

ABSTRACT

The goal of the doctor appointment system is to automate the current manual system with the aid of computerized tools and comprehensive computer software, meeting their needs in order to store their important data and information for a longer period of time with simple access and manipulation. The necessary gear and software are readily available and simple to use.

The above-described doctor appointment system can result in an error-free, secure, dependable, and quick management system. It can help the user focus on their other tasks rather than record-keeping so that they can better accomplish their goals. As a result, it will aid organizations in making better use of their resources. The company can keep computerized records updated without making duplicate inputs. This implies that while one can access the information, they do not need to be side-tracked by irrelevant information.

The goal is to automate its current manual method with the aid of computerized tools and comprehensive computer software, meeting their needs, so that their important data and information may be saved for a longer period of time with simple access and manipulation. In essence, the project outlines how to manage for superior performance and better client services.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION	1
1.1 PROJECT DESCRIPTION.....	1
1.2 COMPANY PROFILE.....	1
CHAPTER 2: LITERATURE SURVEY	2
2.1 EXISTING AND PROPOSED SYSTEM.....	2
2.2 FEASIBILITY STUDY.....	2
2.3 TOOLS AND TECHNOLOGIES USED.....	3
CHAPTER 3: SOFTWARE REQUIREMENTS SPECIFICATION	4
3.1 INTRODUCTION	4
3.2 OVERALL DESCRIPTION.....	5
3.3 PRODUCT FUNCTIONALITY.....	8
3.4 EXTERNAL INTERFACE REQUIREMENTS	8
3.5 OTHER NON-FUNCTIONAL REQUIREMENTS.....	9
3.6 SPECIFIC REQUIREMENTS.....	10
3.7 DELIVERY PLAN.....	10
CHAPTER 4: SYSTEM DESIGN	11
4.1 SYSTEM DESIGN.....	11
4.2 SOFTWARE PRODUCT ARCHITECTURE.....	11
4.3 COMPONENT ARCHITECTURE.....	12
4.4 DATA FLOW DIAGRAM	13
CHAPTER 5: DETAILED DESIGN.....	15
5.1 USE CASE DIAGRAM	15
5.2 SEQUENCE DIAGRAM.....	16
5.3 ACTIVITY DIAGRAMS.....	18
5.4 DATABASE DESIGN	19
5.4.1 ER DIAGRAM.....	21
CHAPTER 6: IMPLEMENTATION	22
6.1 INTRODUCTION	22
6.2 PSEUDO CODES	22
CHAPTER 7: SOFTWARE TESTING	24
7.1 INTRODUCTION	24
7.2 TESTING OBJECTIVE.....	24
7.3 UNIT TESTING.....	24

7.4 INTEGRATION TESTING.....	26
CHAPTER 8: CONCLUSION.....	28
CHAPTER 9: FUTURE ENHANCEMENTS	29
APPENDIX A: REFERENCES	30
APPENDIX B: USER MANUAL.....	31

CHAPTER 1: INTRODUCTION

1.1 PROJECT DESCRIPTION

The goal of the doctor appointment system is to automate the current manual system with the aid of computerized hardware and comprehensive computer software, meeting their needs so that their valuable data/information can be stored for a longer period with simple access and manipulation. It is simple to find and use the necessary software and hardware.

The technique for scheduling appointments with doctors stated above can result in a quick, secure, and error-free management system. It can let the user transfer their focus from record keeping to other responsibilities so they can better accomplish their goals. It will therefore aid businesses in better resource management. With computerized records, the company can avoid entering the same data twice. As a result, there is no need for unrelated material to divert attention from obtaining the information.

The objective is to automate their manual procedure while still satisfying their needs, allowing them to maintain their crucial data and information for a longer period of time with straightforward access and manipulation. Computerized tools and advanced software are used to do this. The project outlines how to manage for improved performance and customer service.

1.2 COMPANY PROFILE

RineX Technologies is a leading global technology company specializing in providing innovative solutions and services in artificial intelligence, data analytics, and software development. Established in 2005, RineX has grown to become a trusted partner for businesses across various industries, empowering them to make data-driven decisions and stay ahead in the digital era. The headquarters is situated in Bengaluru, India.

CHAPTER 2: LITERATURE SURVEY

2.1 EXISTING AND PROPOSED SYSTEM

For the clinic to become more productive and patient-friendly, there are a number of significant issues with the manual appointment system that need to be resolved. Lack of knowledge regarding doctors' availability and specialization is one of the main problems, which results in long waiting lines and uncertainty about getting an appointment. Due to the lack of an effective system for making appointments, patients frequently struggle to identify doctors who are right for them and experience problems.

Patients may find it difficult to find an appropriate time slot that works with their schedules due to the inconvenient appointment schedule, which exacerbates the issue. Additionally, clinics lack a reliable method for keeping records that will enable them to track their progress and produce reports, which limits their ability to be more efficient and successful.

An online method for scheduling appointments can be used to overcome these difficulties and speed up the procedure. With the help of this solution, patients can quickly check a doctor's availability and areas of expertise, enabling them to choose the best healthcare professional for their needs. Additionally, individuals who choose not to use online services now have an alternative with the introduction of telephonic offline appointment possibilities.

By implementing such a system, patients will be able to request appointments at their leisure, get thorough information on doctors, and comprehend the services offered. Additionally, it makes medications and medical records easy to store and retrieve, guaranteeing that healthcare workers can easily access patient information.

Overall, integrating an online appointment booking system and better record-keeping procedures can increase the clinic's productivity, give patients a seamless experience, and eventually lead to better patient results and satisfaction.

2.2 FEASIBILITY STUDY

This feasibility study's goal is to determine whether creating an e-Doc appointment system is feasible and likely to be successful. With the use of the system, patients will be able to make appointments with doctors on a comfortable and user-friendly platform. This study assesses a number of factors, such as technical, economic, and operational considerations.

Technical Feasibility

E-Doc Appointment System is developed using Visual Studio Code. Here the language used is PHP. User Interface Design used CSS, and HTML. The server used is the XAMPP server. The Database used here is MySQL.

Economic Feasibility

To use the E-Doc Appointment System user just needs a computer with an internet connection. There's nothing called a subscription charge. It's free of cost. The user just needs to sign up and provide their details and set a password.

Operational Feasibility

E-Doc Appointment is a system that can be used by individuals in an easy manner as its design is user-friendly.

2.3 TOOLS AND TECHNOLOGIES USED

E-Doc Appointment System is developed using Visual Studio Code. PHP scripting language is used to develop this application. HTML and CSS are used to design the User interface. The server used is Xampp Server. The database used here is MySQL.

CHAPTER 3: SOFTWARE REQUIREMENTS SPECIFICATION

3.1 INTRODUCTION

3.1.1 PURPOSE

The main purpose behind this SRS is to translate the ideas in the mind of a client into a formal document. It is developed to fulfill the needs of the customer. It keeps track of customer details. It is meant for use by the developer and will be the basis for validating the final delivered system any changes made to the requirement in the future will have to go through the formal change approval process.

3.1.2 DOCUMENT CONVENTIONS/ DEFINITIONS AND ABBREVIATIONS

- **CFD:** Context Flow Diagram
- **HTTP:** Hypertext Transfer Protocol
- **PHP:** PHP Hypertext Pre-processor
- **CSS:** Cascading Style Sheet
- **ER:** Entity Relationship Diagram
- **RAM:** Random Access Memory
- **DFD:** Data Flow Diagram
- **SRS:** Software Requirements Analysis and Specification
- **SQL:** Structured Query Language

3.1.3 INTENDED AUDIENCE

3.1.3.1 Developer

If the developer requires any changes or updation to be made in the system he/she should refer to this document to make the necessary changes and then pass these modifications to other phases of the development process.

3.1.3.2 Administrator

They are the ones responsible for adding any individual to the system and also responsible for providing better security features.

3.1.3.3 User

The individuals using the system would have a clear idea of the requirements for the system to operate.

3.1.4 PROJECT SCOPE

The SRS captures the entire requirement in a single document. The admin can easily access the system and check for customer needs. It is also developed to fulfill the needs of the customer. It keeps track of customer details. It is meant for use by the developer and will be the basis for validating the final delivered system any changes made to the requirement in the future will have to go through the formal change approval process. This document contains a complete description of the functioning of the e-Doc Appointment System. This is to ensure that the person reading the document understands in brief what the system is all about.

3.1.5 BENEFITS

The Benefit of this document is to serve as a guide to the developer and testers who are responsible for the development of the system.

This project also helps to

- **Effective scheduling:** The appointment system aids medical professionals in making the most of their time, minimizing wait times, and increasing the number of patients they may see in a single day. Better time management and improved patient flow result from this.
- **Less waiting:** By scheduling appointments ahead of time, patients can avoid standing in huge lines at the doctor's office or hospital. This improves the general experience and patient happiness.
- **Less paperwork:** By electronically storing patient information and making it simpler to access and update healthcare records, digital appointment systems can reduce paperwork.

3.2 OVERALL DESCRIPTION

3.2.1 IDENTIFICATION OF PRE-EXISTING WORK

This project basically deals with managing appointments between patients and doctors. It is a web-based project. This system is designed for the use of patients. It will give the details of the person who has booked appointments with selected doctors. It will have the collection of all

the patients and doctors who have access to the application. This project eliminates the paperwork.

3.2.2 PRODUCT PERSPECTIVE

The E-Doc Appointment System is a piece of software made to make arranging and organizing appointments between patients and medical professionals easier and more efficient. It acts as a conduit for good communication between patients and doctors. The following are some crucial elements of the product perspective:

Roles for users

- **Patients:** The majority of system users are patients. They can register, view doctor profiles, determine the doctors' availabilities, and make appointments.
- **Doctors:** The system is used by doctors and other healthcare professionals to manage their schedules, view appointments, and adjust their availability.

3.2.3 PRODUCT FEATURES

The user interface for this website is easy to use.

It can be accessed by patients, and doctors at any time on any platform.

- **Login:** Using this module the admin can log in with the given username and password.
- **Registration:** This module is used to register by providing his personal or required information.
- **Make an Appointment:** Users can create appointments with doctors.
- **Add Doctor:** The admin can add newly joined doctors to the system by providing their area of specialization.
- **Schedule:** The users can schedule a session with doctors.
- **View Appointments:** Users can view their appointments at what time it is scheduled and the name of the patient and the doctor.
- **Cancel Sessions:** In this module, users can cancel their scheduled appointments with doctors and vice versa.
- **All Doctors:** Here in this module patients can view all the doctors available or registered by the admin. And they can book their appointments with the required doctors.
- **My Bookings:** Here in this module patients can view their appointments with doctors and also, they can cancel their bookings.

3.2.4 END-USER CHARACTERISTICS

The system is designed with the intention to provide easy to use simple system so that no cumbersome and elaborate training for operation is needed.

E-Doc Appointment System has 3 levels of users.

3.2.4.1 Admin:

- Admin can manage the system.
- Admin can add or delete sessions of appointment.
- Admin can add or remove doctor details and provide them access to the system.

3.2.4.2 Doctor:

- The doctor can cancel the scheduled session by the admin.
- The doctor can edit personal details if there are any errors from the admin.
- Doctors can view appointments with patients.
- Doctors can also view different sessions on specific days.

3.2.4.3 Patient:

- Patients can register themselves.
- Patients can schedule appointments.
- Patients can cancel their appointments.

3.2.5 OPERATING ENVIRONMENT

3.2.5.1 General Constraints:

- Requires all the mandatory fields to be filled with the proper information

3.2.6 DESIGN AND IMPLEMENTATION CONSTRAINTS

- All the required fields should be filled.
- Fields should be filled with a proper value.
- Should enter a valid username and password.
- Enter a valid email ID.

3.2.7 ASSUMPTIONS AND DEPENDENCIES

- Application used only by those users, who have Access Permissions
- User should have a basic knowledge of computer systems.
- Requires internet connectivity to access the application.

3.3 PRODUCT FUNCTIONALITY

3.3.1 USER MODULE

3.3.1.1 User registration and authentication:

Patients, doctors, and administrators can set up accounts and use the proper credentials to log in safely.

