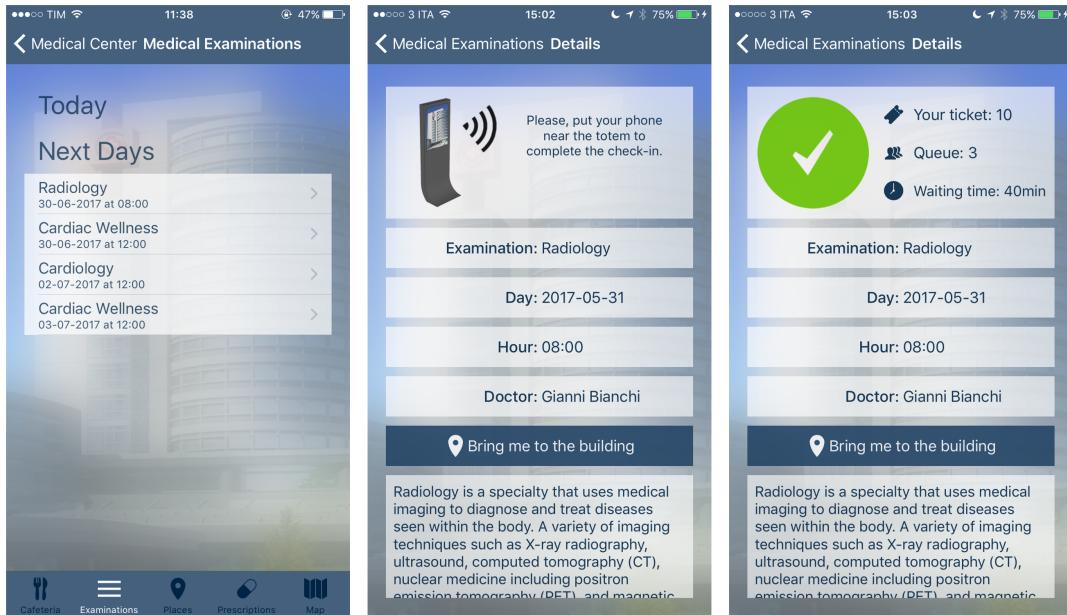


FIGURE 1.15: Next examination views of Medical Center

Medical examinations are presented in two views: the first one is a list of examinations, divided into *Today* and *Next Days*. Tapping on each medical examination, a further view is opened, with its description and some details. This view implements beacons technology in order to complete the check-in when a patient goes to the place of the visit. User can put the phone near a totem (equipped with beacons) the day of the examination to confirm his presence and do the check-in. Then, some information are provided, such as the number of his ticket, the number of people in queue and the waiting time. Also a button can be pressed to start the navigation inside the hospital to reach the place of the examination.

1.6.5 Places View

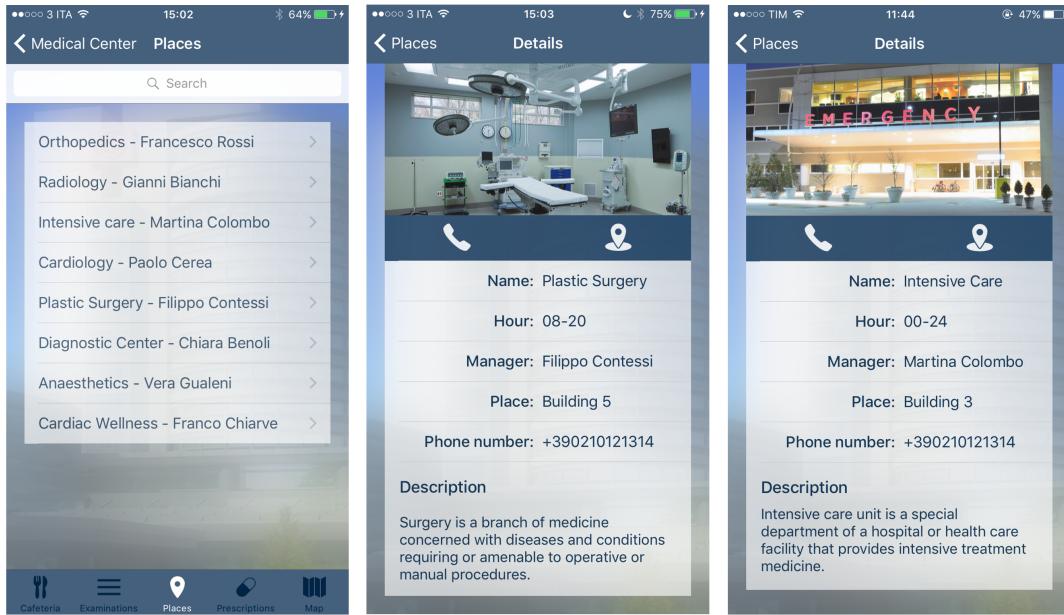


FIGURE 1.16: Places views of Medical Center

Also places are divided into two views, the first one being a list of all the Point Of Interest of the Hospital. This view also features a search bar to perform a quick research between all the places of the hospital while the details view ipresents some information of the place with quick buttons to call the reception or start a navigation to the Point Of Interest.

