Faculty Of Engineering

Computer Engineering Department

Helwan University

Graduation Project Book For Bachelor's Degree

Smart Integrated Electronic Service System

Supervised By

Professor Dr. El Sayed Mustafa

Prepared By

Amira Bahaa El-Din Maher Abdelmoniem

Aya El Sayed Mohamed Abdelrazek

Aya Yehia Khaled Hamouda Ahmed

Eman Amr Ali Hamed

Gehad Ehab Kamal Hassanien

2022-2023

Acknowledgment

We would like to express our sincere gratitude and appreciation to **Professor Dr. El Sayed Mostafa**, who oversaw our graduation project. You have given us valuable suggestions, discussions, knowledge, and advice throughout the entire journey. Our project has been greatly shaped by your advice and persistence.

We deeply appreciate your unwavering faith in our competence and your encouragement of us to achieve even greater heights in our graduation project. Your endless support has inspired us to work towards excellence.

We also want to thank the teaching staff in the computer engineering department for their ongoing help and support over the past five years. It has been a pleasure to share our graduation project with all of you here.

Once more, we would like to express our deepest thanks to **Professor Dr. El Sayed Mostafa** for his exceptional mentoring and guidance. We are incredibly fortunate to have had the chance to work with you.

TABLE OF CONTENTS

Table of contents

TABLE OF CONTENTS

TABLE OF ABBREVIATIONS

Table of Abbreviations

The Abbreviation	Stands For
EHR	Electronic Health Repository
HU	Health Unit
ERD	Entity Realtionship Diagram
DFD	Data Flow Diagram
UI	User Interface
DTO	Data Transfere Object
JWT	Json Web Token
Sql	Structured Query Language
DBMS	Data Base Management System
Xampp	X-Operating System, Apache, Mysql, Php, Perl
TC	Test Case

LIST OF FIGURES:

List of figures:

LIST OF FIGURES:

CHAPTER 1: INTRODUCTION

Chapter 1: Introduction

ABSTRACT

In the current world, healthcare has become an essential part of our lives. Due to the increase in patients and medical staff, a system that integrates all duties and data related to healthcare has become essential. Our graduation project aims to develop an application that integrates the roles of patients, doctors, lab technicians, and pharmacists to improve healthcare services.

This system will streamline emergency procedures, ensure accurate patient information collection, and maintain comprehensive medical histories. Additionally, it will help prevent the illegal dispensing of medications without verified prescriptions and enable tracking of doctors' diagnoses for improved accuracy.

The system will also serve as a memory aid for patients who struggle to recall their medical history in detail. In emergency situations, healthcare professionals can quickly access a patient's records through the system and provide appropriate first aid. Furthermore, they can use the system to contact the patient's family or friends for support.

Overall, our project aims to enhance healthcare delivery by leveraging technology and facilitating efficient communication among all stakeholders in the medical system.

CHAPTER 1: INTRODUCTION

INTRODUCTION

Healthcare has grown to be a vital component of our lives in the modern world. A system that combines all the responsibilities and data involved in healthcare has become crucial due to the growth in both patients and medical personnel. To deliver better healthcare services, our graduation project intends to create an application that combines the patient and doctor roles with lab and pharmacy data.

Smart integrated electronic service system, Before talking about the project itself let's talk about the idea behind it first, Since the beginning we had the desire to make a project that applies the sustainability concept, To serve the entire country not only a few categories of the community, We didn't want our project to be a conventional graduation project or just an idea that can be so sophisticated to be applied in the real. Our Project is a service for each citizen, no matter his gender, marital status, age, job, or even his level, it's literally for each one!

We've chosen the medical field as nothing more important than Man's health, however there are also a lot of fields that need to be covered but comparing to a good health system? There're no competitors. In our system all the actors in the medical system are included starting with the doctor, pharmacist, Lab tech, emergency doctor, Patient. The service aim is to quick the procedure of the emergency, to help the clinic staff to gather the patient's information accurately, to record all the patients history, To stop printing papers and stop the pollution caused by all those papers. As we all aim to go green and to go digital. so, we don't need hardcopies anymore either for lab results or prescriptions.

The System aims also to control the discharging of medicines without verified prescription, to limit the illegal discharging that can be done. It also follows each doctor's diagnosis so that if a patient has a complain it could be proved if the doctor has made a wrong diagnosis at the first place, and that will make doctors more accurate to each diagnosis and medicines they right as everything is recorded!

CHAPTER 1: INTRODUCTION

When the clinic staff asks you for your previous medicines or your previous illness or even your surgeries, most of the people can't remember everything in detail and of course that's normal and that's what computers are invented to be our memory helper!

When there is an emergency case with the aid of the System, they can easily scan the card and view all his records and if he suffers from serious diseases so that they can help him with the suitable first aid, they also can get use of the option of contact members and contact with his family or friends to come and check on him.

The system makes everything user friendly, smart, and quick with only one scan.

CHAPTER 1: INTRODUCTION

PROBLEM STATEMENT

The excessive consumption of paper in the medical sector is contributing to deforestation, increased energy usage, and water shortages, thus harming the environment and sustainability efforts. Countries like Egypt import large amounts of paper pulp, leading to a per capita paper product usage of 30 kg. The adverse effects of paper production and waste disposal on the environment are evident. There is a pressing need for alternative solutions to reduce reliance on paper-based systems. Additionally, the current paper-heavy healthcare processes lead to inefficiency, data loss, and unnecessary resource consumption.

Solution:

1. To address the environmental and sustainability challenges posed by excessive paper consumption in the medical sector, our proposed solution is a digital healthcare application. This application aims to replace paper-based medical records and documentation with a user-friendly, secure, and efficient digital platform. By doing so, we can achieve the following objectives:
2. Resource Conservation: By eliminating the need for physical paper storage, the application contributes to conserving trees, reducing water consumption, and minimizing energy usage associated with paper production.
3. Efficient Resource Management: The digital platform enables healthcare providers to access patient information and medical records instantly, reducing the need for physical infrastructure and large-scale medical facilities, leading to more sustainable resource management.
4. Minimized Transportation and Emissions: Patients can remotely access their medical records and lab results through the application, reducing the need for physical visits to healthcare facilities and lowering carbon emissions from transportation.
5. Enhanced Collaboration and Waste Reduction: The platform centralizes patient information, eliminating unnecessary duplication of tests and procedures, promoting efficient use of resources, and reducing waste generation.

6. Data-driven Insights and Evidence-based Medicine: By securely storing patient data over time, the application enables healthcare providers to analyze health trends and optimize treatment plans, contributing to sustainable healthcare delivery.
7. Improved Patient Safety: The platform ensures accurate and up-to-date medical records are readily available, reducing the risk of errors, unnecessary treatments, and adverse environmental impacts.
8. Long-term Sustainability Planning: Comprehensive patient data available through the application allows healthcare organizations to conduct research, plan healthcare services, allocate resources effectively, and design public health initiatives to promote sustainable healthcare systems.
9. By implementing this digital healthcare solution, we aim to reduce paper consumption, improve healthcare efficiency, and contribute to a more sustainable and environmentally friendly medical sector.

Chapter 2: Related Work

1. EpicCare, In the United States Epic Systems Corporation provides an EHR software.

   It integrates various healthcare stakeholders such as hospitals, clinics, laboratories, and pharmacies into a single platform for efficient data sharing.

   Main Points:

   - EHRs streamline communication between healthcare providers by providing a centralized repository of patient information.

   - Integration of laboratory results with EHRs allows for quick analysis and interpretation by physicians.

   - Improved accessibility to patient data reduces duplication of tests and minimizes medical errors.[7]

2. Babylon Health, a telemedicine platform in the United Kingdom that integrates with existing health systems.

   It allows patients to consult doctors remotely via video calls while ensuring seamless integration of their medical records.

   Main objectives:

   - Telemedicine platforms provide convenient access to healthcare services for patients in remote areas or those with limited mobility.

   - Integration with EHRs ensures continuity of care by enabling physicians to review past medical records during virtual consultations.

3. The Danish Health Data Network (MedCom)

   An HIE platform in Denmark that connects hospitals, general practitioners, pharmacies, and laboratories. It enables seamless data exchange and collaboration among healthcare stakeholders.

   Main Points:

   - HIEs promote interoperability by standardizing data formats and protocols for secure information exchange.

   - Integration of HIEs with EHRs allows healthcare providers to access comprehensive patient records from various sources.

   - Improved collaboration among health stakeholders leads to better care coordination, reduced redundancies, and improved patient outcomes.[6]

Chapter 3: Analysis And Requirement

STAKEHOLDERS

Actor	Role
Administrator	- Each admin has an interface on the website that makeshim see the new users who wants to create new accounts. - Each admin has the ability that helps him to approve andaccept the new users. - Only admins have an admin code to help them entering. the website
Patient	- The patient needs a profile for all his information, old and new, and has the access to see them whenever he wants. - Each patient has an option to report any wronginformation. - Any patient can see his prescriptions, lab results,diagnoses results. - The patients cannot edit any of this information.
Doctors	- Any type of doctor can make a request to have an accounton the system. - These doctors will be classified according to their jobs. - Each type of doctor must have a reader to read the smart card of the patient and do what he wants according to his position.

Diagnostic doctor	- This type of doctors can see the history of the patients,overview states and their old prescriptions. - Each doctor has a specialization. - Each doctor must have an official workplace like a hospitalor private clinic. - Each diagnostic doctor can edit and update the information of the patient like adding a new prescription, new diagnostic and updating the old information.
Pharmacist	- The pharmacists can see only the prescriptions of patientsto get them medicines. - Each pharmacist has an attribute of the pharmacy namethat he works in. - Each pharmacist has an address for this pharmacy.
Lab technician	- The lab technician can see only the lab results which thepatient needs to do. - The lab technician can just upload the lab results for the patient on the website. - Each lab technician has 2 attributes which include the lab. name and the lab address.

Emergency doctor	- This type of doctors works in the hospital like the diagnostic doctors but has a different functionality. - These doctors can see the overview state of the patient so they can rescue them for death due to having importantinformation from the card. -

CHAPTER 3: ANALYSIS AND REQUIREMENT

SYSTEM REQUIREMENTS

1. FUNCTIONAL REQUIREMENTS

Main Functions	Description	Actor
1. Sign up	User should create an account once, to be able to use the application, so he should input the required credentials to become a user.	Doctors, emergency doctors, lab technician, pharmacists and patients.
2. Log in	After registration, user will be able to login whenever he needs, by entering the username and password he registered with.	Doctors, emergency doctors, lab technician, pharmacists and patients.
3. Emergency information	This function is accessed by the Emergency doctor, to show the important and critical information for each patient that can help the emergency team to provide the best aid for the patient, like the chronic diseases, tumors, relatives contacts and recent medicines or diagnosis.	Emergency Doctor
4. Patient Basic Information	This function enables user to see his own basic information	patient
5. Scan Results	This function enables the patient and the specialist doctor to see the results of the scans.	Specialist doctor and Patient
6. Uploading scan results	This function enables the lab technician to upload their reports for the patient on a specific scan type.	Lab Technician
7. Test Results	This function enables the patient and the specialist doctor to see the results of the Tests required.	Specialist doctor and Patient

8. Uploading test results	This function enables the lab technician to upload their reports for the patient on a specific Tests required.	Lab Technician

2. Add prescription	This function enables the specialist doctor to add the prescription for the patient, add the diagnosis, write if required tests or scans, any medical requirements, medicines and their doses.	Specialist Doctor
3. Get Doctor Information	This function enables the doctor to see his profile after logging in.	Specialist Doctor
4. Contact Us	This function enables the patient to send any query or question.	Patient
5. Lost Card	This function enables the patient to request for a new card and write in details how his card is lost , to allow the administrator to stop the previous card.	Patient
6. Get all contact us requests	This function enables the administrator to see all the contact us requests, read to be able to contact them back.	Administrator
7. Get all lost card requests	This function enables the administrator to see all the lost card requests.	Administrator
8. Get Medicines required	This function enables the pharmacist and the patient to see all the medicines required by the specialist doctor, their doses and period.	Pharmacist and patient.

9. Get current medicines	This function shows the current medicine names taken by the patient, their period and doses	Patient and specialist doctor.
10. Check for medicine discharged	This function provides the pharmacist the ability to mark that the medicine is discharged, so that the patient can not discharge it again without a new prescription.	pharmacist
11. Stop lost card	This function enables the administrator to stop the lost cards that has been requested to be stopped.	Administrator
12. Reply to the contact us request messages	This function enables the administrator to reply to the users and contacts them.	Administrator
13. Forget Password	This function enables the user to send a forget password request to be able to reset password.	Users
14. Reset Password	This function enables the user to reset their passwords.	Users

CHAPTER 3: ANALYSIS AND REQUIREMENT

2. NON-FUNCTIONAL REQUIREMENTS

Requirements	Description
1- Availability	Our app is available every time and wherever there is an internet connection, and it is a great privilege as some apps. would have downtime for updates, backups, and recovery
2- Usability	Our app is easy to use, without too many instructions or obstacles.
3- Performance	Our app has short response time to respond to user's request and low utilization of phone resources.
4- Reliability	Our app keeps updating its performance and accuracy of its results and responses by the time. That avoids failure and downtimes.
5- Efficiency	Our app uses low resources of storage space, memory and bandwidth.
6- Security	Our app provides user verification by email and allows easy password changing in case of user forgets his password or try to protect himself from hacking.

