

**Report PiDev 2018**

WRITTEN BY:

* *BEN HAJ YOUSSEF Haythem*
* *MEGDICHE Hamdi*
* *MNASRI Haythem*
* *ZOUAOUI Ilyes*
* *ZOUGHLAMI Imen*

CLASS: *4TWIN2*

TOPIC: *Epione*

**Table of contents**

[General introduction 1](#_Toc526805133)

[CHAPTER 1 : Main conception & Used methodology (Sprint 0) 3](#_Toc526805134)

[Introduction 4](#_Toc526805135)

[1. Study of existing solutions 4](#_Toc526805136)

[1.1. Issue 4](#_Toc526805137)

[1.2. Target audience 4](#_Toc526805138)

[1.3. Similar solutions 5](#_Toc526805139)

[2. Analysis 7](#_Toc526805140)

[2.1. Functional requirements 7](#_Toc526805141)

[2.2. Actors 8](#_Toc526805142)

[2.3. General Use Case Diagram 9](#_Toc526805143)

[2.4. Analysis Class Diagram 10](#_Toc526805144)

[2.5. System sequence diagram 11](#_Toc526805145)

[2.6. Non-functional requirements 12](#_Toc526805146)

[Conclusion 13](#_Toc526805147)

[3. Physical architecture 14](#_Toc526805148)

[3.1. Global architecture 14](#_Toc526805149)

[3.2. Environment 14](#_Toc526805150)

[3.2.1. Technologies 14](#_Toc526805151)

[3.2.2. Web service 15](#_Toc526805152)

[3.3. Deployment Diagram 15](#_Toc526805153)

[Conclusion 16](#_Toc526805154)

[4. Project Management 17](#_Toc526805155)

[Introduction 17](#_Toc526805156)

[4.1. Methodology 17](#_Toc526805157)

[4.1.1. Scrum 17](#_Toc526805158)

[4.1.2. Product Backlog 18](#_Toc526805159)

[4.1.3. Conclusion 18](#_Toc526805160)

[4.2. Version control repository Manager (GIT) 19](#_Toc526805161)

[Conclusion 19](#_Toc526805162)

[5. Mockups 20](#_Toc526805163)

[General conclusion 25](#_Toc526805164)

**List of Figures**

[Figure 1 : Web site Tuotempo 5](#_Toc526804941)

[Figure 2 : Web site DocteurDirect 6](#_Toc526804942)

[Figure 3 : Web site Bonjour-sante 6](#_Toc526804943)

[Figure 4 : General Use case diagram 9](#_Toc526804944)

[Figure 5 : Analysis class diagram 10](#_Toc526804945)

[Figure 6 : Make appointment system sequence diagram 11](#_Toc526804946)

[Figure 7 : Add schedule sequence diagram 12](#_Toc526804947)

[Figure 8 : Global architecture 14](#_Toc526804948)

[Figure 9 : Deployment diagram 15](#_Toc526804949)

[Figure 10 : Home page 20](#_Toc526804950)

[Figure 11 : Search Result Page 21](#_Toc526804951)

[Figure 12 : Sign IN / Sign UP interface 22](#_Toc526804952)

[Figure 13 : Appointment booking page 23](#_Toc526804953)

[Figure 14 : Profile page 24](#_Toc526804954)

**List of tables**

[Table 1 : Sprints technologies 18](#_Toc526803781)

# General introduction

In the context of our integration and development project of the 4th year web and internet technology (TWIN), we were assigned the realization of a project in which we directly apply the skills acquired throughout this semester. We chose Health caring project, which is a platform that offers making appointments between doctors and patients.

Many people do not realize, and often underestimate the importance of good health.  
Health, as they say, is wealth and good health is necessary to carry out daily tasks.

The everyday choices you make can have a big effect on your health, your life and your wellbeing. Eating well, keeping active, not smoking or drinking too much alcohol and looking after your mental health can all help you stay well and enjoy life.

When discussing about health, we say doctors also, they save lives, but their importance goes far beyond that. Doctors also make a difference by helping patients minimize pain, recover from a disease faster or learn to live with a disabling injury. A patient's ability to enjoy life, even if they cannot be healed, makes a huge difference to them and to their families. If they can go back to work after an illness that benefits their employer too, and that is only part of what makes doctors important to society. Our platform will provide many services to facilitate the doctor and patient daily live.

“Health caring” is a project undertaken as a mandatory requirement for the course “PI-dev 4TWIN” that is being conducted mutually by MNASRI Haythem, MEGDICHE Hamdi, ZOUAOUI Ilyes, ZOUGHLAMI Imen and BEN HAJ YOUSSEF Haythem all of whom are members of 4TWIN2 class of 2018/2019 and the EPIONE team.   
The aim of the course is to provide a distributed environment to develop a Web application using multiple different technologies and synchronize the communication between them using APIs.

