

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
Belgavi, Karnataka, 590 014.



A Mini Project Report on
“Hospital Management System”
Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering
in
Computer Science and Engineering
Semester V

(18CSL58)

Academic Year 2020-2021

Submitted By

Mr. Suchit B Ijantakar

(USN: 2KE18CS045)

Mr. Shreyas Kulkarni

(USN: 2KE18CS042)

Under the Guidance of

Mr. Manohar Madgi

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

K. L. E. SOCIETY'S

K. L. E. INSTITUTE OF TECHNOLOGY.

Opp. Airport, Gokul, Hubballi-580 030

Phone: 0836-2232681

Website: www.kleit.ac.in





K. L. E. SOCIETY'S

K. L. E. INSTITUTE OF TECHNOLOGY.

Opp. Airport, Gokul, Hubballi-580 030

Phone: 0836-2232681

Website: www.kleit.ac.in



Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the Mini Project work entitled “**Hospital Management System**” has been completed successfully by **Mr. Suchit B Ijantakar, Mr. Shreyas Kulkarni**, bearing USN number **2KE18CS045, 2KE18CS042** in partial fulfillment for the award of degree of **Bachelor of Engineering in V Semester, Computer Science and Engineering of Visvesvaraya Technological University, Belagavi**, during the year **2020-2021**. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the said degree.

Signature of the Guide
(Mr. Manohar Madgi)

Signature of the HOD
(Dr. Yerriswamy T.)

Signature of the Principal
(Dr. B. S. Anami)

Name of the Examiners

Signature with Date

- 1.
- 2.

ACKNOWLEDGEMENT

The mini project report on “**Hospital Management System**” is the outcome of guidance, moral support and devotion bestowed on us throughout our work. For this we acknowledge and express our profound sense of gratitude and thanks to everybody who have been a source of inspiration during the project work.

First and foremost, we offer our sincere phrases of thanks with innate humility to our **Principal Dr. B. S. Anami** who has been a constant source of support and encouragement. We would like to thank our **Dean Academics Dr. S. G. Joshi** for his constant support and guidance. We feel deeply indebted to our **H.O.D. Dr. Yerriswamy sir** for the right help provided from the time of inception till date. I would take this opportunity to acknowledge our **Guide Prof. Manohar Madgi**, who not only stood by us as a source of inspiration, but also dedicated his time to enable us to present the project on time. We would be failing in endeavor, if we do not thank our **Coordinator** who have helped us in every aspect of our mini project work.

1. Suchit B Ijantakar
2. Shreyas Kulkarni

CONTENT	PAGE NO
1. Introduction	1
1.1 Background Study	2
1.2 Problem Statement	2
1.3 Aims and Objectives	2
1.4 Project Overview	3
1.5 Project Scope	3
1.6 Literature Survey	4
1.7 Report Organization	6
2. Requirement Specification	7
2.1 General Description	7
2.2 Software Requirements	7
2.3 Hardware Requirements	7
2.4 Functional Requirements	8
2.5 Non-Functional Requirements	8
2.6 Summary	8
3. Normalization	9
3.1 Functional Dependencies	10
3.2 Summary	10
4. Conceptual Representation of Application	11
4.1 ER Diagram	11
4.2 Schema Diagram	12
4.3 Summary	12
5. Design and Implementation	14
5.1 Front End	14
5.2 Back End	20
6. Testing and Validation	31
6.1 Use Case Diagram	31
6.2 Sequence Diagram	32
6.3 Result	33
6.4 Summary	33
7. Advantages and Disadvantages	34
7.1 Advantages	34

7.2 Disadvantages	34
8. Conclusion and Future Scope	35
8.1 Conclusion	35
8.2 Future Scope	35
9. Bibliography	36

ABSTRACT

Hospital Management System provides the benefits of streamlined operations, enhanced administration & control, superior patient care, strict cost control and improved profitability. HMS is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. More importantly it is backed by reliable and dependable support.

