

**PROJECT REPORT**  
**Doctor to Door Step**

At  
**SGMS Infotech LLP**  
**Pimpri-Chinchwad, Maharashtra 411044**

By  
**Vaibhav Vanire**  
**Seat no 12853**  
**MCA – II, SEM – IV**  
**A.Y. 2023-2025**

Under The Guidance Of  
**Prof. Arati Kadam**

Submitted To  
**Savitribai Phule Pune University**

**In partial fulfillment of the requirement for the**  
**award of the degree of Master of Computer**  
**Application (MCA)**  
**Batch 2023 25**

Through  
**Suryadatta Education Foundation's Suryadatta Institute of Management &**  
**Mass Communication (SIMMC) Pune-411021**

Date: -

## **PROJECT COMPLETION CERTIFICATE**

This is to certify that Mr. Vaibhav Vanire, of MCA II Year at SIMMC has successfully completed his project “Doctor to Door Step” to my satisfaction and as per the requirements of the two-year full-time MCA program (2023-2025) of the Savitribai Phule Pune University.

**Prof. Arati Kadam**

**(Project Guide)**

**HoD-MCA,SGI**

**Examiner 1**

**Examiner 2**

# **DECLARATION**

I hereby declare that the project titled **Doctor to Door Step**, is an original piece of work carried out by me under the guidance of **Arati Kadam**.

All the necessary information, facts and figures contained herein has been collected from genuine and authentic sources.

The information provided herein below is true and correct to the best of my knowledge and belief and nothing has been falsely stated or concealed therein.

The work has been submitted in partial fulfillment of the requirement of degree Master of Computer Application (MCA) to Savitribai Phule Pune University.

Vaibhav Vanire

Seat No: 12853

Date:

Place:

# **ACKNOWLEDGEMENT**

I am glad to take this opportunity to acknowledge to all those who helped me in designing, developing and successful execution of my project on **Doctor to Door Step**.

I would like to extend my thanks and gratitude to my project guide **Prof.Arati Kadam and Dr. vidya Gavekar Project Co-ordinator** for their valuable guidance and timely assistance throughout the development of this project.

Furthermore, I would like to express my appreciation to **Dr. Manisha Kumbhar**, our Head of Department, whose help and support this project would not have been possible. I am also grateful to **Dr. Shailesh Kasande, CEO, Group Director of our institute** for his valuable time and suggestions in the success of our project.

I also show my extended gratefulness towards SGI, **Chairman Prof Dr. Sanjay Chordiya sir** for his continues support and valuable guidance in the institute. Thank you to everyone involved for your valuable contributions and support throughout this work.

## **Thank You**

**Vaibhav Vanire**

(Seat No: 12853)

**Place: Pune**

**Date:**

# Index

<b>Chapter No.</b>		<b>Chapter</b>	<b>Page number</b>
<b>1</b>		<b>INTRODUCTION</b>	
	1.1	Company Profile / Institute Profile / Client Profile	2
	1.2	Abstract	6
	1.3	Existing System and Need for System	7
	1.4	Scope of System	9
	1.5	Operating Environment - Hardware and Software	11
	1.6	Brief Description of Technology Used 1.6.1 Operating systems used (Windows or Unix) 1.6.2 RDBMS/No Sql used to build database (mysql/ oracle, Teradata, etc.)	12
<b>2</b>		<b>PROPOSED SYSTEM</b>	
	2.1	Feasibility Study	43
	2.2	Objectives of Proposed System	46
	2.3	Users of System	48
<b>3</b>		<b>ANALYSIS &amp; DESIGN</b>	
	3.1	System Requirements (Functional and Non-Functional requirements)	51
	3.2	Entity Relationship Diagram (ERD)	55
	3.3	Table Structure	56
	3.4	Object Diagram	64
	3.5	Class Diagram	65
	3.6	Use Case Diagrams	66
	3.6	Sequence Diagram	72
	3.7	Activity Diagram	75
	3.8	State Diagram	82
	3.9	Interface Diagram	83
	3.10	Deployment Diagram	84
	3.11	Collaboration Diagram	85
	3.12	Module Hierarchy Diagram	86
	3.13	Sample Input and Output Screens	90
<b>4</b>		<b>Coding</b>	
	4.1	Code snippets	110

<b>5</b>		<b>Testing</b>	
	5.1	Test Strategy	158
	5.2	Unit Test Plan	160
	5.3	Acceptance Test Plan	160
	5.4	Test Case / Test Script	161
<b>6</b>		<b>Limitations of Proposed System</b>	173
<b>7</b>		<b>Proposed Enhancements</b>	175
<b>8</b>		<b>Conclusion</b>	178
<b>9</b>		<b>Bibliography</b>	179
<b>10</b>		<b>User Manual</b> (All screens with proper description/purpose Details about validations related to data to be entered.)	181

# **1. Introduction**

## **1.1 Company Profile/Institute Profile/Client Profile:**

SGMS INFOTECH LLP works on Web Application Project, Academic Project & Mobile Application Development, also develop the Connection in Hardware and Software Project and Provide the Best Service to client & Student, also Provide Internship to Computer Sciences and Information Technologies Branches Student like B.E, MCA, MCS and BCA.

Rather than business we develop relation. So for any software and web site development contact us and we provide you satisfactory services. We are motivating our distribution channel aggressively by serving your engineering need in sharp time, which protects your time management.

You give us the requirements we cater it promptly & punctually, quality & price is in our veins which is most competitive, you can contented. Just call us we are very near to your voice. Experience our services & then define us, whether we fit to your esteemed organization or not. We have excellent marketing network with technical support. We also deal with application development were we develop different application which help you to do work efficiently and effectively.

### **Services:**

1. Website Design & development
2. Software Development
3. Digital Marketing
4. Search Engine Optimization
5. Logo Design & Animation
6. E-commerce Web Design

7. Android App Development

8. About Internship

## About Internship:

SGMS INFOTECH LLP offers internship projects to MCA, BE, MCS, MBA students to give them an exposure to real time work environment by letting them work on live projects. Objective of SGMS INFOTECH LLP Internship Program is to assist MCA, BE, MCS, MBA graduates in enhancing their overall skills set to improve their employability level to match industry expectations. SGMS INFOTECH LLP is offering internship programs to MCA, MBA and Engineering students, which will be totally a practical exposure to the students. Topics Covered Experience of working on live projects. The internship program is designed to enable interns work on live projects to give them an opportunity to apply the theoretical knowledge they have gained in the classroom. Interns get to see the immediate impact of programming tools and techniques applied by them who should attend.

## Benefits for students

More and more corporations around the world recognize that, in order to gain a competitive advantage, they need to make sure their people know how to handle themselves at work and how to relate with the new technologies. Some of the things, which students will learn during their internship in SGMS INFOTECH LLP are:

## Real time trainees

Always the employer prefers to go for a person who has an experience, because,

by hiring an inexperienced person the company has to incur an additional cost on training of the candidate, so always they try to avoid going for a fresher. To remove this barrier for the fresher's SGM SOURCE INFOTECH is giving them an opportunity to become a real time trainee in the best and top most companies.

## Assured Opportunities in Top Companies

Doing an internship in SGMS INFOTECH LLP can fetch you a better and bright future by giving an opportunity of being placed in the top companies. We have a track record of giving placement assistance to the outperformed students in their internship program to the well-known companies in the IT sector. Come grab opportunities and make your carrier successful.

## Exposure to current technologies:

In today's world, academic knowledge alone is not sufficient to grow and excel in life; you need something more which adds up to your qualification. As we already told, you will be treated as a real time trainee; you will be exposed to the latest technologies in the market, which will give you a better chance of placement.

Corporate culture Along with the knowledge and skills, it is very important to know the corporate culture that you have to follow in the corporate world. Even though you are not an employee of the company, you will be working with the employees of the company through which you can learn about the corporate culture and the organization structure.

## Selection Process:

- Aptitude Test

- Technical Round
- HR Round

## Key Takeaways

At the end of the internship, you will be providing internship certificate. High performers will be considered for job opportunities. Interested candidates please send your applications along with your CV or can contact us on following numbers.

### **Contact Person:**

Mr. Dinesh Surange

SGMS Infotech LLP

**Mob No:** +91 8275329929

**Email ID:** [sgmsinfotech@gmail.com](mailto:sgmsinfotech@gmail.com)

## **1.2 Abstract:**

The Doctor To Door Steps application is an innovative healthcare platform designed to bridge the gap between patients, doctors, and pathology labs by providing a seamless digital experience. This web-based system enables patients to search for doctors based on their specialization and educational background, book online consultations, and access medical prescriptions and pathology reports from the comfort of their homes.

The system consists of three core modules: Patient, Doctor, and Pathology Lab. Patients can register, log in, book appointments, view doctor prescriptions, request pathology tests, and make online payments for reports. Doctors can manage appointments, prescribe medicines, recommend lab tests, and track patient history. Pathology labs can register, offer diagnostic services, upload reports, and process payments securely.

This project leverages modern web technologies to ensure user-friendly navigation, secure data handling, and real-time updates. By integrating medical consultations and diagnostic services into a single platform, Doctor To Door Steps enhances healthcare accessibility, reduces wait times, and improves patient outcomes.

## **1.3 Existing System and Need for System:**

In the traditional healthcare system, patients must visit hospitals or clinics physically to consult doctors, book appointments, and receive medical prescriptions. The process is often time-consuming and inefficient due to long waiting times, limited availability of specialists, and difficulties in managing patient records. Similarly, pathology tests require patients to visit diagnostic centers, wait for tests, and collect reports manually.

Challenges in the existing System Include:

**Manual Appointment Booking:** Patients need to visit hospitals or clinics to schedule an appointment.

**Limited Access to Specialists:** Finding doctors based on specialization and availability is cumbersome.

**Long Waiting Times:** Patients must wait for extended periods for consultations and test reports.

**Manual Record Management:** Medical prescriptions and test reports are often maintained on paper, increasing the risk of data loss.

**Lack of Integrated System:** Patients, doctors, and pathology labs operate independently, leading to miscommunication and delays in treatment.

### Need for the System

The Doctor To Door Steps application aims to digitalize healthcare services, making them more accessible and efficient. By integrating online appointment booking, virtual consultations, pathology test management, and digital report access, this system offers several benefits:

- **Convenience & Accessibility:** Patients can consult doctors and access medical services from anywhere.
- **Efficient Appointment Management:** Eliminates the need for physical visits for scheduling consultations.
- **Faster Diagnosis & Treatment:** Doctors can provide prescriptions online and suggest immediate pathology tests.
- **Secure Digital Records:** Patients and doctors can access past prescriptions and reports anytime.
- **Online Payment Integration:** Enables hassle-free transactions for consultation fees and pathology reports.
- **Enhanced Patient Experience:** Reduces waiting time, improves communication, and ensures timely medical care.

## **1.4 Scope of System:**

The Doctor To Door Steps application is designed to provide a comprehensive and user-friendly platform for patients, doctors, and pathology labs. It aims to improve healthcare accessibility, streamline appointment management, and enhance medical service delivery. The system's scope covers various functional aspects, ensuring seamless interaction among all stakeholders.

### **1. Patient Module**

- User Registration & Authentication: Patients can register and log in securely with personal details and addresses.
- Doctor Search & Appointment Booking: Patients can search for doctors based on specialization (e.g., Dentist, Cardiologist) or education (MBBS, MD) and book appointments online.
- Prescription & Pathology Services: Patients can view prescriptions from doctors, request pathology tests, and download reports after payment.
- Online Payment: Secure transactions for consultations and test reports.
- Feedback & Rating: Patients can provide ratings and feedback on doctors and pathology labs.

### **2. Doctor Module**

- Doctor Registration & Profile Management: Doctors can register by providing personal and professional details.
- Appointment Management: Doctors can view and update appointment status.
- Prescription Management: Doctors can add prescriptions, suggest medicines, and recommend pathology tests.

- Patient History Access: Doctors can access previous prescriptions and treatment history.
- Feedback Review: Doctors can view patient ratings and feedback.

### 3. Pathology Lab Module

- Lab Registration & Login: Pathology labs can register by entering lab details and services provided.
- Report Upload & Payment Integration: Labs can upload test reports, and patients can make payments before downloading them.
- Feedback & Rating System: Labs can view feedback provided by patients.

### 4. System-Wide Features

- User-Friendly Interface: The application will have an intuitive UI for easy navigation.
- Secure Data Management: Encryption techniques will be used to secure patient records and transactions.
- Real-Time Notifications: Patients, doctors, and pathology labs will receive updates on appointments and test reports.
- Scalability & Future Enhancements: The system can be extended to include telemedicine features, emergency medical services, and AI-based health recommendations.

## **1.5 Operating Environment-Hardware and Software:**

Software Requirements:-

- Operating system: Windows XP/07/08/10.
- Coding Language: JAVA/J2EE
- Frontend: HTML,CSS,JSP,Javascript
- IDE : Eclipse Kepler
- Database : MYSQL

Hardware Requirements:-

- System : core i3
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor: 15 VGA Colour.
- Mouse: Logitech.
- Ram: 512 Mb.

## **1.6 Brief Description of Technology Used:**

### **Java Technology:**

Java technology is both a programming language and a platform.

Java is the primary programming language used for back-end development in the project. It is a versatile, object-oriented language known for its platform independence and scalability. Java provides a wide range of libraries and APIs, making it suitable for building complex web applications.

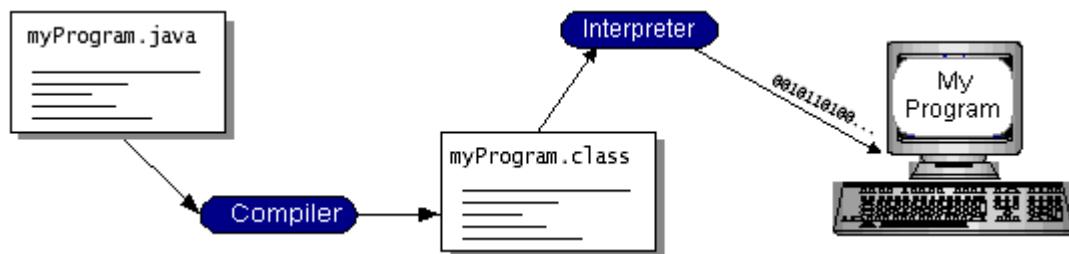
### **The Java Programming Language**

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

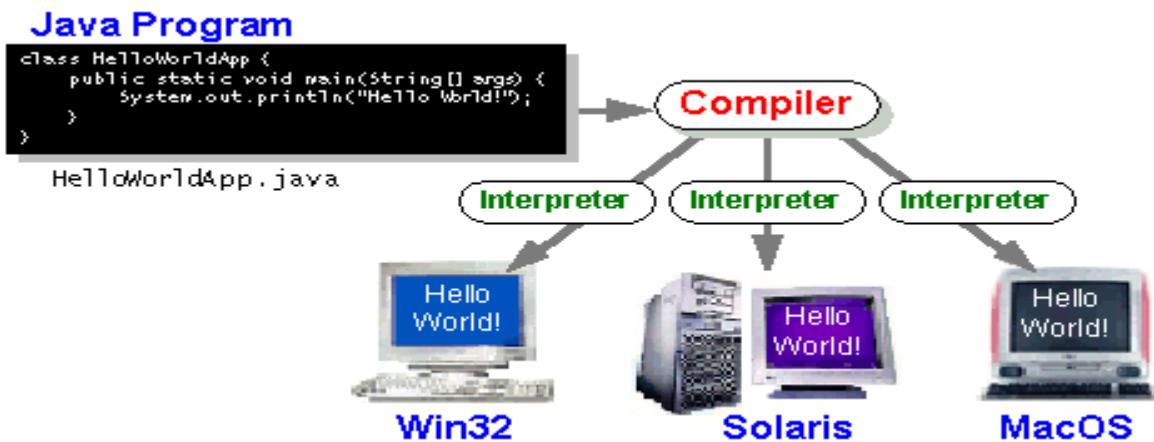
- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the

platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed. The following figure illustrates how this works.



You can think of Java byte codes as the machine code instructions for the *Java Virtual Machine* (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere” possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM.



## The Java Platform

A *platform* is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

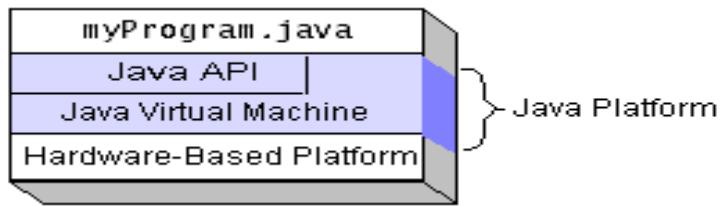
- The *Java Virtual Machine* (Java VM)
- The *Java Application Programming Interface* (Java API)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these

libraries are known as *packages*. The next section, What Can Java Technology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.



Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and just-in-time byte code compilers can bring performance close to that of native code without threatening portability.

## What Can Java Technology Do?

The most common types of programs written in the Java programming language are *applets* and *applications*. If you've surfed the Web, you're probably already familiar with applets. An applet is a program that adheres to certain conventions that allow it to run within a Java-enabled browser.

However, the Java programming language is not just for writing cute, entertaining applets for the Web. The general-purpose, high-level Java programming language

is also a powerful software platform. Using the generous API, you can write many types of programs.

An application is a standalone program that runs directly on the Java platform. A special kind of application known as a *server* serves and supports clients on a network. Examples of servers are Web servers, proxy servers, mail servers, and print servers. Another specialized program is a *servlet*. A servlet can almost be thought of as an applet that runs on the server side. Java Servlets are a popular choice for building interactive web applications, replacing the use of CGI scripts. Servlets are similar to applets in that they are runtime extensions of applications. Instead of working in browsers, though, servlets run within Java Web servers, configuring or tailoring the server.

How does the API support all these kinds of programs? It does so with packages of software components that provides a wide range of functionality. Every full implementation of the Java platform gives you the following features:

**The essentials:** Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.

**Applets:** The set of conventions used by applets.

**Networking:** URLs, TCP (Transmission Control Protocol), UDP (User Data gram Protocol) sockets, and IP (Internet Protocol) addresses.

**Internationalization:** Help for writing programs that can be localized for users worldwide. Programs can automatically adapt to specific locales and be displayed in the appropriate language.

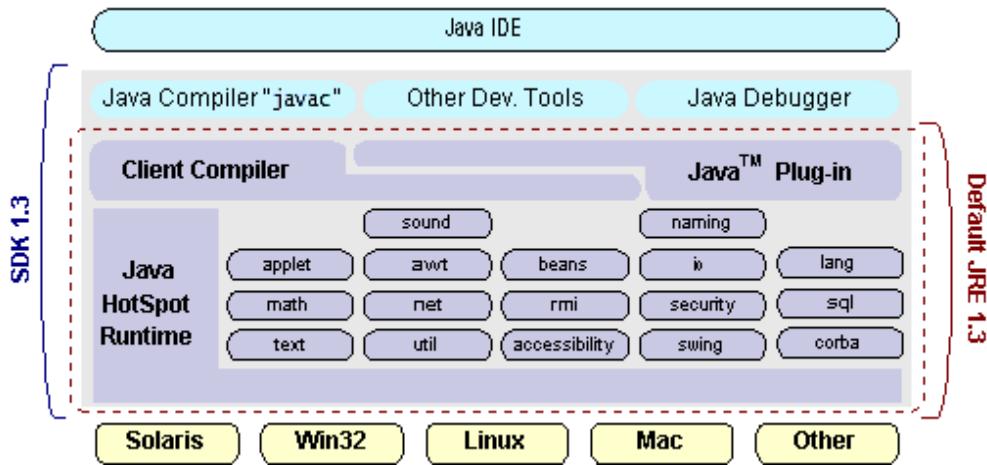
**Security:** Both low level and high level, including electronic signatures, public and private key management, access control, and certificates.

**Software components:** Known as JavaBeans<sup>TM</sup>, can plug into existing component architectures.

**Object serialization:** Allows lightweight persistence and communication via Remote Method Invocation (RMI).

**Java Database Connectivity (JDBC<sup>TM</sup>):** Provides uniform access to a wide range of relational databases.

The Java platform also has APIs for 2D and 3D graphics, accessibility, servers, collaboration, telephony, speech, animation, and more. The following figure depicts what is included in the Java 2 SDK.



## How Will Java Technology Change My Life?

We can't promise you fame, fortune, or even a job if you learn the Java programming language. Still, it is likely to make your programs better and requires less effort than other languages. We believe that Java technology will help you do the following:

**Get started quickly:** Although the Java programming language is a powerful object-oriented language, it's easy to learn, especially for programmers already familiar with C or C++.

**Write less code:** Comparisons of program metrics (class counts, method counts, and so on) suggest that a program written in the Java programming language can be four times smaller than the same program in C++.

**Write better code:** The Java programming language encourages good coding practices, and its garbage collection helps you avoid memory leaks. Its object orientation, its JavaBeans component architecture, and its wide-ranging, easily extendible API let you reuse other people's tested code and introduce fewer bugs.

**Develop programs more quickly:** Your development time may be as much as twice as fast versus writing the same program in C++. Why? You write fewer lines of code and it is a simpler programming language than C++.

**Avoid platform dependencies with 100% Pure Java:** You can keep your program portable by avoiding the use of libraries written in other languages. The 100% Pure Java™ Product Certification Program has a repository of historical process manuals, white papers, brochures, and similar materials online.

**Write once, run anywhere:** Because 100% Pure Java programs are compiled into machine-independent byte codes, they run consistently on any Java platform.

**Distribute software more easily:** You can upgrade applets easily from a central server. Applets take advantage of the feature of allowing new classes to be loaded “on the fly,” without recompiling the entire program.

## ODBC

Microsoft Open Database Connectivity (ODBC) is a standard programming interface for application developers and database systems providers. Before ODBC became a *de facto* standard for Windows programs to interface with database systems, programmers had to use proprietary languages for each database they wanted to connect to. Now, ODBC has made the choice of the database system almost irrelevant from a coding perspective, which is as it should be. Application developers have much more important things to worry about than the syntax that is needed to port their program from one database to another when business needs suddenly change.

Through the ODBC Administrator in Control Panel, you can specify the particular database that is associated with a data source that an ODBC application program is written to use. Think of an ODBC data source as a door with a name on it. Each door will lead you to a particular database. For example, the data source named Sales Figures might be a SQL Server database, whereas the

Accounts Payable data source could refer to an Access database. The physical database referred to by a data source can reside anywhere on the LAN.

The ODBC system files are not installed on your system by Windows 95. Rather, they are installed when you setup a separate database application, such as SQL Server Client or Visual Basic 4.0. When the ODBC icon is installed in Control Panel, it uses a file called ODBCINST.DLL. It is also possible to administer your ODBC data sources through a stand-alone program called ODBCADM.EXE. There is a 16-bit and a 32-bit version of this program and each maintains a separate list of ODBC data sources.

From a programming perspective, the beauty of ODBC is that the application can be written to use the same set of function calls to interface with any data source, regardless of the database vendor. The source code of the application doesn't change whether it talks to Oracle or SQL Server. We only mention these two as an example. There are ODBC drivers available for several dozen popular database systems. Even Excel spreadsheets and plain text files can be turned into data sources. The operating system uses the Registry information written by ODBC Administrator to determine which low-level ODBC drivers are needed to talk to the data source (such as the interface to Oracle or SQL Server). The loading of the ODBC drivers is transparent to the ODBC application program. In a client/server environment, the ODBC API even handles many of the network issues for the application programmer.

The advantages of this scheme are so numerous that you are probably thinking there must be some catch. The only disadvantage of ODBC is that it isn't as efficient as talking directly to the native database interface. ODBC has had many detractors make the charge that it is too slow. Microsoft has always claimed that

the critical factor in performance is the quality of the driver software that is used. In our humble opinion, this is true. The availability of good ODBC drivers has improved a great deal recently. And anyway, the criticism about performance is somewhat analogous to those who said that compilers would never match the speed of pure assembly language. Maybe not, but the compiler (or ODBC) gives you the opportunity to write cleaner programs, which means you finish sooner. Meanwhile, computers get faster every year.

## JDBC

In an effort to set an independent database standard API for Java; Sun Microsystems developed Java Database Connectivity, or JDBC. JDBC offers a generic SQL database access mechanism that provides a consistent interface to a variety of RDBMSs. This consistent interface is achieved through the use of “plug-in” database connectivity modules, or *drivers*. If a database vendor wishes to have JDBC support, he or she must provide the driver for each platform that the database and Java run on.

To gain a wider acceptance of JDBC, Sun based JDBC’s framework on ODBC. As you discovered earlier in this chapter, ODBC has widespread support on a variety of platforms. Basing JDBC on ODBC will allow vendors to bring JDBC drivers to market much faster than developing a completely new connectivity solution.

JDBC was announced in March of 1996. It was released for a 90 day public review that ended June 8, 1996. Because of user input, the final JDBC v1.0 specification was released soon after.

The remainder of this section will cover enough information about JDBC for you to know what it is about and how to use it effectively. This is by no means a complete overview of JDBC. That would fill an entire book.

## JDBC Goals

Few software packages are designed without goals in mind. JDBC is one that, because of its many goals, drove the development of the API. These goals, in conjunction with early reviewer feedback, have finalized the JDBC class library into a solid framework for building database applications in Java.

The goals that were set for JDBC are important. They will give you some insight as to why certain classes and functionalities behave the way they do. The eight design goals for JDBC are as follows:

**1. SQL Level API:** The designers felt that their main goal was to define a SQL interface for Java. Although not the lowest database interface level possible, it is at a low enough level for higher-level tools and APIs to be created. Conversely, it is at a high enough level for application programmers to use it confidently. Attaining this goal allows for future tool vendors to “generate” JDBC code and to hide many of JDBC’s complexities from the end user.

### ***1. SQL Conformance***

SQL syntax varies as you move from database vendor to database vendor. In an effort to support a wide variety of vendors, JDBC will allow any query statement to be passed through it to the underlying database driver. This allows the

connectivity module to handle non-standard functionality in a manner that is suitable for its users.

JDBC must be implemented on top of common database interfaces

The JDBC SQL API must “sit” on top of other common SQL level APIs. This goal allows JDBC to use existing ODBC level drivers by the use of a software interface. This interface would translate JDBC calls to ODBC and vice versa.

Provide a Java interface that is consistent with the rest of the Java system

Because of Java’s acceptance in the user community thus far, the designers feel that they should not stray from the current design of the core Java system.

Keep it simple

This goal probably appears in all software design goal listings. JDBC is no exception. Sun felt that the design of JDBC should be very simple, allowing for only one method of completing a task per mechanism. Allowing duplicate functionality only serves to confuse the users of the API.

Use strong, static typing wherever possible

Strong typing allows for more error checking to be done at compile time; also, less errors appear at runtime.

