

Pi-DEV Report:

EPIONE:

A Medical Appointment’s Portal

**Submitted By**



**Team Members**

* **Ben Aziza Chaima**
* **Bouhija Mayssa**
* **Garouachi Emel**
* **Louati Souha**
* **Majoul Abdeljalil**

**Table of Contents**

[Chapter I: Project context 7](#_Toc527315494)

[1- Our Client 8](#_Toc527315495)

[2- Problematic 8](#_Toc527315496)

[3- Existing Solutions 8](#_Toc527315497)

[4- Our Solution 9](#_Toc527315498)

[Conclusion 9](#_Toc527315499)

[Chapter II: Global Analysis 11](#_Toc527315500)

[1- Functional requirements 12](#_Toc527315501)

[2- Nonfunctional requirements 13](#_Toc527315502)

[3- Actors 13](#_Toc527315503)

[4- Global Use Case Diagram 13](#_Toc527315504)

[5- Sequence Diagram 14](#_Toc527315505)

[6- Class Diagram 15](#_Toc527315506)

[7- Deployment Diagram 16](#_Toc527315507)

[Conclusion 17](#_Toc527315508)

[Chapter III: Work Environment 18](#_Toc527315509)

[1- Technologies 19](#_Toc527315510)

[1.1 – Java EE 19](#_Toc527315511)

[1.2- .NET 19](#_Toc527315512)

[1.3 - Rest Web Services 19](#_Toc527315513)

[2- Development Tools and Servers: 19](#_Toc527315514)

[2.1- JBoss Developer Studio 20](#_Toc527315515)

[2.2- Visual Studio 20](#_Toc527315516)

[2.3 - Wildfly Server 20](#_Toc527315517)

[2.4 – MySQL /XAMPP 21](#_Toc527315518)

[2.5 – GitHub 21](#_Toc527315519)

[3- Logical Architecture 21](#_Toc527315520)

[Conclusion 22](#_Toc527315521)

**Table of Figures**

[Figure 1: MonDocteur Interface 8](#_Toc527315384)

[Figure 2: Doctolib Interface 9](#_Toc527315385)

[Figure 3: Global Use Case Diagram 14](#_Toc527315386)

[Figure 4: Make Appointement Sequence Diagram 15](#_Toc527315387)

[Figure 5:Analyses Class Diagram 16](#_Toc527315388)

[Figure 6:Deployment Diagram 16](#_Toc527315389)

[Figure 7:Architecture Layer 22](#_Toc527315390)

**Introduction**

Nowadays, computer science occupies a full place in the daily life of the human being as well as on the personal and professional level. Indeed, information technology can provide solutions to various complex problems.

Each company is looking for the best way to increase production and to confront the different conflicts that can be caused by the lack of coordination and organization within its different teams in order to satisfy its clients.

This project arises as an initiative proposing the realization of a medical scheduling platform.

Everyone needs to schedule their time, meetings, appointments in our modern world. Therefore, there are many different online services and apps specially designed for people with the dynamic lifestyle. Such apps allow users to stay connected to everything they need on-the-go. The custom online medical scheduling app should allow users to manage their health information and schedule their visits using any device.

The present work is part of the realization of a semi-annual project by computer engineering students aiming to improve our skills in design and development in computer science field but also to learn ,to identify , and meet the needs of the client in a better way .

The aim is to build a solution with different technologies, this solution provides interaction between doctor and patient involved in make an appointment, update an appointment

Our custom software development company was required to develop a simple-to-use and informative app that will be beneﬁcial both for patients and medical specialists.

We will be interested in the first part of our report to put the project in its general context. In the second part, we will treat the state of the art. Then we will analyze the functional and non-functional needs of our website. Finally, we will handle the design of our project where there will be a general design and another detailed.

# Chapter I: Project context

## Our Client

DeepOR is a powerful platform to monitor and optimize the use of Hospital Operating Rooms (ORs). It offers advanced features based on Big Data and Artificial Intelligence to help make real-time resource allocation decisions. DeepOR's target is to improve quality of healthcare delivery and increase financial outcomes for a hospital.