The project was carried out in four steps:

* The definition of the project, which consists of editing the functional specifications: The design and implementation of the project.
* The Implementation Business party and the exposition of web services using JEE technology
* The realization of a part of the project using .NET technology and the movable part using a cross platform Ionic
* The implementation Customer part respecting standards of Angular JS.

In the next chapters, we will mention details about the development of each of these parts being as specific as possible for the reader of this report.

CHAPTER 1 : Main conception   
&   
Used methodology (Sprint 0)

Introduction

In this report we have to explain the initial phase of the project process, we will call it sprint 0’s review. We will focus, as you will notice below on client’s needs, whishes, main features, vision, obviously technologies and environment.

The tutors proposed the theme and we are tasked to expand the main feature of the application, which can be resumed in facilitating the process of making an appointment.  
In this document, there is overall description of the Web Project. It includes what we aim to do. There is also description of the analyzing phase, whole descriptions of the design intended, scenarios and technologies are included in this document.

1. Study of existing solutions
   1. Issue

In everyday life, each individual is faced with a busy schedule. He has to wait for hours to make an appointment, there are days when we are afraid of having to make an appointment with a certain doctors, because he fears the waiting time and in the end, he does not find an appointment available or appropriate with their availability.

To overcome this lack of flexibility and speed, we thought to create this application, which consists of making an appointment online in just a few clicks in order to ensure better management of everyday life and change the spirit of our community to respect and understand the principles of the appointment. Together, we can make a difference.

* 1. Target audience

The main actors as we will see in detail later would be the patients seeking appointments. To describe the market better, we tried to understand the client’s needs, thereafter we were able to identify to which community the client will belong and it’s simply anyone with a Computer or Mobile Phone and obviously an Internet connection.

* 1. Similar solutions

This step is designed to start any IT project, define the operating context, or the business process, and identify the different imperfections in existing applications to correct and improve them in our new application. To detect existing problems, we have identified the following deficiencies in these applications:

* The confirmation of the appointment is not immediate and it requires the intervention doctor to confirm the date chosen by the client.
* The application may not be understandable to the user.
* Lack of options that provide comfort to clients to encourage them to choose to take online appointments.
* There is a lack of tracking appointments and sending reminder notifications.
* Non-computerized data (Customer records).



Figure 1 : Web site Tuotempo



Figure 2 : Web site DocteurDirect



Figure 3 : Web site Bonjour-sante

1. Analysis

During this phase, we will explain the main features, different actors and aimed interactions between users of the application according to the product owner and the scrum master.

* 1. Functional requirements
* **Manage Client Appointement**

The patient selects the specialty of the doctor and the city (for the place he can choose geolocation), he consults the doctor's availability and make an appointment with him, he can optionally write a message to that doctor. The confirmation of the appointment is done by sending an e-mail to the patient and by notification also the patient can cancel or update his appointments with doctors.

* **Manage Doctor Appointment**

The doctor can consult his daily program, specifies his availability by a 15 min slot, also he can cancel an appointment with a patient, and take a vacation.

* **Rate**

The patient can rate his visit to the doctor from 1 to 5.

* **Manage Patient Profile**

The patient can consult and add and update his personal information, also he can consult his paths.

* **Claims**

The patient can send claims about a doctor.

* **Search Doctors**

Visitors, patients and doctors, can search for doctors in our application.

* **Contact Support**

Visitors, patients and doctors, can contact support.

* **Manage Patient Path**

The doctor can create a patient path, and he can recommend other specialists for that patient.

* **Manage Doctor Profile**

The doctor can consult and add and update his personal information, also he can consult his dashboard and analytics.

* **Manage Accounts**

The administrator can consult, activate or deactivate patients and doctors accounts.

* **Manage Claims**

The administrator can consult the claims and send warnings to the doctors.

* 1. Actors

Indeed our application includes the following actors:

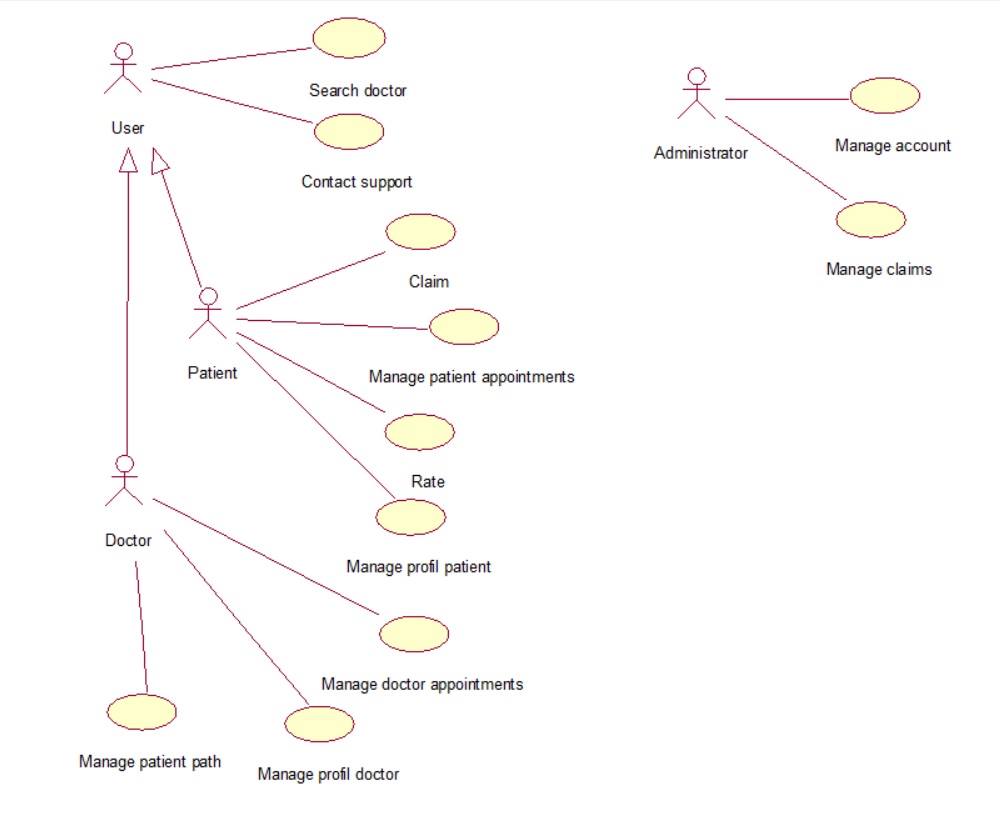
* **The simple user**  
  Who can only access services that do not require an authentication.
* **The administrator** It is the actor who manages the users of the system (activation / deactivation of accounts) and the different specialties and modules found in the application, visualize the statistics and has the right of access to the application.
* **Doctor**It's the user who has his own schedule to put his slots available to receive appointments and notification from the system, he is also the responsible for the patient's medical history.
* **Patient**   
  It is the user who makes appointments with professionals from freeway easy and immediate.
  1. General Use Case Diagram

