**HOSPITAL MANAGEMENT SYSTEM**

**A PROJECT REPORT**

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**Dangar Rajdeep (2020033800100996) Patel Unique (2020033800100393)**

**Damor Dipak (2020033800101005) Rathva Yuvraj (2020033800111197)**

**Under the Esteemed Guidance of**

**Dr.Mamta Padole ,**

**Associate Professor**

**Department of Computer Science and Engineering**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

Faculty of Technology and Engineering

❖ **TECHNOLOGIES USED:**

• Fronted:HTML ,CSS,Javascript,Bootstsrap

• Backend:Servlet,JDBC

• Database:MySQL

• Server:Apache Tomcat

• Tools:Eclipse,MySQL Workbench

❖ **DESCRIPTION:**

➢ Hospital Management System aims to automate and manage the various operations of

a hospital and reduces the work of documentation.

➢ Key features of this project :

• Patient Management: This feature allows healthcare facilities to manage

patient information such as personal details, medical history, diagnosis,

treatment, and medication.

• Appointment Scheduling: This feature enables healthcare providers to schedule

appointments with patients, track their attendance, and reschedule

appointments if necessary.

• Staff Management: This feature enables healthcare facilities to manage

employee information and schedules .

• Based on the disease type system allocates specialist doctor to a patient.

• Keep records of Doctors in the hospital.

• Improves interaction between the patient and the Hospital.

• Doctor can easily make the reports of the patient by retrieving the data of the

Patient .

• Keep track of the appointment dates and status.

➢ Hospital Management System makes it easy to get access to the management system

facilities for the authorized users and keep it safe from unauthorized users.

**INDEX**

1. INTRODUCTION

1.1 Introduction

1.2 Modules in the project

1.2.1 Admin Module

1.2.2 User Module

1.2.3 Doctor Module

1.2.4 Appointment Module

2.Requirement specification

2.1Introduction

2.1.1 Software Requirements for Present Project

2.1.2 Functional Requirements

2.1.3 Software Specification

2.2 Project planning and scheduling

2.2.1 methodology

2.2.2 Project Management Life Cycle

3.Design

3.1 SYSTEM DESIGN:

3.1.1 INTRODUCTION TO UML:

3.2 UML Approach

3.2.1 Use Case Diagram

3.2.2 ER Diagram

4.System Implementation

5. Sample Screenshots

6.Testing

6.1 Introduction to System Testing:

7.Conclusion

1.**INTRODUCTION**

1.1 Introduction

1.2 Modules in the project

**1.1 Introduction:**

The project Hospital Management system includes registration of patients, storing their details into the system. The software has the facility to give a unique id for every patient and stores the details of every patient and the doctor automatically. User can easily see their appointment status also.

The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or receptionist. Only they can add data into the database. The data can be retrieved easily. The interface is very user-friendly. The data are well protected for personal use and makes the data processing very fast.

Hospital Management System is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals.

Hospital Management System is designed for multispeciality hospitals, to cover a wide range of hospital administration and management processes. It is an integrated end-to-end Hospital Management System that provides relevant information across the hospital to support effective decision making for patient care, hospital administration.

Hospital Management System is a software product suite designed to improve the quality and management of hospital management in the areas of clinical process analysis and activity-based costing. Hospital Management System enables you to develop your organization and improve its effectiveness and quality of work. Managing the key processes efficiently is critical to the success of the hospital helps you manage your processes.

**1.2 MODULES:**

The entire project mainly consists of 4 modules, which are

* Admin module
* User module (patient)
* Doctor module
* Appointment

**1.2.1 Admin module:** Admin has the full access to the system which means he is able to manage any activity with regard to the system. He is the highest privileged user who can access to the system.

**Key functions:**

* manage department of hospitals, user, doctor, total appointment and specialist.
* We can add new doctor,removing doctor,also we can editing the doctor detailed.
* Watch total no of doctor ,user,total appointment,total no of specialist which we have add.
* Watch the doctor detailed in view doctor.
* We can also see inside patient bar patient detailed and patient medical status.
* Logout from admin module also.

**1.2.2 User module:**

**Key Functions:**

* Using email and password user can login into user module.
* New user can also register.
* User can easily add their appointment and also see their appointment.
* User can easily change their password
* Also user can easily logout from user Module.
* User can Easily see Their Appointment.

**1.2.3 Doctor module:** Doctors can view the patient appointment list and provide the confirmation or make changes in the appointment list if required. Doctors have access to only records of those patients whom they are treating.Using email and password Doctor can login into Doctor module.

**Key Functions:**

* Doctor can see total number of appointment of patient.
* Doctor can also change the status of the patient.
* Doctor can modified their profile detailed.
* Doctor can also change their password.

**1.2.4 Appointment module:**

**Key functions:**

* Using this module user can add their appointment.
* If user hasn’t login in user module then appoint can’t be added.

**2 .Requirement Specification**

**2.1 INTRODUCTION:**

To be used efficiently, all computer software needs certain hardware components or the other software resources to be present on a computer.

**2.1.1 Software Requirements for Present Project**

• Fronted:HTML ,CSS,Javascript,Bootstsrap • Server:Apache Tomcat

• Backend:Servlet,JDBC • Database:MySQL

• Tools:Eclipse,MySQL Workbench

**2.1.2 Functional Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No.** | **Module Name** | **Applicable Roles** | **Description** |
| **1** | Admin | Admin | ADMIN:Admin can manage doctor,add specialist,view appointment of patient ,view number of registered user,login and logout.Admin can add doctor details and provide id and password to the doctor for first time login. |
| **2** | Doctor | Doctor | DOCTOR : Can view appointment , update status of Patient ,login and logout.Doctor can also update profile and password. |
| **3** | User | User | PATIENT: Can login using unique Id and Password after this system shall show his/her profile.Patient can view or make appointment. |
| **4** | Appointment | User  Doctor | PATIENT: Can Select doctor, date time and make an appointment .  DOCTOR : Can view appointment and update the status of Patient. |

**2.1.3 Software Specification:**

**HTML:**

**HTML** or **Hypertext Markup Language** is the standard [markup language](http://en.wikipedia.org/wiki/Markup_language) used to create [web pages](http://en.wikipedia.org/wiki/Web_page).

HTML is written in the form of [HTML elements](http://en.wikipedia.org/wiki/HTML_element) consisting of *tags* enclosed in [angle brackets](http://en.wikipedia.org/wiki/Angle_brackets) (like <html>). HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent *empty elements* and so are unpaired, for example <img>. The first tag in a pair is the *start tag*, and the second tag is the *end tag* (they are also called *opening tags* and *closing tags*). Though not always necessary, it is best practice to append a slash to tags which are not paired with a closing tag.

The purpose of a [web browser](http://en.wikipedia.org/wiki/Web_browser) is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website [semantically](http://en.wikipedia.org/wiki/Semantic) along with cues for presentation, making it a [markup language](http://en.wikipedia.org/wiki/Markup_language) rather than a [programming language](http://en.wikipedia.org/wiki/Programming_language).

