University of Cassino and Southern Lazio

Distributed Programming and Networking

Project Report

Husam Nujaim **Anastasiia Rozhyna** **Tewele Weletnsea Tareke**

June 9, 2021

1 Acknowledgement:

We are really grateful for having a chance to meet so many wonderful people and professionals who led us through this semester period in Italy. We want to use this opportunity to express our deepest gratitude and special thanks to our Prof. Mario Molinara who is extraordinary in teaching and guiding students. Also, we would like to say thank you to professor Mario Molinara for his careful and precious guidance which are extremely valuable to our lessons(theoretically and practically) and helpful for the next path of our studies. The teaching staffs of university di Cassino also deserve our sincere thanks for sharing us their discussion and exchange of ideas.Their advice, assistance and patience are greatly appreciated.

Contents

2 Abstract

MAIA EyeCare system provides a computerized platform to manage the online booking of eye clinics. Also, the platform provides a computer-aided diagnosis system for Diabetic Retinopathy (DR). The system has three main actors, the first one is the admin. S/he adds ophthalmologists details and allocate their appointments on the platform. The second actor is the patient. S/he registers in the platform, and books an available appointment with the desired ophthalmologist. The last actor is the ophthalmologist. S/he uploads the retinal Fundus images of the patient's retina to the platform, and then the platfrom uses deep learning to diagnose the retinal Fundus image of the patient. Only then can the patient access the diagnosis of his case. Moreover, the system manages the financial transactions of the patient.

3 Introduction

3.1 Preface

In this project, we provide a system that computerizes MAIA clinic EyeCare for online healthcare system. Online Eyecare system have significant advantage in terms of time and cost for diabetic Retinopathy patients. And also, This system helps for the ophthalmologist to save their time in handling and processing the result of the patient. The system provide good environment for booking an appointment with doctors at anytime and with little effort. In this project, We have developed an online platform that connects the patients, ophthalmologist and system administrative of the hospital. It provides confidentiality for the users of the platform which makes the online booking system very secure for both sides(the

Patient and Doctor).

3.2 Problem Statement

The Existed booking system is manual as all the work is done and kept in files. The bookings are done by filling in forms manually which are submitted to health officers therefore taking a lot of time to book the right time slot for the patients to be seen by the ophthalmologist. This is obvious that the performance of the current system is slow, complex and insufficient.They face the problem of data accuracy, ambiguity,costy and not being able to collect the required data in time. The patient can't know in which time she/he is going to be and the name of the ophthalmologist.

Therefore it is necessary to develop an automated online system that helps patients to automatically book an appointment with preferred doctor in anytime and see the number of patients who are registered for treatment. This system can easily track the manually booking and processing system.The patients have an authentication to view the result of the examination online and check any update information from the doctor.

3.3 General Objective

To develop a MAIA EyeCare online booking system for patients and ophthalmologist.The patient can book an appointment with preferred doctor and view the result after diagnosed and uploaded by the ophthalmologist.

3.4 Specific Objectives

To analyze the existed booking system.

To design a convenient and flexible online EyeCare clinic system.

To implement the online EyeCare clinic.

To test and validate the system.

3.5 Functional Requirements

This system manages the processes of online booking system platform, it contains three main actors which are Admin, patient and Ophthalmologist. The admin is responsible for adding ophthalmologists details, allocate their appointments on the platform and updating the system with time. The patient is responsible for registration and booking an appointment with available doctors. The Last actor is Ophthalmologist and his/her role is to diagnosed the image with deep Learning and upload the result to system so that the patient can see it directly.