Figure 4 : General Use case diagram

* 1. Analysis Class Diagram

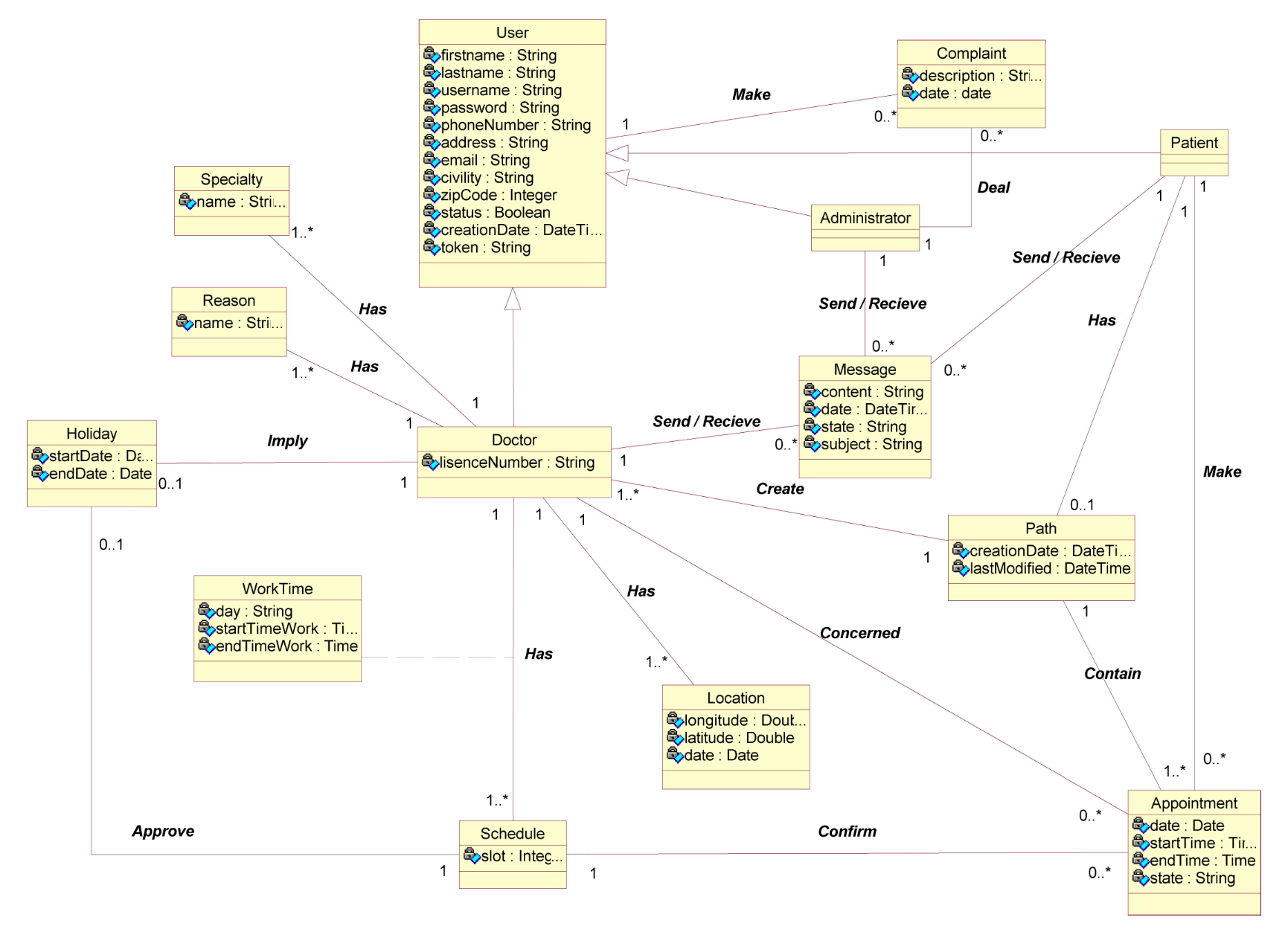


Figure 5 : Analysis class diagram

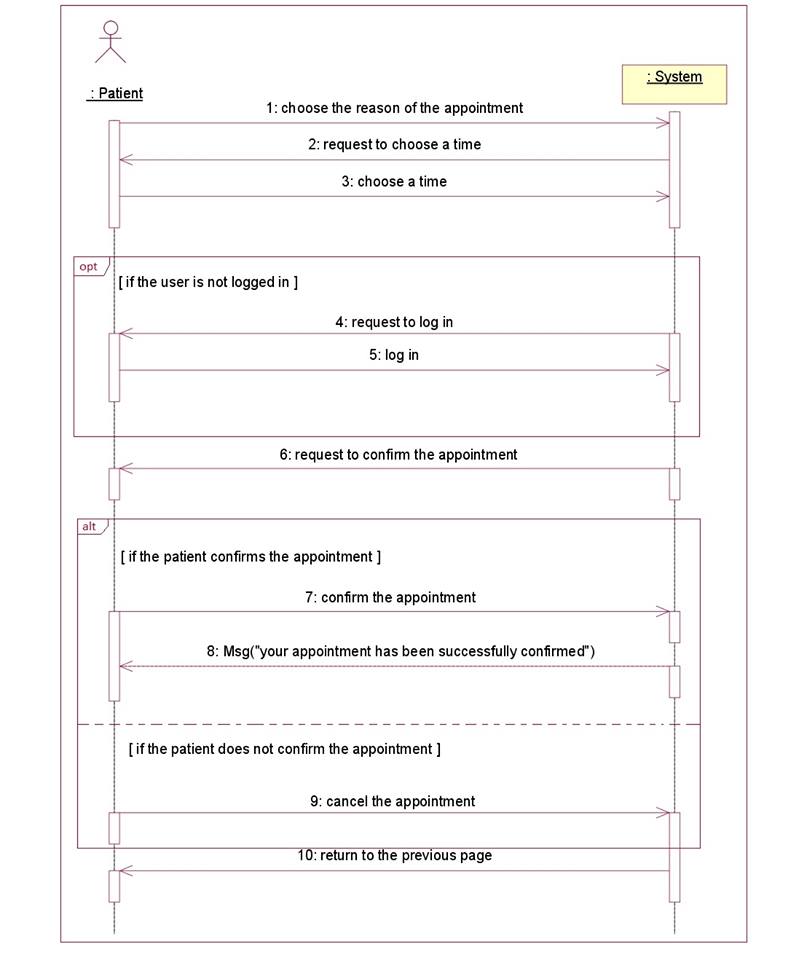
* 1. System sequence diagram

Figure 6 : Make appointment system sequence diagram

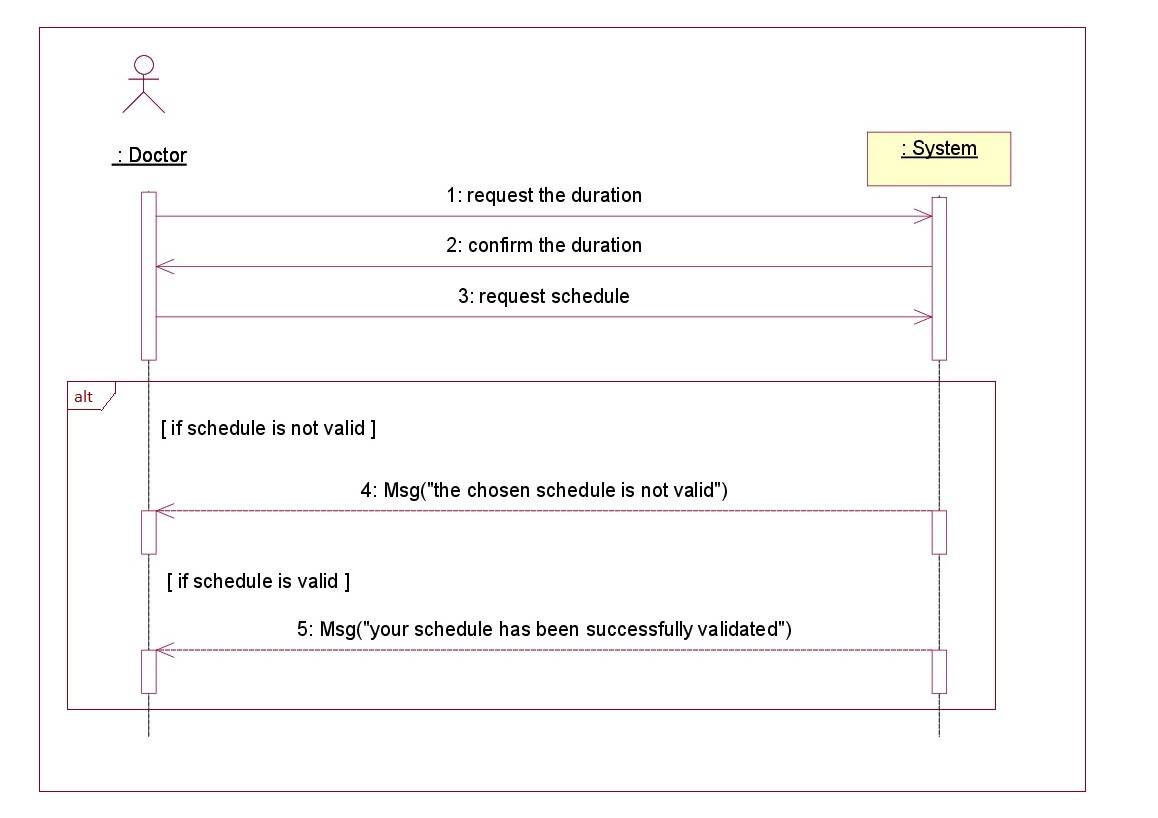


Figure 7 : Add schedule sequence diagram

* 1. Non-functional requirements

These are requirements that do not relate specifically to the behavior of the system but instead identify internal and external constraints of the system.

* **Simplicity**

The graphical interface of our application must be simple clear and easy to handle, to ensure easy and intuitive navigation and since users are not computer scientists.

* **Effectiveness**

The execution of the application must be done without errors and without crashes.

* **Performance**Attention should be paid to loading time, processing time and refreshment.
* **Availability**

The application must be available regardless of the information flows entered.

* **Reliability**

The application must work in a consistent manner without errors.

* **The validity**

It is the fact of responding to the functions expressed in the specifications.

* **Security**