CHAPTER 3: ANALYSIS AND REQUIREMENT

4. SOFTWARE DESIGN

4.1 Project schematic Diagram (Doctors and Administrator aspect)

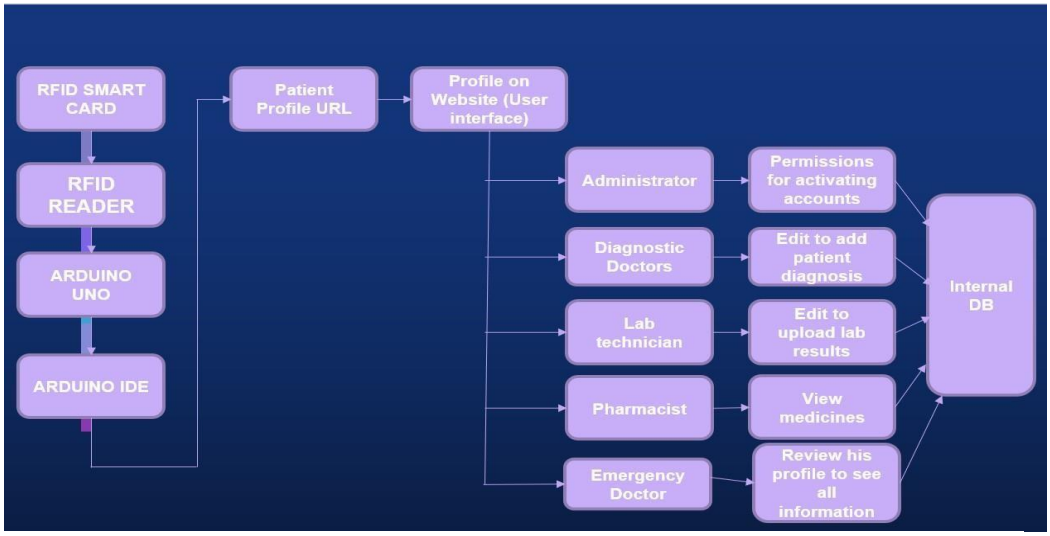


Figure 1 Project schematic Diagram

CHAPTER 3: ANALYSIS AND REQUIREMENT

4.2 Entity Relationship Diagram

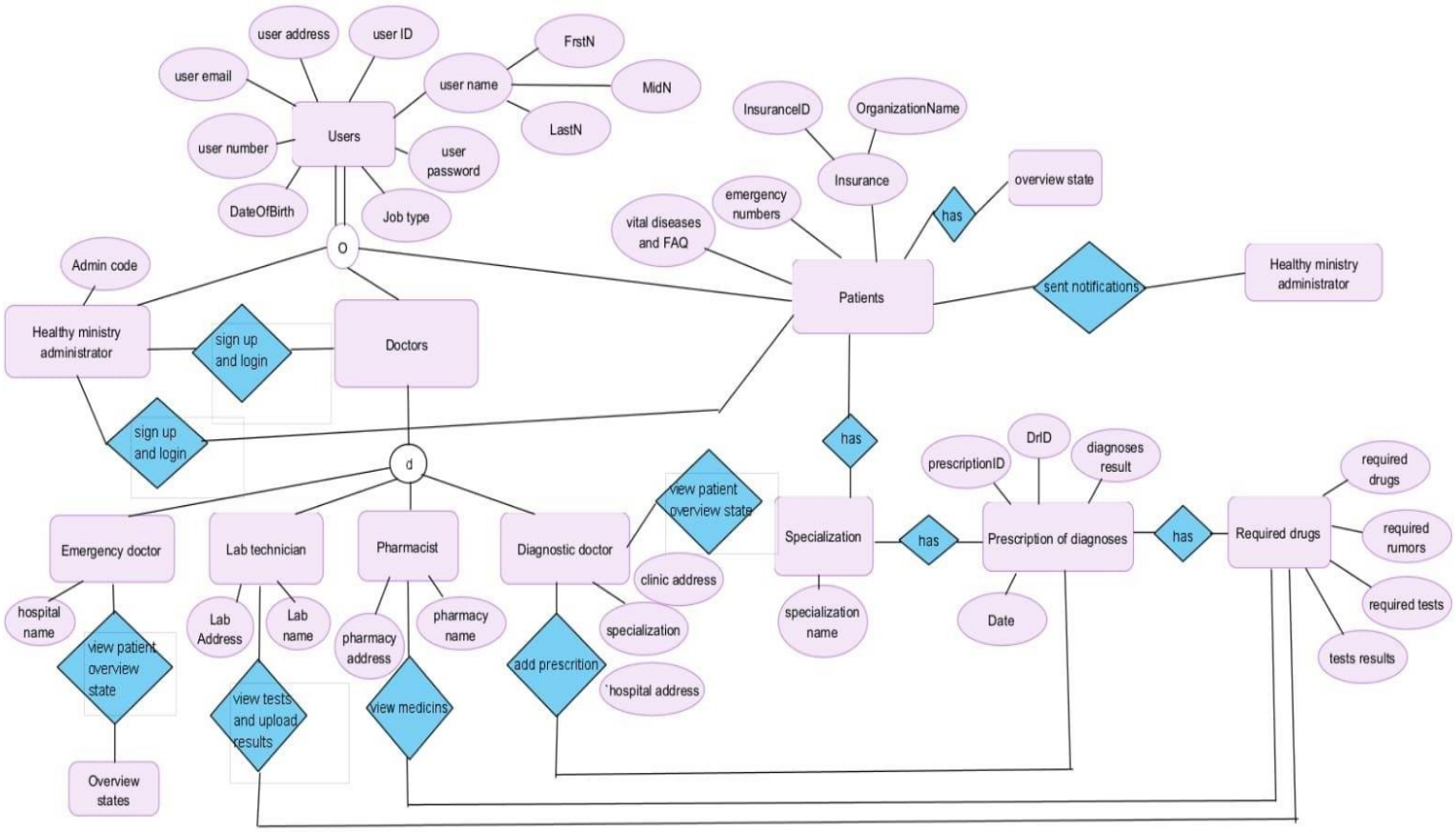


Figure 2 ERD

CHAPTER 3: ANALYSIS AND REQUIREMENT

4.3Use Case Diagram

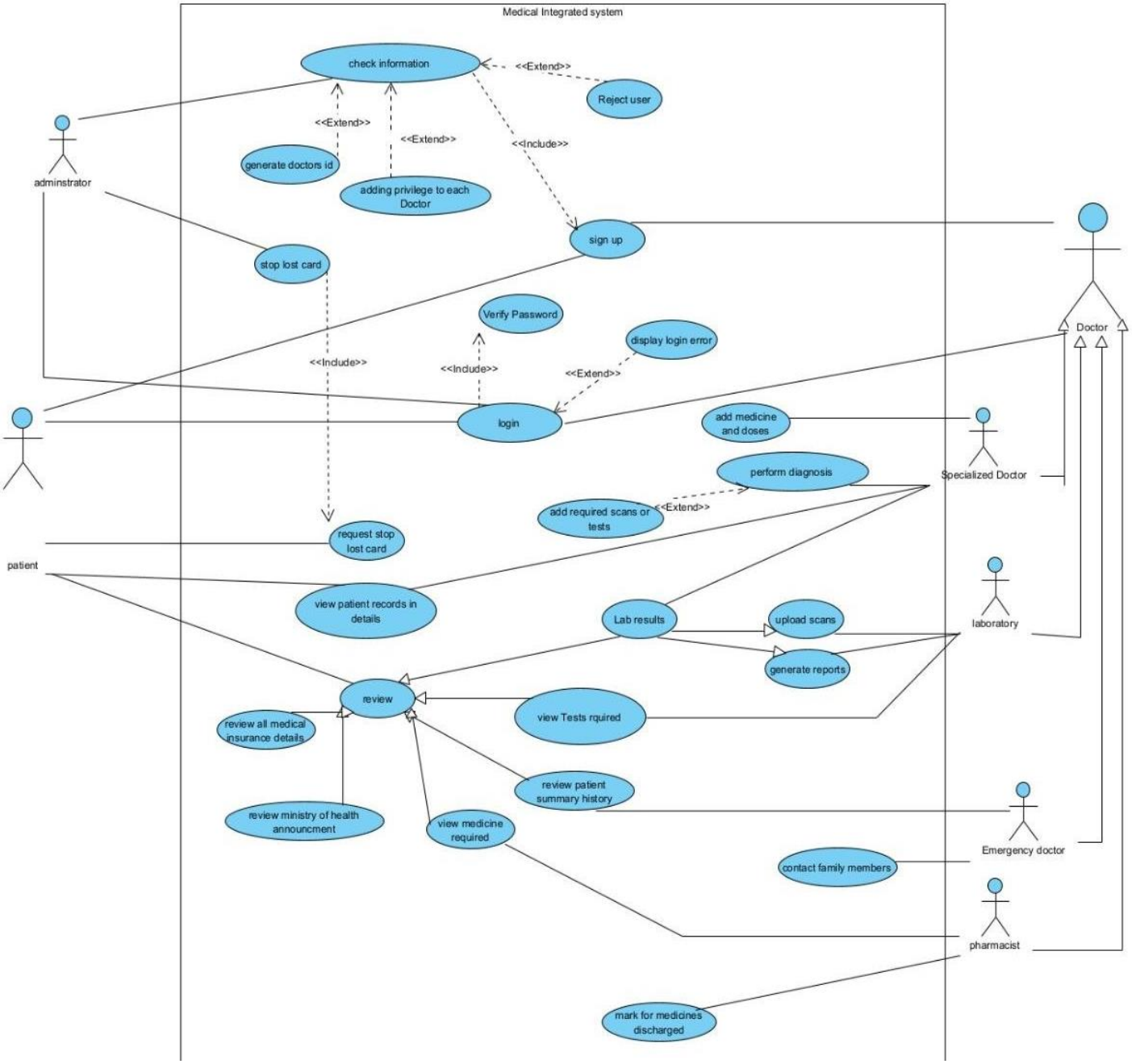


Figure 3 Use Case Diagram

CHAPTER 3: ANALYSIS AND REQUIREMENT

4.4 Sequence Diagram

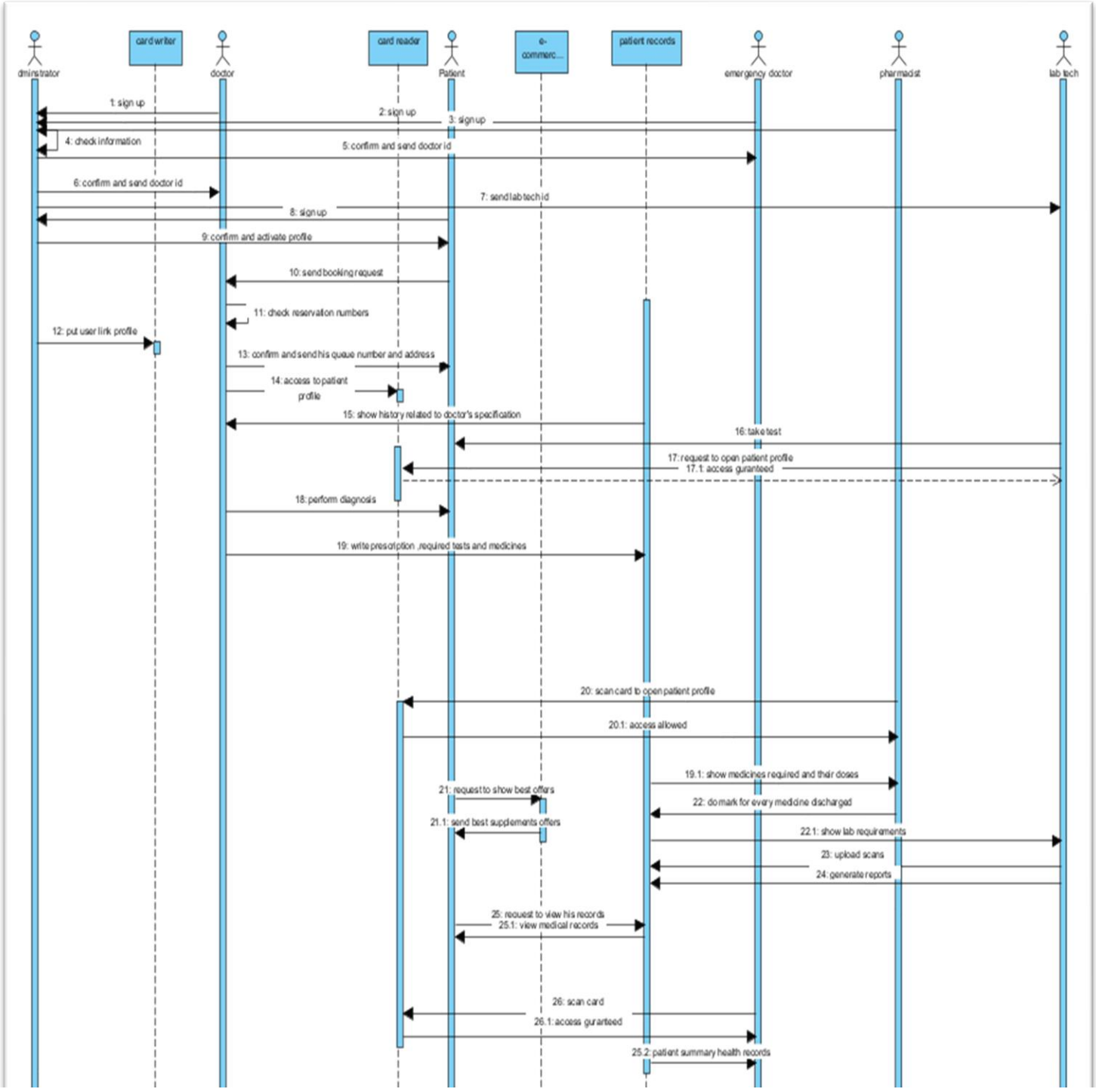


Figure 4 sequence diagram

4.5 Data Flow Diagram

1. Context Diagram

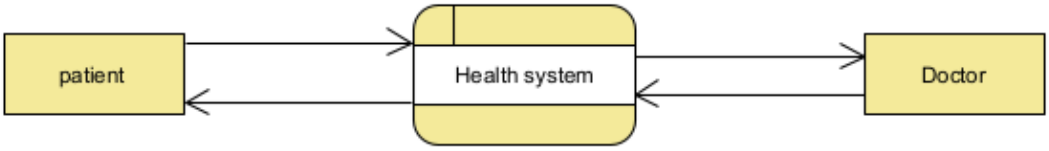


Figure 5 DFD Context level

2. Second level diagram

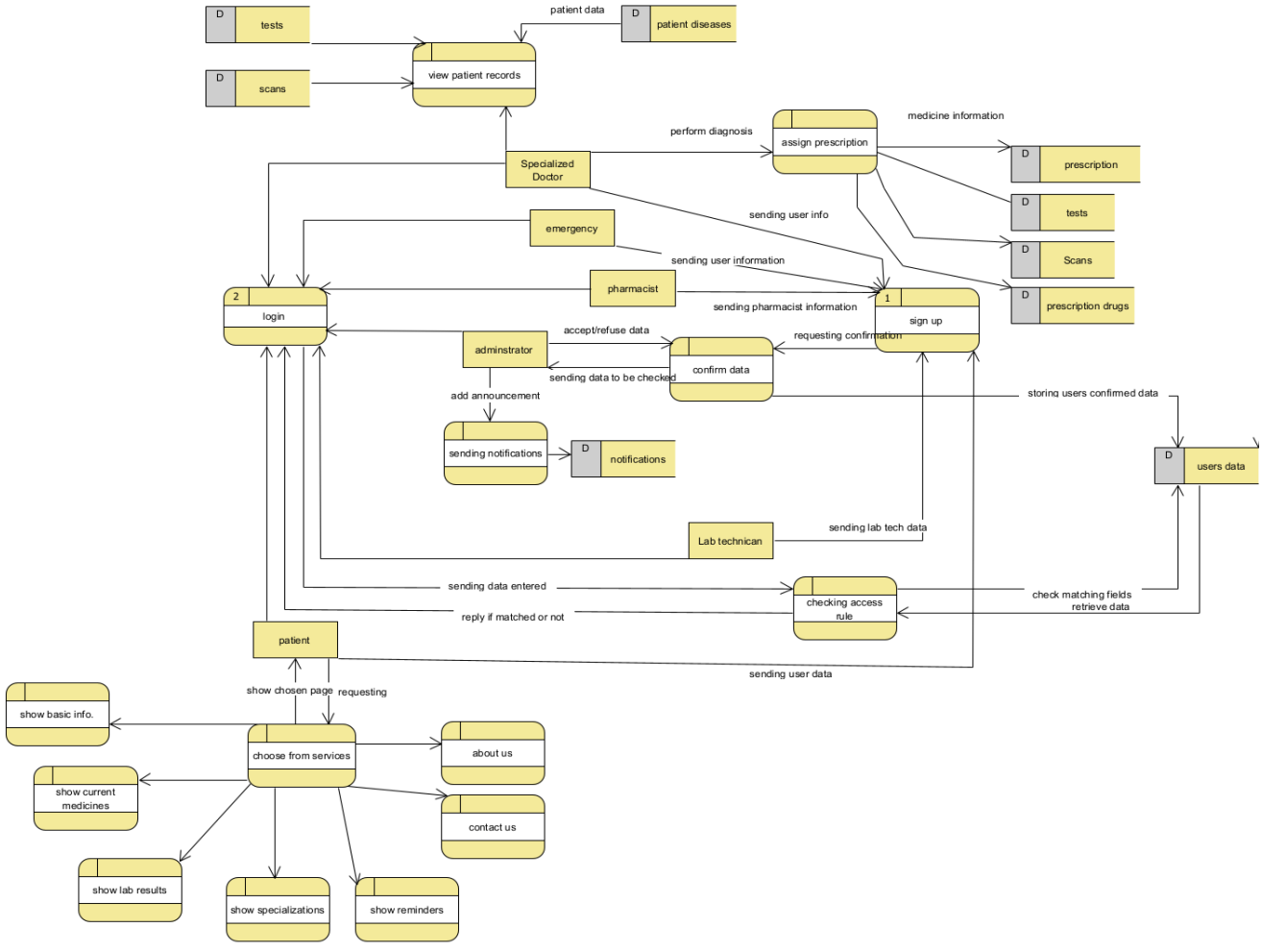


Figure 6 DFD second level

Second level DFD Continued:

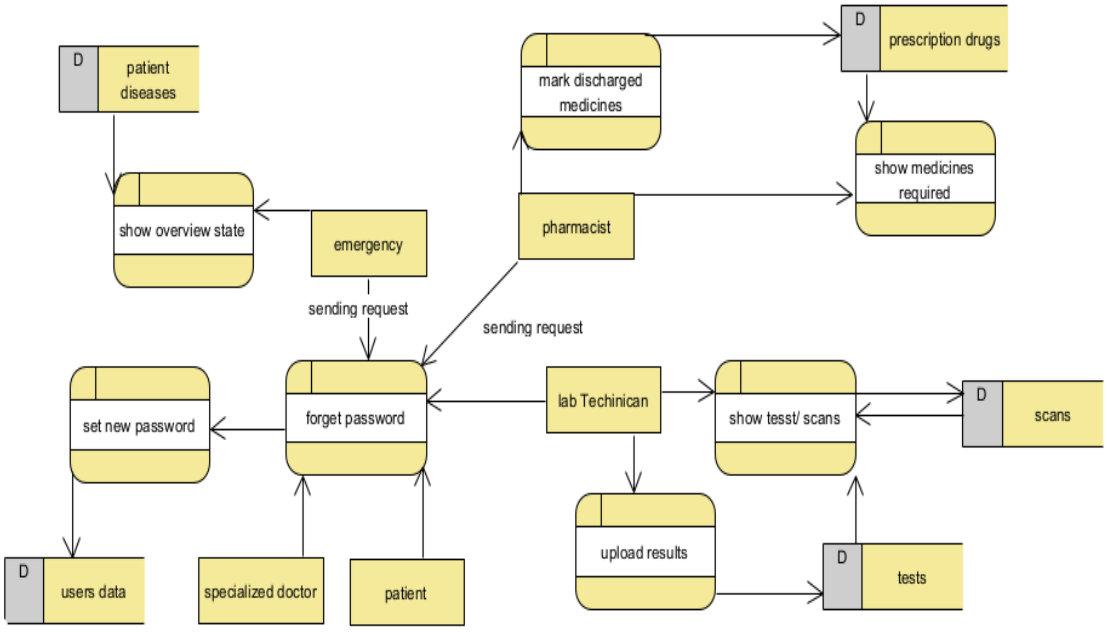


Figure 7 DFD Second level

CHAPTER 4: IMPLEMENTATION

Chapter 4: Implementation

1. FRONT-END

1.1 Angular Framework

Two essential technologies for creating web pages are HTML and CSS. While CSS takes care of the page's visual and auditory layout, HTML supplies the page's framework. Building on JavaScript, TypeScript is a tightly typed programming language.

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. It is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications. It's good for building excellent, adaptable, and seamless web, mobile, or desktop apps, to sum up. It provides the tools necessary to create complete applications, going beyond simply being a framework. Solutions including Single Page Apps (SPAs), Progressive Web Apps (PWAs), expansive business software products, cross-platform mobile solutions, and more may be designed with Angular. [8]

The architecture of an Angular application relies on certain fundamental concepts. The basic building blocks of the Angular framework are Angular components that are organized into Ng-Modules. Ng-Modules collect related code into functional sets; an Angular application is defined by a set of Ng Modules. An application always has at least a root module that enables bootstrapping, and typically has many more feature modules.

- Components define views, which are sets of screen elements that Angular can choose among and modify according to your program logic and data.

- Components use services, which provide specific functionality not directly related to views. Service providers can be injected into components as dependencies, making your code modular, reusable, and efficient.

1.1.1 Modules

Angular *Ng-Modules* differ from and complement JavaScript (ES2015) modules. An Ng-Module declares a compilation context for a set of components that is dedicated to an application domain, a workflow, or a closely related set of capabilities. An Ng-Module can associate its components with related code, such as services, to form functional units.

Every Angular application has a *root module*, conventionally named App-Module, which provides the bootstrap mechanism that launches the application. An application typically contains many functional modules.

Like JavaScript modules, Ng-Modules can import functionality from other Ng-Modules and allow their own functionality to be exported and used by other Ng-Modules. For example, to use the router service in your app, you import the Router Ng-Module.

1.1.2 Components

Every Angular application has at least one component, the *root component* that connects a component hierarchy with the page document object model (DOM). Each component defines a class that contains application data and logic and is associated with an HTML *template* that defines a view to be displayed in a target environment.

The @Component() decorator identifies the class immediately below it as a component and provides the template and related component-specific metadata.

1.1.3 Routing

The Angular Router Ng-Module provides a service that lets you define a navigation path among the different application states and view hierarchies in your application. It is modeled on the familiar browser navigation conventions:

- Enter a URL in the address bar and the browser navigates to a corresponding page.
- Click links on the page and the browser navigates to a new page
- Click the browser's back and forward buttons and the browser navigates backward and forward through the history of pages you've seen.

The router maps URL-like paths to views instead of pages. When a user performs an action, such as clicking a link, that would load a new page in the browser, the router intercepts the browser's behavior, and shows or hides view hierarchies.

CHAPTER 4: IMPLEMENTATION

2. BACK-END

2.1 Spring Boot Framework

Spring Boot is an open source Java-based framework used to create a micro Service. It is developed by Pivotal Team and is used to build stand-alone and production ready spring applications, widely used in the development of Java-based web applications due to its numerous advantages and features.

One of the key benefits of Spring Boot is its ability to reduce the amount of boilerplate code required to set up a web application. It provides a streamlined and opinionated approach to application development, allowing developers to focus on writing business logic rather than dealing with the complexities of configuring and setting up the application infrastructure.

Spring Boot also comes with built-in support for embedded servers, such as Tomcat, Jetty, and Undertow. This means that developers can run their applications as standalone executables without the need for deploying them on external servers. This simplifies the deployment process and makes it easier to package and distribute the application.

Also, it has its auto-configuration feature. It automatically configures various components and dependencies based on the class path and application properties. This eliminates the need for manual configuration and reduces the chances of configuration errors. Additionally, Spring Boot provides a wide range of starter dependencies that automatically configure common libraries and frameworks, such as Spring Data, Spring Security, and Spring MVC. This simplifies the integration of these components into the application.

Spring Boot also offers production-ready features out of the box. It provides metrics, health checks, and other monitoring capabilities that help developers monitor the health and performance of their applications.

Furthermore, Spring Boot has a vibrant and active community that regularly releases updates and new features, ensuring that developers have access to the latest advancements in web application development.[9]

Advantages

Spring Boot offers the following advantages to its developers −

- Easy to understand and develop spring applications.
- Increases productivity.
- Reduces the development time.

Goals

Spring Boot is designed with the following goals −

- To avoid complex XML configuration in Spring
- To develop a production ready Spring applications in an easier way
- To reduce the development time and run the application independently.
- Offer an easier way of getting started with the application.

CHAPTER 4: IMPLEMENTATION

Why Spring Boot?

You can choose Spring Boot because of the features and benefits it offers:

- It provides a flexible way to configure Java Beans, XML configurations, and Database Transactions.
- It provides powerful batch processing and manages REST endpoints.
- In Spring Boot, everything is auto configured; no manual configurations are needed.
- It offers annotation-based spring application.
- Eases dependency management.
- It includes an Embedded Servlet Container.