3.3.1.2 Doctor profiles:

Admins have the ability to establish and manage doctor profiles, which can include contact information, personal details, speciality, credentials, and working hours.

3.3.1.3 Appointment Scheduling:

Patients can examine the schedules of the doctors who are available and make appointments based on those schedules.

3.3.1.4 Appointment Management:

Doctors may manage their availability and monitor their planned appointments. They can also accept or reject requests for appointments.

3.3.1.5 Rescheduling and Cancelling Appointments:

Both patients and doctors have the right to cancel appointments.

3.3.1.6 Administrative Dashboard:

To manage user accounts, address technical difficulties, and keep an eye on how the system is running overall, administrators have access to a feature-rich dashboard.

3.4 EXTERNAL INTERFACE REQUIREMENTS

3.4.1 USER INTERFACES

- The user can access the site through a web browser.
- Home page which has links to other pages.
- He can navigate to the various icons and view the system.
- A start validation is provided to the login form. On successful validation, permission to use the system is provided.

3.4.2 HARDWARE INTERFACES

Any device which has an operating system installed in it.

3.4.3 SOFTWARE INTERFACES

Requires web browser support.

3.4.4 COMMUNICATION INTERFACES

The system will require a web browser, an internet connection that supports HTTP, and Server.

3.5 OTHER NON-FUNCTIONAL REQUIREMENTS

3.5.1 PERFORMANCE REQUIREMENTS

There may be a chance of the system getting crashed, which leads to the loss of data. To avoid this there will be a backup to recover the data.

The performance of the overall system should be faster and error-free, with built-in

- Error checking and correction facilities.
- In order to run this application, we require.
- An Internet with a minimum 56 kbps bandwidth.
- To access this page, we require IE6 or any higher version browser.

3.5.2 SAFETY REQUIREMENTS

Users can access the system only after successful login. Only authorized users can make use of the application so we can prevent unauthorized access to the system.

3.5.3 SOFTWARE QUALITY ATTRIBUTES

3.5.3.1 Reliability

Measure if the product is reliable enough to sustain in any condition. Should give consistently correct results. Product reliability is measured in terms of project work under different working environments and conditions.

3.5.3.2 Maintainability

Different versions of the product should be easy to maintain. For development, it should be easy to add code to the existing system and should be easy to upgrade for new features and new technologies from time to time. Maintenance should be cost-effective and easy. The system is easy to maintain and correct defects or make a change in the software.

3.5.3.3 Portability

This can be measured in terms of Costing issues related to porting, technical issues related to porting, and Behavioural issues related to porting.

3.5.3.4 Usability

This can be measured in terms of ease of use. The application should be user-friendly. Should be easy to learn. Navigation should be simple. The system must be:

- Easy to use for input preparation, operation, and interpretation of the output.
- Provide consistent user interface standards or conventions with our other frequently used systems.
- Easy for new or infrequent users to learn to use the system.

3.5.3.5 Flexibility

Should be flexible enough to modify. Adaptable to other products with which it needs interaction. Should be easy to interface with other standard 3rd party components.

3.6 SPECIFIC REQUIREMENTS

3.6.1 OPERATING ENVIRONMENT

3.6.1.1 Hardware

Processor	:	2.4 GHZ or above
RAM	:	1GB or above
Hard Disk	:	10GB or above

3.6.1.2 Software

Language	:	PHP, HTML
Frontend	:	Visual Studio Code
Database (Backend)	:	MySQL Server

3.7 DELIVERY PLAN

The system should satisfy all the needs of the clients and components should work properly according to the specification and should satisfy all the tests.

CHAPTER 4: SYSTEM DESIGN

4.1 SYSTEM DESIGN

4.1.1 INTRODUCTION

System Design is used to define the elements of the system such as its modules, architecture, and its data. It can also be known as the development of the sub-application. In System design emphasis is on deciding which segments are required, the terms of these modules should be consistent is called System Design.

4.1.2 SCOPE

The scope of this document plays a very important role, especially for the end users in the way to book appointments with available doctors. It further describes the context flow of the system where we can observe it. Data flow of the system in which we can easily find the way of data flow. It also describes the relationship of each module of this application and use-case realization to help the developer of this application use the best technique in the coding phase.

4.2 SOFTWARE PRODUCT ARCHITECTURE

4.2.1 ARCHITECTURAL DESIGN

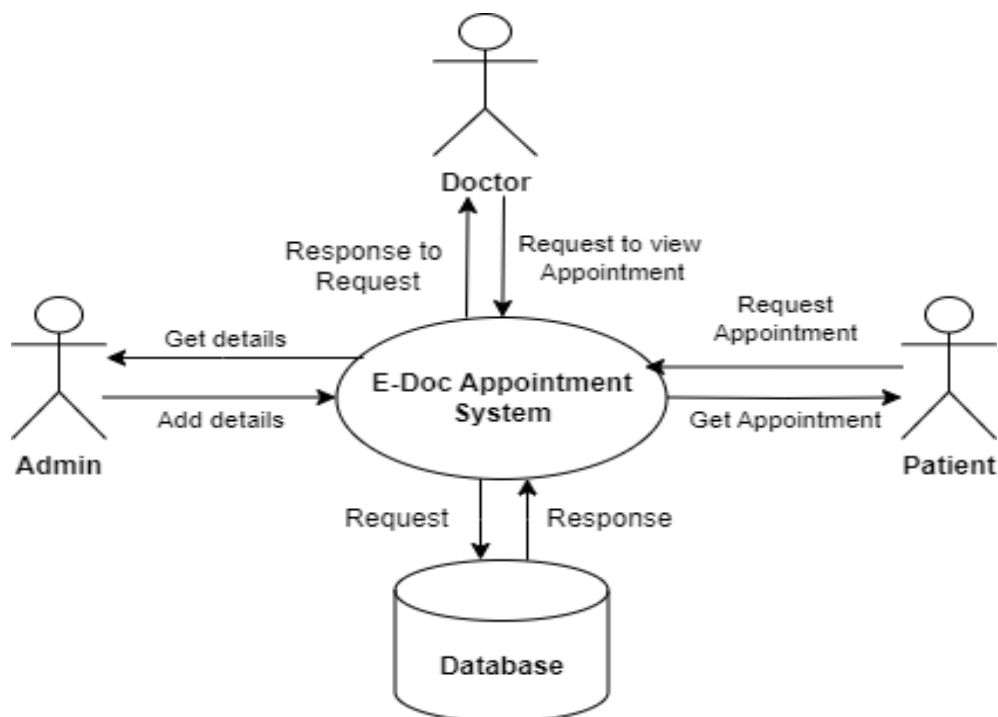


Fig 4.2.1: Architectural Design for E-Doc Appointment System

4.2.1.1 View Layer or Presentation Layer

This Layer includes the participating Users as well as Admin. These Users are directly involved with the System. The Users here include casual end-Users.

4.2.1.2 Business Layer

The business Layer interacts with the Database to store and retrieve Data.

4.1.1.3 Access Layer

Access Layer involves the Database and is used to store and retrieve the Data pertaining to the Users.

4.3 COMPONENT ARCHITECTURE

4.3.1 User Interface:

- Patient Interface: This allows patients to register, log in, search for doctors, view available time slots, and book appointments.
- Doctor Interface: This enables doctors to manage their schedules, view patient appointments, and access patient information.
- Admin Interface: Used by system administrators to manage user accounts, handle system configurations, and monitor system performance.

4.3.2 Authentication and Authorization:

- Authentication Module: Handles user login, registration, and authentication processes to ensure secure access to the system.
- Authorization Module: Manages user permissions and access levels based on their roles (patient, doctor, admin).

4.3.3 Appointment Management:

- Appointment Scheduler: Handles the scheduling of appointments, checking for available time slots.

4.3.4 Doctor and Patient Profile Management:

- Doctor Profile: Allows doctors to manage their personal information, specialization, and availability.
- Patient Profile: Allows patients to update their personal information and preferences.

4.3.5 Database Management:

- Patient Database: Stores patient information, and appointment details.
- Doctor Database: Stores details of doctors, including their specialties and availability.
- Appointment Database: Holds information about scheduled appointments, including date, and time.

4.3.6 Search and Filter:

- Doctor Search: Enables patients to search for doctors based on specialization, availability, etc.
- Appointment Filter: Allows patients to filter available appointment slots based on their preferences.

