



**INSTITUTE FOR ADVANCED
COMPUTING AND SOFTWARE
DEVELOPMENT AKURDI, PUNE**

Documentation On

AROGYAM

(Doctor Appointment System)

PG-DAC MARCH

2023

Submitted By:

Group No: G-39

Roll No:

233082

233074

Name:

Sarang Rohankar

Roshan Dahaghane

Mr. Rohit Puranik
Centre coordinator

Mrs. Gauri Kadam
Project Guide

ABSTRACT

The "Arogyam: Doctor Appointment System" is a web-based application designed to bridge the gap between patients and doctors, revolutionizing the healthcare industry by streamlining appointment scheduling, enhancing patient-doctor interactions, and promoting efficient healthcare management. The system facilitates seamless communication, reduces waiting times, and provides patients and doctors with a user-friendly platform to manage appointments and healthcare-related tasks.

In a world where technology continues to reshape various sectors, the healthcare industry often faces challenges in ensuring smooth patient-doctor interactions and efficient appointment management. The "Arogyam" project addresses these challenges by introducing a comprehensive online platform that caters to the needs of patients, doctors, and administrators.

The system comprises three distinct modules: the Admin Module, the Doctor Module, and the Patient Module. The Admin Module empowers administrators to oversee and manage doctor profiles, user accounts, and system configurations. The Doctor Module empowers doctors to create and manage their profiles, set availability, view patient appointments, and interact with patients through reviews and feedback. The Patient Module provides patients with the ability to register, search for doctors, book appointments, leave reviews, and manage their medical history.

The project follows a meticulous analysis and design process, ensuring that the system addresses the existing challenges faced by both patients and doctors in the current healthcare landscape. By leveraging modern technologies such as React, Spring Boot, and MySQL, the system offers a dynamic and responsive user interface, robust backend functionality, and efficient data management.

ACKNOWLEDGEMENT

Apart from the efforts of the team, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

The completion of any inter-disciplinary project depends upon cooperation, coordination and combined efforts of several sources of knowledge.

We are eternally grateful to our guide Mrs.Geeta Darunte for her even willingness to give us valuable advice and direction under which we executed this project. Her constant guidance and willingness to share her vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks. . I extend my sincere thanks to our respected Centre Co-ordinator Mr.Rohit Puranik for allowing us to use the facilities available. I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for the support and encouragement they have given me during the course of our work.

Sarang Rohankar (233127)

Roshan Dahaghane(233074)

Table of Contents

ABSTRACT.....	2
ACKNOWLEDGEMENT.....	3
INTRODUCTION.....	6
1.1 PROJECT OBJECTIVE.....	7
1.2 PROJECT OVERVIEW.....	7
1.3 PROJECT SCOPE.....	7
1.4 STUDY OF THE SYSTEM.....	8
1.4.1 MODULES:.....	8
SYSTEM ANALYSIS.....	14
2.1 EXISTING SYSTEM.....	14
2.2 PROPOSED SYSTEM.....	14
2.3 SYSTEM REQUIREMENT SPECIFICATION.....	15
2.3.1 GENERAL DESCRIPTION.....	15
2.3.2 SYSTEM OBJECTIVES.....	17
2.3.3 SYSTEM REQUIREMENTS.....	17
SYSTEM DESIGN.....	20
3.1 INPUT AND OUTPUT DESIGN.....	20
3.1.1 INPUT DESIGN:.....	20
3.1.2 OUTPUT DESIGN:.....	21
DATABASE DESIGN.....	21
3.2 DATABASE.....	21
3.3 SYSTEM TOOLS.....	21
3.3.1 FRONT END:.....	21
3.3.2 BACKEND:.....	22
TABLE STRUCTURE:.....	27
CONCLUSION.....	37
REFERENCES.....	38

LIST OF FIGURES

FIGURE 1: ADMIN ACTIVITY DIAGRAM	09
FIGURE 2: USER ACTIVITY DIAGRAM	11
FIGURE 3: PROFESSIONAL ACTIVITY DIAGRAM	13
FIGURE 4: DFD DIAGRAM	23
FIGURE 4: SEQUENCE DIAGRAM	24
FIGURE 5: E-R DIAGRAM	25
FIGURE 6: CLASS DIAGRAM	27
FIGURE 7: TABLE STRUCTURE	28
FIGURE 8: PROJECT DIAGRAMS	31

INTRODUCTION

"Arogyam" is a sophisticated web-based application designed to revolutionize the process of scheduling and managing medical appointments within healthcare facilities. The application aims to provide a seamless platform that connects administrators, doctors, and patients, allowing them to efficiently coordinate appointments, manage doctor approvals, and facilitate patient reviews. By leveraging cutting-edge technology, "Arogyam" seeks to streamline the appointment process, enhance patient-doctor interactions, and ultimately improve the overall healthcare experience.

The Arogyam(Doctor Appointment System) is a comprehensive web-based application designed to streamline the management of medical appointments and reviews within a healthcare facility. The system aims to provide efficient and user-friendly tools for administrators, doctors, and patients to interact and manage various aspects of the medical appointment process. By automating appointment scheduling, doctor approvals, and patient feedback, the system aims to enhance the overall experience for both medical professionals and patients.

In the traditional healthcare setting, manual appointment scheduling processes often lead to scheduling conflicts, miscommunications, and extended waiting times for patients. Furthermore, the absence of an integrated platform for managing doctor approvals and patient feedback impedes the facility's ability to enhance its services based on user experiences.

1.1 PROJECT OBJECTIVE

The main objectives of the "Arogyam" project are:

- To provide a user-friendly and accessible platform for patients to book appointments with doctors.
- To enable doctors to manage their schedules, view patient appointments, and respond to patient requests.
- To allow patients to leave reviews and feedback for doctors after their appointments.
- To enhance the overall healthcare experience by reducing waiting times and improving communication between patients and doctors.
- To provide an organized and efficient system for both patients and doctors to manage appointments and healthcare-related tasks.

1.2 PROJECT OVERVIEW

"Arogyam" is a web-based Doctor Appointment System designed to streamline medical appointment scheduling and management. This platform aims to connect administrators, doctors, and patients through a user-friendly interface, enhancing the efficiency of appointment coordination, doctor approvals, and patient reviews. By leveraging modern technology, "Arogyam" seeks to revolutionize healthcare service delivery and improve patient experiences.

1.3 PROJECT SCOPE

"Arogyam" is a web-based Doctor Appointment System designed to streamline medical appointment scheduling and management. This platform aims to connect administrators, doctors, and patients through a user-friendly interface, enhancing the efficiency of appointment coordination, doctor approvals, and patient reviews. By leveraging modern technology, "Arogyam" seeks to revolutionize healthcare service delivery and improve patient experiences.

1.4 STUDY OF THE SYSTEM

1.4.1 MODULES:

1. Admin Module:

- Approve and manage doctor profiles.
- Monitor and manage user accounts.
- Handle system-wide configurations.
- Generate reports and insights about system usage.

2. Doctor Module:

- Create and manage their profiles, including personal details and specialization.
- Set availability and manage appointment slots.
- View patient appointments and access their medical history.
- Receive notifications of appointment bookings.
- Respond to patient reviews and feedback.

3. Patient Module:

- Register and create accounts.
- Search for doctors based on specialization, location, and availability.
- Book appointments with preferred doctors.
- View appointment details, including date, time, and doctor information.
- Leave reviews and feedback for doctors.

1.4.1.1 Administrator:

DESCRIPTION - The admin can add Doctor, update Doctor status, create/update/delete Doctor.

MAIN FLOW OF EVENTS

1. Admin logs in the system.
2. Admin can add doctor.
3. Admin maintains doctor record.
 - 3.1 List of registered doctor details is displayed.
4. Admin View the blood services available and also can manage them.
5. Admin can add doctor and manage and see them in a list form.