Chapter 5: Database

Database management is an essential aspect of web application development, and MySQL is one of the most widely used relational database management systems. In our project, we utilized MySQL in conjunction with Spring Boot and Spring Data to enhance the functionality and efficiency of our Spring web application.

One of the key advantages of using MySQL with Spring Boot is its seamless integration with the Spring Data framework. Spring Data provides a high-level abstraction layer that simplifies the interaction with databases, including MySQL. It eliminates the need for writing complex SQL queries and instead allows developers to work with Java objects and annotations. This significantly reduces the amount of boilerplate code required for database operations, making the application more maintainable and easier to understand.

Another advantage of using MySQL with Spring Boot is its performance and scalability. MySQL is known for its robustness and ability to handle large amounts of data efficiently. With Spring Boot's support for connection pooling and caching, our application can effectively manage database connections and optimize query execution. This ensures that our application can handle a high volume of concurrent requests without compromising performance.

Additionally, MySQL offers various features that enhance data integrity and security. It supports transactions, which allow multiple database operations to be grouped together as a single unit of work. This ensures that the data remains consistent even in the event of failures or errors. MySQL also provides built-in security mechanisms, such as user authentication and access control, to protect sensitive data from unauthorized access.[10]

CHAPTER 5: DATABASE

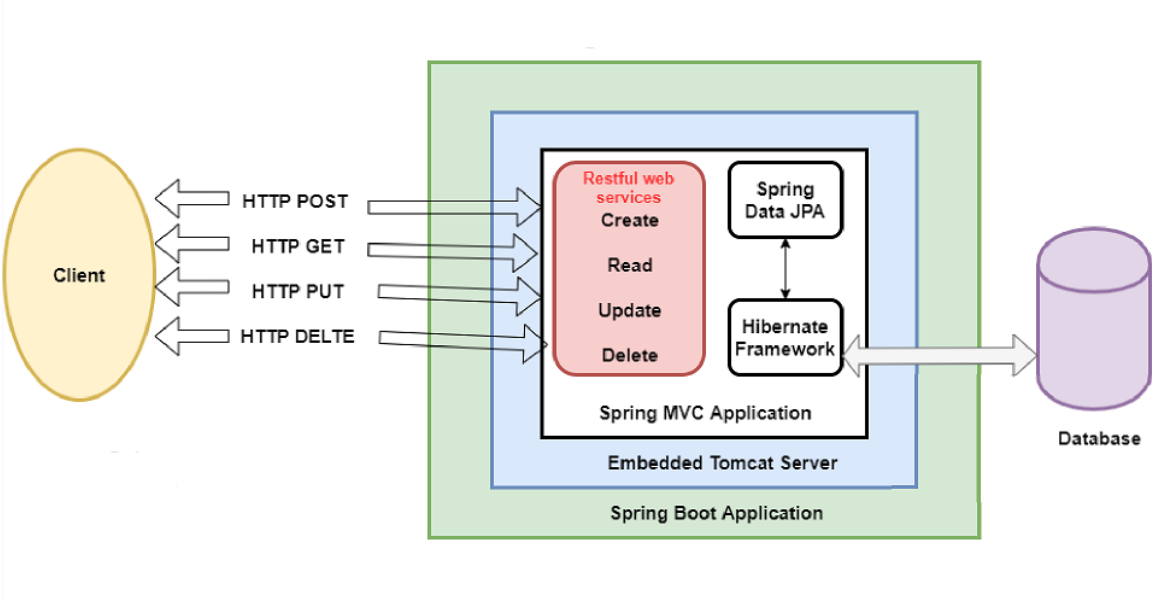


Figure 8 Spring Data

To facilitate the development and testing process, we used XAMPP as a local development environment. XAMPP is a software package that includes Apache web server, MySQL database, and PHP scripting language. It provides a convenient way to set up a local server environment without the need for manual configuration. With XAMPP, we were able to quickly deploy and test our Spring Boot application locally before deploying it to a production environment.[11]

Figure 9 XAMPP

CHAPTER 5: DATABASE

RELATIONAL DATABASE USING XAMPP

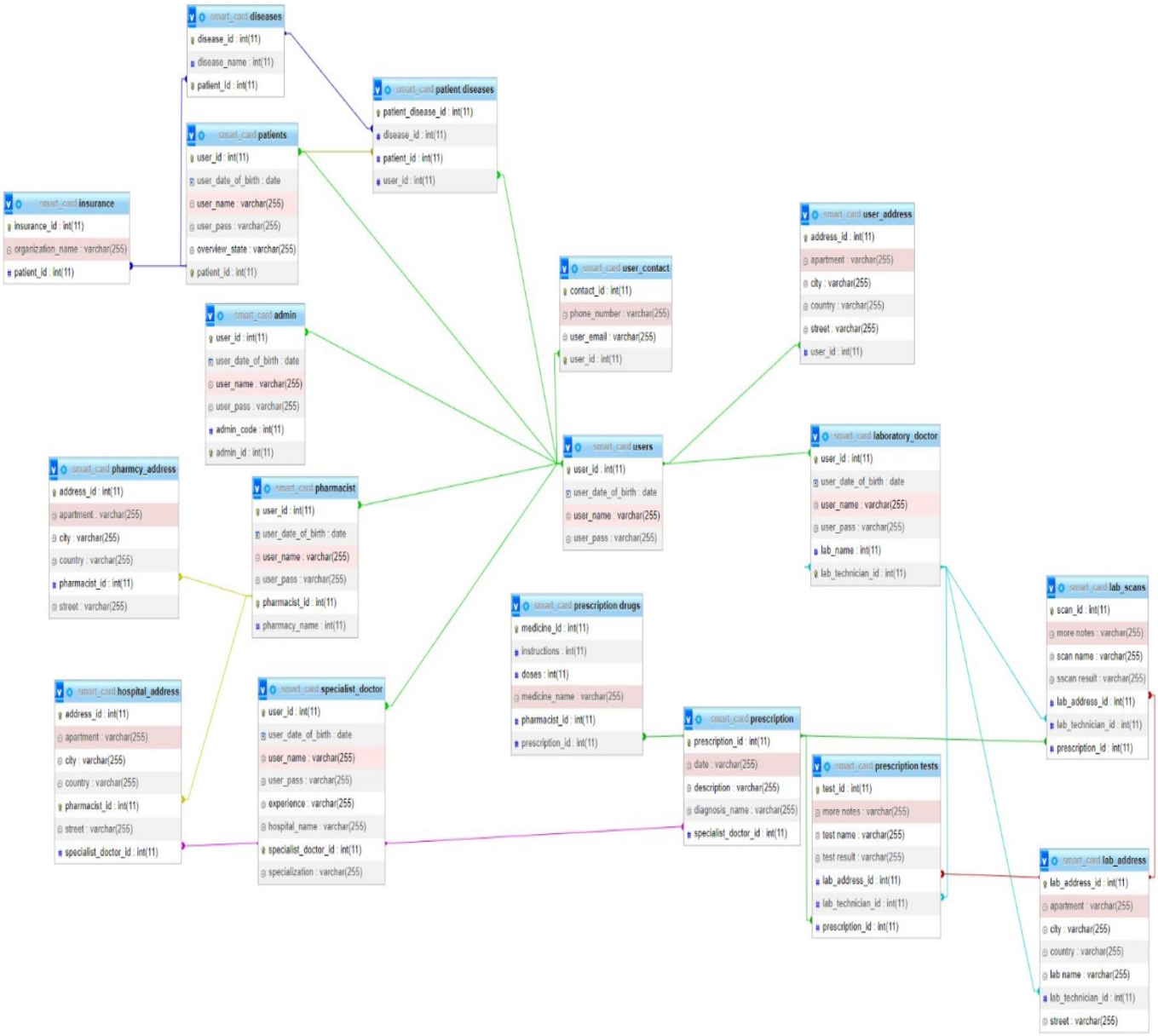


Figure 10 Relational DB

CHAPTER 5: DATABASE

Generating tables using Spring data, for example on Prescription entity.

```java
@Entity
@Table(name = "Prescription")
public class PrescriptionEntity {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "prescription_id")
    private Integer PrescriptionId;

    @Column(name = "diagnosis_Name")
    private String diagnosisName;

    @Column(name = "Date")
    private String date;

    @Column(name = "description")
    private String description;

    1 usage
    @JsonBackReference
    @OneToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "specialist_doctor_id", insertable = false, updatable = false)
    private SpecialistDoctor specialistDoctor;

    @JsonManagedReference
    @OneToMany(mappedBy = "prescriptionEntities")
    private List<RequiredMedication> requiredDrugs;

    @JsonManagedReference
    @OneToMany(mappedBy = "prescriptionEntity")
    private List<LabTests> labTests;

    @JsonManagedReference
    @OneToMany(mappedBy = "prescriptionEntity")
    private List<LabScans> labScans;
}
```

Figure 11 snippet from Prescription class

Chapter 6: Security

Spring Security is a powerful framework that offers authentication and authorization functionalities. It is constructed on top of the Spring framework and seamlessly incorporates Spring Boot. A complete framework for adding authentication and authorization to web applications is called Spring Security. It offers customizable options for managing passwords, user authentication, and authorization. We can use its features to secure our web application and distinguish between different users based on their roles and permissions by integrating it with Spring Boot.

2.2 Authentication and Authorization

- Authentication is the process of verifying the identity of a user.

- Authorization determines what actions a user is allowed to perform within an application. Spring Security helps in implementing both these functionalities in a web application.

1. Configure Spring Security: We need to add the necessary dependencies to our project's build file and configure Spring Security in the application's configuration file. This includes specifying the login and logout URLs, defining access rules for different URLs, and configuring authentication providers.

2. User Authentication: Spring Security supports various authentication mechanisms such as form-based authentication, HTTP basic authentication, and OAuth. For a typical web application, we can use form-based authentication, where users provide their credentials

(username and password) through a login form. Spring Security handles the validation of these credentials and authenticates the user.[13]

3. User Authorization: Once a user is authenticated, we can define access rules to restrict certain URLs or resources based on the user's roles or permissions. Spring Security provides annotations and configuration options to define these access rules. For example, we can specify that only users with the "ADMIN" role can access certain URLs.

4. User Management: Spring Security integrates with various user management systems such as databases, LDAP servers, and custom user repositories. We can configure Spring Security to authenticate users against these systems and retrieve their roles and permissions. Additionally, we can implement features like login and sign up by creating appropriate controllers and views.

5. Password Management: Spring Security provides utilities for secure password storage and management. It supports features like password hashing, salting, and password strength validation. We can implement a "forget password" functionality by allowing users to reset their passwords through a secure process.

Chapter 7: Testing and Validation

Software testing is an essential part of the software development life cycle. It involves the process of evaluating a system or application to ensure that it meets the specified requirements and functions as expected. Testing helps identify defects, errors, or gaps in the software, allowing developers to fix them before the product is released to users.

There are various forms of software testing that we've used to test our project, such as:

1.UNIT TESTING:

This type of testing focuses on checking that individual parts or units of code operate as intended. It assists in early bug detection and is typically carried out by developers themselves.[12]

benefits:

- JUnit can help you keep your code organized and easy to read.
- JUnit can help you detect and fix errors in your code.
- JUnit can help you improve the quality of your software.
- JUnit can help you work more efficiently and improve your testing process.

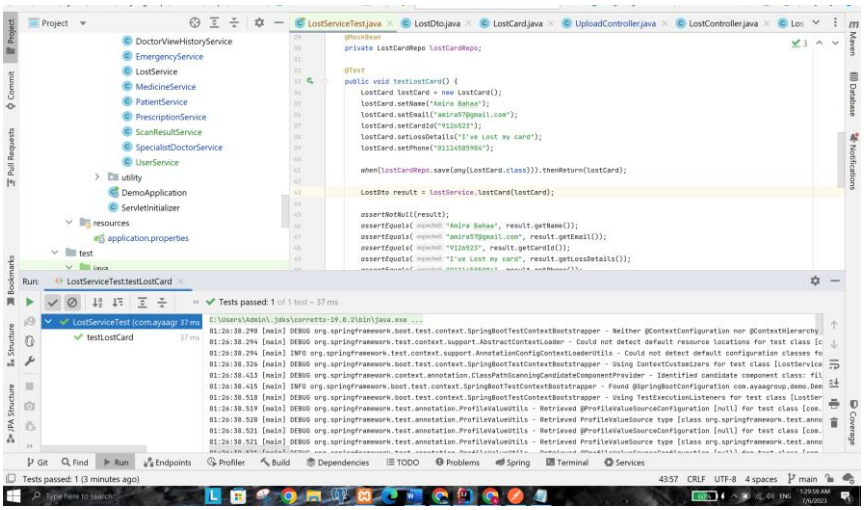


Figure 12 Unit testing on Lost card Functions

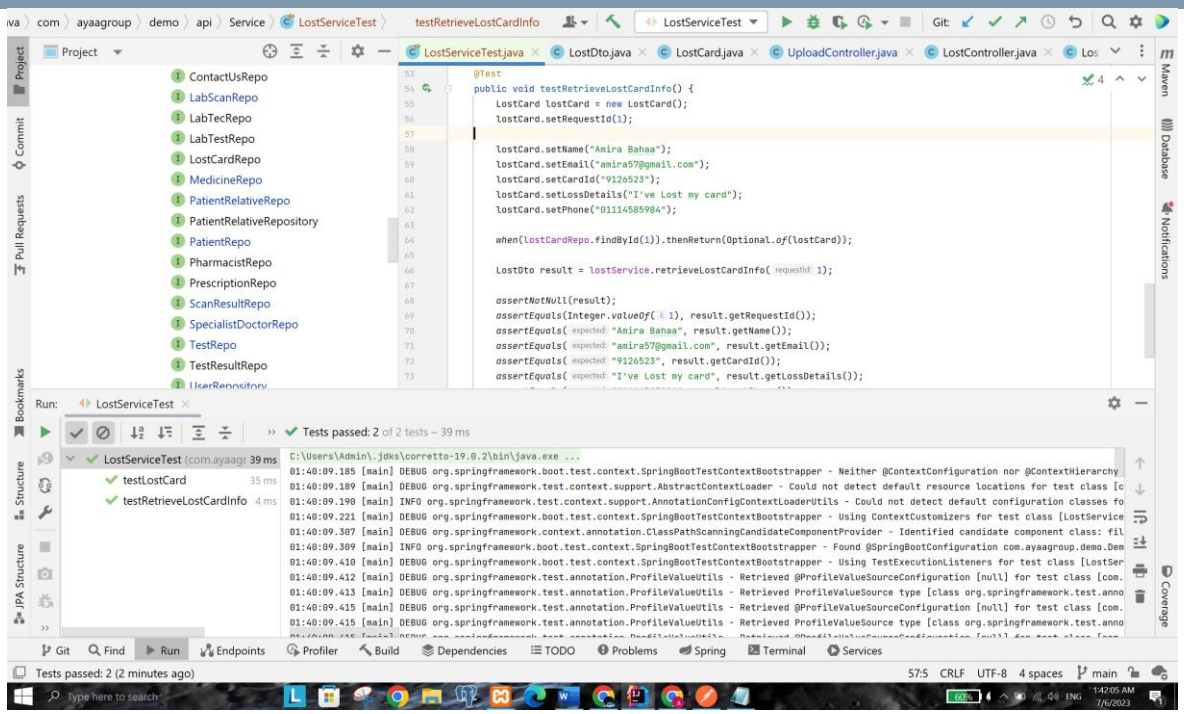


Figure 13 additional tests on Lost card functions

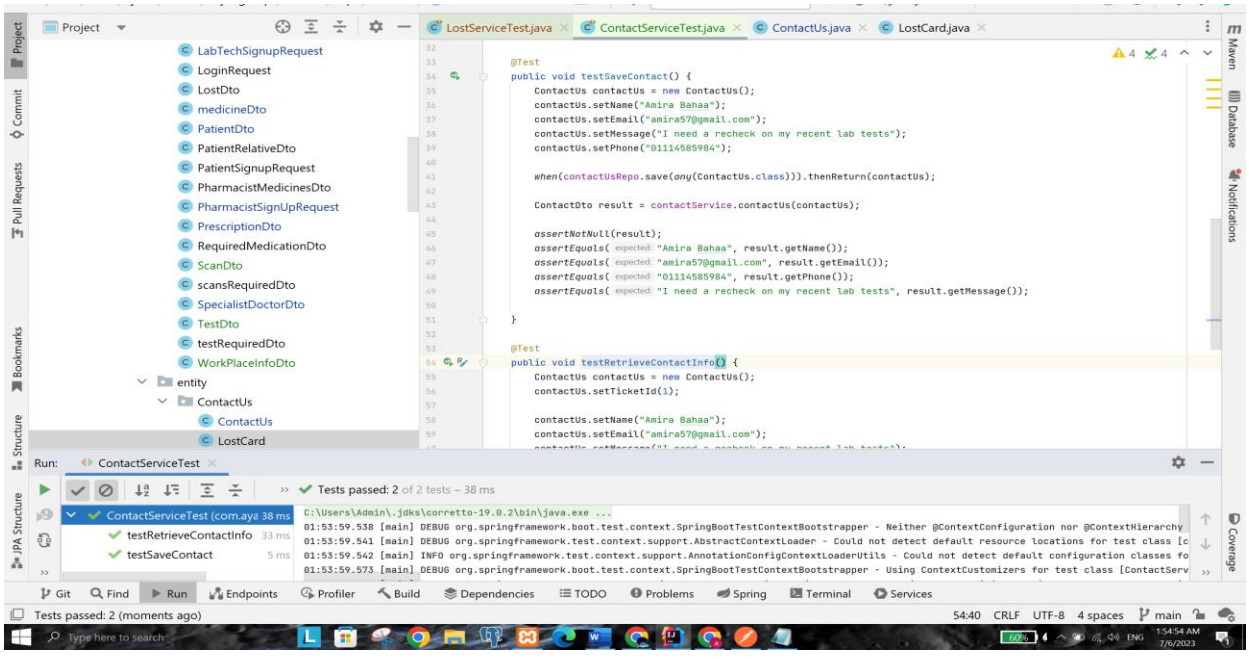


Figure 14 Contact us functions Tests passed.