HTML elements form the building blocks of all [websites](http://en.wikipedia.org/wiki/Website). HTML allows [images and objects](http://en.wikipedia.org/wiki/Img_(HTML_element)) to be embedded and can be used to create [interactive forms](http://en.wikipedia.org/wiki/Fieldset). It provides a means to create [structured documents](http://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](http://en.wikipedia.org/wiki/Semantic) for text such as headings, paragraphs, lists, [links](http://en.wikipedia.org/wiki/Hyperlink), quotes and other items. It can embed [scripts](http://en.wikipedia.org/wiki/Scripting_language) written in languages such as [JavaScript](http://en.wikipedia.org/wiki/JavaScript) which affect the behavior of HTML web pages.

**CASCADING STYLE SHEETS** (**CSS**):

It is a [style sheet language](http://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [look and formatting](http://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](http://en.wikipedia.org/wiki/Markup_language). While most often used to style [web pages](http://en.wikipedia.org/wiki/Web_page) and [interfaces](http://en.wikipedia.org/wiki/Interface_(computing)) written in [HTML](http://en.wikipedia.org/wiki/HTML) and [XHTML](http://en.wikipedia.org/wiki/XHTML), the language can be applied to any kind of [XML](http://en.wikipedia.org/wiki/XML) document, including [plain XML](http://en.wikipedia.org/wiki/Plain_Old_XML), [SVG](http://en.wikipedia.org/wiki/Scalable_Vector_Graphics) and [XUL](http://en.wikipedia.org/wiki/XUL). CSS is a cornerstone specification of [the web](http://en.wikipedia.org/wiki/The_web) and almost all web pages use CSS style sheets to describe their presentation.

CSS is designed primarily to enable the separation of document content from document presentation, including elements such as the [layout](http://en.wikipedia.org/wiki/Page_layout), [colors](http://en.wikipedia.org/wiki/Color), and [fonts](http://en.wikipedia.org/wiki/Typeface).[[1]](http://en.wikipedia.org/wiki/Cascading_Style_Sheets#cite_note-1) This separation can improve content [accessibility](http://en.wikipedia.org/wiki/Accessibility), provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content .

CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or [screen reader](http://en.wikipedia.org/wiki/Screen_reader)) and on [Braille-based](http://en.wikipedia.org/wiki/Braille_display), tactile devices. It can also be used to allow the web page to display differently depending on the screen size or device on which it is being viewed. While the author of a document typically links that document to a CSS file, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. However if the author or the reader did not link the document to a specific style sheet the default style of the browser will be applied.

**MySQL:**

MySQL is developed, distributed, and supported by Oracle Corporation. MySQL is a database system used on the web it runs on a server. MySQL is ideal for both small and large applications. It is very fast, reliable, and easy to use. It supports standard SQL. MySQL can be compiled on a number of platforms.

The data in MySQL is stored in tables. A table is a collection of related data, and it consists of columns and rows. Databases are useful when storing information categorically.

**FEATURES OF MySQL:**

Internals and portability:

* Written in C and C++.
* Tested with a broad range of different compilers.
* Works on many different platforms.
* Tested with Purify (a commercial memory leakage detector) as well as with Val grind, a GPL tool.
* Uses multi-layered server design with independent modules.

#### Security:

* A privilege and password system that is very flexible and secure, and that enables host-based verification.
* Password security by encryption of all password traffic when you connect to a server.

#### Scalability and Limits:

* Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.
* Support for up to 64 indexes per table (32 before MySQL 4.1.2). Each index may consist of 1 to 16 columns or parts of columns. The maximum index width is 767 bytes for **InnoDB** tables, or 1000 for **MyISAM**; before MySQL 4.1.2, the limit is 500 bytes. An index may use a prefix of a column for [**CHAR**](https://dev.mysql.com/doc/refman/5.0/en/char.html), [**VARCHAR**](https://dev.mysql.com/doc/refman/5.0/en/char.html), [**BLOB**](https://dev.mysql.com/doc/refman/5.0/en/blob.html), or [**TEXT**](https://dev.mysql.com/doc/refman/5.0/en/blob.html) column types.

**Jsp:**

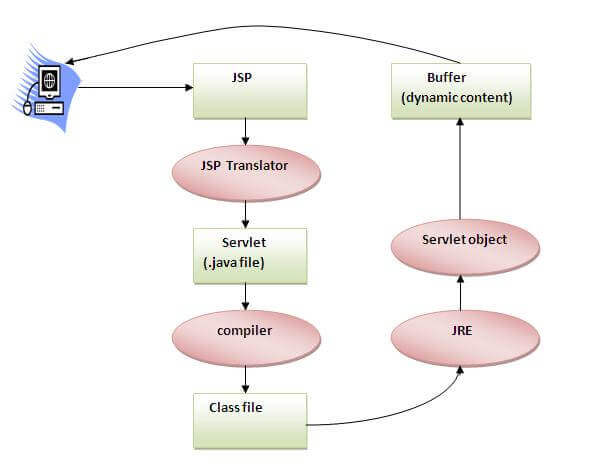
**JSP** technology is used to create web application just like Servlet technology. It can be thought of as an extension to Servlet because it provides more functionality than servlet such as expression language, JSTL, etc.

A JSP page consists of HTML tags and JSP tags. The JSP pages are easier to maintain than Servlet because we can separate designing and development. It provides some additional features such as Expression Language, Custom Tags, etc.

### **The Lifecycle of a JSP Page**

The JSP pages follow these phases:

* Translation of JSP Page
* Compilation of JSP Page
* Classloading (the classloader loads class file)
* Instantiation (Object of the Generated Servlet is created).
* Initialization ( the container invokes jspInit() method).
* Request processing ( the container invokes \_jspService() method).
* Destroy ( the container invokes jspDestroy() method).

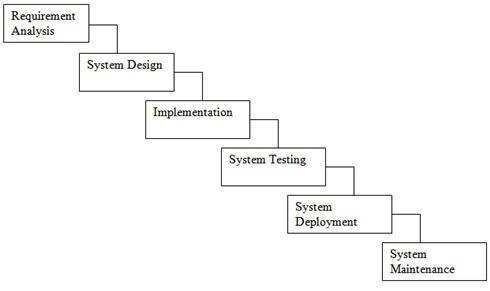


#### **2.2 Project planning and scheduling :**

Project planning is part of project management, which relates to the use of schedules such as Gantt charts to plan and subsequently report progress within the project environment. Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure. The logical dependencies between tasks are defined using an activity network diagram that enables identification of the critical path.