## Problematic

Today a patient in the yellow pages. He needs to the doctor’s cabinet and checks his availability then wait for the confirmation which costs time and money.

We can find some appointment-making applications such as doctolib, mondocteur, keldoc ... These solutions don’t guarantee time optimization, they are not provided with synchronized tools such as google calendar

and do not allow an intelligent readjustment of the appointments in case of hindrance.

## Existing Solutions

The most known platforms are Mondocteur and Doctolib.

* Mondocteur:

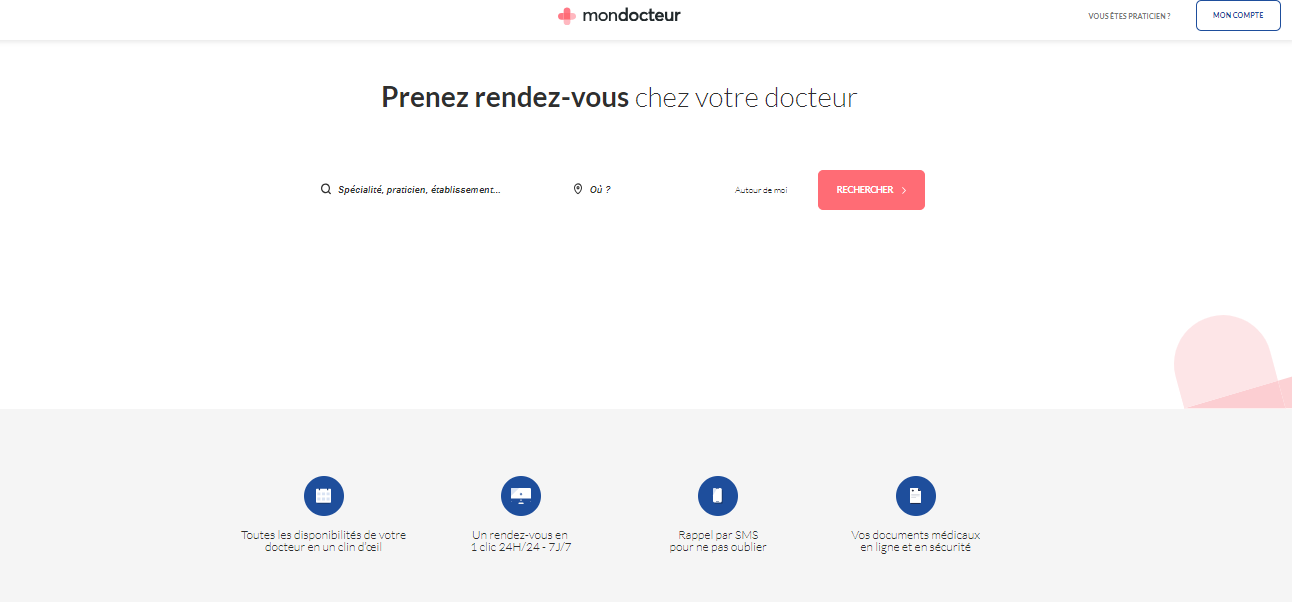


Figure 1: MonDocteur Interface

* Doctolib:

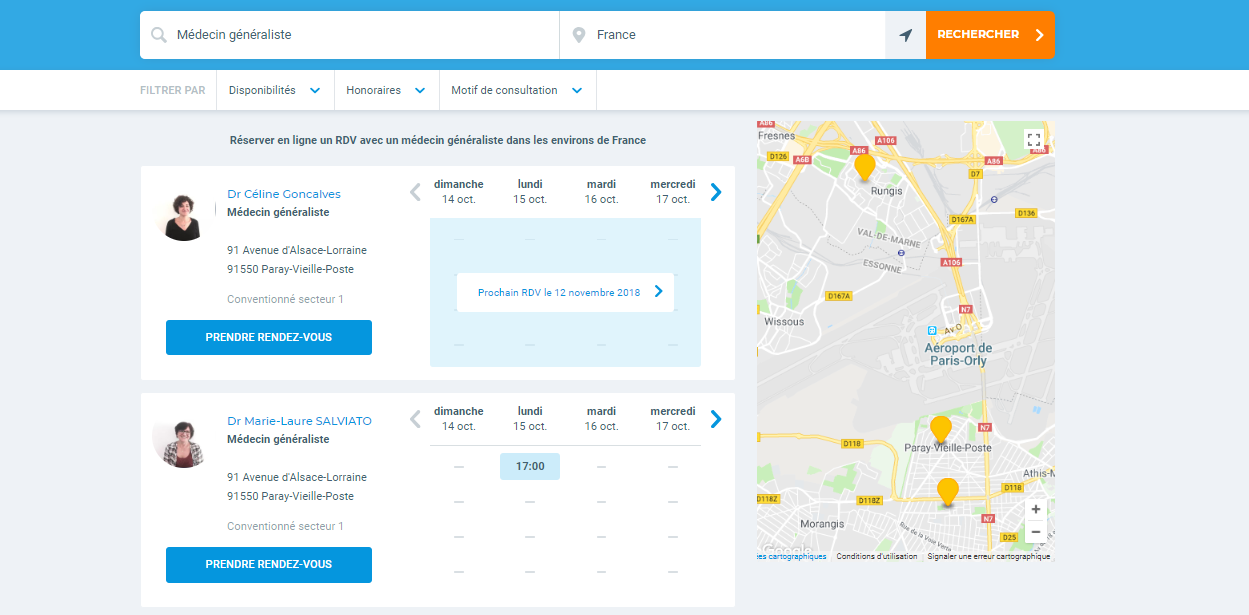


Figure 2: Doctolib Interface

These solutions have several disadvantages, the doctor has no idea about the course of the patient so he can not recommend to the patient other doctors or treatments. Also, the patient cannot ask directly the doctor (chat) in emergency case.

## Our Solution

**Epione** is an intelligent portal for making medical appointments which offers to its users the ability to readjust their appointments. The solution consists in a mobile and a web application. The patient chooses his location and the doctor specialty. He enters his information and chooses a date from the doctor’s calendar. The doctor visualizes all his appointments in his space. In case of cancellation of an appointment, the system oﬀers an alternative appointment. The application also manages saves the patient course which helps the doctors to suggest a better treatment for their patients. Also, the patient is able to ask directly the doctor through the chat option.

## Conclusion

In this chapter, we have carefully studied our competitors in order to analyze their strengths and weaknesses, so we can take advantage of their qualities and avoid their deficiencies.

# Chapter II: Global Analysis

## Functional requirements

**Patient space :**

* Manage an appointment:
  + make an appointment.
  + edit an appointment.
  + cancel an appointment.
  + Check doctor disponibility
* Rate visit.
* View list of doctor.
* Manage account:
  + Edit his account.
  + Consult his account.
  + Registration.
* Chat.
* Visualize course .
* Consult notification .

**Doctor space :**

* Manage appointment :
  + Consult program
  + Define visit
  + Edit visit
  + Cancel visit
* Synchronize program
* Register
* Consult dashbord
  + Write report
  + Edit report
  + Show statistic
  + Export report
* Confirm appointment.
* Specify disponibility .
* Manage patient course .
* Consult notification .

**Administrator Space :**

1. Manage account :
   * Activate account .
   * Desactivate account.
   * Confirm account .
   * Create doctor account .

## Nonfunctional requirements

o **Efficiency**: Specifies how well the application utilizes resources.

o **Flexibility**: If the organization intends to increase or extend the functionality of the software after it is deployed.

o **Performance:** The performance constraints specify the timing characteristics of the software. Certain tasks or features are more time-sensitive than others.

o **Ergonomic:** Design factors must be on point and things well arranged.

o **Robustness:** A robust system is able to handle error conditions gracefully, without failure. This includes a tolerance of invalid data, software defects, and unexpected operating conditions

## Actors

The actors represent the people who are directly going to interact with our system. Our Application is going to have three actors who are the following ones:

• The simple user Who can only access services that do not require an authentication.