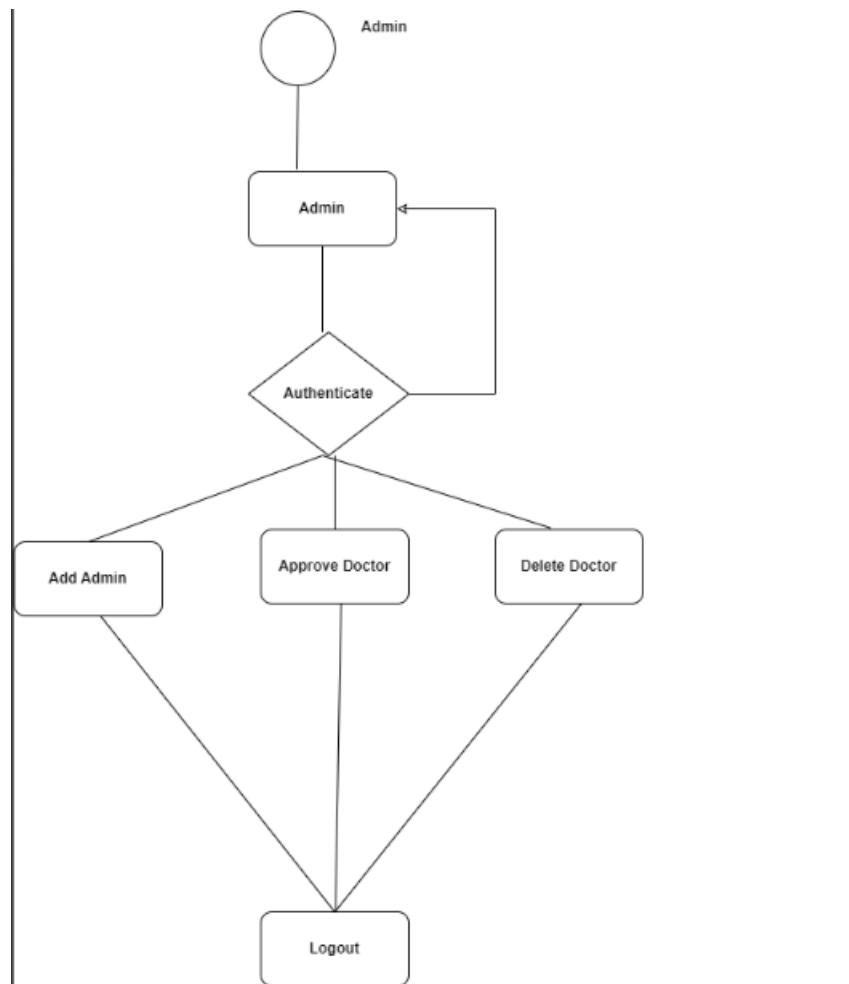


Figure 1 Admin Activity Diagram

➤ **Add Doctor**

Admin can add new professionals and register him with details like name, Email, age, gender and its number etc.

➤ **Edit Doctor**

Admin will have a list view of all the existing Doctor. He can also edit for a particular Doctor.

➤ **Add/Remove Doctor**

Admin can add /remove Doctor.

➤ **View Doctor**

Admin will have a dashboard where he/she can view all the Doctor.

1.4.1.2 Patient:

MAIN FLOW OF EVENTS

1. Patient can sign up in the system.
2. Patient can sign up by filling their details like name, Email, age, mobile number , etc.
3. Patient can sign in to the application and manage their profile.
4. Patient can view a list of all appointments.
5. Patient can update and cancel their appointments (bookings).

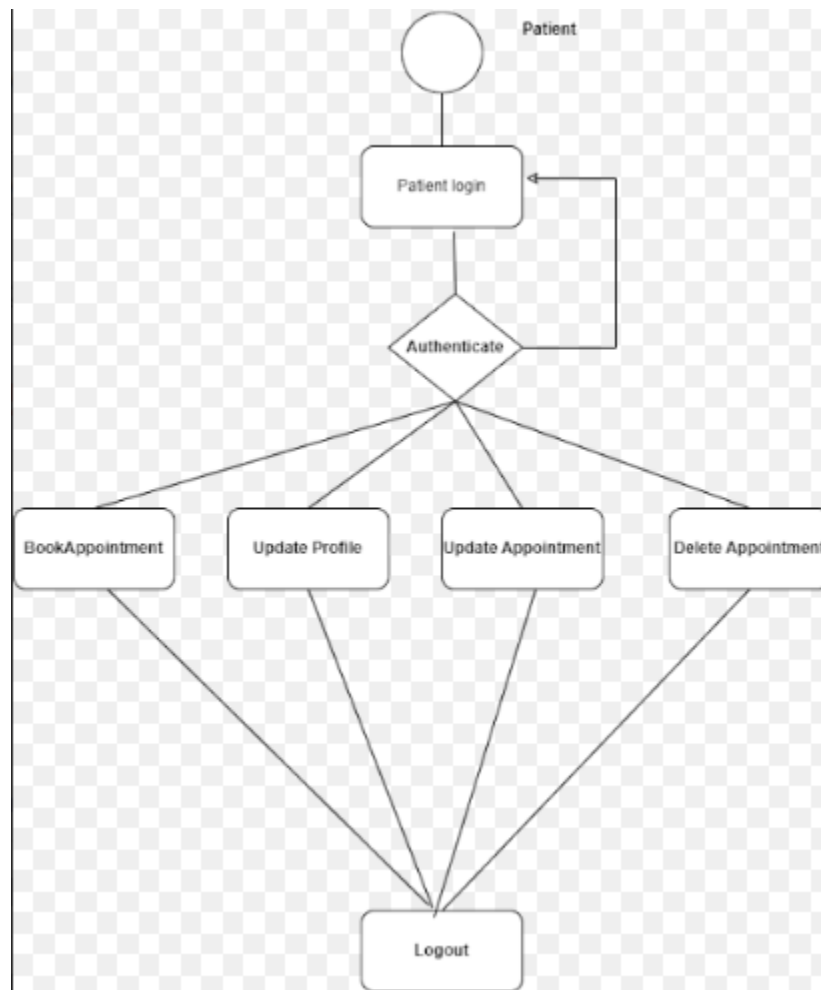


Figure 2 Patient's Activity Diagram

➤ **Create Account**

Patient can create an account by filing proper details.

➤ **List All Appointments**

Patient will have a list view of all the appointments of him/her.

➤ **Manage Profile**

Patient can manage their profile and can update their details.

➤ **Book Appointments**

Patient can book an appointment with the professional according to their preference.

1.4.1.3 Doctors:

MAIN FLOW OF EVENTS

1. Doctors can register with correct details.
2. Doctors can sign in by using their email and password.
3. Doctors can edit their profiles.
4. Doctors can manage their bookings and appointments.

➤ **Create Account**

Doctors can create an account by filing proper details.