The system in question must provide an authentication and access control function by specifying the access rights for each user to ensure the security and integrity of the data.

The application must respect the confidentiality of users’ data.

Conclusion

Throughout this chapter, we projected the purpose of the system to be developed, our application manages to save time by gathering all the existing solutions and providing several additional features all related to healthcare that satisfy most of the important needs. In the upcoming chapter, we start the development process, beginning with Java Enterprise Edition and the web service.

1. Physical architecture

Before the start of the project, we were informed about the technologies we will have to work with or use and according to these technologies a certain Agile Method will be used to better manage the work process.

* 1. Global architecture

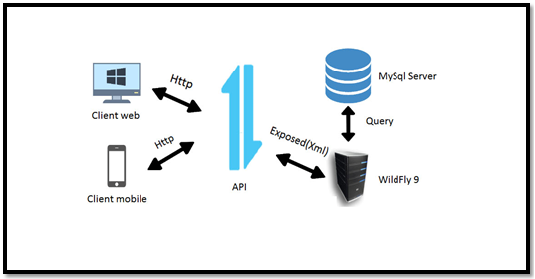


Figure 8 : Global architecture

* 1. Environment
     1. Technologies
  + **Java Enterprise Edition**

Java EE will be the back-end system of our application. Using Java EE to build all necessary functionalities regarding this sprint, then exposing them as RESTful API that shall be consumed in an upcoming sprints.