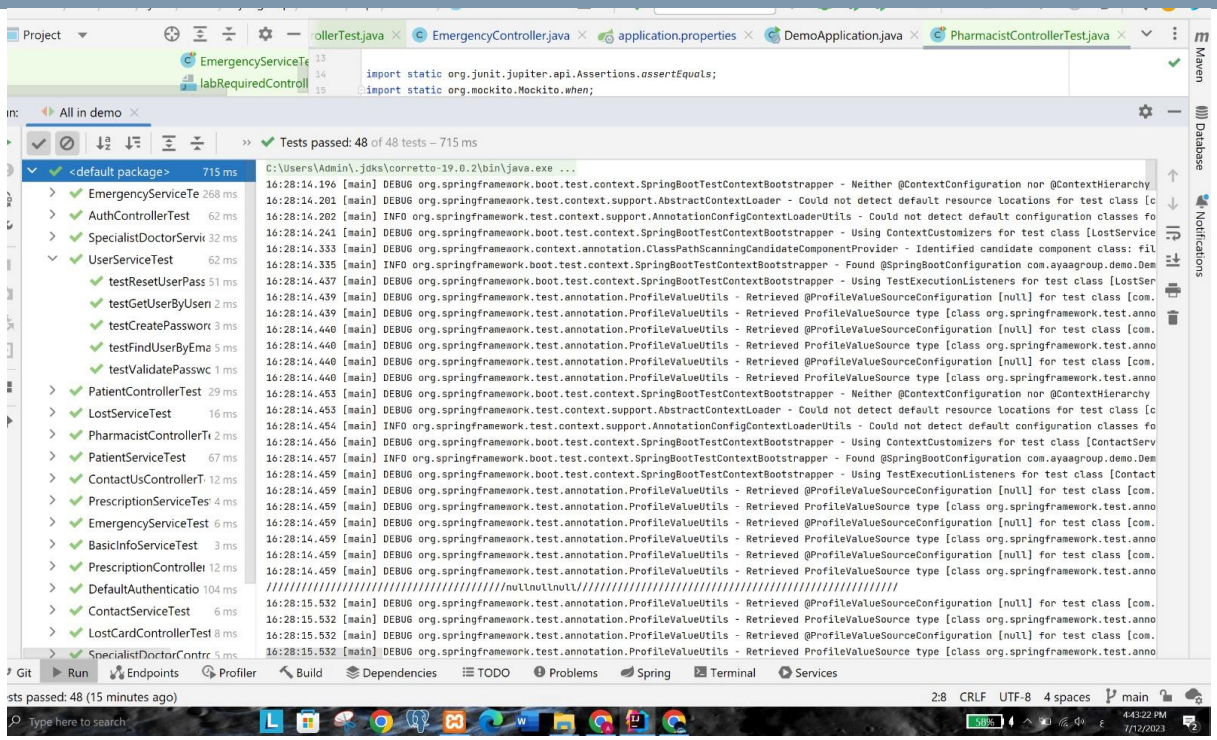


Figure 15 Rest of functions Tests passed.

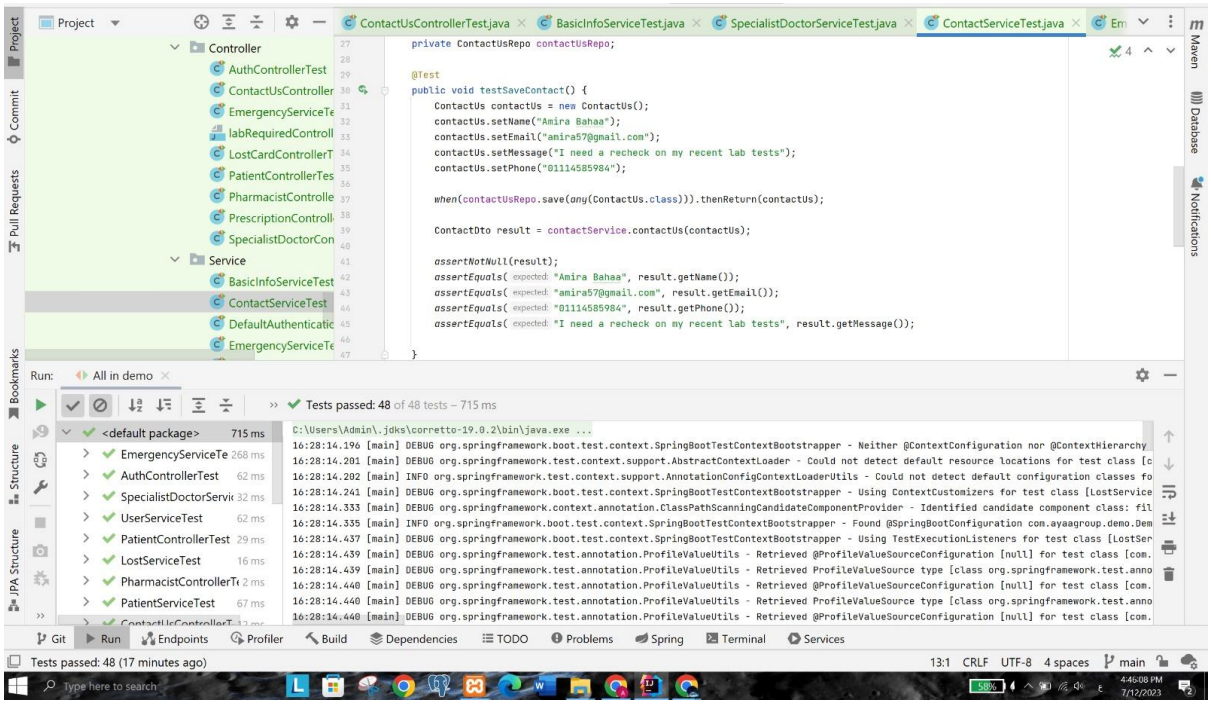


Figure 16 Service functions Tests passed

CHAPTER 7: TESTING AND VALIDATION

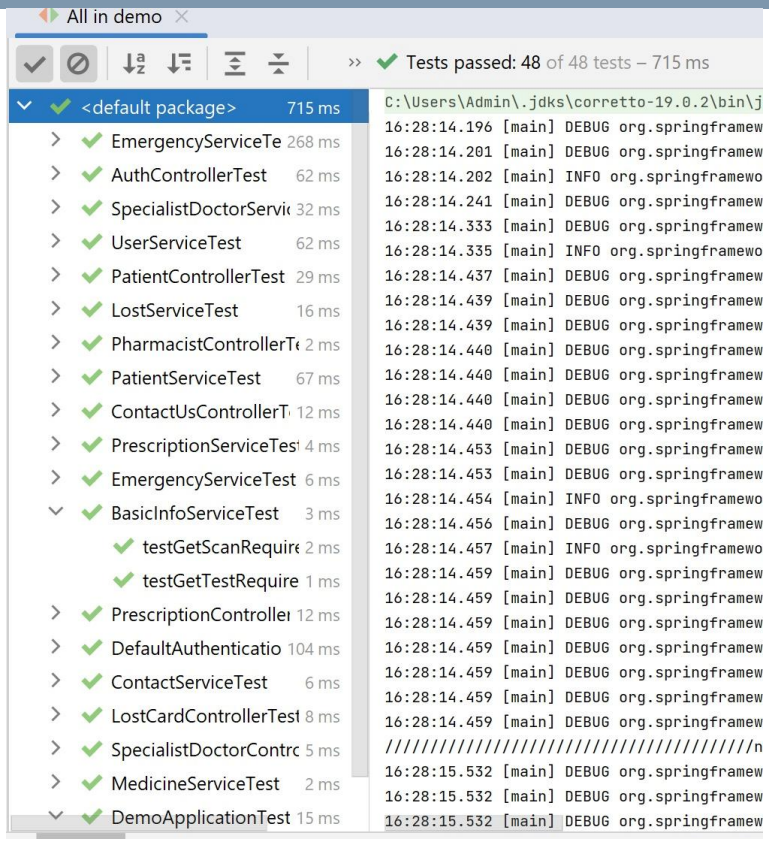


Figure 17 the running output of the unit tests

API TESTING AND TESTCASES:

The code sets up the following test cases:

- one for asserting the status code,
- one for validating the response body data.
- one for checking the response headers.
- The assertions ensure that the expected values match the actual values in the response.

POSTMAN:

- Postman is a standalone software testing API (Application Programming Interface) platform to build, test, design, modify, and document APIs. It is a simple Graphic User Interface for sending and viewing HTTP requests and responses.

- While using Postman, for testing purposes, one doesn't need to write any HTTP client network code. Instead, we build test suites called collections and let Postman interact with the API.
- In this tool, nearly any functionality that any developer may need is embedded. This tool can make various types of HTTP requests like GET, POST, PUT, PATCH, and convert the API to code for languages like JavaScript and Python.

Why we're using Postman?

Postman is based on a wide range of extremely user-friendly power tools. For more than 8 million users, Postman has become a tool of convenience. Following are the reasons why Postman is used:

1. **Accessibility-** One can use it anywhere after installing Postman into the device by simply logging in to the account.
2. **Use Collections**-Postman allows users to build collections for their API-calls. Every set can create multiple requests and subfolders. It will help to organize the test suits.
3. **Test development-** To test checkpoints, verification of successful HTTP response status shall be added to every API- calls.
4. **Automation Testing-**Tests can be performed in several repetitions or iterations by using the Collection Runner or Newman, which saves time for repeated tests.
5. **Creating Environments-** The design of multiple environments results in less replication of tests as one can use the same collection but for a different setting.
6. **Debugging-** To effectively debug the tests, the postman console helps to track what data is being retrieved.

7. **Collaboration-** You can import or export collections and environments to enhance the sharing of files. You may also use a direct connection to share the collections.

SYSTEM TESTING:

System testing consists of comprehensive testing of the entire software system. It confirms that the system satisfies the set requirements and that all components are correctly integrated.[12]

WRITING TEST CASES ON POSTMAN:

1. **pm.test("Status code is 200", function ())** This line introduces a new test case with the statement
2. **"Status code is 200."** Postman offers a function called **pm.test** that lets you specify a test case. The test function, where the assertions are written, is the one that is given as the second argument.
3. **pm.response.to.have.status(200);** This line states that the status code for the response is 200.
4. **pm.response** refers to the request's response object.
5. The Postman **assertion.to.have.status(200)** determines whether the response status code corresponds to the expected value.

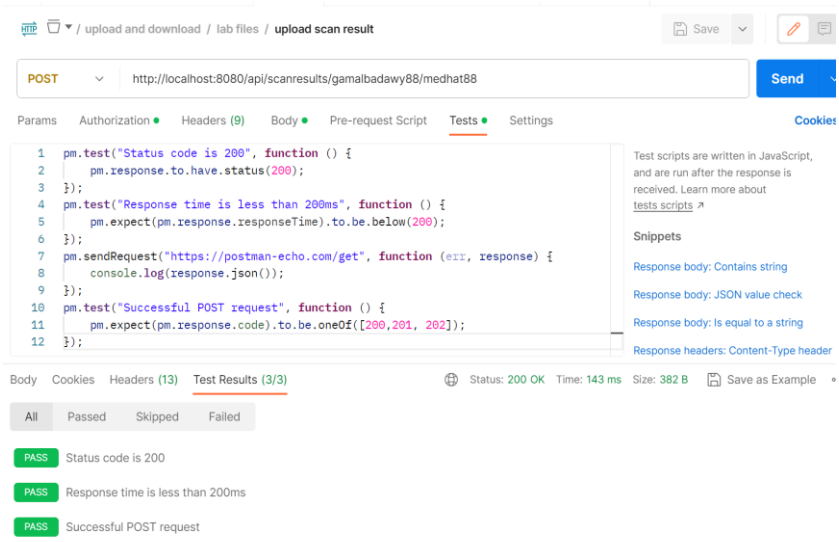


Figure 18 Test cases on postman for our application

2. API TESTING:

1. Testing functions upload and download using postman , where the username is omarrezk77, and got response with 200 ok indicates that the profile image is successfully uploaded.
   The long blob type in database means, Binary large object is basically a collection of binary data stored in the form of a single entity in the database.

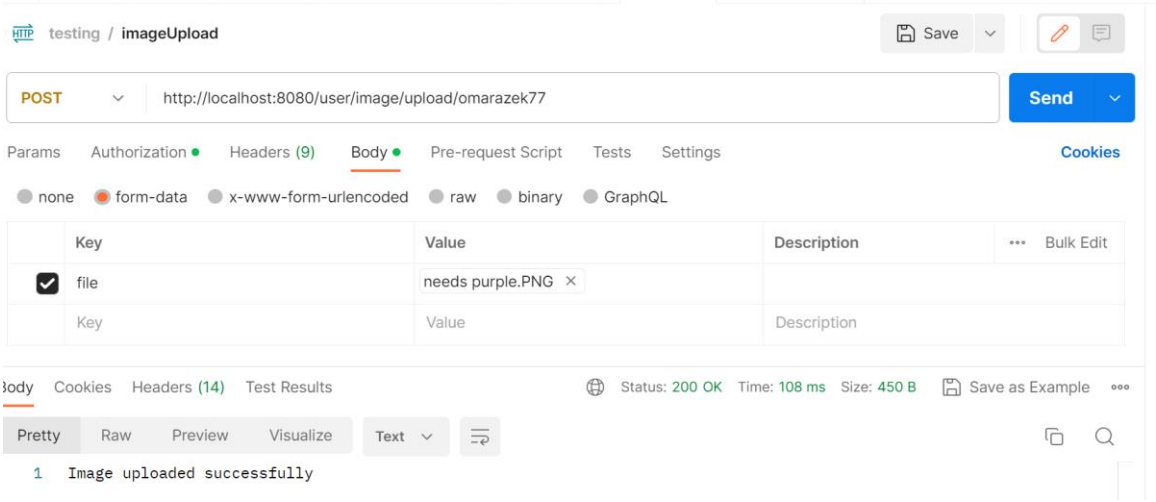


Figure 20 Api testing on profile image upload

```java
@RestController
@RequestMapping("/user")
@CrossOrigin(origins = "http://localhost:4200")
public class ImageController {
    //image upload
    @Autowired
    private UserService userService;

    @PostMapping("/image/upload/{username}")
    public ResponseEntity<String> uploadImage(@PathVariable String username, @RequestParam("file") MultipartFile file) {
        try {
            User user = userService.getUserByUsername(username);
            user.setImage(file.getBytes());
            userService.saveUser(user);
            return ResponseEntity.ok().body("Image uploaded successfully");
        } catch (IOException e) {
            return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("Failed to upload image");
        }
    }
```

Figure 19 controller code of the upload image

2. Testing the download image API that returns the profile picture of the user.

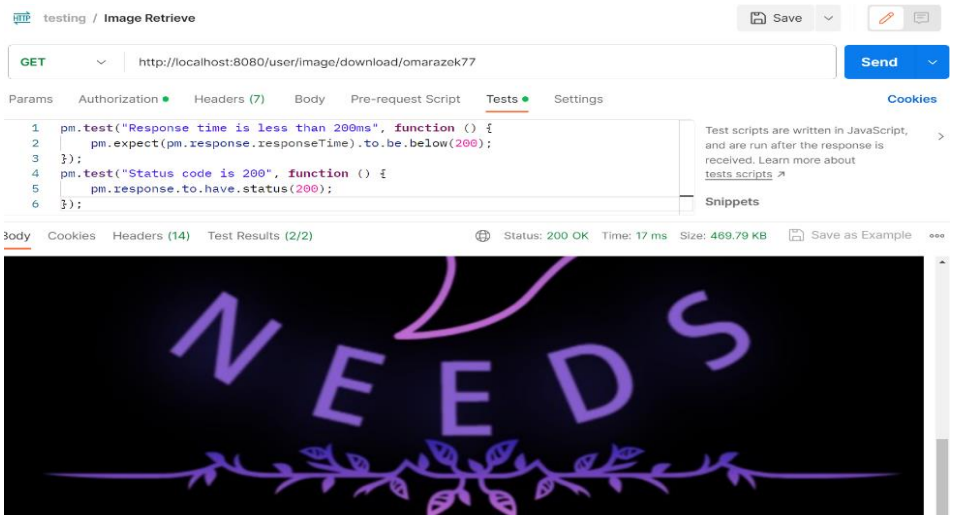


Figure 21 api test passed on image download

```
@GetMapping("/image/download/{username}")
public ResponseEntity<byte[]> getImage(@PathVariable String username) {
    User user = userService.getUserByUsername(username);
    if (user != null && user.getImage() != null) {
        HttpHeaders headers = new HttpHeaders();
        headers.setContentType(MediaType.IMAGE_JPEG);
        return new ResponseEntity<>(user.getImage(), headers, HttpStatus.OK);
    } else {
        return ResponseEntity.notFound().build();
    }
}
```

Figure 22 controller code for the image download

3. Testing to add prescription, from the specialist doctor to the username of the patient, and got successfully added prescription.

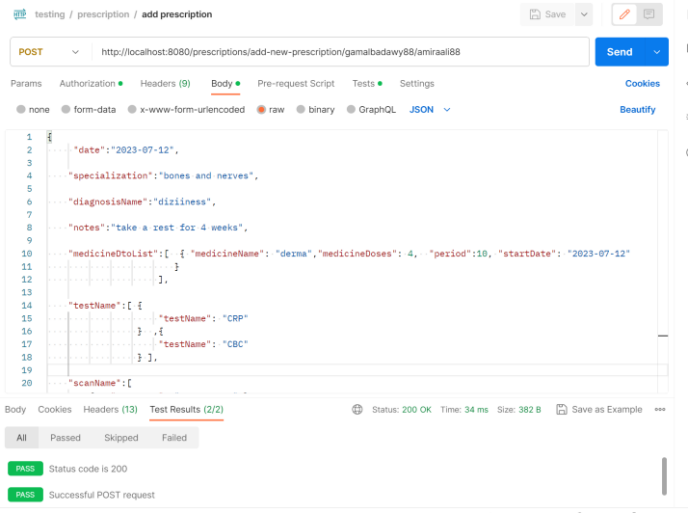


Figure 23 API testing add prescription

4. Testing to show the prescription for specific patient, and got successful responses.

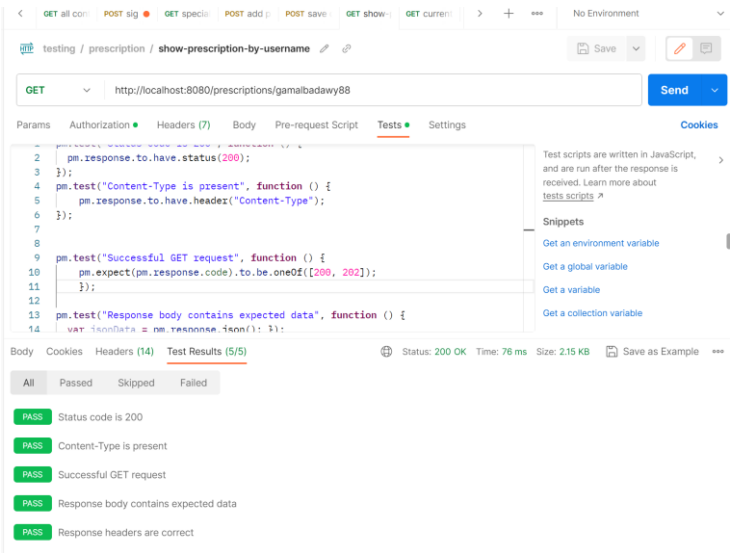


Figure 24 API testing on add prescription

5. Testing retrieving the profile data of the doctor, in this example we are testing the pharmacist profile data and passed with 17 test cases. These test cases include checking the header response, status code 200, content type is correct, response body contains the expecting data and finally checking the main information that it matches the expected results.

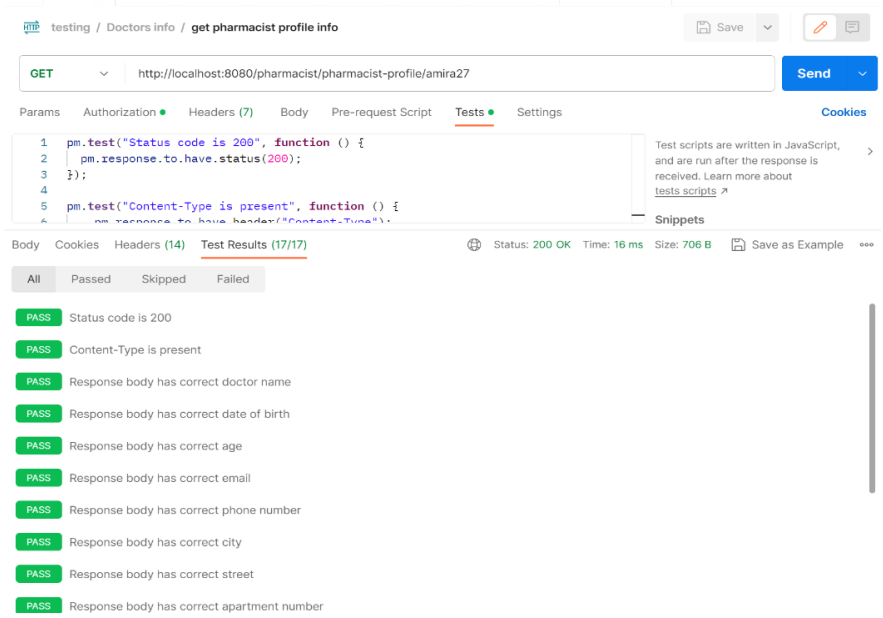


Figure 25 API 17 test cases passed on the profile page data retrieving for specific user

6. Testing the save contact us form, where the user will enter his data. The test scripts checks the header response, status code 200, content type is correct, response body contains the expecting data and test the matching input data and the saved data.