Hospital Management System facilitates online appointment booking for the patients from the doctors they want to seek medication. The task sometimes become very tedious for the compounder or the doctor himself in manually allotting the appointments for the patients as per the availability and hence is time consuming. Manual storing of patient information is a difficult work and requires large amount of paperwork hence, our system automatically stores the information in the database. It also maintains the data of the doctors and the new doctor information is added whenever he/she joins the hospital and provides ease and comfort to the patients while taking appointment from the doctors thereby resolving the problem that the patient face while making an appointment. The information is stored in the database.

The patient and doctor information are secured as every user has a unique username and password and a doctor can view the patient past history in order to treat a patient. Patient can request for the appointment in the hospital or the clinic as per the availability of doctor. Our system informs the doctor and the patient about the appointment confirmation. Admin is responsible for adding new doctors. The project 'Hospital Management System' is based on the database, object oriented and networking techniques. As there are many areas where we keep the records in database for which we are using MY SQL software which is one of the best and the easiest software to keep our information. This project uses HTML, CSS and PHP as the front-end software which is an Object-Oriented Programming and has connectivity with MY SQL

Hospital Management System is custom built to meet the specific requirement of the mid and large size hospitals across the globe. All the required modules and features have been particularly built to just fit in to your requirement. This package has been widely accepted by the clients in India and overseas. Not stopping only to this but they are highly satisfied and appreciating. Entire application is web based and built on 3 tier architecture using the latest technologies. The sound database of the application makes it more users friendly and expandable. The package is highly customizable and can be modified as per the needs and requirements of our clients. Prolonged study of the functionalities of the hospital and its specific requirement has given it a wonderful shape both technically and usability wise. It covers all the required modules right from Patient Registration, Medicine details, Doctor, Admin, Patient appointment, record modification, discharge details etc.

CHAPTER 1

INTRODUCTION

Hospital Management System is a smart web application. The purpose of Hospital Management System is to automate the existing manual system by help of computerized equipments and full-fledged computer software, fulfilling their requirements, so their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. This provides a registration and login for both doctors and patients and only login for admin. After the successful registration of patient, the patient can login by giving username and password.

Admin is the one who maintains the database. He has a privilege to add and remove the doctors as well as patients. Patient must create an account by filling the required details in the sign-up form for the first time. Patient must login to book an appointment later whenever he wants to take an appointment. Our system allows the patients to select the medical field like dermatologist, homeopathy, general physician, etc. The search results will show the list of doctors matching patient's requirement criteria and he can request the required doctor. The patient can also cancel the appointment if he changes his mind. If both patient and doctor are satisfied, then the patient receives the appointment confirmation.

Hospital Management System as described above, can lead to free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on record keeping. Basically, the project describes how to manage for good performance and better services for the patients.

The application is reduced as much as possible to avoid errors while entering the data. Also, for those busy executives who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all time. This system will ultimately allow you to better manage resources.

1.1 Background study

Traditionally, appointment was done manually, and it required numerous paper forms with data stores spread throughout the hospital management infrastructure. This made the management and maintenance of critical information complex.

We do not usually visit hospitals unless we need to, but this visit does not always give us good experience. We face serious problems such as not getting appointment at required time. Even the doctors do not have access to their patient's data while they are at home. It will be great for the patients to have an application that will keep them close to their personal data like prescriptions, test reports and past history.

1.2 Problem statement

Lack of immediate retrievals: It is very tedious to go through the records of any patient as and when it is needed because patient's information is written in the registers manually.

Lack of immediate information storage: The information generated by various transactions takes time and efforts to be stored at the right place.

Lack of prompt updating: Various changes to information like patient details are difficult to update as paperwork is involved and the process is more tedious.

Preparation of accurate and prompt reports: This becomes a difficult task as information is difficult to collect from various register.