* + **.NET**

The .Net side of the application will be the Front-end product for the back-office of the main application as it will consume the exposed Metadata from the Wildfly server.

* + **Angular/Ionic**

The Ionic side of the application will be the Front-end product for the front office for mobile users and the Angular side of the application will be available for web users and both will consume the exposed Metadata from the Wildfly server.

* + 1. Web service

A client application can access remote distributed resources. There are several ways to access these resources, and web services are the most portable. We will use REST services (Representational State Transfer) with a Java API, which has an extraordinary evolution in the last versions of the Java Enterprise platform.

* 1. Deployment Diagram

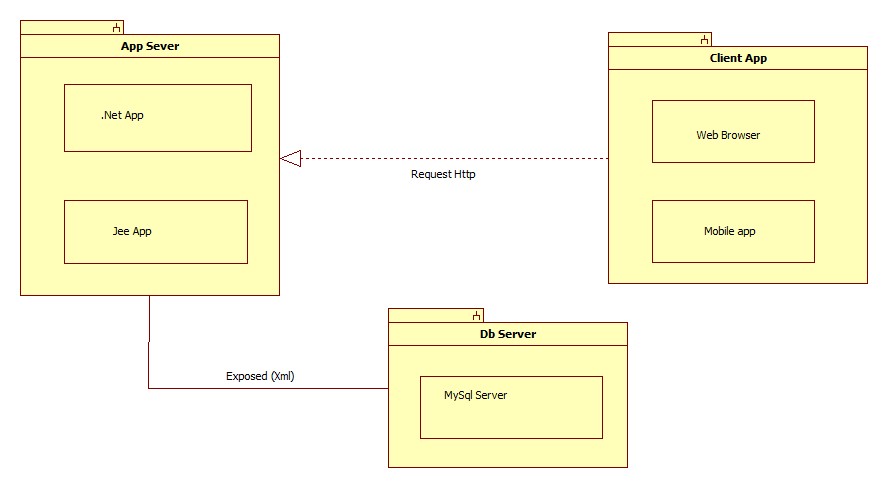


Figure 9 : Deployment diagram

Conclusion

This diagram resumes the global architecture of our application. During the sprint 1, we will develop the Java EE resources, deploying in the WildFly server that shall be exposed via a REST API. In the sprint after, we will tackle the ASP.NET application, deployed on IIS express and provides the back-office system. In the final sprint, we will consume the previously exposed REST resources with Angular as a Front-end application along with an Ionic mobile application.

1. Project Management

Introduction

Since our project falls in the software criteria, we chose to follow a method of work that has proven to be highly successful in similar domains which is the scrum methodology. Scrum is an agile project management concept used for software development that revolves around the use of multiple small teams working in an intensive, interdependent manner to produce a functional product.

* 1. Methodology

Scrum prescribes four formal events for inspection and adaptation:

* Sprint Planning.
* Daily Scrum.
* Sprint Review.
* Sprint Retrospective.

The scrum team consists of:

* Product owner.
* Scrum master.
* The Development Team.

Throughout this project, we will exchange roles during each sprint to guarantee our aptitude for managing a team and committing to an agile methodology.

* + 1. Scrum

The sprints planning was pre-assigned by the tutors of the course depending on the use of certain technologies as we will see below.

|  |  |  |
| --- | --- | --- |
| SPRINTS | TECHNOLOGIES | NATURE |
| SPRINT 0 | DATA Gathering & Analysis | |
| SPRINT 1 | Java EE | Back-end |
| SPRINT 2 | .NET | Front-end |
| SPRINT 3 | Angular/Ionic | Front-end |

Table 1 : Sprints technologies

* + 1. Product Backlog

The agile product backlog in Scrum is a prioritized features list, containing short descriptions of all functionality desired in the product. Typically, a Scrum team and its product owner begin by writing down everything they can think of for agile backlog prioritization. This agile product backlog is usually more than enough for a first sprint. The Scrum product backlog is then allowed to grow and change as more is learned about the product and its customers. Refer to the annex below to find our up-to-date Product Backlog.

* + 1. Conclusion

When practicing Scrum, we can make the sprint backlogs visible by putting it on a Scrum task board. We will update the task board continuously throughout every sprint. To make it more practical for us, using tools such as Trello.