## Keep the common cases simple

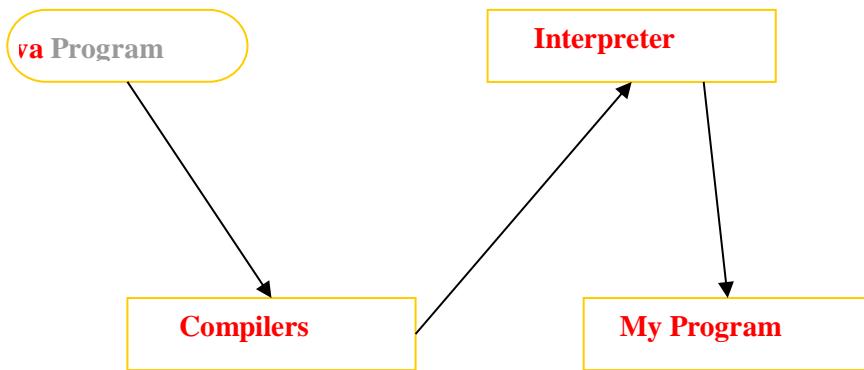
Because more often than not, the usual SQL calls used by the programmer are simple SELECT's, INSERT's, DELETE's and UPDATE's, these queries should be simple to perform with JDBC. However, more complex SQL statements should also be possible.

Java has two things: a programming language and a platform. Java is a high-level programming language that is all of the following

Simple	Architecture-neutral
Object-oriented	Portable
Distributed	High-performance
Interpreted	multithreaded
Robust	Dynamic
Secure	

Java is also unusual in that each Java program is both compiled and interpreted. With a compile you translate a Java program into an intermediate language called Java byte codes the platform-independent code instruction is passed and run on the computer.

Compilation happens just once; interpretation occurs each time the program is executed. The figure illustrates how this works.

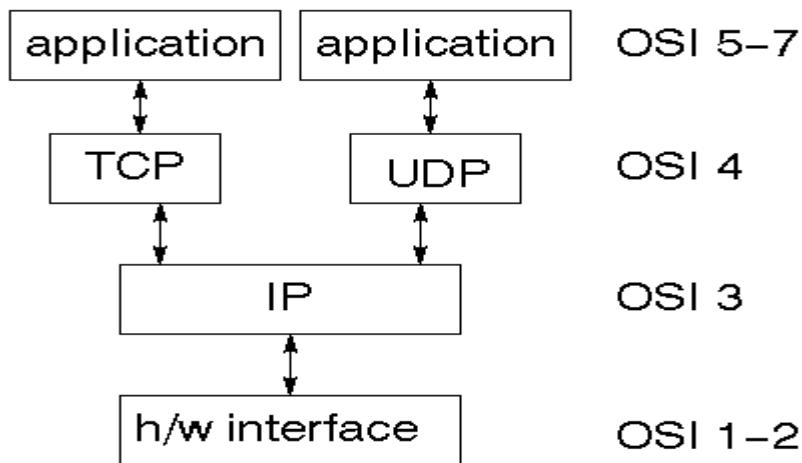


Java byte codes help make “write once, run anywhere” possible. You can compile your Java program into byte codes on my platform that has a Java compiler. The byte codes can then be run any implementation of the Java VM. For example, the same Java program can run Windows NT, Solaris, and Macintosh.

## Networking

### TCP/IP stack

The TCP/IP stack is shorter than the OSI one:



TCP is a connection-oriented protocol; UDP (User Datagram Protocol) is a connectionless protocol.

### IP datagram's

The IP layer provides a connectionless and unreliable delivery system. It considers each datagram independently of the others. Any association between datagram must be supplied by the higher layers. The IP layer supplies a checksum that includes its own header. The header includes the source and destination addresses. The IP layer handles routing through an Internet. It is also responsible for breaking up large datagram into smaller ones for transmission and reassembling them at the other end.

### UDP

UDP is also connectionless and unreliable. What it adds to IP is a checksum for the contents of the datagram and port numbers. These are used to give a client/server model - see later.

### TCP

TCP supplies logic to give a reliable connection-oriented protocol above IP. It provides a virtual circuit that two processes can use to communicate.

### Internet addresses

In order to use a service, you must be able to find it. The Internet uses an address scheme for machines so that they can be located. The address is a 32 bit integer which gives the IP address. This encodes a network ID and more addressing. The network ID falls into various classes according to the size of the network address.

### Network address

Class A uses 8 bits for the network address with 24 bits left over for other addressing. Class B uses 16 bit network addressing. Class C uses 24 bit network addressing and class D uses all 32.

## Subnet address

Internally, the UNIX network is divided into sub networks. Building 11 is currently on one sub network and uses 10-bit addressing, allowing 1024 different hosts.

## Host address

8 bits are finally used for host addresses within our subnet. This places a limit of 256 machines that can be on the subnet.

## Total address:



## Port addresses

A service exists on a host, and is identified by its port. This is a 16 bit number. To send a message to a server, you send it to the port for that service of the host that it is running on. This is not location transparency! Certain of these ports are "well known".

## Sockets

A socket is a data structure maintained by the system to handle network connections. A socket is created using the call socket. It returns an integer that is like a file descriptor. In fact, under Windows, this handle can be used with Read File and Write File functions.

```
#include <sys/types.h>
```

```
#include <sys/socket.h>
```

```
int socket(int family, int type, int protocol);
```

Here "family" will be AF\_INET for IP communications, protocol will be zero, and type will depend on whether TCP or UDP is used. Two processes wishing to communicate over a network create a socket each. These are similar to two ends of a pipe - but the actual pipe does not yet exist.

## J Free Chart

J Free Chart is a free 100% Java chart library that makes it easy for developers to display professional quality charts in their applications. J Free Chart's extensive feature set includes:

A consistent and well-documented API, supporting a wide range of chart types;

A flexible design that is easy to extend, and targets both server-side and client-side applications;

Support for many output types, including Swing components, image files (including PNG and JPEG), and vector graphics file formats (including PDF, EPS and SVG);

J Free Chart is "open source" or, more specifically, free software. It is distributed under the terms of the GNU Lesser General Public Licence (LGPL), which permits use in proprietary applications.

## 1. Map Visualizations

Charts showing values that relate to geographical areas. Some examples include: (a) population density in each state of the United States, (b) income per capita for each country in Europe, (c) life expectancy in each country of the world. The tasks in this project include:

Sourcing freely redistributable vector outlines for the countries of the world, states/provinces in particular countries (USA in particular, but also other areas);

Creating an appropriate dataset interface (plus default implementation), a rendered, and integrating this with the existing XY Plot class in J Free Chart;

Testing, documenting, testing some more, documenting some more.

## 2. Time Series Chart Interactivity

Implement a new (to J Free Chart) feature for interactive time series charts --- to display a separate control that shows a small version of ALL the time series data, with a sliding "view" rectangle that allows you to select the subset of the time series data to display in the main chart.

## 3. Dashboards

There is currently a lot of interest in dashboard displays. Create a flexible dashboard mechanism that supports a subset of J Free Chart chart types (dials, pies, thermometers, bars, and lines/time series) that can be delivered easily via both Java Web Start and an applet.

## 4. Property Editors

The property editor mechanism in JFree Chart only handles a small subset of the properties that can be set for charts. Extend (or re implement) this mechanism to provide greater end-user control over the appearance of the charts.

## What is a Java Web Application?

A Java web application generates interactive web pages containing various types of markup language (HTML, XML, and so on) and dynamic content. It is typically comprised of web components such as Java Server Pages (JSP), servlets and JavaBeans to modify and temporarily store data, interact with databases and web services, and render content in response to client requests.

Because many of the tasks involved in web application development can be repetitive or require a surplus of boilerplate code, web frameworks can be applied to alleviate the overhead associated with common activities. For example, many frameworks, such as Java Server Faces, provide libraries for templating pages and session management, and often promote code reuse.

## What is Java EE?

Java EE (Enterprise Edition) is a widely used platform containing a set of coordinated technologies that significantly reduce the cost and complexity of developing, deploying, and managing multi-tier, server-centric applications. Java EE builds upon the Java SE platform and provides a set of APIs (application programming interfaces) for developing and running portable, robust, scalable, reliable and secure server-side applications.

Some of the fundamental components of Java EE include:

- Enterprise JavaBeans (EJB): a managed, server-side component architecture used to encapsulate the business logic of an application. EJB technology enables rapid and simplified development of distributed, transactional, secure and portable applications based on Java technology.

- Java Persistence API (JPA): a framework that allows developers to manage data using object-relational mapping (ORM) in applications built on the Java Platform.

## JavaScript and Ajax Development

JavaScript is an object-oriented scripting language primarily used in client-side interfaces for web applications. Ajax (Asynchronous JavaScript and XML) is a Web 2.0 technique that allows changes to occur in a web page without the need to perform a page refresh. JavaScript toolkits can be leveraged to implement Ajax-enabled components and functionality in web pages.

## Web Server and Client

Web Server is a software that can process the client request and send the response back to the client. For example, Apache is one of the most widely used web server. Web Server runs on some physical machine and listens to client request on specific port.

A web client is a software that helps in communicating with the server. Some of the most widely used web clients are Firefox, Google Chrome, Safari etc. When we request something from server (through URL), web client takes care of creating a request and sending it to server and then parsing the server response and present it to the user.

## HTML and HTTP

Web Server and Web Client are two separate softwares, so there should be some common language for communication. HTML is the common language between server and client and stands for **Hyper Text Markup Language**.

Web server and client needs a common communication protocol, HTTP (**Hyper Text Transfer Protocol**) is the communication protocol between server and client. HTTP runs on top of TCP/IP communication protocol.

Some of the important parts of HTTP Request are:

- HTTP Method – action to be performed, usually GET, POST, PUT etc.
- URL – Page to access
- Form Parameters – similar to arguments in a java method, for example user, password details from login page.

Sample HTTP Request:

T /First Servlet Project/jsp/hello.jsp HTTP/1.1

st: localhost:8080

he-Control: no-cache

Some of the important parts of HTTP Response are:

- Status Code – an integer to indicate whether the request was success or not. Some of the well known status codes are 200 for success, 404 for Not Found and 403 for Access Forbidden.

- Content Type – text, html, image, pdf etc. Also known as MIME type
- Content – actual data that is rendered by client and shown to user.

**MIME Type or Content Type:** If you see above sample HTTP response header, it contains tag “Content-Type”. It’s also called MIME type and server sends it to client to let them know the kind of data it’s sending. It helps client in rendering the data for user. Some of the mostly used mime types are text/html, text/xml, application/xml etc.

## Understanding URL

URL is acronym of Universal Resource Locator and it’s used to locate the server and resource. Every resource on the web has it’s own unique address. Let’s see parts of URL with an example.

`http://localhost:8080/FirstServletProject/jsp/hello.jsp`

`http://` – This is the first part of URL and provides the communication protocol to be used in server-client communication.

`localhost` – The unique address of the server, most of the times it’s the hostname of the server that maps to unique IP address. Sometimes multiple hostnames point to same IP addresses and web server virtual host takes care of sending request to the particular server instance.

`8080` – This is the port on which server is listening, it’s optional and if we don’t provide it in URL then request goes to the default port of the protocol. Port numbers 0 to 1023 are reserved ports for well known services, for example 80 for HTTP, 443 for HTTPS, 21 for FTP etc.

First Servlet Project/jsp/hello.jsp – Resource requested from server. It can be static html, pdf, JSP, servlets, PHP etc.

## Why we need Servlet and JSPs?

Web servers are good for static contents HTML pages but they don't know how to generate dynamic content or how to save data into databases, so we need another tool that we can use to generate dynamic content. There are several programming languages for dynamic content like PHP, Python, Ruby on Rails, Java Servlets and JSPs.

Java Servlet and JSPs are server side technologies to extend the capability of web servers by providing support for dynamic response and data persistence.

## Web Container

Tomcat is a web container, when a request is made from Client to web server, it passes the request to web container and it's web container job to find the correct resource to handle the request (servlet or JSP) and then use the response from the resource to generate the response and provide it to web server. Then web server sends the response back to the client.

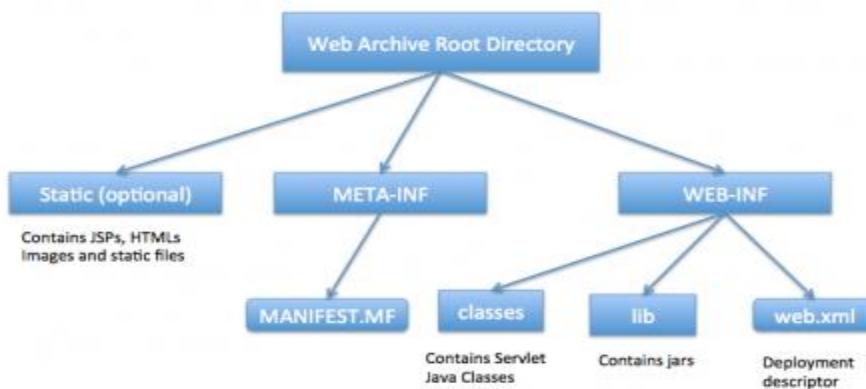
When web container gets the request and if it's for servlet then container creates two Objects HTTP Servlet Request and HTTP Servlet Response. Then it finds the correct servlet based on the URL and creates a thread for the request. Then it invokes the servlet service() method and based on the HTTP method service() method invokes doGet() or doPost() methods. Servlet methods generate the dynamic page and write it to response. Once servlet thread is complete, container converts the response to HTTP response and sends it back to client.

Some of the important works done by web container are:

- Communication Support – Container provides easy way of communication between web server and the servlets and JSPs. Because of container, we don't need to build a server socket to listen for any request from web server, parse the request and generate response. All these important and complex tasks are done by container and all we need to focus is on our business logic for our applications.
- Lifecycle and Resource Management – Container takes care of managing the life cycle of servlet. Container takes care of loading the servlets into memory, initializing servlets, invoking servlet methods and destroying them. Container also provides utility like JNDI for resource pooling and management.
- Multithreading Support – Container creates new thread for every request to the servlet and when it's processed the thread dies. So servlets are not initialized for each request and saves time and memory.
- JSP Support – JSPs doesn't look like normal java classes and web container provides support for JSP. Every JSP in the application is compiled by container and converted to Servlet and then container manages them like other servlets.
- Miscellaneous Task – Web container manages the resource pool, does memory optimizations, run garbage collector, provides security configurations, support for multiple applications, hot deployment and several other tasks behind the scene that makes our life easier.

## Web Application Directory Structure

Java Web Applications are packaged as Web Archive (WAR) and it has a defined structure. You can export above dynamic web project as WAR file and unzip it to check the hierarchy. It will be something like below image.



## Deployment Descriptor

**Web.xml** file is the deployment descriptor of the web application and contains mapping for servlets (prior to 3.0), welcome pages, security configurations, session timeout settings etc.

That's all for the java web application startup tutorial, we will explore Servlets and JSPs more in future posts.

### 1.6.2 Operating systems used (Windows or Unix):

#### Windows:

Windows operating systems, such as Windows Server or Windows 10, are widely used for hosting Java applications. Windows provides a user-friendly environment, and many developers and organizations are familiar with it. Some advantages of using Windows for hosting Java applications include:

- a. Familiarity: Many developers are already accustomed to the Windows environment, which can simplify setup and configuration.
- b. Development Tools: There are various integrated development environments (IDEs) available for Java development on Windows, such as Eclipse, IntelliJ IDEA, and NetBeans.
- c. Compatibility: Java applications developed on Windows are generally compatible with other operating systems, allowing for easy portability.
- d. Easy Deployment: Windows provides straightforward deployment options, making it relatively simple to install and configure Java applications.

## **UNIX:**

Unix-based operating systems, such as Linux and are also popular choices for hosting Java applications. Unix systems are known for their stability, security, and performance. Some advantages of using Unix for hosting Java applications include:

- a. Stability: UNIX systems are renowned for their stability and uptime, making them suitable for hosting critical applications.
- b. Performance: UNIX systems are efficient and often outperform Windows in terms of resource management and overall performance.
- c. Security: UNIX systems have a strong security foundation, which is essential for hosting web applications securely.
- d. Command Line: UNIX systems offer powerful command-line interfaces, providing greater control over the system and facilitating automation.

### **1.6.3 RDBMS/No Sql used to build database (mysql / oracle, Tera data, etc.):**

#### **MySQL:**

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

The MySQL Web site (<http://www.mysql.com/>) provides the latest information about MySQL software.

MySQL is a database management system.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

MySQL databases are relational.

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL-92” refers

to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

MySQL software is Open Source.

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), <http://www.fsf.org/licenses/>, to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information

The MySQL Database Server is very fast, reliable, scalable, and easy to use.

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

You can find a performance comparison of MySQL Server with other database managers on our benchmark page.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

MySQL Server works in client/server or embedded systems.

The MySQL Database Software is a client/server system that consists of a multi-threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

A large amount of contributed MySQL software is available.

MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favorite application or language supports the MySQL Database Server.

The official way to pronounce “MySQL” is “My Ess Que Ell” (not “my sequel”), but we do not mind if you pronounce it as “my sequel” or in some other localized way.

## **2. Proposed System**

## **2.1 Feasibility Study:**

A feasibility study is conducted to evaluate the practicality and effectiveness of the Doctor To Door Steps system before its implementation. The study analyzes the project's viability from different perspectives, including technical, economic, operational, legal, and scheduling feasibility.

### **1. Technical Feasibility**

The system requires a robust and scalable technology stack to ensure smooth operation. The feasibility of the technology is assessed based on:

- **Programming Languages & Frameworks:** The system can be developed using Java Spring Boot (backend), JSP/Servlets (frontend), and MySQL (database) for efficient data management.
- **Platform Compatibility:** The application will be accessible via web browsers and can be extended for mobile support.
- **Security & Data Protection:** Secure authentication, data encryption, and access control mechanisms will be implemented to protect patient records.
- **Hosting & Scalability:** Cloud-based hosting (AWS, Azure, or DigitalOcean) will ensure the system can handle a large number of users.

### **2. Economic Feasibility**

The financial viability of the project is analyzed based on development costs, operational costs, and return on investment.

- **Development Costs:** The system will require investment in software development, database setup, and cloud hosting.

- **Operational Costs:** Maintenance, server costs, and support services will incur ongoing expenses.
- **Revenue Model:** The system can generate revenue through **subscription** plans for doctors and pathology labs, appointment booking charges, **and** commission on pathology services.
- **Cost-Benefit Analysis:** The system reduces healthcare costs for patients by minimizing travel and consultation expenses while providing a scalable business model.

### 3. Operational Feasibility

The system's operational feasibility is assessed based on its ability to meet user requirements and integrate into existing workflows.

- **Ease of Use:** The application will have a **user-friendly interface** to ensure smooth navigation for patients, doctors, and pathology labs.
- **Efficiency Improvement:** The system reduces manual appointment booking, paperwork, and waiting times, improving overall efficiency.
- **User Acceptance:** Since digital healthcare solutions are increasingly adopted, users are likely to accept and use the platform.

### 4. Legal Feasibility

Legal considerations are crucial to ensure compliance with medical and data protection laws.

- **Data Privacy & Security:** The system must comply with HIPAA (Health Insurance Portability and Accountability Act) or similar regulations to protect patient data.
- **Medical Compliance:** Doctors and pathology labs must be verified before registration to ensure authentic services.
- **Terms of Service & Liability:** Legal agreements will define the responsibilities of patients, doctors, and pathology labs to prevent disputes.

## 5. Scheduling Feasibility

The project timeline must be reasonable to ensure successful implementation.

- **Project Phases:** The development lifecycle includes requirement gathering, design, development, testing, deployment, and maintenance.
- **Estimated Timeline:** The system can be developed within **4-6 months**, depending on complexity and resources.
- **Risk Management:** Any delays due to technical challenges or requirement changes can be managed by proper planning and agile development.

## **2.2 Objectives of Proposed System:**

The Doctor To Door Steps application aims to enhance healthcare accessibility by providing an online platform for doctor consultations, appointment management, and pathology lab services. The key objectives of the proposed system are as follows:

### **1. Improve Healthcare Accessibility**

- Enable patients to find and consult doctors online based on specialization and availability.
- Provide access to medical services without the need for physical visits.

### **2. Efficient Appointment Management**

- Allow patients to book, update, or cancel appointments conveniently.
- Help doctors manage their schedules and keep track of patient bookings.

### **3. Digital Prescription & Pathology Integration**

- Allow doctors to prescribe medicines and suggest pathology tests digitally.
- Enable patients to request pathology services and download reports online.

### **4. Secure Online Payment System**

- Facilitate online payments for consultations and pathology reports.
- Ensure secure transactions through payment gateways.

### **5. Seamless Patient-Doctor Communication**

- Provide a platform for doctors to share prescriptions and medical advice with patients.
- Allow patients to review past prescriptions and medical history.

## **6. Improve Pathology Lab Services**

- Enable pathology labs to register, manage services, and upload reports online.
- Reduce the need for patients to visit labs for report collection.

## **7. User-Friendly & Secure System**

- Ensure an intuitive interface for patients, doctors, and pathology labs.
- Implement data security measures to protect patient records and transactions.

## **8. Feedback & Rating System**

- Allow patients to rate doctors and pathology labs based on service quality.
- Help improve healthcare services through user feedback.

## **9. Scalability & Future Enhancements**

- Design a scalable system that can integrate telemedicine, emergency consultations, and AI-driven health recommendations in the future.

## **2.3 Users of System:**

### **1. Patients (End Users)**

Patients are the main beneficiaries of the system, utilizing it for booking doctor consultations and pathology tests. Their key functionalities include: Register & Login – Create an account and securely log in. Search & Book Doctors – Find doctors based on specialization and schedule appointments.

View & Manage Appointments – Check, update, or cancel bookings.

Access Prescriptions & Pathology Reports – View medical prescriptions and request lab tests.

Make Dummy Online Payments – Pay for pathology reports or consultations.

Provide Feedback & Ratings – Share service reviews for doctors and labs.

### **2. Doctors (Healthcare Providers)**

Doctors use the system to manage patient appointments, provide prescriptions, and track patient history. Their main functionalities include:

Register & Login – Sign up by providing personal and professional details.

Manage Appointments – View, confirm, or update appointments.

Provide Prescriptions & Lab Recommendations – Suggest medicines and refer patients for tests.

Access Patient History – View past prescriptions and medical records.

Review Feedback & Ratings – Check patient reviews to improve service quality.

### **3. Pathology Labs (Diagnostic Service Providers)**

Pathology labs use the system to manage lab tests, upload reports, and handle payments. Their key functionalities include:

**Register & Login** – Register as a pathology lab and manage profile details.

**Manage Lab Services** – List available tests like X-ray, MRI, Blood Tests, etc.

**Handle Appointments & Payments** – View patient test requests and process payments.

**Upload Test Reports** – Provide digital reports that patients can download.

**View Feedback & Ratings** – Monitor patient reviews to improve lab services.

### **4. System Administrators (Platform Managers)**

System admins oversee platform operations, ensuring smooth functioning and security. Their key responsibilities include:

**User Management** – Approve or verify doctors and pathology labs.

**Monitor Transactions & Security** – Ensure secure payments and prevent fraudulent activities.

**Manage System Data** – Handle backups, data security, and platform updates.

# **3. Analysis and Design**

### **3.1 System Requirements (Functional and Non-Functional requirements)**

The Doctor To Door Steps application requires a well-defined set of functional and non-functional requirements to ensure its proper operation, security, and user experience.

#### **Function Requirement:**

Functional requirements define the core features and operations of the system.

##### **Patient Module**

User Registration & Authentication – Patients must register with personal details and securely log in.

Doctor Search & Appointment Booking – Patients can search for doctors based on specialization and book appointments.

Prescription & Pathology Requests – Patients can view doctor prescriptions and request lab tests.

Online Payment Processing – Patients must be able to pay for consultations and reports.

View & Manage Appointments – Patients can modify or cancel appointments.

Feedback & Ratings – Patients can provide reviews for doctors and pathology labs.

## Doctor Module

Doctor Registration & Login – Doctors register with credentials and log in securely.

Appointment Management – Doctors can view, confirm, or update appointments.

Prescription Management – Doctors can add prescriptions and recommend lab tests.

Patient History Access – Doctors can check previous prescriptions and reports.

View Feedback – Doctors can monitor patient ratings and reviews.

## Pathology Lab Module

Lab Registration & Login – Labs must register with details and securely log in.

Appointment & Payment Handling – Labs can view test requests and process payments.

Report Upload System – Labs must be able to upload test reports for patients to download.

Feedback & Rating System – Labs can view patient feedback and improve services.

## Administrator Module

User Management – Admins approve and manage doctor and pathology lab registrations.

System Monitoring – Admins oversee user activities, security, and transactions.

Data Backup & Security Management – Admins handle system backups and prevent unauthorized access.

## Non-Functional Requirement:

Non-functional requirements define the system's performance, security, and usability.

### Performance Requirements

Scalability – The system should support multiple users simultaneously without performance issues.

Response Time – The system should process user actions (e.g., booking appointments) within **3 seconds**.

Data Processing – Patient records, prescriptions, and reports should be stored and retrieved efficiently.

### Security Requirements

User Authentication & Authorization – Secure login mechanisms should be implemented for patients, doctors, and labs.

Data Encryption – Patient records, prescriptions, and payment details should be encrypted for security.

Role-Based Access Control – Patients, doctors, and labs should have different permissions to access system features.

### Usability Requirements

User-Friendly Interface – The platform should be intuitive and easy to navigate for all users.

Mobile & Web Compatibility – The system should be accessible via desktop and mobile browsers.

Multi-Language Support (Optional) – The system should support different languages for better accessibility.

### Reliability & Maintenance Requirements

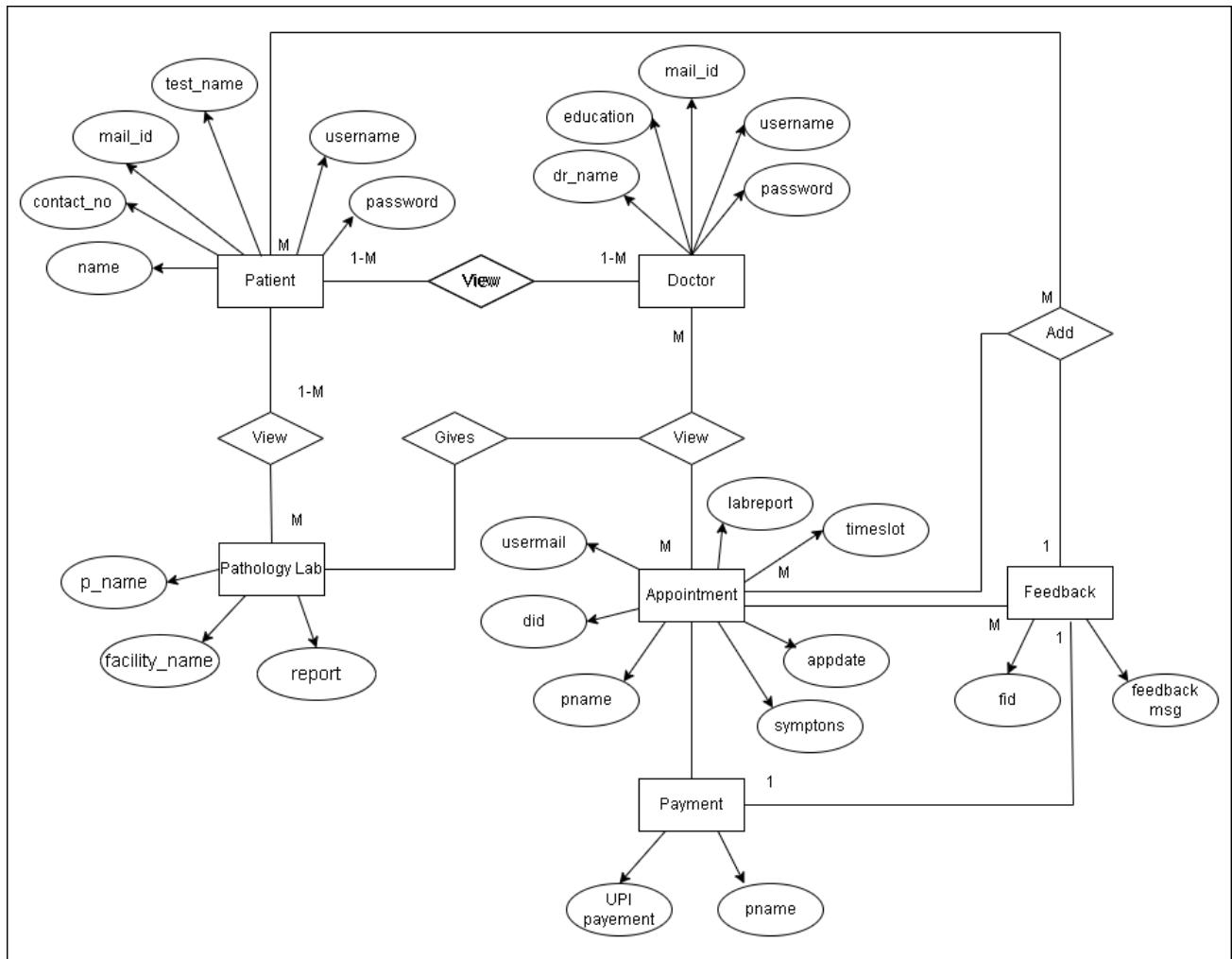
System Uptime – The system should be available **24/7 with at least 99% uptime.**

Data Backup – Regular backups should be maintained to prevent data loss.