4 System Analysis and Design

4.1 software development life cycle

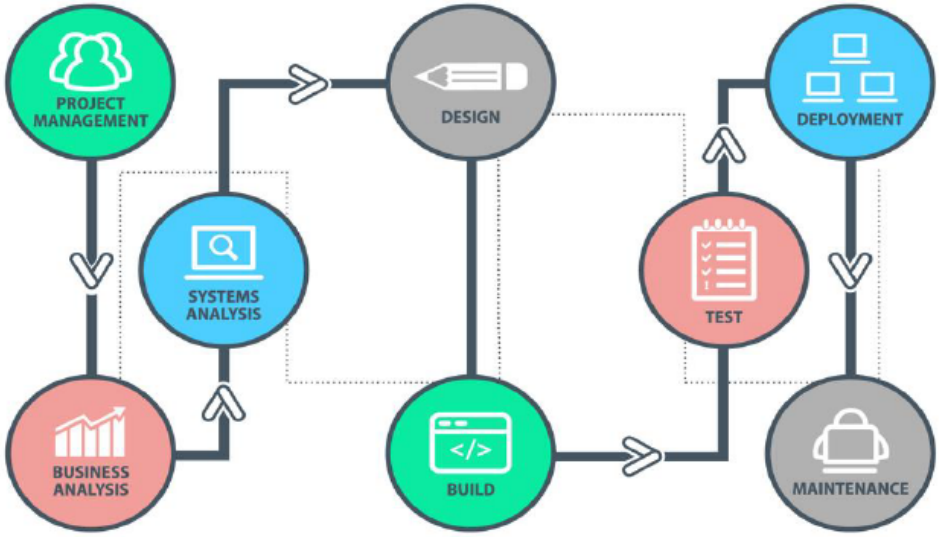

Figure 1: software development life cycle

4.2 Project development Architecture:

This platform used as web application services and for that we use MVC software design pattern that separates an application into three components that have communication among them to complete process. In the MVC when we built the system as back-end and front-end that divide the view into external part it represents the same client/server architecture, In the following picture the view part it means the way of representing the data after getting them for the database moreover it would be JSON.