* 1. Version control repository Manager (GIT)

The aim of GIT is to manage web development projects and files. As they change over time. Such a git repository contains a set of commit objects and a set of references to commit objects. A git repository is a central place where developers store, share, test and collaborate on web projects. We will be using GitLab over GitHub for the simple reason, which is Issue Tracking.

If you are using GitHub issues, you might think that it lacks some functionality. GitLab provides a powerful Issue Tracker that lets you change status and assignee for multiple issues at the same time. The repository is owned by Git Master “MEGDICHE Hamdi” as the rest of the group are assigned Developers.

Conclusion

The global view of the architecture shown above will help the team during the phases of designing and implementing methods and functions giving them a better understanding of the logical architecture of each of the applications especially when having to deal with RestAPI.

1. Mockups

The Mockups below resume the process to make an appointment. We opt for “Balsamiq mockup“ software for interface design.

This is the home page of our application, the visitor can easily search for a doctor by his name, specialty or address.

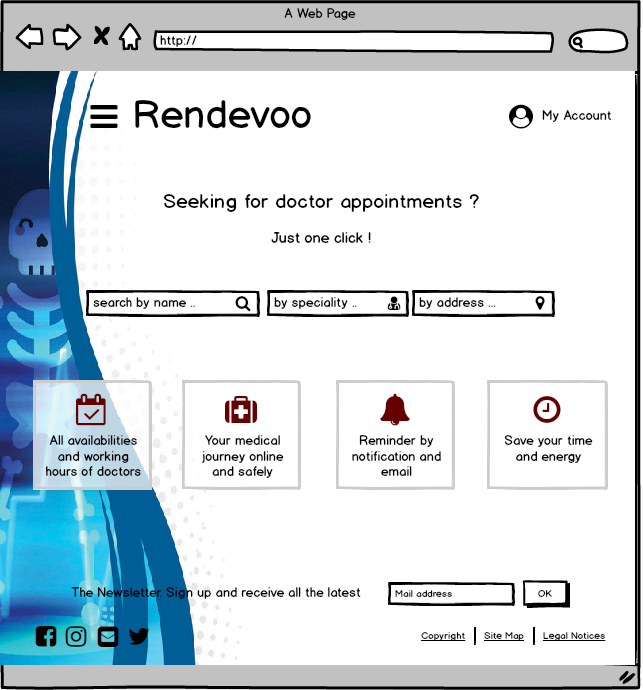


Figure 10 : Home page

This page displays the result of the search in a map and a list and allows the customer to choose a doctor to make an appointment.

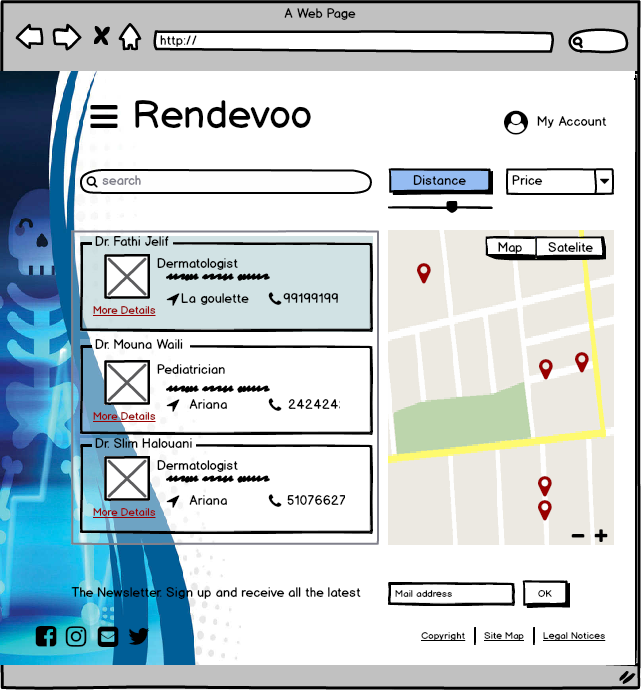


Figure 11 : Search Result Page

In this case, the visitor must register or sign in as a patient to be able to make one.