Bug Fixing & Updates – The system should be updated regularly to fix bugs and enhance performance.

### 3.2 Entity Relationship Diagram (ERD):

The ER model defines the conceptual view of a database. It works around real-world entities and the associations among them. At view level, the ER model is considered a good option for designing databases.



### **3.13 Table Structure:**

Table Name: Patient Details:

Sr. No	Field	Size	Type	Constraints
1	Id	11	Number	Primary Key
2	Fname	30	String	Not Null
3	Mname	30	String	Not Null
4	Lname	30	String	Not Null
5	Dob	30	String	Not Null
6	Gender	30	String	Not Null
7	Mobile	30	String	Not Null
8	adhar_id	30	String	Not Null
9	Address	1000	String	Not Null
10	Email	30	Email	Not Null
11	Password	30	Password	Not Null
12	reg_date	30	Date	Not Null

## Table Name: Doctor Details

Sr. No	Field	Size	Type	Constraints
1	Id	11	Number	Primary Key
2	Fname	30	String	Not Null
3	Mname	30	String	Not Null
4	Lname	30	String	Not Null
5	Dob	30	String	Not Null
6	Gender	30	String	Not Null
7	Mobile	30	Number	Not Null
8	adhar_id	30	Number	Not Null
9	acc_type	30	String	Not Null
10	Qualification	100	String	Not Null
11	Address	1000	String	Not Null
12	Email	30	Email	Not Null
13	Password	30	Password	Not Null
14	reg_date	30	Date	Not Null
15	Status	30	String	Not Null
16	area_name	30	String	Not Null
17	Speciality	100	String	Not Null

Table Name: doctor specialties

Sr. No	Field	Size	Type	Constraints
1	Id	11	String	Primary Key
2	specialty_name	225	String	Not Null

Table Name: Feedback Details

Sr. No	Field	Size	Type	Constraints
1	Id	11	Number	Primary Key
2	app_id	30	Number	Not Null
3	p_id	30	Number	Not Null
4	dr_id	30	Number	Not Null
5	Feedback	30	String	Not Null
6	Rating	30	Number	Not Null
7	Cdate	30	Date	Not Null
8	fd_for	30	String	Not Null

## Table Name: Medical Details

Sr. No	Field	Size	Type	Constraints
1	Id	11	Number	Primary Key
2	medical_name	225	String	Not Null
3	Address	225	String	Not Null
4	pin_code	10	Number	Not Null
5	contact_person	225	String	Not Null
6	contact_number	20	Number	Not Null

## Table Name: Pathology Report

Sr. No	Field	Size	Type	Constraints
1	Id	11	Number	Primary Key
2	patient_id	30	Number	Not Null
3	Pname	100	String	Not Null
4	dr_id	30	Number	Not Null
5	app_id	30	Number	Not Null
6	time_slot	30	Time	Not Null
7	app_date	30	Date	Not Null
8	Status	30	String	Not Null
9	c_date	30	Date	Not Null
10	refer_by_dr_id	30	Number	Not Null
11	feedback_status	30	String	Not Null

## Table Name: Doctor Details

Sr. No	Field	Size	Type	Constraints
1	Id	11	Number	Primary Key
2	User_email	30	Email	Not Null
3	D_id	30	String	Not Null
4	pname	100	String	Not Null
5	Symptons	300	String	Not Null
6	app_date	30	Date	Not Null
7	Time_slot	30	String	Not Null
8	Status	30	Number	Not Null
9	Medicine	30	String	Not Null
10	Feedback	300	String	Not Null
11	Lab_report	3000	String	Not Null
12	Lab_report_sts	300	String	Not Null
13	fees	30	String	Not Null
14	Fees_status	30	String	Not Null

Table Name: Payment Details

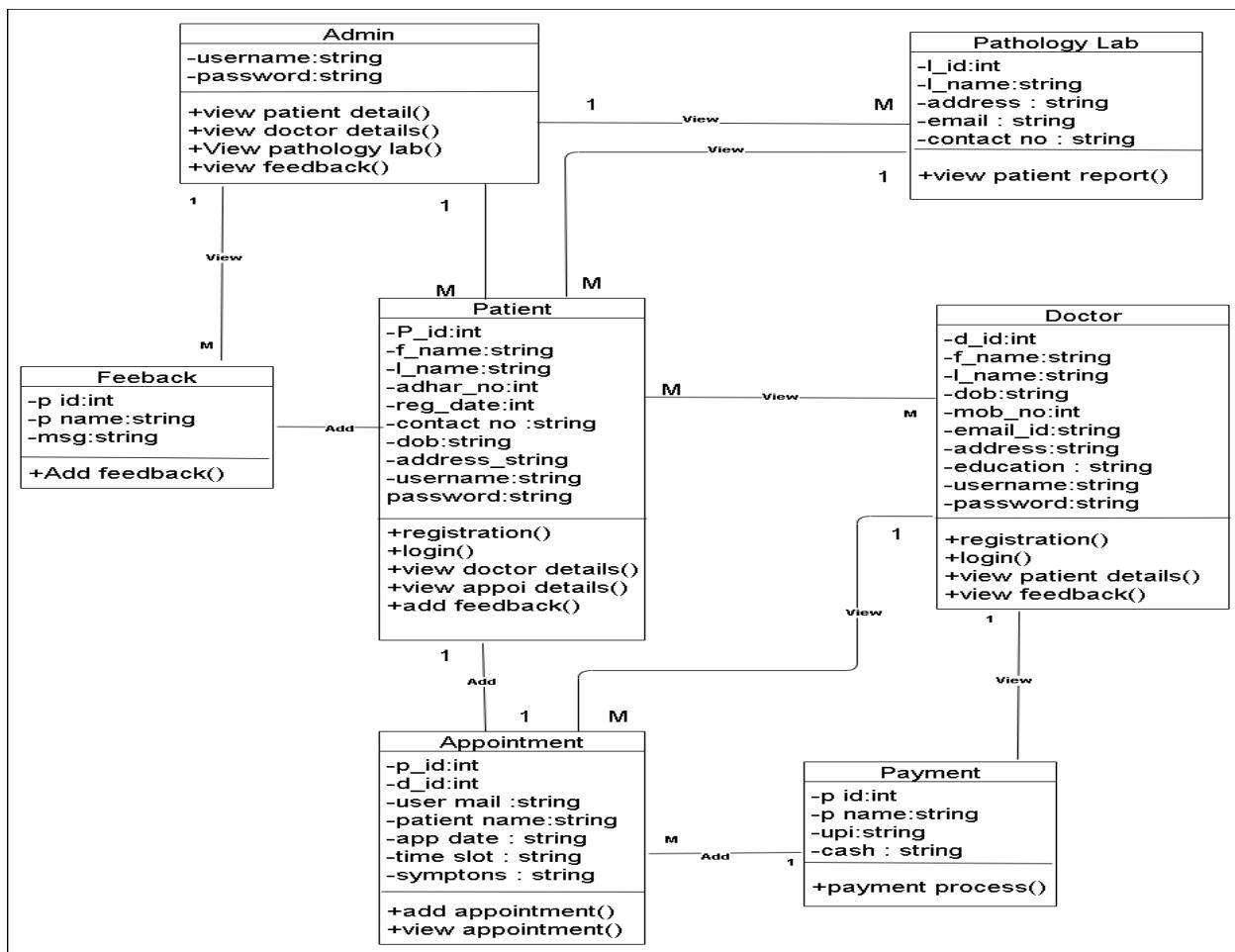
Sr. No	Field	Size	Type	Constraints
1	Id	30	Number	Primary Key
2	app_id	30	Number	Not Null
3	d_id	30	Number	Not Null
4	p_id	30	Number	Not Null
5	Fees	30	Number	Not Null
6	Status	30	String	Not Null
7	c_date	30	Date	Not Null

Table Name: Admin Details

Sr. No	Field	Size	Type	Constraints
1	Id	11	Number	Primary Key
2	Username	30	String	Not Null
3	password	30	Password	Not Null

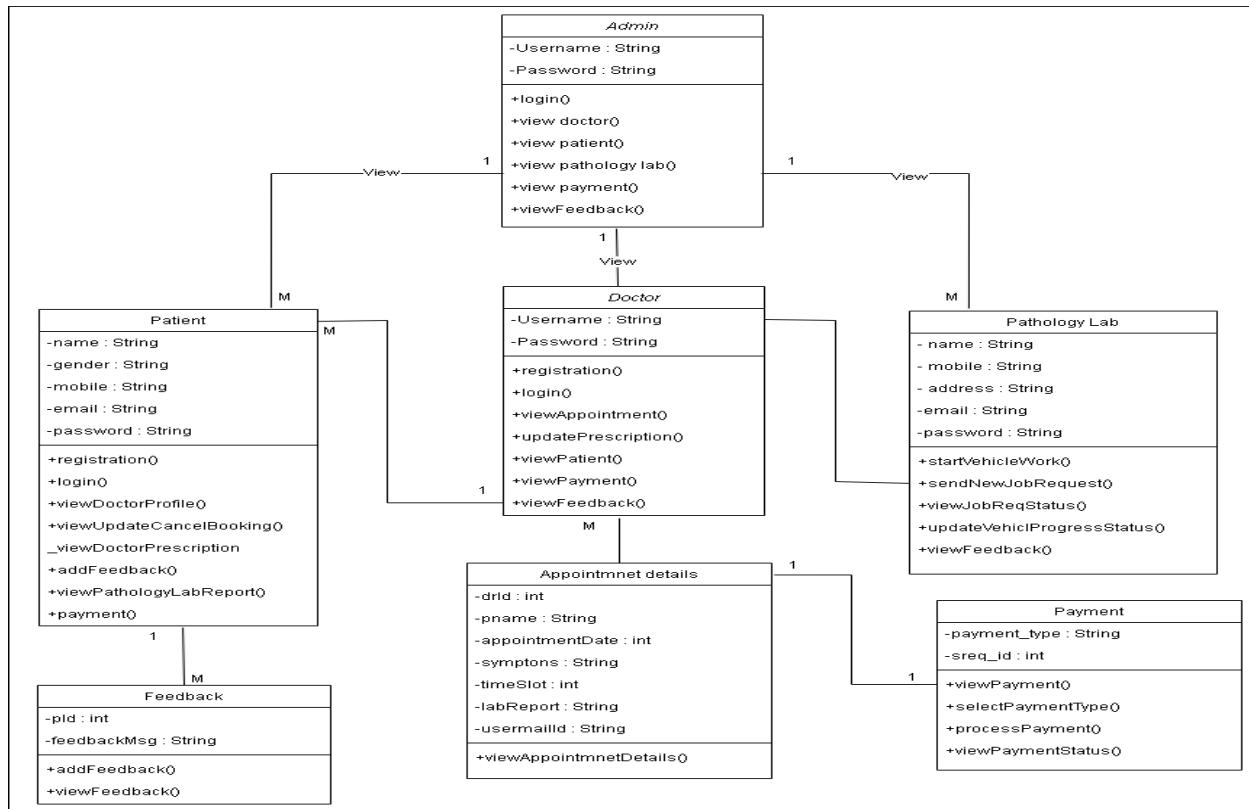
### 3.3 Object Diagram:

An object diagram is a graph of instances, including objects and data values. A static object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time. The use of object diagrams is fairly limited, namely to show examples of data structure.



### 3.4 Class Diagram:

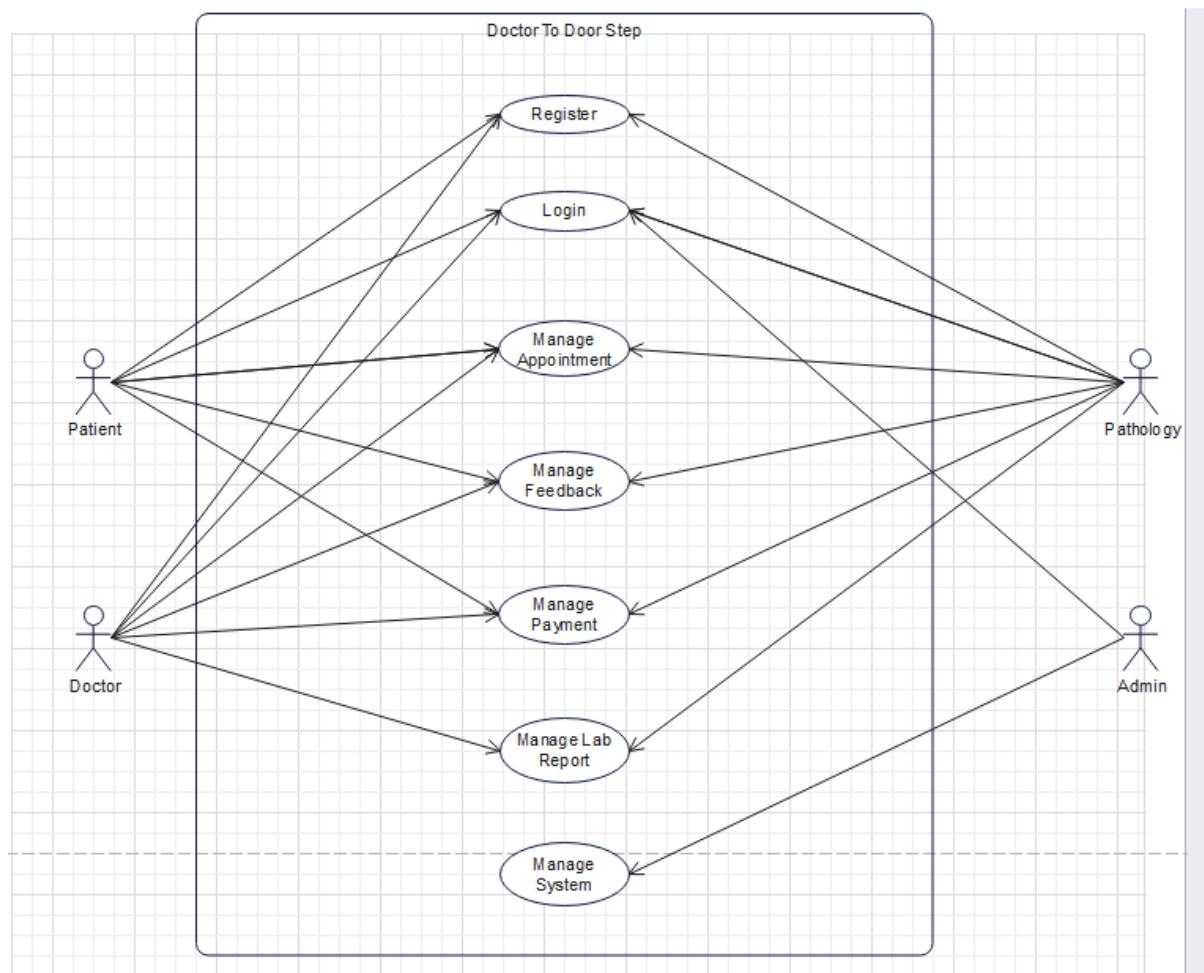
In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system, classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



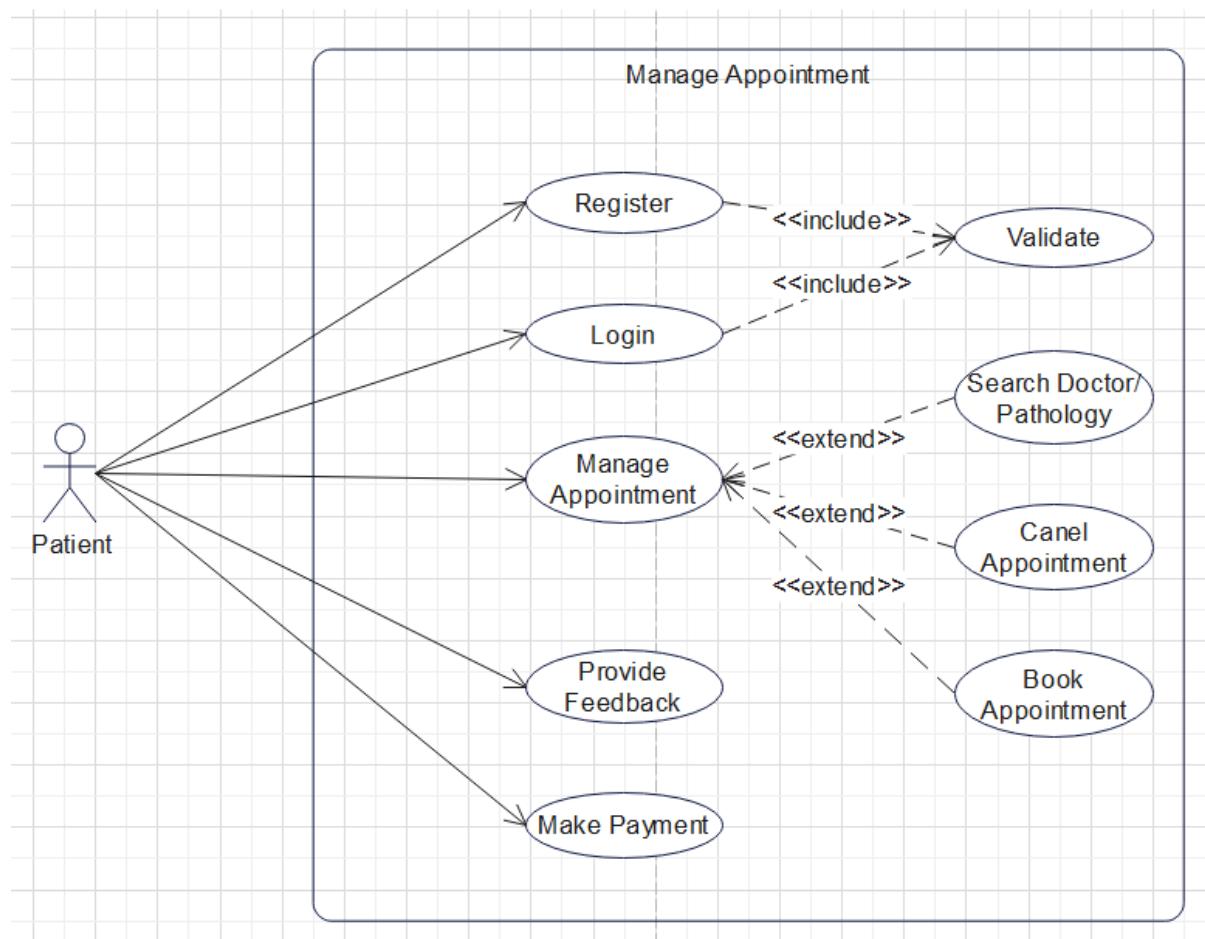
### 3.5 Use Case Diagrams:

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system.

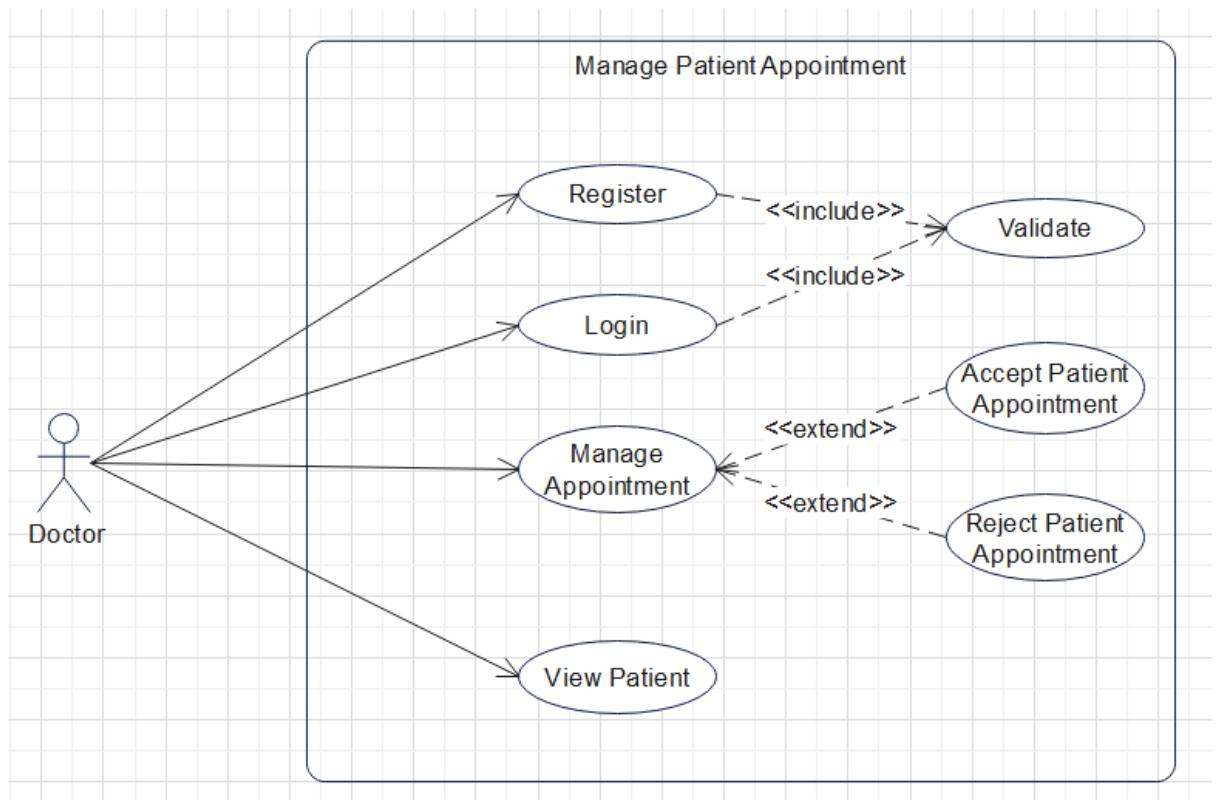
#### Global Use Case Diagram:



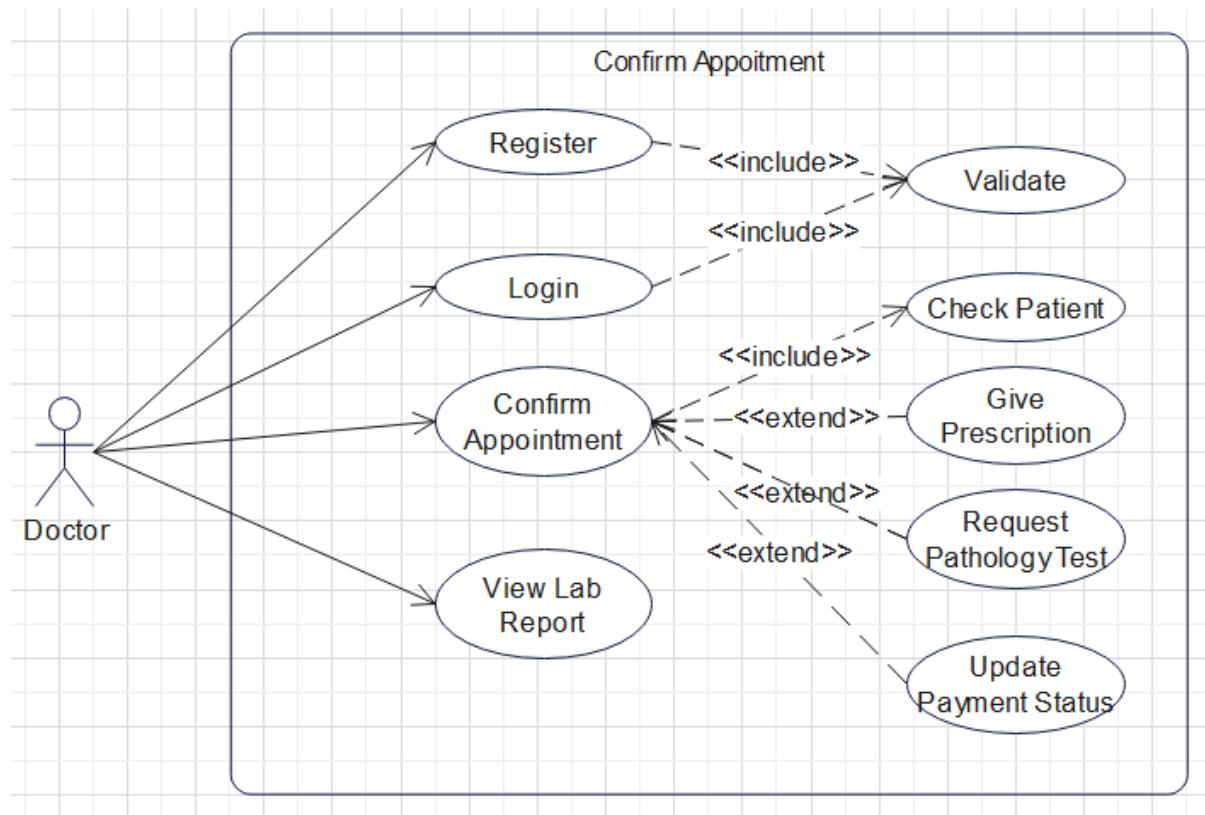
#### Manage Appointment Use Case:



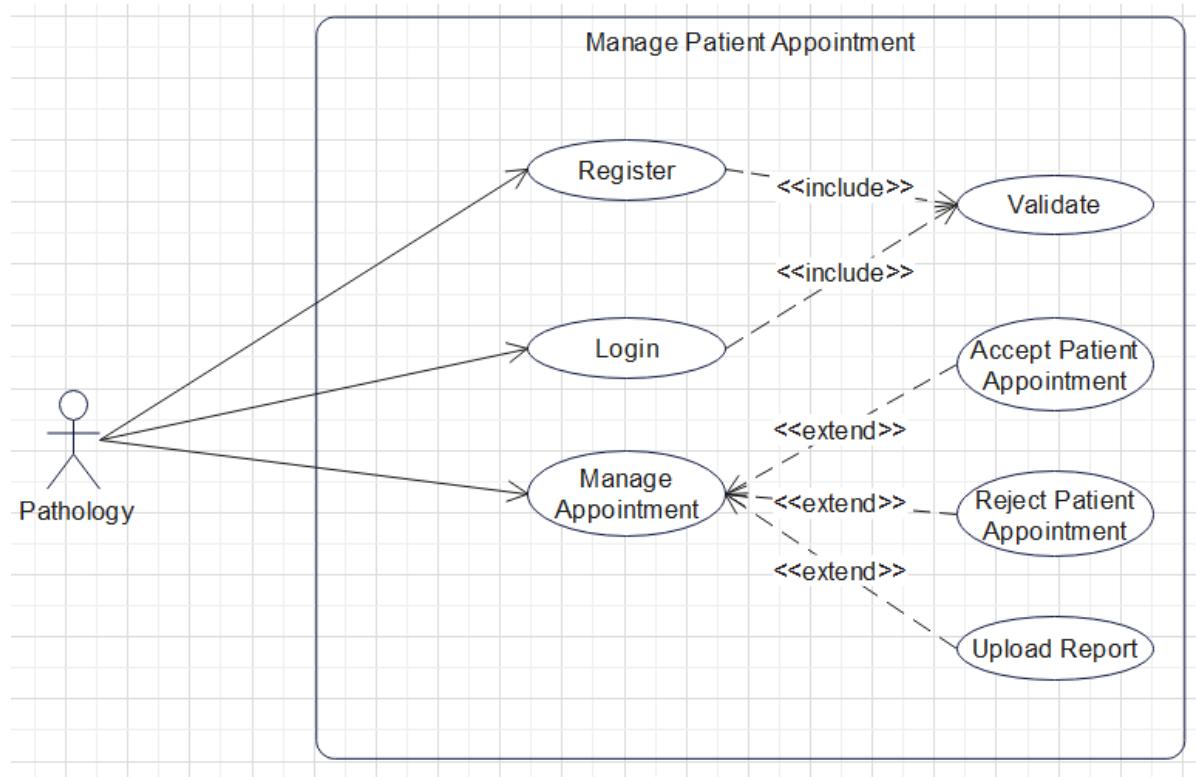
## Manage Patient Appointment Use Case:



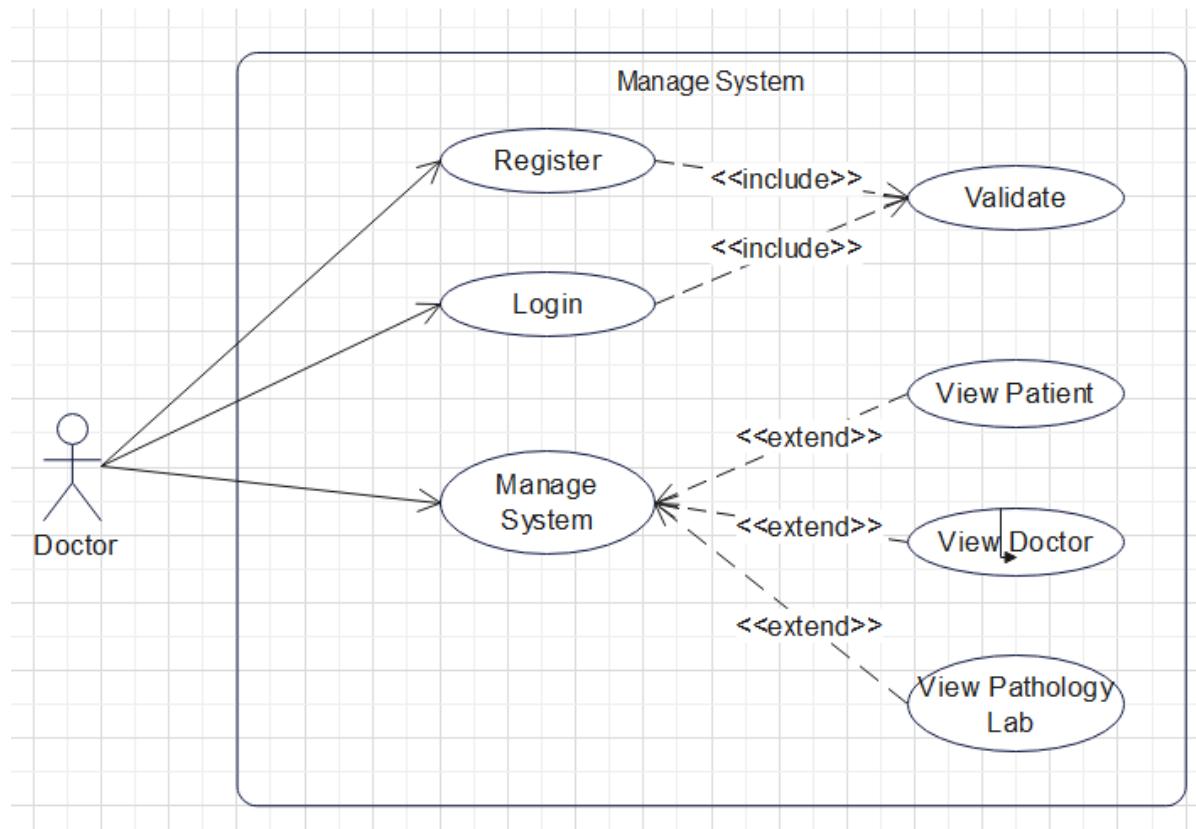
## Confirm Appointment Use Case:



## Manage Pathology Appointment Use Case:



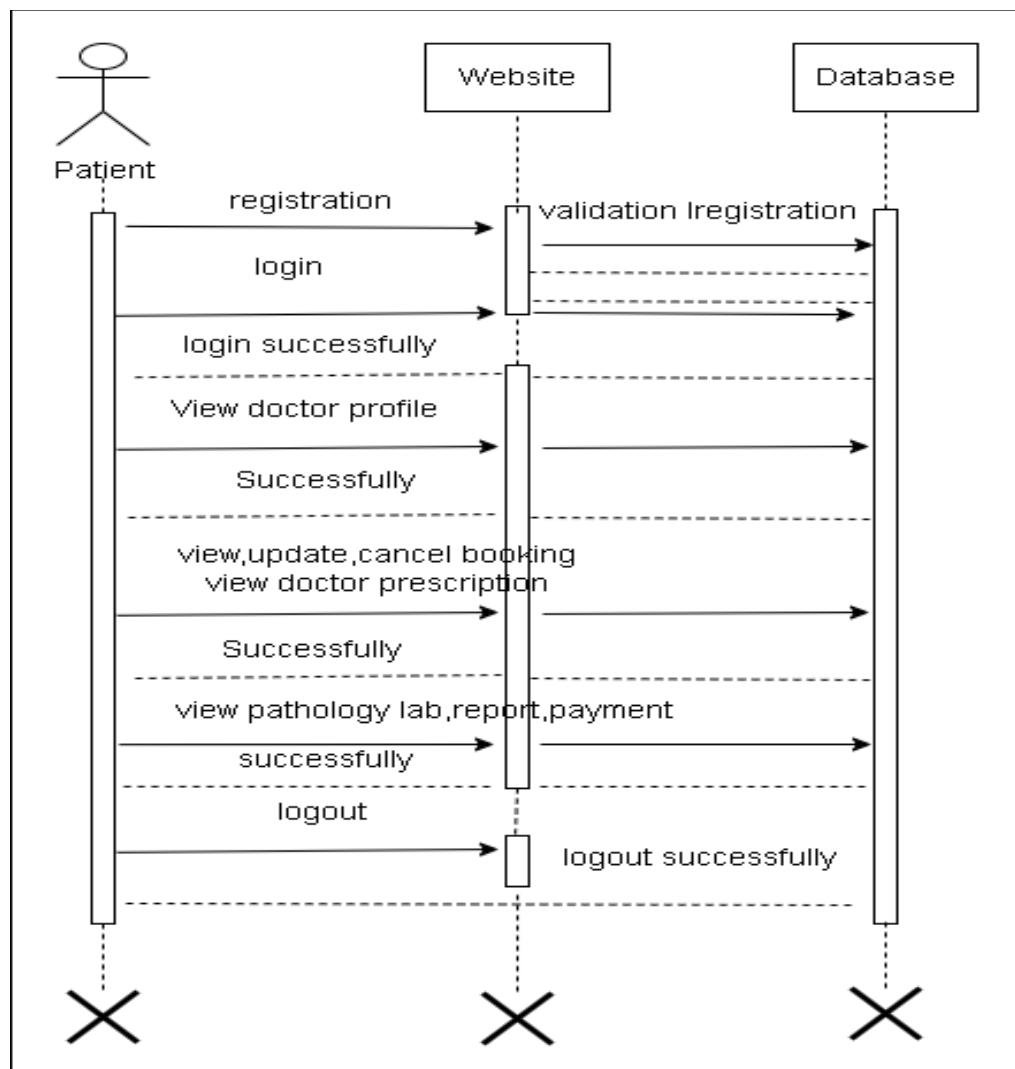
## Manage System Use Case:



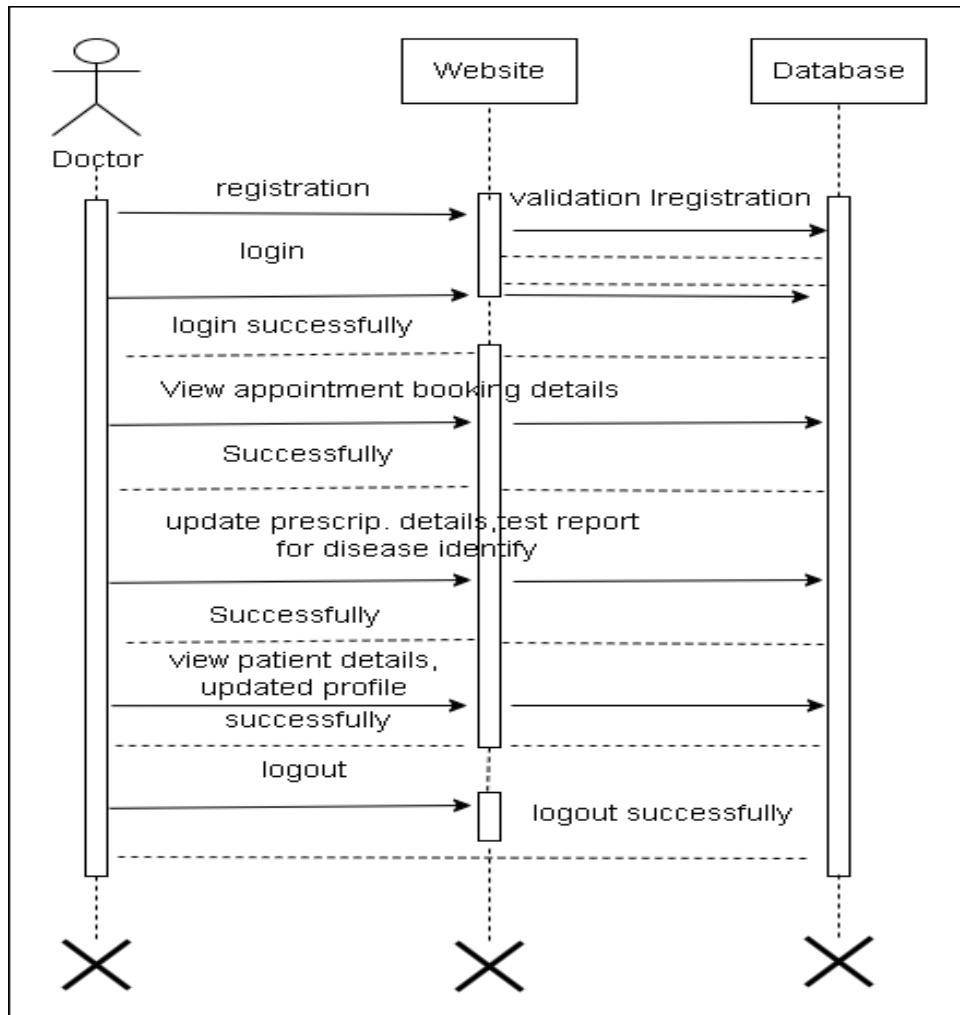
### 3.6 Sequence Diagram:

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence.

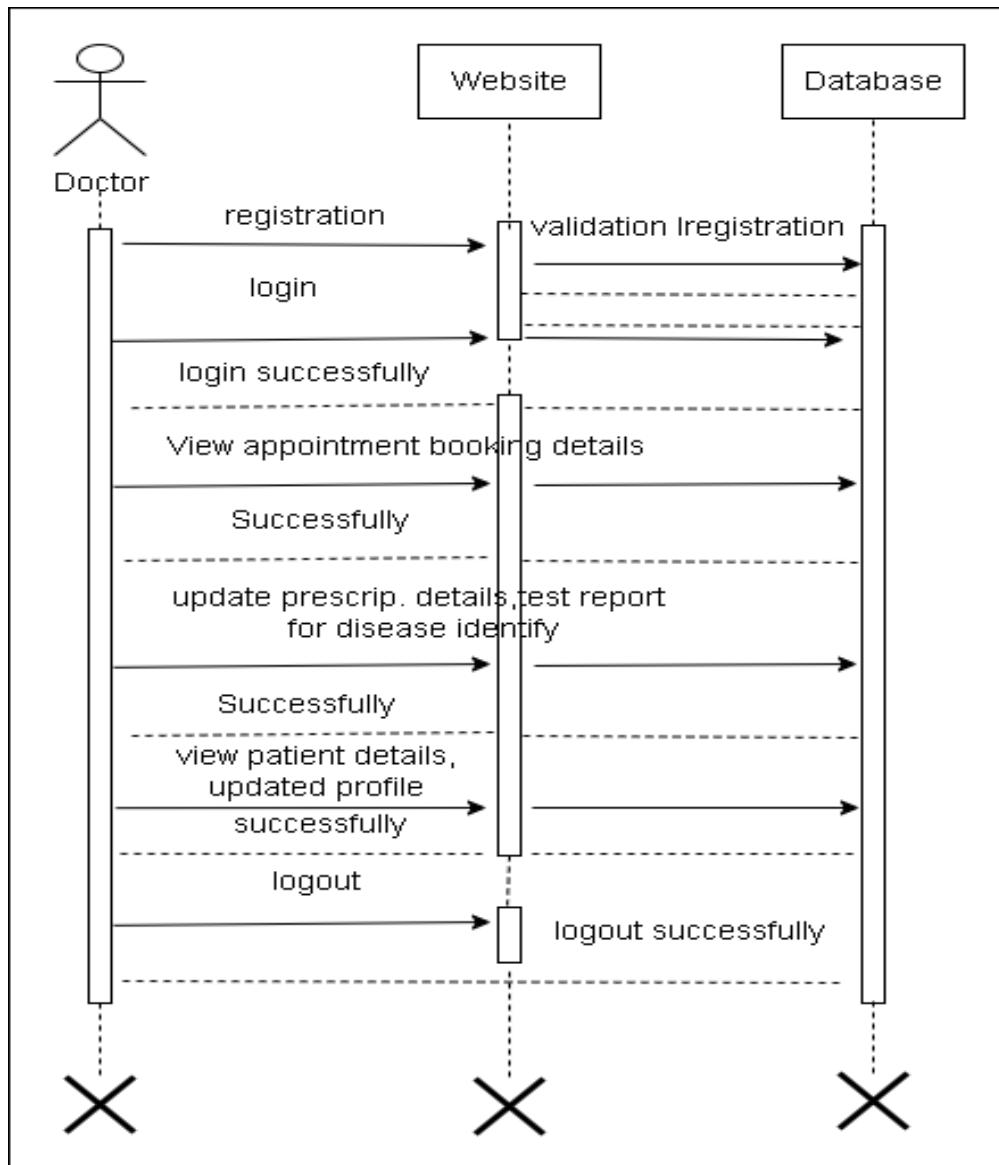
Patient Sequence Diagram:



## Doctor Sequence Diagram:



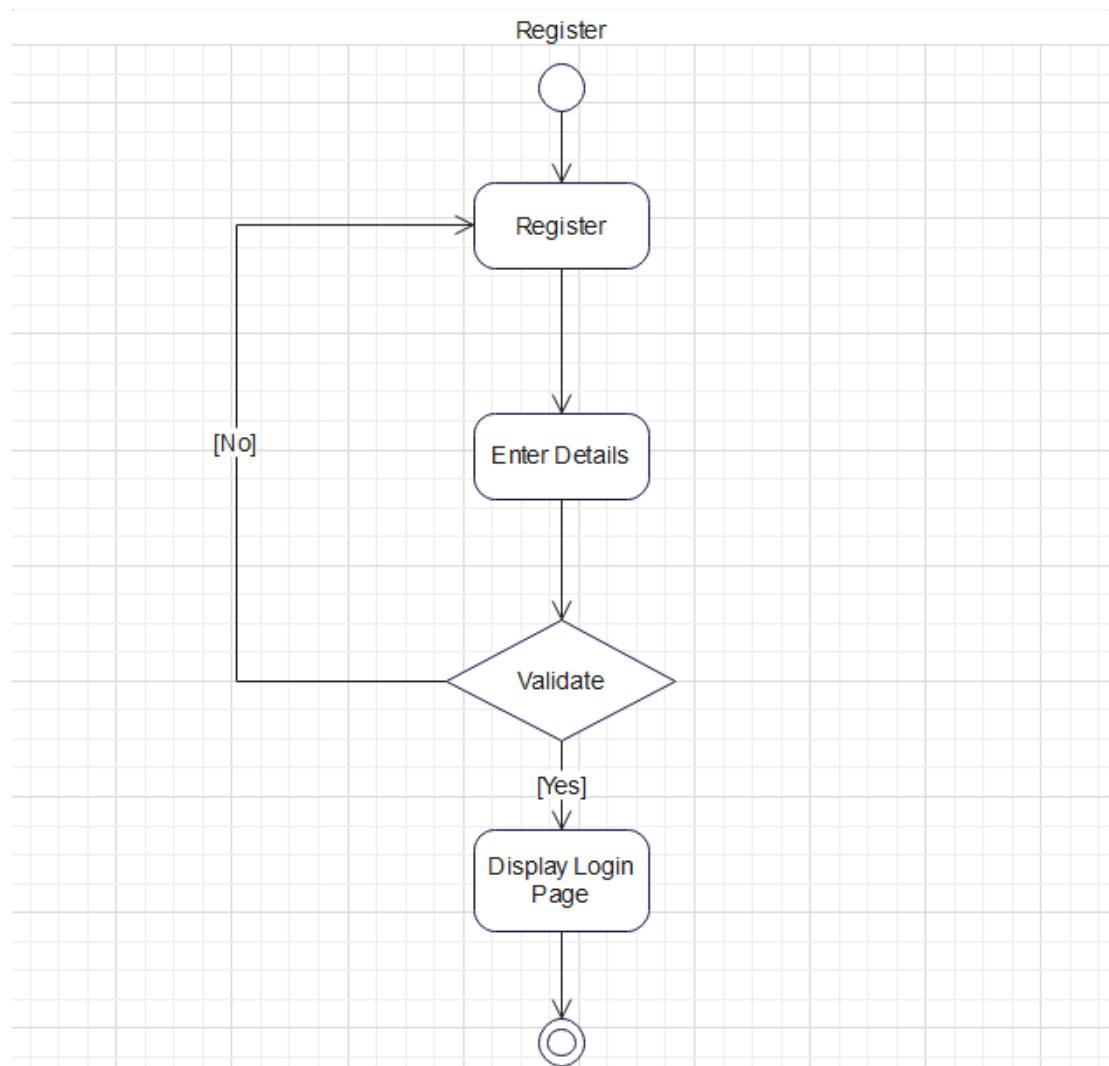
## Pathology Lab Sequence Diagram:



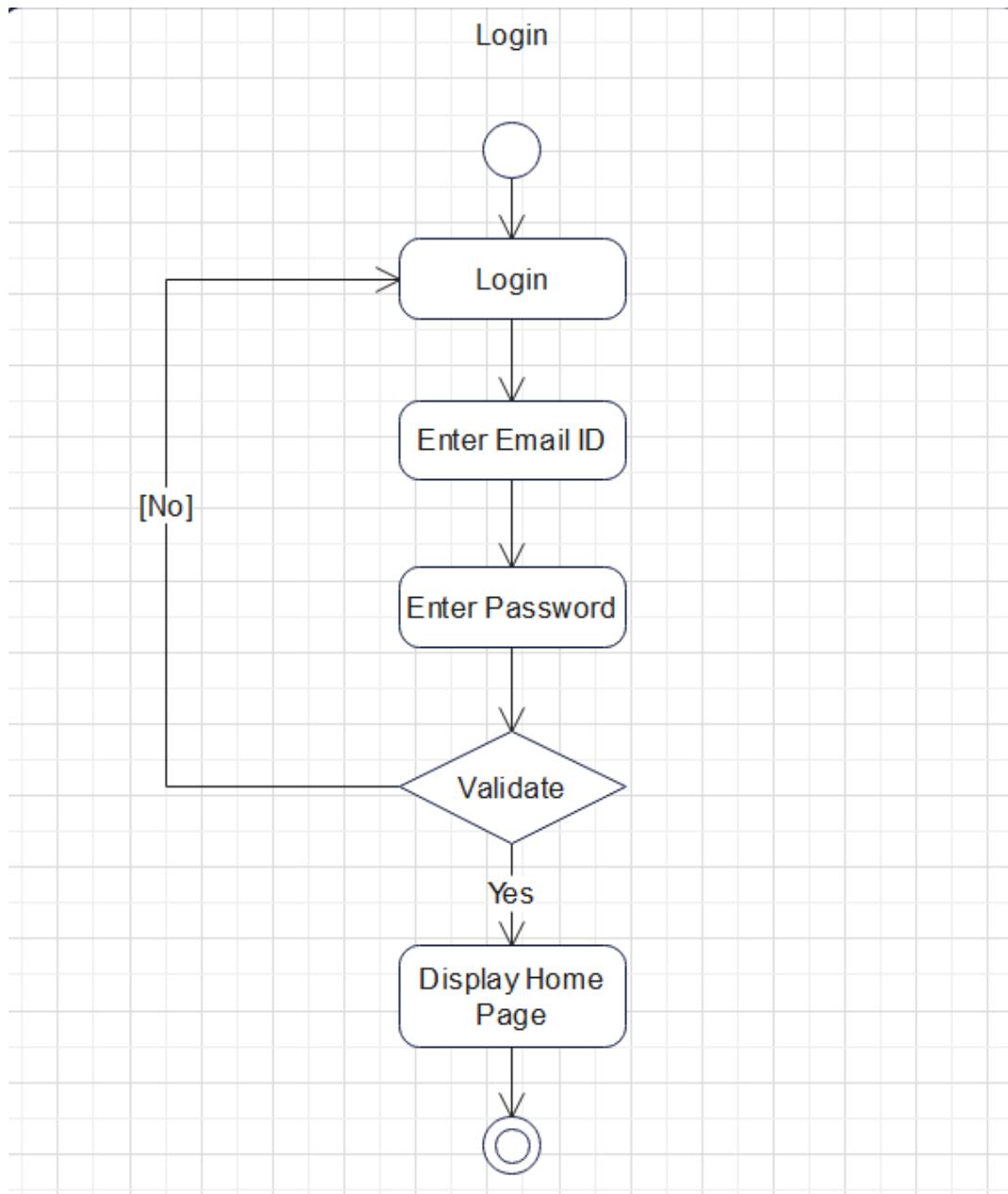
### 3.7 Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes (i.e. workflows). Activity diagrams show the overall flow of control.