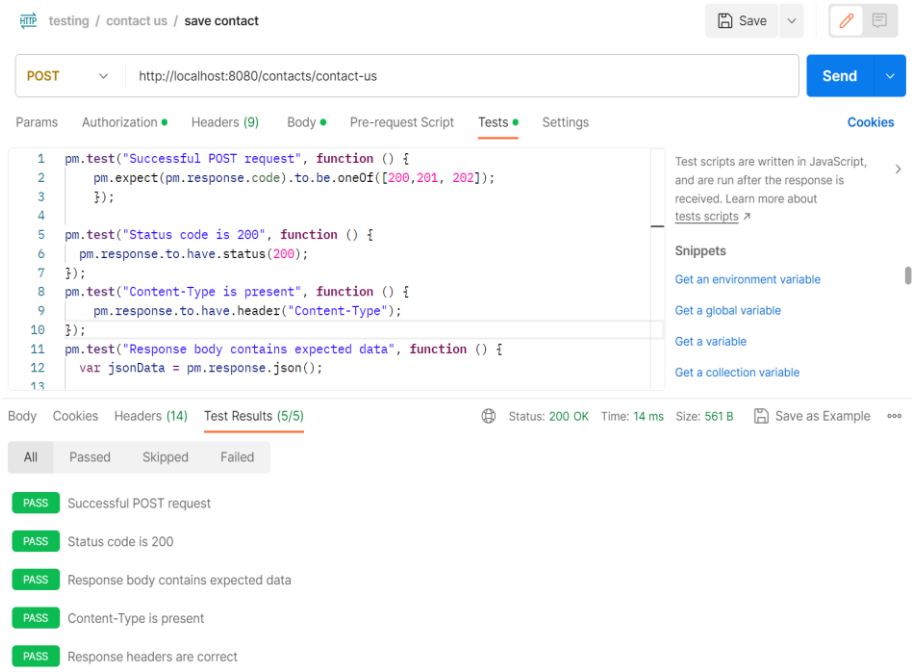


Figure 26 API testing on contact us saving form

and so on we've tested with the same techniques for the rest of the apis, here's the summary of the test results of the apis:

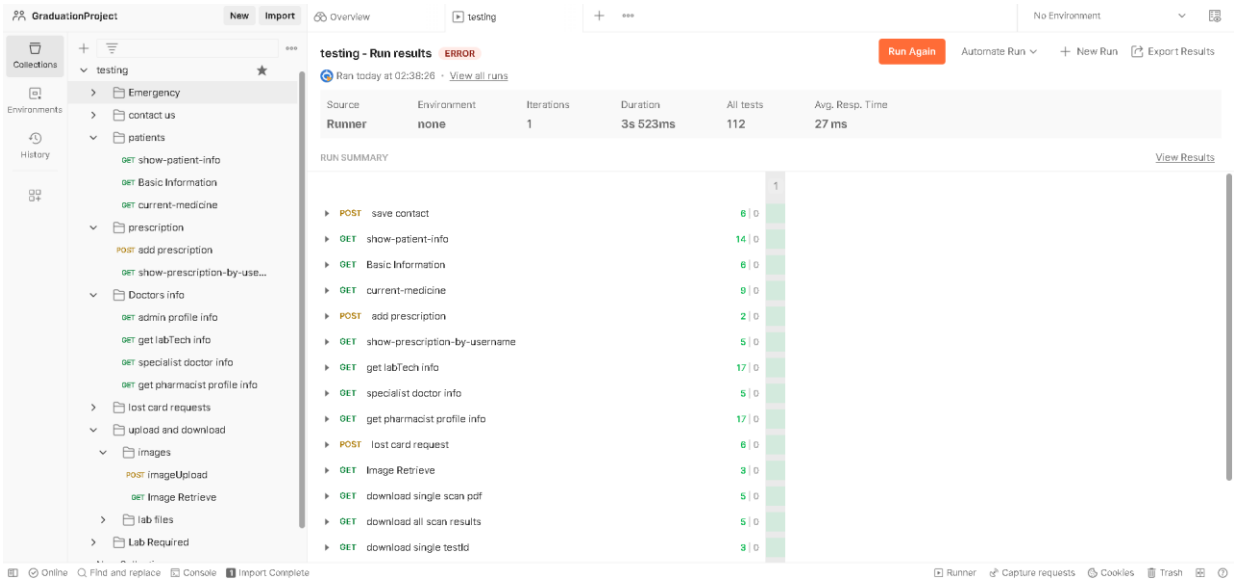


Figure 27 final results on API Testing

3.REQUIREMENTS TESTING

Register:

For register an account:

Test case ID	Test scenario	Expected Results	Actual Results	Pass/Fail
TC1	All fields are blank	"invalid data" error message should be displayed	As Expected.	**Pass**
TC2	All fields are filled with valid data	Registration done successfully	As Expected	**Pass**
TC3	Register with already existing username	"invalid data" error message should be displayed	As Expected	**Pass**
TC4	Wrong email format	"Invalid data" error message should be displayed	As Expected	**Pass**

CHAPTER 7: TESTING AND VALIDATION

Login:

For login to an account:

Test case ID	Test scenario	Expected Results	Actual Results	Pass/Fail
TC1	Email and Password are blank	"invalid data " error message should be displayed	As Expected.	**Pass**
TC2	Email is blank	"Invalid data" error message should be displayed	As Expected.	**Pass**
TC3	Password is blank	"invalid data" error message should be displayed	As Expected.	**Pass**
TC4	Wrong email format	"invalid data" error message should be displayed	As Expected.	**Pass**
TC5	Wrong email	"Invalid data" error message should be displayed	As Expected.	**Pass**
TC6	Wrong password	"Invalid data" error message should be displayed	As Expected.	**Pass**
TC7	Valid email and password	Login successfully	As Expected.	**Pass**

Chapter 8: Hardware

RFID stands for Radio Frequency Identification which was invented basically for defense and war purposes. It claims to have patented by Mario W. Cardullo for an active rewritable RFID tag on January 23, 1973. It helps in automatically tracking and identifying objects. It actually works on the principle of exchange of radio signals between the identification medium and a RFID reader with a combination of microchip and radio frequency technologies.

RFID technology here helps in tracking the books and materials, reducing management costs and increases the time the library staffs spend with patrons in accessing the library resources more effectively and efficiently. RFID technology, during the past few decades have been playing a vital role in redefining library processes in all aspects and increased users' satisfaction. Here are some of the major advantages of implementing RFID technology in Libraries:

- RFID tags replace both EM security chips and Barcode.
- Simplifies patron self-check-out and check-in.
- Radio Frequency and anti-theft detection is innovative and safe.
- Helps in verifying stock with high speed and accuracy.
- Long term development guaranteed when using open standards.

RFID technology for Libraries includes RFID tags, a self-check-out station, self-return book drops with an automatic check in feature, a tagging station, a set of security gates, and shelf scanner for stock verification and administrative reasons.[15]

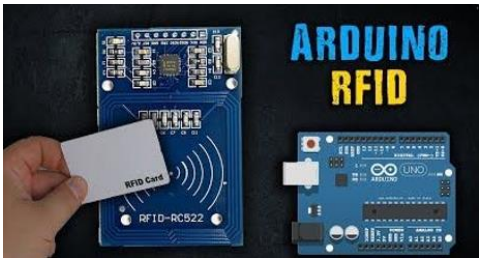


Figure 28 Arduino RFID

CHAPTER 8: HARDWARE

THE WORKING IDEA:

In a passive RFID system, the tags do not use a battery; instead, they receive their energy to run from the reader. The reader emits an energy field of a few feet, providing the energy for any tag in the vicinity. The tag gathers the electromagnetic energy from the card reader, powers up, and responds with 'hello world' and its identification information.

MAIN COMPONENTS

RFID reader (MFRC522):

- RFID reader is a device used to gather information from an RFID tag, which is used to track Library books. Radio waves are used to transfer data from the RFID tag to the reader. It is also called as n interrogator.
- The RC522 RFID module based on the MFRC522 IC from NXP usually comes with an RFID card tag and a key fob tag with 1KB of memory. And the best part is that it can write a tag that means you can store any message in it.
- The RC522 RFID reader module is designed to create a 13.56MHz electromagnetic field and communicate with RFID tags. The reader can communicate with a microcontroller over a 4-pin SPI with a maximum data rate of 10 Mbps.

Figure 29 RFID-RC522 module.

- Reader pins:

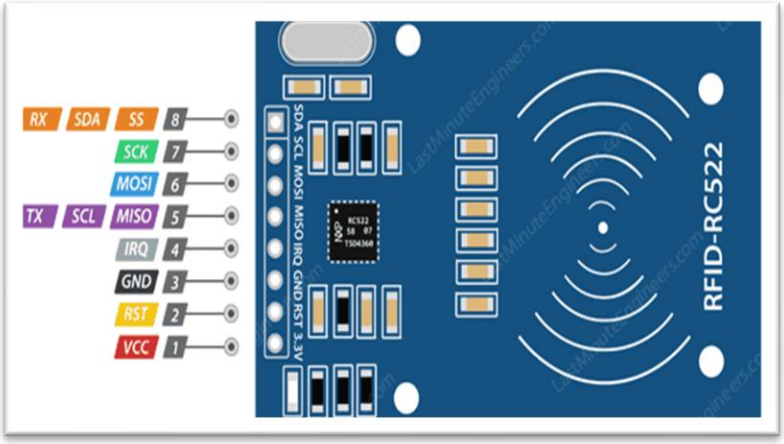


Figure 20 RC522 module pins

PINS	Definition
VCC	supplies power to the module. This can be anywhere from 2.5 to 3.3 volts. You can connect it to the 3.3V output from your Arduino.
RST	is an input for reset and power-down. When this pin goes low the module enters power-down mode. In which the oscillator is turned off and the input pins are disconnected from the outside world. Whereas the module is reset on the rising edge of the signal.
GND	is the ground pin and needs to be connected to the GND pin
MISO / SCL / Tx	pin acts as master-in-slave-out when SPI interface is enabled, as serial clock when I2C interface is enabled and as

	serial data output when the UART interface is enabled.
MOSI (Master Out Slave In)	is the SPI input to the RC522 module.
SCK (Serial Clock)	accepts the clock pulses provided by the SPI bus master.
SS / SDA / Rx	pin acts as a signal input when the SPI interface is enabled, as serial data when the I2C interface is enabled and as a serial data input when the UART interface is enabled.

Arduino UNO:

- **Arduino Uno** is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button.
- The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.[14]

CHAPTER 8: HARDWARE

CIRCUIT DESIGN

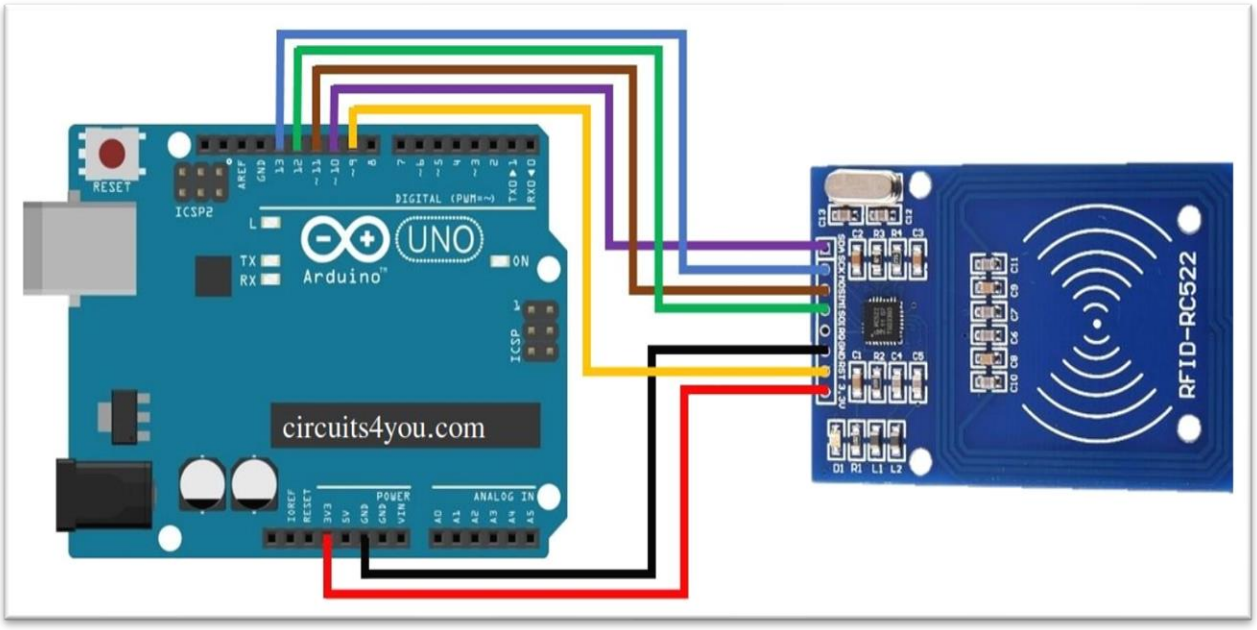


Figure 21 HW circuit design.

Smart Card:

Contact **cards** have a **smart chip** embedded in the surface of the **card**, and it must meet the reader to **work**. Contactless **cards** have a **chip** and antenna system that can wirelessly send the data to the reader once it comes into range.

RFID TAGS:

- It is considered the heart of the RFID technology and can be fixed inside of Library Books or directly for CDs, DVDs, and other Materials.
- It works with a microchip that consists of integrated circuits (silicon chip) that are programmed to store unique identification number of the book or the materials in Library. The size of the tags depends on the antenna, which increases with range of tag and decreases with frequency.[15]

CHAPTER 8: HARDWARE

Figure 30 RFID card

Arduino Ide

It will be the software which is used for controlling the card by reading and writing on it and specifies the main output that will show when the reader connects with the card.

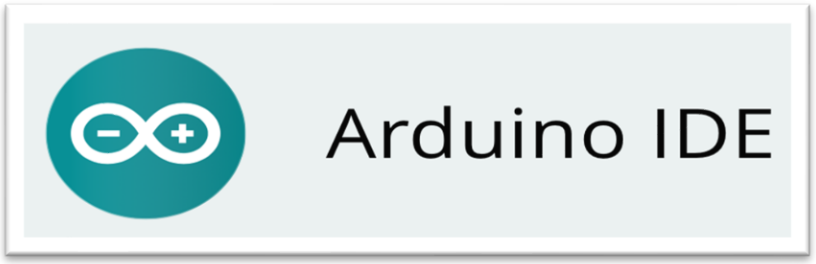


Figure 31 arduino IDE

- MFRC522 library:

  There's a library called MFRC522 library which simplifies reading from and writing to RFID tags. Once the library is installed, open Examples submenu and select MFRC522 > DumpInfo example sketch.

  We will not write any data to the tag. It just tells us if it managed to read the tag and displays some information about it.

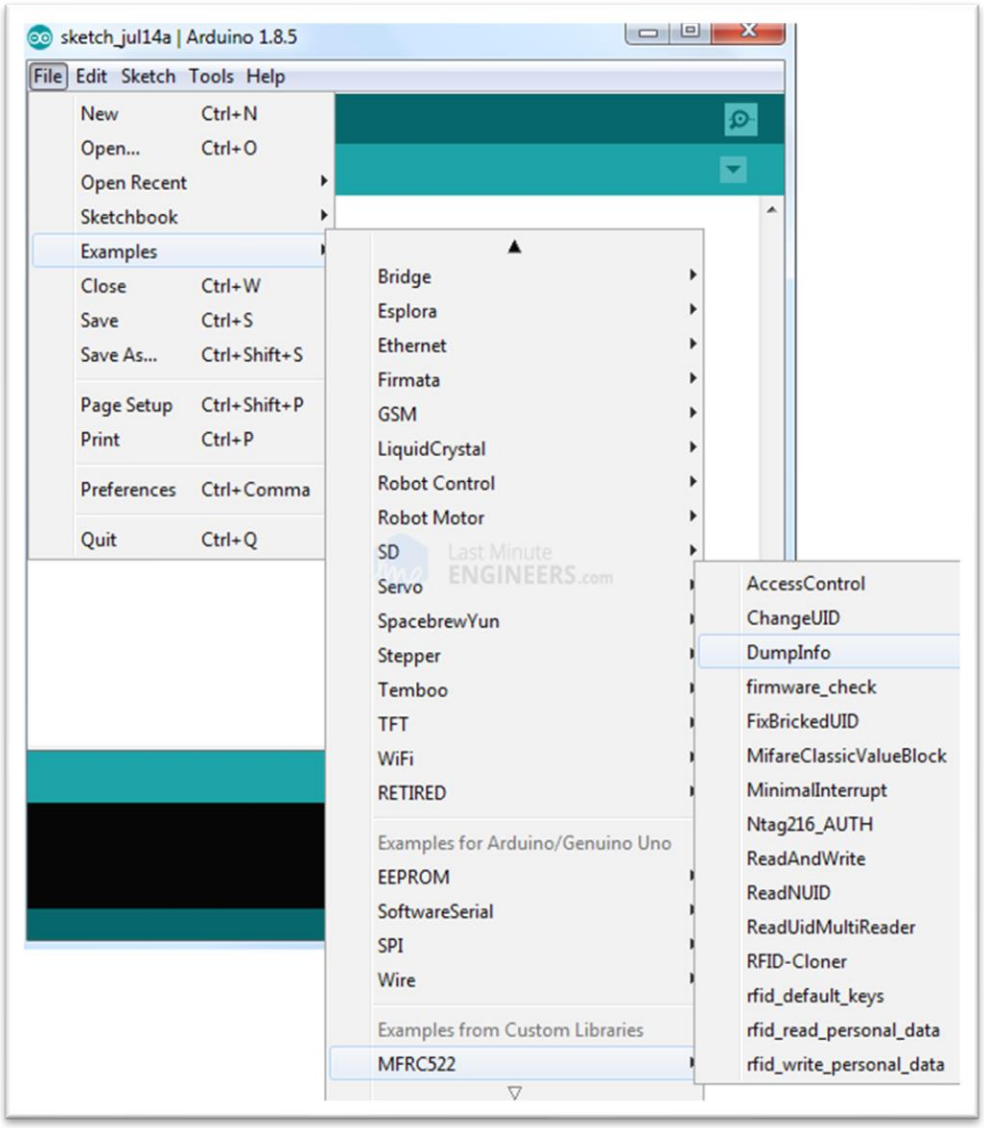


Figure 32 MFRC522 library insertion.