4.4 DATA FLOW DIAGRAM

4.4.1 CONTEXT FLOW DIAGRAM

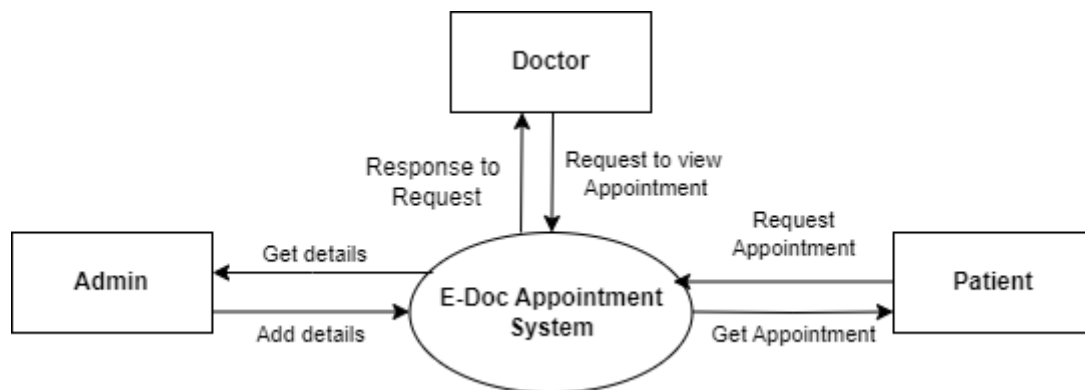


Fig 4.4.1: Context Flow Diagram for E-Doc Appointment System

4.4.2 DFD FOR Admin

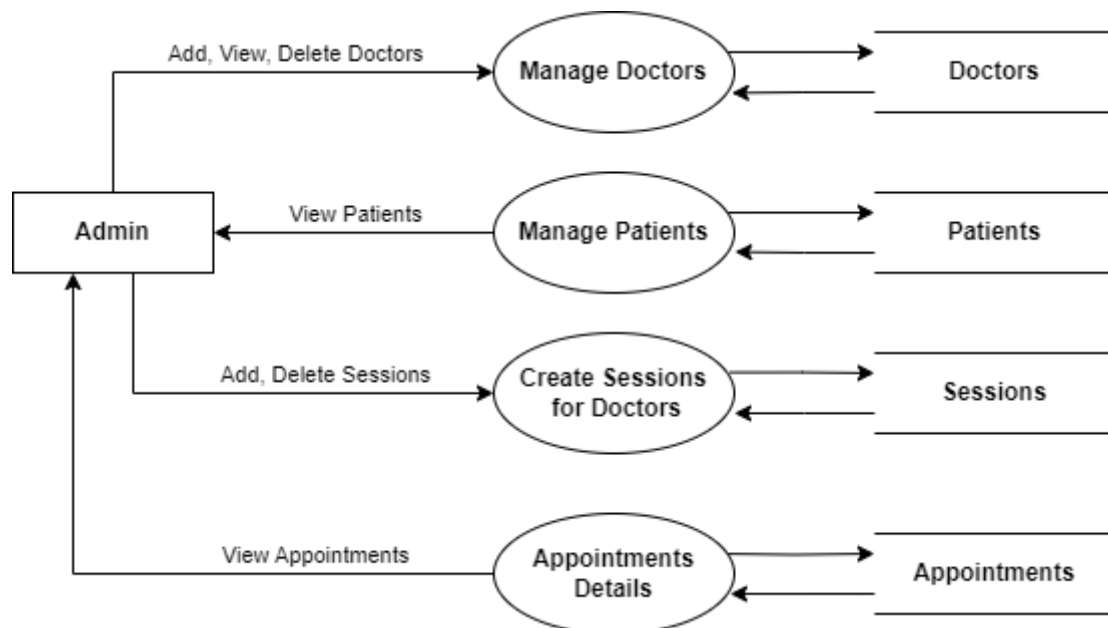


Fig 4.4.2: DFD for Admin

4.4.3 DFD FOR DOCTOR

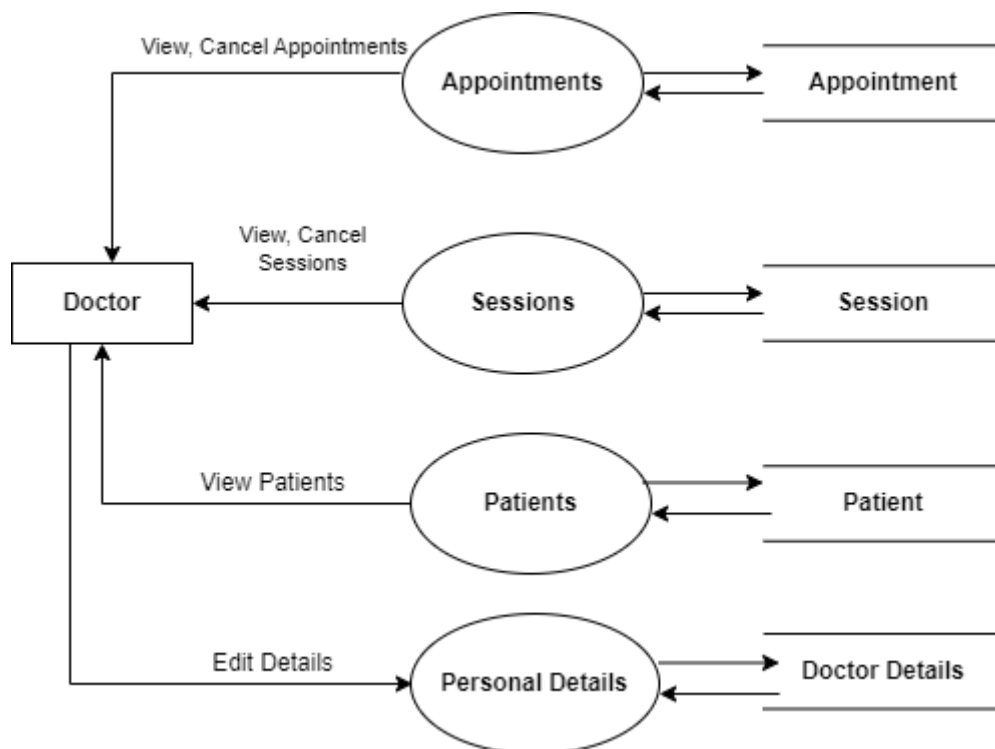


Fig 4.4.3: DFD for Doctor

4.4.3 DFD FOR PATIENT

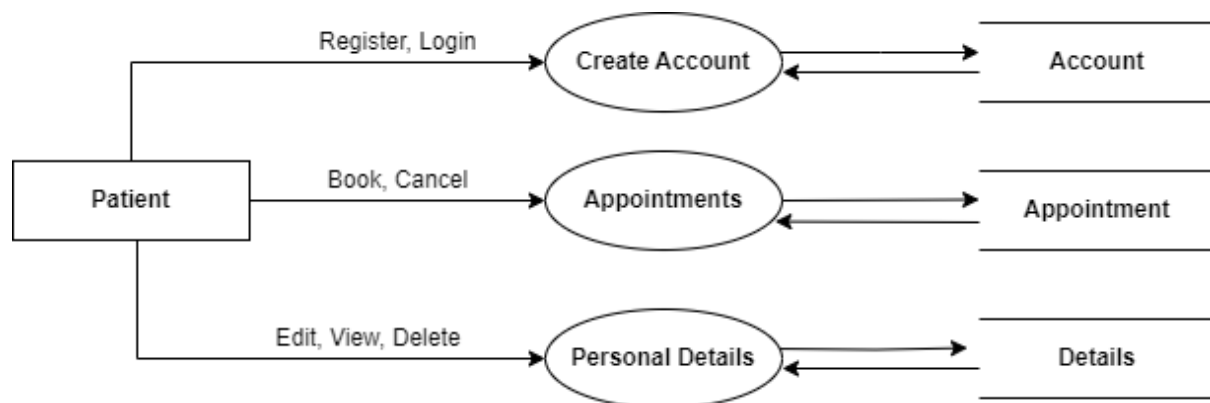


Fig 4.4.3: DFD for Patient

CHAPTER 5: DETAILED DESIGN

5.1 USE CASE DIAGRAM

A Use Case Diagram of the e-Doc Appointment System graphically represents interactions of Actors i.e., Admin and Users who are interacting with the system.

System: The system is depicted as a rectangle which is nothing but an E-Doc Appointment System website.

Actor: Each actor is shown as a stick man they are Admin as well as Users.

Use case: The use case is represented by an oval shape which is nothing but the operations performed on the E-Doc Appointment System Website.

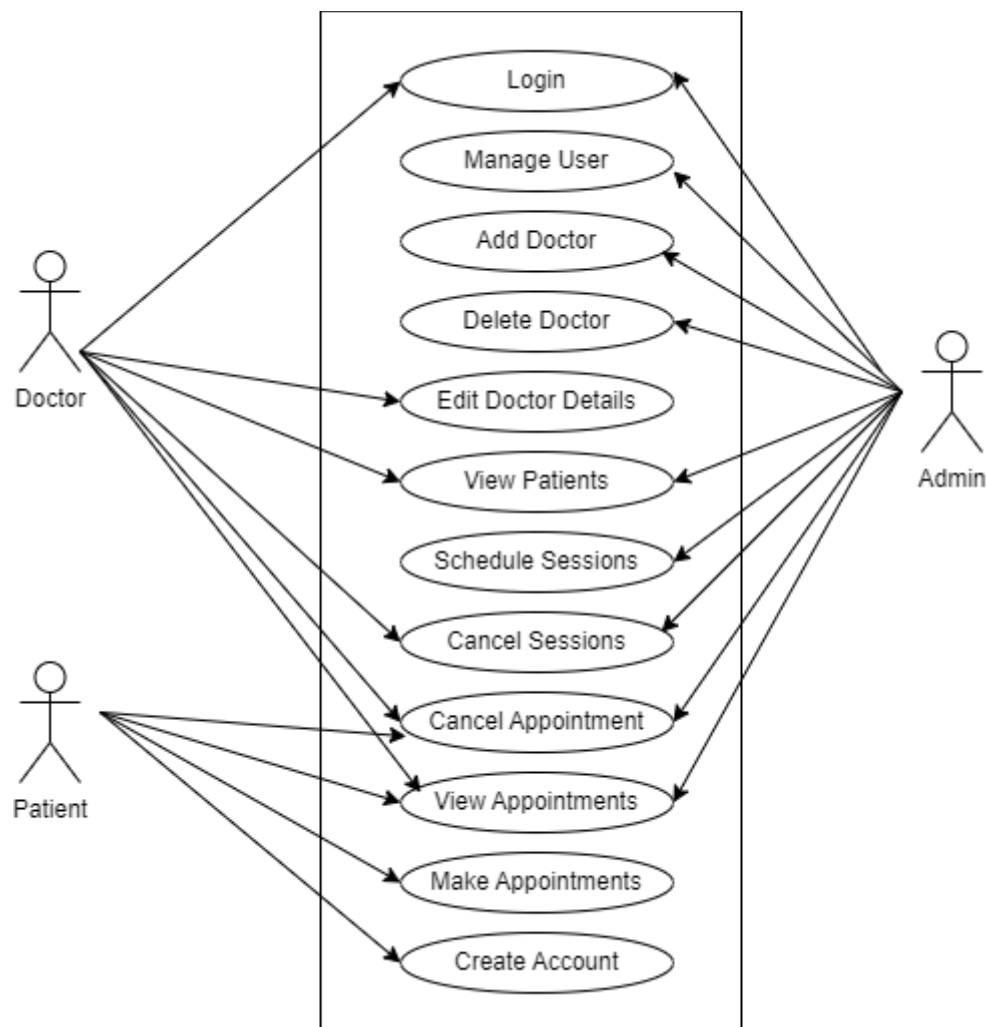


Fig 5.1: Use Case Diagram for E-Doc Appointment system

5.2 SEQUENCE DIAGRAM

The sequence Diagram shows the interaction between the objects according to the time. It is used to show the communication between the objects and the messages exchanged between them. The sequence diagram is also called as an event diagram.