➤ **Manage Profile**

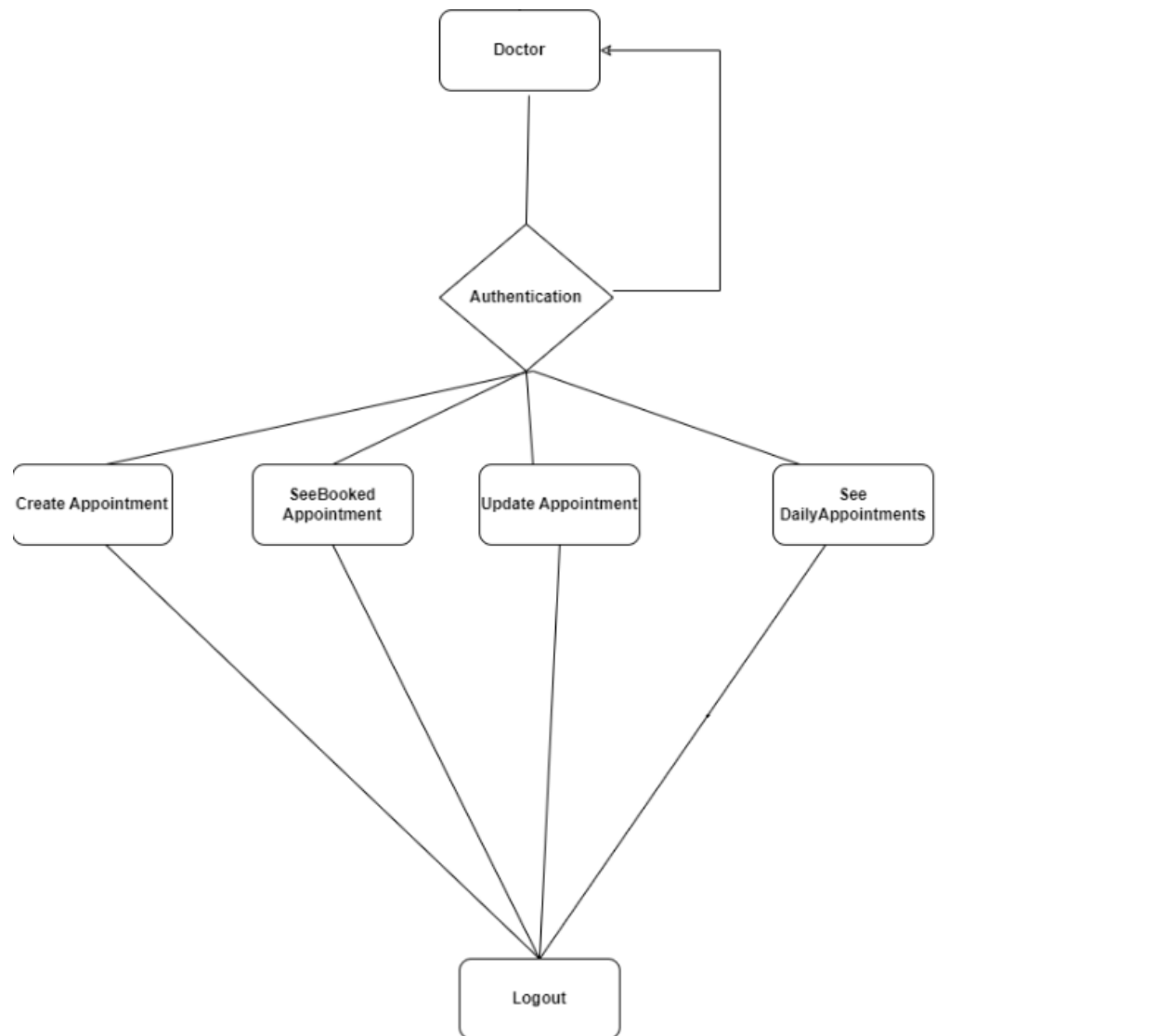
Doctors can manage their profile and can update their details.

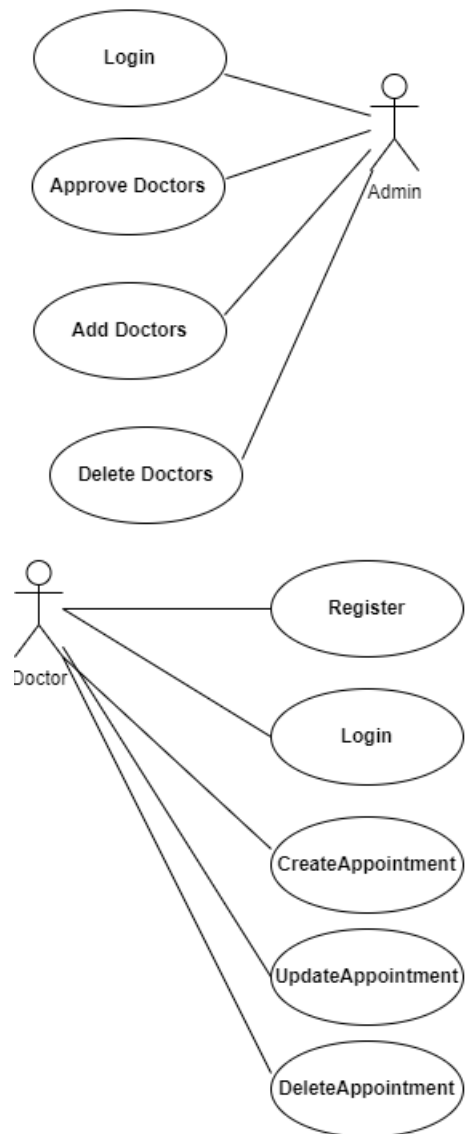
➤ **List All Appointments**

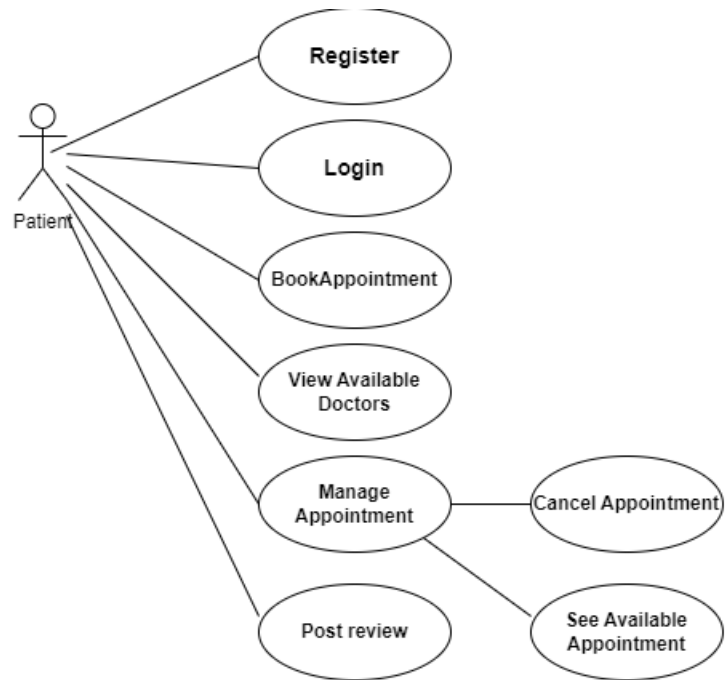
Doctors will have a list view of all the appointments of him/her.

➤ **Manage Appointments**

Doctors can manage appointments with the users according to their preference.



USE Case Diagrams:



SYSTEM ANALYSIS

- Describes the process of gathering and interpreting facts to recommend improvements for the system.
- Covers the existing system's drawbacks and the proposed system's benefits.
- Outlines the system requirement specifications including general description, objectives, and requirements.

2.1 EXISTING SYSTEM

In the existing system on demand home service functionalities are handled manually by admin by keeping the records of every doctor in a register. There is no record present for billings of users and no history is available for appointments.

- It is less user-friendly.
- It is difficult to know the user's history.
- Difficult to maintain a user record.

2.2 PROPOSED SYSTEM

In the proposed system, we are developing a web application for on demand Doctor Appointment. In which three modules have access to the system. The proposed system allows to keep the record for every patient and Doctor. It will also have all the records of patient and doctor booking and appointment.

2.3 SYSTEM REQUIREMENT SPECIFICATION

2.3.1 GENERAL

DESCRIPTION

Product Description:

The "Arogyam: Doctor Appointment System" is an innovative and user-friendly web application designed to revolutionize the way patients and doctors interact within the healthcare domain. This system aims to provide a comprehensive platform that simplifies the appointment booking process, fosters effective communication between patients and doctors, and enhances overall healthcare management.

Problem Statement:

Furthermore, the lack of a centralized and user-friendly platform for doctor-patient interactions hinders the potential for seamless healthcare management. The absence of a digital solution for appointment scheduling, medical history storage, and efficient communication prevents healthcare providers from delivering optimal services and patients from accessing healthcare conveniently.