1.3 Aims and Objectives

Main objectives of Hospital Management System are:

- Design a system for better patient care.
- Reduce hospital operating costs.
- Facilitates taking appointment from any place at any desirable time.
- Recording the information about the patient's that come.
- Recording information related to diagnosis given to patients.
- Maintaining the patient's past history.
- Providing the facility to doctor to access the patient's information.

1.4 Project Overview

Hospital Management System provides the benefits of streamlined operations, enhanced administration & control, superior patient care and improved profitability. HMS is powerful, flexible, and easy to use and is designed and developed to deliver real conceivable benefits to hospitals. More importantly it is backed by reliable and dependable support.

The project ‘Hospital Management System’ is based on the database, object oriented and networking techniques. As there are many areas where we keep the records in database for which we are using MY SQL software which is one of the best and the easiest software to keep our information. This project uses HTML, CSS and PHP as the front-end software which has connectivity with MY SQL.

Hospital Management System is custom built to meet the specific requirement of the mid and large size hospitals across the globe. All the required features have been particularly built to just fit in to the requirement. Entire application is web based and built on 3 tier architecture using the latest technologies. The sound database of the application makes it more users friendly and expandable.

The patient registers himself and logs into his account. He requests an appointment from the doctor with a particular specialty at desired date and time. The doctor logs in and checks if there are any appointment requests. If the patient wishes to, he can also cancel the appointment. If the appointment is confirmed he can visit the doctor on the approved time and date. The entire system is managed by the admin.

1.5 Project Scope

This system can be implemented to any mid and large sized hospital across the globe. This HMS is designed for multispecialty 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 and critical financial accounting, in a seamless flow.

If hospitals are providing an online portal where the users can have the ability to book an appointment from anywhere, it would save the users time to physically go to the hospital and book a slot. Since the application is available in the web it is easily accessible and always available.

1.6 Literature Survey

The purpose of writing this literature review was to find out how hospital management systems in various hospitals work and compare it to our hospital management system.

1.6.1 CASE STUDY: THE ST. ROSS HOSPITAL

St. Ross hospital is situated in Maputo city in Mozambique. On average, around 500 patients per day tend to visit the hospital, resulting in large queues. With such a diverse range of services offered. St. Ross hospital needed to implement a Patient Management solution.

The Challenge

The challenge was the fact that many patients, particularly those who travel long distances to visit the department, tend to arrive hours before their scheduled appointment, further adding to the existing queues. Patient visibility was a further challenge; with staff time wasted searching for patients through manual records.

Functionality of the new system

When a patient report to the registration point, the clerks on duty must establish whether the patient has a number and whether a file exists as a result of a previous attendance in the system. In many instances it will be possible to establish that a file exists by obtaining the number from the patient's hospital identity card. when a patient does not have an identity card or has forgotten to bring it, the main list will be needed to trace the number so that the patient's file can be retrieved. When the patient's name is a common one, the records clerk will need to interview the patient or a relative to obtain sufficient information to identify the patient from others of the same name who appear in the index. Before opening a new file, clerks must be instructed to make careful checks that no file already exists. The supervisor should regularly monitor that these checks are being done. patients may claim that they have never attended the hospital before, yet a check of the main file reveals that a file already exists.

Administration: The administrator can create User accounts for the staff, define the departments of the hospital and schedule for the doctors and other staff members which can be viewed all through the application.

Registration: The registration module captures the complete patient's information with a unique identification number.

Store: This module keeps the watch over the stock/issue of various medicines to the Dispensary department.

The Solution

Using patients register their arrival by entering the index on their appointment letter. Each patient enters a required queue, which respects the scheduled appointment time. Patients who arrive early are reminded of their actual appointment time. Patients who forget their appointment letter can check in manually by entering their first name into the system. Volunteers are also on hand to assist.