• The administrator who manages the other accounts of the system and visualizes the statistics.

• Doctor who can create his own account by the administrator confirmation.

• Patient the internaut who will register and make appointments.

## Global Use Case Diagram

A use case is used in system analysis to identify, clarify, and organize system requirements and it also defines the relation between users and the elements that the system implants.

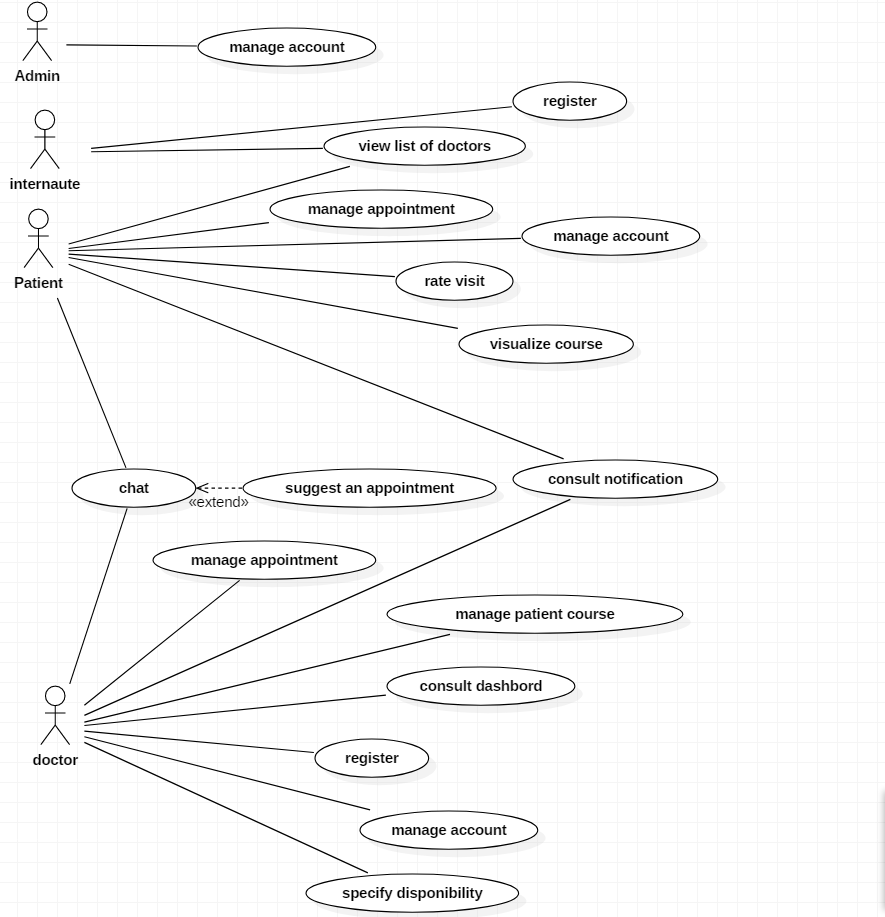


Figure 3: Global Use Case Diagram

## Sequence Diagram

This diagram will describe the steps of making an appointment and the interaction with the system.