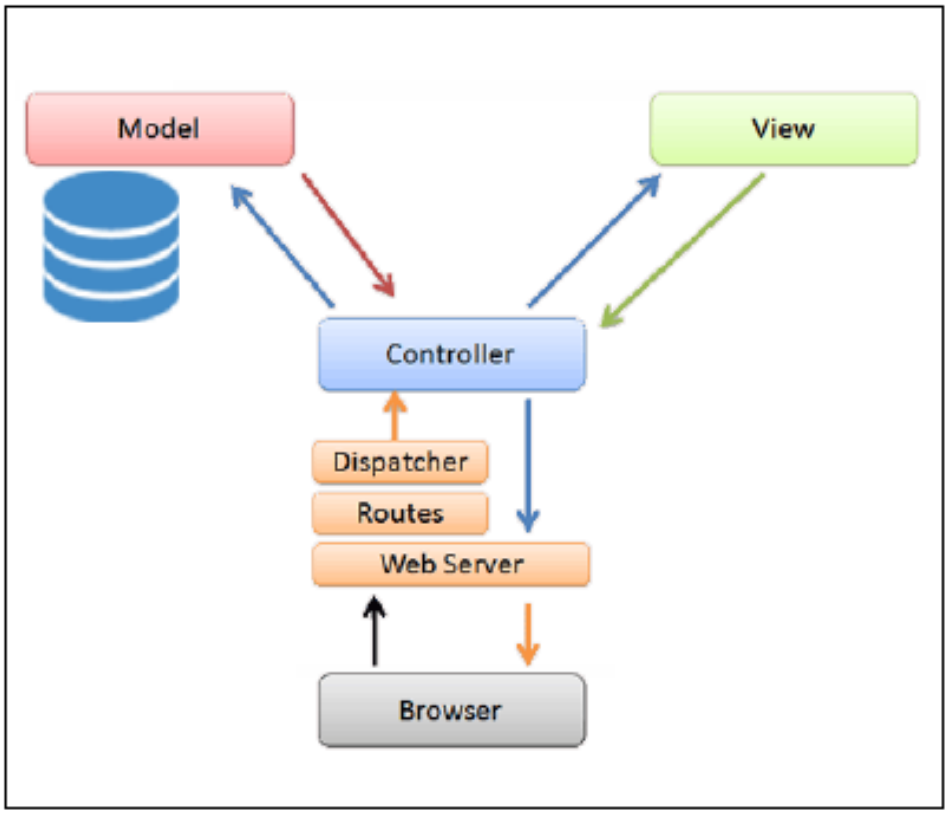

Figure 2: MVC connected to the Internet

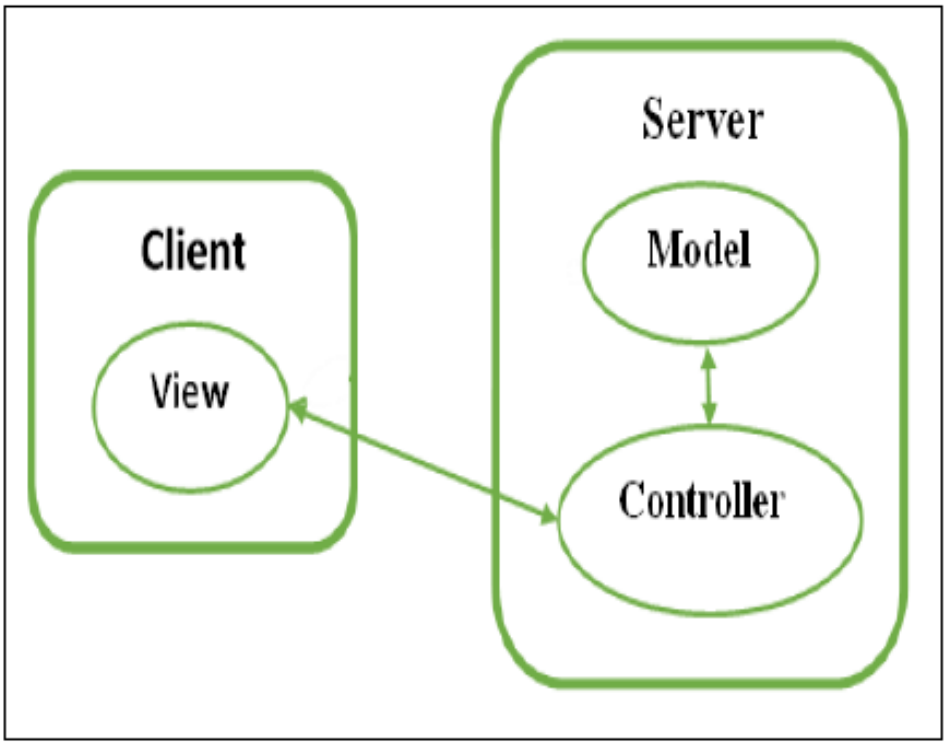

Figure 3: MVC as Client/Server Diagram

4.3 Template Design pattern

TDP is a behavioral design pattern that defines the skeleton of an algorithm in the superclass but lets subclasses override specific steps of the algorithm without changing its structure.This pattern has two main parts: It can be implemented as a method in a base class (usually an abstract class).