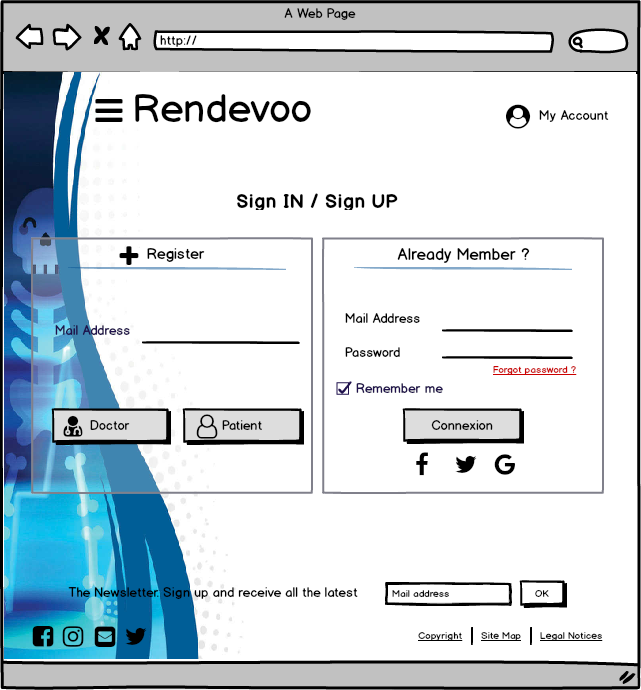


Figure 12 : Sign IN / Sign UP interface

This page allows the patient “Ahmed Hamden” to view doctor “Fathi Jelif”’s availability and choose a free slot to book the appointment online with him.

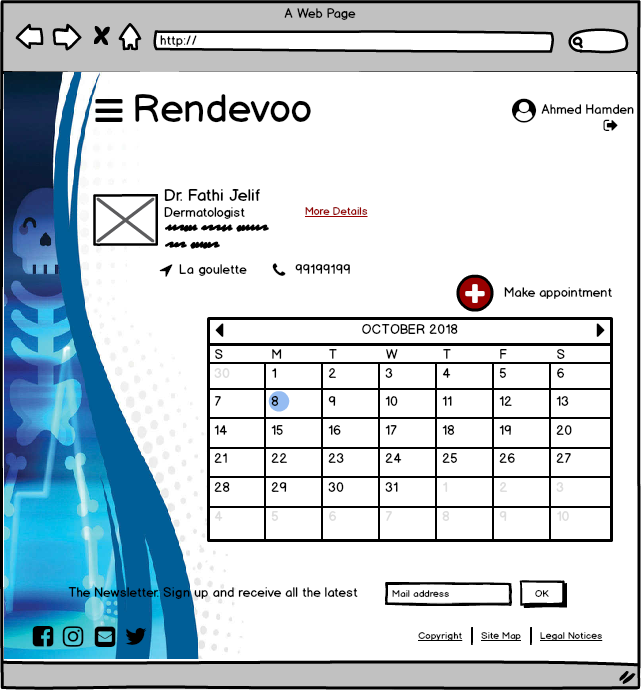


Figure 13 : Appointment booking page

Finally, the patient “Ahmed Hamden” can display his profile page to check the appointment that he made early, also he can manages his account.

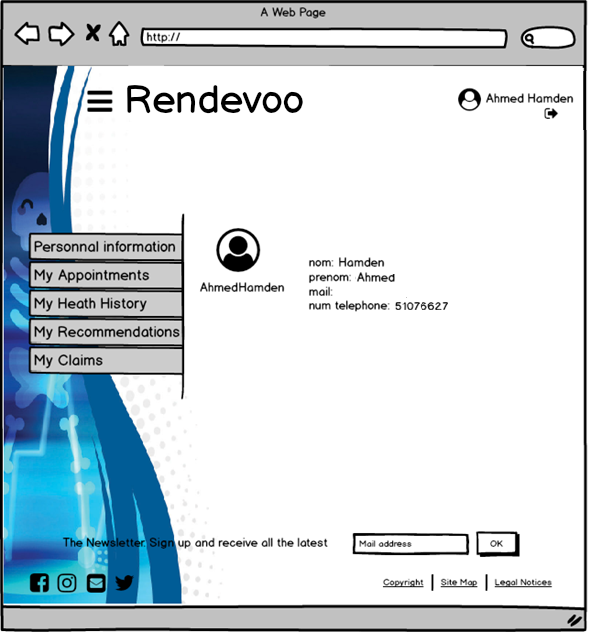


Figure 14 : Profile page

General conclusion

Throughout our project, we will develop an integrated platform dedicated to a wide community willing to help patients to make appointments the best way we can with whatever we can spare

Our mission is to develop an integrated platform (web, desktop and mobile) throughout this semester. Our project will be divided into 3 sprints:

* Sprint 1: Consists of developing a Java EE application to expose all of our functionalities as well as our web services, all of which will be used during the following two sprints.
* Sprint 2: Through which we will be using .Net framework to implement the back-office features for administrators to manage and supervise all of the application events that require special attention and/or authorization.
* Sprint 3: Finally, we will be implementing with Angular and Ionic a client-side mobile application.

By the end of the semester, we will have put all of our knowledge together and we would have a totally integrated and fully functional platform. Together, we can make a difference. Together, we can make this world a better place.