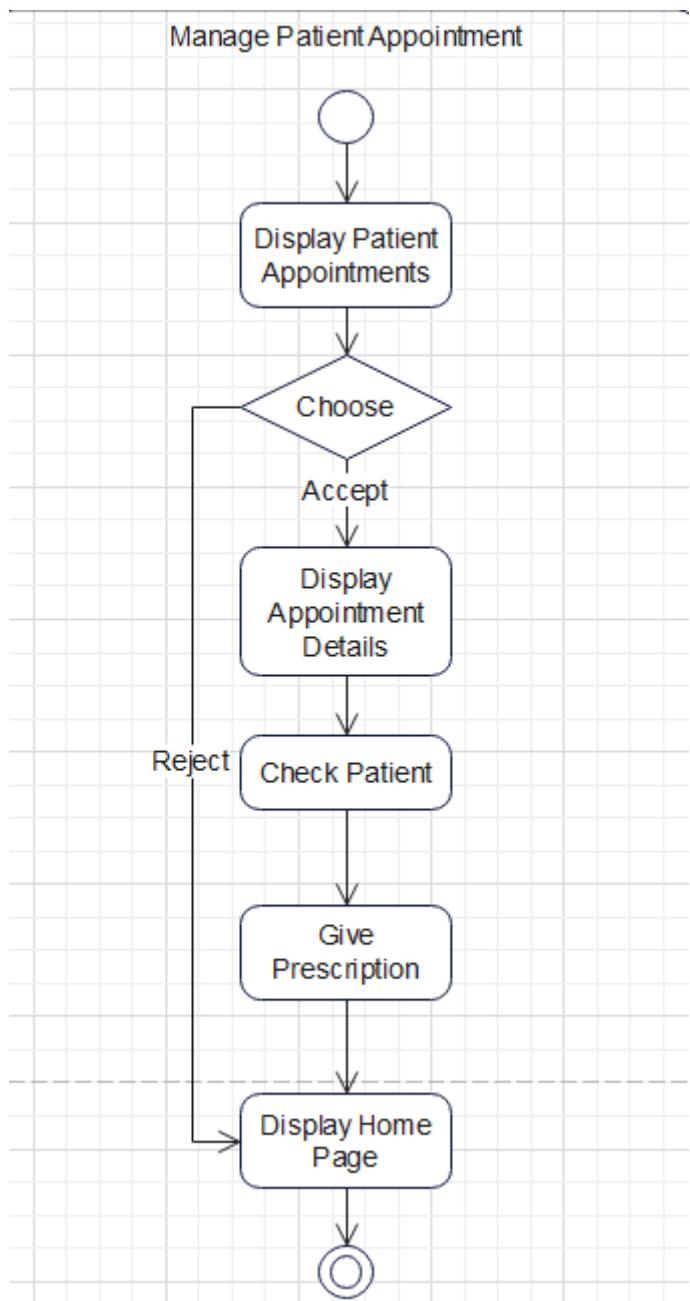
### Register Activity Diagram:



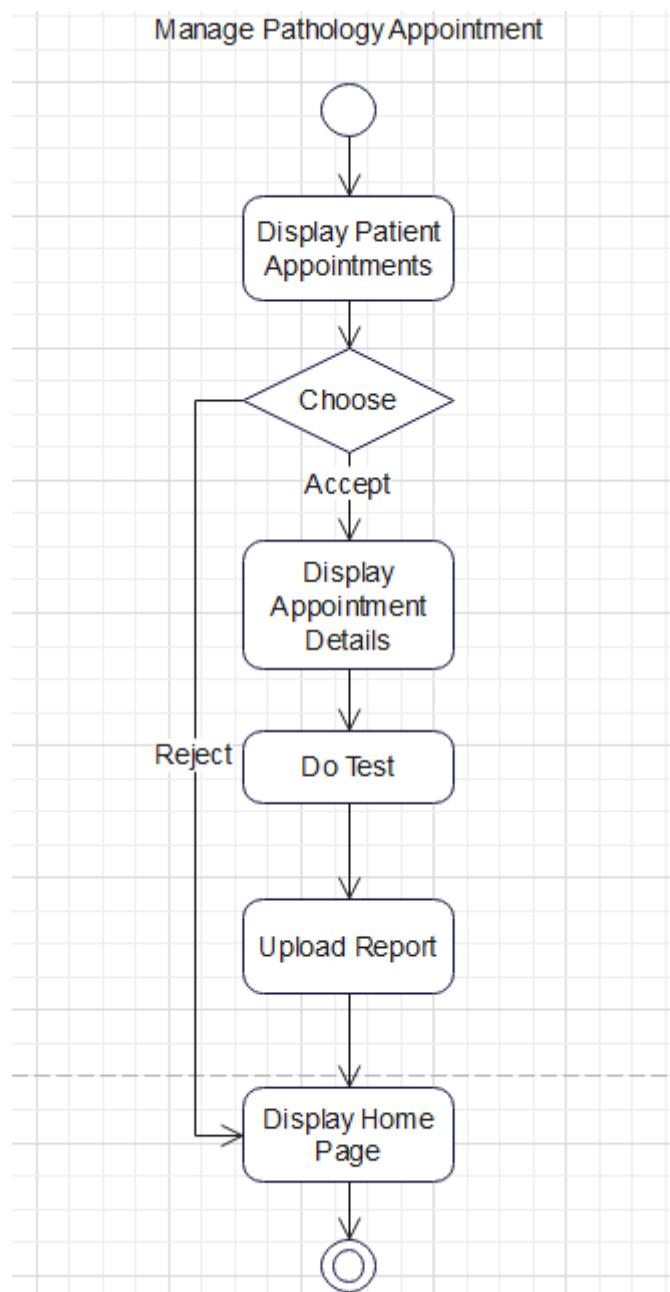
## Login Activity Diagram:



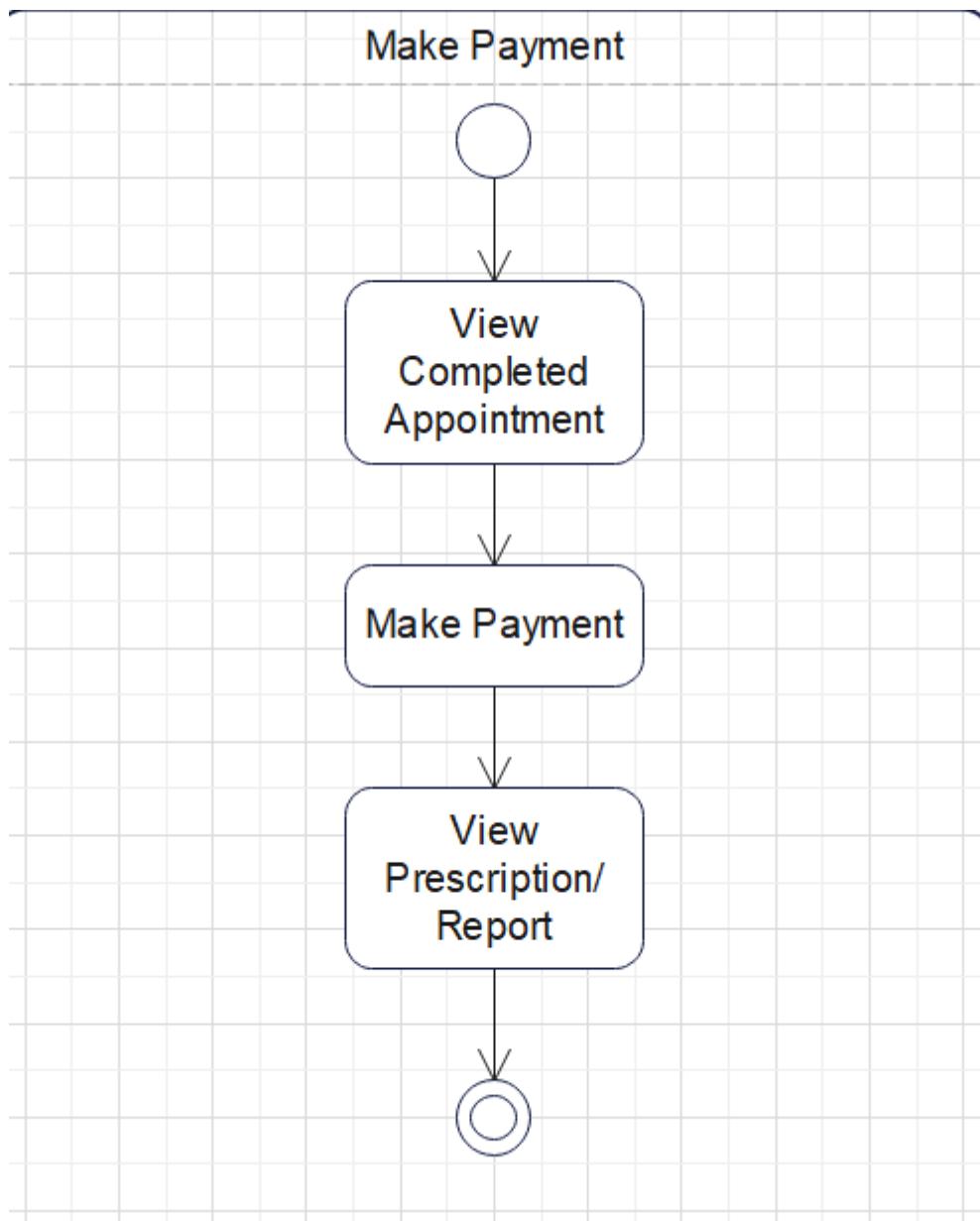
## Manage Patient Activity Diagram:



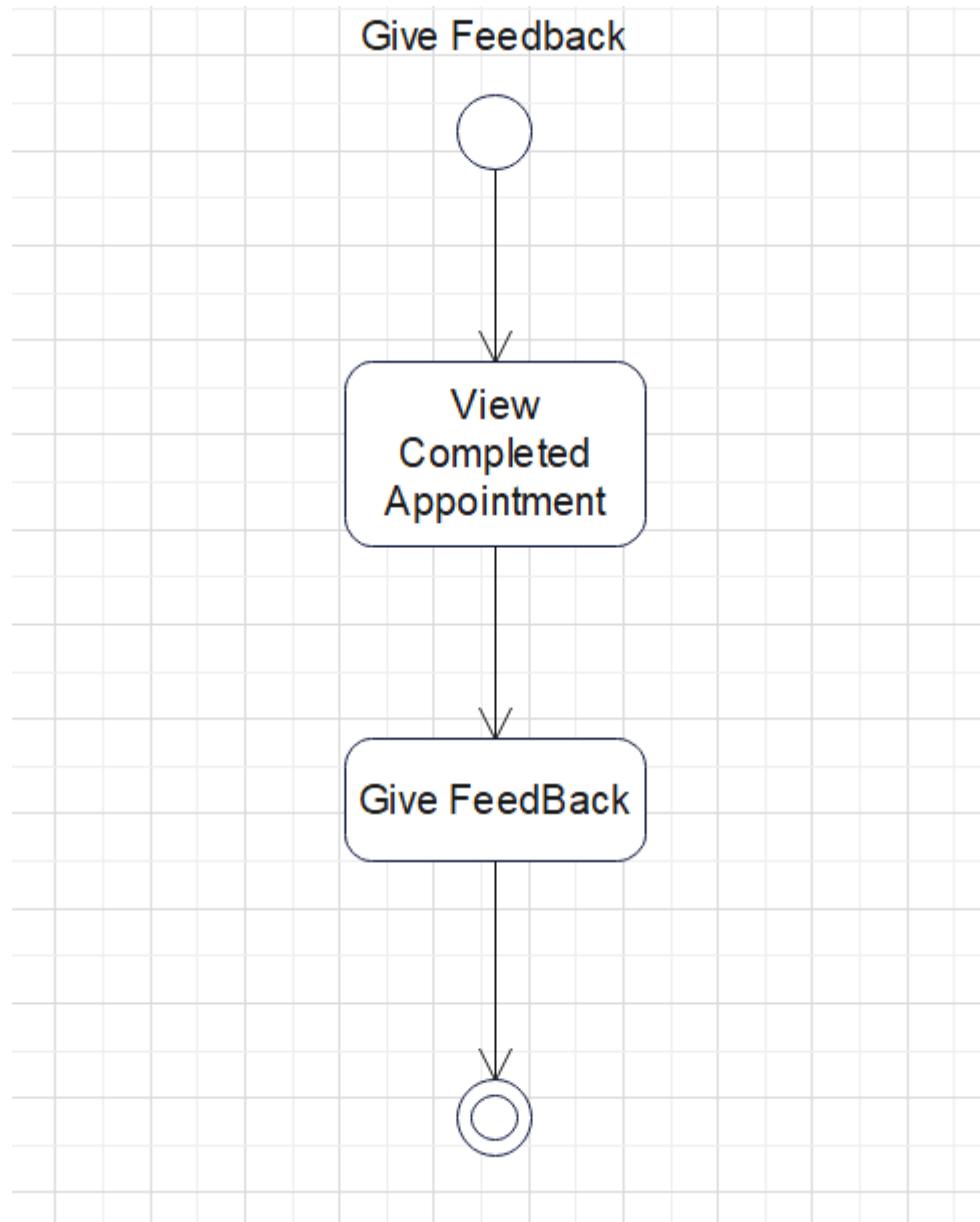
## Manage Pathology Activity Diagram:



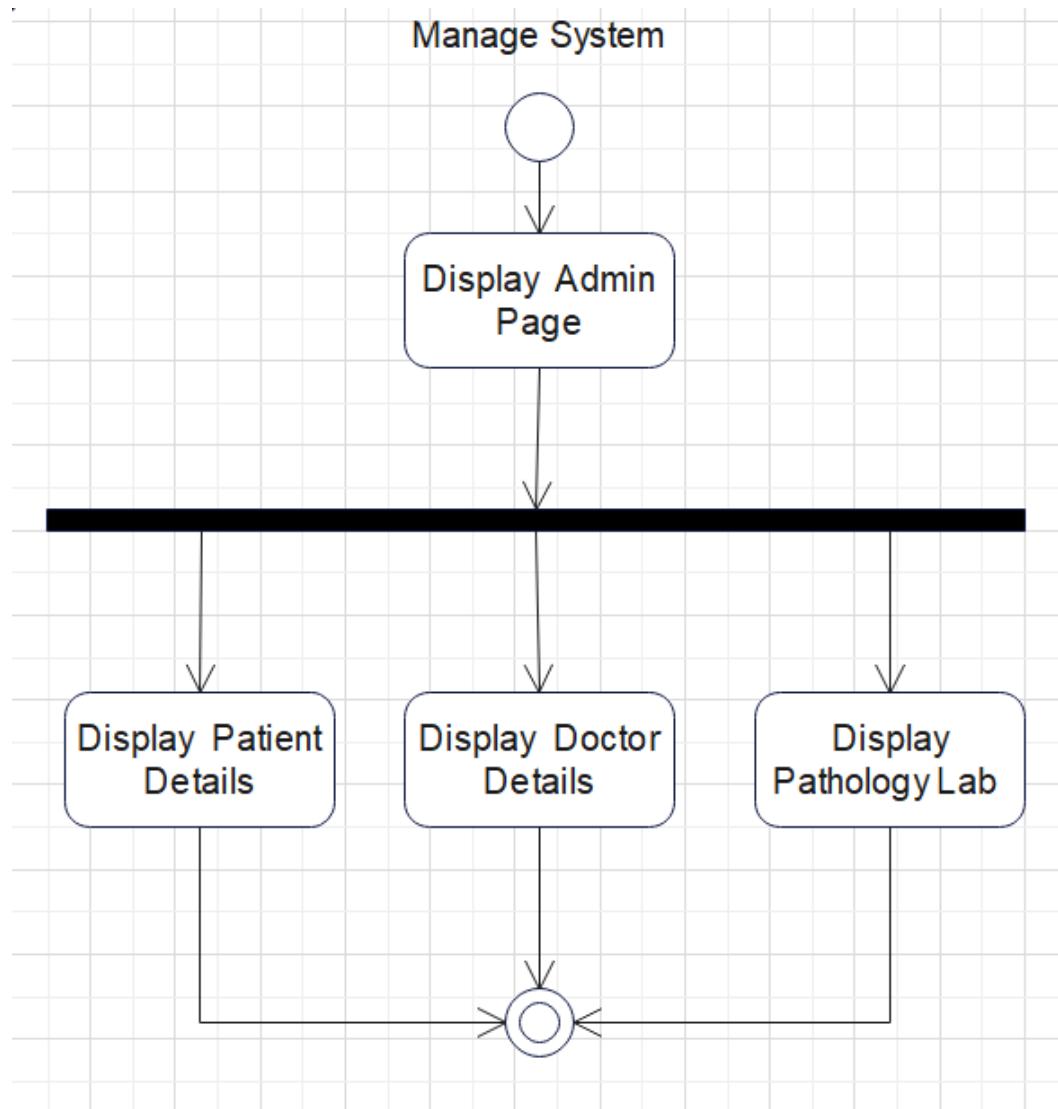
## Make Payment Activity Diagram:



## Give Feedback:

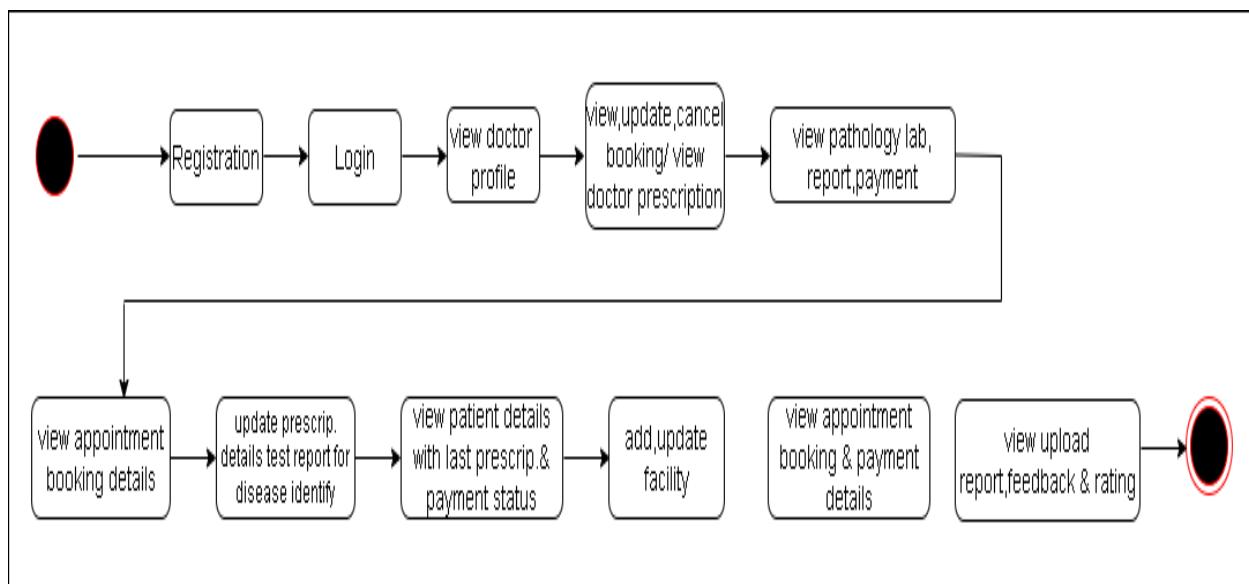


## Manage System:

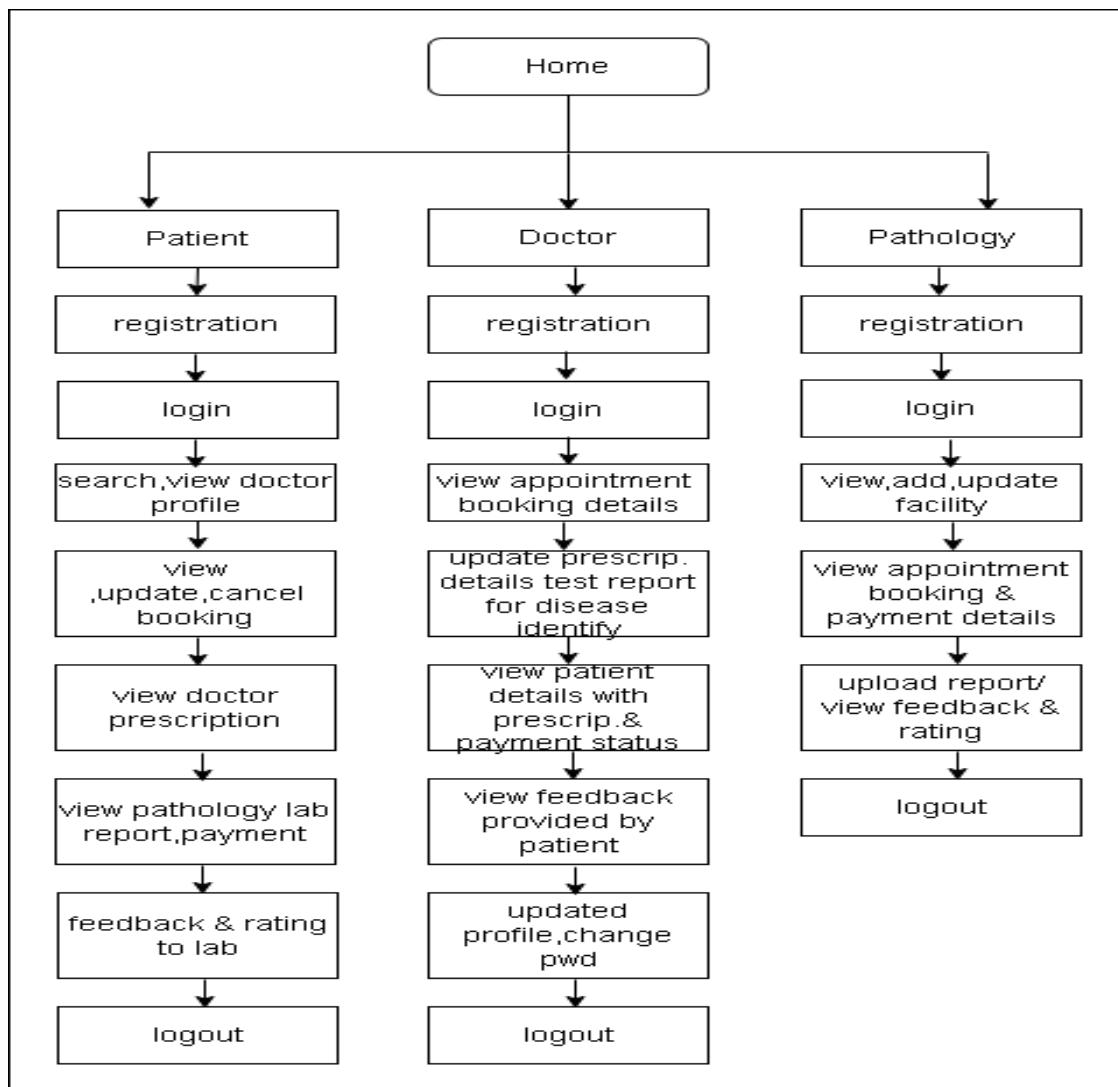


### 3.8 State Chart Diagram:

A state diagram (also known as a state machine or statechart diagram) is an illustration of all the possible behavioral states a software system component may exhibit and the various state changes it's predicted to undergo over the course of its operations.

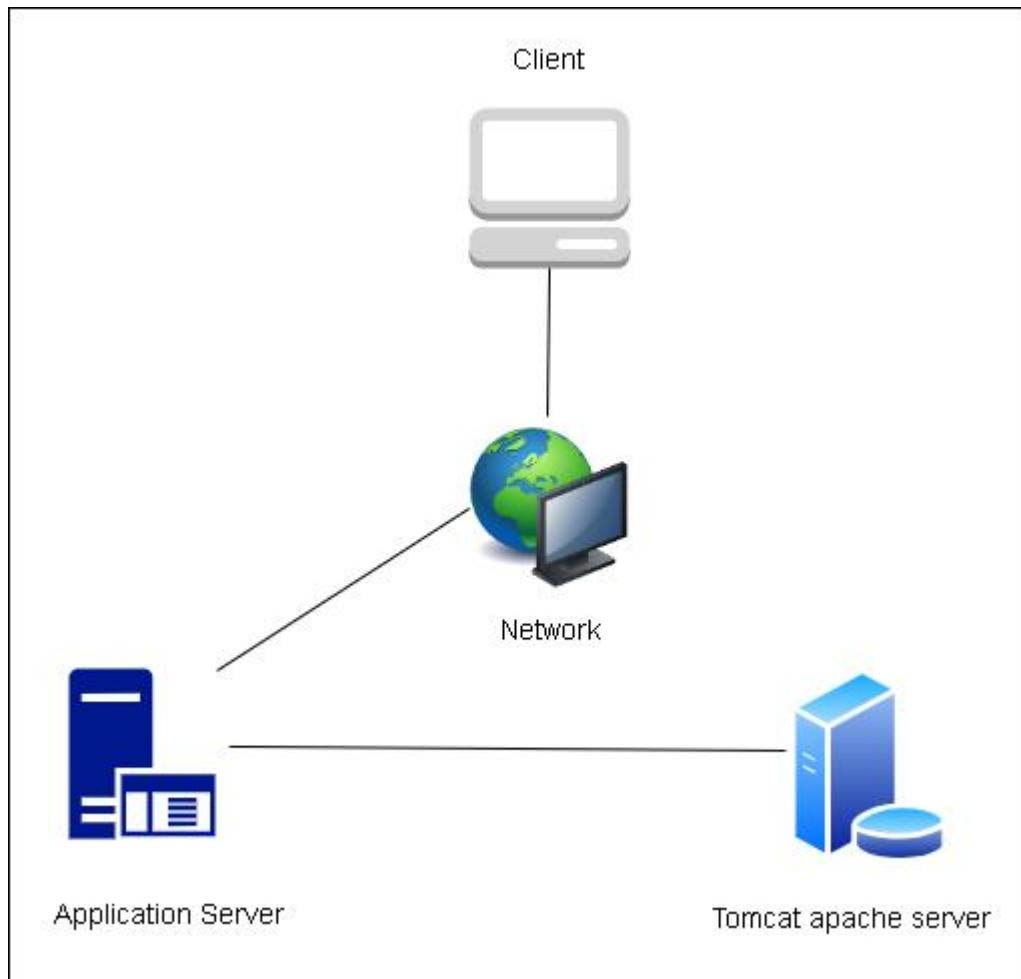


### 3.9 Interface Diagram (In Case of Embedded System):



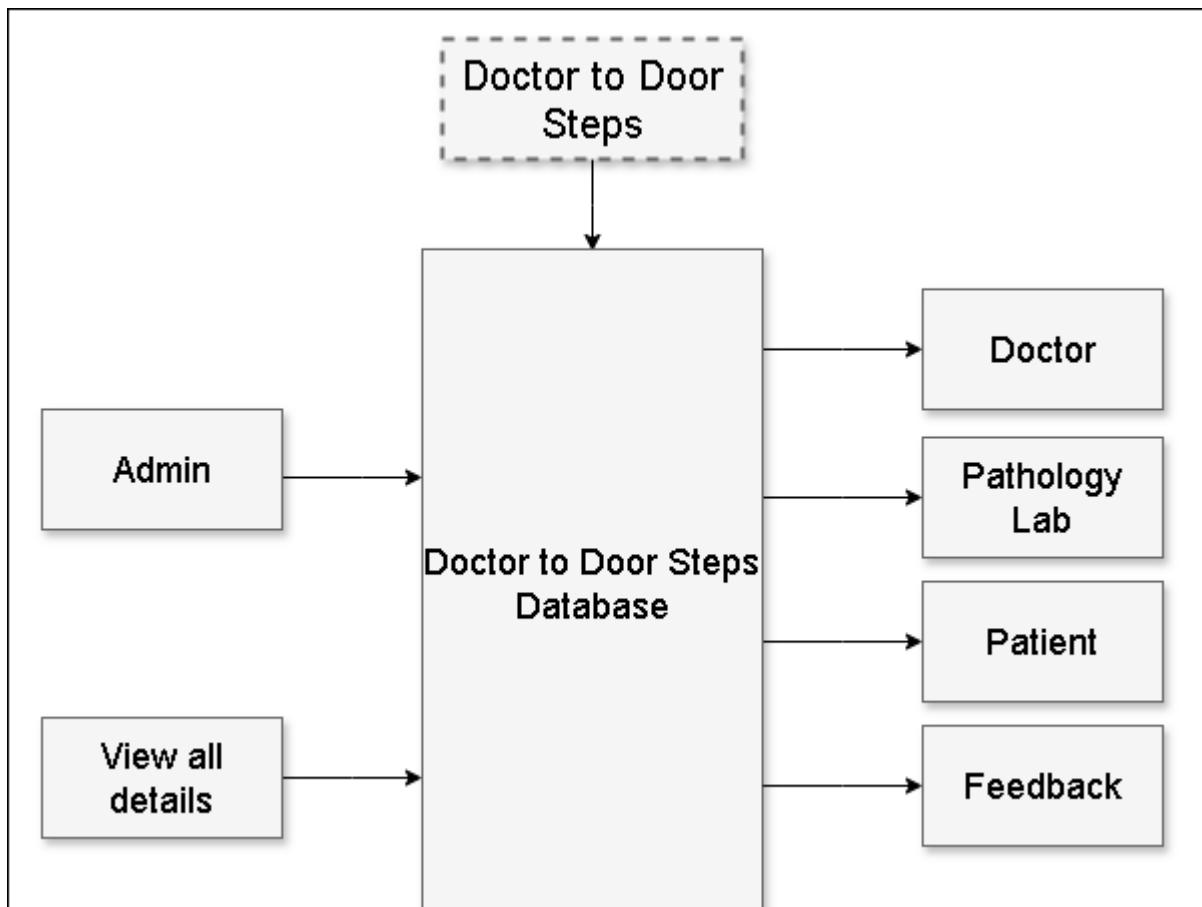
### **3.10 Deployment Diagram:**

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related.



### 3.11 Collaboration Diagram:

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). Developers can use these diagrams to portray the dynamic behavior of a particular use case and define the role of each object.



## **3.12 Module Specifications:**

### Module Specification for "Doctor To Door Steps" Application

The Doctor To Door Steps system is divided into multiple modules, each designed to handle specific functionalities for patients, doctors, pathology labs, and administrators. Below is the detailed module specification, covering key features and interactions.

#### 1. Patient Module

##### Features:

- Patient Registration & Login
  - New users register with name, contact details, address, and IP tracking.
  - Secure login with username and password.
- Search & Book Doctors
  - Search doctors by specialization (MBBS, Dentist, Cardiologist, etc.).
  - View doctor profiles, including experience, consultation fees, and ratings.
  - Book an appointment by selecting available time slots.
- Manage Appointments
  - View upcoming and past appointments.
  - Modify or cancel booked appointments.
- View Prescriptions & Request Pathology Tests
  - Check prescriptions provided by doctors.