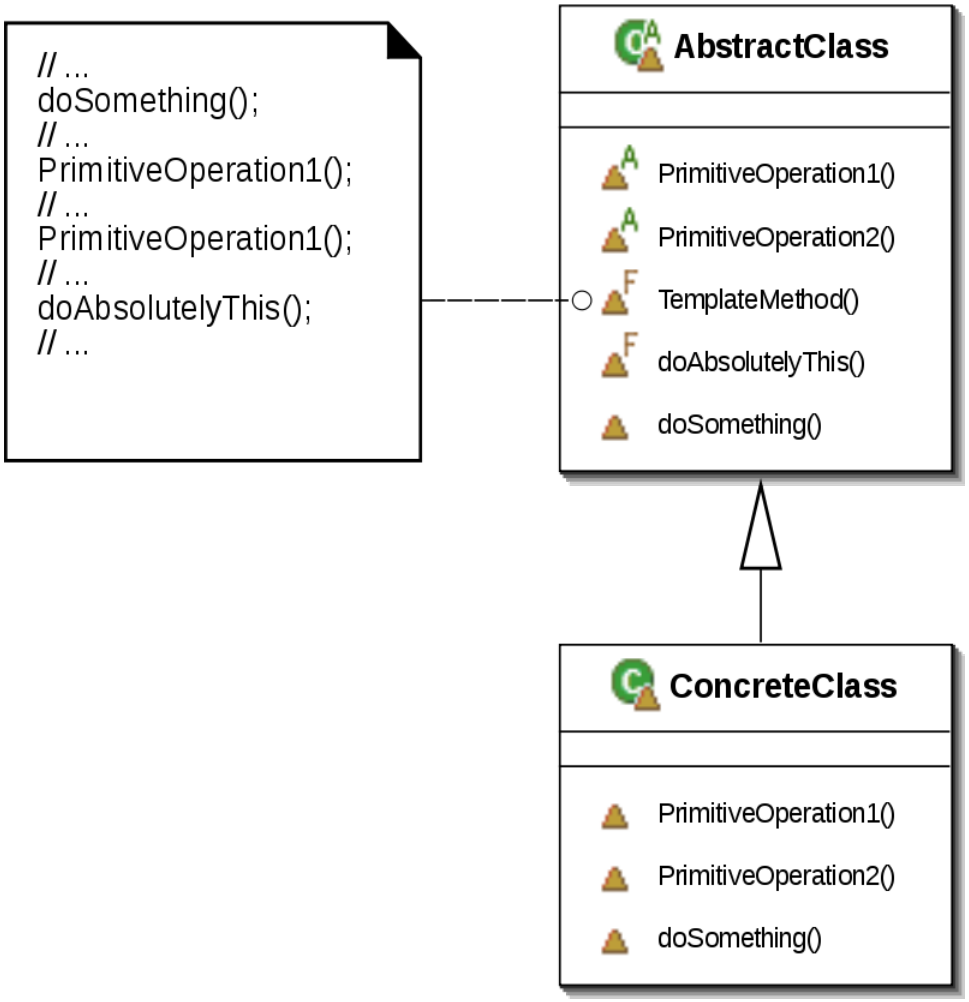

Figure 4: Template Method Pattern

4.4 Use Case Diagram

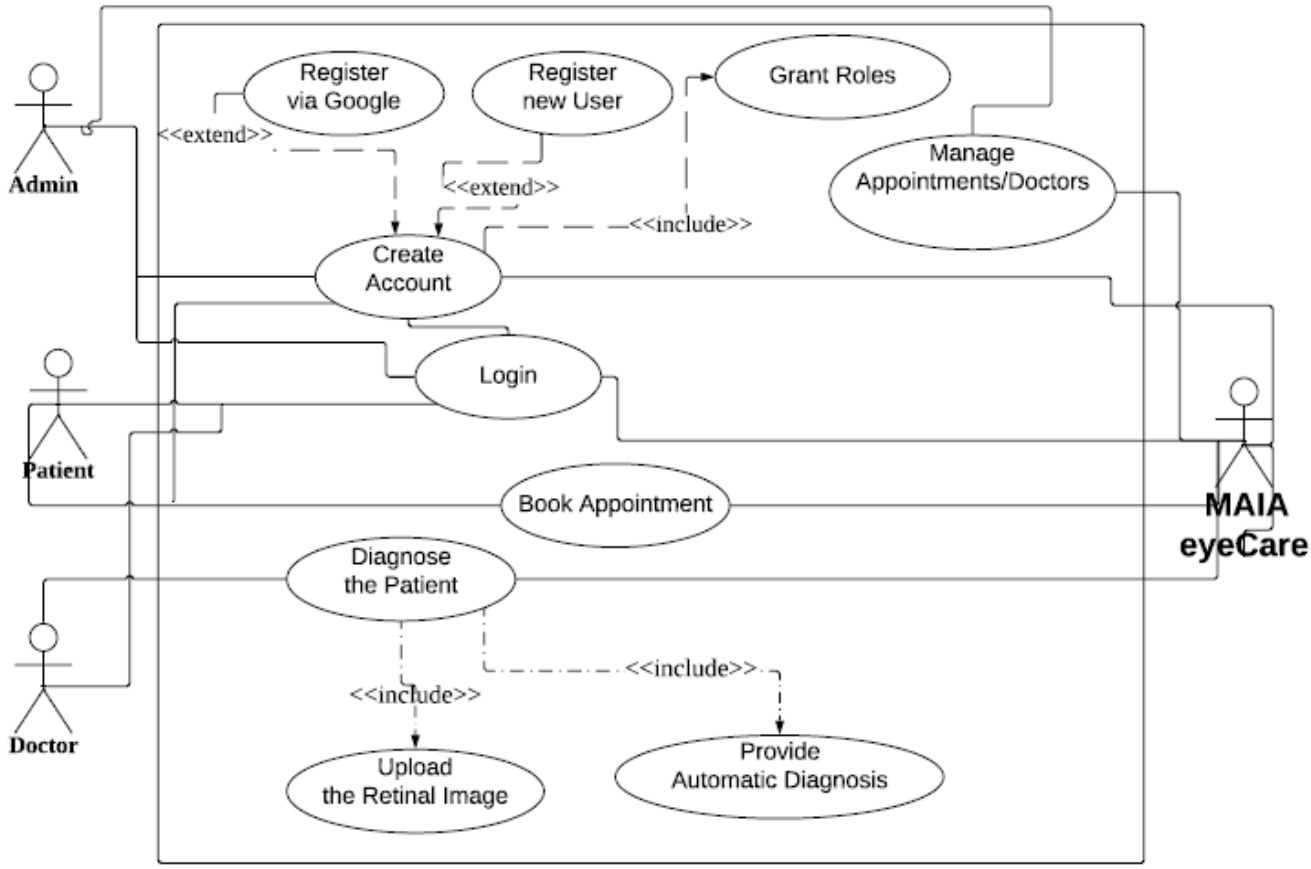

Figure 5: Use case Diagram

5 SYSTEM IMPLEMENTATION

5.1 Preface

The system is developed as a web app that has two parts, the Back-End part that serves as the server side and the Front-End part that serves as the client-side; Moreover, the system

was build based on MVC software design pattern which divides the program into three interconnection element. For processing the database, the system uses CRUD that it used to retrieve and manipulate data.

5.2 System Technology Framework:

1.Visual Studio Code is a free source-code editor made by Microsoft that redefined and optimized for building and debugging modern web and cloud applications.

2. IntelliJ IDEA is an integrated development environment developed by JetBrains and written in Java for developing computer software.

3. MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

4. Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information. It was used to manage the Back-end project that build with Spring Framework.

5. NPM is a package manager for the JavaScript programming language. It is the default package manager for the JavaScript runtime environment Node.js. it was used to manage the Front-End project that build with VueJS Framework.

6. Spring Framework is java application framework that provide foundational support for different application architectures as needed, including messaging, transactional data and persistence, and web. It used for Building a RESTful Web Service.

7. Vue JS is an open-source model–view–viewmodel JavaScript framework for building user interfaces and single-page applications. It used in the front-end for building.

8. Axios is a library that used for mange the HTTP client request. This library can handle all types of requests and between the front-end and the back-end.

9. Vuex used to manage the state of all components of the application and to ensure that the state changes in a predictable manner; we use VUEX library, which is a state management model, developed for VUE to centralized storage.

10. Element UI It is a component library that mainly used for developers, designers, and product managers to make the front end looks professional.