Challenges:

1. **Inefficient Appointment Booking:** Patients find it difficult to identify suitable doctors based on their specialization and availability, leading to long waiting times and frustration.
2. **Limited Patient-Doctor Communication:** Lack of effective communication channels prevents patients from providing timely feedback and doctors from addressing patient concerns and improving their services.
3. **Cumbersome Medical Record Keeping:** Patients struggle to maintain and share their medical history, leading to repetitive information gathering during appointments and potential gaps in healthcare management.
4. **Manual Administrative Tasks:** Administrators are burdened with manual tasks such as managing appointments, doctor profiles, and patient records, which can lead to errors and inefficiencies.
5. **Lack of Transparency:** Patients often lack sufficient information about doctors' qualifications, specializations, and patient reviews, making it challenging to make informed healthcare decisions.

SYSTEM OBJECTIVES

- To provide a Web application for on demand Doctor Appointment
- To provide a web app for patient to book Appointment of any doctor in specific area.
- To provide web app which will save patient history and provide an easy access to hospitals.

2.3.2 SYSTEM REQUIREMENTS

2.3.3.1 NON-FUNCTIONAL REQUIREMENTS

Following Non-Functional Requirements will be there in the insurance to the internet:

1. Efficiency and Performance:

- The system should respond promptly to user actions, ensuring minimal waiting times for loading pages and processing requests.
- The platform should be capable of handling a substantial number of concurrent users without significant performance degradation.

2. Reliability and Availability:

- The system should have high availability, ensuring that users can access the platform 24/7, except during scheduled maintenance.
- Backup and recovery mechanisms should be in place to prevent data loss in case of unexpected system failures.

3. Usability and User Experience:

- The user interface should be intuitive, user-friendly, and easily navigable for users with varying levels of technical expertise.
- The platform should be accessible across different devices and screen sizes, providing a consistent and responsive user experience.

4. Security and Privacy:

- User authentication and authorization mechanisms should be robust, preventing unauthorized access to sensitive data.
- Patient medical records and personal information should be encrypted and stored securely to maintain privacy and compliance with data protection regulations.

5. Scalability:

- The system architecture should be designed to accommodate future growth in terms of user base and data volume without compromising performance.
- Scalability should be achieved through effective database design, load balancing, and resource management.

6. Interoperability and Integration:

- The system should support integration with existing healthcare systems, enabling seamless data exchange between different healthcare providers.
- APIs and standardized protocols should be used to facilitate interoperability with external systems if required.

7. Compliance and Regulations:

- The platform should adhere to relevant healthcare regulations and standards, ensuring the protection of patient information and healthcare data.
- Compliance with data protection laws, such as HIPAA, GDPR, and local healthcare regulations, should be a priority.

8. Compatibility:

- The platform should be compatible with a wide range of browsers, operating systems, and devices to accommodate diverse user preferences.

9. Maintenance and Support:

- The system should have a clear maintenance plan, including regular updates, bug fixes, and enhancements based on user feedback.
- Technical support should be readily available to assist users with any issues they encounter while using the platform.

10. Documentation:

- Comprehensive documentation should be provided for system users, administrators, and developers, covering user guides, system architecture, APIs, and troubleshooting steps.

11. Performance Monitoring and Reporting:

- The system should include tools for monitoring performance metrics and generating reports to analyze user engagement, system usage, and any potential issues.

2.3.3.2 FUNCTIONAL REQUIREMENTS

1. User Authentication and Authorization:

- Users (admins, doctors, patients) should be able to register and log in to the system with unique credentials.
- Admins should have elevated privileges to manage users and approve new doctor registrations.

2. Admin Module:

- Admins should be able to view, add, update, and remove doctor profiles.
- Admins should be able to approve doctor registrations after verification.
- Admins should manage patient profiles and their appointments.

3. Doctor Module:

- Doctors should be able to create and manage their profiles, including personal information, specialization, and availability.
- Doctors should have a dashboard to view upcoming patient appointments.
- Doctors should be able to update their availability slots and manage booked appointments.

4. Patient Module:

- Patients should be able to create and manage their profiles, providing personal information and medical history.
- Patients should be able to search for available doctors based on specialization, location, and other criteria.
- Patients should be able to view doctors' profiles, available time slots, and book appointments.
- Patients should receive confirmation and reminders for their booked appointments.

5. Appointment Booking:

- Patients should be able to choose a suitable doctor and available time slot for booking appointments.
- The system should prevent double booking of time slots and ensure accurate scheduling.

6. Doctor Timetable Management:

- Doctors should be able to set their working hours, available time slots, and update them as needed.
- Doctors should have the ability to mark specific time slots as unavailable (e.g., lunch breaks).

7. Slot Booking and Management:

- Patients should be able to view available time slots for a chosen doctor and book appointments accordingly.
- Doctors should have a view of their booked appointments and the patient details.

8. Review and Rating System:

- Patients should be able to provide reviews and ratings for doctors after their appointments.
- Doctors should be able to view their reviews and ratings.

9. Notification and Communication:

- Patients should receive notifications and reminders for their upcoming appointments via email or SMS.
- Doctors and patients should be able to communicate through in-app messaging for non-urgent queries.

10. Data Management:

- The system should securely store and manage user profiles, medical records, appointment details, and reviews.
- Personal and sensitive information should be encrypted and protected.

11. Search and Filter Functionality:

- The system should provide advanced search and filter options for patients to find doctors based on specialization, location, ratings, etc.

12. **Reporting and Analytics:

- Admins should have access to analytics and reports on appointment trends, doctor availability, patient engagement, and system usage.

13. System Administration:

- System admins should have the ability to manage user accounts, reset passwords, and handle technical issues.

SYSTEM DESIGN

System design is the solution for the creation of a new system. This phase focuses on the detailed implementation of the feasible system. Its emphasis is on translating design specifications to performance specification. System design has two phases of development.

- Logical Design
- Physical Design

During logical design phase the analyst describes inputs (sources), outputs (destinations), databases (data stores) and procedures (data flows) all in a format that meets the user requirements. The analyst also specifies the needs of the user at a level that virtually determines the information flow in and out of the system and the data resources. Here the logical design is done through data flow diagrams and database design. The physical design is followed by physical design or coding. Physical design produces the working system by defining the design specifications, which specify exactly what the candidate system must do. The programmers write the necessary programs that accept input from the user, perform necessary processing on accepted data and produce the required report on a hard copy or display it on the screen.

3.1 INPUT AND OUTPUT DESIGN

3.1.1 INPUT DESIGN:

User Registration and Login:

- User registration forms will collect information such as name, email, password, and contact details.
- Input validation will ensure that the entered data meets required criteria (e.g., valid email format, strong password).
- Captcha or other security measures may be implemented to prevent automated registrations.

Doctor Profile Creation:

- Doctors will provide information about their specialization, clinic address, availability hours, fees, and other relevant details.
- The input form will ensure completeness and accuracy of information to create comprehensive doctor profiles.

Appointment Booking:

- Patients will select their preferred doctor, appointment date, time slot, and any specific concerns.
- The system will validate the availability of the chosen slot and confirm the booking to the patient.

Patient Medical History:

- Patients can input their medical history, including previous diagnoses, allergies, medications, and other relevant information.
- The input form may provide predefined options or free-text fields for patients to provide detailed information.

Review and Rating Submission:

- Patients can submit reviews and ratings for doctors they have visited.
- The input form will allow patients to provide qualitative feedback and a numerical rating based on their experience

3.1.2 OUTPUT DESIGN:

Computer output is the most important and direct source of information to the user. Output design is a very important phase since the output needs to be in an efficient manner. Efficient and intelligible output design improves the system relationship with the user and helps in decision making. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output. The output module of this system is the selected notifications.

