Project Report On

**Laboratory Management System**

Submitted by

**LATHIYA NIKHIL KUMAR RAJIVBHAI**

**CHAUHAN AYUSH VRAJLALBHAI**

**Group No:** **10**

Bachelor of Computer Application (Sem – VI)

Submitted To:

DNV INTERNATIONAL EDUCATION ACADEMY, PLOT NO.3, WARD 9/A, NEAR SUBHASH NAGAR, GANDHIADHAM, KUTCH

**Year :- 2022-23**

**Acknowledgement**

It was a highly eventful session at the **DNV INTERNATIONAL EDUCATION ACADEMY** & working with **KSKV KACHCHH UNIVERSITY** and highly devoted computer teacher’s community and will probably remain the memorable experience of our life. Hence this acknowledgement is a humble attempt to earnestly thank all those who were directly or indirectly helping us.

We are lifelong indebted to our project guide **Prof. Shreedutt Sharma** for providing us with the environment that is best suited for outgrowth of the students’ personality and boosting their morale.

Above all, how can we forget **Prof. Shreedutt Sharma**, our respected project coordinator who gave directions to our work and shaped our imagination. We are deeply indebted to him for providing immeasurable guidance and reposing faith is us when things went wrong.

And finally, there was and always will be love for all the professors who have taught us as everything that is written in this report has been the fruit of the knowledge that they gave us.

Thanking All Professors,

**Nikhil Lathiya and Ayush Chauhan.**

**Logo

Description automatically generated**

DNV INTERNATIONAL

EDUCATION ACADEMY

BACHELOR OF COMPUTER APPLICATION

**Certificate**

This is to certify that **Nikhil Lathiya** Exam Seat Number: \_\_\_\_\_\_\_\_\_\_\_\_ and **Ayush Chauhan** Exam Seat Number: \_\_\_\_\_\_\_\_\_\_\_\_ has satisfactorily completed his project work entitled **Laboratory Management System** as a fulfillment of the requirements, during the academic year 2022 – 2023.

Date:

Place:

Principal: Dr. Rashmi Chandnani

BCA, DNV INTERNATIONAL EDUCATION

ACADEMY, Gandhidham, Kutch.

Project Coordinator: Prof. Shreedutt Sharma

BCA, DNV INTERNATIONAL EDUCATION

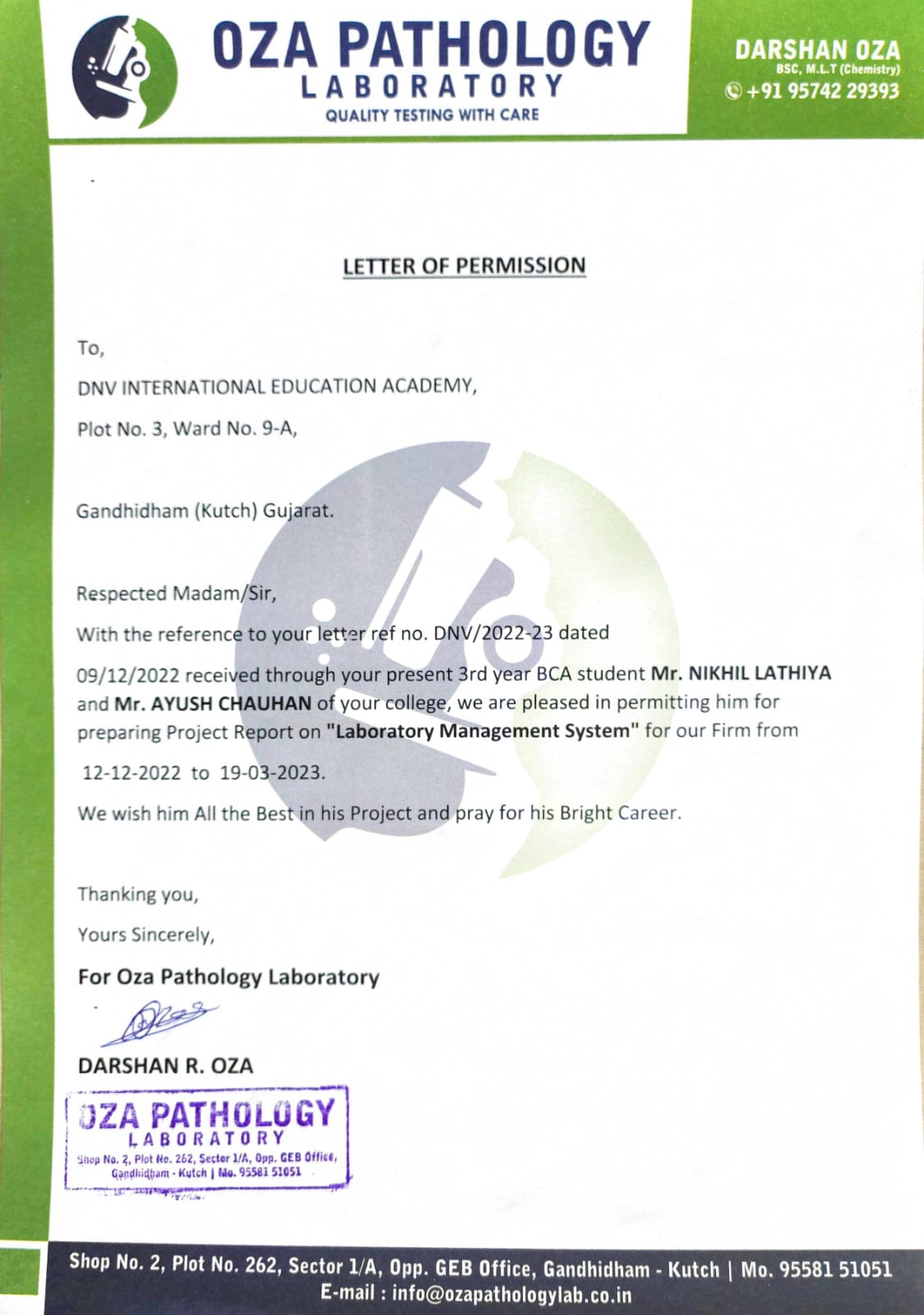
ACADEMY, Gandhidham, Kutch.

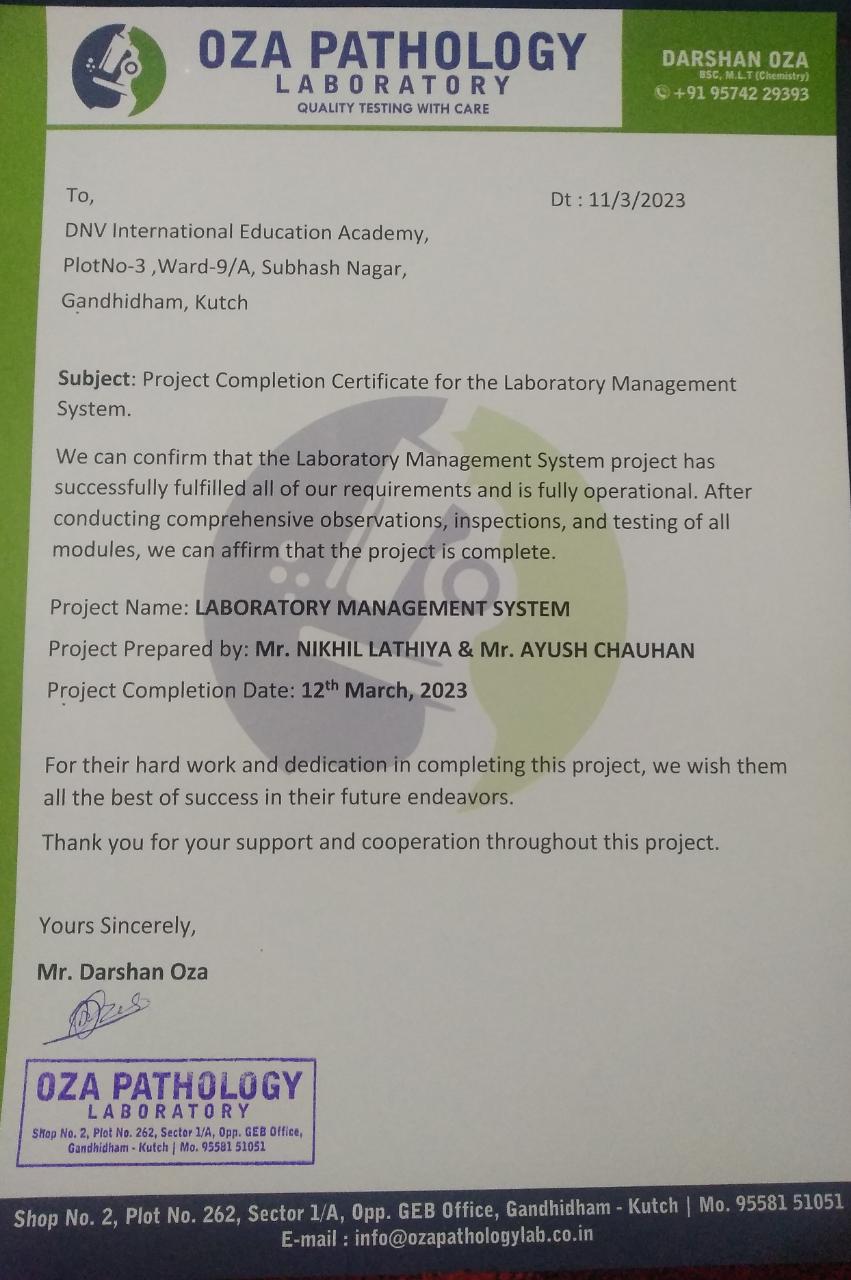
**Preface**

It is the trace that study cannot be complete without practical knowledge. It is a necessity of a technical student to gain practical knowledge. Technical knowledge is like a coin with two sides. One side represents theoretical knowledge, and the other side represents practical/technical knowledge.

This is Semester – 6 BCA during which students are trained in Web Application Development. Along with different computer languages & technologies, topics such as service-oriented architecture, smart device computing and wireless technologies are covered in this course.

Hence, I have taken up this project of **LABORTAORY MANAGEMENT SYSTEM**



****

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr. no.** | **Title** | **Page No.** |
|  | **Introduction**   * 1. Introduction   2. Project Summary | 1 |
| 2. | **Proposed System**  2.1 Introduction to Laboratory Management  (process).  2.2 Advantages & Limitations.  2.3 Project Profile.  2.4 User Profile  2.5 Technical Details.  2.6 Use Case Diagram. | 4 |
| 3. | **Platform/Environment**  3.1 Specification of hardware and software.  3.2 Software Interface | 12 |
| 4. | **Structural Analysis**  4.1 Data Flow Diagram.  4.2 Sequence Diagram.  4.3 Activity Diagram.  4.4 ER Diagram.  4.5 Time Line Chart.  4.6 Relationship Between Tables | 15 |

|  |  |  |
| --- | --- | --- |
| **Sr. no.** | **Title** | **Page No.** |
| 5. | **Database Table Design** | 30 |
| 6. | **Interface Design** | 35 |
| 7. | **Testing**  7.1 Unit Testing.  7.2 Interface Testing.  7.3 Document Testing.  7.4 Final Testing. | 53 |
| 8. | **Bibliography** | 55 |

**Introduction**

**Laboratory Management System**

* 1. **Introduction**
* The Laboratory Management System provides an efficient way to manage laboratory operations.
* The software simplifies the management process and handles various elements related to laboratory business.
* With the use of this software, paperless records can be maintained, and laboratory workflows can be streamlined.
* The software enables users to book appointments online and download their reports, making the process more convenient.
* The Laboratory Management System provides an integrated platform to manage laboratory workflows, including data management.
  1. **Project Summary**

**PURPOSE**

The Laboratory Management System is specifically designed for "**OZA PATHOLOGY LABORATORY**" to reduce the reliance on physical documents. The system offers patients the convenience of online access to their reports, bills and appointment booking, as well as the ability to download and view them.