Chapter 9: The Application user interface

1. HOME PAGE

The home page includes sign in, sign up, about us and contact us services.

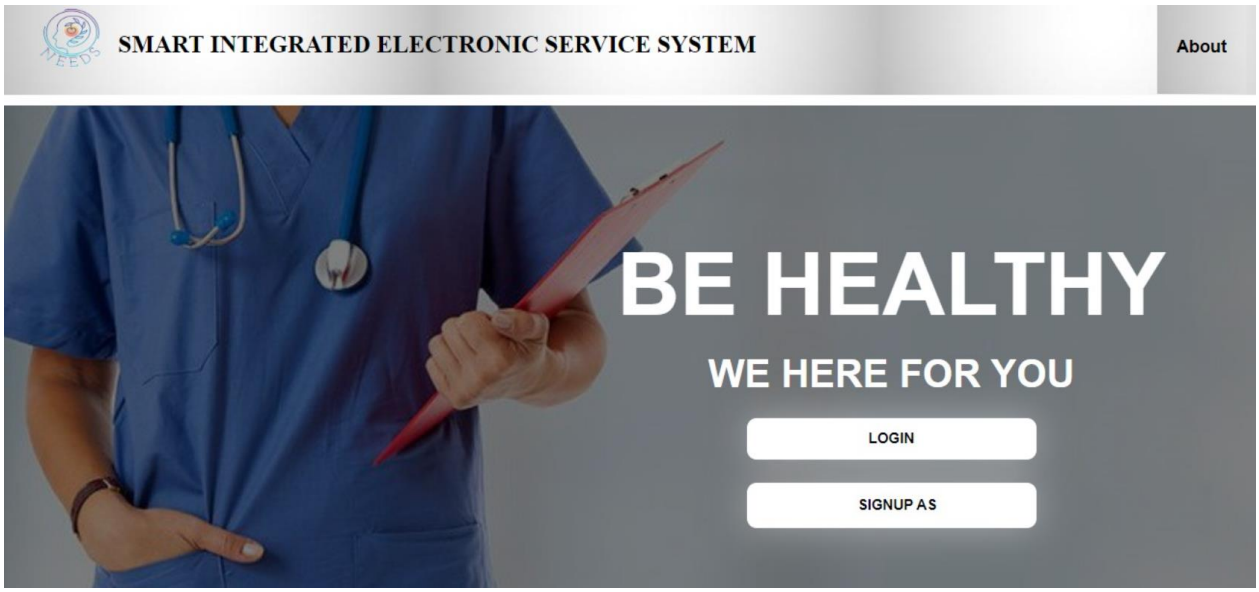


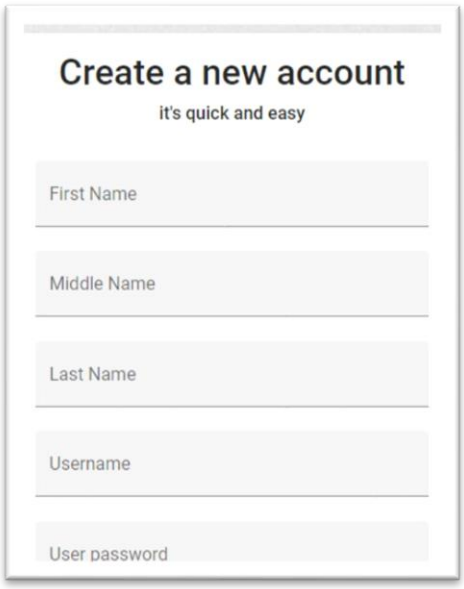
Figure 33 home page

2. SIGNUP PAGE

The signup page, also called the registration page is a really importantpage if it is first time to user to use application.

→ The required data are:

- First name, middle name, last name, username
- Phone Number, street, apartment number, city
- Email, password, relative email
- If signing up as doctor, entering working place information as well.



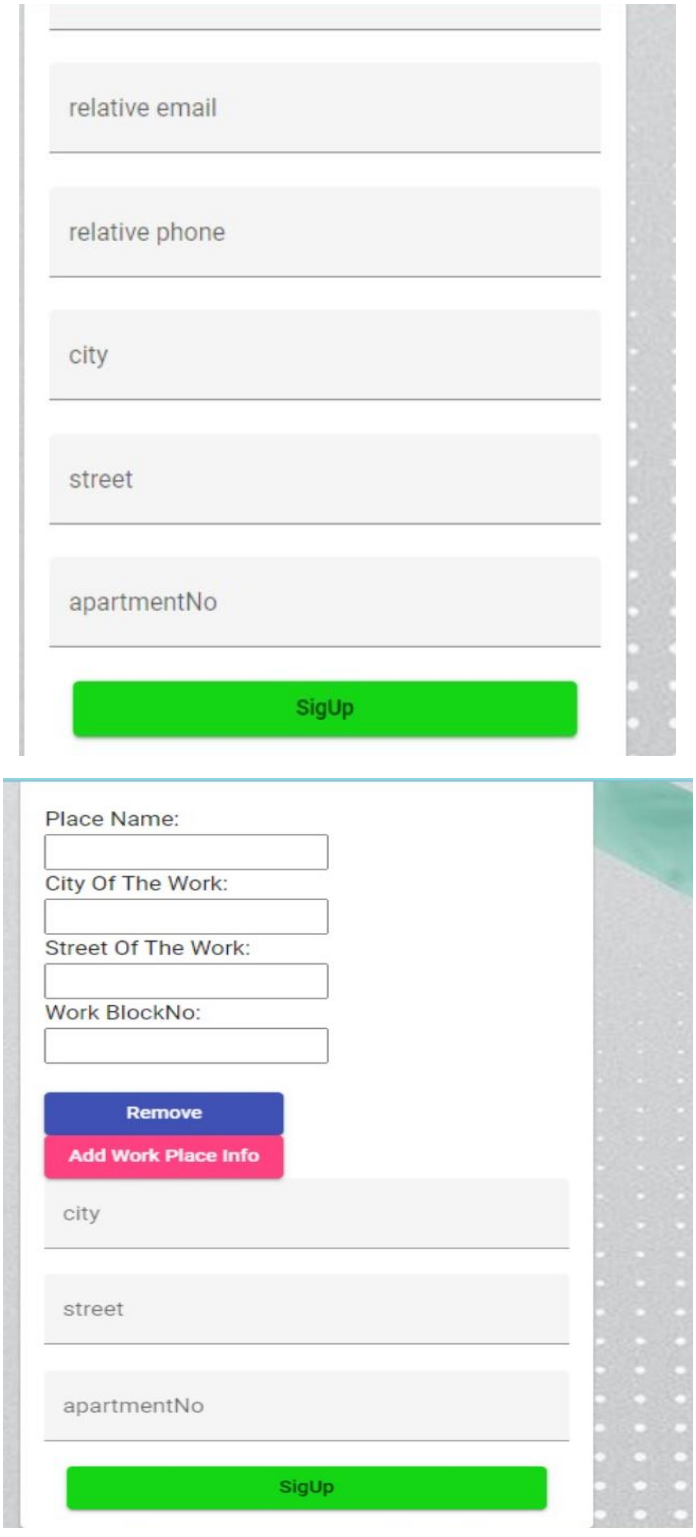


Figure 34 sign up pages

CHAPTER 9: THE APPLICATION USER INTERFACE

3. LOGIN PAGE

User must Enter the username and password.

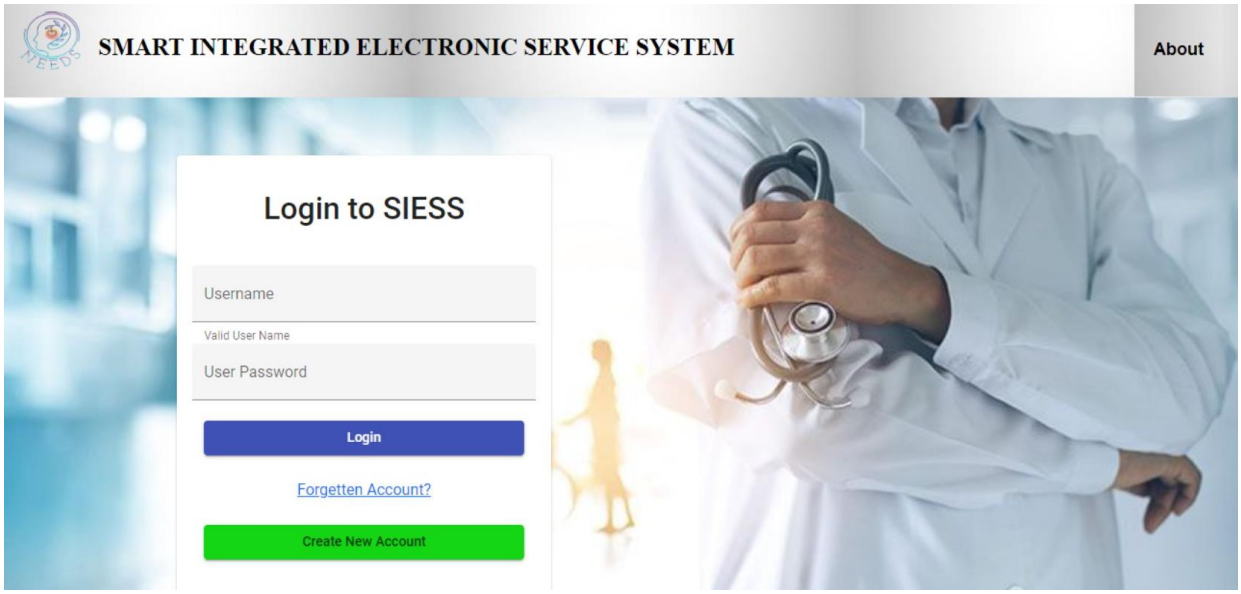


Figure 35 login page

4. CONTACT US PAGE

The contact us page allows users to contact with us if there is any problemor something that user want to inform us with, This form requires the following:

- User name
- Email address
- Phone number
- The message itself

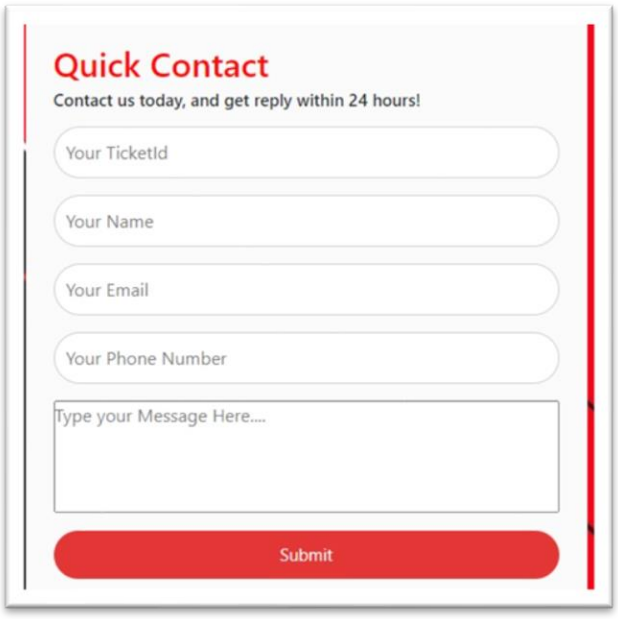


Figure 36 contact us page

5. ADMIN CONTACT US VIEW REQUESTS PAGE

This page shows all Contact us requests to the admin, to let him contact them back.

Figure 37 Admin viewing the contact us requests.

6. LOST-CARDS PAGE PATIENT VIEW

This page allow patient to send us all details for his card loss to disable it.

This form requires the following:

- The user name
- Email address
- Id
- How did he lost his card, and any additional information.

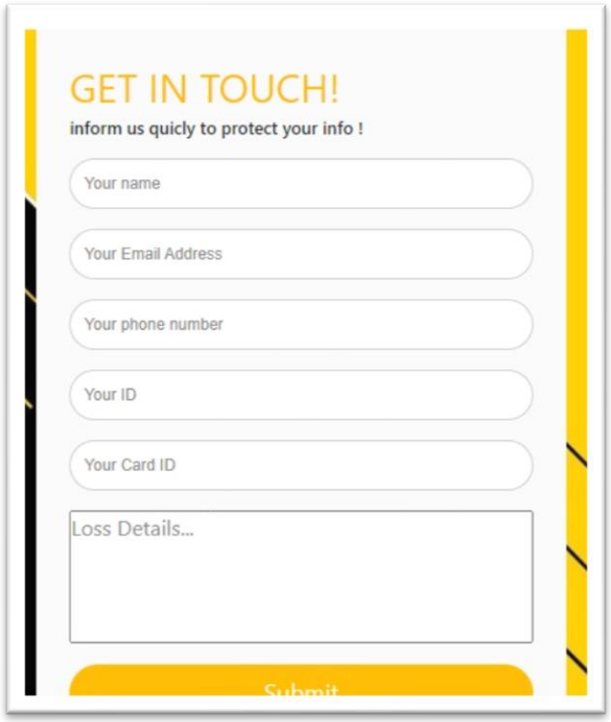


Figure 38 lost card page

7. ADMIN LOST-CARD VIEW REQUESTS

This page shows all requests of lost cards to the admin to take the legal steps.

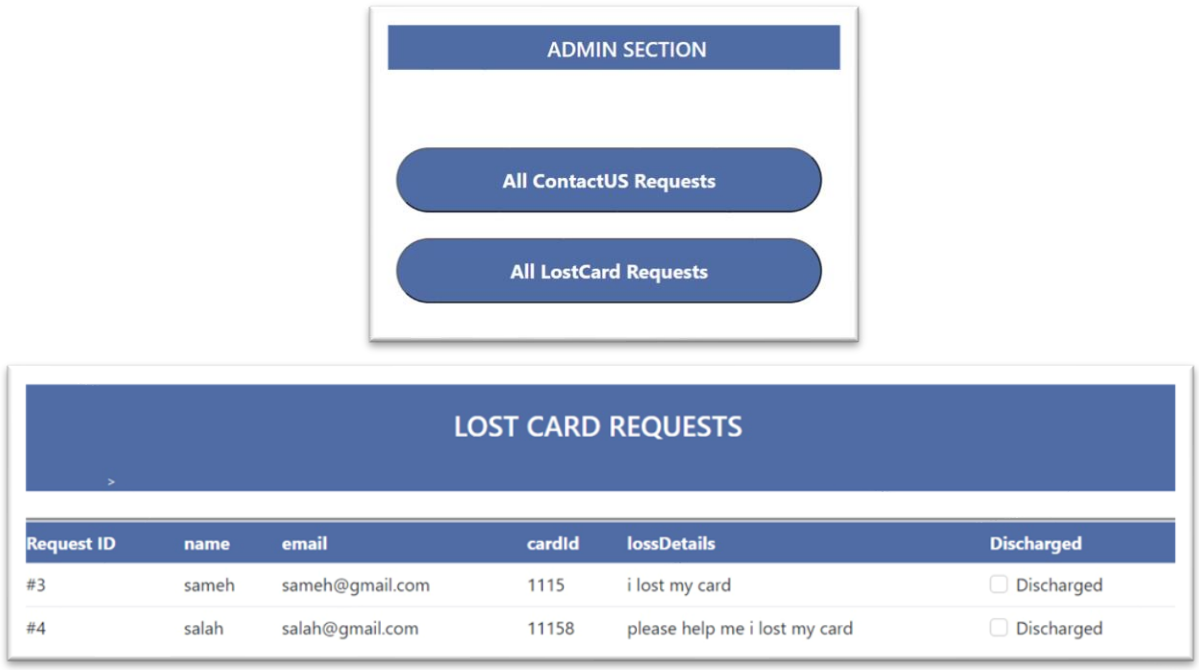


Figure 39 lost card requests admin view

7. PATIENT SHOW SERVICES PAGE

Shows all options for patient to choose the section he wants to discover. The application offers the patient to show the following:

- His Basic and necessary information.
- Current Medicines that user take
- Lab Results, including the test results or scan results.
- Reminders for the medicines to be taken
- Lost card request page

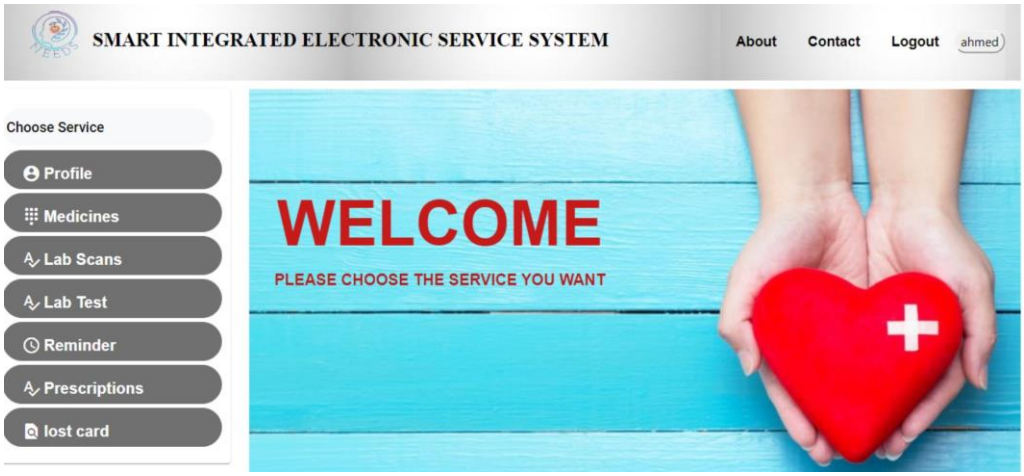


Figure 40 select services page

8. PATIENT BASIC INFORMATION PAGE

All necessary information for the patient to make it easy for emergency doctors and specialist doctor to take a quick review about the patient and contact his relative.