DATABASE DESIGN

3.2 DATABASE

Databases are the storehouses of data used in the software systems. The data is stored in tables inside the database. Several tables are created for the manipulation of the data for the system. Two essential settings for a database are

- Primary key - the field that is unique for all the record occurrences
 - Foreign key - the field used to set relation between tables
- Normalization is a technique to avoid redundancy in the tables.

3.3 SYSTEM TOOLS

The various system tools that have been used in developing both the front end and the back end of the project are being discussed in this chapter.

3.3.1 FRONT END:

React is a library which is developed by Face book is utilized to implement the frontend. React (also known as React.js or React JS) is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Face book and a community of individual developers and companies. React can be used as a base in the development of single page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

3.3.2 BACKEND:

The back end is implemented using MySQL which is used to design databases.

MySQL:

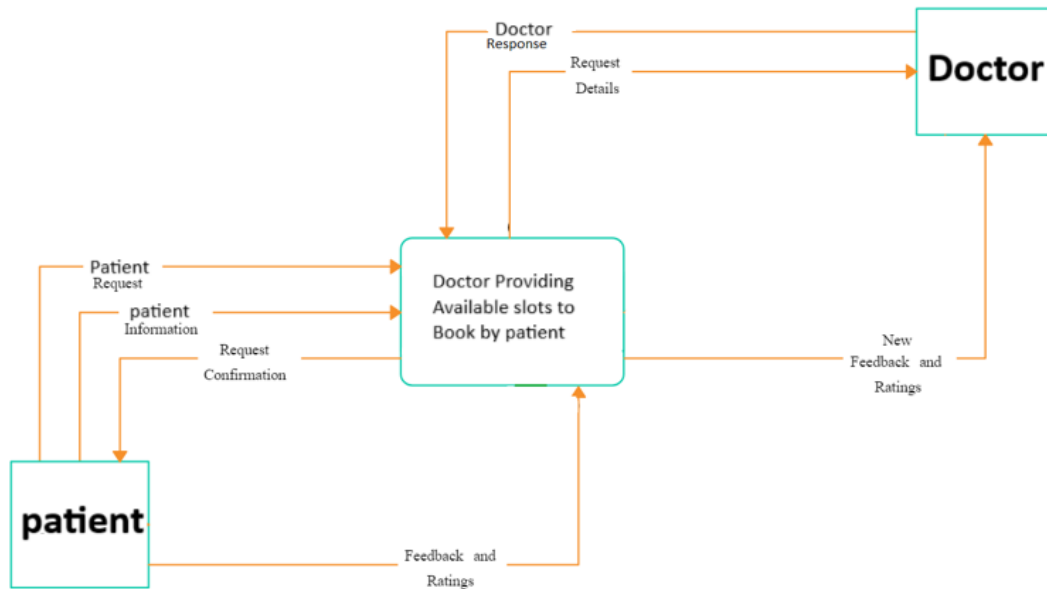
MySQL is the world's second most widely used open-source relational database

management system (RDBMS). The SQL phrase stands for Structured Query Language.

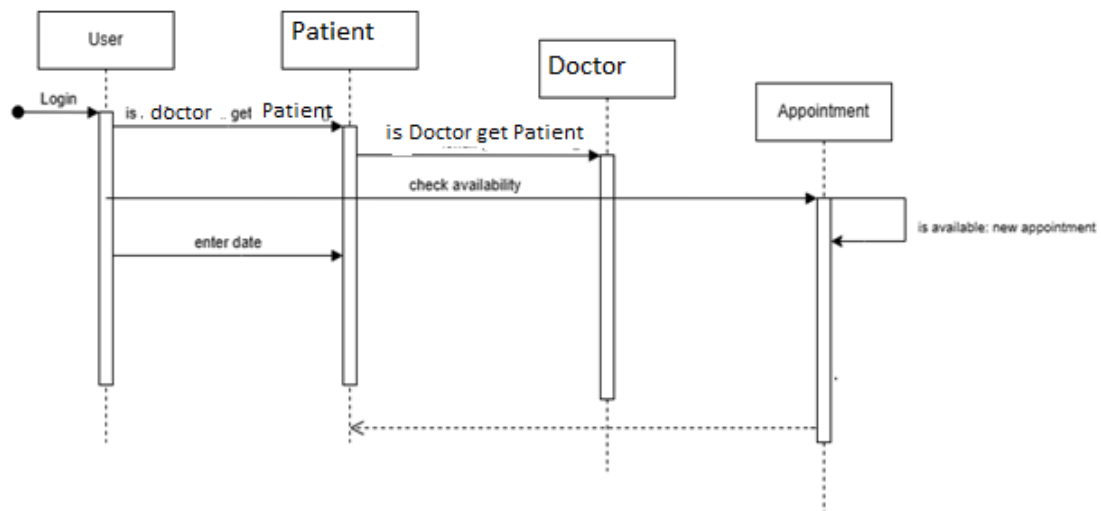
Spring-Boot:

This is used to connect MYSQL and fetch data from database and store the data in database. The Spring Framework is an application framework and inversion of control container for the Java platform. The framework's core features can be used by any Java application, but there are extensions for building web applications on top of the Java EE (Enterprise Edition) platform. Although the framework does not impose any specific programming model, it has become popular in the Java community as an addition to the Enterprise JavaBeans (EJB) model. The Spring Framework is Open-source Framework.

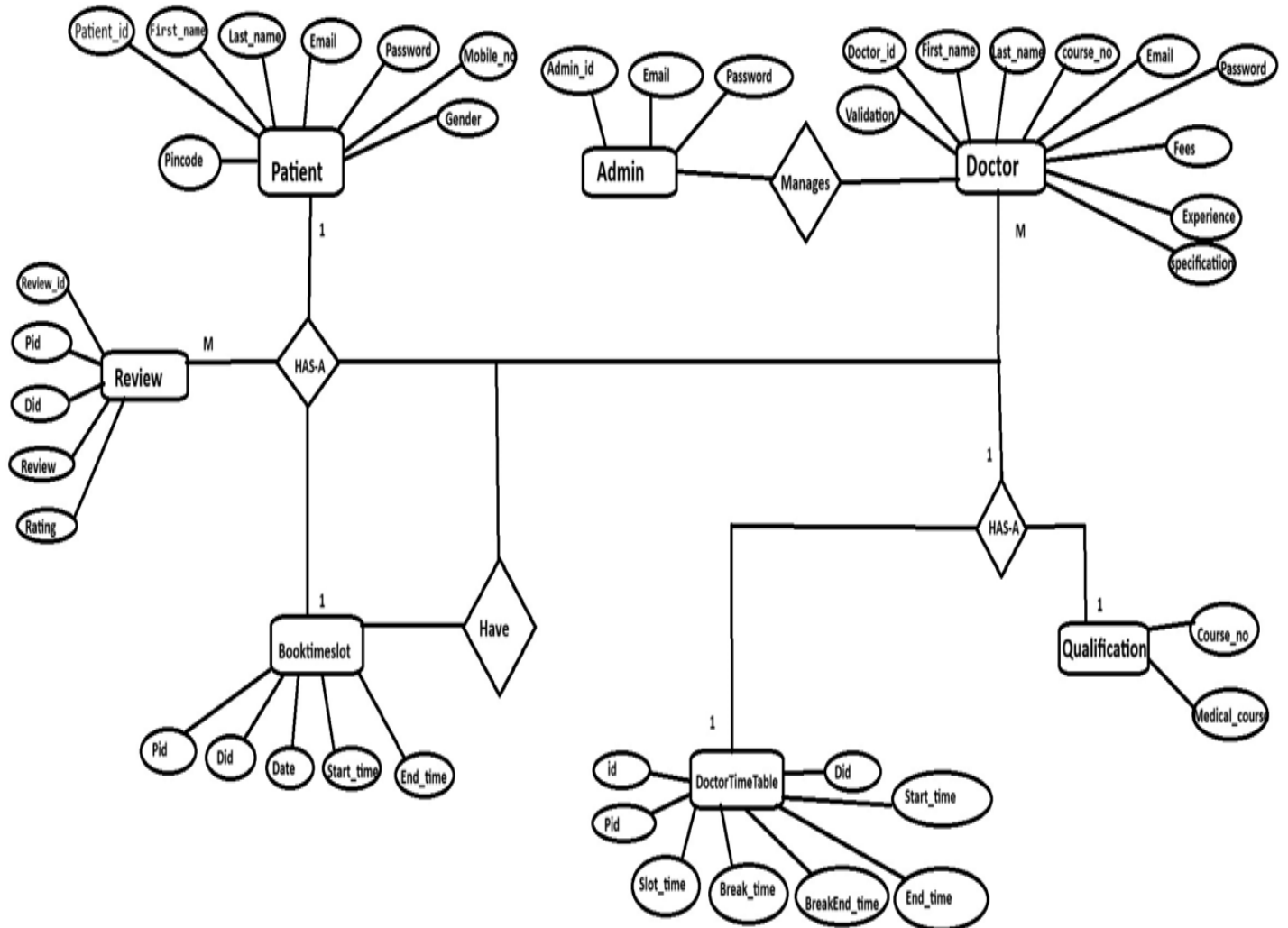
Data Flow Diagram (DFD):