The Results

The system has reduced the queue completely, greatly reducing stress for staff and ensuring a good experience for patients. Data captured using the system will be used to improve appointment scheduling in the future as it provides a clear picture on patient flow from entry to exit. The system provides information on daily work flow, which has the benefit of enabling roster makers to improve staff rosters in the future and ensure high traffic areas are appropriately looked upon.

1.6.2 CASE STUDY: NYERERE NATIONAL HOSPITAL

Dr Korir Mohamed is Chairman of Nyerere Hospital, in Tanzania that I interviewed the following version of the interview.

I would like to know a little about Nyerere Hospital-its origin, history and how it's come to Tanzania.

Nyerere hospital is an international hospital currently with operations and developments in the whole country of Tanzania. It is situated along Mamlaka road in Dodoma town. It started in the year 1934 with a small number of patients attending it per day and it has grown until now with an average of 5000 patients visiting various parts of the hospital each day.

Tell me, what was the kind of system your hospital was using before.

The old system registration of patients is done by just writing the Patients name, age and gender on a paper. Whenever the Patient comes up his information is stored freshly no reference. Bills are generated by recording price for each facility provided to Patient on a separate sheet and at last they all are summed up. Diagnosis information to patients is generally recorded on the document, which contains patient information. All this work is done manually by the receptionist and other operational staff and lot of papers are needed to be handled and taken care of. The information is very difficult to retrieve and to find particular information like to find out about the patient's history, the user has to go through various registers. This results in inconvenience and wastage of time. The information generated by various transactions takes time and efforts to be stored at right place. Various changes to information like patient details or immunization details of child are difficult to make as paper work is involved. This becomes a difficult task as information is prepared manually.

What is the kind of system your hospital is using now?

The administrator can create User accounts for the staff, define the departments of the hospital and schedule for the doctors and other staff members which can be viewed all through the application. The registration module can capture the complete patient's information with a unique identification number. Can store and keep the watch over the stock/issue of various medicines to the Dispensary department. Can generate the bill is generated once the patient is discharged by the Doctor.

Conclusion

In this information age, it is therefore essential that hospital management system be done with the utmost efficiency and accuracy. This is the point at which records management in hospital is integrated with computer science in order to develop a computer-based records management system. The conclusion is that efficient and comprehensive records keeping is as good as guaranteed when the art of recordkeeping is simulated and integrated into a computerized records management system.

1.7 Organization of report

The report is organized as follows:

Chapter 1: This chapter presents introduction, problem statement, aims and objectives of the project, project overview, project scope and literature survey

Chapter 2: This chapter explains general description, software requirements, hardware requirements, functional and non-functional requirements.

Chapter 3: This chapter explains normalization and functional dependency.

Chapter 4: This chapter shows the conceptual representation including ER diagram and Schema diagram.

Chapter 5: This chapter explains front end and back end implementation details.

Chapter 6: This chapter showcases the testing and validation process with use case diagram and Sequence diagram and with result obtained for the proposed project.

Chapter 7: This chapter explains the advantages and disadvantages concerning the hospital management system.

Chapter 8: This chapter talks about conclusion and future scope.

Chapter 9: Reference and Bibliography.

CHAPTER 2

REQUIREMENT SPECIFICATION

In this chapter, we will discuss and analyses about developing process of online shopping Management System including software requirement specification. The functional and non-functional requirements are included in requirement part to provide complete description and overview of system requirement before the developing process is carried out.

2.1 General Description

The system is made up of units that are put together to work as one in order to achieve a common goal. The requirements for implementation of the new system are:

1. The Hardware
2. The Software

2.2 Software Requirements

For the effective implementation of the system, the following software must be installed on the computer.

Operating system: Windows Operating system

Database: my SQL

Web Server: XAMPP

IDE: Brackets

Web Technologies: HTML, CSS, php

2.3 Hardware Requirements

Processor: Intel core i3 or above

Processor Speed: 1.0 GHz or above

RAM: 1GB or above

Operating system: Windows 7, 8, 8.1, 10

2.4 Functional Requirements:

A Functional Requirement defines a function of system or its component. It includes descriptions of the data to be entered into the system, operations to be performed, workflows and who can enter the data into the system.