Through this system, patients can view their lab reports and bills anytime and anywhere, without the need for physical copies. They can also book appointments online, which saves time and effort for both the patients and the laboratory staff.

Additionally, the Laboratory Management System helps in organizing and managing lab reports, bills and appointments, thus reducing the risk of errors and delays. It also ensures secure storage of data and access to authorized personnel only. Overall, the Laboratory Management System streamlines the lab process and improves the patient experience, while ensuring efficient and accurate reporting.

**Proposed**

**System**

**2.1** **Introduction to Laboratory Management System**

The Laboratory Management System provides different levels of access for **Super Admin, Admin, and User**.

**Super Admin and Admin:**

* Super Admin and Admin can add, view, delete, and edit patient details with OTP verification, ensuring the accuracy and security of the data.
* Super Admin and Admin can generate, view, edit, and delete patient reports, making it easy to manage and retrieve patient data.
* **Super Admin can add or remove Admin, ensuring that only authorized individuals have access to the system.**
* Super Admin and Admin can view appointment requests submitted by patients and take action such as confirming or rejecting them.
* Super Admin and Admin can view test details, with **Super Admin being the only one who can add and activate or deactivate tests.**

**User:**

* Users can book appointments online for a specified date.
* Users can download their reports and bills online, making the process more convenient and saving time for both patients and staff.
* Overall, the Laboratory Management System provides an efficient and secure solution for managing laboratory operations, with different levels of access provided to ensure the system is used only by authorized individuals.

**2.2 Advantages and limitations**

|  |  |
| --- | --- |
| **Advantages** | **Limitations** |
| * Paperwork can be eliminated. * Managing data becomes easy. * Large number of data can be accessed with ease. * Patient Can book their appointment online. * E-bills and E-Reports Can be generated. * Patient can download their report by this website. * Analysis the ratio of patients, Reports & tests. | * E-mail is Mandatory. * Test Measurement should be input in JSON format |

**2.3 Project Profiles**

|  |  |
| --- | --- |
| **Project Title** | **Laboratory Management System** |
| Project Definition | It allows to manage Laboratory and related services digitally. |
| Team Members | 2 |
| Team Member Name | Nikhil Lathiya &  Ayush Chauhan |
| Time Duration | 3 Months |

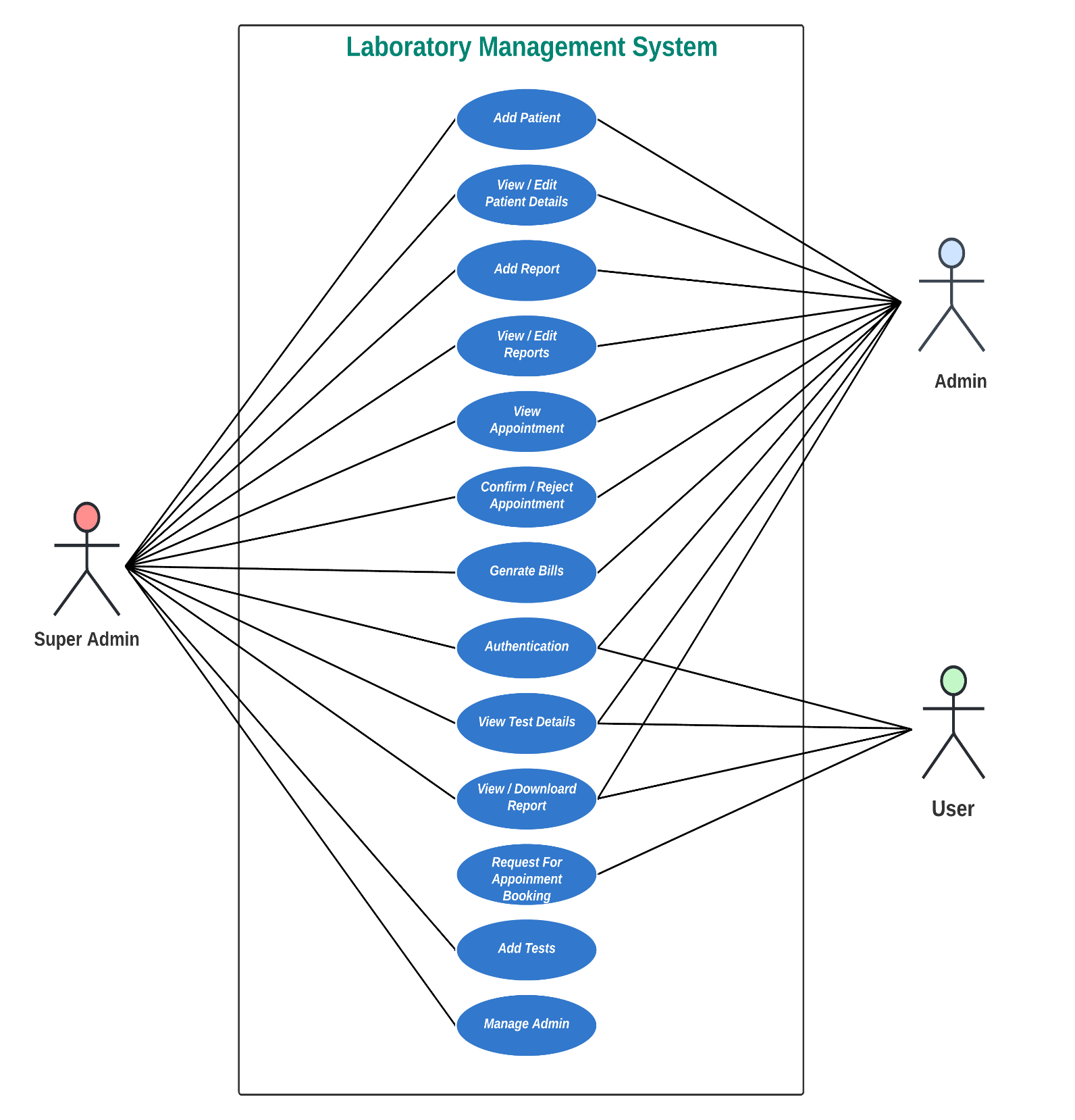
**2.4 User Profile**

|  |  |
| --- | --- |
| Super Admin | Manage patients, Manage Reports, Manage Tests,  Manage Appointments (Confirm/Reject Appointments),  Manage Admin,  Generate Bills, Reports |
| Admin | Manage patients, Manage Reports, Manage Appointments (Confirm/Reject), View Tests, Generate Bills |
| User | Book their Appointments, View/Download the Reports |

**2.5 Technical Details**

|  |  |
| --- | --- |
| Front End | React-Js, JSX, Swagger, CSS , JavaScript, Bootstrap, Material-UI |
| Back End | Node-Js, Express-Js Postman, Prisma(ORM), PostgreSQL |
| OS | Windows 11 |
| Web Server | Node-Js,  Express-Js(Nodemon) |
| Browsers | Chrome, Brave |
| API’s | Node-Js , Express-Js |
| Tools | VS Code, Postman,  Git & GitHub |

**2.6 Use Case Diagram**



**Platform/**

**Environment**

**3.1 SPECIFICATION OF H/W & S/W USED**

3.1.1. SOFTWARE INTERFACE (FOR

DEVELOPMENT)

|  |  |
| --- | --- |
| Software | Chrome & Brave |
| Operating System | Windows -11 |
| Server | Node-Js , Express-Js |
| Development tool | JavaScript  (Node-Js , Express-Js) |
| Back End | DBeaver, pgAdmin4 |

FOR CLIENT SIDE:

* Software Required \_> Any Web Browser

3.2 HARDWARE AND SOFTWARE DETAILS

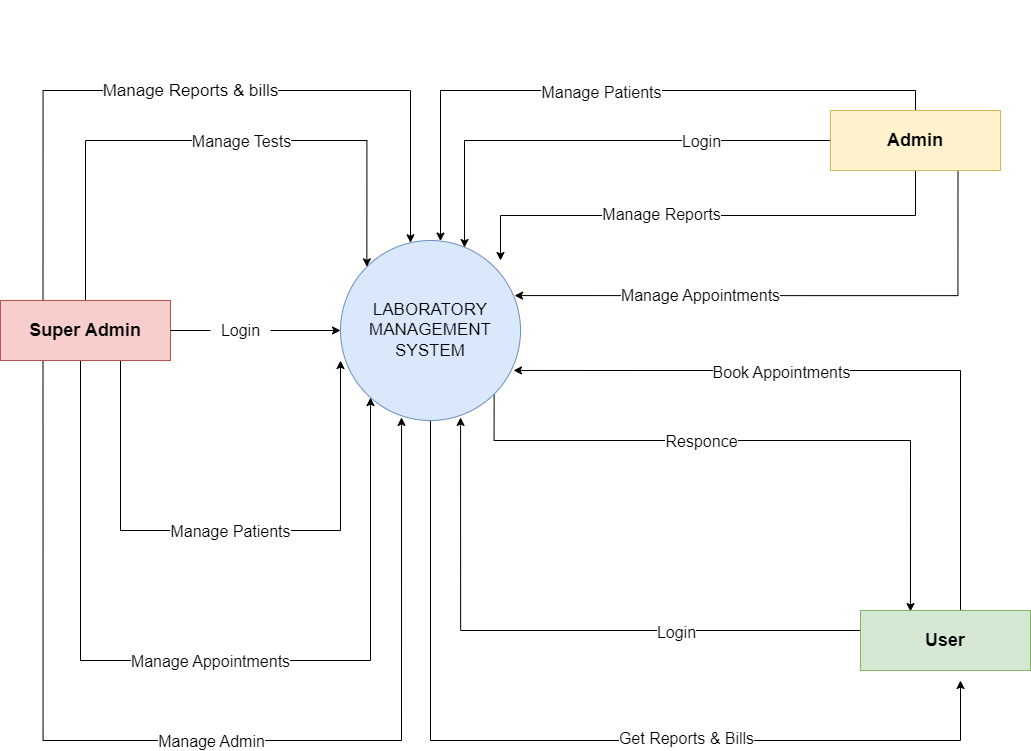
OF SYSTEM USED

|  |  |
| --- | --- |
| Operating System | Windows -11 |
| Front End | React-Js, JSX, CSS, JavaScript, Bootstrap |
| Back End | Node-Js, Express-Js, Postman, PostgreSQL |
| Ram | 8 GB |
| Browser | Brave, Chrome |