Sequence Diagram:



E-R Diagram



Class Diagram:

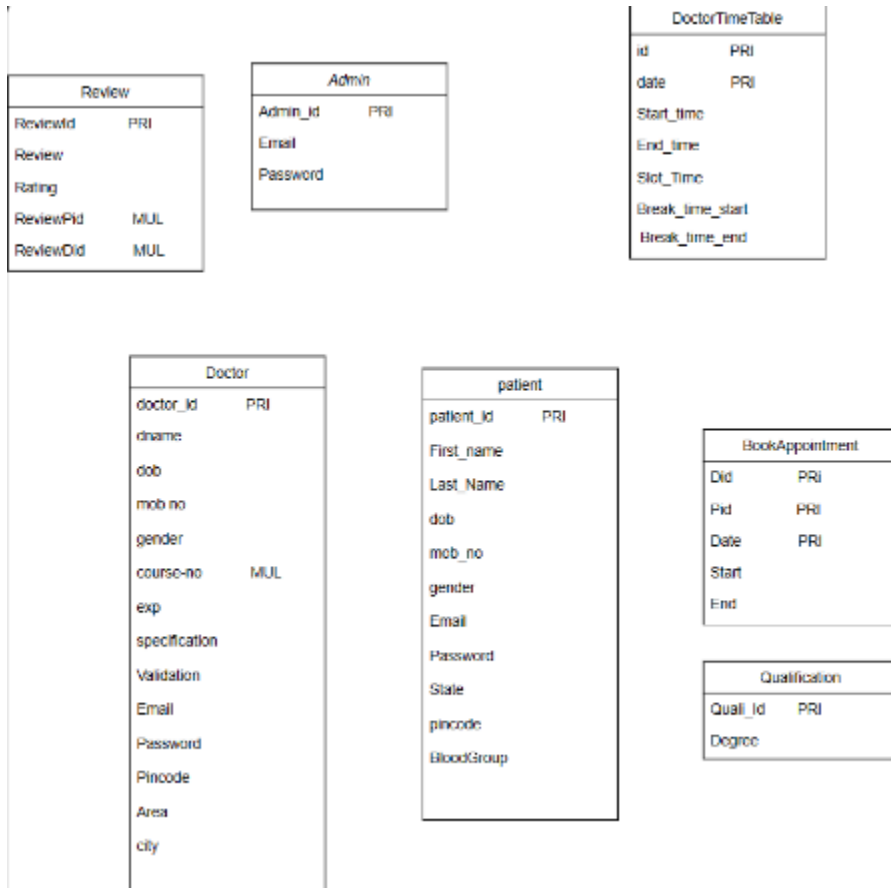


TABLE STRUCTURE:

Admin:

```
mysql> desc admin;
```

Field	Type	Null	Key	Default	Extra
aid	int	NO	PRI	NULL	auto_increment
email	varchar(255)	YES		NULL	
name	varchar(255)	YES		NULL	
pass_word	varchar(255)	YES		NULL	

4 rows in set (0.01 sec)

Patient:

```
mysql> desc patient;
```

Field	Type	Null	Key	Default	Extra
patient_id	bigint	NO	PRI	NULL	auto_increment
area	varchar(255)	YES		NULL	
blood_group	varchar(255)	YES		NULL	
city	varchar(255)	YES		NULL	
country	varchar(255)	YES		NULL	
mobilenno	varchar(255)	YES		NULL	
pincode	varchar(255)	YES		NULL	
state	varchar(255)	YES		NULL	
dob	date	YES		NULL	
email	varchar(255)	YES		NULL	
first_name	varchar(255)	YES		NULL	
gender	varchar(255)	YES		NULL	
last_name	varchar(255)	YES		NULL	
middle_name	varchar(255)	YES		NULL	
password	varchar(255)	YES		NULL	
profilepic	longblob	YES		NULL	

16 rows in set (0.00 sec)

Doctor:

```
mysql> desc doctor;
```

Field	Type	Null	Key	Default	Extra
doctor_id	bigint	NO	PRI	NULL	auto_increment
area	varchar(255)	YES		NULL	
city	varchar(255)	YES		NULL	
country	varchar(255)	YES		NULL	
dob	date	YES		NULL	
first_name	varchar(255)	YES		NULL	
gender	varchar(255)	YES		NULL	
langauges	varchar(255)	YES		NULL	
last_name	varchar(255)	YES		NULL	
middle_name	varchar(255)	YES		NULL	
mob_no	varchar(255)	YES		NULL	
specialisation	varchar(255)	YES		NULL	
state	varchar(255)	YES		NULL	
email_id	varchar(255)	YES		NULL	
fees	int	YES		NULL	
password	varchar(255)	YES		NULL	
pincode	varchar(255)	YES		NULL	
practice_start_year	date	YES		NULL	
profile_pic	varchar(255)	YES		NULL	
validation	bit(1)	YES		NULL	
qualifi	int	YES	MUL	NULL	

21 rows in set (0.01 sec)

Book slots:

```
mysql> desc bookslots;
```

Field	Type	Null	Key	Default	Extra
cid	bigint	NO	PRI	NULL	
date_of_appointment	date	NO	PRI	NULL	
pid	bigint	NO	PRI	NULL	
end	time	YES		NULL	
start	time	YES		NULL	

5 rows in set (0.00 sec)

Doctor Timetable:

```
mysql> desc bookslots;
```

Field	Type	Null	Key	Default	Extra
date_of_appointment	date	NO	PRI	NULL	
did	bigint	NO	PRI	NULL	
pid	bigint	NO	PRI	NULL	
end	time	YES		NULL	
start	time	YES		NULL	

```
5 rows in set (0.00 sec)
```

```
mysql> desc doctimetable;
```

Field	Type	Null	Key	Default	Extra
date_of_appointment	date	NO	PRI	NULL	
doc_did	bigint	NO	PRI	NULL	
break_time_end	time	YES		NULL	
break_time_start	time	YES		NULL	
end_time	time	YES		NULL	
slot_duration	time	YES		NULL	
start_time	time	YES		NULL	

```
7 rows in set (0.00 sec)
```

Review

```
mysql> desc qualification;
```

Field	Type	Null	Key	Default	Extra
quali_id	int	NO	PRI	NULL	auto_increment
medical_course	varchar(255)	YES		NULL	