5.2.1 SEQUENCE DIAGRAM FOR ADMIN

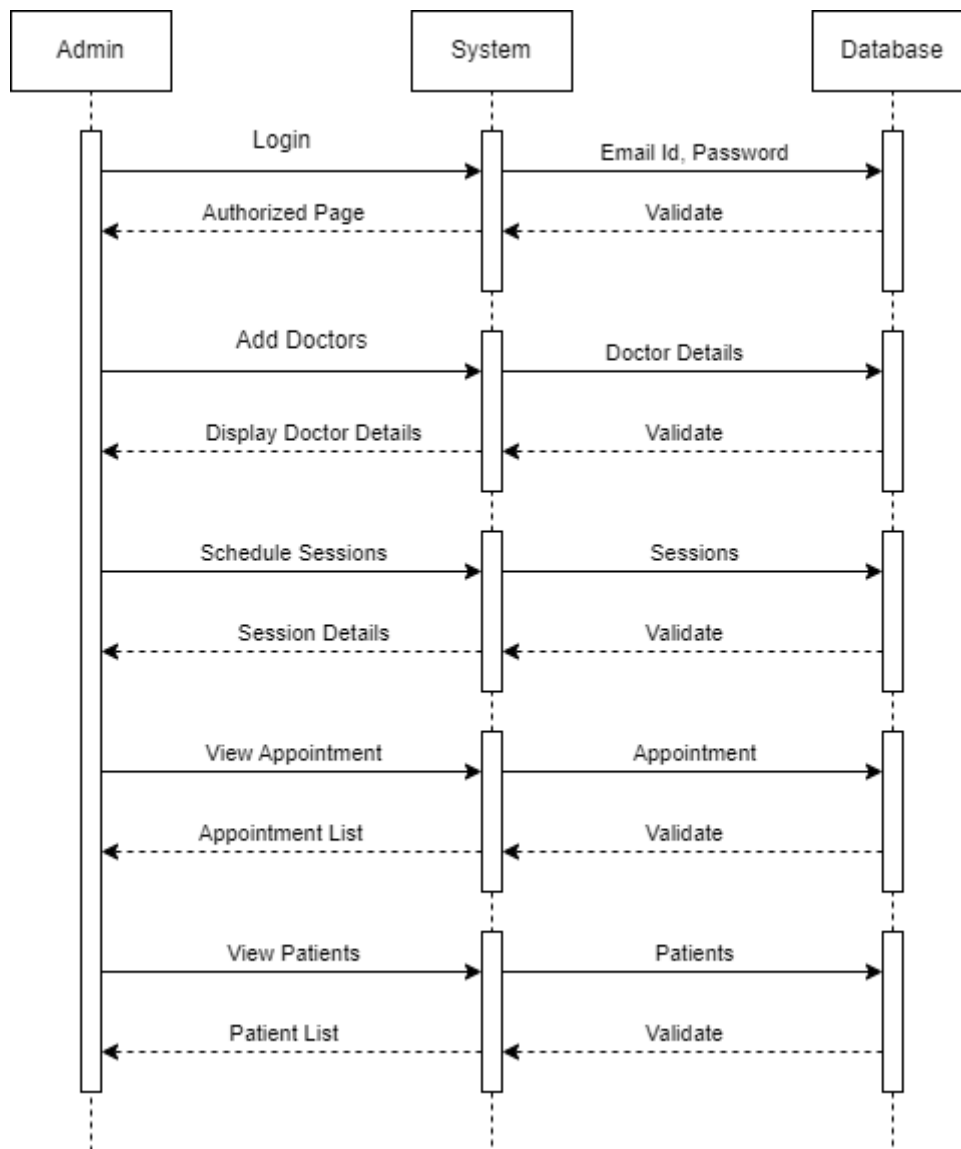


Fig 5.2.1: Sequence Diagram for Admin

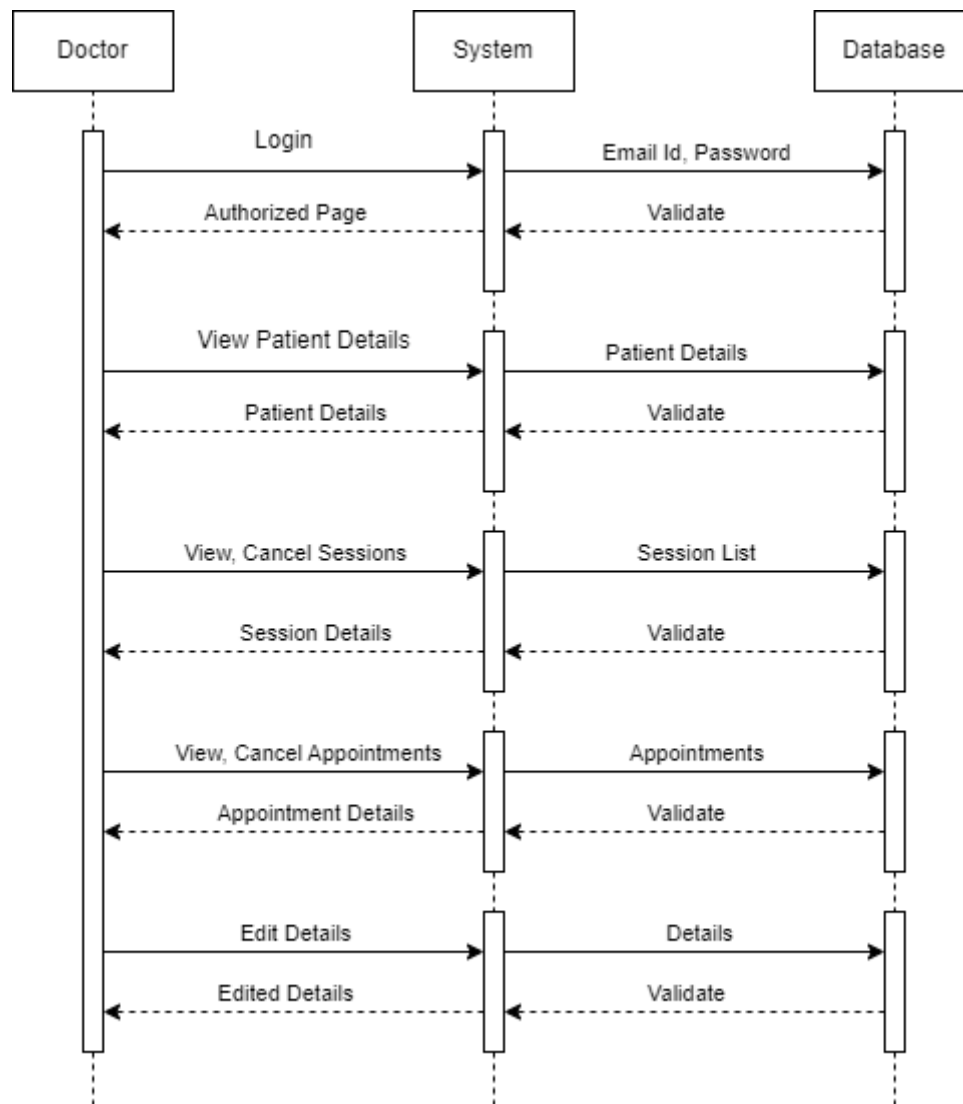
5.2.2 SEQUENCE DIAGRAM FOR DOCTOR

Fig 5.2.2: Sequence Diagram for Doctor

5.2.3 SEQUENCE DIAGRAM FOR PATIENT

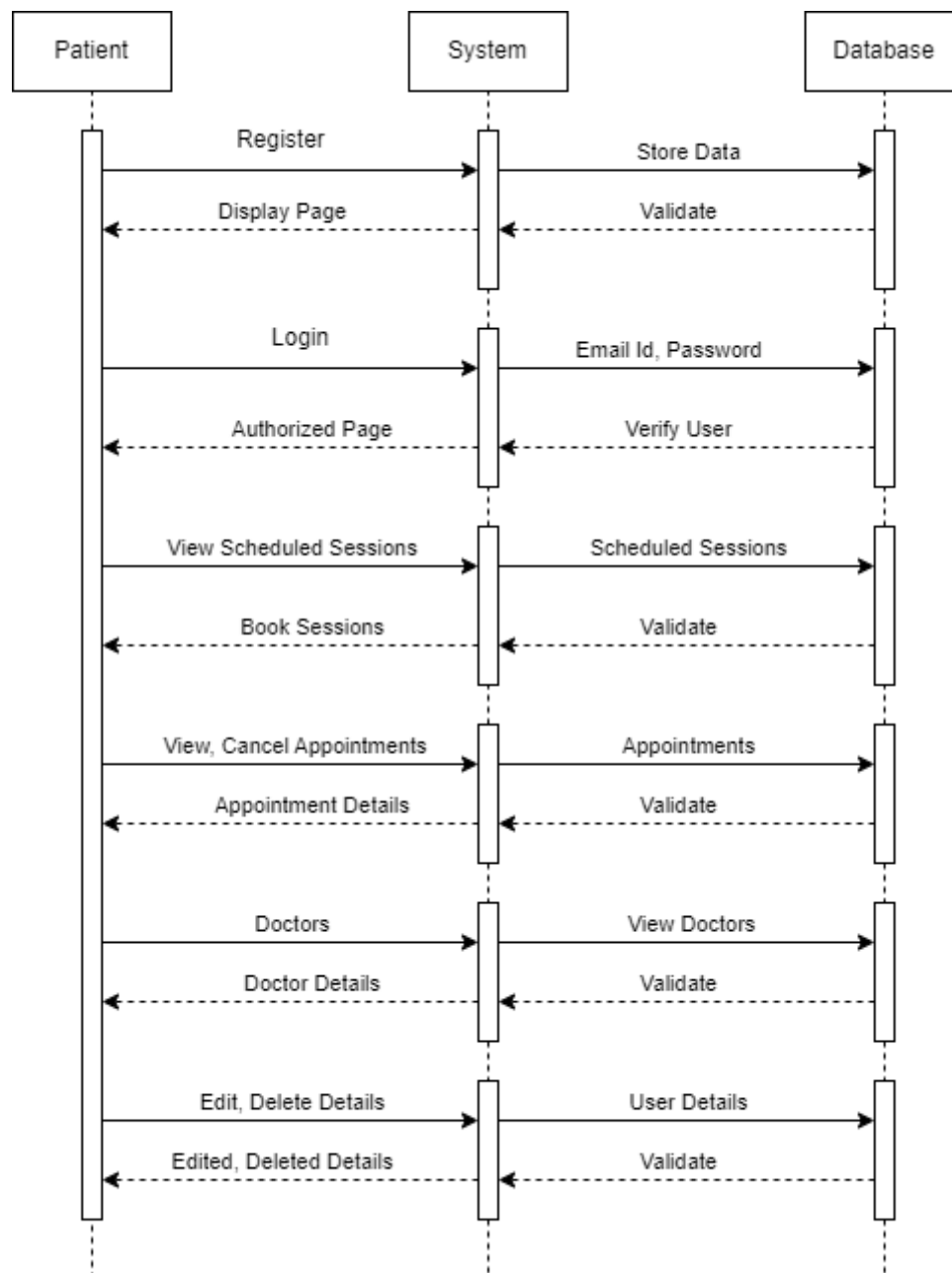


Fig 5.2.3: Sequence Diagram for Patient

5.3 ACTIVITY DIAGRAMS

Activity Diagram is a kind of flowchart in which it represents the flow of an activity to a new activity. The flow of the activity is also known as the process of the system, the way in which the system works.

5.3.1 ADMIN, DOCTOR, AND PATIENT OPERATIONS

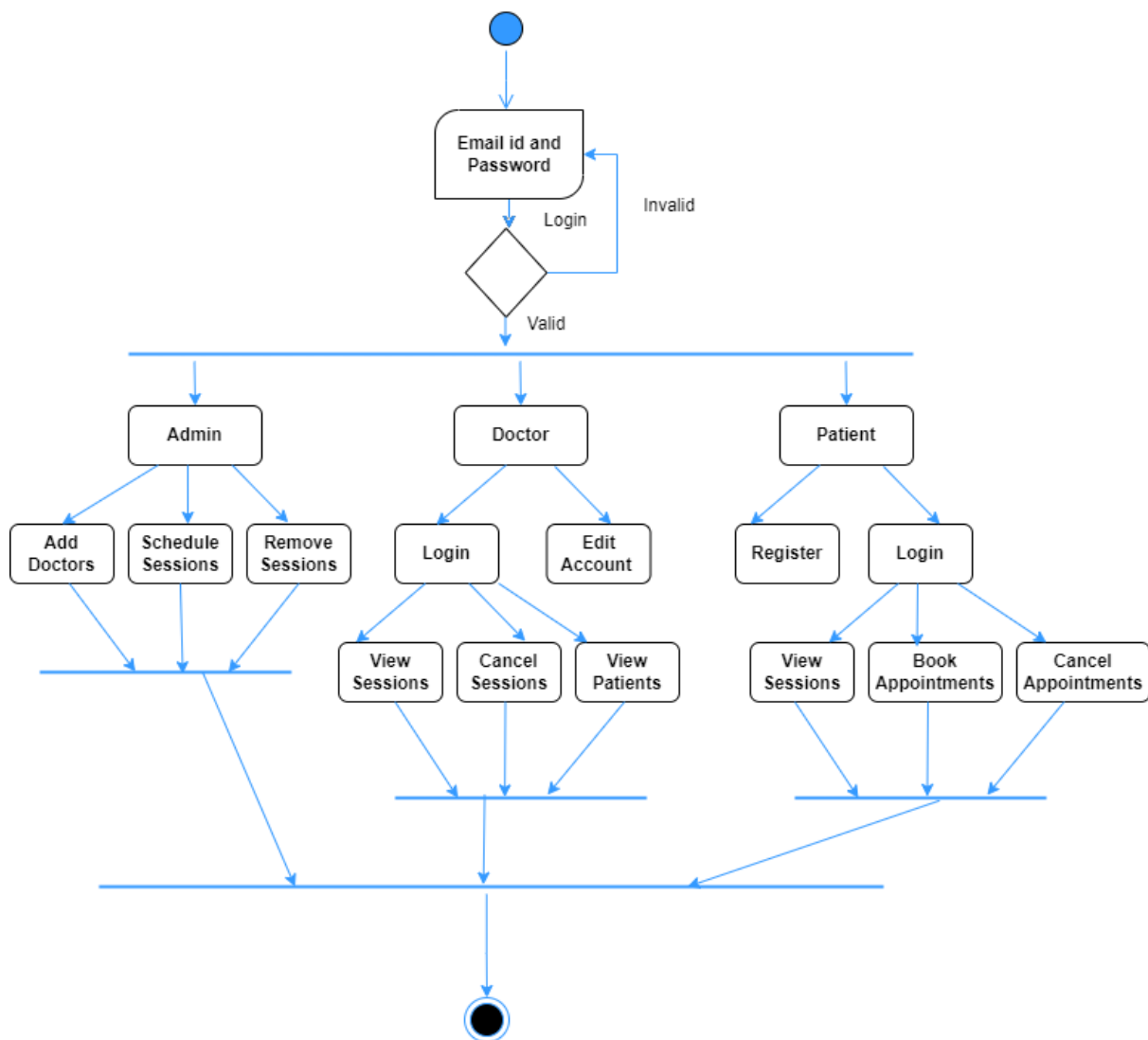


Fig 5.3.1: Activity Diagram for E-Doc Appointment System

5.4 DATABASE DESIGN

The Database of E-Doc Appointment System all together includes many tables, among which the most important tables relevant to the system are shown here.