- Request pathology tests based on doctor's recommendations.
- Online Payments for Pathology Reports
  - Secure payment gateway for test reports.
  - Generate digital receipts for transactions.
- Provide Feedback & Ratings
  - Rate doctors and pathology labs based on service experience.
- Logout

## 2. Doctor Module

Features:

- Doctor Registration & Login
  - Register with personal details, qualifications, specialization, experience.
  - Secure login with credentials.
- Manage Appointments
  - View patient bookings.
  - Accept, reschedule, or cancel appointments.
  - Update appointment status (completed, pending, canceled).
- Prescriptions & Lab Recommendations
  - Provide prescriptions with medication details.
  - Suggest lab tests for patient diagnosis.
- Patient History Management
  - View previous consultations, prescriptions, and test results.
  - Check patient payment status.
- View Feedback & Ratings
  - Access patient reviews and ratings to improve services.
- Logout

### **3. Pathology Lab Module**

Features:

- Lab Registration & Login
  - Register with lab name, location, contact details.
  - Secure login with credentials.
- Handle Appointments & Payments
  - View patient requests for tests.
  - Accept payments before allowing report downloads.
- Upload & Manage Test Reports
  - Upload patient reports in PDF or image format.
  - Allow patients to download reports upon successful payment.
- View Feedback & Ratings
  - Monitor patient reviews to enhance service quality.
- Logout

## 4. Admin Module

Features:

- User Management
  - Manage patient, doctor, and lab accounts.
- System Monitoring & Security
  - Track transactions and prevent fraudulent activities.
- Data Management & Backup
  - Maintain system database backups.
  - Ensure smooth operation by handling system issues.
- Logout

### **3.14 Sample Input and Output Screens (Screens must have valid data. All reports must have at-least 5 valid records.)**

Input Screens:

1. Patient Registration Page:

The screenshot shows a web browser window with the URL `localhost:8080/DoctorToDoorR1/patientRegistration.jsp`. The page title is "Doctor To Door Step". The main content is titled "Patient Registration". The form fields are as follows:

- First Name: test
- Middle Name: test
- Last Name: test
- Date of Birth: 29/01/2025
- Gender:
  - Male
  - Female
- Mobile Number: 1231231232
- Adhar Card Number: 121212121212
- Address in Details:  
test test test test

The browser status bar shows the date as 30-04-2025 and the time as 23:18.

**Doctor To Door Step**

Gender  
 Male  
 Female

Mobile Number  
 1231231232

Adhar Card Number  
 121212121212

Address in Details  
 test test test test

Email ID  
 test@gmail.com

Password  
 \*\*\*\*\*

**Register** **Reset**

© Copyright Doctor To Door Step. All Rights Reserved  
 Designed by SGMS INFOTECH LLP

## 2. Search Doctor/Pathology Page:

**Doctor To Door Step**

Welcome Patients Home

Welcome: Ritesh Rama Surange

Search : Doctor

MD Hematology Talegaon 15 / 05 / 2025 **Search**

© Copyright Doctor To Door Step. All Rights Reserved  
 Designed by SGMS INFOTECH LLP

**Twitter** **Facebook** **Instagram** **YouTube** **LinkedIn**



### 3. Doctor Appointment Details Page:

The screenshot shows a web browser window for 'Doctor To Door Step' on localhost. The page title is 'Appointment Booking Details'. It displays form fields for a doctor's appointment. The doctor's information includes Name: Rahul A Jadhav, DOB: 1993-02-12, Gender: Male, Mobile: 8900890076, Email-ID: rahul@gmail.com, Qualification: BHMS, and Address: Akurdi Pradhikaran Nigadi. The patient's information includes Name: Ritesh Rama Surange, Symptoms: test test, Appointment Date: 16 / 05 / 2025, and Time Slot: 12:00PM To 01:00PM. A 'Submit' button is at the bottom right. The browser status bar shows the date as 30-04-2025.

Doctor Name	Rahul A Jadhav	Patient Name	Ritesh Rama Surange
DOB	1993-02-12	Symptoms	test test
Gender	Male	Appointment Date	16 / 05 / 2025
Mobile	8900890076	Time Slot	12:00PM To 01:00PM
Email-ID	rahul@gmail.com		
Qualification	BHMS		
Address	Akurdi Pradhikaran Nigadi		

### 4. Pathology Appointment Details Page:

The screenshot shows a web browser window for 'Doctor To Door Step' on localhost. The page title is 'Appointment Booking Details'. It displays form fields for a pathology appointment. The doctor's information includes Name: Rahul Santosh Ratho, DOB: 2022-06-30, Gender: Male, Mobile: 8900890089, Email-ID: rahul.lab@gmail.com, Qualification: MD, and Address: Nigadi. The patient's information includes Patient Name: Ritesh Rama Surange, Select Report: Ritesh Rama Surange--test, Appointment Date: 13 / 05 / 2025, and Time Slot: 01:00PM To 02:00PM. A 'Submit' button is at the bottom right. The browser status bar shows the date as 30-04-2025.

Doctor Name	Rahul Santosh Ratho	Patient Name	Ritesh Rama Surange
DOB	2022-06-30	Select Report	Ritesh Rama Surange--test
Gender	Male	Appointment Date	13 / 05 / 2025
Mobile	8900890089	Time Slot	01:00PM To 02:00PM
Email-ID	rahul.lab@gmail.com		
Qualification	MD		
Address	Nigadi		

## 5. Feedback Details Page:

The screenshot shows a web browser window for 'Doctor To Door Step' on localhost:8080. The title bar says 'localhost:8080/DoctorToDoorR1/pl...' and 'localhost / 127.0.0.1 / doctor\_to...'. The main content area is titled 'Patient Provide Feedback related to Doctor'. It has a text input field labeled 'Feedback' containing 'test test test test' and a rating section with five yellow stars. A 'Submit Feedback' button is below. The top navigation bar includes links for Home, Search, Book Appointment, Lab Report, Download Report, Feedback, and Logout. The bottom footer shows copyright information, social media icons, and a Windows taskbar.

## 6. Doctor/Pathology Registration Page:

The screenshot shows a web browser window for 'Doctor To Door Step' on localhost:8080. The title bar says 'localhost:8080/DoctorToDoorR1/doctorRegistration.jsp'. The main content area is titled 'Doctor / Pathology Lab Registration'. It contains several form fields: First Name (Test), Middle Name (Test), Last Name (Test), Date of Birth (09/05/2023), Gender (Male selected), Mobile Number (1221121121), Adhar Card Number (421342234234), Account Type (Doctor selected), Working Area Name (test test). The top navigation bar includes links for Home, Doctor / Pathology, Patients, Admin, and Near Medical Details. The bottom footer shows a Windows taskbar.

**Doctor To Door Step**

localhost:8080/DoctorToDoorR1/doctorRegistration.jsp

Import bookmarks... Finish setup

Home Doctor / Pathology Patients Admin Near Medical Details

Doctor Pathology Lab

Working Area Name: test test

Qualification: MD

Speciality: Gastroenterology

Email ID: test@gmail.com

Address in Details: test test test test

Password: •••••

Register Reset

© Copyright Doctor To Door Step. All Rights Reserved  
Designed by SGMS INFOTECH LLP

Twitter Facebook Instagram LinkedIn Up arrow

Windows taskbar: 23:15 30-04-2025 ENG IN

## 7. Doctor/Pathology Login Page:

**Doctor To Door Step**

localhost:8080/DoctorToDoorR1/doctorLogin.jsp

Import bookmarks... Finish setup

contact@healthcare.com +91 1234567890 Twitter Facebook LinkedIn

Home Doctor / Pathology Patients Admin Near Medical Details



Doctor / Pathology Lab Login

Email ID: test@gmail.com

Password: •••••

Login

© Copyright Doctor To Door Step. All Rights Reserved  
Designed by SGMS INFOTECH LLP

Twitter Facebook Instagram LinkedIn Up arrow

Windows taskbar: 23:15 30-04-2025 ENG IN

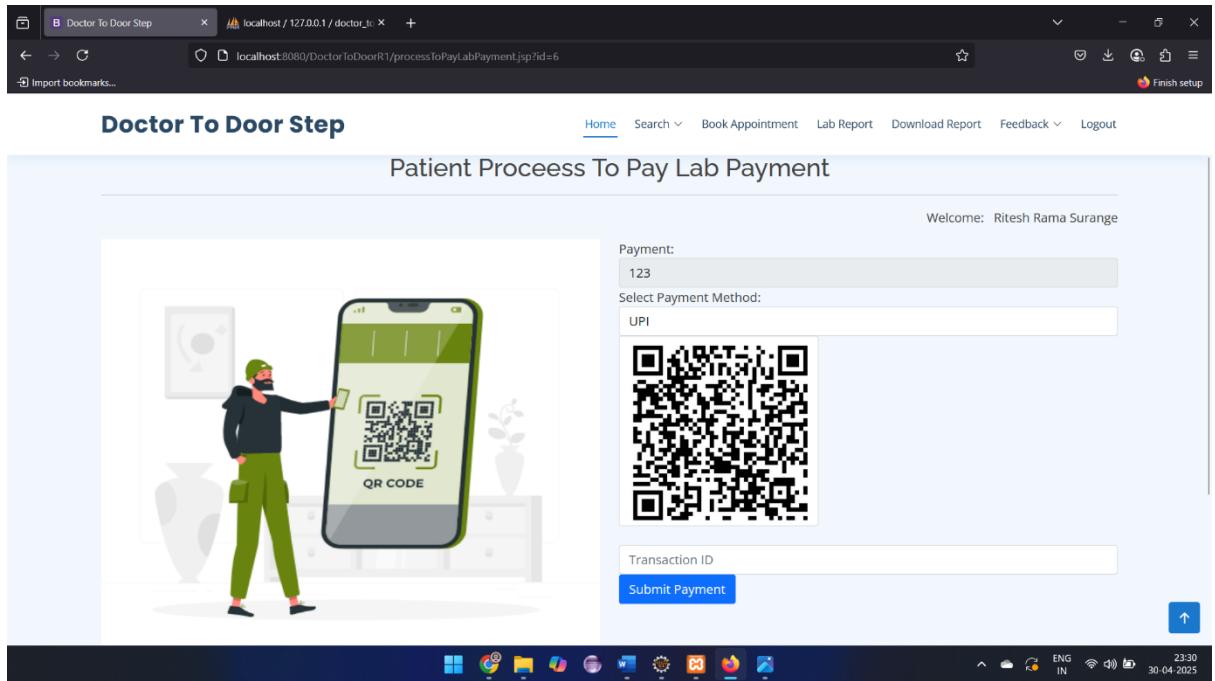
## 8. Patient Prescription Details Page:

The screenshot shows a web browser window for 'Doctor To Door Step' on localhost. The page title is 'Appointment Booking Details'. It displays patient information: Name - Ritesh Rama Surange, Symptoms - test test, Appointment Date - 2025-05-07, Time - 01:00PM To 02:00PM. The prescription field contains 'test'. The pathology lab report field also contains 'test'. Fees are listed as 1200, and the fees status is Pending. A green 'Submit' button is at the bottom right. The top navigation bar includes links for Home, Appointment Request, Confirm Appointment, View Patient, Lab Report, and Logout. The status bar at the bottom shows system icons and the date 30-04-2025.

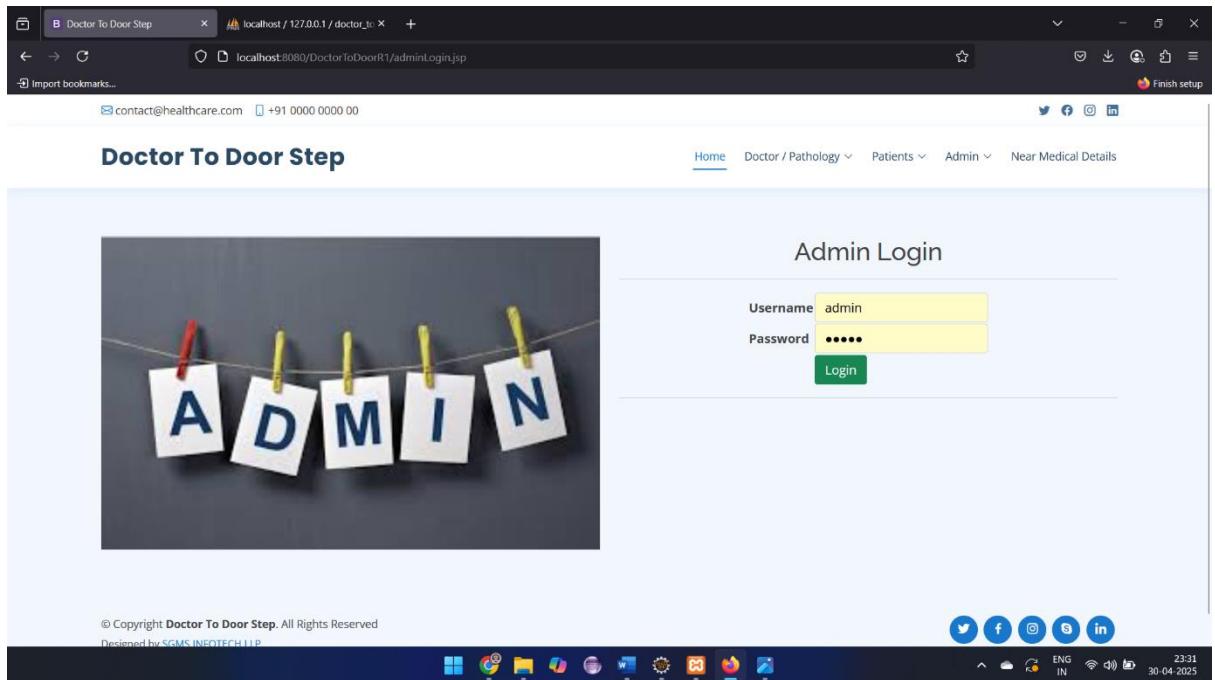
## 9. Upload Lab Report Page:

The screenshot shows a web browser window for 'Doctor To Door Step' on localhost. The page title is 'Upload Pathology Lab Report'. It features a background image of a scientist in a lab. The form fields include ID (4-Ritesh Rama Surange), File (Browse... 1234.png), Fees (123), and Fees Status (Pending). A green 'Upload' button is at the bottom right. The top navigation bar includes links for Home, View Appointment, Upload Report, and Logout. The status bar at the bottom shows system icons and the date 30-04-2025.

## 10. Lab Payment Details Page:

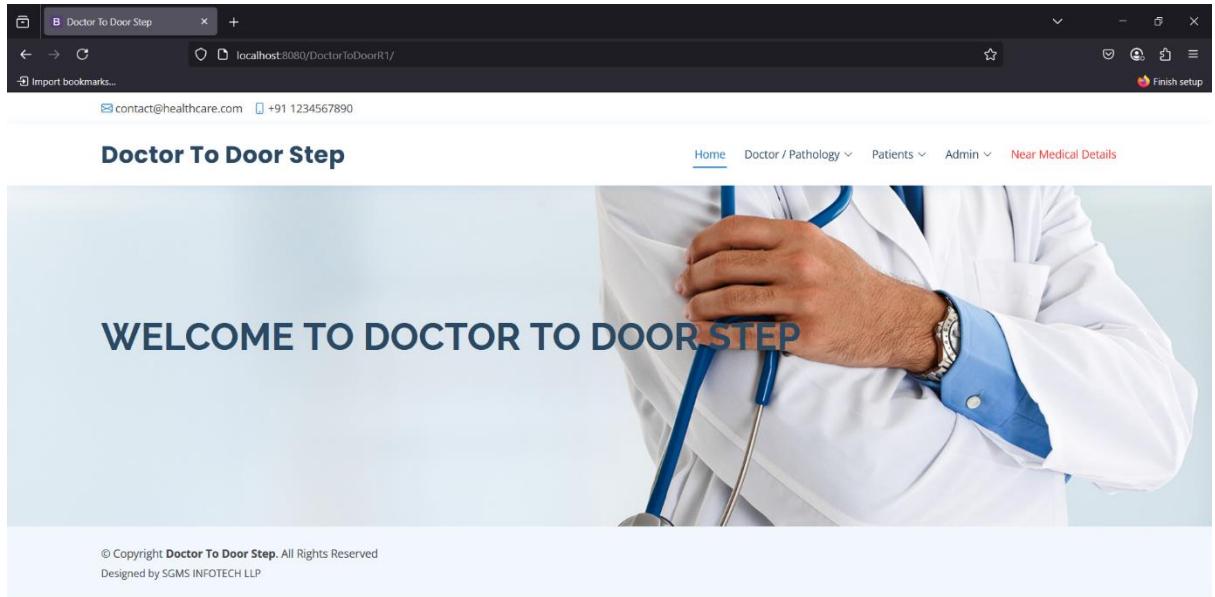


## 11. Admin Login Page:

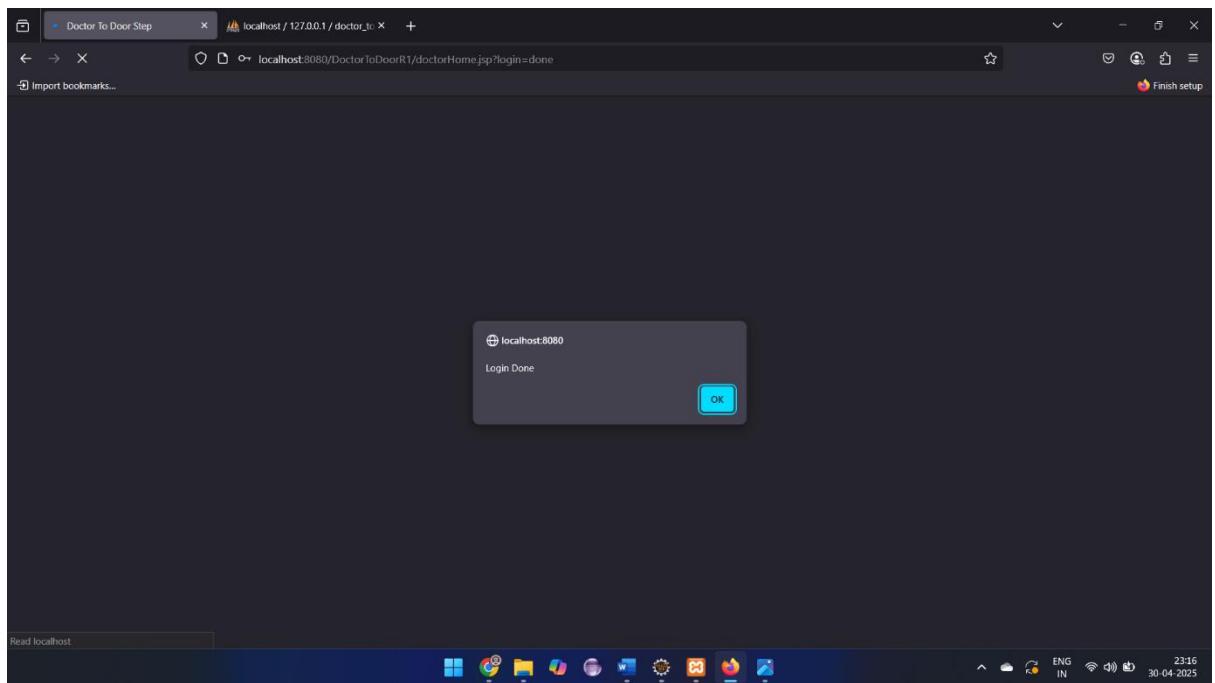


## Output:

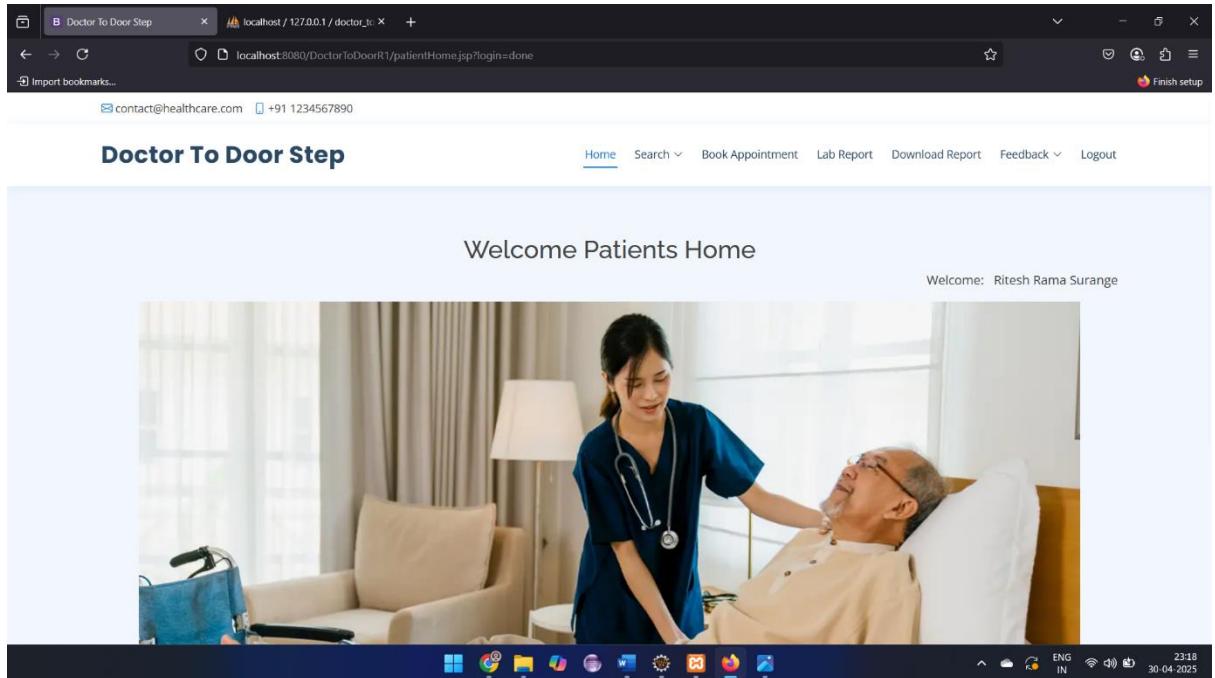
### 1. Home Page



### 2. Notification Page



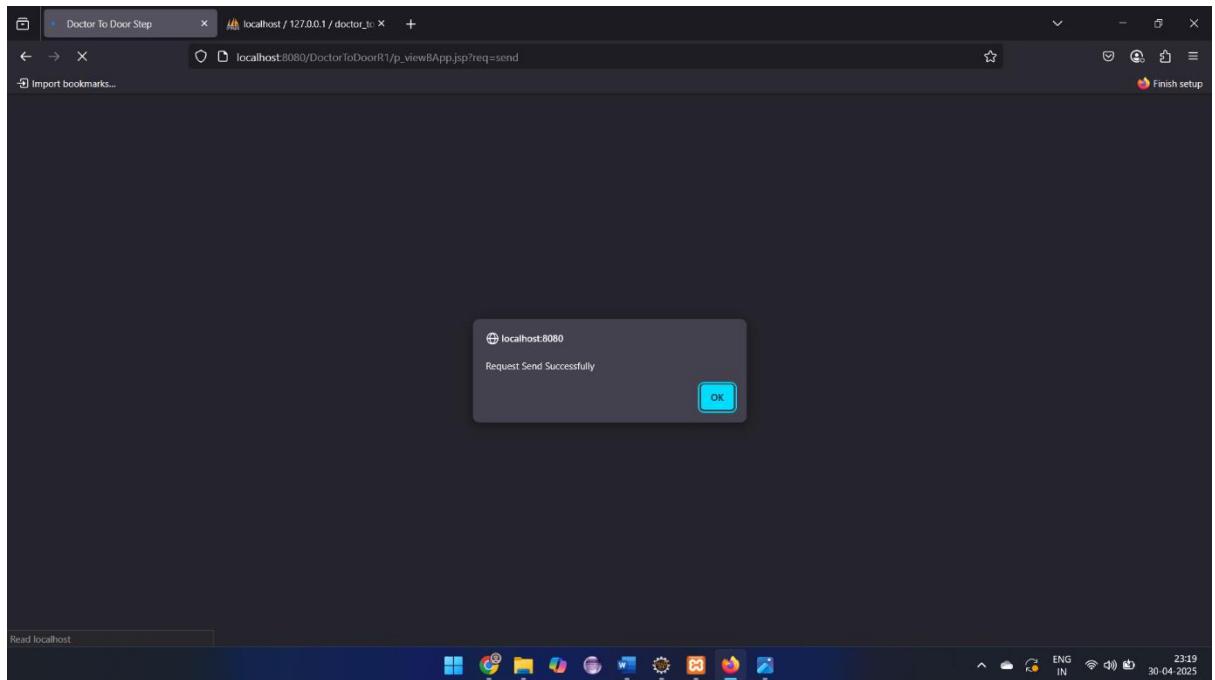
### 3. Patient Home Page:



### 4. Searched Doctor/Pathology Page:

A screenshot of a web browser window showing the 'Doctor To Door Step' search doctor page. The URL in the address bar is 'localhost:8080/DoctorToDoorR1/searchDoctor.jsp'. The page title is 'Doctor To Door Step'. The main content area features a heading 'Search : Doctor' and a search form with four dropdown fields labeled '--Select--' and a date input field 'dd / mm / yyyy' followed by a calendar icon and a green 'Search' button. Below the search form is a table listing four doctors. The table columns are: Sr.No, Doctor Details, Type, Mobile, Email-ID, and Address. Each row contains a doctor's name, qualifications, gender, age, fees, and contact information. The last column of each row has a 'Book Appoinment' link. The browser taskbar at the bottom shows various application icons.

## 5. Submitted Appointment Notification page



## 6. View Appointment page:

A screenshot of a web application titled 'Doctor To Door Step'. The main header includes links for 'Home', 'Search', 'Book Appointment', 'Lab Report', 'Download Report', 'Feedback', and 'Logout'. Below the header, a section titled 'View Appointment' lists four patient records in a table format. Each row contains details such as Sr.No, Doctor Details, Patient Details, Appointment Details, Status, and Operation (with links like 'Cancel Appointment', 'Check Prescription', and 'Provide Feedback').

Sr.No	Doctor Details	Patient Details	Appointment Details	Status	Operation
1	Rahul A Jadhav BHMS Address: Akurdi Pradhikaran Nigadi Mb. 8900890076	Ritesh Rama Surange <b>Symptoms:</b> test test	Date: 2025-05-16 Time: 12:00PM To 01:00PM	Waiting	<a href="#">Cancel Appointment</a>
2	Yugandhara L Rothe MD Address: Akurdi Nigadi Pradhikaran-411044 Mb. 9800090000	Ritesh Rama Surange <b>Symptoms:</b> test	Date: 2025-04-24 Time: 01:00PM To 02:00PM	Check	<a href="#">Check Prescription</a> <a href="#">Provide Feedback</a>
3	Pratik D Sonawane MD Address: Akurdi Pune Mb. 9800980098	Ritesh Rama Surange <b>Symptoms:</b> High fever	Date: 2024-03-25 Time: 01:00PM To 02:00PM	Check	<a href="#">Check Prescription</a>
4	Pratik D Sonawane MD Address: Akurdi Pune	Ritesh Rama Surange <b>Symptoms:</b> High Fever and Ometting	Date: 2024-03-25 Time: 01:00PM To 02:00PM	Check	<a href="#">Check Prescription</a>

## 7. View Prescription Page:

The screenshot shows a web browser window for 'Doctor To Door Step' on localhost. The title bar says 'localhost / 127.0.0.1 / doctor\_to...'. The main content area is titled 'Patient View Prescription'. It displays patient information: Name (Ritesh Rama Surange), Symptoms (test), Appointment Date (2025-04-24), Time (01:00PM To 02:00PM), Precision (test), Lab Report (test), Fees (122), and Fees Status (Received). The bottom status bar shows system icons and the date 30-04-2025.