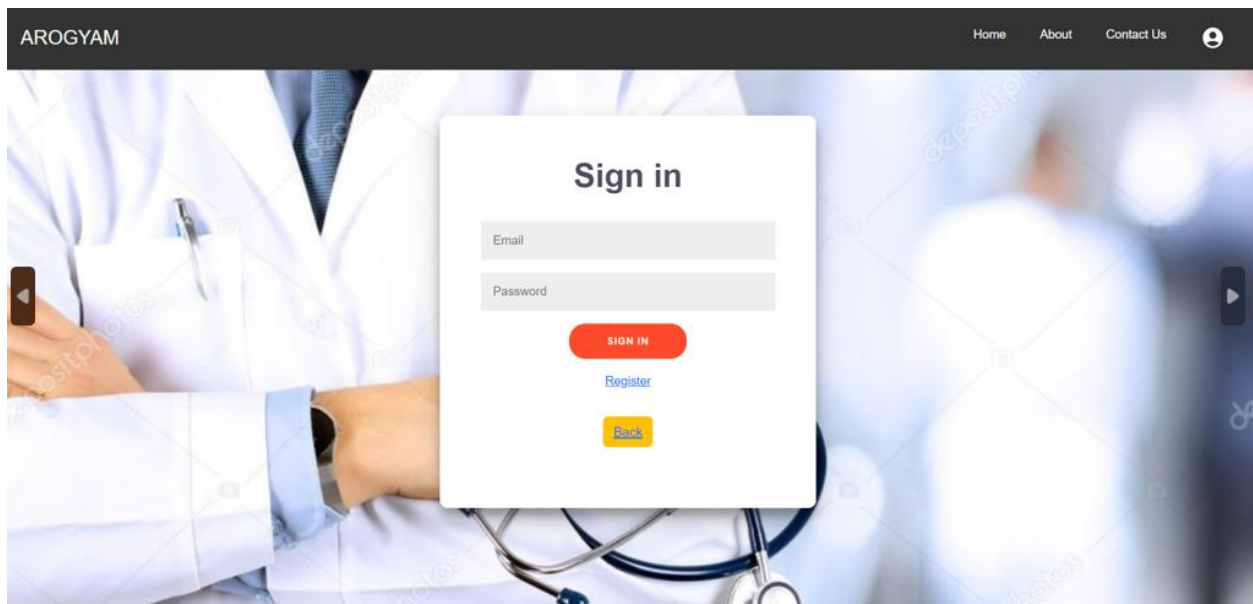
```
2 rows in set (0.00 sec)
```

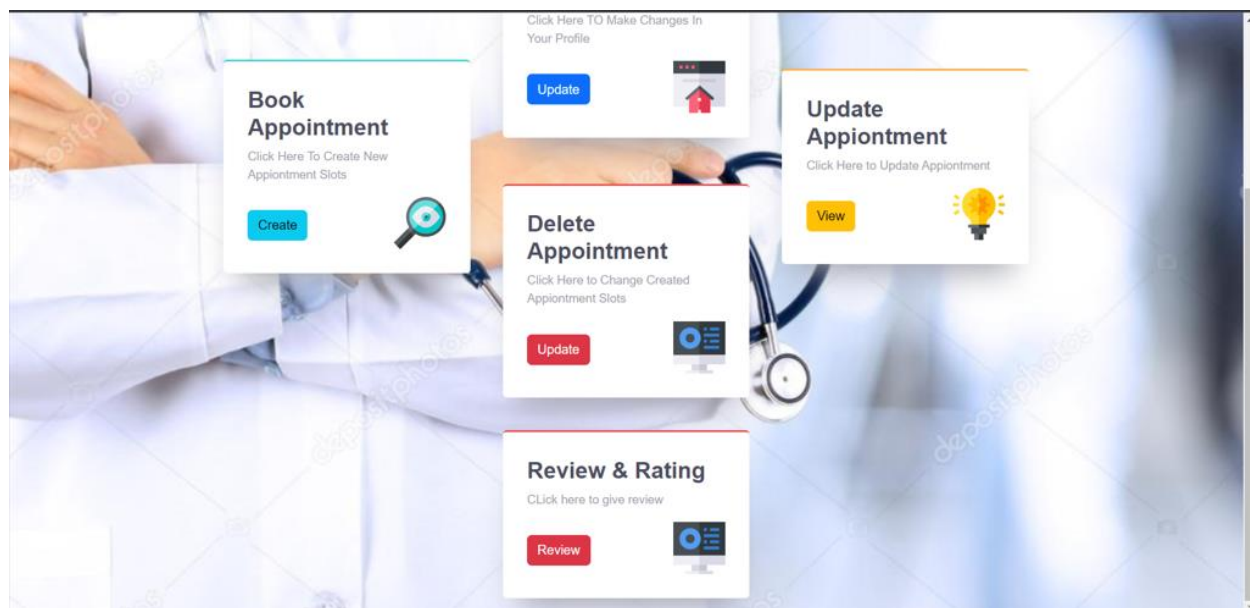
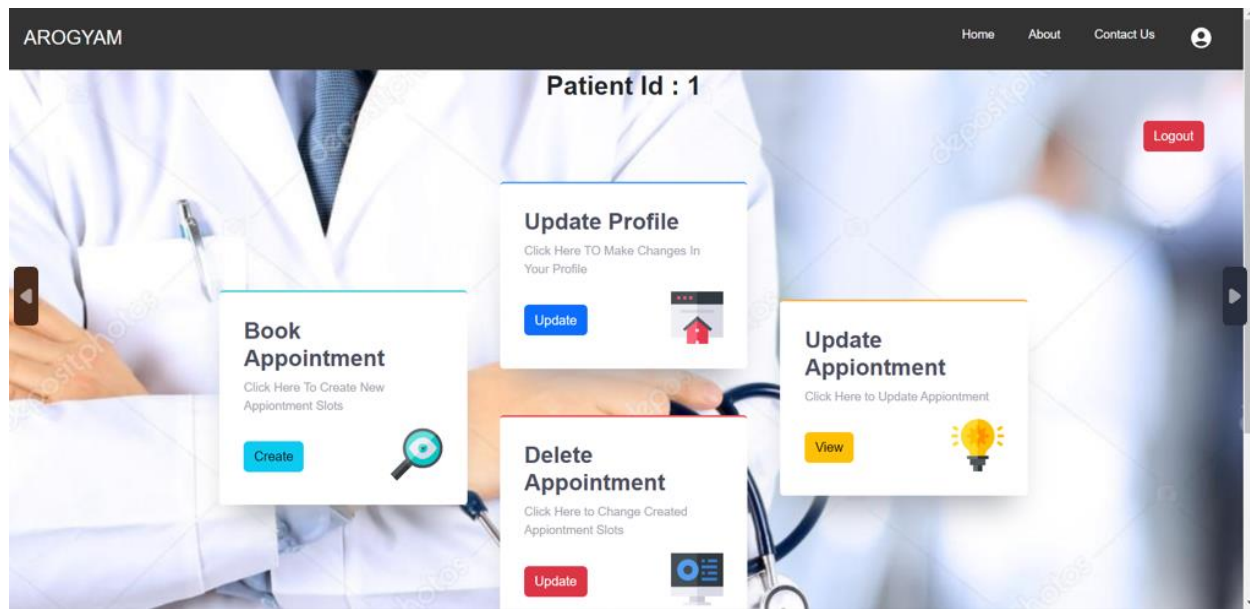
```
mysql> desc review_and_rating;
```

Field	Type	Null	Key	Default	Extra
review_id	bigint	NO	PRI	NULL	auto_increment
rating	varchar(20)	YES		NULL	
review	varchar(255)	YES		NULL	
doctor_id	bigint	YES	MUL	NULL	
patient_id	bigint	YES	MUL	NULL	


```
5 rows in set (0.00 sec)
```


Project Screenshots:





AROGYAM

[Home](#) [About](#) [Contact Us](#) 

Logout

Enter Pincode

SUBMIT

Back

Available Doctor

Doctor	city	area	pincode	fees	practicing Since	specialisation	Action
Akshay khadse	pune	pune	123456	1000	2023-08-25	Heart Specialist	Slots
sarang Rohankar	Pune	Akurdi	123456	200	2023-08-23	Heart Specialist	Slots

sarang Rohankar

Pune

Akurdi

123456

200

2023-08-23

Heart Specialist

[Slots](#)