Table 5.1: Admin Table			
Attribute	Data Type	Constraints	Description
aemail	Varchar	Primary Key	Admin Email Id
apassword	Varchar	Not Null	Admin Password

Table 5.2: Doctor Table

Attribute	Data Type	Constraints	Description
docid	Varchar	Primary Key	Doctor id
specialities	int	Foreign Key	Specialties
docemail	Varchar	Not Null	Doctor Email Id
docname	Varchar	Not Null	Doctor Name
docpassword	Varchar	Not Null	Doctor Password
doctel	Varchar	Not Null	Doctor Phone No.

Table 5.3: Patient Table

Attribute	Data Type	Constraints	Description
pid	int	Primary Key	Patient id
pemail	Varchar	Not Null	Patient Email Id
pname	Varchar	Not Null	Patient Name
ppassword	Varchar	Not Null	Patient Password
paddress	Varchar	Not Null	Patient Address
pdob	Date		Date of Birth
ptel	Varchar	Not Null	Patient Phone No.

Table 5.4: Appointment Table

Attribute	Data Type	Constraints	Description
appoid	int	Primary Key	Appointment id
pid	int	Foreign Key	Patient id
scheduleid	int	Foreign Key	Schedule id
apponum	int	Not Null	Appointment No.
appodate	Date		Appointment Date

Table 5.5: Schedule Table

Attribute	Data Type	Constraints	Description
scheduleid	int	Primary Key	Schedule id
docid	Varchar	Foreign Key	Doctor id
title	Varchar	Not Null	Title

scheduledate	Date		Schedule Date
scheduletime	Time	Not Null	Appointment Date
nop	int	Not Null	Maximum No. of Patients

Table 5.6: Specialities Table

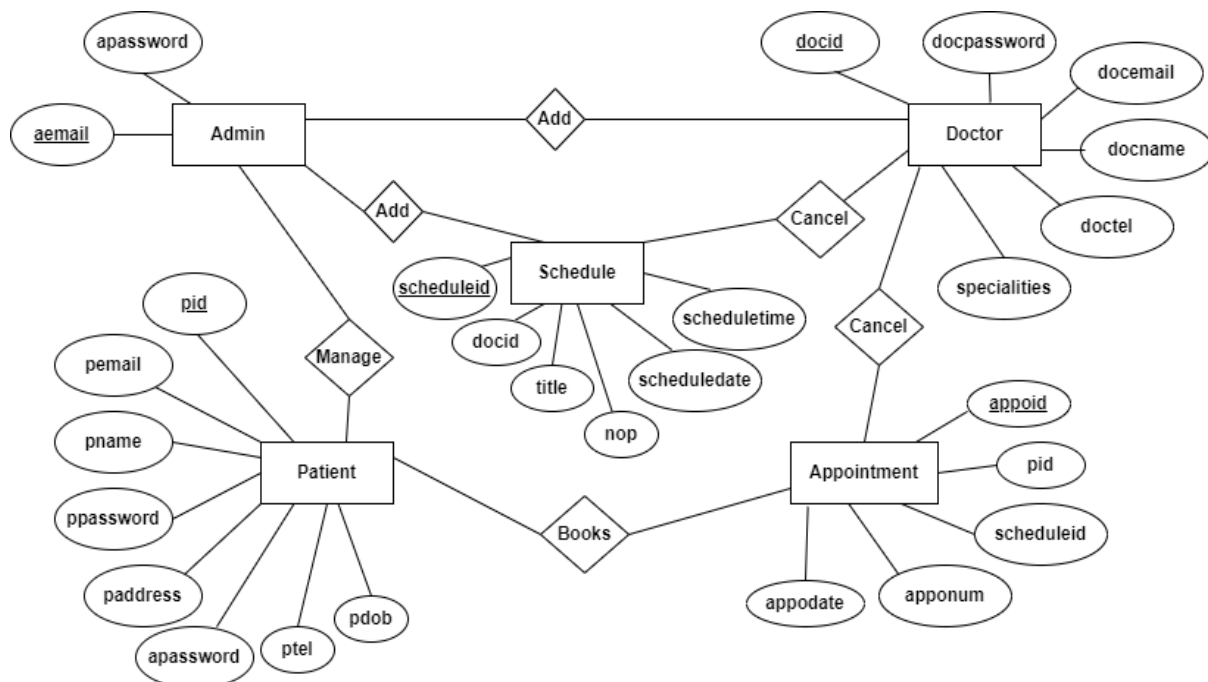
Attribute	Data Type	Constraints	Description
id	int	Primary Key	Specialities id
sname	Varchar	Not Null	Specialities name

Table 5.7: Web User Table

Attribute	Data Type	Constraints	Description
email	Varchar	Primary Key	Email Id
usertype	char	Not Null	Type of User

5.4.1 ER DIAGRAM

Here I have taken only the main entities of my project and depicts the relationship between them.

**Fig 5.4.1: ER Diagram for E-Doc Appointment System**

CHAPTER 6: IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the process of putting a plan into an execution. The implementation plan gives us the overview, brief description and task involved in the process of implementation. In implementation the different kind of test cases will be tested and then implementation into the system. Coding standards is necessary for a system because, people other than programmers can easily understand the orderly convention of the system. Standards facilitate easy maintenance of the system, as someone else other than the actual developers, can later modify or upgrade the system. The naming convention used here is similar to that of windowclasses.

6.2 PSEUDO CODES

6.2.1 PSEUDO CODE FOR REGISTRATION PAGE

Users must register to the system. While registration users need to input name, email id, address, date of birth, contact number, password. If the data is valid then user will login to user page.

Begin

Input Name, Email-id, Address, Date of Birth, Contact Number, Password, Confirm Password
Check whether data is valid and mandatory fields are filled.

If data valid, then

 Update to database

Else

 Show error

End if

6.2.2 PSEUDO CODE FOR LOGIN PAGE

The user needs to login with the Email Address and password, if the Email Address exists then we need to see whether the password matches, if the password matches then there will be successful login or else an error message will be displayed.

Begin

Input Email, Password

Check whether data is valid and mandatory fields are filled.

If Email and Password exists, then

```
    checkUserType ()  
    If user equals admin  
        showAdminHome()  
    Else if user equals doctor  
        showDoctorHome()  
    Else  
        showUserHome()  
    End If  
Else  
    Show error  
End if
```

CHAPTER 7: SOFTWARE TESTING

7.1 INTRODUCTION

Testing is a technique that is used to determine whether generated computer software is accurate, complete, and of high quality. The system's success depends on testing. Software failure detection is a key goal of testing, allowing for the identification and correction of flaws. This project is evaluated in accordance with the user's detailed requirements, and numerous validations and tests are carried out to satisfy those needs. It is offered to users who are unfamiliar with the software and its procedures so they can understand what is expected of the software. It has undergone unit and integration testing. It establishes the parameters under which the test was conducted, generates test data, and creates an expected results schedule. At first, various units are tested and then the system as a whole is tested.

7.2 TESTING OBJECTIVE

- Testing is done to make sure that this software is created with the user's needs in mind.
- Testing ensures that the programme is error-free and continues to improve its quality.
- To verify that all of the validation and verification has been completed.
- Verify that each module is implemented in accordance with the specifications.

This project involves unit testing, integration testing, functional testing, and regression testing for the programme. The project's module-by-module implementation is displayed below.

7.3 UNIT TESTING

A unit testing also known as component testing, refers to tests that verify the functionality of a specific section of code, usually at the functional level. These types of tests are usually done by developers or programmers as they work on code to ensure that the specific function is working as expected. It ensures defect prevention and detection strategies in order to reduce software development risks, time and costs. The purpose of unit testing is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output.

7.3.1 UNIT TEST CASE

Unit testing involves testing the smallest possible unit of an application. Detecting and correcting errors at this stage is simple, less time consuming and inexpensive. In this project

each and every module is tested individually with proper inputs and has got the expected outputs.

7.3.2 TESTING FOR VALID USER NAME

Test Case	Input	Text Description	Output
1.	User name starts with number	User name cannot start with number	Appropriate error
2.	User name is left blank	User name cannot be left blank	Enter user name
3.	User name entered	Checks whether the entered username and password is present in database or not	Successful

Table 7.3.2 Testing for valid Username

7.3.3 TESTING FOR VALID PASSWORD

Test Case	Input	Text Description	Output
1.	Password is left blank	Password cannot be blank	Enter password again
2.	First letter is space	No spaces allowed in the password	Incorrect password
3.	If Invalid password entered	Valid password must be entered	Password mismatch
4.	Valid password entered	Password matches	Password accepted successful

Table 7.3.3 Testing for valid Password

7.3.4 TESTING FOR CHANGE PASSWORD

Test Case	Input	Text Description	Output
1.	If any field left blank	All field are entered compulsory	Appropriate error message generated
2.	Invalid password	Valid password must be entered	Appropriate error message generated
3.	If retyped password does not match	Retyped password must match	Appropriate error message generated

4.	Valid input	Valid input	Password changed
----	-------------	-------------	------------------

Table 7.3.4 Testing for change Password

7.3.5 TESTING FOR PHONE NUMBER

Test Case	Input	Text Description	Output
1.	Phone number entered with alphabets	Phone number cannot have alphabet	Appropriate error message generated
2.	Phone number entered with spaces	Phone number cannot have spaces	Appropriate error message generated
3.	Valid Phone number	Valid Phone number	Number accepted

Table 7.3.5 Testing for Phone number

7.3.5 TESTING FOR EMAIL ADDRESS

Test Case	Input	Text Description	Output
1.	Email address with an @ symbol	Email address must have @ symbol	Appropriate error message generated
2.	Email address entered with spaces	Email address cannot have spaces	Appropriate error message generated
3.	Email address without above faults	Valid email address	Email address accepted

Table 7.3.5 Testing for Email address

7.4 INTEGRATION TESTING

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. In this project it is considered better since it allows interface issues to be localized more quickly and fixed. Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively large groups of tested components corresponding to elements of the Architectural design are integrated and tested until the software works as a system.

7.4.1 INTEGRATION TEST CASES

In an application one module can have adverse effect on other module, when combined, which may not produce the desired results. So, in this project all the modules are integrated together and tested as a whole.

No	Test Cases	Expected Output	Observed Output	Result
1.	Mandatory fields	Mandatory fields cannot be left empty	Result as expected	Pass
2.	Schedule dates	Schedule dates must be upcoming dates	Result as expected	Pass

Table 7.4.1: Integration Test Cases for Data Insertion

7.4.2 OTHER CASES:

Test Cases	Input	Test Description	Output
1.	Click on Logout	Application will close	Application Closes
2.	Click on Reset	Entering Details	Record is not submitted to the database and text boxes gets cleared
3.	Click on save	Entering Details	Record is submitted to the database
4.	Click on save	Not Entering the Details	Appropriate error message displayed
5.	Click on drop down list	Selecting data	Data is displayed

CHAPTER 8: CONCLUSION

In conclusion, the E-Doc appointment system provides a simple and effective way to manage appointments across a variety of industries. Real-time updates, automated scheduling, and a simple user interface improve both the user and administrative experiences. The solution significantly improves overall productivity and client satisfaction by minimizing manual effort, decreasing scheduling conflicts, and optimizing resource allocation. As technology develops, organizations looking to offer efficient and easy services must adopt such digital appointment systems. The E-Doc appointment system is evidence of the beneficial effects of incorporating technology into conventional appointment procedures, and it represents an important step towards a more organized and smooth future.

This project was successfully tested and implemented under Windows 10 and above. In E-Doc Appointment System admin logs in and manages all the data. Admin can manage Doctor details, Add Sessions, delete doctor details, and delete scheduled sessions. Thus, it is useful for the admin so that the paperwork is reduced. The doctor should log in to the system using the Email Address and password. Doctors can edit their personal details, they can cancel appointments, and also, they can cancel sessions generated through admin. Patients should register themselves and log in to book an appointment. Since it's an economic friendly and helps to store the data securely.