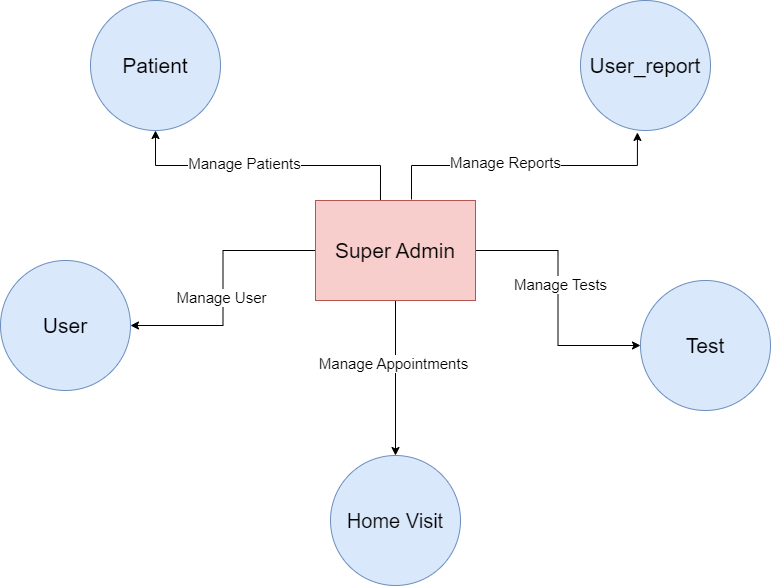
**Structural**

**Analysis**

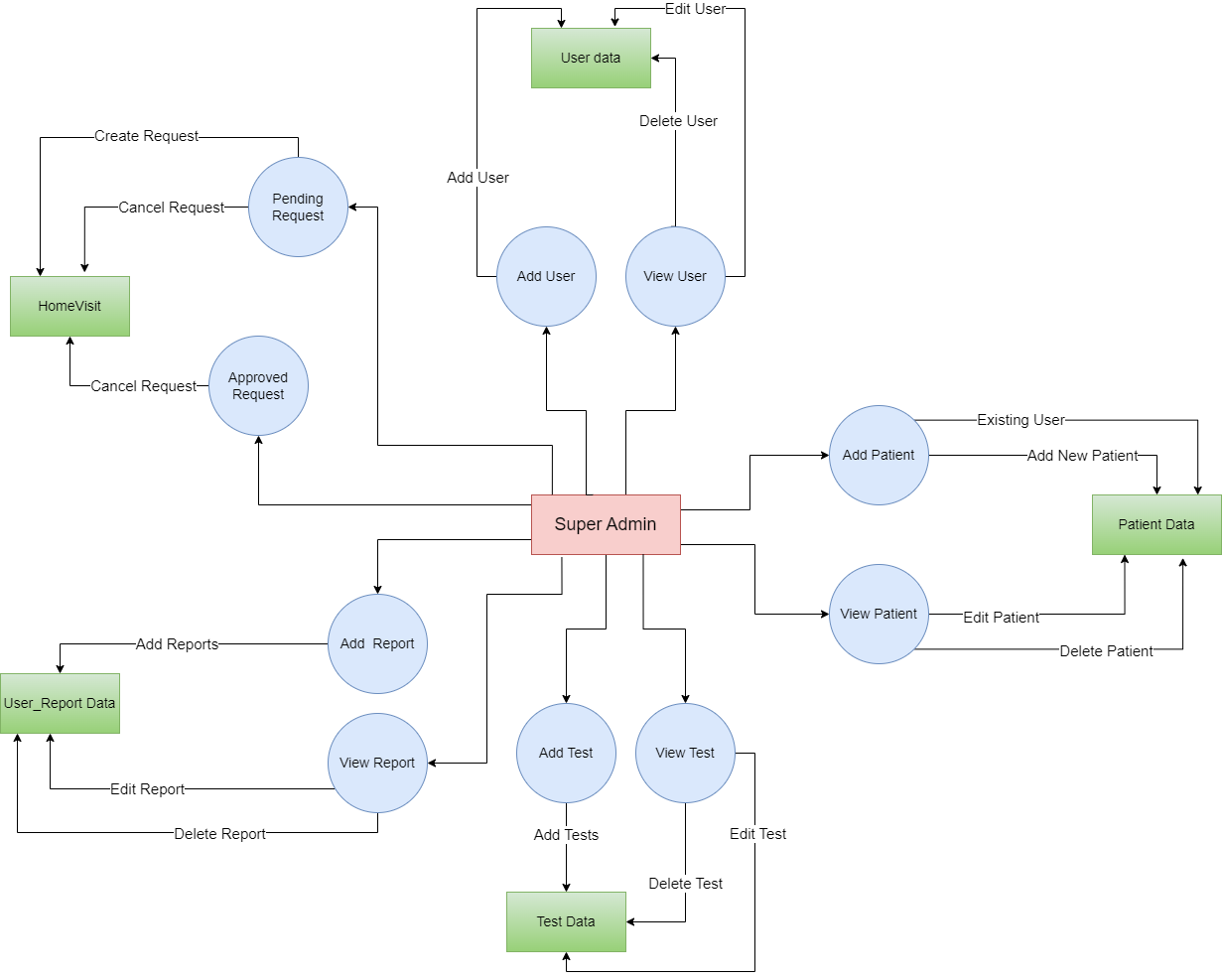
**Level 0 – Data Flow Diagram**



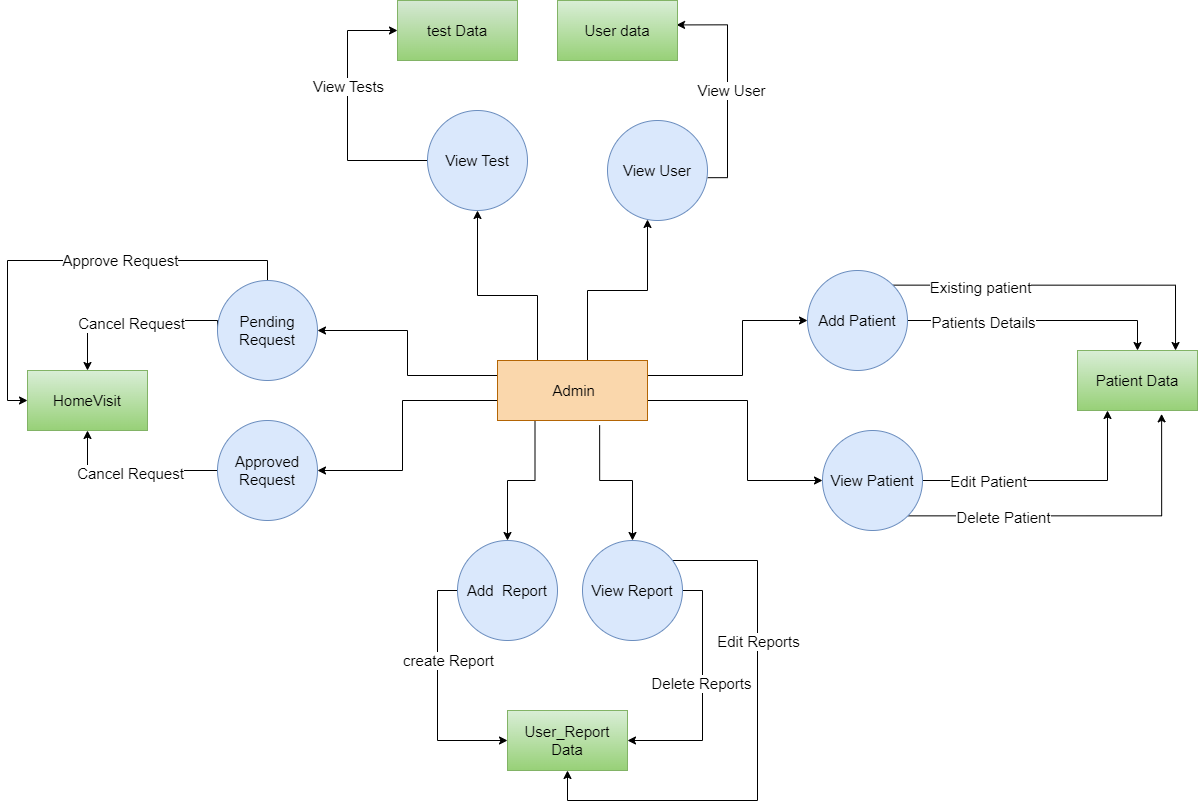
**Super Admin Level 1 – Data Flow Diagram**



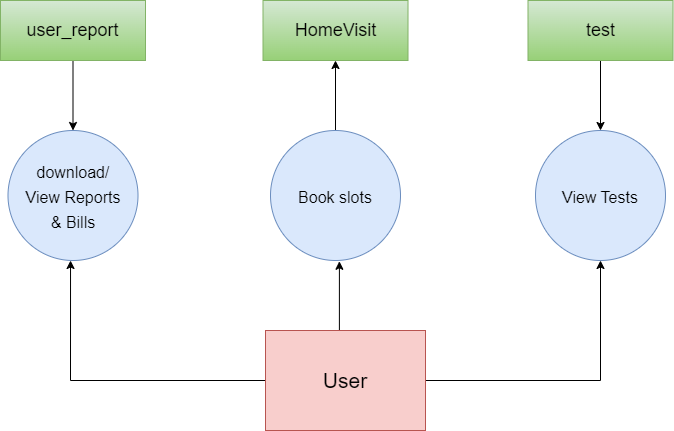
**Super Admin Level 2 – Data Flow Diagram**