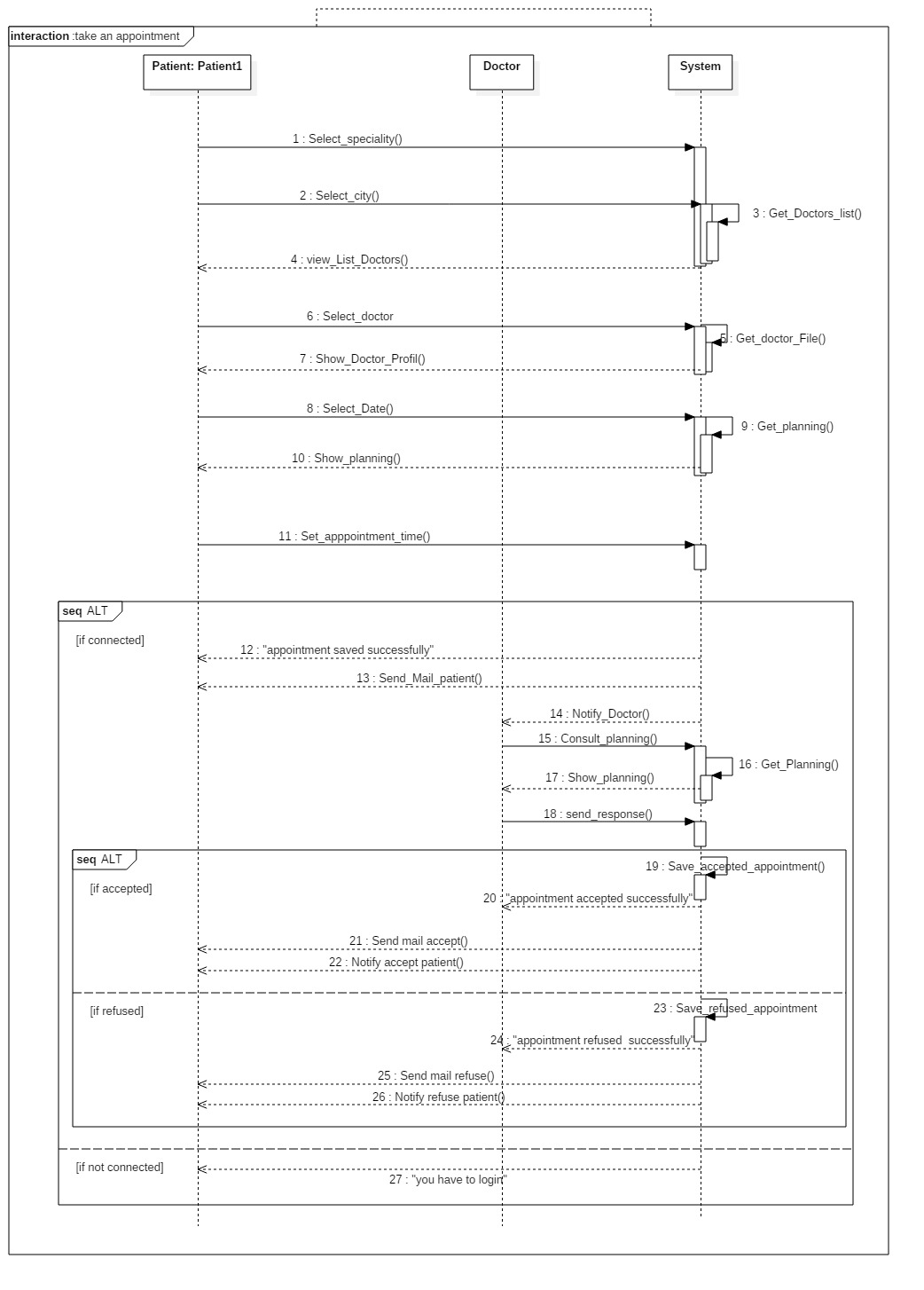


Figure 4: Make Appointement Sequence Diagram

## Class Diagram

In this section we are going to represent the class diagram of the use cases presented above.