Security: This Application prevent unauthorized to access the personal information of both Doctors and Patients. It supports administrator-controlled data access feature capabilities i.e. viewing, entry, adding, deleting.

Administrative Functions: In this Application, the administrators use software to store and organize the data entered by the users.

Authentication: The Admin is given the entire authentication to view and insert the data into the tables.

Historical Data: This system stores the data generated either manually or automatically within the system. The data is also secured within the system.

2.5 Non-Functional Requirements:

Non-Functional requirement is a requirement that specifies criteria that can be used to judge the operation of the system.

Availability: The Hospital Management System is 100% available for the user(patient) and can be used 24 hours a day and a patient can take an appointment at any time and at any desired date.

Accuracy and Validity: The system has various data quality assurance elements, including search functions to match records to identifiers.

Compatibility: The Hospital Management System is able to operate on various versions of windows operating systems and browsers.

Usability: The Application can provide a pleasing and satisfying interaction for the user. It is user friendly and not at all complicated to handle.

2.6 Summary

In this chapter we explained general description, software requirements, hardware requirements, functional and non-functional requirements.

CHAPTER 3

NORMALIZATION

Normalization is the process of organizing the columns (attributes) and tables (relations) of a relational database to reduce data redundancy and improve data integrity. Normalization is also the process of simplifying the design of a database so that it achieves the optimal structure composed of atomic elements.

Normalization rules are divided into following normal form.

- **First Normal Form:** A relation is said to be in 1NF, if all values in given relation are atomic.
- **Second Normal Form:** A relation is said to be in 2NF, if it satisfies 1NF and the non-prime attributes should be fully functionally dependent on key of relation.
- **Third Normal Form:** A relation is said to be in 3NF, if it satisfies 2NF and non-prime attribute is transitively dependent on primary key.
- **Boyce Codd Normal Form:** A relation is said to be in BCNF, if and only if there are no non-trivial functional dependencies of attributes on anything or other than a super key or candidate key.

Database Type	Tables
Denormalized	Admin, doctor, patient.
1NF	Admin, doctor, patient, appointment, doctor specialization
2NF	Admin, doctor, patient, appointment, doctor specialization, medical history, user
3NF	Admin, doctor, patient, appointment, doctor specialization, medical history, user, doctor log, user log, contact us.

Table 3.1. Normalized Table and its Attributes

3.1 Functional Dependencies

There is a main authenticated admin who manages the system. Admin can access the system through Admin name, Admin password. To gain the access and to confirm the appointment admin should register into the system.

There can be one or more patients who can access the system by registering themselves into the system and login by providing user id, user name, contact no, email.

3.2 Summary

In this chapter we explained normalization and functional dependency.

CHAPTER 4

CONCEPTUAL REPRESENTATION OF APPLICATION

In this chapter we will study detailed design used to build the online shopping system.

4.1 ER Diagram

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes

An entity-relationship model (ERM) in software engineering is an abstract and conceptual representation of data. Entity-relationship modeling is a relational schema database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion.

ER Diagram Description:

The above is an entity relationship diagram of doctor appointment scheduling system. It consists of the following tables and their functionalities.

Admin: An admin can view all the details of patients and doctors.

Patient: A patient can view the doctors list based on requirement criteria and can book an appointment.

Doctor: A doctor confirms the appointment request and can view patient past history.

Hospital: If the patient is given the appointment in hospital, the patient can visit the hospital for his treatment.

Schedule Appointment: After both the patient and doctors are satisfied with the appointment the data is stored in confirmation appointment.