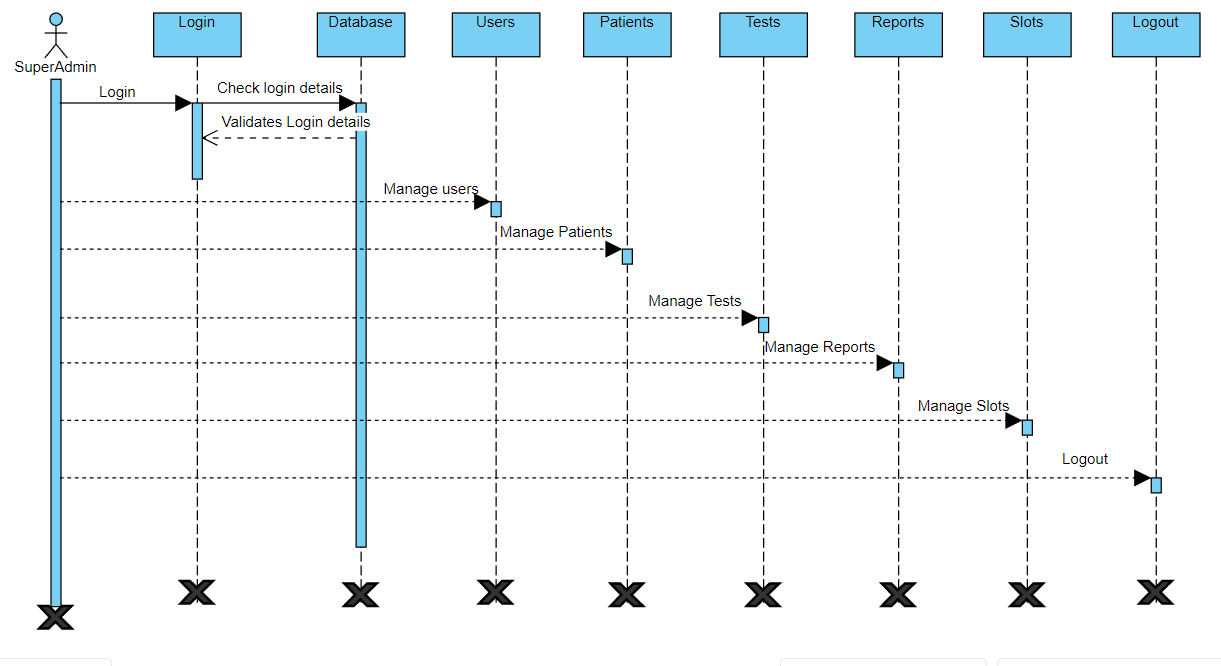
**Admin Level 1 – Data Flow Diagram**

****

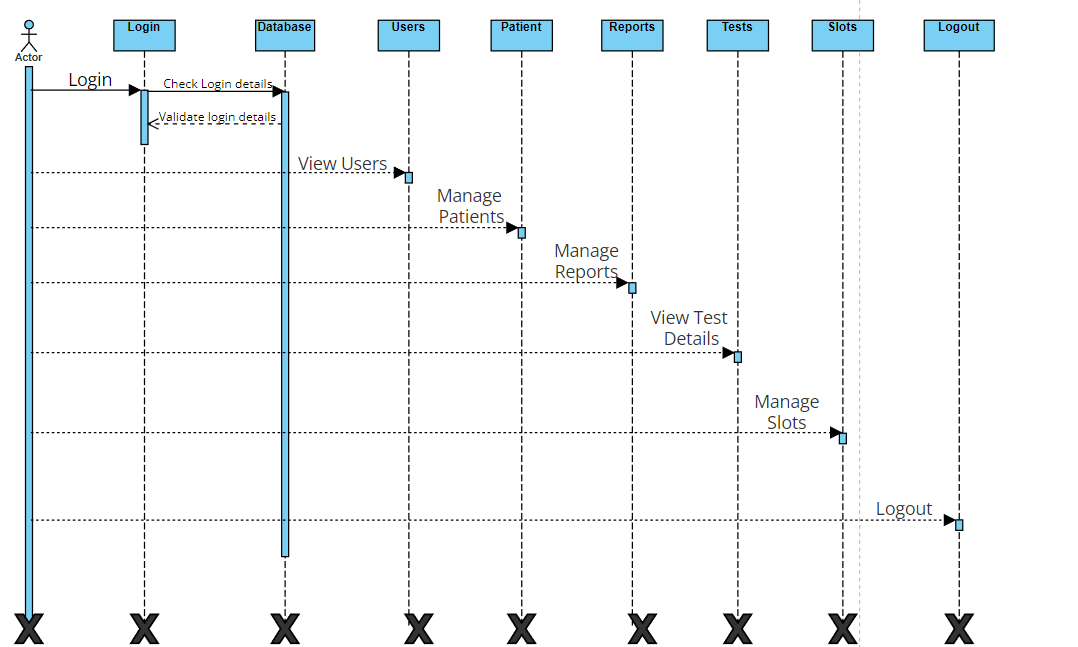
**User Level 1 – Data Flow Diagram**



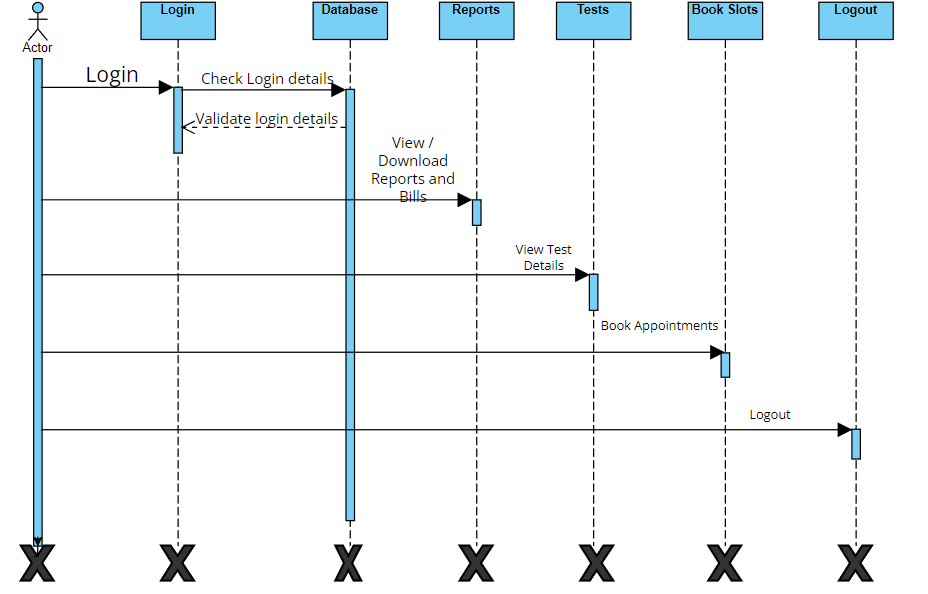
**Sequence Diagram of Super Admin**

****

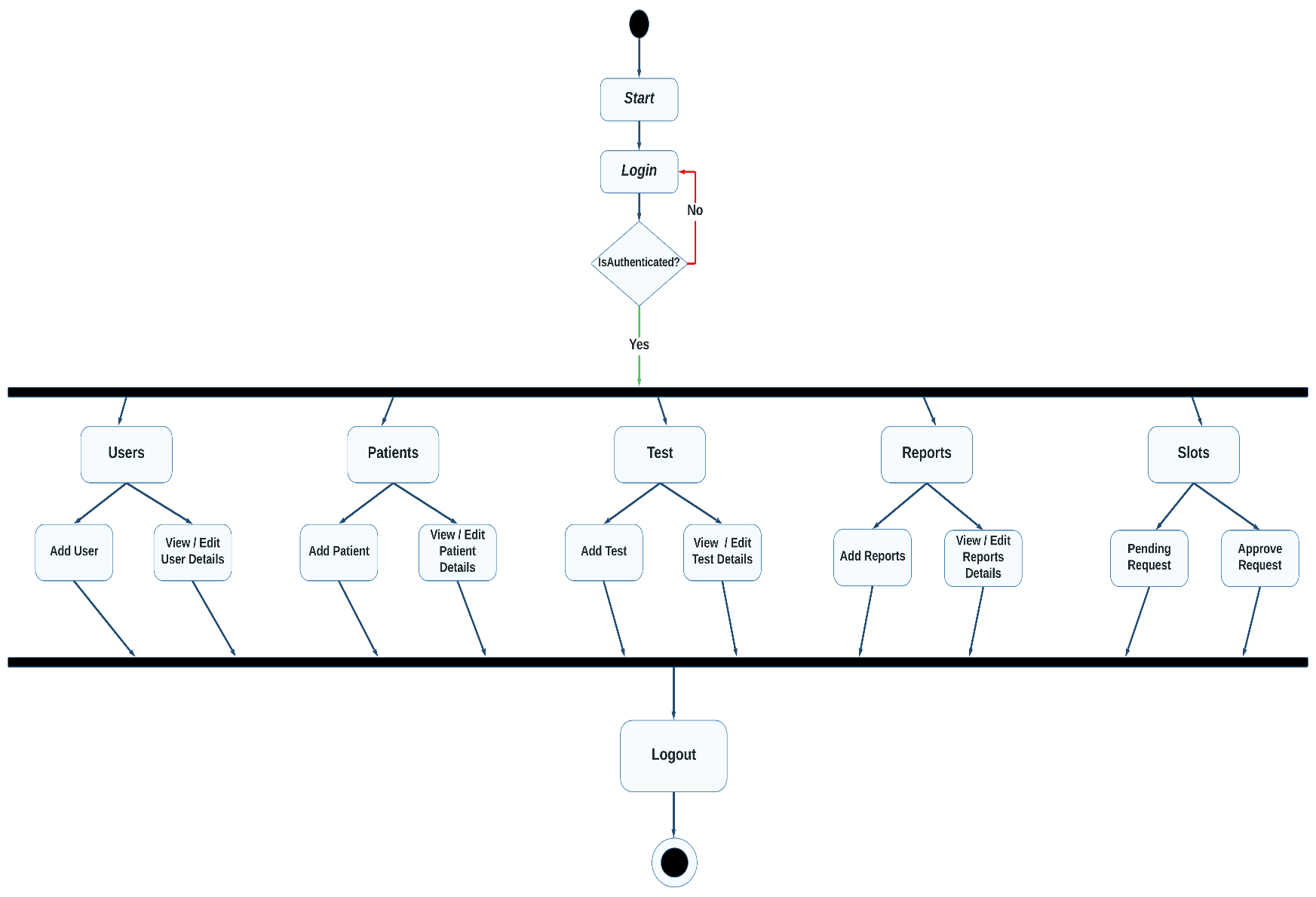
**Sequence Diagram of Admin**

****

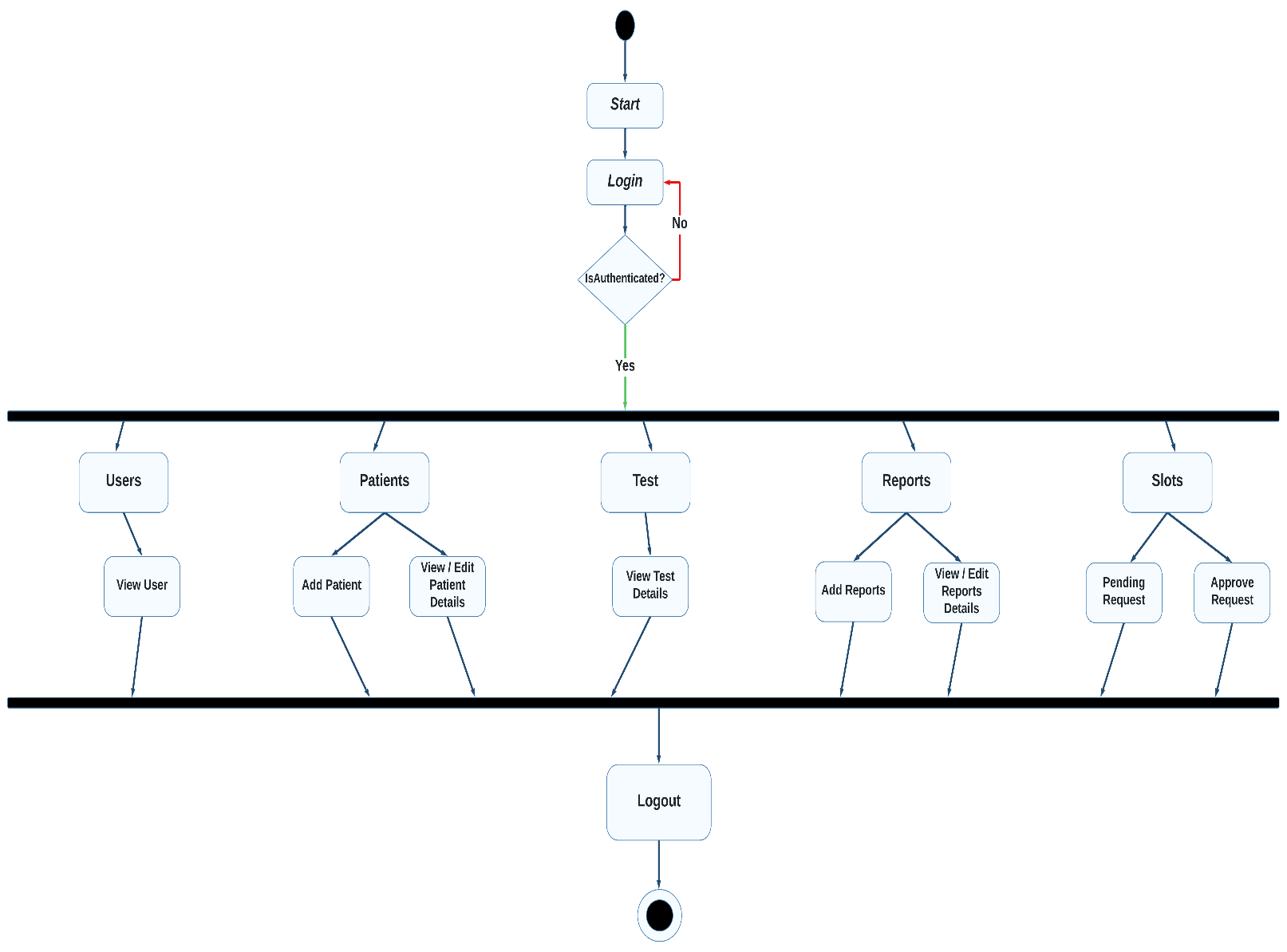
**Sequence Diagram of User**

****

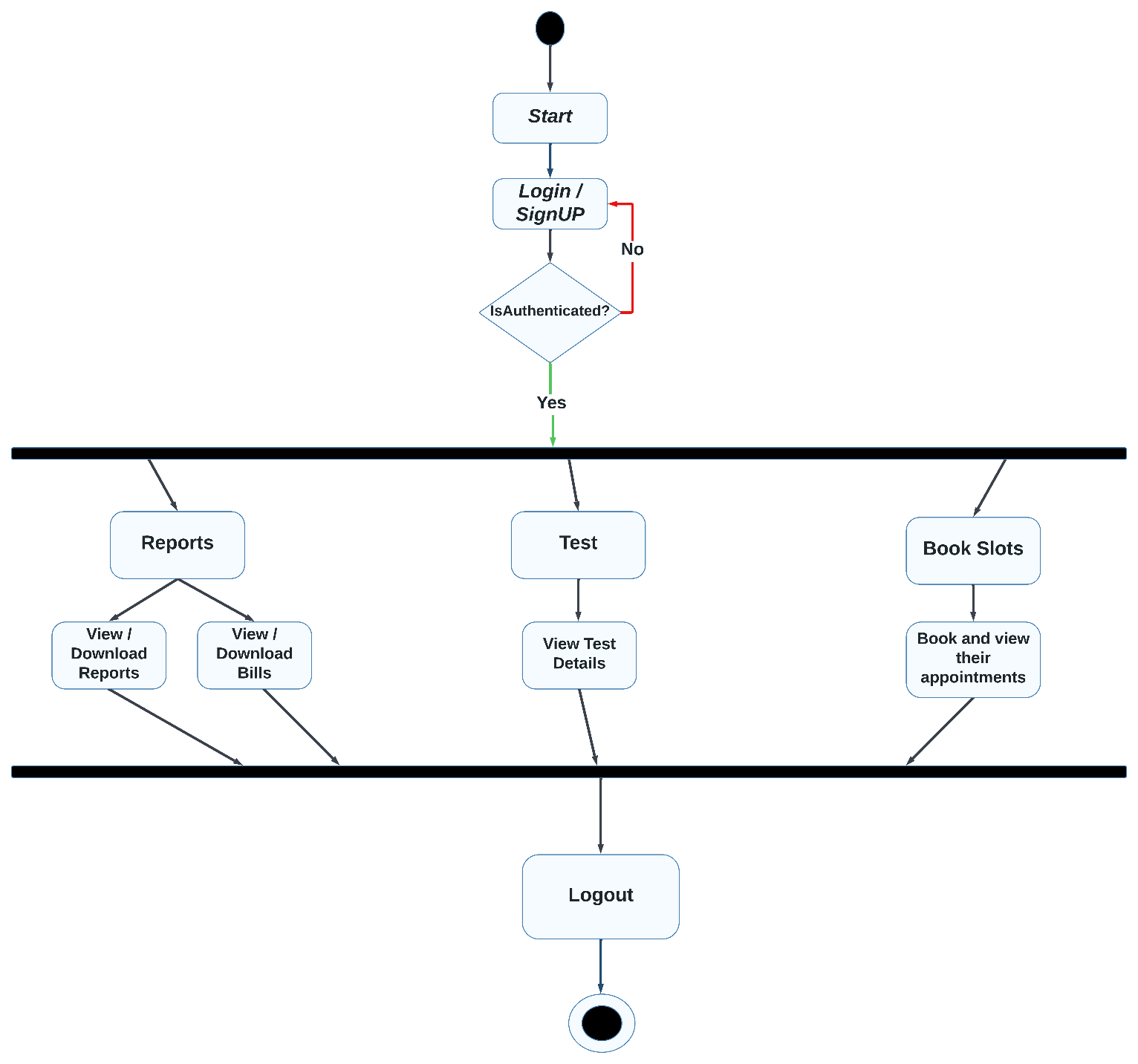
**Activity Diagram of Super Admin**

****

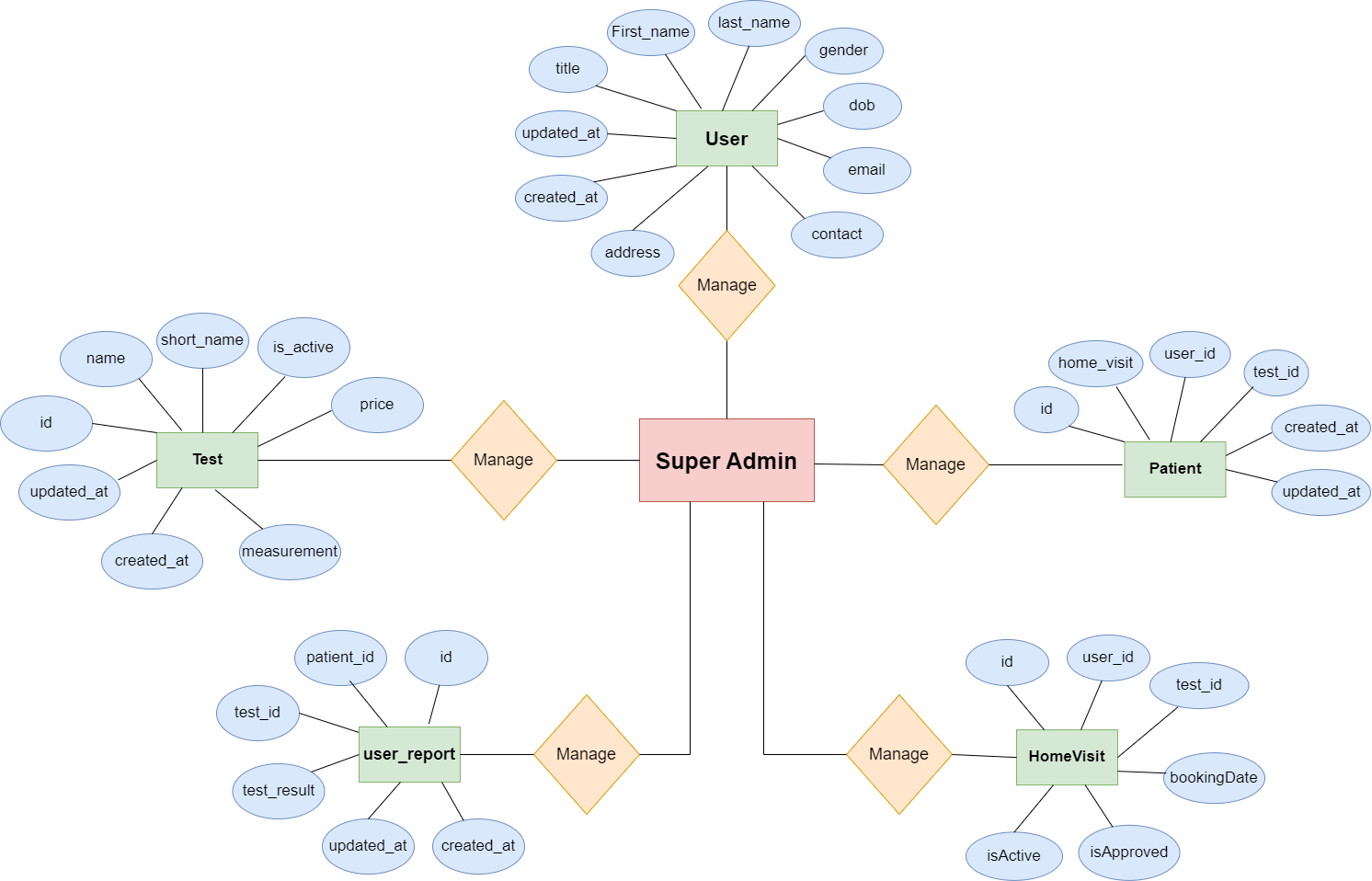
**Activity Diagram of Admin**

****

**Activity Diagram of User**



**Er Diagram**

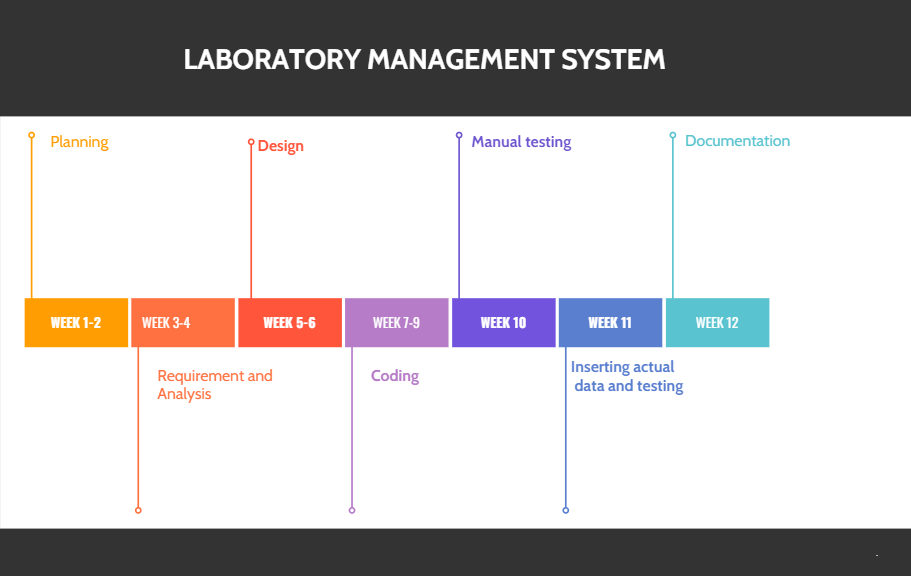


**Relationship Between Tables**

Diagram

Description automatically generated

**TimeLine Chart**

****

**Database Design**

Role:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr No | Column Name | Datatype | Constraint | Description |
| 1 | Key  id | Int | Auto Inc/  Not null | PK of the role table. |
| 2 | role | String | \_ | Role of the user  (E.g. Super Admin / Admin / User) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr No | Column name | Datatype | Constraint | Description |
| 1 | Keyid | int | Auto inc /  Not null | PK of the table |
| 2 | name | varchar | \_ | Full Name of the test |
| 3 | short\_name | varchar | \_ | Short name of test |
| 4 | is\_active | boolean | \_ | Test is active or not |
| 5 | price | float | \_ | Price of the test |
| 6 | measurement | Json | \_ | Measurements |
| 7 | created\_at | Datetime | By default | Creation date with current time stamp |
| 8 | updated\_at | Datetime | By default | date when updated |