CHAPTER 9: FUTURE ENHANCEMENTS

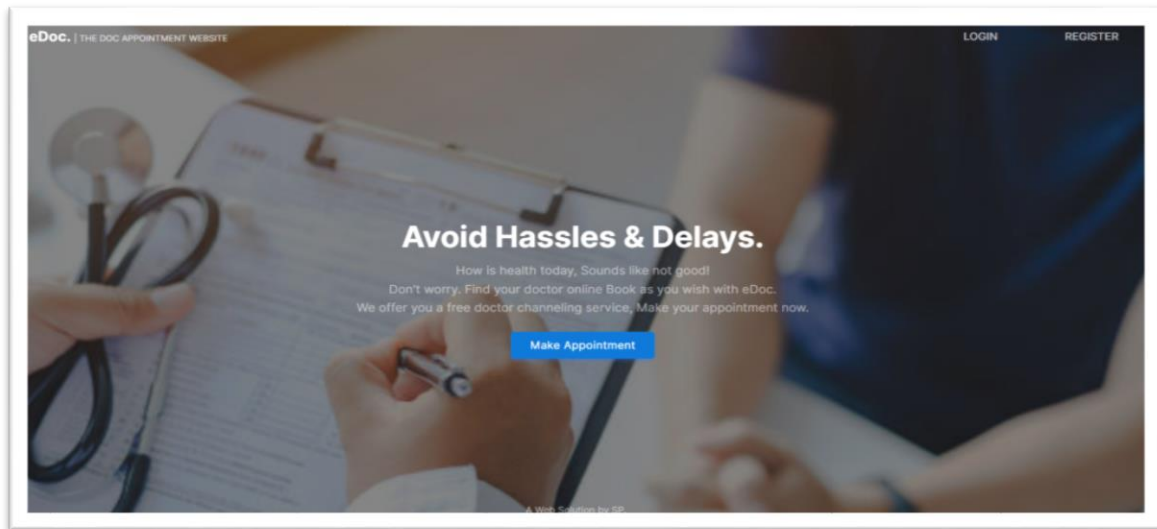
The E-Doc appointment system might give customers a synchronized and efficient experience by enabling seamless interaction with popular external calendars like Google Calendar and Outlook. Users would benefit from the convenience of easily syncing their appointments across all of their devices with this improvement, ensuring that they stay current regardless of their preferred platform. The mobile app development would advance unmatched ease of use and versatility by enabling users to manage their appointments straight from their cell phones. Users would be given the freedom to easily make, change, or cancel appointments while on the go, free from the limitations of conventional desktop-based interactions. A beneficial approach to improve user involvement and service improvement would be to include a thorough feedback and rating system in the E-Doc appointment system. Through allowing users to exchange their opinions and past experiences. The E-Doc appointment system has the potential to revolutionize how transactions are conducted by easily connecting with dependable payment channels, adding a new level of convenience and security. Users and businesses alike would gain from a streamlined payment procedure with this improvement, enabling quick and easy settlement for services provided during visits. By incorporating safe online payment options, manual currency exchanges, and subsequent follow-ups would no longer be necessary, saving time and reducing the possibility of mistakes.

APPENDIX A: REFERENCES

- [1] Hassan, Md Arif, et al. "Enhanced Security of User Authentication on Doctor E-Appointment System." *Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2021*. Singapore: Springer Nature Singapore, 2022. 47-63.
- [2] Jadhav, Prakash, et al. "Smart Pulmonary System with Doctor Appointment."
- [3] Zhang, Min, et al. "The Impact of Narrative Reviews on Patient E-doctor Choice in Online Health Communities." *INQUIRY: The Journal of Health Care Organization, Provision, and Financing* 60 (2023): 00469580231183695.
- [4] Rawabdeh, Ali Ahamd Awad. "An e-health trend plan for the Jordanian health care system: a review." *International journal of health care quality assurance* 20.6 (2007): 516-531.
- [5] Nixon (Robin), *Learning PHP MySQL and JavaScript*,
Publication: Shroff publishers & Distributors Pvt Pankaj Jalote Ltd.
- [6] <https://www.google.com/>
- [7] <https://www.wikipedia.org/>
- [8] <https://www.w3schools.com/>

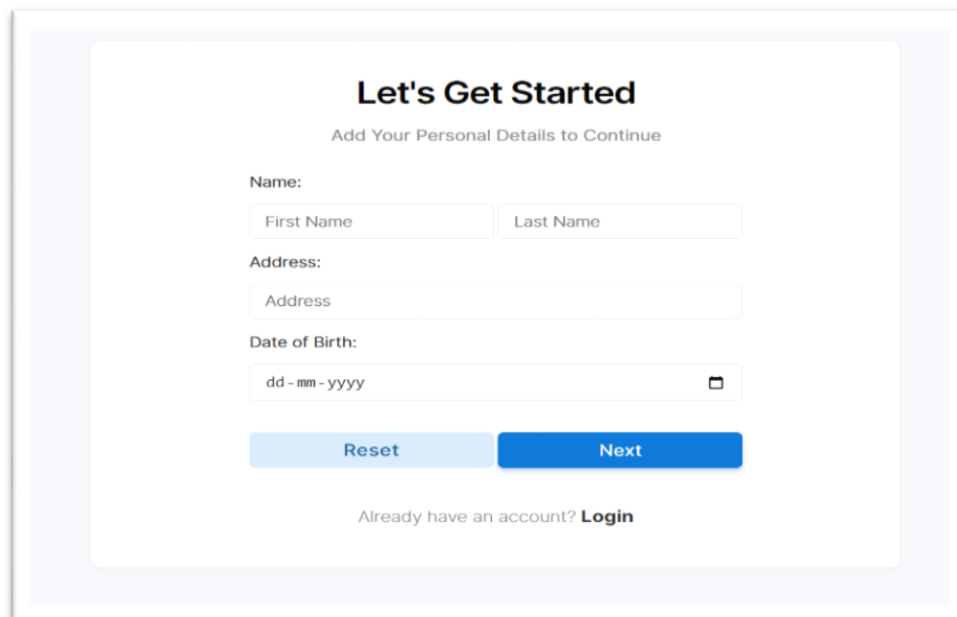
APPENDIX B: USER MANUAL

1. MAIN PAGE



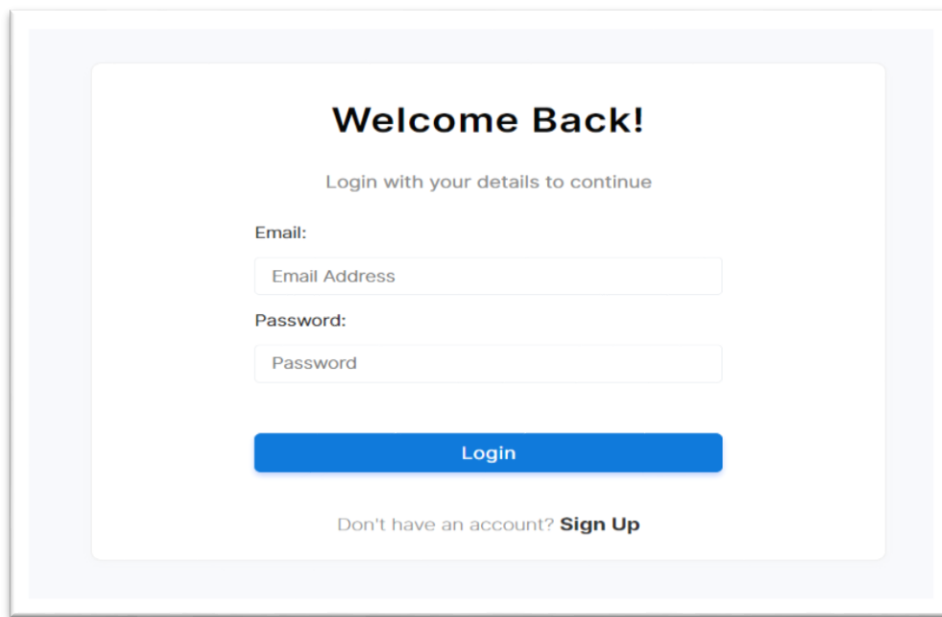
Screenshot 1: Main Page

2. REGISTER PAGE

A screenshot of the eDoc registration page. The page has a light blue background. The main heading is "Let's Get Started" with the subtext "Add Your Personal Details to Continue". The form fields are: "Name:" with two input boxes for "First Name" and "Last Name"; "Address:" with one input box for "Address"; and "Date of Birth:" with one input box for "dd - mm - yyyy" and a calendar icon. Below the form are two buttons: "Reset" (light blue) and "Next" (dark blue). At the bottom, there is a link: "Already have an account? Login".

Screenshot 2: Register Page

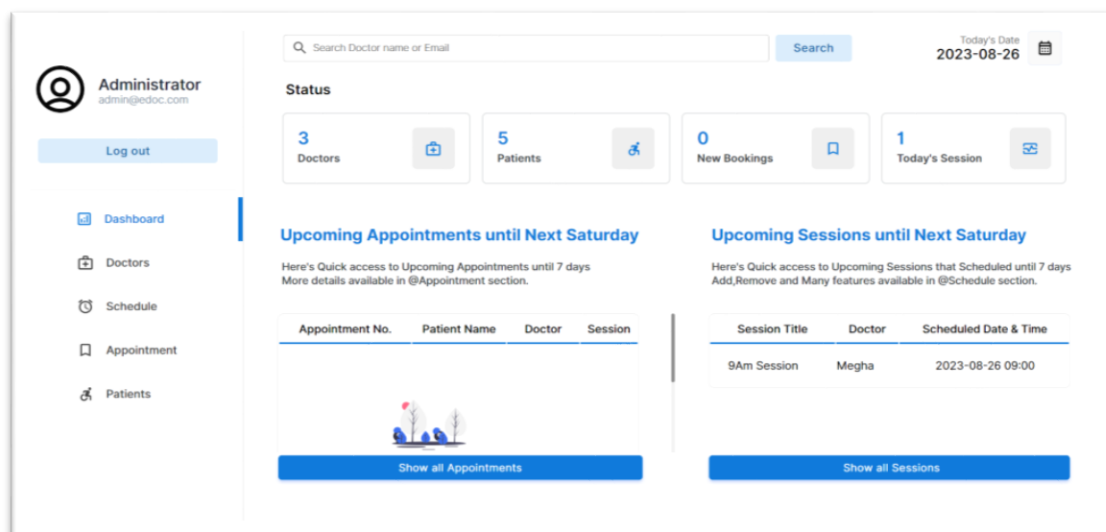
3. LOGIN PAGE



The login page features a central white card on a light blue background. At the top, it says "Welcome Back!" in bold. Below this, it prompts the user to "Login with your details to continue". There are two input fields: "Email:" with a placeholder "Email Address" and "Password:" with a placeholder "Password". A blue "Login" button is positioned below the password field. At the bottom, it says "Don't have an account? Sign Up".

Screenshot 3: Login Page

4. ADMIN DASHBOARD



The admin dashboard is a comprehensive interface for managing the appointment system. It includes a sidebar with navigation links: Dashboard, Doctors, Schedule, Appointment, and Patients. The main content area is divided into several sections:

- Status:** A row of four cards showing key metrics: 3 Doctors, 5 Patients, 0 New Bookings, and 1 Today's Session.
- Upcoming Appointments until Next Saturday:** A section with a quick access note and a table of upcoming appointments. Below the table is a "Show all Appointments" button.
- Upcoming Sessions until Next Saturday:** A section with a quick access note and a table of upcoming sessions. Below the table is a "Show all Sessions" button.

The top of the dashboard features a search bar for "Search Doctor name or Email" and a "Search" button. The date "Today's Date 2023-08-26" is displayed in the top right corner.

Appointment No.	Patient Name	Doctor	Session
[Illustration of a doctor and patient]			

Session Title	Doctor	Scheduled Date & Time
9Am Session	Megha	2023-08-26 09:00

Screenshot 4: Admin Dashboard

5. ADD NEW DOCTOR (ADMIN)

Add New Doctor.