Available Day For Appointment

Start Time	End Time	Duration	Date	Action
08:00:00	20:00:00	00:30:00	2023-08-30	View Slots

Available Slots

16:30:00 - 17:00:00

08:00:00 - 08:30:00

10:30:00 - 11:00:00

17:00:00 - 17:30:00

15:30:00 - 16:00:00

09:00:00 - 09:30:00

10:00:00 - 10:30:00

17:30:00 - 18:00:00

13:00:00 - 13:30:00

11:30:00 - 12:00:00

18:00:00 - 18:30:00

13:30:00 - 14:00:00


14:00:00 - 14:30:00

09:30:00 - 10:00:00

08:30:00 - 09:00:00

14:30:00 - 15:00:00

AROGYAM

[Home](#) [About](#) [Contact Us](#) 

Review:

Patient ID:


Doctor ID:

Rating:

BEST

SUBMIT

AROGYAM

[Home](#) [About](#) [Contact Us](#) 


Doctor Id : 4

Logout

Create Appointments

Click Here To Create New Appointment Slots


Create



Update Profile

Click Here TO Make Changes In Your Profile


Update



Update Appointment

Click Here to Change Created Appiontment Slots


Update

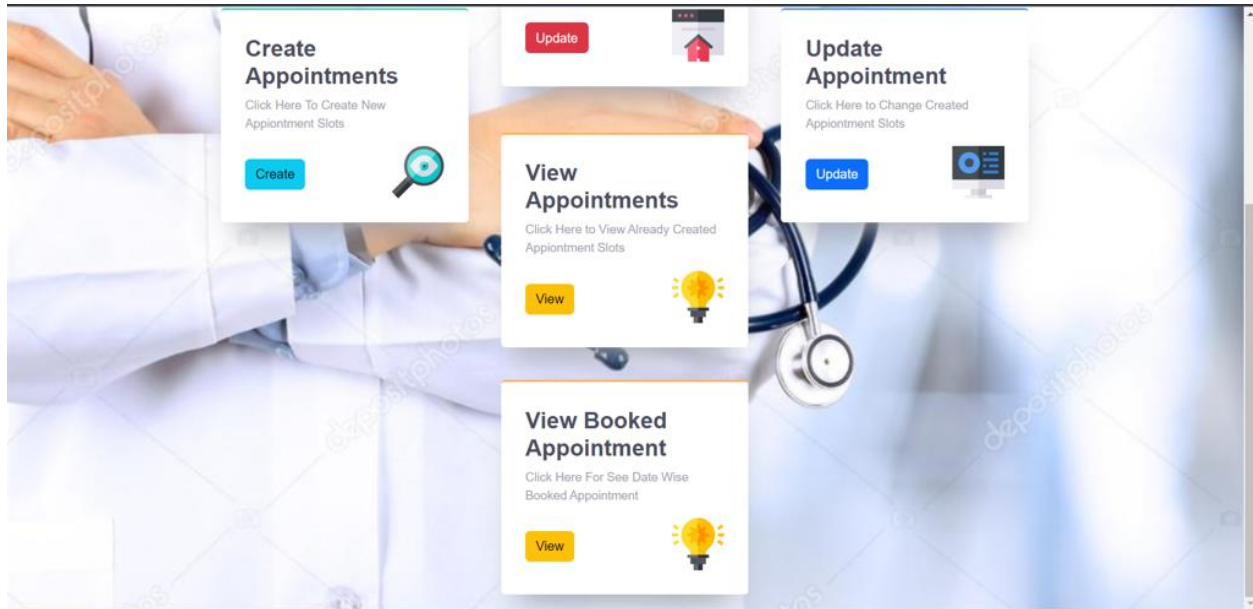


View Appointments

Click Here to View Already Created Appiontment Slots

View

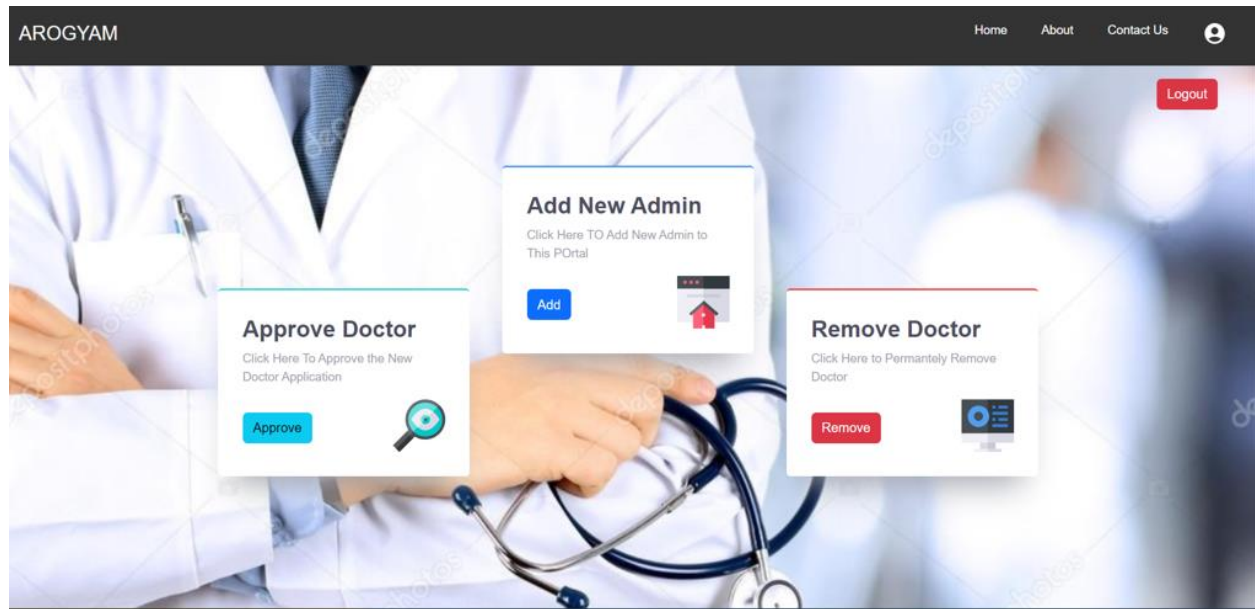




The form is used to configure appointment slots. It includes the following fields and components:

- Start Time**: Input field with "5 am".
- EndTime**: Input field with "5 am".
- SlotTime**: Input field with "10 min".
- Break Start Time**: Input field with "11 am".
- Break End Time**: Input field with "12 pm".
- Calendar**: A monthly calendar for August 2023. The 30th is highlighted in yellow.
- SUBMIT**: A red button.
- Back**: A yellow button.

MON	TUE	WED	THU	FRI	SAT	SUN
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3



CONCLUSION

In conclusion, the "Arogyam" Doctor Appointment System offers a modern and efficient solution to the challenges of traditional appointment booking. By providing a user-friendly platform for patients, doctors, and administrators, the system simplifies appointment scheduling, medical record management, and communication.

Through the integration of advanced technologies and responsive design, the system ensures seamless user experience across various devices and platforms. Patients can easily book appointments and access medical history, doctors can manage schedules and patient profiles, and administrators can oversee the system's functionality.

The "Arogyam" system aims to enhance healthcare services by bridging the gap between patients and doctors, making appointments more accessible and hassle-free. This project showcases both technical expertise and the commitment to improving healthcare, offering a glimpse into the future of efficient medical appointment systems.

ONLINE REFERENCES

- [1] www.Google.com
- [2] www.w3school.com
- [3] www.javatpoint.com
- [4] <https://docs.spring.io/spring-framework/docs>
- [5] <https://getbootstrap.com/docs/4.0>