Test:

Permission:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr No | Column Name | Datatype | Constraint | Description |
| 1 | Key  Id | Int | Auto Inc/  Not null | PK of the Permission Table. |
| 2 | permissionName | String | \_ | Name of the Permission |
| 3 | role\_id | Int | \_ | FK of role table |

User\_report:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr No | Column name | Data type | Constraint | Description |
| 1 | Keyid | Int | Auto inc /  Not null | PK |
| 2 | patient\_id | Int | \_ | FK from patient |
| 3 | test\_id | Int | \_ | FK from tests |
| 4 | testResult | JSON | \_ | Result data |
| 5 | created\_at | Datetime | By default | Creation date with current time stamp |
| 6 | updated\_at | Datetime | By default | Date when updated |

User:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr No | Column Name | Datatype | Constraint | Description |
| 1 | Keyid | Int | Auto Inc/  Not null | PK of the user table |
| 2 | role\_id | Int | \_ | Fk of role table |
| 3 | title | String | \_ | Title of the user  (E.g. mr , ms , mrs) |
| 4 | first\_name | String | \_ | FirstName of the user |
| 5 | lastName | String | \_ | LastName of the user |
| 6 | gender | String | \_ | Gender of the user |
| 7 | dob | DateTime | \_ | Date of birth |
| 8 | email | String | \_ | Email of the user |
| 9 | contact | String | \_ | Contact number of the user |
| 10 | address | String | \_ | Address of user |
| 11 | created\_at | Datetime | By default | Creation date with current time stamp |
| 12 | updated\_at | Datetime | By default | Date & time when updated |

Patient:

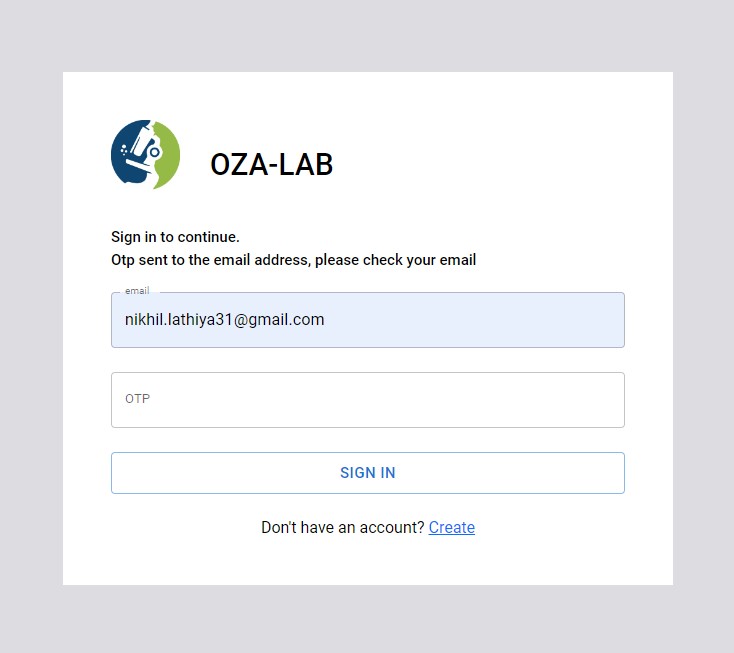
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr No | Column Name | Datatype | Constraint | Description |
| 1 | Keyid | Int | Auto inc/  Not null | PK of the table |
| 2 | user\_id | Int | \_ | Fk from user id |
| 3 | home\_visit | Boolean | \_ | Home visit [yes / no] |
| 4 | test\_id | Int | \_ | Fk from test id |
| 5 | created\_at | Datetime | By default | Creation date with current time stamp |
| 6 | updated\_at | Datetime | By default | date when updated |

Home\_visit:

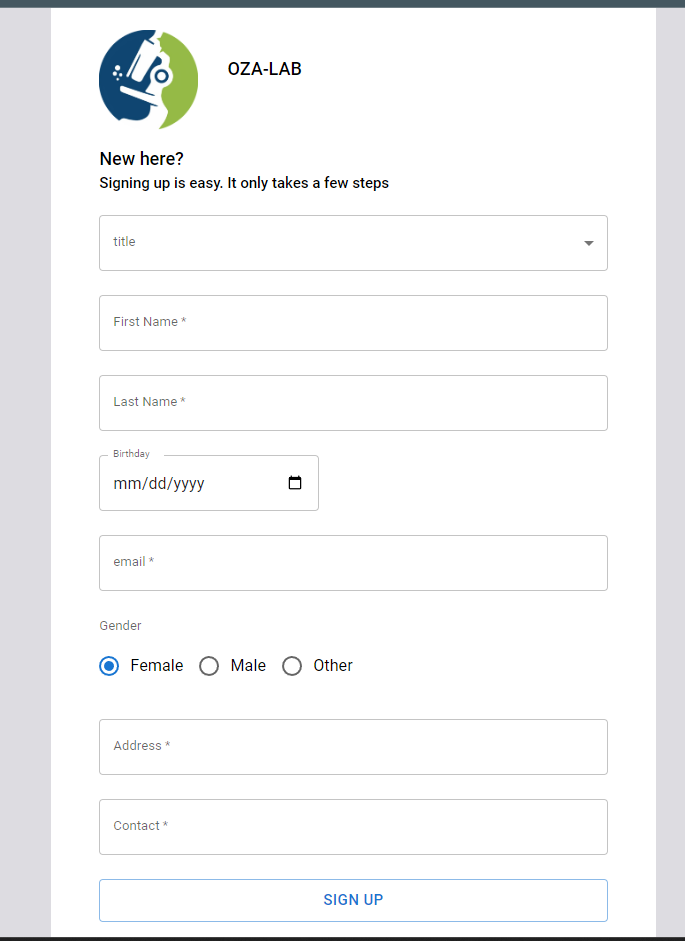
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr No | Column Name | Data type | constraint | Description |
| 1 | Keyid | Int | Auto inc /  Not null | PK |
| 2 | user\_id | Int | \_ | FK from patient |
| 3 | test\_id | Int | \_ | FK from tests |
| 4 | bookingDate | JSON | \_ | Result data |
| 5 | isApproved | Datetime | By default | Creation date with current time stamp |
| 6 | isActive | Datetime | By default | Date when updated |