##### **2.2.1 Methodology**

We have used Iterative and Incremental Development model (IID) for our project development. This development approach is also referred to as Iterative Waterfall Development approach. Iterative and Incremental Development is a software development process developed in response to the more traditional waterfall model. This model is designed to take care of such big project. The large and complicate project chiefly demand better development and testing procedure. The waterfall model is well known for its repeated testing process. Hence I choose the waterfall model for developing my software.



**Fig. 2.1:** Waterfall model

Some advantages of waterfall model:

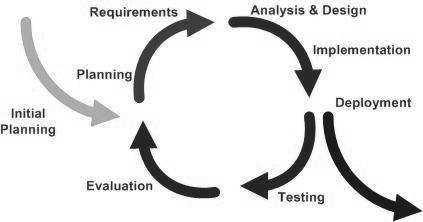
o Simple and easy to understand and use. o Easy to manage due to the rigidity of the model. o Phases are processed and completed one at a time o Works well for smaller projects where requirements are very well understood.

##### **2.2.2 Project Management Life Cycle**

The Project Management Life Cycle has four phases. Each project life cycle phase is described along with the tasks need to complete it

The four phases is

1. Initiation
2. Planning
3. Execution
4. Closure.



**Fig. 2.2:** Iterative and Incremental Life Cycle

**3.DESIGN**

**3.1 SYSTEM DESIGN:**

**3.1.1 INTRODUCTION TO UML:**

UML Design

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the software system and its components. It is a graphical language , which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

The UML is a language for:

\*Visualizing

\*Specifying

\*Constructing

\*Documenting

**Visualizing**

Through UML we see or visualize an existing system and ultimately we

visualize how the system is going to be after implementation. Unless we think,

we cannot implement. UML helps to visualize, how the components of the

system communicate and interact with each other.

**Specifying**

Specifying means building, models that are precise, unambiguous and complete

UML addresses the specification of all the important analysis design,

implementation decisions that must be made in developing and deploying a

software system.

**Constructing**

UML models can be directly connected to a variety of programming language

through mapping a model from UML to a programming language like JAVA or

C++ or VB. Forward Engineering and Reverse Engineering is possible through

UML.

**Documenting**

The Deliverables of a project apart from coding are some Artifacts, which are

critical in controlling, measuring and communicating about a system during its

developing requirements, architecture, desire, source code, project plans, tests,

prototypes releasers, etc..

**3.2 UML Approach**

UML Diagram

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices and arcs . you draw diagram to visualize a system from different perspective, so a diagram is a projection into a system. For all but most trivial systems, a diagram represents an elided view of the elements that make up a system. The same element may appear in all diagrams, only a few diagrams , or in no diagrams at all. In theory, a diagram may contain any combination of things and relationships. In practice, however, a small number of common combinations arise, which are consistent with the five most useful views that comprise the architecture of a software-intensive system. For this reason, the UML includes nine such diagrams:

1. Class diagram

2. Object diagram

3. Use case diagram

4. Sequence diagram

5. Collaboration diagram

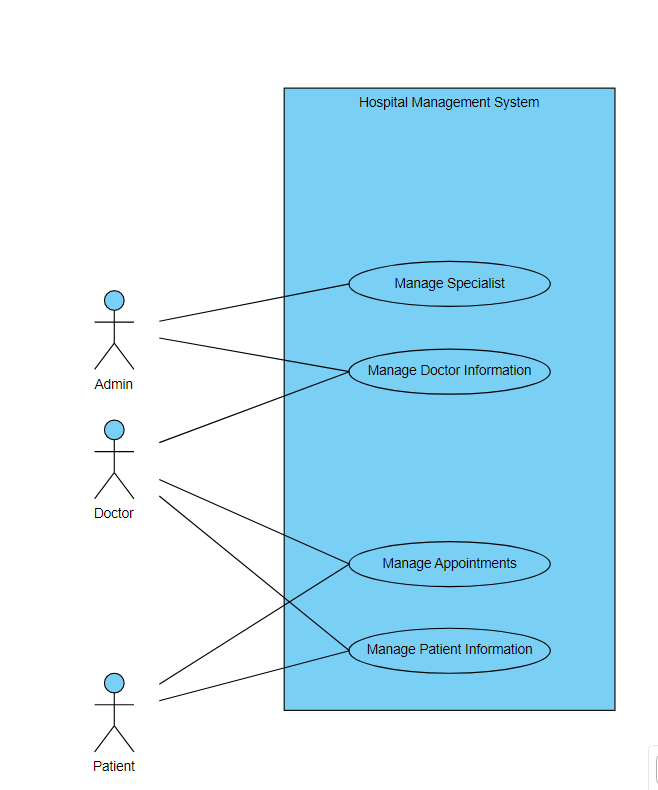
6. State chart diagram

7. Activity diagram

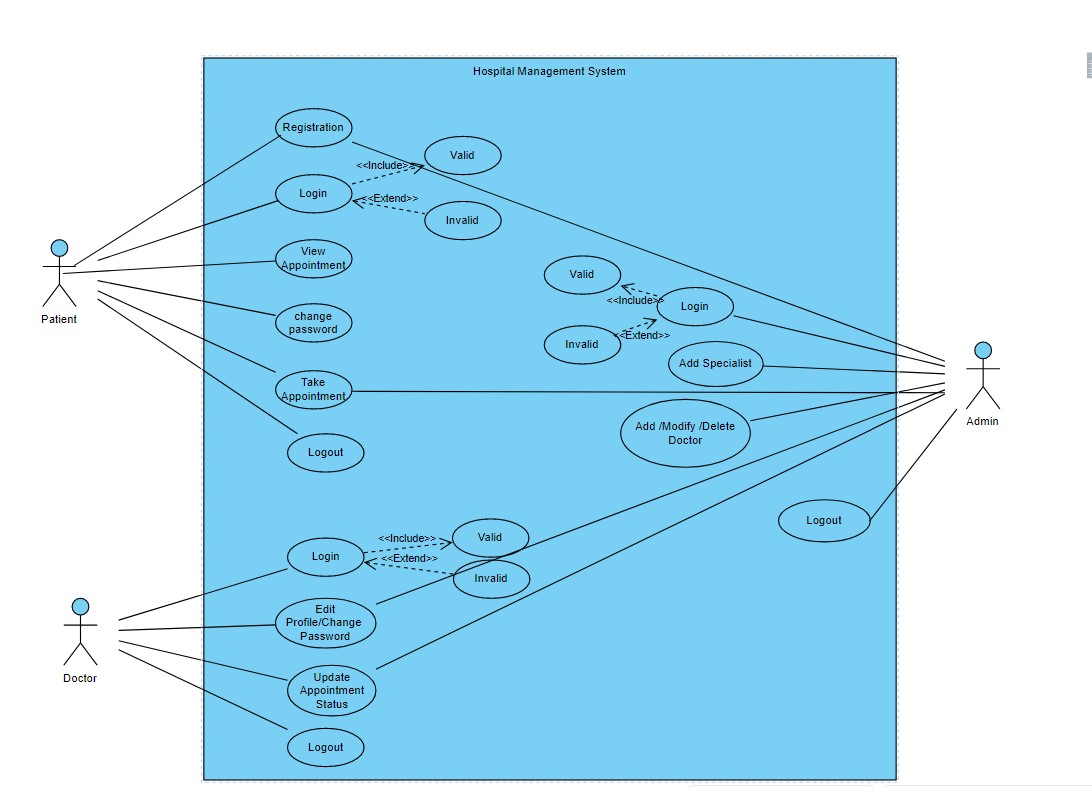
8. Component diagram

9. Deployment diagram

**3.2.1 USE CASE DIAGRAM:**

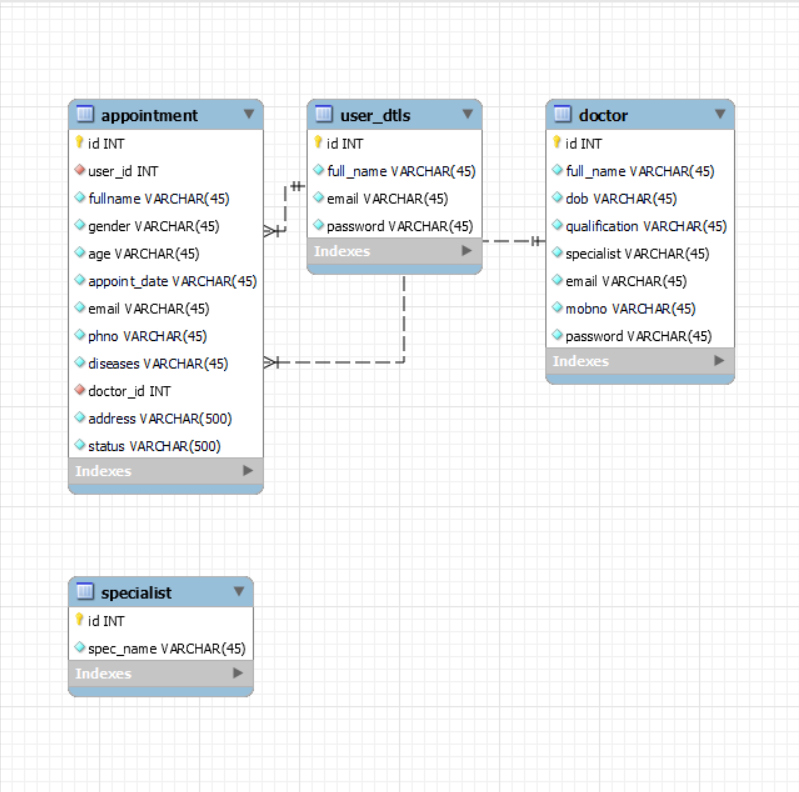
A usecase diagram in the Unified Modeling Language(UML) is atype of behavioral diagram defined by and created from a use-case analysis.its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals(represented as use cases),and any dependencies between those use cases. Use case diagrams are formally included in two modeling languages defined by the OMG:theunfied modeling language(UML) and the systems modeling language(sysML). 

*Use Case Diagram*



**3.2.2 ER Diagram**

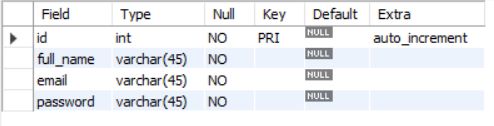
Database is absolutely an integral part of software system. To fully utilize ER Diagram in database engineering guarantee you to produce high quality database design to use in database creation, management and maintenance. An ER model also provides a means for communication.



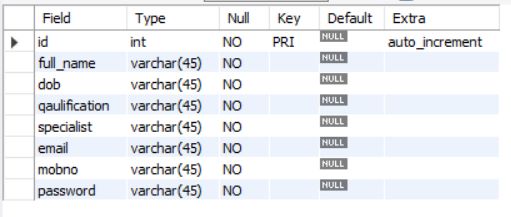
*EER Diagram (Hospital Management System)*

## Database Table:

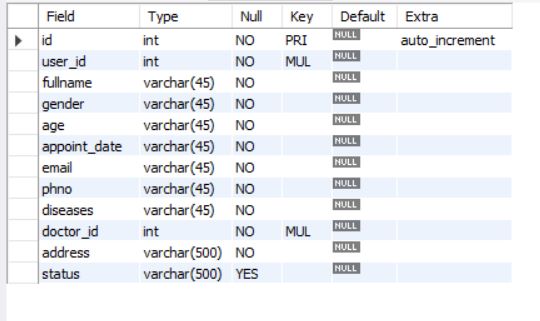
**User\_dtls table:**



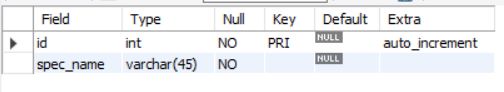
**Doctor table:**



**Appointment Table:**



**Specialist Table:**



**4.SYSTEM IMPLEMENTATION**

**Inroduction:**

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Source Code:

**GitHub Link:** [**https://github.com/dipakdamor417/Hospital\_management**](https://github.com/dipakdamor417/Hospital_management)