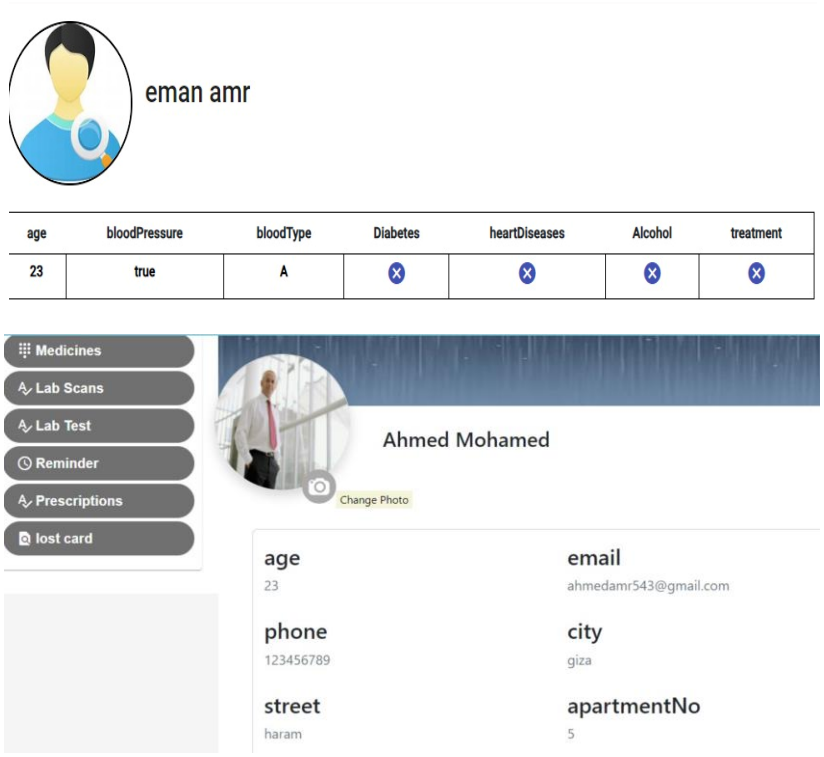
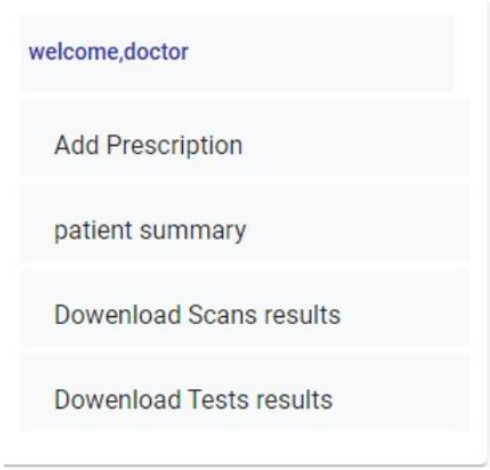
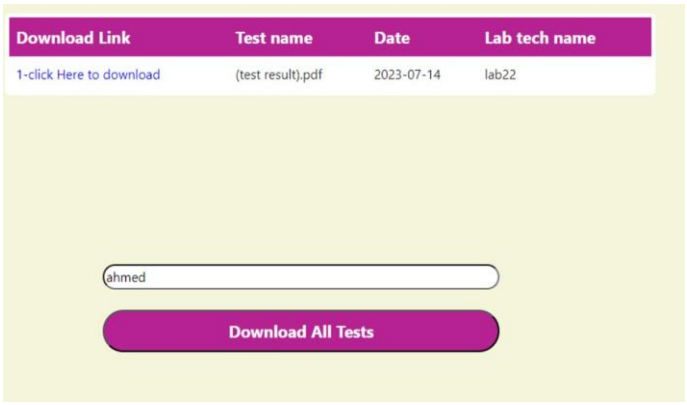


Figure 41 patient profile

CHAPTER 9: THE APPLICATION USER INTERFACE

9. DOCTOR PROFILE PAGE





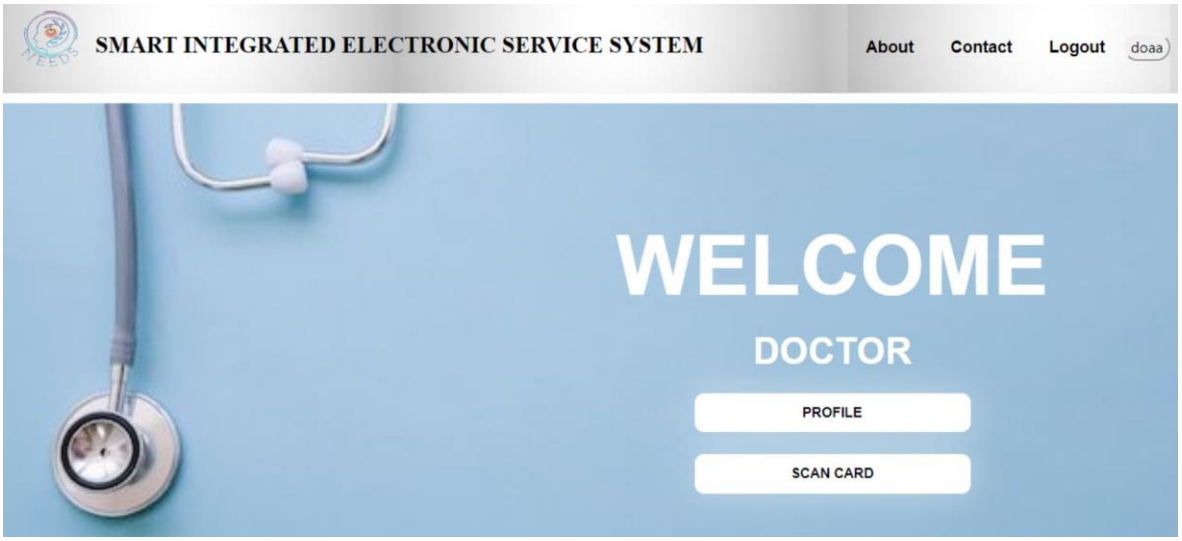


Figure 42 profile welcome page

CHAPTER 9: THE APPLICATION USER INTERFACE

10. AFTER SCANNING CARD, DOCTOR CAN SEE THE PATIENT'S SUMMARY INFO

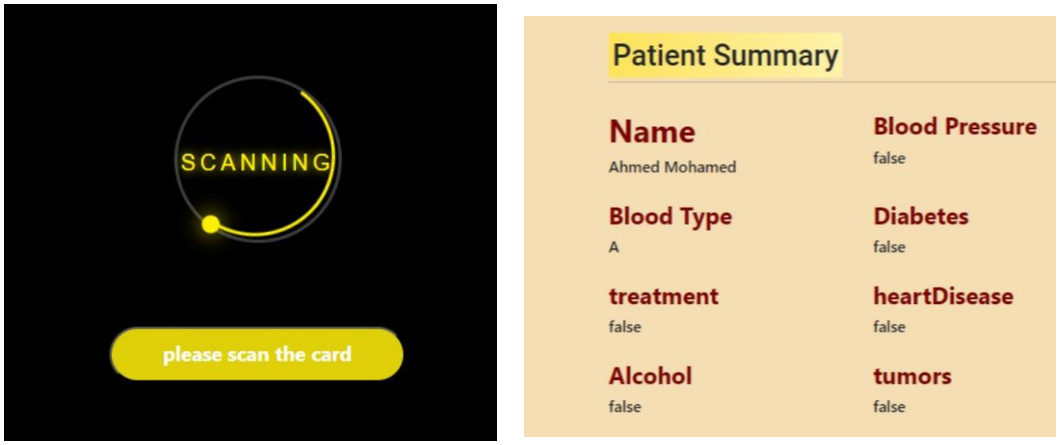
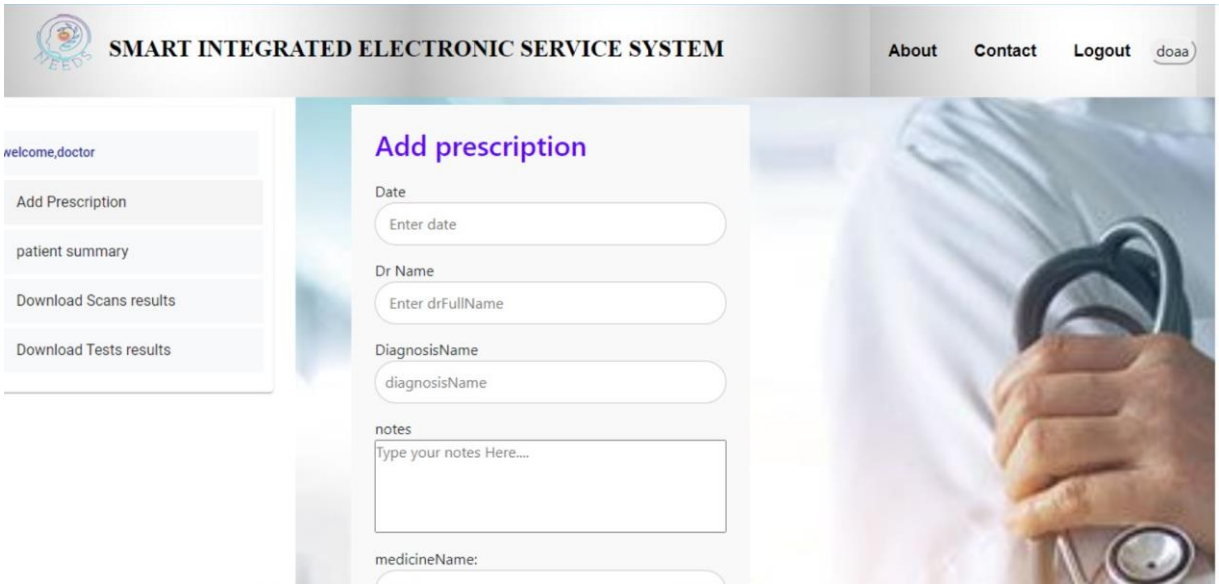


Figure 43 patient summary

11. ADD NEW PRESCRIPTIONS

Allow the specialist doctor to write the prescription, for a specific patient



CHAPTER 9: THE APPLICATION USER INTERFACE

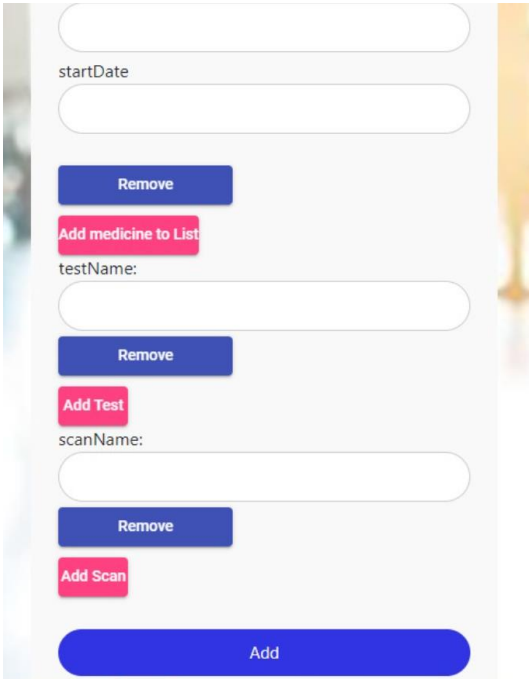


Figure 44 add prescription

12. PRESCRIPTIONS VIEW

The patient can see all the previous prescriptions with details.

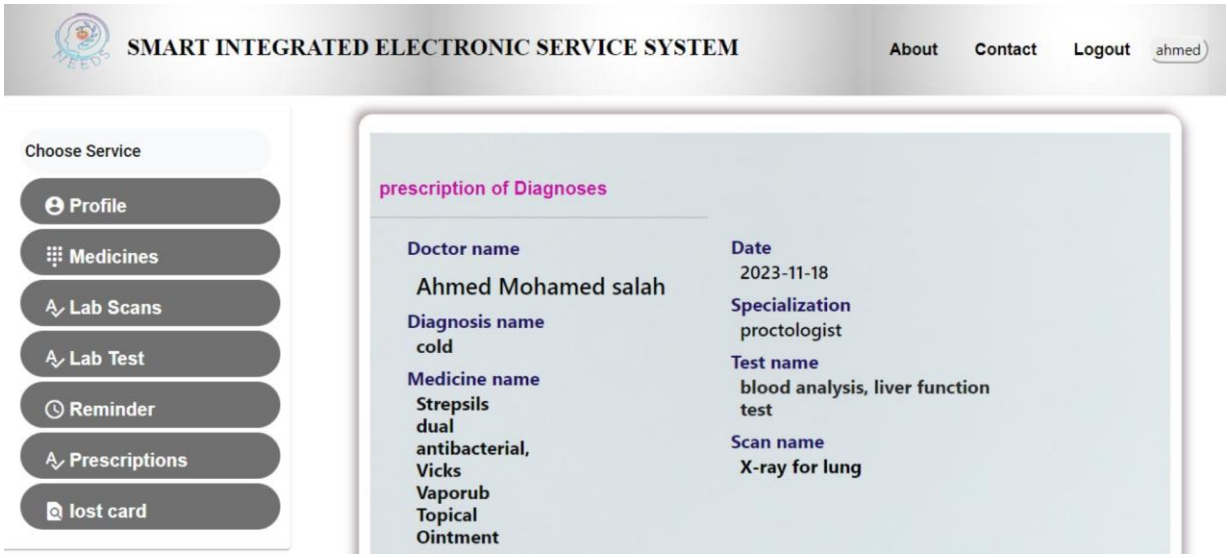
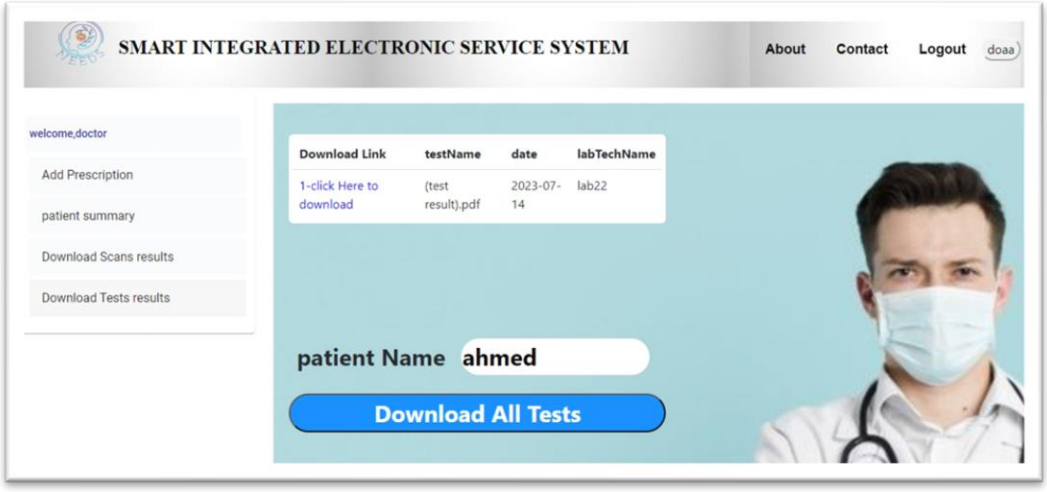


Figure 45 show prescription

CHAPTER 9: THE APPLICATION USER INTERFACE

13. SHOW SCAN RESULTS AND TEST RESULTS



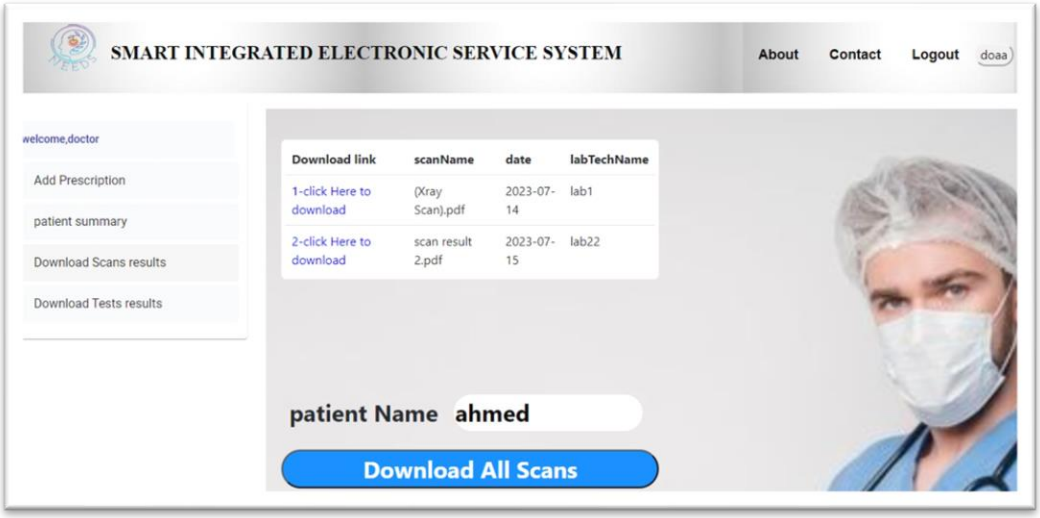
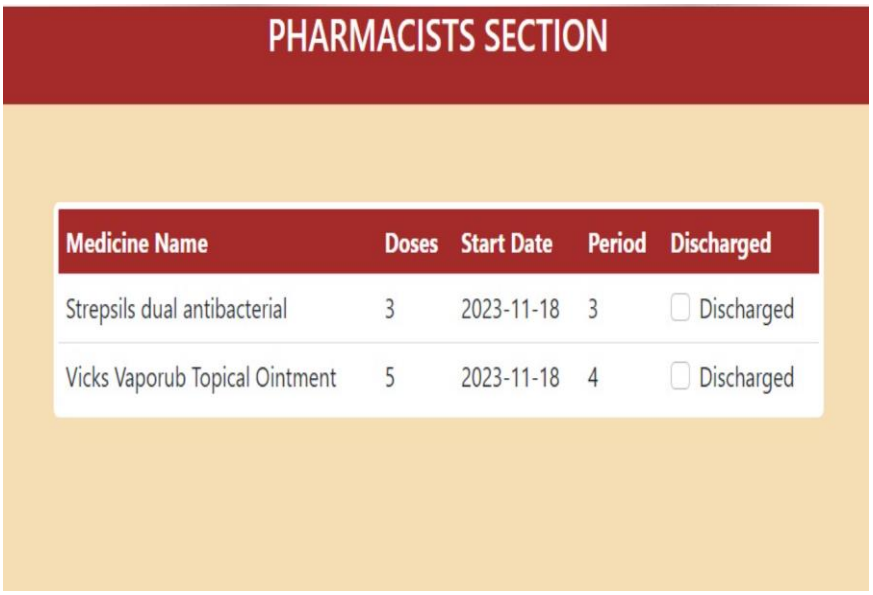


Figure 46 download all tests and scan

CHAPTER 9: THE APPLICATION USER INTERFACE

14. PHARMACISTS

For all pharmacists to view the prescriptions.



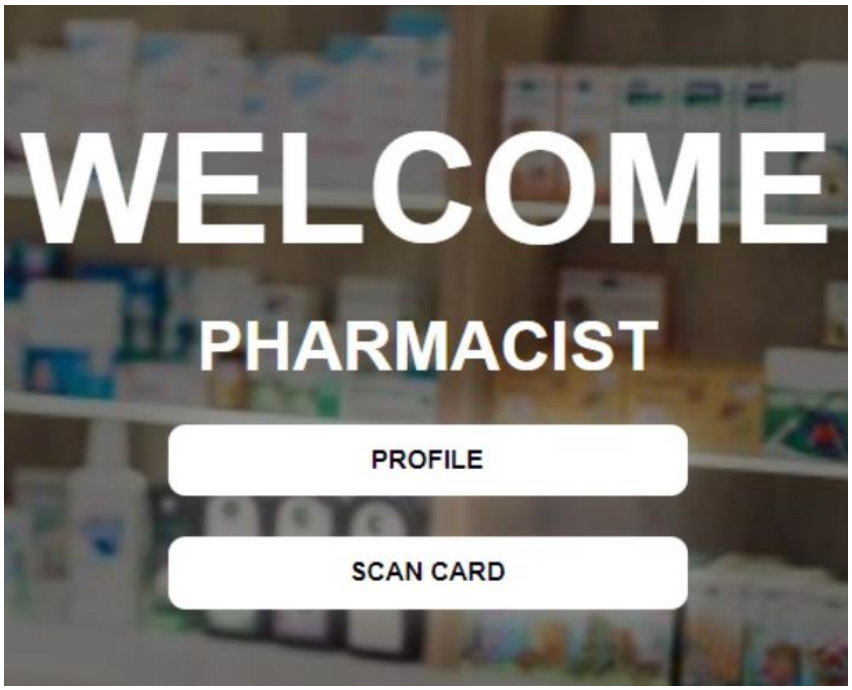


Figure 47 pharmacist

CHAPTER 9: THE APPLICATION USER INTERFACE

15. EMERGENCY DOCTOR SECTION

The information for the emergency doctors to view which helps them to save the patient faster.