Name:
Doctor Name

Email:
Email Address

Telephone:
Telephone Number

Choose specialties:
Accident and emergency medicine

Password:
Define a Password

Confirm Password:
Reconfirm Password

Screenshot 5: Add New Doctor (Admin)

6. ADD NEW SESSION (ADMIN)

Add New Session.

Session Title :
Name of this Session

Select Doctor:
Choose Doctor Name from the list

Number of Patients/Appointment Numbers :
The final appointment number for this session depends on this number

Session Date:
dd - mm - yyyy

Schedule Time:
-- : --

Reset Place this Session

Screenshot 6: Add New Session (Admin)

7. VIEW APPOINTMENT (ADMIN)

Appointment Manager

All Appointments (5)

Date: dd - mm - yyyy Doctor: Choose Doctor Name from the list Filter

Patient name	Appointment number	Doctor	Session Title	Session Date & Time	Appointment Date	Events
Test Patient1	1	Test Doctor	Morning Session	2023-07-30 09:30	2023-07-29	Cancel

Screenshot 7: View Appointment (Admin)

8. VIEW PATIENT (ADMIN)

Name	Telephone	Email	Date of Birth	Events
David Alex	9638547210	david@gmail.com	1999-06-26	View
Soemya Arya	9876543210	soemya@gmail.com	1999-09-04	View
Anushree Sanjay	8575455122	anush@gmail.com	1998-09-21	View
Sahana .	9663548212	sahana@gmail.com	2000-07-19	View
Test Patient1	9638527410	patient@edoc.com	2000-01-01	View

Screenshot 8: View Patient (Admin)

9. DOCTOR DASHBOARD

Session Title	Scheduled Date	Time
9Am Session	2023-08-26	09:00

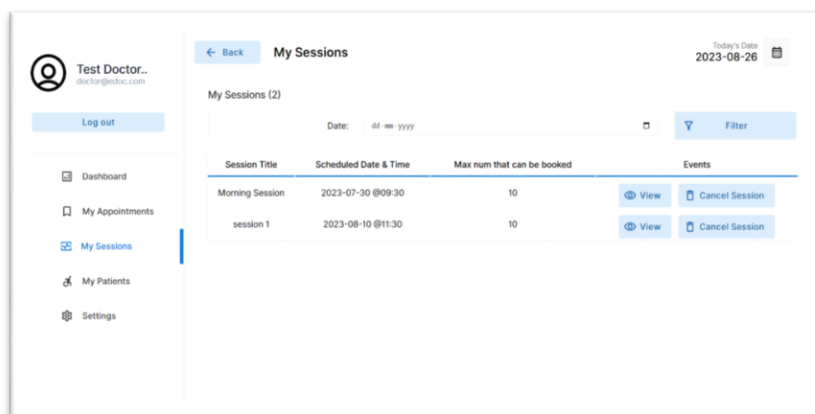
Screenshot 9: Doctor Dashboard

10. VIEW APPOINTMENT (DOCTOR)

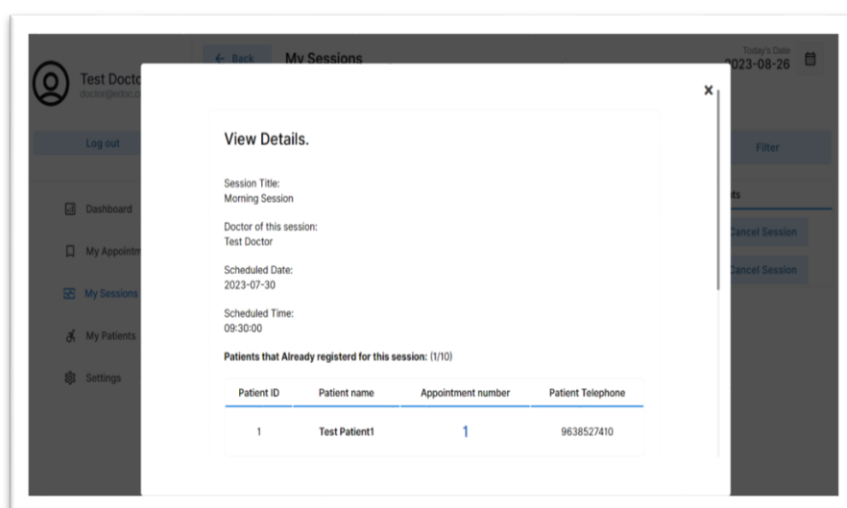
Patient name	Appointment number	Session Title	Session Date & Time	Appointment Date	Events
Test Patient1	1	Morning Session	2023-07-30 @09:30	2023-07-29	Cancel

Screenshot 10: View Appointment (Doctor)

11. VIEW SESSIONS (DOCTOR)

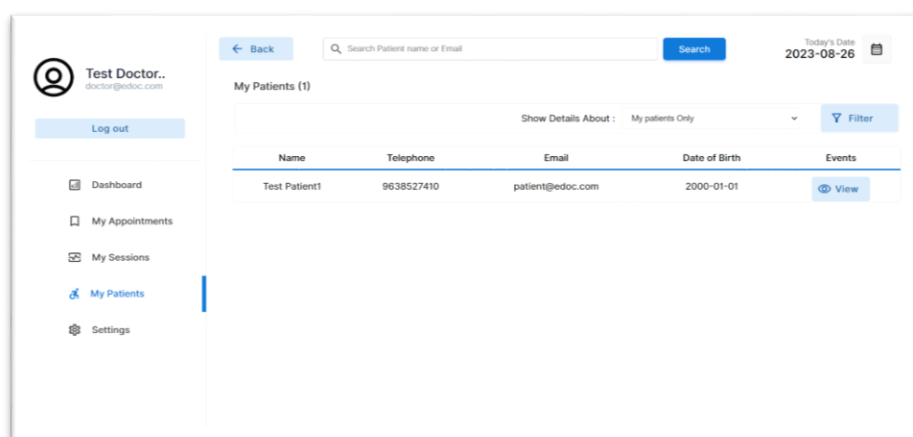


Screenshot 11.1: View Sessions (Doctor)

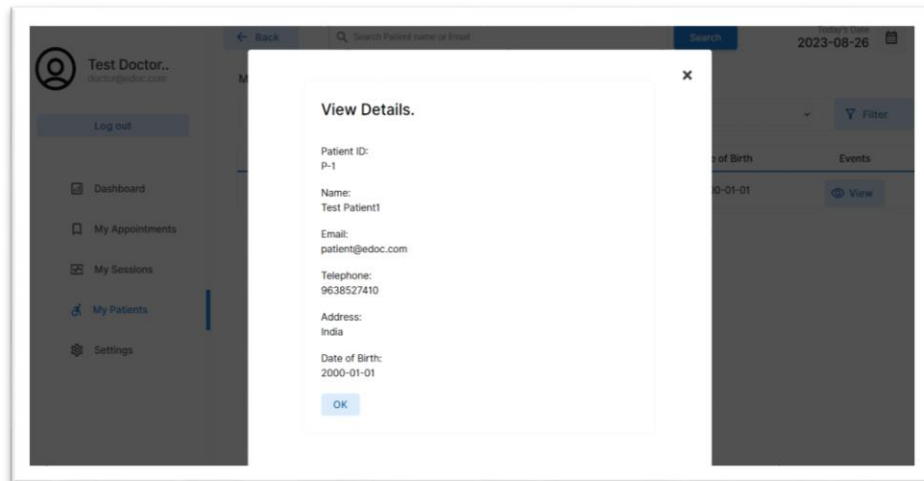


Screenshot 11.2: View Session Details (Doctor)

12. VIEW PATIENTS (DOCTOR)

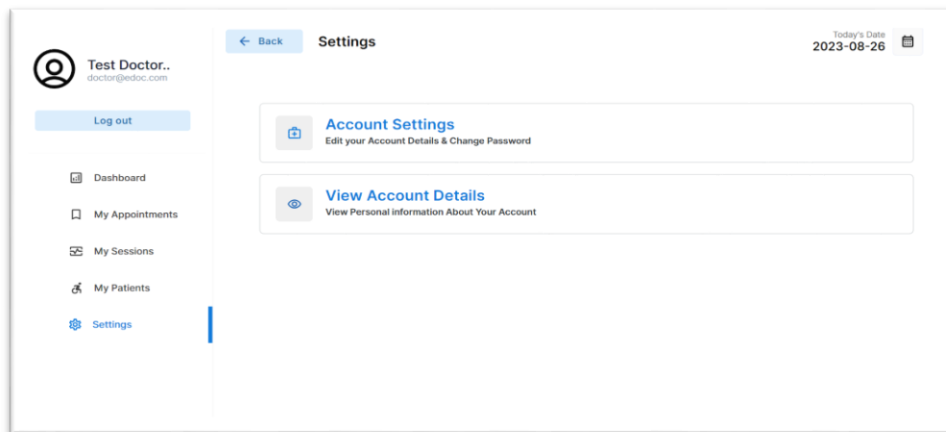


Screenshot 12.1: View Patients (Doctor)

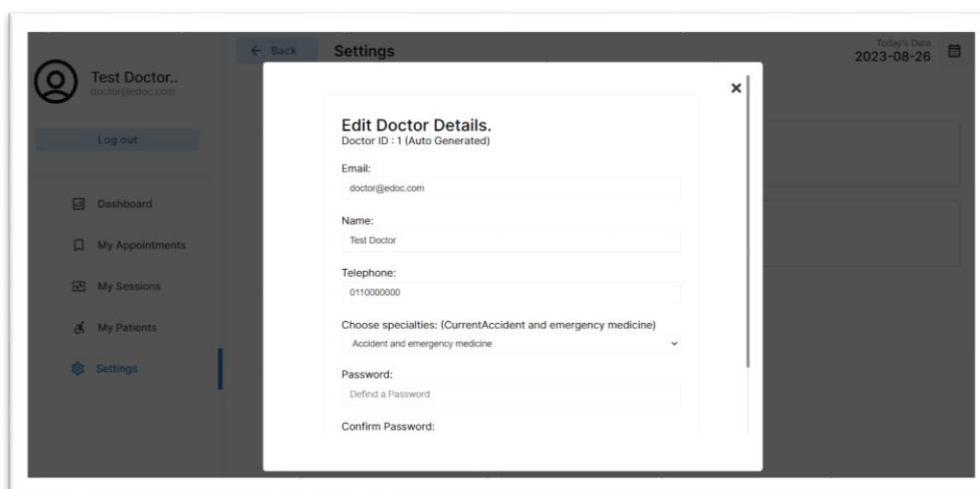


Screenshot 12.2: View Patient Details (Doctor)

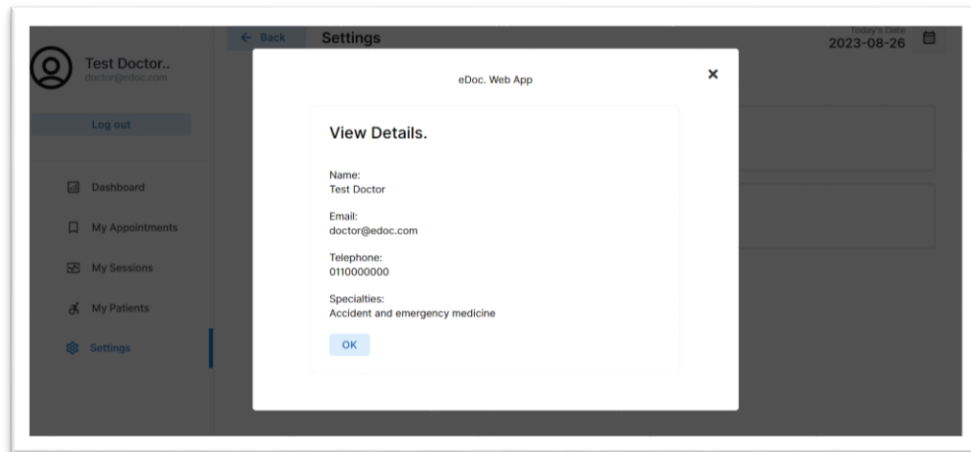
13. SETTINGS (DOCTOR)