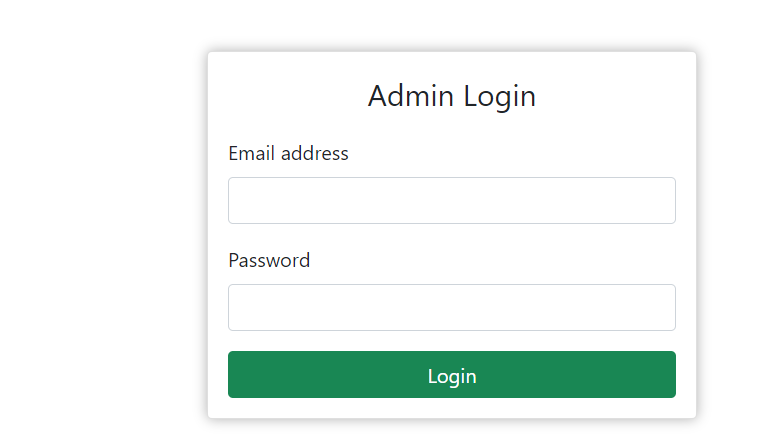
## 5.SAMPLE SCREENSHOTS

**5.1 Home page:**



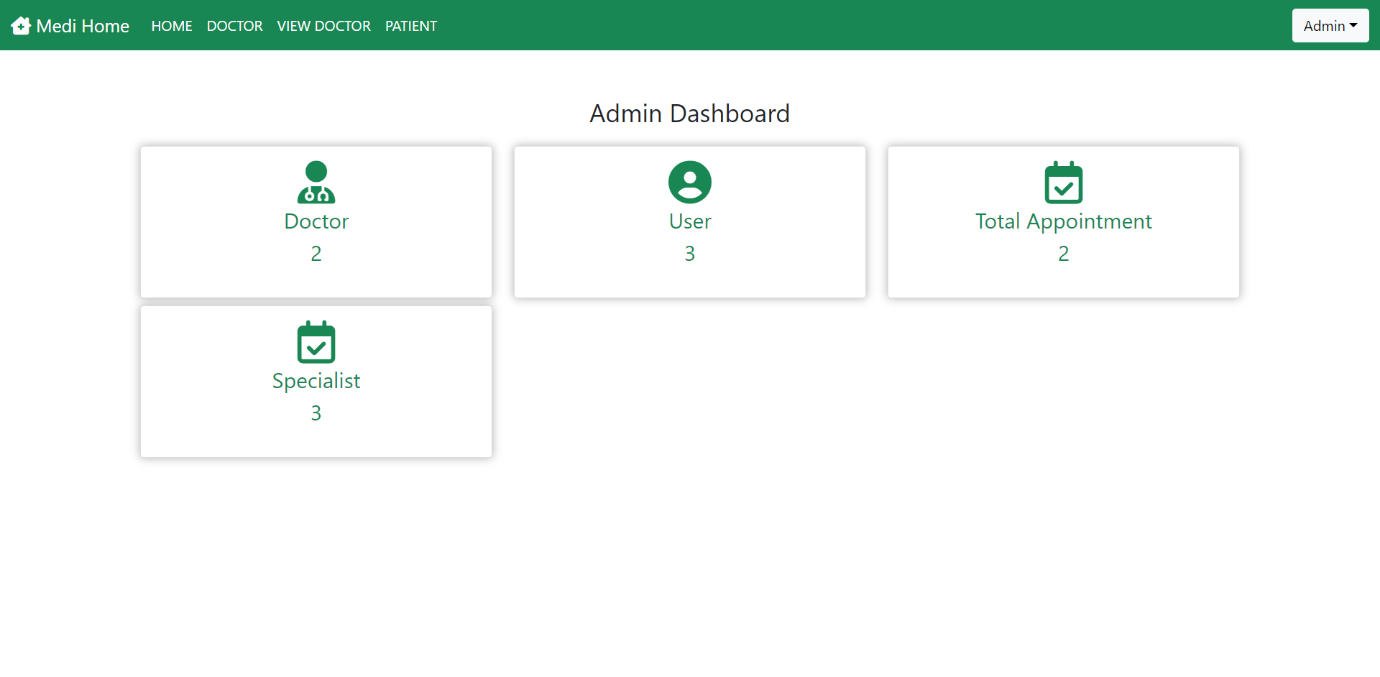
**5.2 Admin**

Admin login:

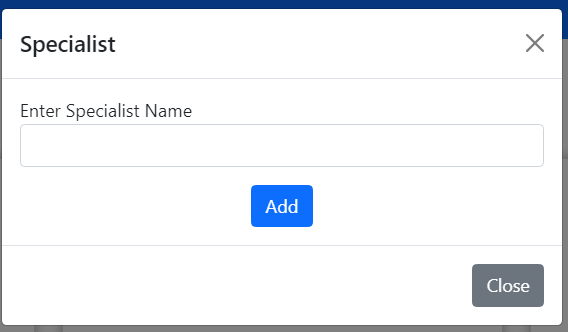


User Registration: User Login:

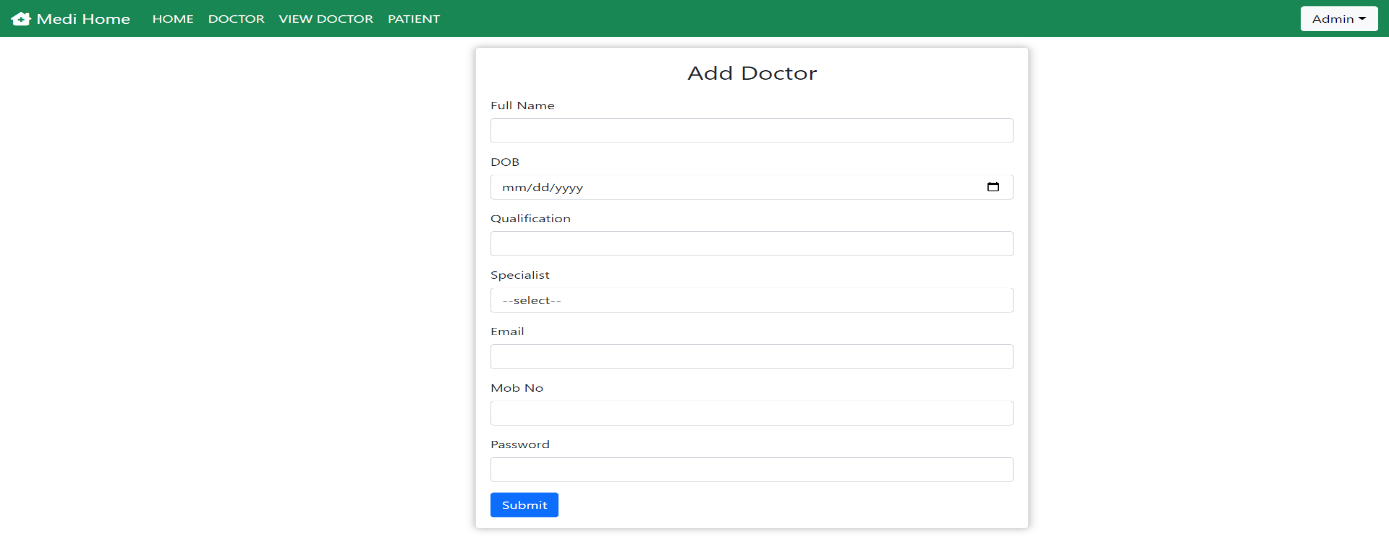
Admin Dashboard:



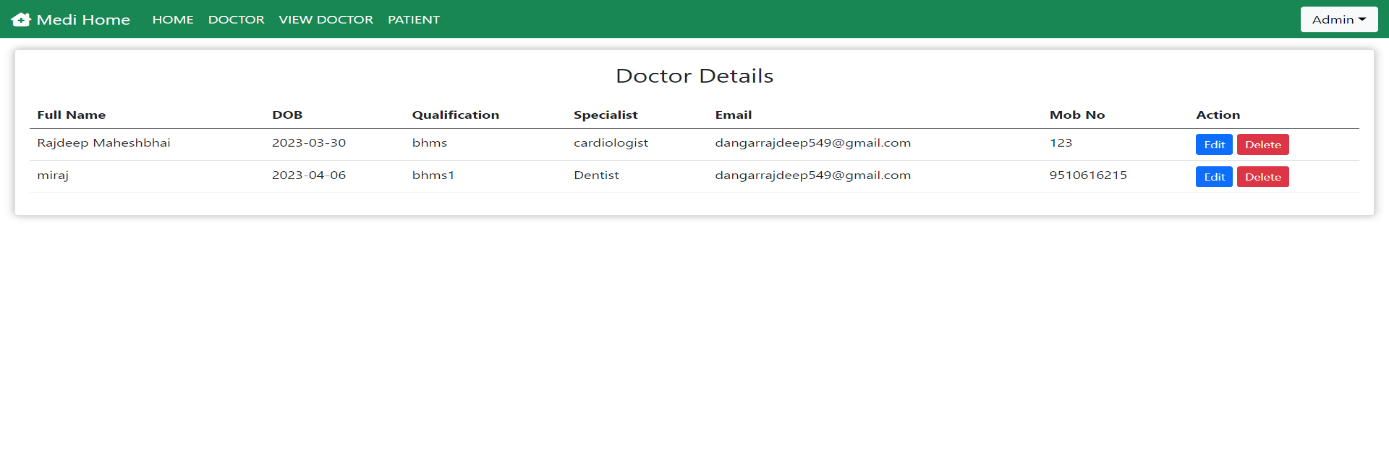
Adding Specialist:



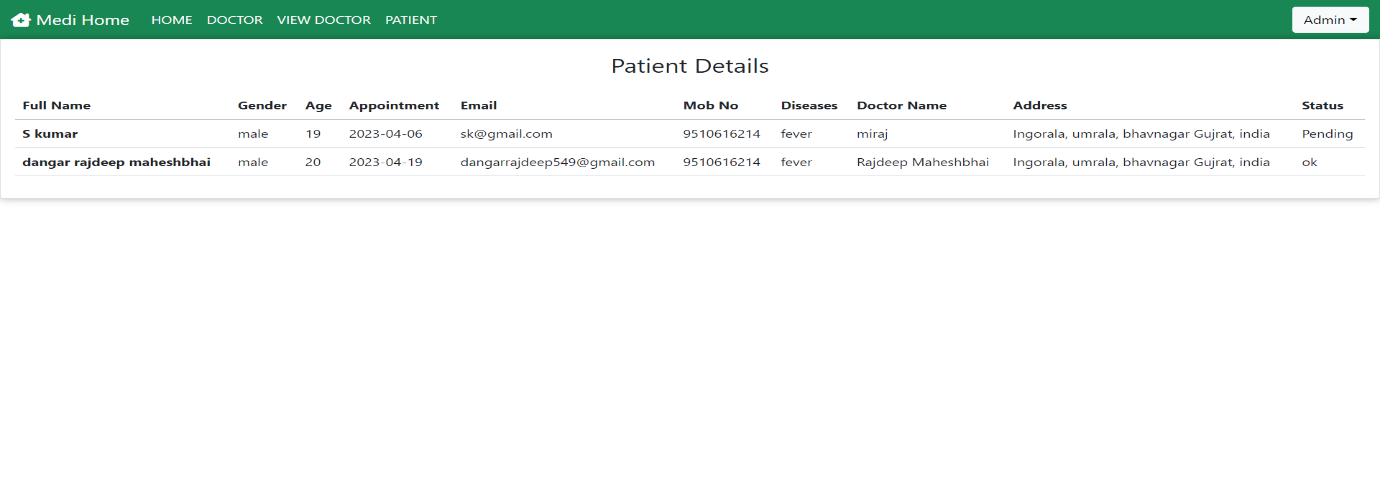
Adding Doctor:



View\_doctor:

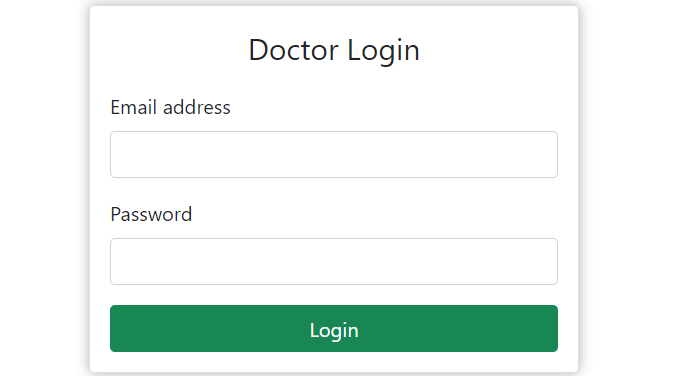


Patient status:

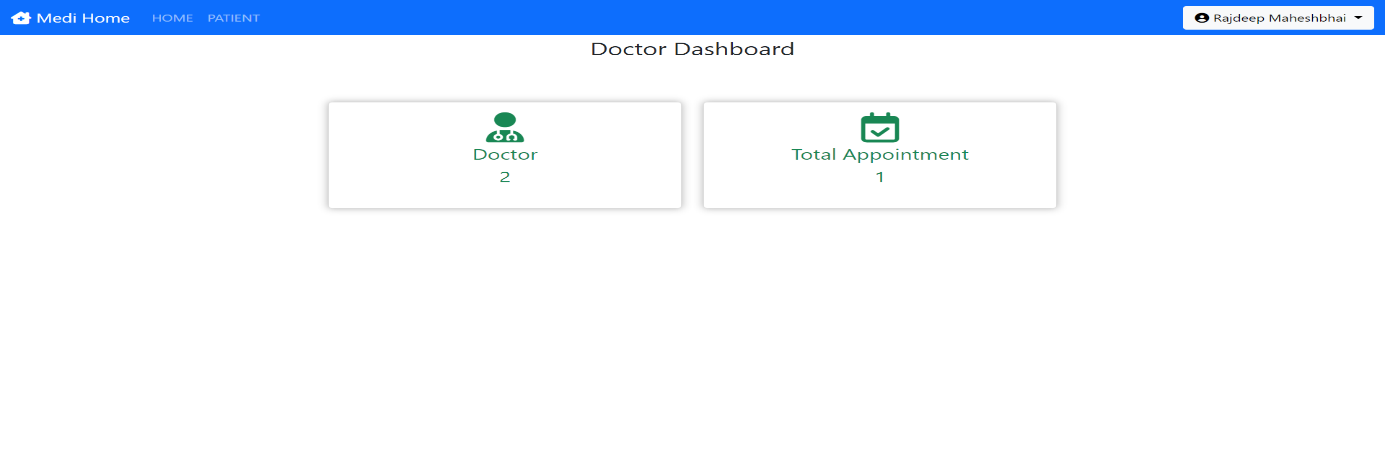


**5.3 Doctor**

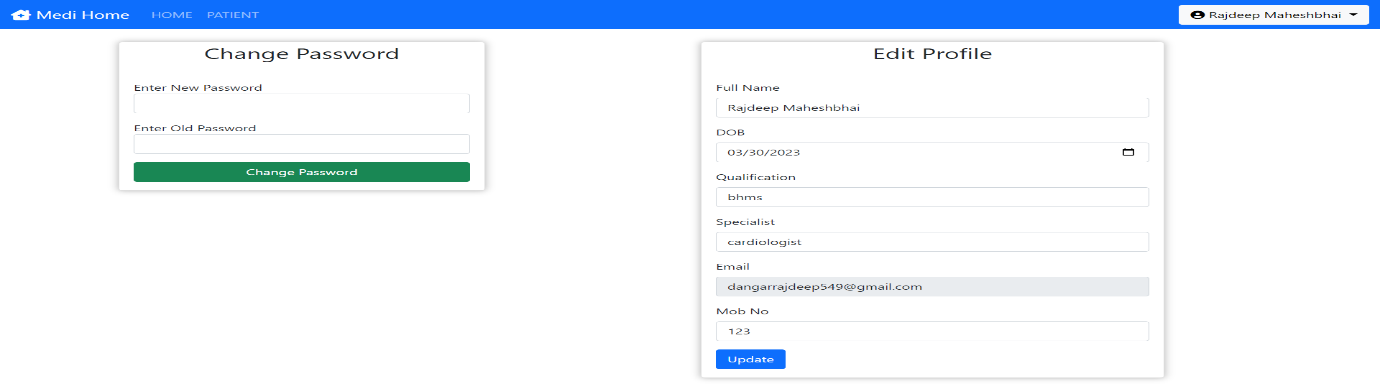
Doctor login:



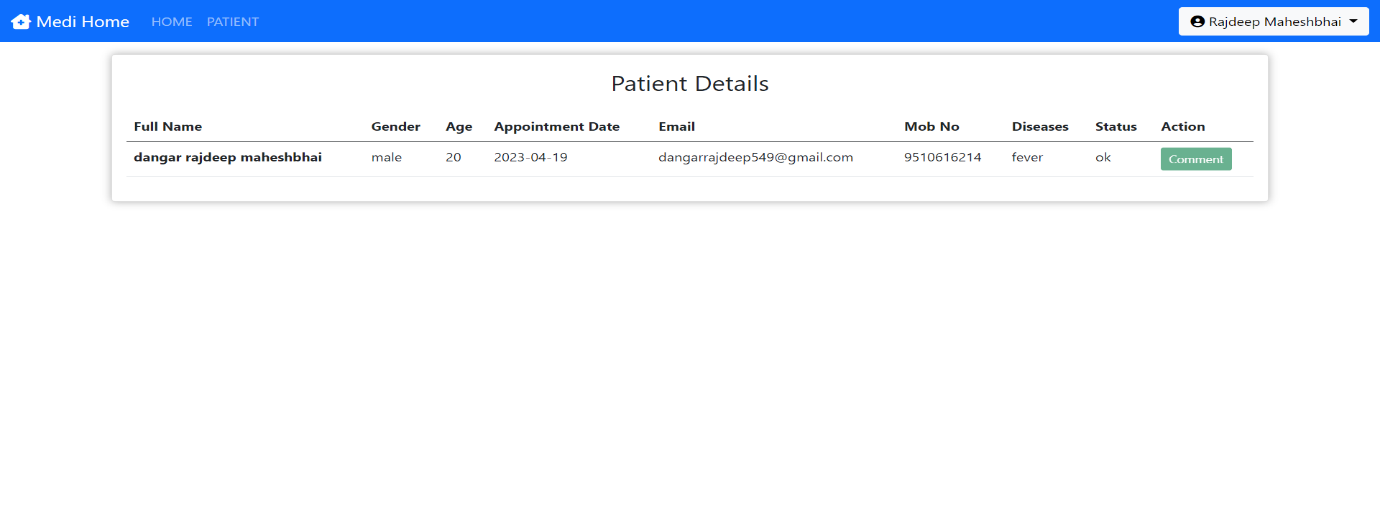
Doctor Dashboard:



Edit Doctor profile and Change Password:

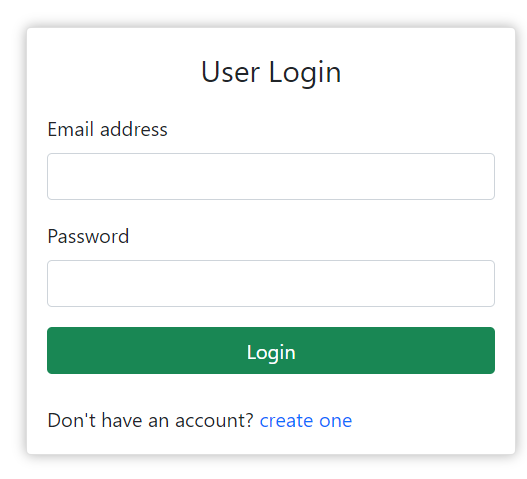
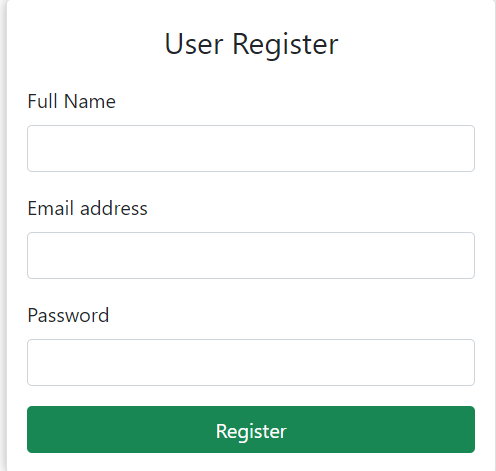


Patient Detailed:

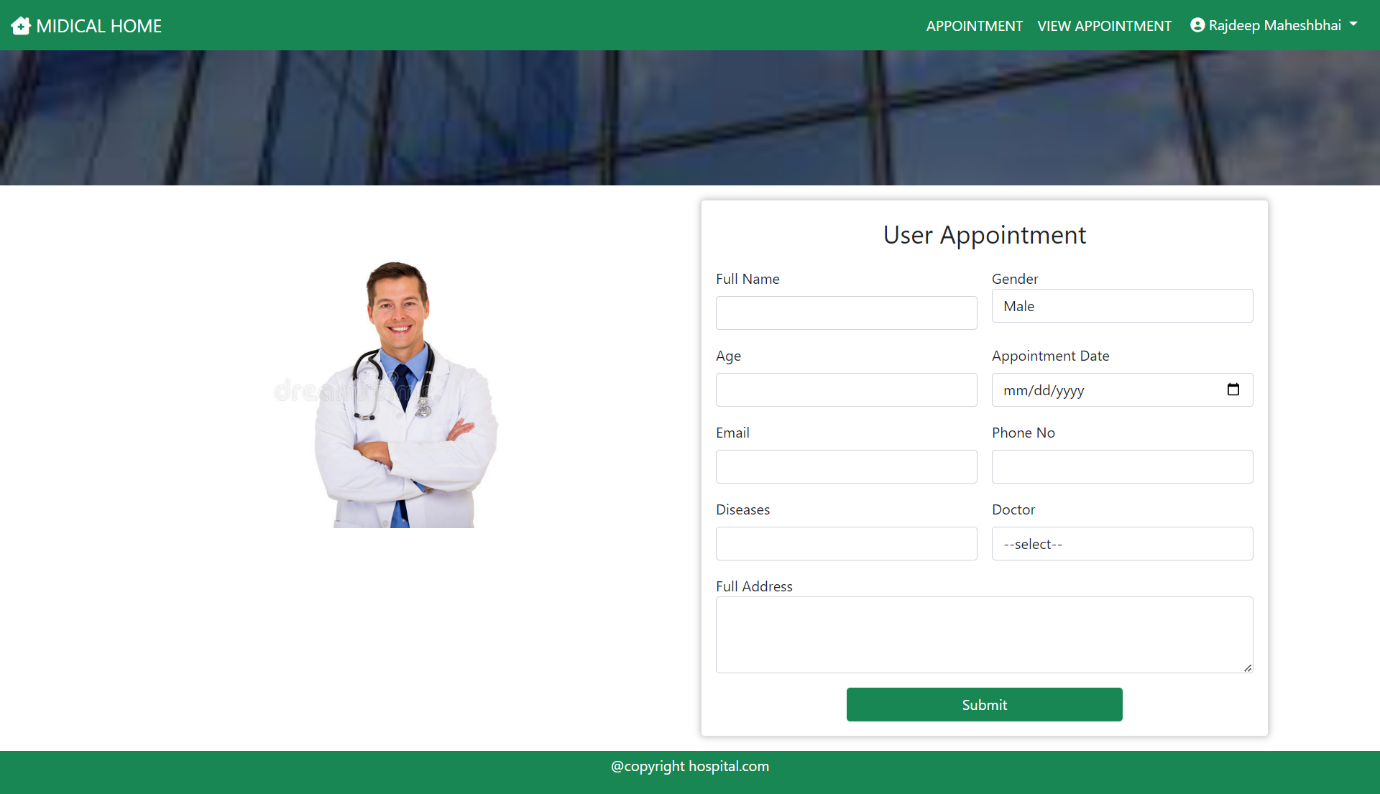


**5.4 User**

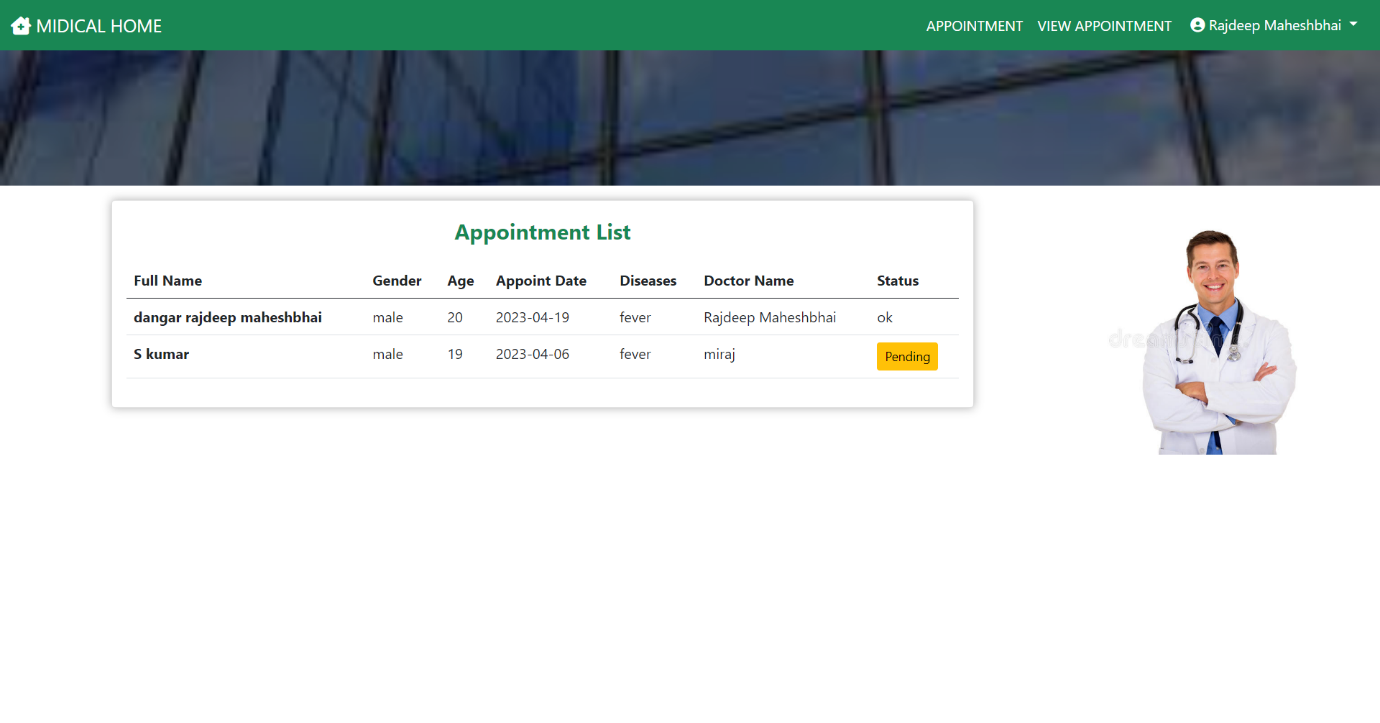
User register: User login:



User Dashboard:



View Appointment:



**6. TESTING**

**6.1 INTRODUCTION TO SYSTEM TESTING:**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTING:

Unit testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

• All field entries must work properly.

• Pages must be activated from the identified link.

• The entry screen, messages and responses must not be delayed.

Features to be tested

• Verify that the entries are of the correct format

• No duplicate entries should be allowed

• All links should take the user to the correct page.

Integration Testing:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results:

All the test cases mentioned above passed successfully. No defects encountered.

## 7.CONCLUSION

Working on the project was an excellent experience. It helped us to understand the importance of planning, designing and implementation so far we have learnt in our theory books. It helped us unleashing our creativity while working in a team. It also realized the importance of team working, communication as a part of this project.

The project was successfully completed after a lot of efforts and work hours. This project underwent number of compiling, debugging, removing errors, making it bug free, adding more facilities in Hospital Management System and interactivity making it more reliable and useful.

This project focused that scheduling a project and adhering to that schedule creates a hard sense of time- management. It has also let us known that co-operative teamwork always produce effective results.

There are also few features which can be integrated with this system to make it more flexible. Below list shows the future points to be consider :

* Paytm System.
* Including a different module for pharmacy, LAB, Bed Allotment and many more.
* Including a Frequently Asked Questions Section.

Finally, we like to conclude that we put all our efforts throughout the development of our project and tried to fulfill most of the requirements of the user.