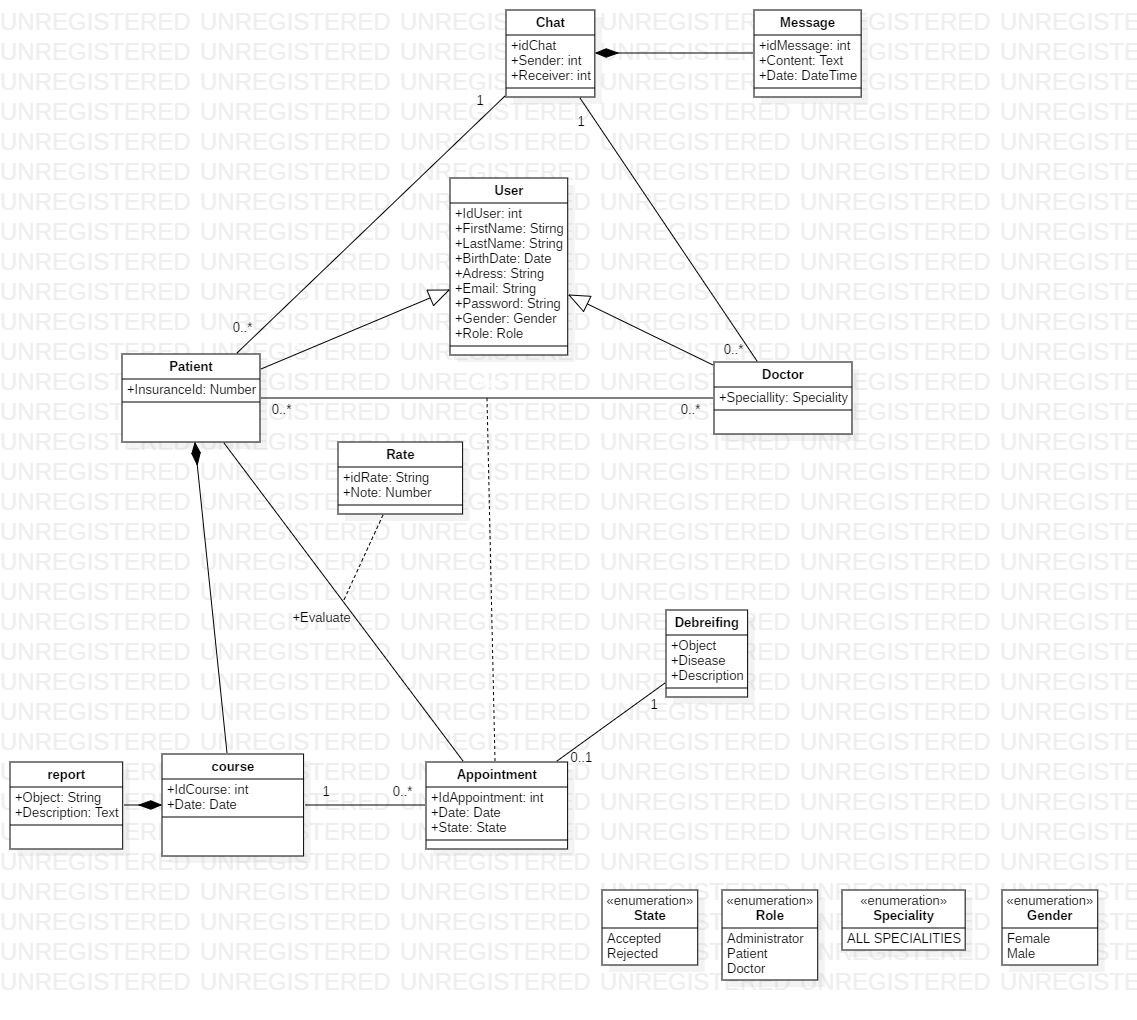


Figure 5:Analyses Class Diagram

## 

## Deployment Diagram

Figure 6:Deployment Diagram

### Conclusion

In this chapter we mentioned the different requirements of our application and the diagrams used to realize the modeling of the application.

# Chapter III: Work Environment

## Technologies

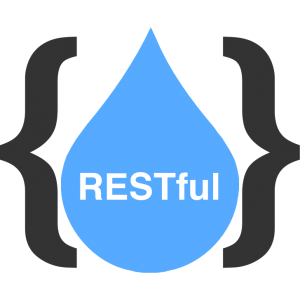
### 1.1 – Java EE

The Java 2 Platform, Enterprise Edition (J2EE) is a set of coordinated specifications and practices that together enable solutions for developing, deploying, and managing multi-tier server-centric applications. Building on the Java 2 Platform, Standard Edition (J2SE), the J2EE platform adds the capabilities necessary to provide a complete, stable, secure, and fast Java platform to the enterprise level.

### 1.2- .NET

The .Net framework is a software development platform developed by Microsoft. The framework can be used to create both  **Form-based**and**Web-based** applications. Web Services can also be developed using the .Net framework. The framework also supports various programming languages such as Visual Basic and C#. So developers can choose and select the language to develop the required application. In this chapter, you will learn some basics of the .Net framework.