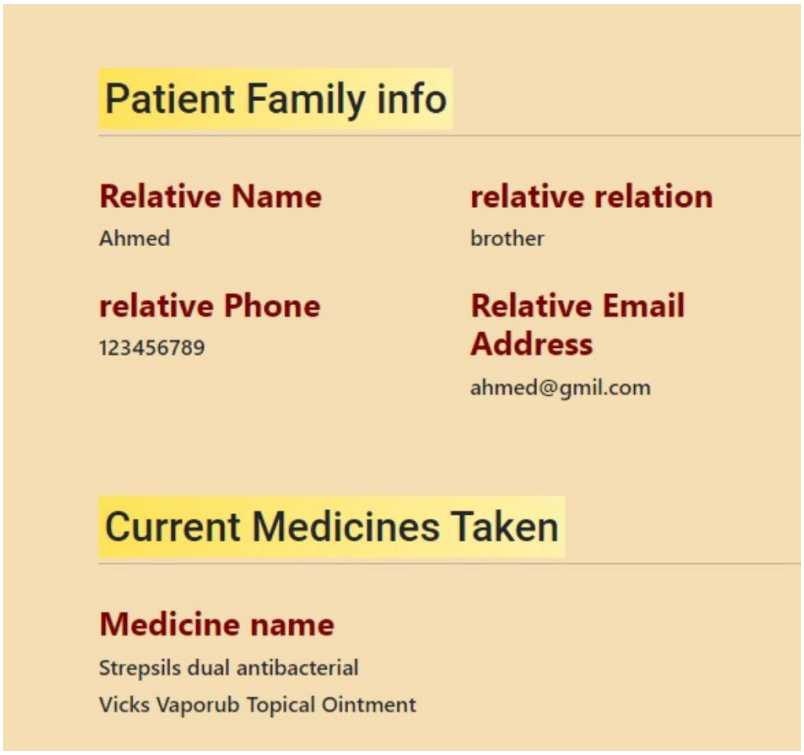


Figure 48 patient family info

16. CURRENT MEDICINES

Shows the patient the current medicines he take.

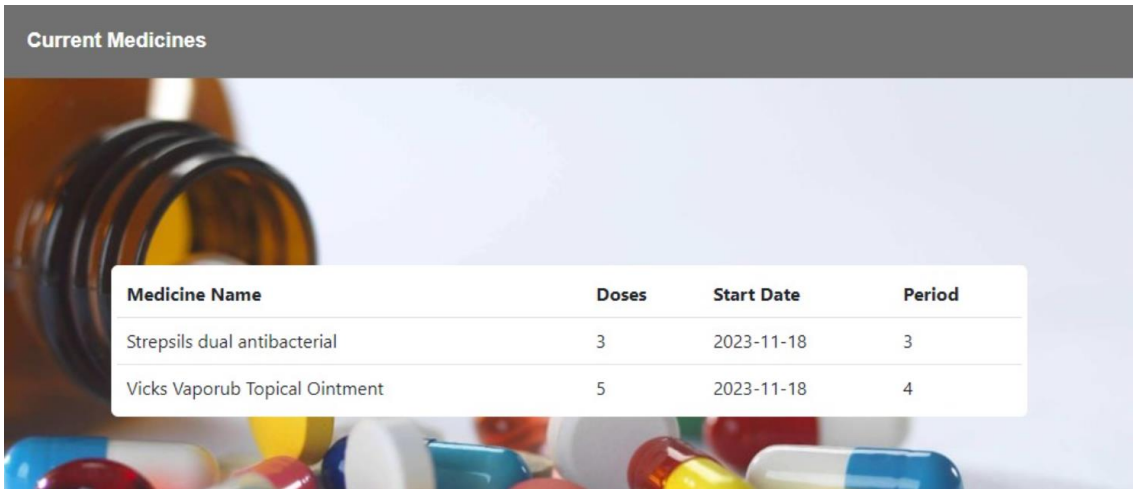


Figure 49 current medicines

CHAPTER 9: THE APPLICATION USER INTERFACE

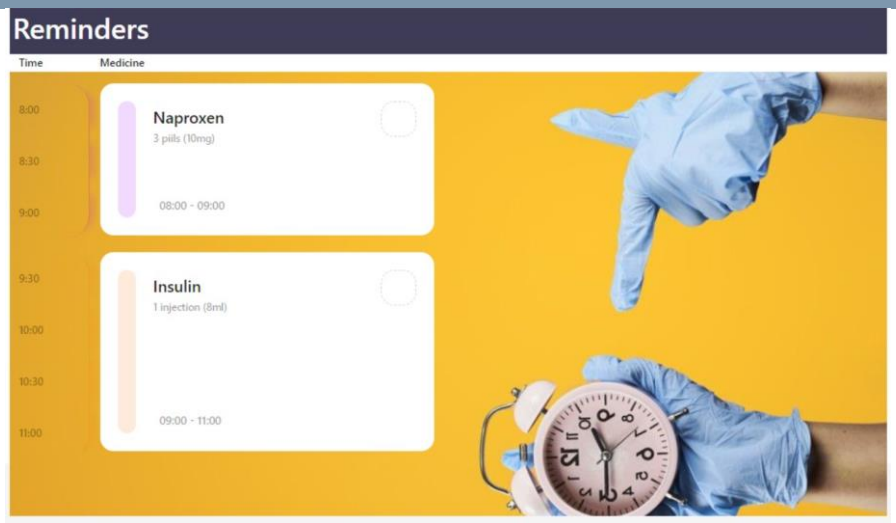


Figure 50 reminders

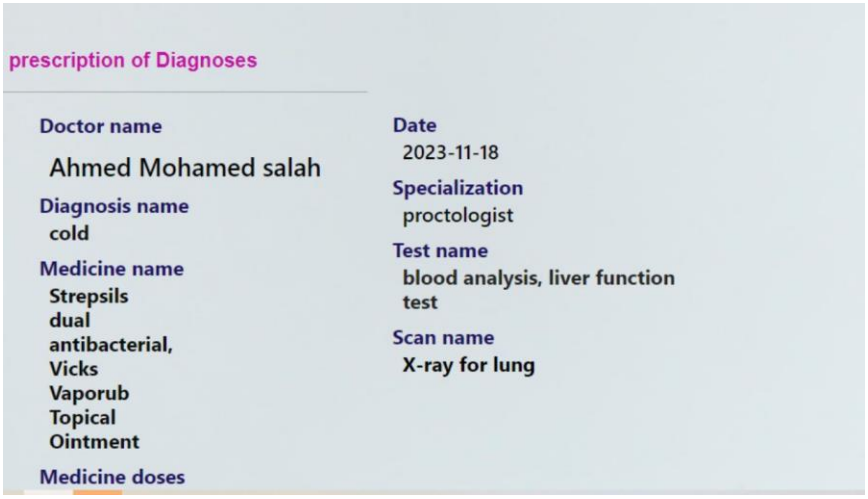


Figure 51 view prescription

CHAPTER 9: THE APPLICATION USER INTERFACE

17.LAB TECHNICIAN SECTION

This page shows the tests and scans required to the technician and allow him to upload the results of either the scan or tests.

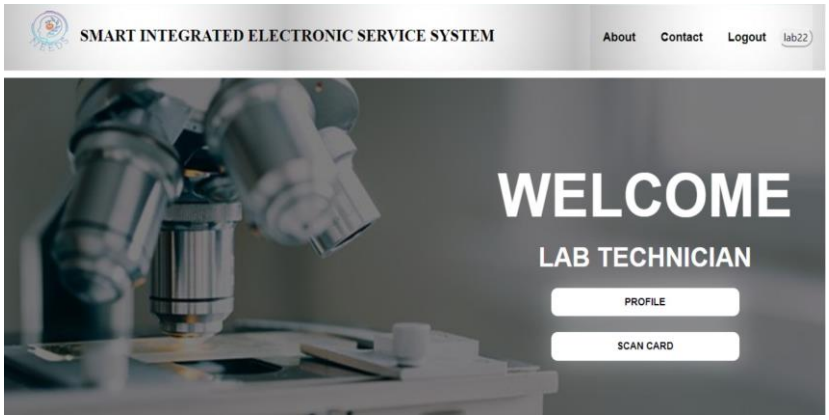


Figure 52 lab technician welcome page

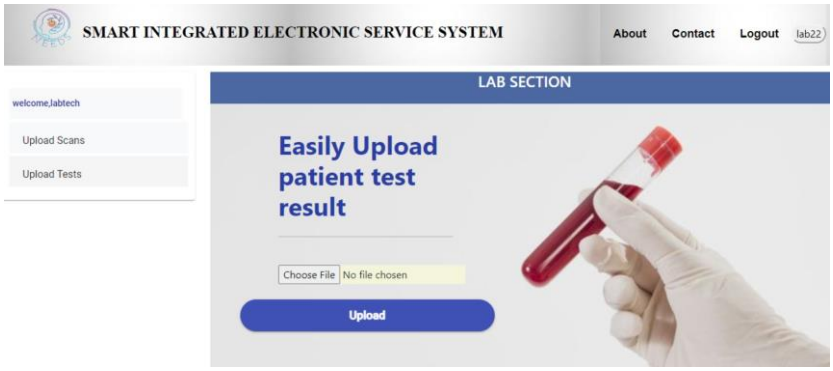


Figure 53 upload tests results pages

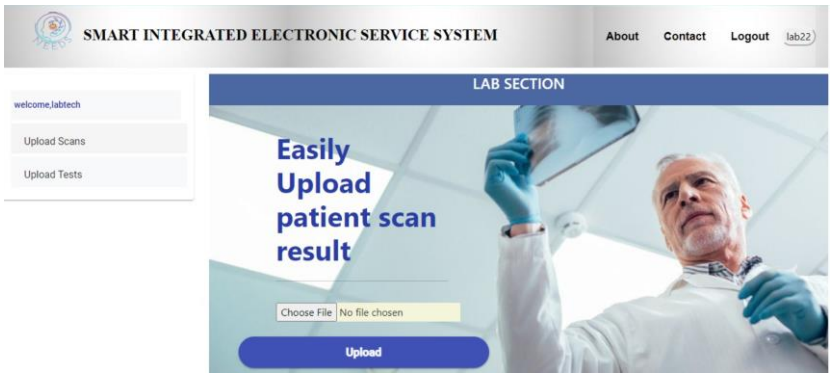


Figure 54 upload scan results pages

CHAPTER 9: THE APPLICATION USER INTERFACE

18.FORGOT PASSWORD

Forget password and reset password pages, you can request to reset your password by entering the verified email.

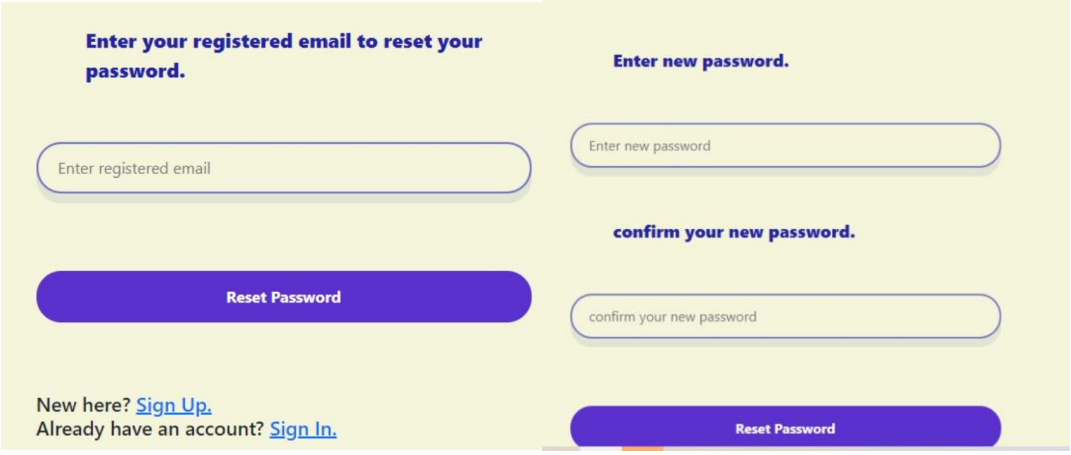


Figure 55 reset password page

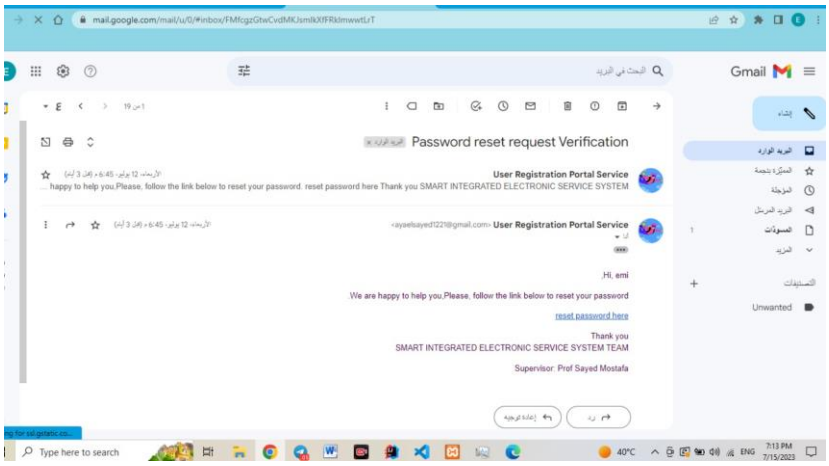


Figure 56 password reset token

19.ABOUT US PAGE

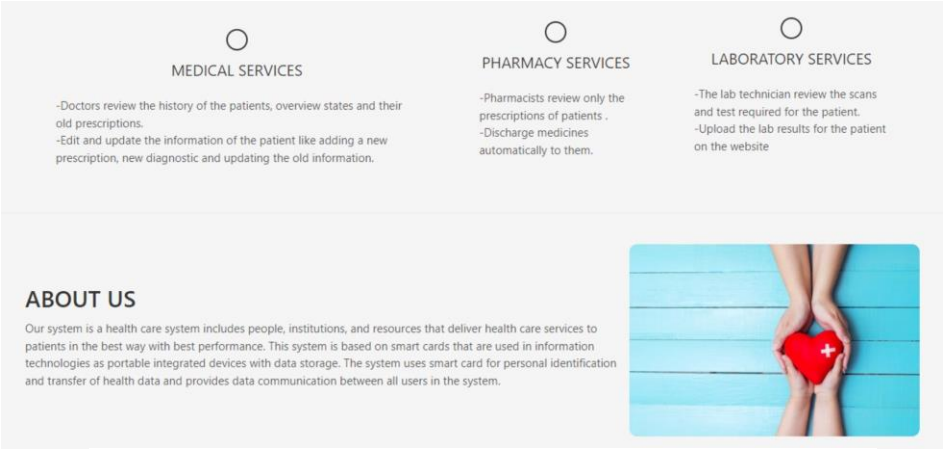


Figure 57 about us

CONCLUSION

Conclusion

In conclusion, the smart integrated electronic service system we developed aims to revolutionize the medical field by providing efficient and convenient services to all citizens. By utilizing new technologies we were able to enhance the functionality and efficiency of our system.

The integration of angular with Spring Boot framework offers numerous advantages, including simplified database operations, reduced boilerplate code, and improved performance and scalability. This ensures that our system can effectively manage large amounts of data and handle a high volume of concurrent requests.

Furthermore, the system promotes sustainability by eliminating the need for printing papers and reducing pollution. It also helps control the discharging of medicines without verified prescriptions, ensuring patient safety and preventing illegal practices.

The user-friendly interface and smart features of the system make it quick and easy to access and manage patient information. In emergency cases, the system provides immediate access to a patient's medical history, allowing healthcare professionals to provide appropriate first aid and contact the patient's family or friends for support.

Overall, our smart integrated electronic service system is a comprehensive solution that addresses the needs of the medical field while promoting sustainability and improving patient care. With its user-friendly interface and advanced features, it provides a seamless experience for both healthcare professionals and patients.

Future Work

In the future, this project can serve as a prototype for the digitalization of all citizens' data across various sectors. One potential area where this system can be expanded is in airports and travel. By integrating the smart integrated electronic service system with airport databases, travelers can have a seamless experience from check-in to boarding. This would eliminate the need for physical tickets and reduce waiting times at security checkpoints.

Additionally, this system can be extended to include information related to cars, such as vehicle registration, insurance details, and maintenance records. By digitizing these documents, it would be easier for individuals to access and manage their car-related information. This could also help in reducing fraud related to stolen or forged documents.

Furthermore, policies, licenses, and fees can also be integrated into the system. Citizens would be able to access information about various policies and licenses required for different activities such as starting a business or obtaining a driver's license. They would also be able to pay fees online, eliminating the need for physical visits to government offices.

Another area where this system can be expanded is in managing police records and paperwork. By digitizing police reports and other relevant documents, law enforcement agencies can have quick access to information during investigations. This would streamline processes and improve efficiency in solving crimes.

Moreover, by expanding the smart integrated electronic service system to include other sectors such as education and healthcare, citizens would have a centralized platform for accessing all their important information. For example, students could access their academic records and apply for scholarships through the system.

References

1. White CM, Coleman CI, Jackman K, et al. AHRQ series on improvingtranslation of evidence: linking evidence reports and performance measures to help learning health systems use new information for improvement .Jt Comm J Qual Patient Saf. 2019;45(10):706-10.
2. Borsky AE, Flores EJ, Berliner E, et al. AHRQ Evidence-based PracticeCenter Program: applying the knowledge to practice to data cycle to strengthen the value of patient care. Journal of Hospital Medicine. 2019.
3. https://www.researchgate.net/publication/267323193_The_Use_of_Smart_Cards_in_Health_Care
4. Van Gennip E.M.S.J., Talmon J.L., Assessment and Evaluation of Information Technologies in Medicine, IOS Press, Amsterdam 1995
5. R. Beuscart and P. Paradinas. Smart Cards for Health Care, in Telematics in Medicine, Elsevier Science Publishers B.V., North-Holland,1991.
6. P. Peyret. RISC-Based, Next-Generation Smart Card Microcontroller Chips, in proceedings of CardTech'94, Washington D.C., U.S.A., April 1994.
7. Chiaretta, S. (2018). *Front-end Development with ASP.NET Core, Angular, and Bootstrap*.
8. E. Gordons and G. Grimonprez. A card as element of a distributed database, IFIP WG 8.4 Workshop, P.Paradinas and G. White : The portable office. Microprocessor cards as elements of distributed offices,Ottawa, Canada, 1992.
9. Guntupally, K., Devarakonda, R., & Kehoe, K. (2018). *Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example*.
10. P. Paradinas. The CQL Database Smart Card, GMD, Smart Card Workshop, Darmstadt, Germany, February 1994.
11. Ribeiro, J. L. P., Zenha-Rela, M., & Vega, F. (2007). *Using Dynamic Analysis Of Java Bytecode For Evolutionary Object-Oriented Unit Testing*.
12. Almeida, M. P., & Canedo, E. D. (2022). Authentication and Authorization in Microservices Architecture: A Systematic Literature Review. *Applied Sciences, 12*(6), 3023. https://doi.org/10.3390/app12063023.
13. Agarwal, S., Singh, A. K., Aniraj, V., Pandey, A., Prasad, D., & Nath, V. (2020). RFID (MF-RC522) and Arduino Nano-Based Access Control System. In *Lecture notes in electrical engineering* (pp. 561–567). Springer Science + Business.
14. Dr. L. Santhi1, S. Lakshmi2, R.Sakthivel3. RFID Technology and Its Applications with Reference to Academic Libraries (pp. 1-6). IOSR Journal of Computer Engineering (IOSR-JCE).