**Interface Design**

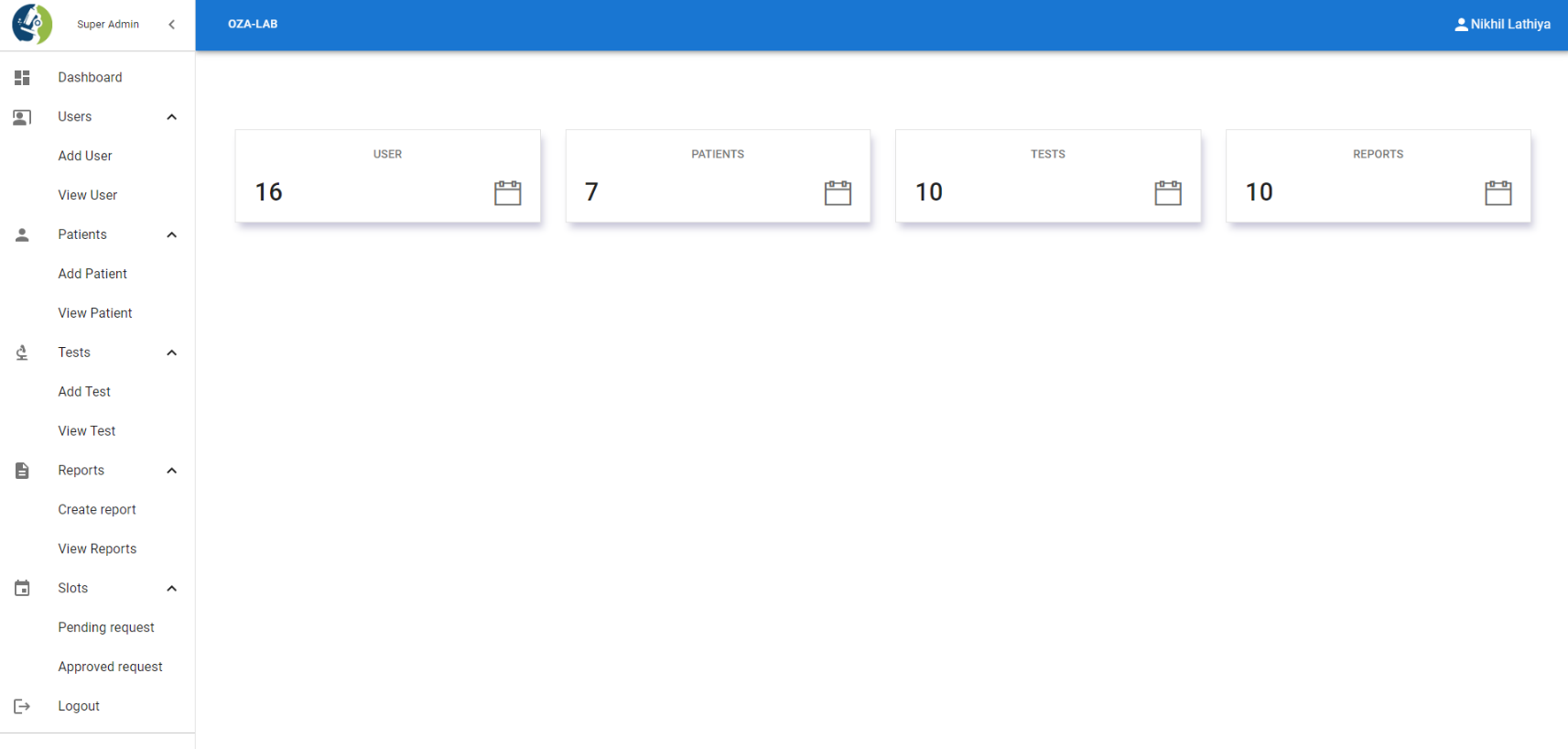
**Login Page**

****

**Sign\_Up Form**



**Dashboard [Super Admin]**



**Add User [Super Admin]**

Graphical user interface, application

Description automatically generated

**View User [Super Admin]**

**Graphical user interface, application

Description automatically generated**

**Edit User**

Graphical user interface, text, application, email

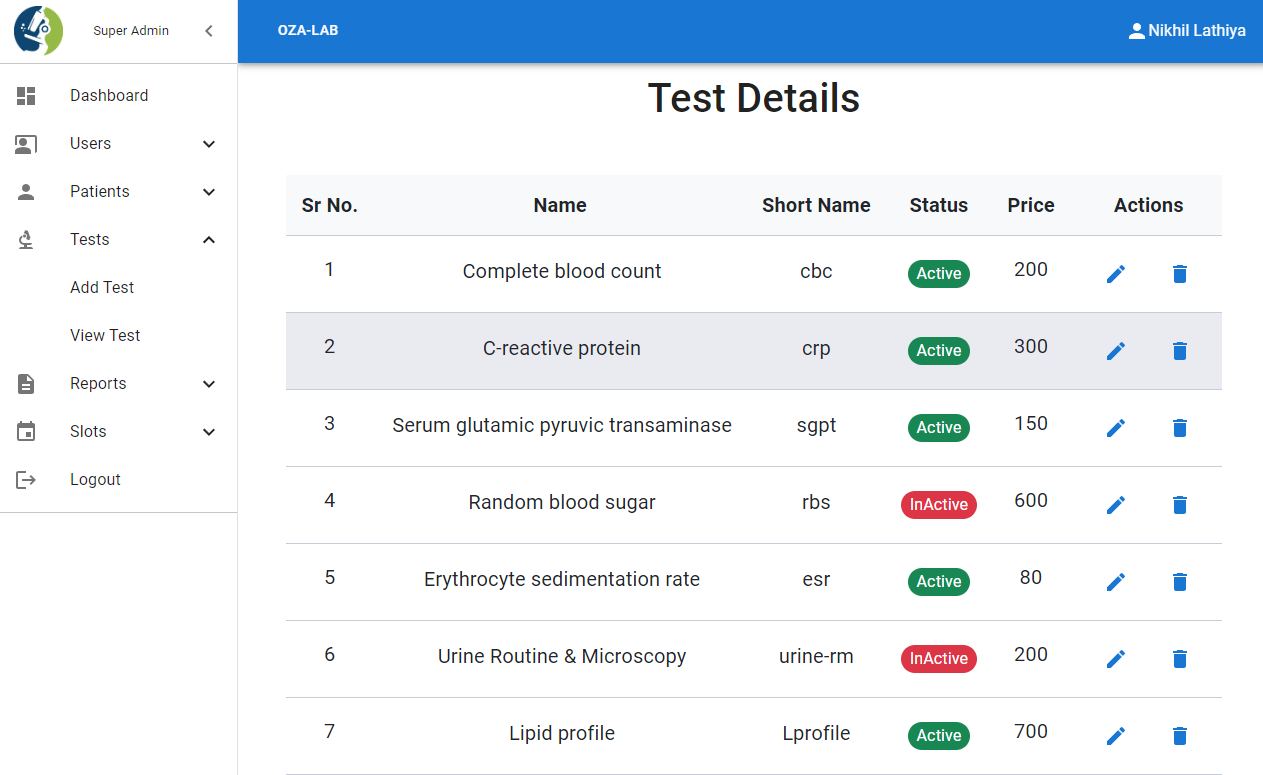
Description automatically generated

**Add Test**

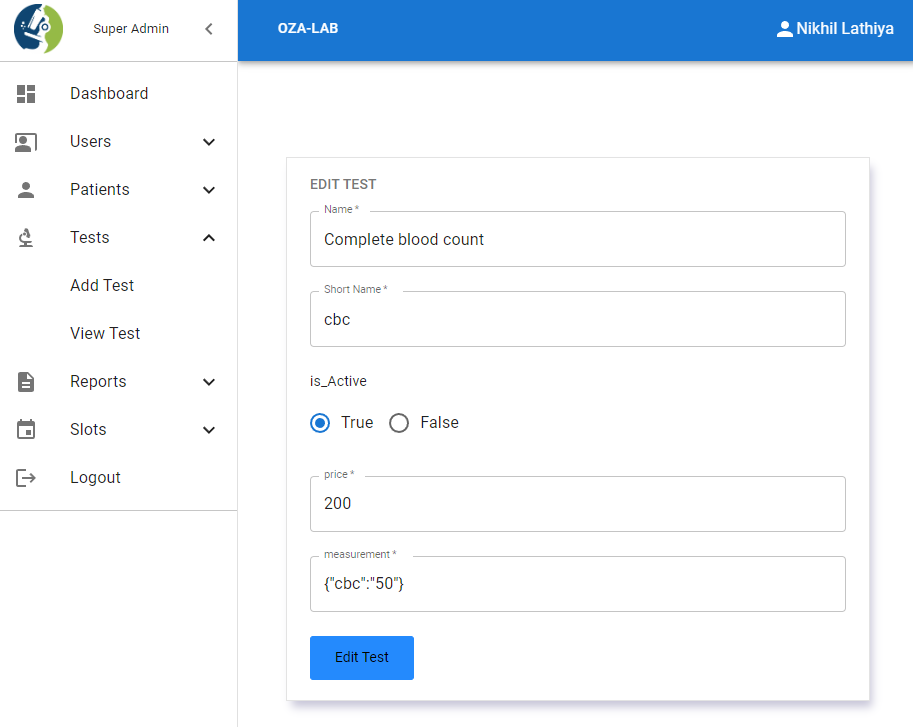
Graphical user interface, application

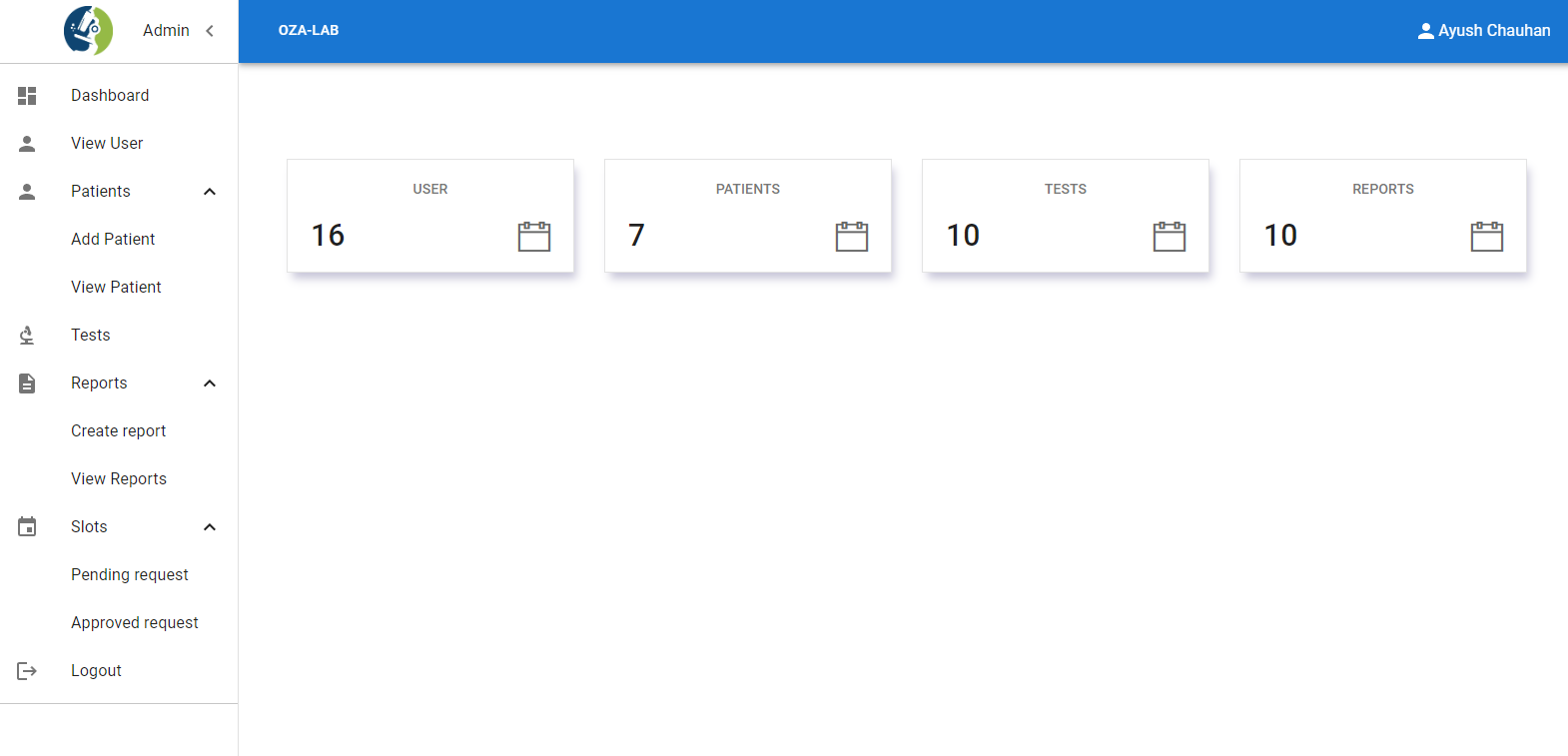
Description automatically generated

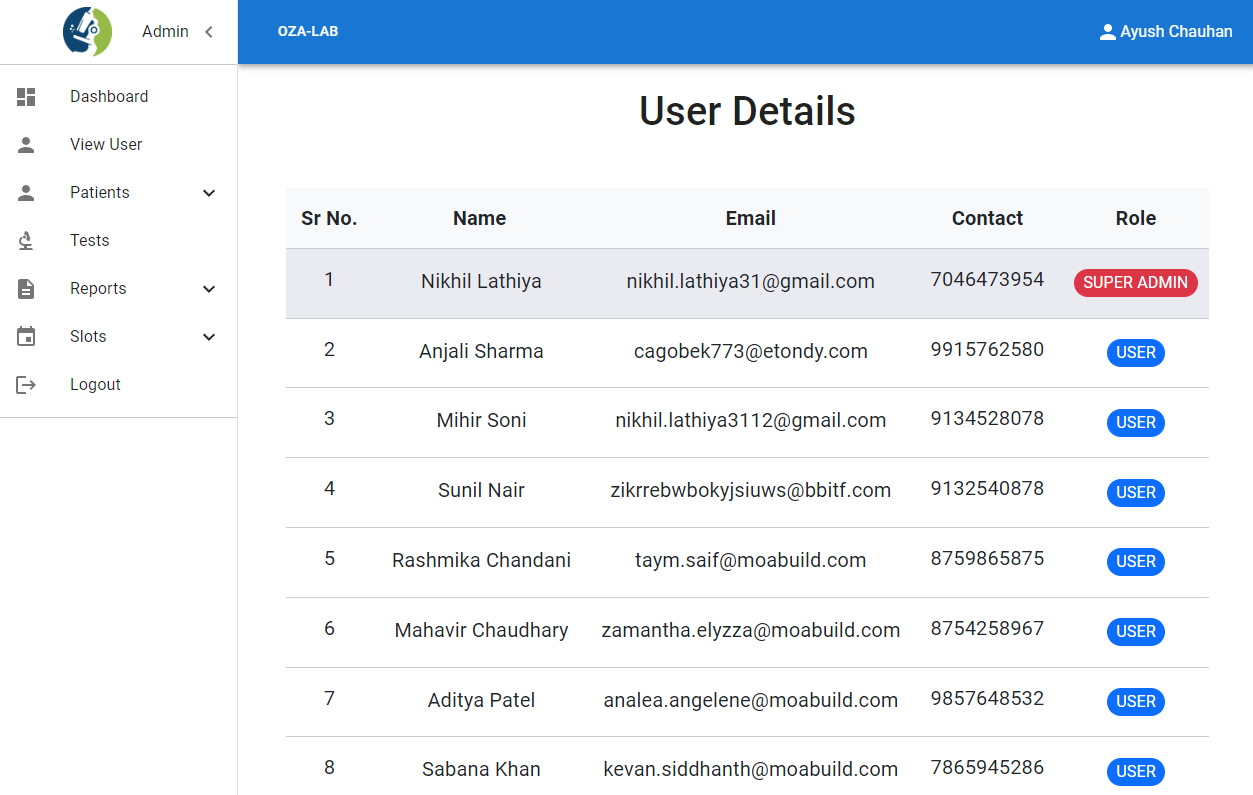
**View Test**

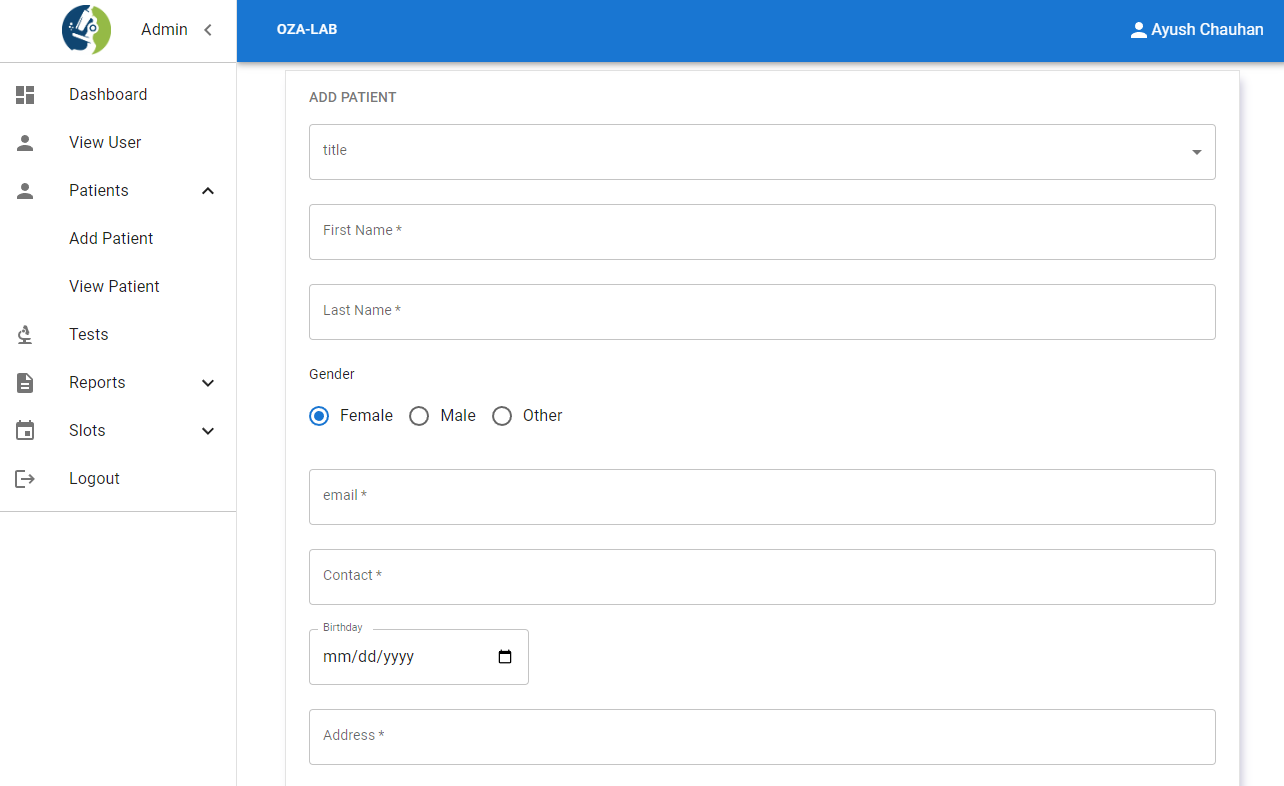


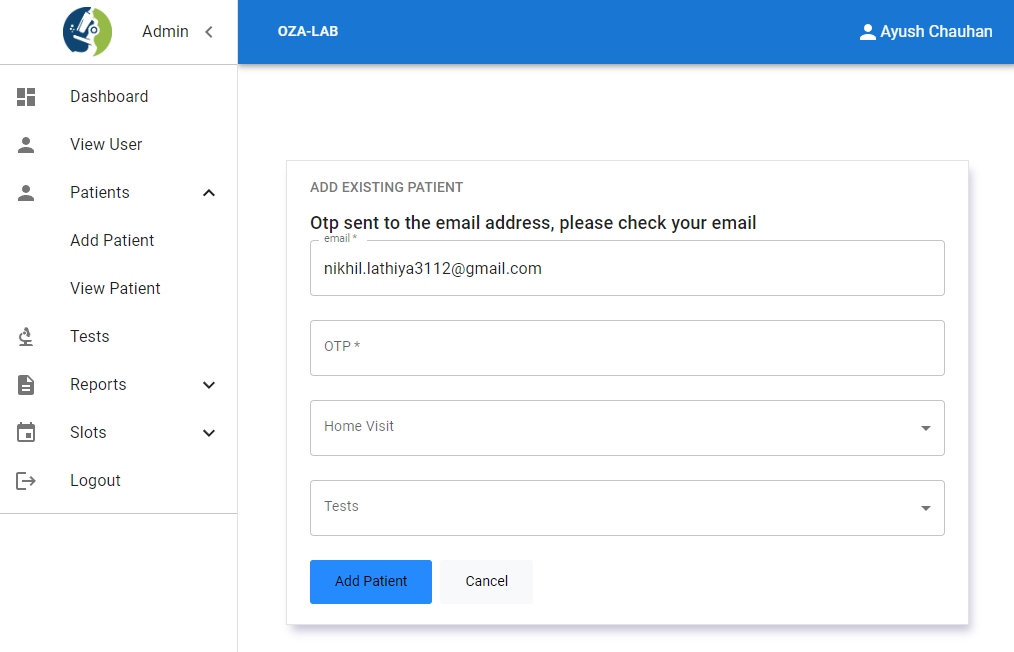
**Edit Test**



 **Dashboard [Admin]**

 **View User [Admin]**

**Add Patient [Admin]**

 **Add Existing patient [Admin]**

Graphical user interface, text, application, email

Description automatically generated **Add User Report**

**\**

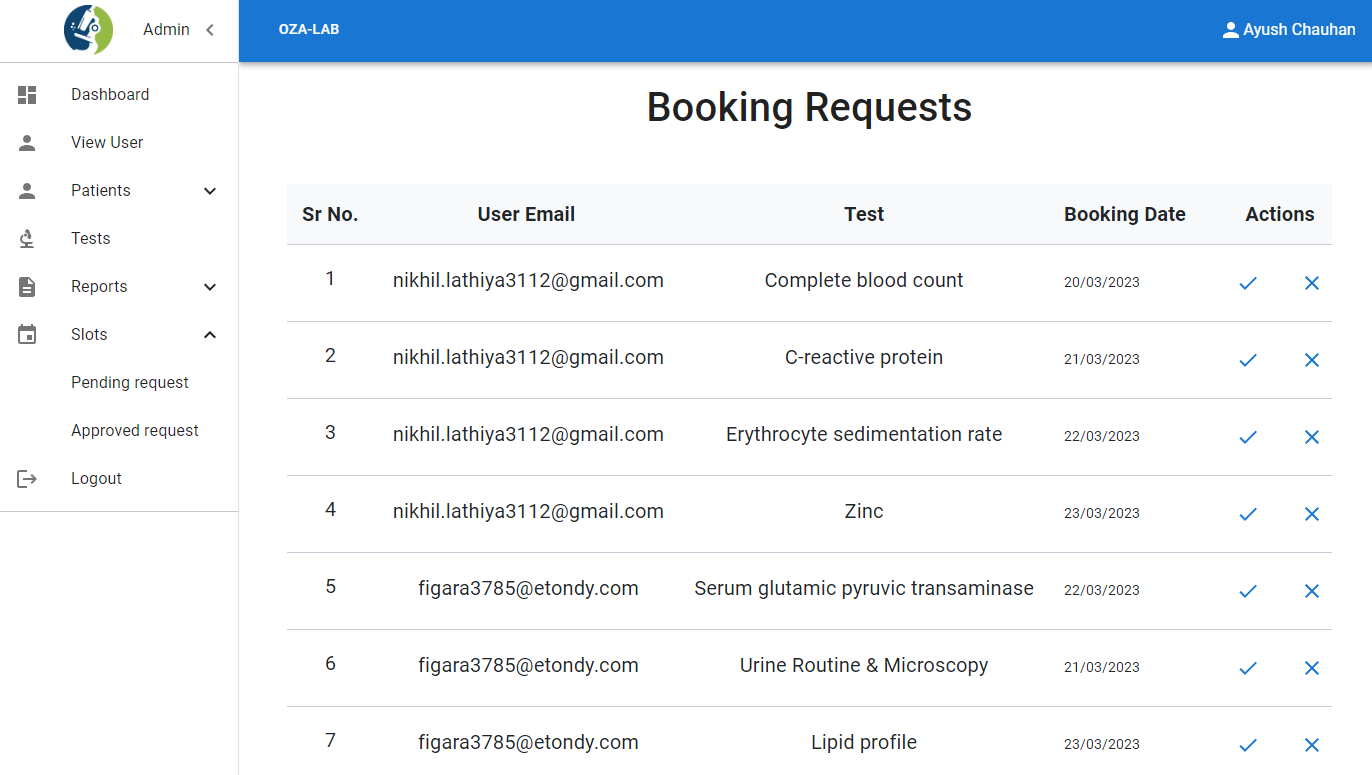
Graphical user interface, application

Description automatically generated **Edit User Report**

**View Reports**

Graphical user interface, table

Description automatically generated

**Pending Request**

**Approved Request**

Graphical user interface, text, application

Description automatically generated

**Dashboard [User]**

Graphical user interface

Description automatically generated with medium confidence

Graphical user interface, text, application, email