## 8. View Request for Pathology page:

The screenshot shows a web browser window for 'Doctor To Door Step' on localhost. The title bar says 'localhost / 127.0.0.1 / doctor\_to...'. The main content area is titled 'View Request for Pathology Lab'. It displays a table of pathology requests:

Sr.No	Patient Name	Dr.Name (Lab)	Appointment Date	Time	Request Date	Report Required	Status
1	Ritesh Rama Surange	Sakshi A Jadhav	2024-03-25	04:00PM To 05:00PM	2024-03-24 00:35:01.0	Blood Test Urin Test	Accept
2	sapna surange	A B C	2024-03-23	03:00PM To 04:00PM	2024-03-22 15:27:26.0	Blood Tesh	Accept

At the bottom, there is a copyright notice: '© Copyright Doctor To Door Step. All Rights Reserved' and 'Designed by SGMS INFOTECH LLP'. There are also social media sharing icons for Twitter, Facebook, LinkedIn, and others.

## 9. Download Report Page:

Sr.No	Patient Name	Dr Name(Lab)	Dr. Name(Refere By)	Upload Time	Report Name	Operation
1	Ritesh Rama Surange Ref/ Dr. Pratik D Sonawane	A B C	2024-03-22	Blood Tesh	Fees: NA Fees Status: NA	
2	Ritesh Rama Surange Ref/ Dr. Pratik D Sonawane	Sakshi A Jadhav	2024-03-24	Blood Test Urin Test	Fees: NA Fees Status: NA	
3	Ritesh Rama Surange Ref/ Dr. Pratik D Sonawane	Sakshi A Jadhav	2024-03-24	Blood Test Urin Test	Fees: NA Fees Status: NA	
4	Ritesh Rama Surange Ref/ Dr. Pratik D Sonawane	Sakshi A Jadhav	2024-03-24	Blood Test Urin Test	Fees: 400 Fees Status: Received	<a href="#">Download</a>

© Copyright Doctor To Door Step. All Rights Reserved  
Designed by SGMS INFOTECH LLP

Sr.No	Patient Name	Dr Name(Lab)	Dr. Name(Refere By)	Upload Time	Report Name	Operation
1	Ritesh Rama Surange Ref/ Dr. Pratik D Sonawane	A B C	2024-03-22	Blood Tesh	Fees: NA Fees Status: NA	
2	Ritesh Rama Surange Ref/ Dr. Pratik D Sonawane	Sakshi A Jadhav	2024-03-24	Blood Test Urin Test	Fees: NA Fees Status: NA	
3	Ritesh Rama Surange Ref/ Dr. Pratik D Sonawane	Sakshi A Jadhav	2024-03-24	Blood Test Urin Test	Fees: NA Fees Status: NA	
4	Ritesh Rama Surange Ref/ Dr. Pratik D Sonawane	Sakshi A Jadhav	2024-03-24	Blood Test Urin Test	Fees: 400 Fees Status: Received	<a href="#">Download</a>
5	Ritesh Rama Surange Ref/ Dr. Yugandhara L Rothe	Rahul Santosh Ratho	2025-04-30	test	Fees: 123 Fees Status: Pending	<a href="#">Process To Pay</a>

## 10. Downloaded Report Page:

The screenshot shows a web browser window for 'Doctor To Door Step' on localhost. The title bar says 'localhost / 127.0.0.1 / doctor\_to...'. The main content area is titled 'Download Reports' and displays a table of uploaded reports. The table columns are: Sr.No, Patient Name, Dr Name(Lab), Dr. Name(Refere By), Upload Time, Report Name, and Operation. The fourth row shows a report for 'Ritesh Rama Surange' with a 'Download' link in the 'Operation' column. The bottom of the page includes a copyright notice, social media icons, and a system tray.

Sr.No	Patient Name	Dr Name(Lab)	Dr. Name(Refere By)	Upload Time	Report Name	Operation
1	Ritesh Rama Surange Ref/ Dr. Pratik D Sonawane	A B C	2024-03-22	Blood Tesh	Fees: NA Fees Status: NA	
2	Ritesh Rama Surange Ref/ Dr. Pratik D Sonawane	Sakshi A Jadhav	2024-03-24	Blood Test Urin Test	Fees: NA Fees Status: NA	
3	Ritesh Rama Surange Ref/ Dr. Pratik D Sonawane	Sakshi A Jadhav	2024-03-24	Blood Test Urin Test	Fees: NA Fees Status: NA	
4	Ritesh Rama Surange Ref/ Dr. Pratik D Sonawane	Sakshi A Jadhav	2024-03-24	Blood Test Urin Test	Fees: 400 Fees Status: Received	<a href="#">Download</a>

## 11. View Feedback page:

The screenshot shows a web browser window for 'Doctor To Door Step' on localhost. The title bar says 'localhost / 127.0.0.1 / doctor\_to...'. The main content area is titled 'View Provide Feedback' and displays a table of patient feedback. The table columns are: Sr.No, Doctor Details, Patient Details, Appointment Details, and Feedback. The third row shows a feedback entry with a rating of 4 stars. The bottom of the page includes a copyright notice, social media icons, and a system tray.

Sr.No	Doctor Details	Patient Details	Appointment Details	Feedback
1	Pratik D Sonawane MD Address: Akurdi Pune Mb. 9800980098	Patient: Ritesh Rama Surange <b>Symptoms:</b> High Fever and Ometting <b>Fees:</b> 200	Date: 2024-03-25 Time: 01:00PM To 02:00PM	thanks Doctor Saheb <b>Rating:</b> 5 ★
2	Pratik D Sonawane MD Address: Akurdi Pune Mb. 9800980098	Patient: Ritesh Rama Surange <b>Symptoms:</b> High Fever and Ometting <b>Fees:</b> 200	Date: 2024-03-25 Time: 01:00PM To 02:00PM	Thanks Service Provide Lab <b>Rating:</b> 4 ★
3	Pratik D Sonawane MD Address: Akurdi Pune Mb. 9800980098	Patient: Ritesh Rama Surange <b>Symptoms:</b> High fever <b>Fees:</b> NA	Date: 2024-03-25 Time: 01:00PM To 02:00PM	Thanks <b>Rating:</b> 4 ★

## 12. Doctor Home Page:

The screenshot shows a web browser window for 'Doctor To Door Step' on 'localhost:8080'. The title bar says 'localhost / 127.0.0.1 / doctor\_to...'. The page header includes a contact link 'contact@healthcare.com +91 1234567890', social media icons, and a 'Finish setup' button. The main content area has a blue header 'Doctor To Door Step'. Below it, a red banner reads 'Welcome Doctor Home'. On the right, it says 'Welcome: Yugandhara L Rothe'. A table displays the doctor's profile:

Doctor Name	Yugandhara L Rothe MD
Date of Birth	1992-12-12
Gender	Female
Mobile Number	9800090000
UID Number	89008900890089
Email-ID	yuga@gmail.com
Area Name	Pune
Address	Akurdi Nigadi Pradhikaran-411044
Average Rating	0.0 ★

The status bar at the bottom shows system icons, language 'ENG IN', date '30-04-2025', and time '23:17'.

## 13. View Appointment request from patient page:

The screenshot shows a web browser window for 'Doctor To Door Step' on 'localhost:8080'. The title bar says 'localhost / 127.0.0.1 / doctor\_to...'. The page header includes a contact link 'contact@healthcare.com +91 1234567890', social media icons, and a 'Finish setup' button. The main content area has a blue header 'Doctor To Door Step'. Below it, a blue banner reads 'View Appointment Request From Patients'. A table lists the appointment requests:

Sr.No	Patient Name	Symptoms	Appointment Date	Time	Operation
1	Ritesh Rama Surange	test test	2025-05-07	01:00PM To 02:00PM	<a href="#">Accept</a>   <a href="#">Reject</a>

The status bar at the bottom shows system icons, language 'ENG IN', date '30-04-2025', and time '23:23'.

## 14. Confirm Appointment page:

The screenshot shows a web browser window for 'Doctor To Door Step' on localhost. The title bar says 'localhost / 127.0.0.1 / doctor\_tc'. The main content area is titled 'View Confirm Appointment'. A table displays patient details, symptoms, appointment details, and operation status. At the bottom, there's a copyright notice and social media sharing icons.

Sr.No	Patient Details	Symptoms	Appointment Details	Operation
1	Ritesh Rama Surange Mobile: 9800980089; Age: 24	test test	Date: 2025-05-07 Time: 01:00PM To 02:00PM	<a href="#">Check Patient</a>

© Copyright Doctor To Door Step. All Rights Reserved  
Designed by SGMS INFOTECH LLP

Twitter Facebook Instagram LinkedIn



## 15. View Patient Page:

The screenshot shows a web browser window for 'Doctor To Door Step' on localhost. The title bar says 'localhost / 127.0.0.1 / doctor\_tc'. The main content area is titled 'Doctor View Patients'. A table displays patient details, symptoms, appointment details, fees details, and operation status. At the bottom, there's a copyright notice and social media sharing icons.

Sr.No	Patient Details	Symptoms	Appointment Details	Fees Details	Operation
1	Ritesh Rama Surange Mobile: 9800980089; Age: 24	test test	Date: 2025-05-07 Time: 01:00PM To 02:00PM	Fees:1200 Fees Status: Pending	<a href="#">Check Prescription</a>
2	Ritesh Rama Surange Mobile: 9800980089; Age: 24	test	Date: 2025-04-24 Time: 01:00PM To 02:00PM	Fees:122 Fees Status: Received	<a href="#">Check Prescription</a>

© Copyright Doctor To Door Step. All Rights Reserved  
Designed by SGMS INFOTECH LLP

Twitter Facebook Instagram LinkedIn



## 16. Pathology Home Page:

Welcome PathologyLab Home

Doctor Name	Abhishek AA Bidkar BSc-Pathology
Date of Birth	1900-12-12
Gender	Male
Mobile Number	9800098800
UID Number	1231231231231
Email-ID	abhi@gmail.com
Area Name	Akurdi
Address	Akurdi
Average Rating	0.0

## 17. Pathology Appointment Page:

Pathology Lab View Request for Sample

Sr.No	Patient Details	Appointment Details	Request Date	Report Required	Status
1	Ritesh Rama Surange Ref. Dr.Yugandhara L Rothe	Date: 2025-05-13 Time: 01:00PM To 02:00PM	2025-04-30 23:26:16.0	test	waiting <a href="#">Accept   Reject</a>

## 18. View Pathology Upload Page:

The screenshot shows a web browser window for 'Doctor To Door Step' on localhost. The title bar says 'localhost / 127.0.0.1 / doctor\_to...' and the address bar shows 'localhost:8080/DoctorToDoorR1/lb\_viewReportRequestJsp'. The page header includes contact information ('contact@healthcare.com +91 1234567890') and navigation links ('Home', 'View Appointment', 'Upload Report', 'Logout'). The main content is titled 'Pathology Lab View Request for Sample' and displays a table with one row of data:

Sr.No	Patient Details	Appointment Details	Request Date	Report Required	Status
1	Ritesh Rama Surange Ref. Dr.Yugandhara L Rothe	Date: 2025-05-13 Time: 01:00PM To 02:00PM	2025-04-30 23:26:16.0	test	Accept <a href="#">Upload Report</a>

At the bottom, there's a copyright notice ('© Copyright Doctor To Door Step. All Rights Reserved. Designed by SGMS INFOTECH LLP') and social media sharing icons.



## 19. Admin Home Page:

The screenshot shows a web browser window for 'Doctor To Door Step' on localhost. The title bar says 'localhost / 127.0.0.1 / doctor\_to...' and the address bar shows 'localhost:8080/DoctorToDoorR1/adminHome.jsp?login=done'. The page header includes contact information ('contact@healthcare.com +91 1234567890') and navigation links ('Home', 'Doctors', 'Pathology Labs', 'View Patients', 'Logout'). The main content displays a welcome message 'Welcome: Admin' and a copyright notice ('© Copyright Doctor To Door Step. All Rights Reserved. Designed by SGMS INFOTECH LLP').



## 20. Doctors View Page:

**Doctor To Door Step**

**Admin View All DoctorsLab Details**

Sr.No	Doctor Details	Mobile	Address	Email ID	Registration Date
1	Dr. Yugandhara L Rothe <b>Qualification:</b> MD <b>Specialist:</b> Cardiology <b>A/C Type:</b> Doctor <b>Gender:</b> Female <b>UID No.:</b> 89008900890089	9800090000	Akurdi Nigadi Pradhikaran-411044	yuga@gmail.com	2024-03-22
2	Dr. Rahul A Jadhav <b>Qualification:</b> BHMS <b>Specialist:</b> Dermatology <b>A/C Type:</b> Doctor <b>Gender:</b> Male <b>UID No.:</b> 987698769876	8900890076	Akurdi Pradhikaran Nigadi	rahul@gmail.com	2024-03-22
3	Dr. Prashant B' Patil <b>Qualification:</b> MD <b>Specialist:</b> Endocrinology	8900890077	Chinchwad Pune	prashant@gmail.com	2024-03-22

## 21. Pathology View Page:

**Doctor To Door Step**

**Admin View All Pathology Lab Details**

Sr.No	Doctor Details	Mobile	Address	Email ID	Registration Date
1	Dr. A B C <b>Qualification:</b> MD <b>Specialist:</b> Neurology <b>A/C Type:</b> PathologyLab <b>Gender:</b> Male <b>UID No.:</b> 1231231231231	9800980098	Nigadi	a.lab@gmail.com	2024-03-22
2	Dr. Rahul Santosh Ratho <b>Qualification:</b> MD <b>Specialist:</b> Oncology <b>A/C Type:</b> PathologyLab <b>Gender:</b> Male <b>UID No.:</b> 1231231231231	8900890089	Nigadi	rahul.lab@gmail.com	2024-03-22
3	Dr. Abhishek AA Bidkar <b>Qualification:</b> BSc-Pathology	9800098800	Akurci	abhi@gmail.com	2024-03-22

## 22. View Patient Page:

The screenshot shows a web browser window titled "Doctor To Door Step" with the URL "localhost:8080/DoctorToDoorR1/adminViewPatients.jsp". The page header includes a logo, contact information ("contact@healthcare.com +91 1234567890"), and navigation links for Home, Doctors, Pathology Labs, View Patients, and Logout. The main content area is titled "Admin View All Patients Details" and displays a table with two patient records. The table columns are Sr.No, Patient Details, Adhar ID, Mobile, Address, Email ID, and Registration Date.

Sr.No	Patient Details	Adhar ID	Mobile	Address	Email ID	Registration Date
1	Ritesh Rama Surange Gender: Male	123412341234	9800980089	Akurdi Pune	ritesh@gmail.com	2024-03-22
2	Yugandhara R Surange Gender: Female	567856785678	9876987667	ofice number 4 ABC Junction SN 26 Akurdi Pune	yuga@gmail.com	2024-03-23

At the bottom of the page, there is a copyright notice: "© Copyright Doctor To Door Step. All Rights Reserved" and "Designed by SGMS INFOTECH LLP". Social media sharing icons for Twitter, Facebook, LinkedIn, and others are also present.



## 23. Nearest Medical Store Page:

The screenshot shows a web browser window titled "Doctor To Door Step" with the URL "localhost:8080/DoctorToDoorR1/medicalDetails.jsp". The page header includes a logo, contact information ("contact@healthcare.com +91 1234567890"), and navigation links for Home, Doctor / Pathology, Patients, Admin, and Near Medical Details. The main content area is titled "Search Nearest Medical Store" and features a search bar with "Enter Pin Code" and a "Search" button. Below the search bar is a table listing six medical centers with their details. To the right of the table is a section titled "Emergency Contacts" with three entries.

Sr.No	Medical Name	Contact Person	Mobile
1	Pune Medical Center 1 <b>Address:</b> 123 Nigadi Road, Nigadi - 411044	John Doe	123-456-7890
2	Pune Medical Center 2 <b>Address:</b> 456 Akurdi Road, Akurdi - 411035	Jane Smith	456-789-0123
3	Pune Medical Center 3 <b>Address:</b> 789 Pimpri Road, Pimpri - 411018	David Johnson	789-012-3456
4	Pune Medical Center 4 <b>Address:</b> 321 Chinchwad Road, Chinchwad - 411019	Mary Brown	012-345-6789
5	Pune Medical Center 5 <b>Address:</b> 654 Dapodi Road, Dapodi - 411012	Michael Lee	987-654-3210
6	Pune Medical Center 6 <b>Address:</b> 987 Aundh Road, Aundh - 411007	Sarah Taylor	654-321-0987

**Emergency Contacts**

1	Police Station	100
2	Ambulance	101
3	Road Accident	103

At the bottom of the page, there is a Windows taskbar showing various pinned icons and system status.

# **4. Coding**

## 4.1 Code snippets

Patient Registration Code:

```
<%@ page language="java" contentType="text/html; charset=ISO-  
8859-1"  
  
pageEncoding="ISO-8859-1"%>  
  
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  
  
<html>  
  
<head>  
  
<meta http-equiv="Content-Type" content="text/html;  
charset=ISO-8859-1">  
  
<meta content="width=device-width, initial-scale=1.0"  
name="viewport">  
  
<title><jsp:include page="projectTitle.jsp"></jsp:include></title>  
  
<meta content="" name="description">  
  
<meta content="" name="keywords">  
  
<!-- Favicons -->
```

```
<link href="assets/img/favicon.png" rel="icon">

<link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">

<!-- Google Fonts -->

<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Raleway:300,300i,400,400i,500,500i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">

<!-- Vendor CSS Files -->

<link href="assets/vendor/fontawesome-free/css/all.min.css" rel="stylesheet">

<link href="assets/vendor/animate.css/animate.min.css" rel="stylesheet">

<link href="assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

<link href="assets/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
```

```
    rel="stylesheet">

<link href="assets/vendor/boxicons/css/boxicons.min.css"
      rel="stylesheet">

<link href="assets/vendor/glightbox/css/glightbox.min.css"
      rel="stylesheet">

<link href="assets/vendor/remixicon/remixicon.css"
      rel="stylesheet">

<link href="assets/vendor/swiper/swiper-bundle.min.css"
      rel="stylesheet">

<!-- Template Main CSS File -->

<link href="assets/css/style.css" rel="stylesheet">

<!--
=====
-->

* Template Name: Medilab - v4.7.1
* Template URL: https://bootstrapmade.com/medilab-free-medical-bootstrap-theme/
* Author: BootstrapMade.com
```

\* License: <https://bootstrapmade.com/license/>

---

---

===== -->

</head>

<script>

```
function AllowAlphabet0() {  
    if      (!ureg.fname.value.match(/^[a-zA-Z]+$/))      &&  
ureg.fname.value != "") {  
  
    ureg.fname.value = "";  
  
    ureg.fname.focus();  
  
    alert("Please Enter only alphabet in text");  
  
}  
  
}
```

</script>

```
<script>

function AllowAlphabet1() {

    if      (!ureg.mname.value.match(/^[a-zA-Z]+$/))    &&
ureg.mname.value != "") {

    ureg.mname.value = "";

    ureg.mname.focus();

    alert("Please Enter only alphabet in text");

}

}

</script>
```

```
<script>

function AllowAlphabet2() {

    if      (!ureg.lname.value.match(/^[a-zA-Z]+$/))    &&
ureg.lname.value != "") {

    ureg.lname.value = "";

    ureg.lname.focus();

    alert("Please Enter only alphabet in text");
```

```
        }

    }

</script>

<script>

function validateEmail() {

    var email = document.ureg.email.value;

    if (email == 0) {

        alert("Enter Valid mail id");

        //document.ureg.email.focus();

        return false;

    }

    if (/^[\w+([.-]?\w+)*@\w+([.-]?\w+)*(\.\w{2,3})+$/.test(email)) {

        } else {

            alert("Enter valid mail id");

            // document.ureg.email.focus();
    }
}
```

```
return false;

}

}

</script>

<script type="text/javascript">

function ValidateMobNumber(txtMobId) {

    var fld = document.getElementById(txtMobId);

    if (fld.value == "") {

        alert("You didn't enter a phone number.");

        fld.value = "";

        // fld.focus();

        return false;

    } else if (isNaN(fld.value)) {

        alert("The phone number contains illegal

characters.");

        fld.value = "";
    }
}
```

```
//      fld.focus();

return false;

} else if (!(fld.value.length == 10)) {

    alert("The phone number is the wrong length. \nPlease
enter 10 digit mobile no.");

fld.value = "";

//      fld.focus();

return false;

}

}

</script>
```

```
<meta charset="utf-8">

<script language="Javascript">

function validation() {

    var email = document.ureg.email.value;
```

```
if (email == 0) {  
  
    alert("Enter Valid mail id");  
  
    document.ureg.email.focus();  
  
    return false;  
  
}  
  
if (/^w+([.-]?\w+)*@\w+([.-]  
]?w+)*(\.\w{2,3})+$/.test(email)) {  
  
} else {  
  
    alert("Enter valid mail id");  
  
    document.ureg.email.focus();  
  
    return false;  
  
}  
  
</script>
```

```
<script type="text/javascript">  
  
function ValidateMobNumber(txtMobId) {
```

```
var fld = document.getElementById(txtMobId);

if (fld.value == "") {

    alert("You didn't enter a phone number.");

    fld.value = "";

    // fld.focus();

    return false;

} else if (isNaN(fld.value)) {

    alert("The phone number contains illegal

characters.");

    fld.value = "";

    // fld.focus();

    return false;

} else if (!(fld.value.length == 10)) {

    alert("The phone number is the wrong length. \nPlease

enter 10 digit mobile no.");

    fld.value = "";

    // fld.focus();

    return false;
```

```
    }

}

function validatepassword() {

    var password = document.ureg.password.value;

    if (password != "") {

        if (password.length < 6) {

            //     document.ureg.password.focus();

            document.ureg.password = "";

            alert("Password must contain at least 6
characters!");

        }

        return false;
    }
}
```

```
</script>
```

```
<%-- <jsp:include page="validation.jsp"/> --%>
```

```
<body>
```

```
<!-- ===== Top Bar ===== -->
```

```
<div id="topbar" class="d-flex align-items-center fixed-top">
```

```
    <div class="container d-flex justify-content-between">
```

```
        <div class="contact-info d-flex align-items-center">
```

```
            <i class="bi bi-envelope"></i> <a
```

```
                href="mailto:contact@example.com">contact@healthcare.com</a>
```

```
>
```

```
            <i class="bi bi-phone"></i> +91 1234567890
```

```
        </div>
```

```
        <div class="d-none d-lg-flex social-links align-items-center">
```

```
<a href="#" class="twitter"><i class="bi bi-twitter"></i></a> <a href="#" class="facebook"><i class="bi bi-facebook"></i></a> <a href="#" class="instagram"><i class="bi bi-instagram"></i></a> <a href="#" class="linkedin"><i class="bi bi-linkedin"></i></a>

</div>

</div>

</div>

<!-- ===== Header ===== -->

<header id="header" class="fixed-top">

<div class="container d-flex align-items-center">

<h1 class="logo me-auto">

<a href="#"><jsp:include page="projectTitle.jsp"></jsp:include></a>

</h1>
```

<!-- Uncomment below if you prefer to use an image logo -->

```
id="why-us" class="why-us">

<div class="container">

    <div class="row">

        <!-- End .content-->

    </div>

</div>

</section> <!-- End Why Us Section --> <!-- ====== Appointment Section ====== -->

<section id="appointment" class="appointment section-bg">

    <div class="container">

        <div class="section-title">

            <h2>Patient Registration</h2>

        </div>

        <form      action="PatientRegistration"      method="post"
name="ureg">

            <div class="row">

                <div class="col">

                    <label for="fname">First Name</label>


```

```
<input type="text" id="fname"
placeholder="First Name" class="form-control" name="fname"
onkeyup="AllowAlphabet0()" required>

</div>

<div class="col">

    <label for="mname">Middle
Name</label>

    <input type="text" id="mname"
placeholder="Middle Name" class="form-control" name="mname"
onkeyup="AllowAlphabet1()" required>

</div>

<div class="col">

    <label for="lname">Last Name</label>

    <input type="text" id="lname"
placeholder="Last Name" class="form-control" name="lname"
onkeyup="AllowAlphabet2()" required>

</div>

</div>

<div class="form-row">
```

```
<div class="form-group">

    <label for="dob">Date of Birth</label>

    <input type="date" id="dob" class="form-control" name="dob" required>

</div>

<div class="form-group">

    <label>Gender</label>

    <div class="form-check">

        <input type="radio" id="male" name="gender" required value="Male">

        <label for="male">Male</label>

    </div>

    <div class="form-check">

        <input type="radio" id="female" name="gender" required value="Female">

        <label
            for="female">Female</label>

    </div>

</div>
```

```

</div>

<div class="form-row">
    <div class="form-group">
        <label for="mobile">Mobile Number</label>
        <input type="text" id="mobile" placeholder="Mobile Number" onblur="return ValidateMobNumber('txtMB')" pattern="\d*" maxLength="10" class="form-control" name="mobile" required>
    </div>
    <div class="form-group">
        <label for="adhar_id">Adhar Card Number</label>
        <input type="text" id="adhar_id" placeholder="Adhar Card Number" pattern="\d*" class="form-control" name="adhar_id" maxLength="12" required>
    </div>
</div>

```

```
<div class="form-group">

    <label for="address">Address in  
Details</label>

    <textarea id="address" rows="3" class="form-control" name="address" required placeholder="Address in Details  
with Pin Code"></textarea>

</div>

<div class="form-group">

    <label for="email">Email ID</label>

    <input type="email" id="email"  
placeholder="Your Email ID" class="form-control" name="email"  
required>

</div>

<div class="form-group">

    <label for="password">Password</label>

    <input type="password" id="password"  
placeholder="*****" class="form-control"  
name="password" required>
```