11. PyTorch is a deep learning framework which we used to train our model.

12. Deep JAVA Library (DJL) is used to deploy the deep learning model in our spring application.

5.3 System common code design

Building all the classes to work in the environment adopting the spring boot to manage the HTTP request and response, as well as, the internet services. As known Spring mainly uses the MVC architecture and for the database use MyBatis that implement the CRUD this part was used to handle working with the data to store, retrieve, and manipulate data.

The first part was the user and their rules that control the way of accessing the data and give every user it is own permission according to what they allowed to do. For the security using WebSecurityConfigurerAdapter that is part of the spring-boot framework to control the working session on the internet according to the configuration that programmer use to customize it with. The first part was blocking the access to the web page and allowing a few links which can be accessed without and authorize.

Figure 6

In this code, shows the first line which tells the WebSecurityConfigurerAdapter this is the entry for the security configuration and the second line tells which link could be accessed without any security while in the third line it tells that any other request needs to be authorized to be accessed.

Figure 7

This piece of code shows the login process and the success Handler with the failure Handler as well as the link which could be used to login

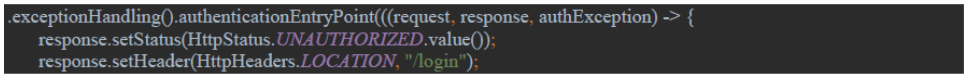

Figure 8

This part of the code handles the unauthorized access to the web page, which redirects the request to the login page.

```
.logout().deleteCookies("JSESSIONID").logoutUrl("/logout")
.logoutSuccessHandler((req, resp, auth) -> {
    resp.getWriter().print(auth.getName());
})
```

Figure 9

This part of the code used to handle the logout process. When we work with the database we need the following part of the code:

After making all of those parts altogether, We needed to test them and see if there is any problem, for that, We use the SpringBootTest package to check that everything is working fine. While building the entity we have some date that saved in the database as symbols and when we send it to the Front-end it has to be as full text, (ex. The Gender in the database we store M for male and F for Female) but to show them for the user it has to be written in the full form, not as a symbol for that purpose I used the emu as showing in the code below.

```
    Gender(String gender) {
        this.gender = gender;
    }
    public String getGender() {
        return gender;
    }
}
```

Figure 10

The last part of the server is the controller which is used to manage the communication between the server application and the outside 1. Those classes have to be declared as a controller by using the annotation "@RestController" to tell that this is a controller that could handle the HTTP request the function in this kind of classes have to be declared using one of the following annotations 2. "@RequestMapping" this manage any kind of request a general mapper 3. "@GetMapping" this manage only the GET request in the http 4.

"@PostMapping" this manage only the POST request in the http 5. "@PutMapping" this manage only the PUT request in the http 6. "@PatchMapping" this manage only the PUT request in the http 7. "@DeleteMapping" this manage only the PUT request in the http This annotation tells the spring to map those function as HTTP request handler and in the annotation, we specify which URL they are handling, moreover, if we want the class to be mapped and a request handler we have to use the same annotation we use for the function in the same way.

```
@RequestMapping("about")
public ResponseEntity<String> aboutPage(){
    return ResponseEntity.ok("this is the MVO & BVO trading platform");
}
```

Figure 11

Front-End

This part of the project represents the client-side and in this part, I used the VUE js to make the web page and handle the views for the data. The system has used a lot of libraries the most important one are:

1. Axios: use to handle the Ajax request and deliver the response to the program, in other words it works the middleware between the Back-end and the Front-End.

2. VueX: this use to store the data in and make them available as general among the component and all the parts in the vue project.

3. Element-ui: which used for the style and for the html component in the project.

4. bootstrap-vue: a style library used especially for the login page.

5. js-cookie: used to manage the cookies for the front-end.

6. nprogress: used to show a progress bar while loading the pages.

```
const service = axios.create({

  timeout: 5000 // request timeout

})
```

Figure 12

The most important one was the AXIOS so We use it as a Services which allow the system to access it without making the configuration every time We want to use it.

References