### 1.3 - Rest Web Services

REST (Representational State Transfer) is an architectural style for developing web services. REST is popular due to its simplicity and the fact that it builds upon existing systems and features of the internet's HTTP in order to achieve its objectives, as opposed to creating new standards, frameworks and technologies.

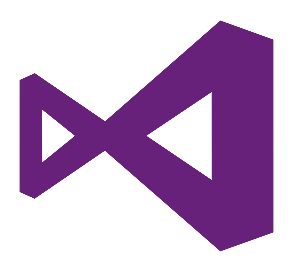
## Development Tools and Servers:

### 2.1- JBoss Developer Studio

 JBoss Developer Studio is a set of eclipse-based development tools that are pre-configured for JBoss Enterprise Middleware Platforms and Red Hat Enterprise Linux. Developers are not required to use JBoss Developer Studio to develop on JBoss Enterprise Middleware and/or Red Hat Linux. But, many and these pre-configured tools offer significant time-savings and value, making them more productive and speeding time to deployment.

This tool is used for the JEE development part.

### 2.2- Visual Studio

Microsoft Visual Studio is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It is used to develop [computer programs](https://en.wikipedia.org/wiki/Computer_program), as well as [websites](https://en.wikipedia.org/wiki/Web_site), [web apps](https://en.wikipedia.org/wiki/Web_app), [web services](https://en.wikipedia.org/wiki/Web_service) and [mobile apps](https://en.wikipedia.org/wiki/Mobile_app). Visual Studio uses Microsoft software development platforms such as [Windows API](https://en.wikipedia.org/wiki/Windows_API), [Windows Forms](https://en.wikipedia.org/wiki/Windows_Forms), [Windows Presentation Foundation](https://en.wikipedia.org/wiki/Windows_Presentation_Foundation), [Windows Store](https://en.wikipedia.org/wiki/Windows_Store) and [Microsoft Silverlight](https://en.wikipedia.org/wiki/Microsoft_Silverlight). It can produce both [native code](https://en.wikipedia.org/wiki/Native_code) and [managed code](https://en.wikipedia.org/wiki/Managed_code).

This tool is used for the .Net Development part.

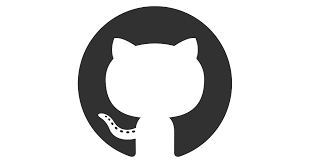
### 2.3 - Wildfly Server

 WildFly formerly JBoss, is an application server Open source Java EE written in Java server can be used on any operating system supplying a virtual machine Java (JVM). (The used version is the 9.0) we will deploy our Web Service on this server.

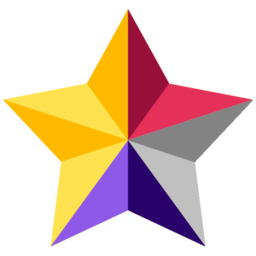
### 2.4 – MySQL /XAMPP

 MySQL A basic relational database server that stores information in tables, in the form of data subjects by groups. The tables are linked by relationships. XAMPP is a platform allowing the development ofdynamic the Web applications, this platformgathers a set of software making it possible to set up a Web server,  MySQL a MySQL database server.

### 2.5 – GitHub

GitHub is a Web-based Git repository hosting service, which oﬀers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.

2.6 StarUML:

StartUml is a UML modeling software available in Open source. Via this platform, it is possible to design diagrams such as creating class diagram, object, activity or sequences.

## Logical Architecture

The software logical architecture is the n-tiers architecture which is a client server architecture in which presentation, application processing, and database management layers are separated.

• Presentation Layer It contains the User Interfaces (UI) that will be show to the user.

• Business Layer It contains treatments representing the business rules and it controls the application’s functionality by performing detailed processing.

• Data Access Layer This layer is responsible for mapping logic objects and data components. It’s the only layer that is linked with the database.



Figure 7:Architecture Layer

## Conclusion

In this chapter we detailed the different tools that we will use to realize our application and the system’s architecture.