</div>

```
<!-- Buttons in One Row with Increased Gap -->

<div    class="form-row    d-flex    justify-content-
between">

    <div class="form-group col-md-5">

        <button type="submit" class="btn  btn-
success btn-block">Register</button>

    </div>

    <div class="form-group col-md-5">

        <!-- Add more margin here to increase the
space between the buttons -->
```



```
<div class="me-md-auto text-center text-md-start">

    <div class="copyright">
        © Copyright <strong><span><jsp:include
page="projectTitle.jsp"></jsp:include></span></strong>. All
        Rights Reserved
    </div>

    <div class="credits">
        <!-- All the links in the footer should remain
intact. -->
        <!-- You can delete the links only if you
purchased the pro version. -->
        <!-- Licensing information:
https://bootstrapmade.com/license/ -->
        <!-- Purchase the pro version with working
PHP/AJAX contact form: https://bootstrapmade.com/medilab-free-
medical-bootstrap-theme/ -->
    Designed by SGMS INFOTECH LLP
</div>
</div>
</div>
```

```
</footer>
```

```
<!-- End Footer -->
```

```
<div id="preloader"></div>
```

```
<a href="#"  
    class="back-to-top    d-flex    align-items-center    justify-  
    content-center"><i  
        class="bi bi-arrow-up-short"></i></a>
```

```
<!-- Vendor JS Files -->
```

```
<script  
src="assets/vendor/purecounter/purecounter.js"></script>  
  
<script  
src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>  
  
<script  
src="assets/vendor/glightbox/js/glightbox.min.js"></script>  
  
<script src="assets/vendor/swiper/swiper-  
bundle.min.js"></script>  
  
<script src="assets/vendor/php-email-  
form/validate.js"></script>
```

```
<!-- Template Main JS File -->

<script src="assets/js/main.js"></script>

</body>

</html>
```

Patient Login Code:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"

pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<meta content="width=device-width, initial-scale=1.0"
name="viewport">
```

```
<title><jsp:include page="projectTitle.jsp"></jsp:include></title>

<meta content="" name="description">

<meta content="" name="keywords">

<!-- Favicons -->

<link href="assets/img/favicon.png" rel="icon">

<link href="assets/img/apple-touch-icon.png" rel="apple-touch-icon">

<!-- Google Fonts -->

<link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Raleway:300,300i,400,400i,500,500i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">

<!-- Vendor CSS Files -->

<link href="assets/vendor/fontawesome-free/css/all.min.css"
```

```
rel="stylesheet">

<link href="assets/vendor/animate.css/animate.min.css"
rel="stylesheet">

<link href="assets/vendor/bootstrap/css/bootstrap.min.css"
rel="stylesheet">

<link href="assets/vendor/bootstrap-icons/bootstrap-icons.css"
rel="stylesheet">

<link href="assets/vendor/boxicons/css/boxicons.min.css"
rel="stylesheet">

<link href="assets/vendor/glightbox/css/glightbox.min.css"
rel="stylesheet">

<link href="assets/vendor/remixicon/remixicon.css"
rel="stylesheet">

<link href="assets/vendor/swiper/swiper-bundle.min.css"
rel="stylesheet">

<!-- Template Main CSS File -->

<link href="assets/css/style.css" rel="stylesheet">
```

```
<!--
=====
=====

* Template Name: Medilab - v4.7.1
* Template URL: https://bootstrapmade.com/medilab-free-
medical-bootstrap-theme/
* Author: BootstrapMade.com
* License: https://bootstrapmade.com/license/

=====
===== -->

</head>

<body>

<!-- ===== Top Bar ===== -->

<div id="topbar" class="d-flex align-items-center fixed-top">
    <div class="container d-flex justify-content-between">
        <div class="contact-info d-flex align-items-center">
```

```
<i      class="bi      bi-envelope"></i>      <a  
href="mailto:contact@example.com">contact@healthcare.com</a  
>  
  
<i class="bi bi-phone"></i> +91 1234567890  
  
</div>  
  
</div>  
  
<!-- ===== Header ===== -->  
  
<header id="header" class="fixed-top">  
  
<div class="container d-flex align-items-center">  
  
<h1 class="logo me-auto">  
  
<a href="#"><jsp:include  
page="projectTitle.jsp"></jsp:include></a>  
  
</h1>  
  
<!-- Uncomment below if you prefer to use an image logo -  
->  
  
<!-- <a href="index.jsp" class="logo me-auto"></a>-->
```

```
<nav id="navbar" class="navbar order-last order-lg-0">
<jsp:include
    page="header/mainMenu.jsp"></jsp:include> <i
    class="bi bi-list mobile-nav-toggle"></i> </nav>

<!-- .navbar -->

</div>

</header>

<!-- End Header -->

<!-- End Hero -->

<jsp:include page="header/alertBox.jsp"></jsp:include>

<main id="main"> <!-- ===== Why Us Section ===== -->
<section
    id="why-us" class="why-us">
    <div class="container">
```

```
<div class="row">

    <!-- End .content-->

</div>

</div>

</section> <!-- End Why Us Section --> <!-- ====== 
Appointment Section ====== -->

<section id="appointment" class="appointment section-bg">

<div class="container">

    <div class="row">

        <div class="col-lg-12">

            <div class="row">

                <div class="col-lg-6" align="right">

                </div>

                <div class="col-lg-6" align="center">

                    <h2>Patients Login</h2>

                    <hr />


```

```
<form action="PatientLogin"
      method="post">

      <table align="center">

          <tr>

              <th>Email ID</th>

              <td><input
type="text" placeholder="Your Email ID"
class="form-control" name="email" required></td>

          </tr>

          <tr>

              <th>Password</th>

              <td><input
type="password" placeholder="*****"
id="password" class="form-control" name="password"
required></td>

          </tr>
```

```
<th></th>
<td><button type="submit" class="btn btn-success">Login</button></td>
</tr>
</table>
</form>
<hr/>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- End Appointment Section -->
</main>
<!-- End #main -->
```

<!-- ===== Footer ===== -->

```
<footer id="footer">
```

```
<div class="container d-md-flex py-4">

    <div class="me-md-auto text-center text-md-start">

        <div class="copyright">
            © Copyright <strong><span><jsp:include
page="projectTitle.jsp"></jsp:include></span></strong>. All
            Rights Reserved
        </div>

        <div class="credits">
            <!-- All the links in the footer should remain
intact. -->
            <!-- You can delete the links only if you
purchased the pro version. -->
            <!-- Licensing information:
https://bootstrapmade.com/license/ -->
            <!-- Purchase the pro version with working
PHP/AJAX contact form: https://bootstrapmade.com/medilab-free-
medical-bootstrap-theme/ -->
        </div>
    </div>
</div>
```

Designed by SGMS INFOTECH LLP

```
</div>

</div>

</div>

</footer>

<!-- End Footer -->

<div id="preloader"></div>

<a href="#"  
    class="back-to-top d-flex align-items-center justify-  
content-center"><i  
        class="bi bi-arrow-up-short"></i></a>

<!-- Vendor JS Files -->

<script  
src="assets/vendor/purecounter/purecounter.js"></script>

<script  
src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<script  
src="assets/vendor/glightbox/js/glightbox.min.js"></script>
```

```
<script src="assets/vendor/swiper/swiper-bundle.min.js"></script>

<script src="assets/vendor/php-email-form/validate.js"></script>

<!-- Template Main JS File -->

<script src="assets/js/main.js"></script>

</body>

</html>
```

Patient Home Page Code:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>
```

```
<head>

<meta http-equiv="Content-Type" content="text/html;
charset=ISO-8859-1">

<meta content="width=device-width, initial-scale=1.0"
name="viewport">

<title><jsp:include page="projectTitle.jsp"></jsp:include></title>

<meta content="" name="description">

<meta content="" name="keywords">

<!-- Favicons -->

<link href="assets/img/favicon.png" rel="icon">

<link href="assets/img/apple-touch-icon.png" rel="apple-touch-
icon">

<!-- Google Fonts -->

<link
    href="https://fonts.googleapis.com/css?family=Open+Sans:300,
300i,400,400i,600,600i,700,700i|Raleway:300,300i,400,400i,500,5
00i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,60
0i,700,700i"
```

```
    rel="stylesheet">

<!-- Vendor CSS Files -->

<link href="assets/vendor/fontawesome-free/css/all.min.css"
      rel="stylesheet">

<link href="assets/vendor/animate.css/animate.min.css"
      rel="stylesheet">

<link href="assets/vendor/bootstrap/css/bootstrap.min.css"
      rel="stylesheet">

<link href="assets/vendor/bootstrap-icons/bootstrap-icons.css"
      rel="stylesheet">

<link href="assets/vendor/boxicons/css/boxicons.min.css"
      rel="stylesheet">

<link href="assets/vendor/lightbox/css/lightbox.min.css"
      rel="stylesheet">

<link href="assets/vendor/remixicon/remixicon.css"
      rel="stylesheet">

<link href="assets/vendor/swiper/swiper-bundle.min.css"
      rel="stylesheet">
```

```
<!-- Template Main CSS File -->

<link href="assets/css/style.css" rel="stylesheet">

<!--
=====
=====

* Template Name: Medilab - v4.7.1
* Template URL: https://bootstrapmade.com/medilab-free-medical-bootstrap-theme/
* Author: BootstrapMade.com
* License: https://bootstrapmade.com/license/

=====
===== -->

</head>

<body>

<!-- ===== Top Bar ===== -->
```

```
<div id="topbar" class="d-flex align-items-center fixed-top">

    <div class="container d-flex justify-content-between">

        <div class="contact-info d-flex align-items-center">

            <i class="bi bi-envelope"></i> <a href="mailto:contact@example.com">contact@healthcare.com</a>

        </div>

        <i class="bi bi-phone"></i> +91 1234567890

    </div>

</div>

<!-- ===== Header ===== -->

<header id="header" class="fixed-top">

    <div class="container d-flex align-items-center">

        <h1 class="logo me-auto">

            <a href="#"><jsp:include page="projectTitle.jsp"></jsp:include></a>

        </h1>

    </div>

</header>
```

```
<!-- Uncomment below if you prefer to use an image logo -->

<!-- <a href="index.jsp" class="logo me-auto"></a>-->

<nav id="navbar" class="navbar order-last order-lg-0">
<jsp:include
    page="header/mainMenu.jsp"></jsp:include> <i
    class="bi bi-list mobile-nav-toggle"></i> </nav>

<!-- .navbar -->

</div>

</header>

<!-- End Header -->

<!-- End Hero -->

<jsp:include page="header/alertBox.jsp"></jsp:include>
```

```

<main id="main"><!-- ===== Why Us Section ===== -->
<section
    id="why-us" class="why-us">

</section> <!-- End Why Us Section --> <!-- =====
Appointment Section ===== -->

<section id="appointment" class="appointment section-bg">
<div class="container">
    <div class="row">
        <div class="col-lg-12">
            <h2 align="center">Welcome Patients
Home</h2>

            <p align="right">
                Welcome:
                &nbsp;&nbsp;<%=session.getAttribute("fname").toString()%>&n
bsp;<%=session.getAttribute("mname").toString()%>&nbsp;<%=s
ession.getAttribute("lname").toString()%>

            <p align="center">
                </p>

```

```
</div>

</div>

</div>

</section> <!-- End Appointment Section --> </main>

<!-- End #main -->

<!-- ===== Footer ===== -->

<footer id="footer">

<div class="container d-md-flex py-4">

<div class="me-md-auto text-center text-md-start">

<div class="copyright">

    © Copyright <strong><span><jsp:include  
page="projectTitle.jsp"></jsp:include></span></strong>. All  
Rights Reserved

</div>

<div class="credits">
```

<!-- All the links in the footer should remain intact. -->

<!-- You can delete the links only if you purchased the pro version. -->

<!-- Licensing information:  
<https://bootstrapmade.com/license/> -->

<!-- Purchase the pro version with working PHP/AJAX contact form: <https://bootstrapmade.com/medilab-free-medical-bootstrap-theme/> -->

Designed by SGMS INFOTECH LLP

</div>

</div>

</div>

</footer>

<!-- End Footer -->

<div id="preloader"></div>

<a href="#"  
class="back-to-top d-flex align-items-center justify-content-center"><i

```
        class="bi bi-arrow-up-short">></i></a>

<!-- Vendor JS Files -->

<script
src="assets/vendor/purecounter/purecounter.js"></script>

<script
src="assets/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<script
src="assets/vendor/glightbox/js/glightbox.min.js"></script>

<script src="assets/vendor/swiper/swiper-
bundle.min.js"></script>

<script src="assets/vendor/php-email-
form/validate.js"></script>

<!-- Template Main JS File -->

<script src="assets/js/main.js"></script>

</body>

</html>
```

Check Patient Code:

```
<%@page import="java.sql.ResultSet"%>  
<%@page import="com.dbcon.ConnectionProvider"%>  
<%@page import="java.sql.Connection"%>  
<%@page import="java.sql.PreparedStatement"%>  
<%@page import="java.util.HashSet"%>  
<%@page import="com.code.global.GlobalFunction"%>  
<%@ page language="java" contentType="text/html; charset=ISO-  
8859-1"  
pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

Styles.css:

```
.rating {  
    unicode-bidi: bidi-override;  
    direction: rtl;  
    text-align: center;  
}
```

```
.rating > input {  
    display: none;  
}
```

```
.rating > label::before {  
    content: "\2605";  
    font-size: 36px;  
    color: grey;
```

```
}

.rating > label {
    display: inline-block;
    margin-right: 10px;
}

.rating > input:checked ~ label::before {
    color: gold;
}

.rating > input:checked ~ label:hover::before,
.rating > label:hover ~ input:checked ~ label::before {
    color: grey;
}
```

# **5. Testing**

## 5.1 Test Strategy:

### Introduction

The purpose of this test strategy is to define the testing approach, objectives, scope, tools, roles, and responsibilities for ensuring the quality of the Healthcare Management System, which includes Patient, Doctor, and Pathology Lab modules.

### Test Objectives

- Verify that all components and modules function as expected.
- Ensure the system meets business and user requirements.
- Detect defects early to reduce cost and risk.
- Validate performance, security, and usability of the application.

### Scope of Testing

#### In-Scope:

- Patient registration, login, appointment, prescription, payment.
- Doctor dashboard: appointments, prescriptions, test suggestions.
- Pathology lab: test management, report uploads.
- Security, access control, performance, and feedback module.

## Testing Types

Testing Type	Description
Unit Testing	Testing individual components such as form validation, DB access layers.
Integration Testing	Validate communication between modules: patient-doctor-lab, DB, APIs.

## **5.2 Unit Test Plan:**

- Verify the functionality of individual components and modules in isolation.
- Focus on testing business logic, data access layers, and utility classes.

### Integration Testing:

- Validate interactions between various modules and components.
- Test integration points such as APIs, database connections, and external service integrations.

### System Testing:

- Verify end-to-end functionality of the system as a whole.
- Test user flows, including registration, appointment scheduling, consultation sessions, and feedback submission.
- Conduct boundary testing, stress testing, and negative testing to ensure system robustness.

## **5.3 Acceptance Test Plan:**

- Validate that the system meets business requirements and user expectations.
- Involve stakeholders and end-users in acceptance testing to gather feedback and ensure alignment with expectations.
- Perform user acceptance testing (UAT) to validate usability, accessibility, and overall satisfaction.

## **5.4 Test Case/Test Script:**

### Patient Module Test Cases

#### 1.1 Registration

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_P_01	Register a new patient with valid details	Enter valid name, email, phone, address, password, and submit	Registration successful, redirected to login page
TC_P_02	Register with an already used email	Enter existing email and submit	Error: "Email already registered"
TC_P_03	Register without mandatory fields	Leave required fields empty and submit	Validation errors for missing fields

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_P_04	Register with invalid email format	Enter incorrect email format (e.g., abc.com)	Error: "Invalid email format"

## 1.2 Login

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_P_05	Login with correct credentials	Enter valid email & password	Redirect to Patient Dashboard
TC_P_06	Login with incorrect credentials	Enter wrong password	Error: "Invalid credentials"
TC_P_07	Login with unregistered email	Enter non-existent email	Error: "User not found"
TC_P_08	Login without entering details	Click login without filling fields	Validation errors displayed

### 1.3 Doctor Search & Appointment

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_P_09	Search doctor by specialization	Select specialization and search	Display list of doctors
TC_P_10	View doctor profile	Click on doctor profile	Show doctor details
TC_P_11	Book appointment with available slot	Select doctor, date, time, confirm booking	Appointment booked successfully
TC_P_12	Book appointment without time	Leave time slot empty	Error: "Please select a time slot"

## 1.4 Prescription & Lab

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_P_13	View doctor's prescription	Click on prescription section	Display prescription details
TC_P_14	Request lab test	Click "Request Test"	Request sent to pathology lab
TC_P_15	Download report after payment	Complete payment and download report	Report downloaded successfully

## 1.5 Payment

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_P_16	Complete payment successfully	Enter payment details & submit	Payment successful, confirmation received

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_P_17	Insufficient funds	Enter low balance card details	Error: "Payment failed due to insufficient balance"
TC_P_18	Cancel appointment with refund	Cancel paid appointment	Refund initiated

## 2. Doctor Module Test Cases

### 2.1 Registration & Login

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_D_01	Register with valid details	Enter valid details & submit	Registration successful, pending admin approval

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_D_02	Login as approved doctor	Enter valid credentials	Redirect to Doctor Dashboard

## 2.2 Manage Appointments & Prescriptions

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_D_03	View booked appointments	Click "Appointments"	List of patient bookings displayed
TC_D_04	Update appointment status	Change appointment status to "Completed"	Status updated successfully
TC_D_05	Add prescription	Enter prescription details & save	Prescription saved and visible to patient

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_D_06	Suggest pathology test	Select test & send request	Test request sent to pathology lab

### 3. Pathology Lab Module Test Cases

#### 3.1 Registration & Login

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_L_01	Register pathology lab	Enter valid lab details & submit	Registration successful, pending admin approval
TC_L_02	Login as approved lab	Enter credentials	Redirect to Lab Dashboard

### 3.2 Manage Lab Services

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_L_03	Add new test service	Enter test name, price, doctor, and save	Test service added successfully
TC_L_04	Update existing test service	Edit test details and save	Changes saved successfully

### 3.3 Reports

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_L_05	View pending test requests	Click "Test Requests"	Display pending requests
TC_L_06	Upload test report	Attach report file & submit	Report uploaded successfully, patient notified
TC_L_07	Prevent report upload without payment	Upload report for unpaid test	Error: "Payment pending"

## 4. Security & System Test Cases

### 4.1 Security

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_S_01	Prevent unauthorized access	Try accessing dashboard	Redirect to login page

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_S_02	Restrict role-based access	without login Login as patient, access doctor module	Error: "Unauthorized Access"
TC_S_03	SQL injection prevention	Enter SQL in login form	System blocks malicious input

## 4.2 Performance

<b>Test Case ID</b>	<b>Test Scenario</b>	<b>Test Steps</b>	<b>Expected Result</b>
TC_PF_01	Handle multiple concurrent bookings	Simulate 100+ users booking appointments	System remains stable and responsive

Test Case ID	Test Scenario	Test Steps	Expected Result
TC_PF_02	Doctor search under load	Search for doctor during high traffic	Results load within 2 seconds

## 5. Feedback & Rating Module

Test Case ID	Test Scenario	Test Steps	Expected Result
TC_FB_01	Provide feedback for doctor	Enter review & submit	Review saved successfully
TC_FB_02	Prevent blank feedback submission	Submit empty feedback field	Error: "Feedback cannot be empty"

# **6. Limitations of Proposed System**

Although the Doctor To Door Steps application enhances healthcare accessibility and efficiency, it has certain limitations that need to be considered.

#### 1. Internet Dependency

- The system requires a stable internet connection for users to access services, book appointments, and make payments.
- Patients in remote areas with poor network connectivity may face difficulties using the platform.

#### 2. Limited Personal Interaction

- Online consultations may not always be as effective as physical checkups, as doctors rely on verbal descriptions instead of physical examinations.
- Some complex medical conditions may require in-person diagnosis, which the system cannot provide.

#### 3. Data Security and Privacy Concerns

- Sensitive patient data, including medical records and payment information, needs strong encryption and security measures.
- Any security breach could lead to unauthorized access to personal health information.

#### 4. Accuracy of Diagnoses

- Virtual consultations may lead to misdiagnosis if patients fail to provide accurate symptoms or medical history.
- The system does not have AI-based automated diagnosis capabilities, relying entirely on human expertise.

#### 5. Dependence on User Input

- The system depends on patients, doctors, and pathology labs to enter and update data correctly.
- Any incorrect or missing information could lead to confusion and errors in appointments, prescriptions, or test results.

# **7. Proposed Enhancements**

To improve the efficiency, security, and user experience of the Doctor To Door Steps system, the following enhancements are proposed:

### 1. AI-Based Symptom Analysis

- Integrate an AI-powered symptom checker to assist patients in determining possible health conditions before booking a doctor.
- Provide recommendations for specialized doctors based on symptoms.

### 2. Telemedicine Support (Video Consultation)

- Enable video consultation for better diagnosis instead of relying only on text-based prescriptions.
- Allow doctors to interact with patients more effectively, improving diagnostic accuracy.

### 3. Offline Mode for Appointment Scheduling

- Introduce an offline mode where patients can book appointments via SMS or call in case of poor internet connectivity.
- Synchronize data automatically when the user goes online.

# **8. Conclusion**

The Doctor To Door Steps application is designed to provide a seamless and efficient online healthcare solution, enabling patients to consult doctors, book pathology tests, and access medical reports from the comfort of their homes. The system enhances the accessibility of healthcare services, reducing the need for physical visits and long waiting times.

Through its Patient, Doctor, Pathology Lab, and Admin modules, the platform streamlines the process of appointment scheduling, prescription management, pathology testing, and secure online payments. By leveraging technology-driven solutions, it ensures better healthcare management and enhances user convenience.

Despite some limitations, such as reliance on internet connectivity, lack of emergency support, and potential security concerns, proposed enhancements like AI-based symptom analysis, video consultations, blockchain-based security, and health monitoring integrations can significantly improve its efficiency and reliability.

In conclusion, the Doctor To Door Steps application is a promising step towards digital healthcare transformation, offering an innovative, patient-centric approach. Future improvements will further enhance its functionality, making it an even more robust, secure, and accessible healthcare platform.

# **9. Bibliography**

1. Lundy, K. S., & Janes, S. (Eds.). (2019). Community Health Nursing: Caring for the Public's Health. Jones & Bartlett Learning.
2. Mahan Buttaro, T., Trybulski, J., Polgar-Bailey, P., & Sandberg-Cook, J. (Eds.). (2016). Primary Care: A Collaborative Practice. Elsevier Health Sciences.
3. Marrelli, T. M. (2019). Home Care Nursing: Surviving in an Ever-Changing Care Environment. Mosby.

# **10. User Manual (All**

**screens with proper**

**description/purpose**

**Details About**

**validations related to**

**data to be entered.)**

## 1. Introduction

The Doctor To Door Steps application is an online healthcare platform that enables patients to consult doctors, book appointments, request pathology tests, and access medical reports from home. This manual explains how different users (Patients, Doctors, and Pathology Labs) can use the system.

## 2. User Roles and Access

The system has three main user roles:

- Patients – Register, search for doctors, book appointments, make payments, and access medical reports.
- Doctors – Manage appointments, provide prescriptions, and recommend pathology tests.
- Pathology Labs – Offer test services, upload reports, and manage patient requests.

## 3. System Requirements

Hardware Requirements:

- PC, Laptop, Tablet, or Smartphone
- Stable Internet Connection

Software Requirements:

- Web Browser (Google Chrome, Firefox, Safari, Edge)
- Windows/Linux/MacOS/Android/iOS

- Internet Access

## 4. How to Use the Application

### 4.1 Patient Module

#### Step 1: Registration & Login

1. Open the application and click "Register".
2. Enter personal details (name, email, mobile number, address, etc.).
3. Create a username and password.
4. Click "Sign Up" and verify the account via OTP.
5. Log in using your credentials.

#### Step 2: Search for a Doctor

1. Enter search criteria such as specialization (Dentist, Cardiologist, etc.) or doctor qualification (MBBS, MD, etc.).
2. Click "Search" to view doctor profiles.
3. Click on a doctor's profile to view their experience, consultation fees, and reviews.

#### Step 3: Book an Appointment

1. Choose a date and available time slot.
2. Confirm the appointment by clicking "Book Now".
3. Receive booking confirmation via email/SMS.

#### Step 4: View & Manage Appointments

1. Navigate to "My Appointments".
2. View scheduled appointments and their status.

3. Click "Cancel" if you want to cancel an appointment.

#### Step 5: View Prescriptions & Request Pathology Tests

1. After consultation, go to "My Prescriptions" to check the doctor's recommendations.
2. If a pathology test is recommended, click "Request Test" to proceed.

#### Step 6: Make Online Payments

1. If a pathology lab requires payment, go to "Payments".
2. Choose payment method (Cash, UPI).
3. Complete the transaction to download the test report.

#### Step 7: Provide Feedback & Ratings

1. Go to "Feedback" section.
2. Rate the doctor or lab and provide written feedback.

#### Step 8: Logout

- Click "Logout" to exit securely.

## 4.2 Doctor Module

#### Step 1: Registration & Login

1. Click "Doctor Register" and enter details like name, qualifications, specialization, and experience.
2. Submit documents for verification.
3. After admin approval, log in using credentials.

## **Step 2: Manage Appointments**

1. Click "Appointments" to view patient bookings.
2. Accept, reschedule, or cancel appointments.

## **Step 3: Provide Prescriptions & Recommend Tests**

1. Open "Patient Details" to check symptoms.
2. Enter prescription details and suggest medications.
3. If needed, recommend a pathology test.

## **Step 4: View Patient History & Payments**

1. Check previous consultations and prescriptions for returning patients.
2. View payment status of appointments.

## **Step 5: Update Profile & Security**

1. Edit personal details in "Profile Settings".
2. Change password if required.

## **Step 6: View Feedback**

- Read patient reviews to improve service quality.

## **Step 7: Logout**

- Click "Logout" to exit the system.

## 4.3 Pathology Lab Module

### Step 1: Registration & Login

1. Click "Lab Register" and enter lab name, location, and services offered.
2. Log in after approval.

### Step 2: Add & Manage Test Services

1. Go to "Services" and click "Add New Test".
2. Enter details (test name, cost, doctor's name, etc.).
3. Save and update anytime.

### Step 3: View Appointment Requests & Payments

1. Open "Bookings" to see patient test requests.
2. Accept or reject based on availability.
3. Ensure payment is received before report upload.

### Step 4: Upload Test Reports

1. Click "Upload Report" under patient's request.
2. Attach a PDF or image of the report.
3. Confirm upload and notify the patient.

### Step 5: View Feedback & Ratings

- Read patient reviews to improve services.

### Step 6: Logout

- Click "Logout" to exit the system securely.

## 4. Troubleshooting & FAQs

Q1: I forgot my password. How do I reset it?

- Click "Forgot Password" on the login page.
- Enter your registered email/phone number.
- Follow the reset instructions sent via email/SMS.

Q2: My appointment payment failed. What should I do?

- Check if the amount was deducted from your account.
- If deducted, wait for 5–10 minutes; the system will retry confirmation.
- If not resolved, contact support team.

Q3: How do I cancel a booked appointment?

- Go to "My Appointments" and click "Cancel" next to the appointment.

Q4: Can I reschedule my appointment?

- Yes, open "My Appointments", select a new time slot, and click "Reschedule".

Q5: Can I download my pathology report anytime?

- Yes, once payment is completed, go to "My Reports" and click "Download".

Q6: What if my doctor doesn't provide a prescription?

- Contact customer support or send a direct message to the doctor from "My Appointments" section.

Q7: How do I change my registered phone number or email?

- Open "Profile Settings", edit your contact details, and save the changes.

## 6. Customer Support

For any issues, contact the support team via:

- Email: support@doctortodoor.com
- Helpline Number: +91-XXXXXXXXXX