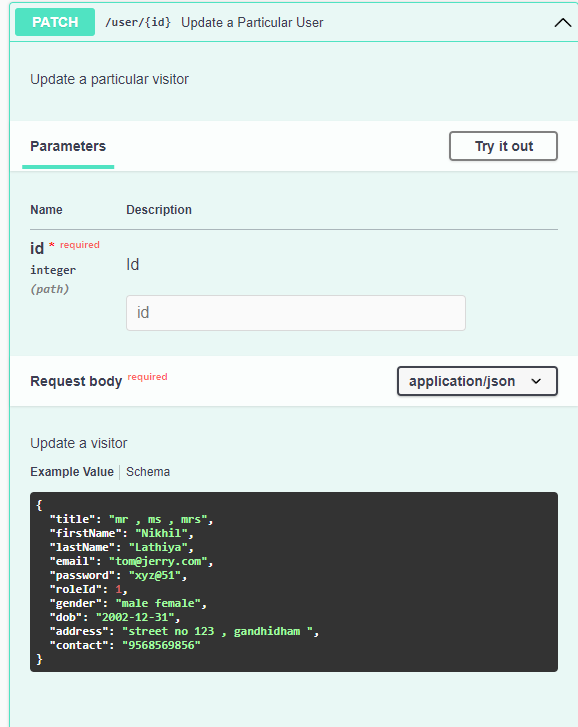
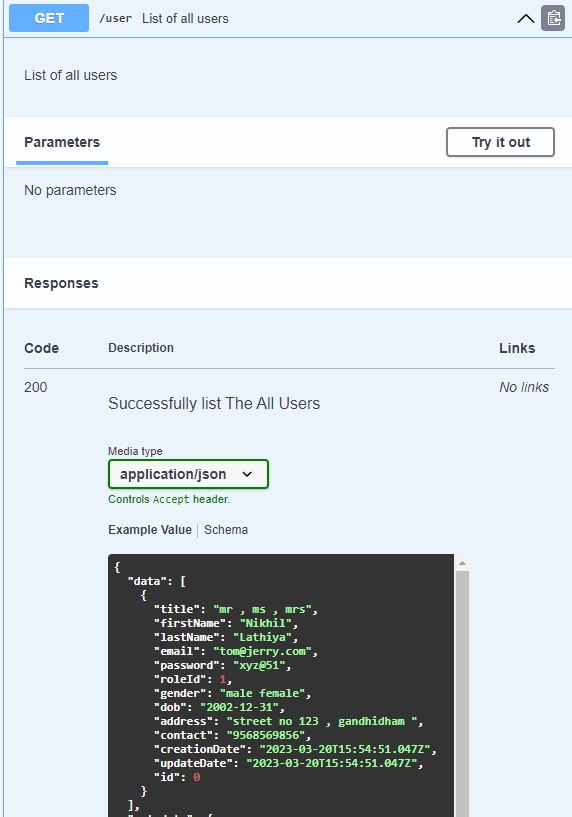
Description automatically generated **Book Slot**

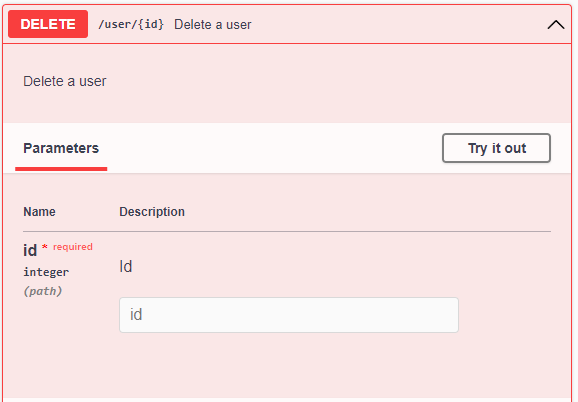
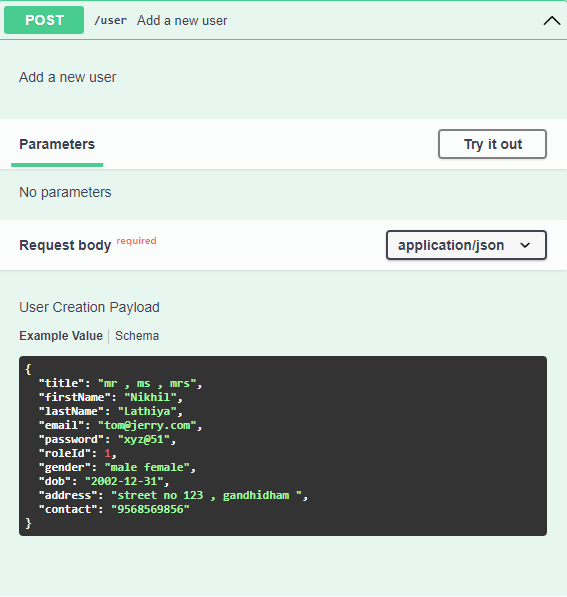
Graphical user interface, application, email

Description automatically generated **Swagger (For API Testing)**

Graphical user interface, application

Description automatically generated



****

Testing

**Unit Testing:**

Unit testing is a type of software testing where individual units or components of the software area tested. All modules are tested properly. The purpose is to validate that each unit of the software code is performing as it is meant to be.

**Interface Testing:**

Interface Testing is defined as a software testing type which verifies whether the communication between two different software system is done correctly. Interface testing is done on a connection that integrates two components called interface.

**Testing Documents:**

Testing documentation is the documentation of artifacts that are created during or before the testing of a software application. Documentation reflects the importance of processes for the customer, individual, and organization. Projects which contain all documents have a high level of maturity.

**Final Testing:**

The Final Test of the project gave some bugs and errors, but I solved it

**Bibliography**

* + - * Postgresql.org
      * Expressjs.com
      * Reactjs**.**org
      * Nodejs.org
      * Prisma.io
      * Swagger**.**io
      * mui.com
      * stackoverflow.com
      * getbootstrap.com