Screenshot 13.1: Settings (Doctor)

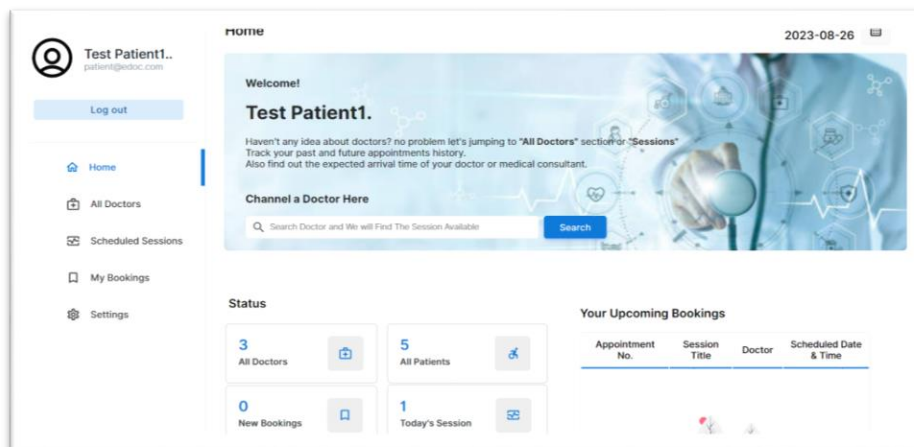


Screenshot 13.2: Edit Doctor Details (Doctor)



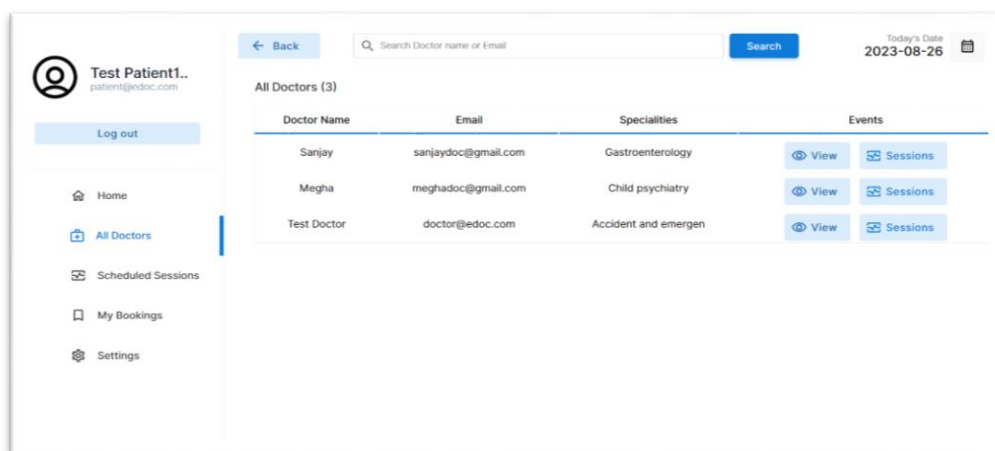
Screenshot 13.3: View Details (Doctor)

14. PATIENT DASHBOARD



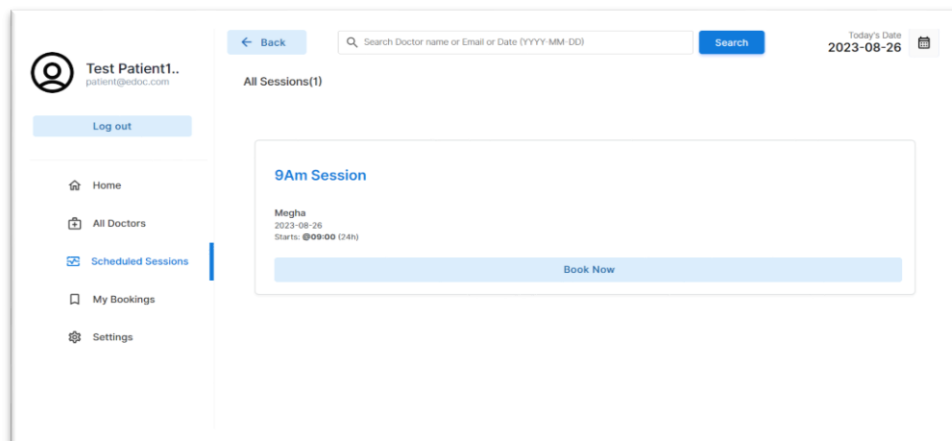
Screenshot 14: Patient Dashboard

15. VIEW DOCTORS (PATIENT)

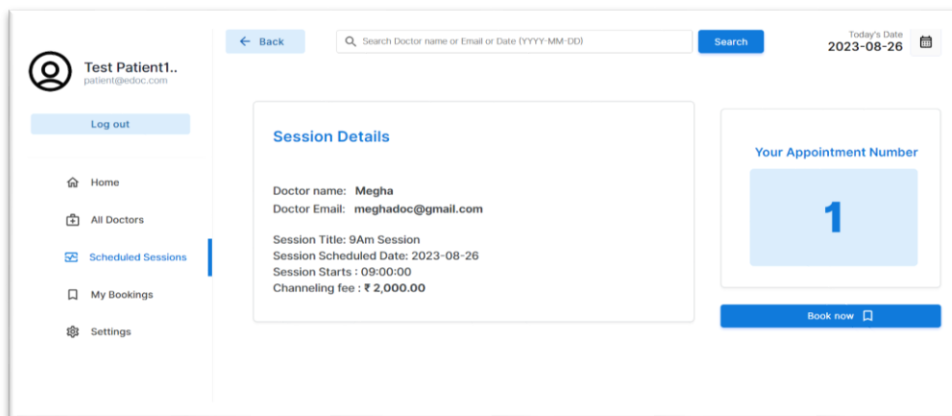


Screenshot 15: View Doctors (Patient)

16. VIEW SESSIONS (PATIENT)

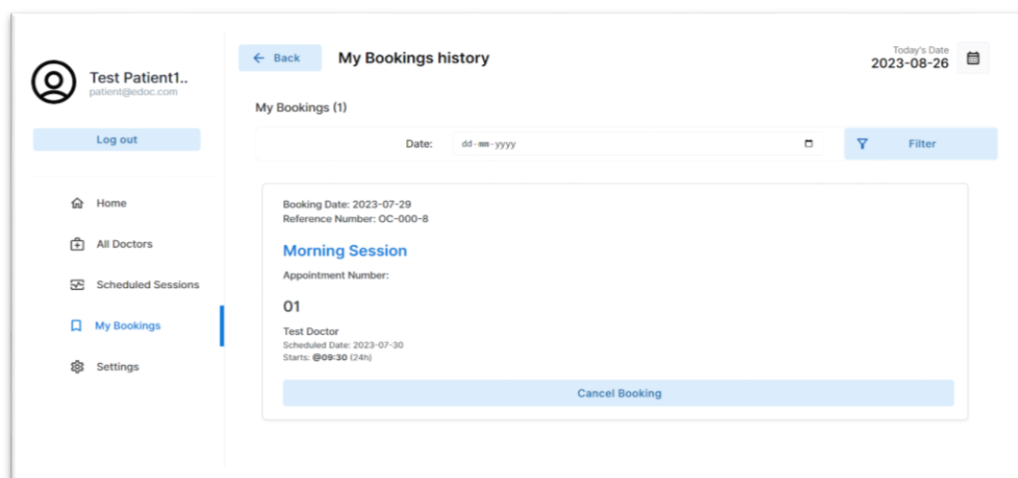


Screenshot 16.1: View Sessions (Patient)



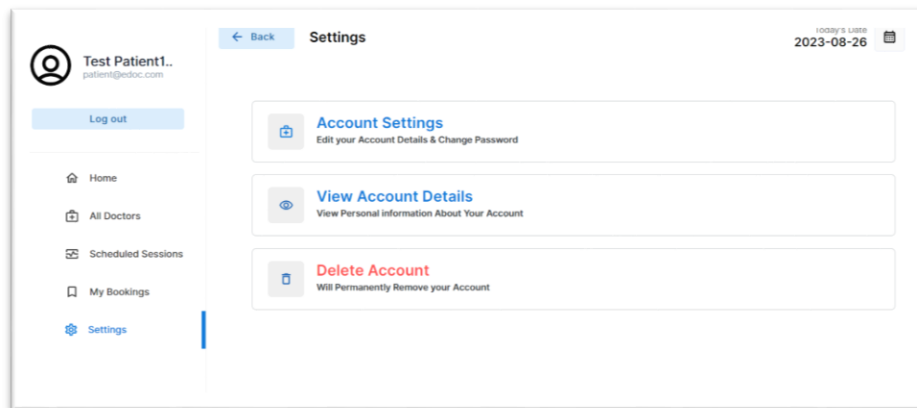
Screenshot 16.2: View Session Details (Patient)

17. VIEW BOOKINGS (PATIENT)

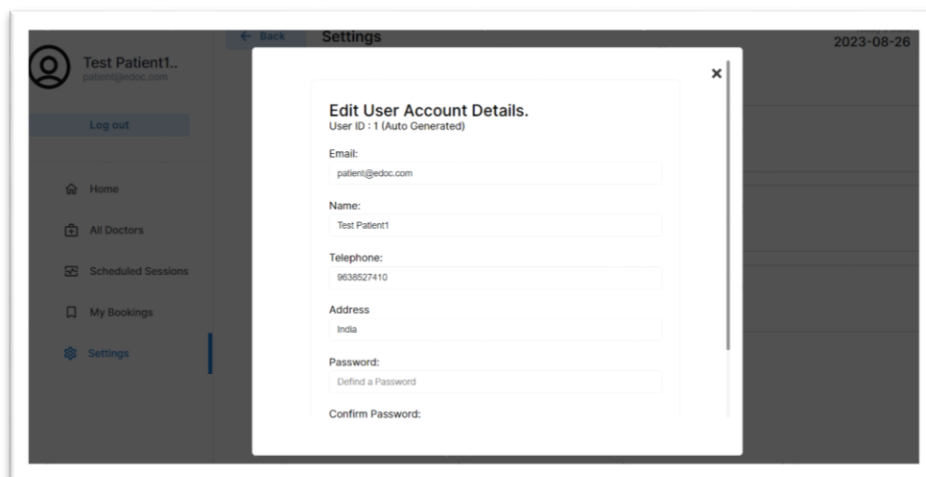


Screenshot 17: View Bookings (Patient)

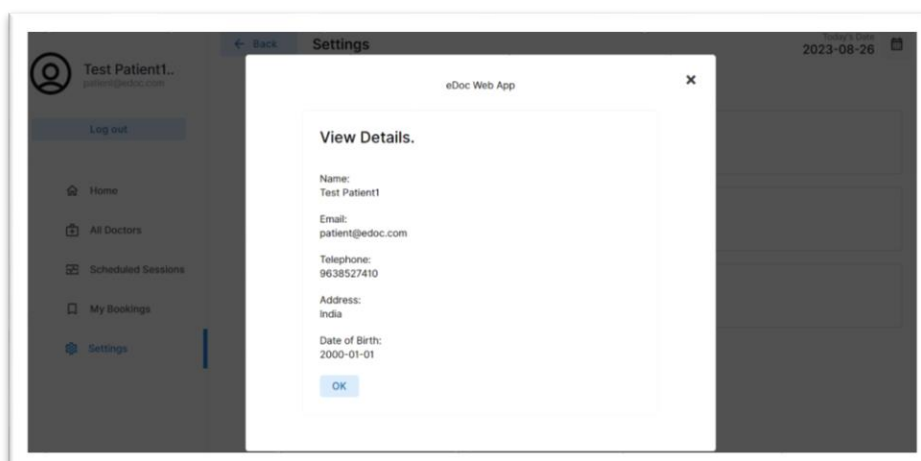
18. SETTINGS (PATIENT)



Screenshot 18.1: Settings (Patient)



Screenshot 18.2: Edit User Details (Patient)



Screenshot 18.3: View Details (Patient)