1.6.6 Prescriptions View

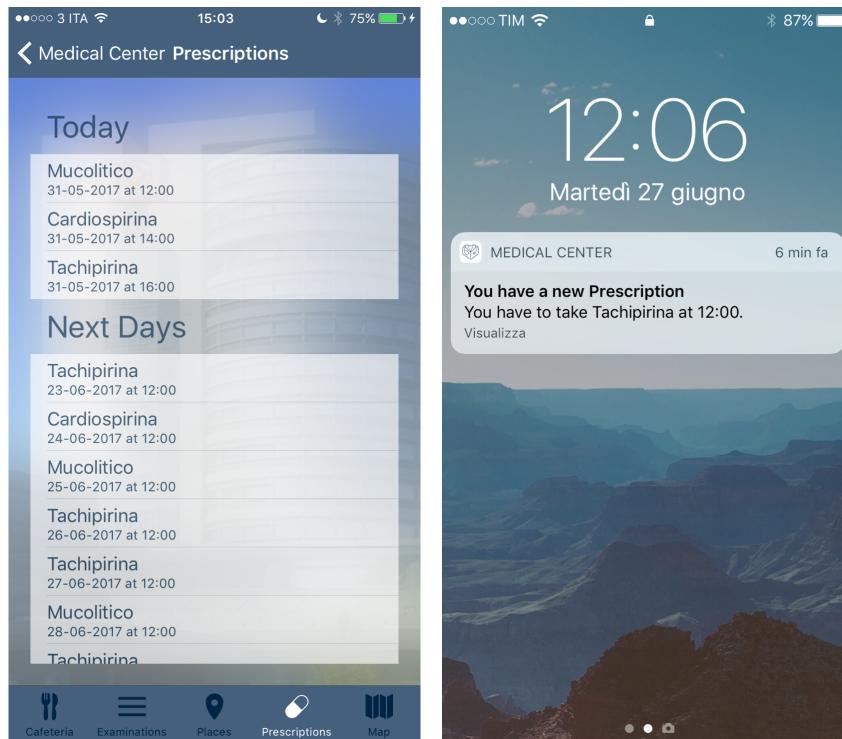


FIGURE 1.17: Prescriptions views of Medical Center

Prescriptions view show all medicines that the patient has to take. They are divided into *Today* and *Next Days*, presenting the name of the prescription and day and hour when the user has to take it.

1.6.7 Map View



FIGURE 1.18: Map view of Medical Center

This view presents the map of the hospital. When the smartphone is connected to a Pozyx Tag, his position and orientation are also shown on the map. At the bottom of the view, users can choose a destination point between all the Point Of Interest of the Hospital. Then, tapping on start button, they can start navigation to the selected place. After pressing the button, the optimal path is calculated and shown on the map. The bottom part of the view changes, showing textual indications place side by side by an arrow that shows the direction to follow. This view is dynamic, because since it updates itself reflecting new directions or possible obstacles on the path. The view also change colour if during the route an obstacle is nearby or if the user leaves the optimal path. Vocal commands to drive the user during the navigation have also been implemented. On the top of the view, some information about the connection state of Pozyx tag are shown in a box, that is hidden when there aren't new communications.

1.6.8 News View

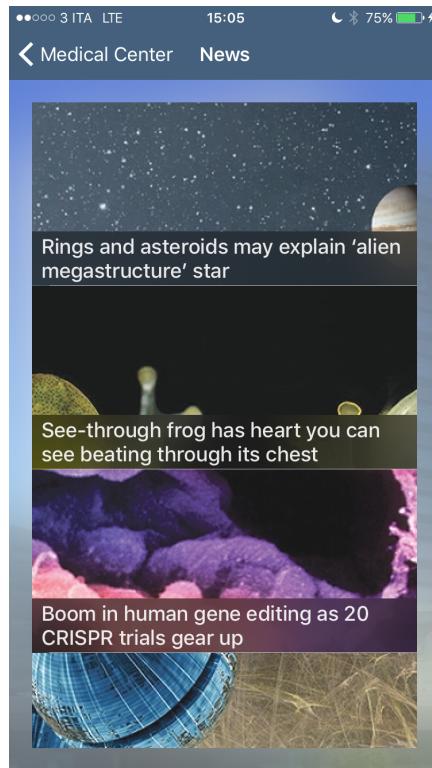


FIGURE 1.19: News view of Medical Center

News view retrieves new information through an HTTPS request and presents them in a collection view with the image of the news and its title. Tapping on each news, a Safari View Controller is created loading the URL of the chosen item.

1.6.9 Settings View

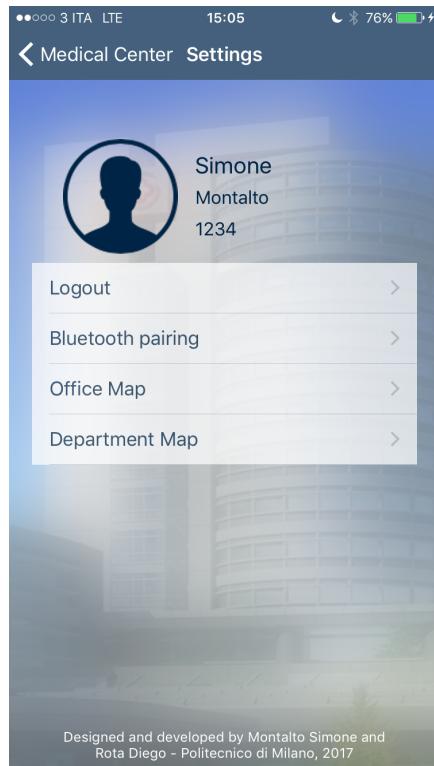


FIGURE 1.20: Settings view of Medical Center

Through the settings view, the user can show his information (name, surname and patient code), logout from his account and perform the bluetooth pairing with the Pozyx Tag.