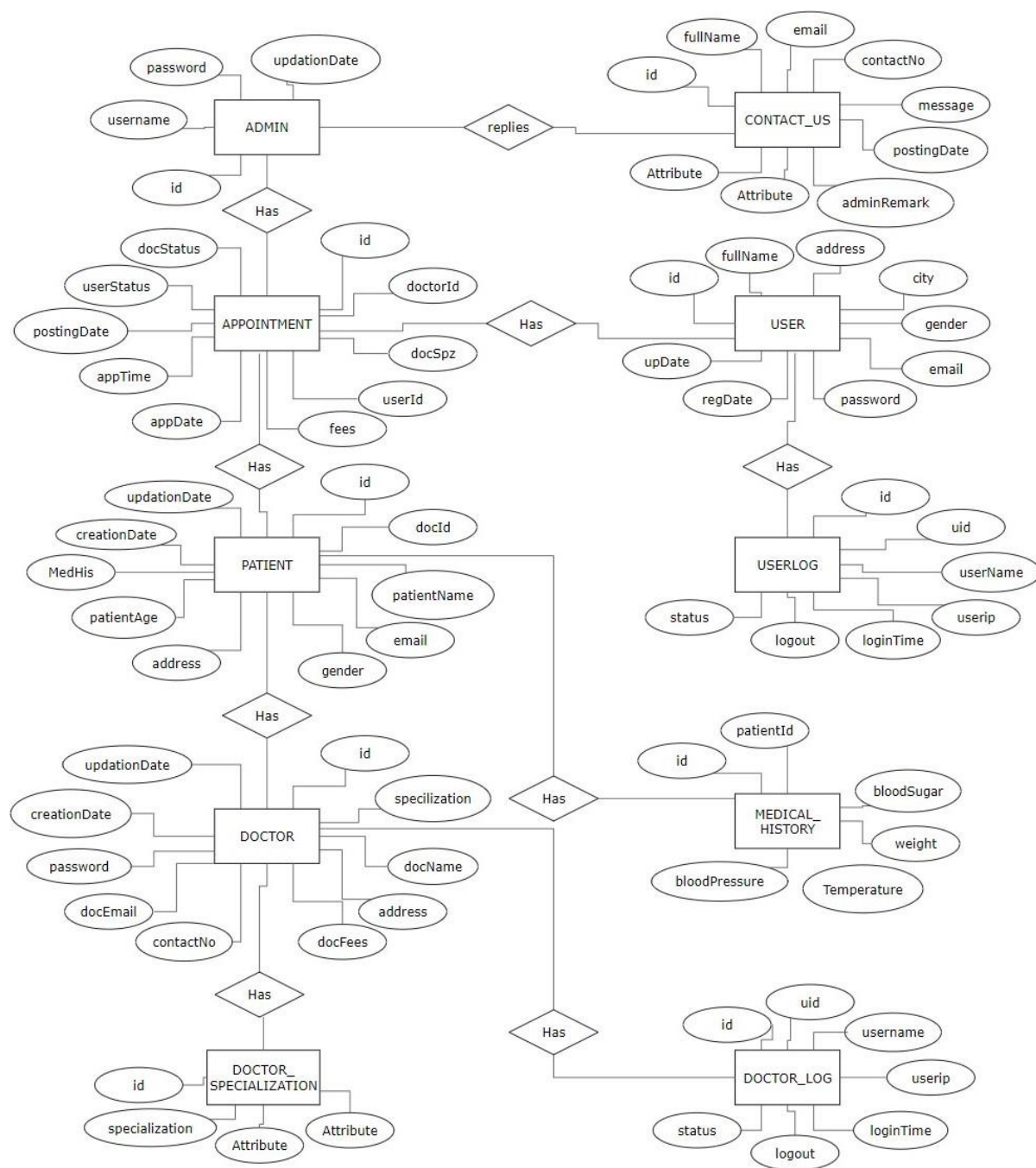


Figure 4.1. ER Diagram for Hospital Management System

4.2 Schema Diagram

A schema is the structure behind data organization. It is a visual representation of how different table relationships enable the schema's underlying mission business rules for which the database is created.

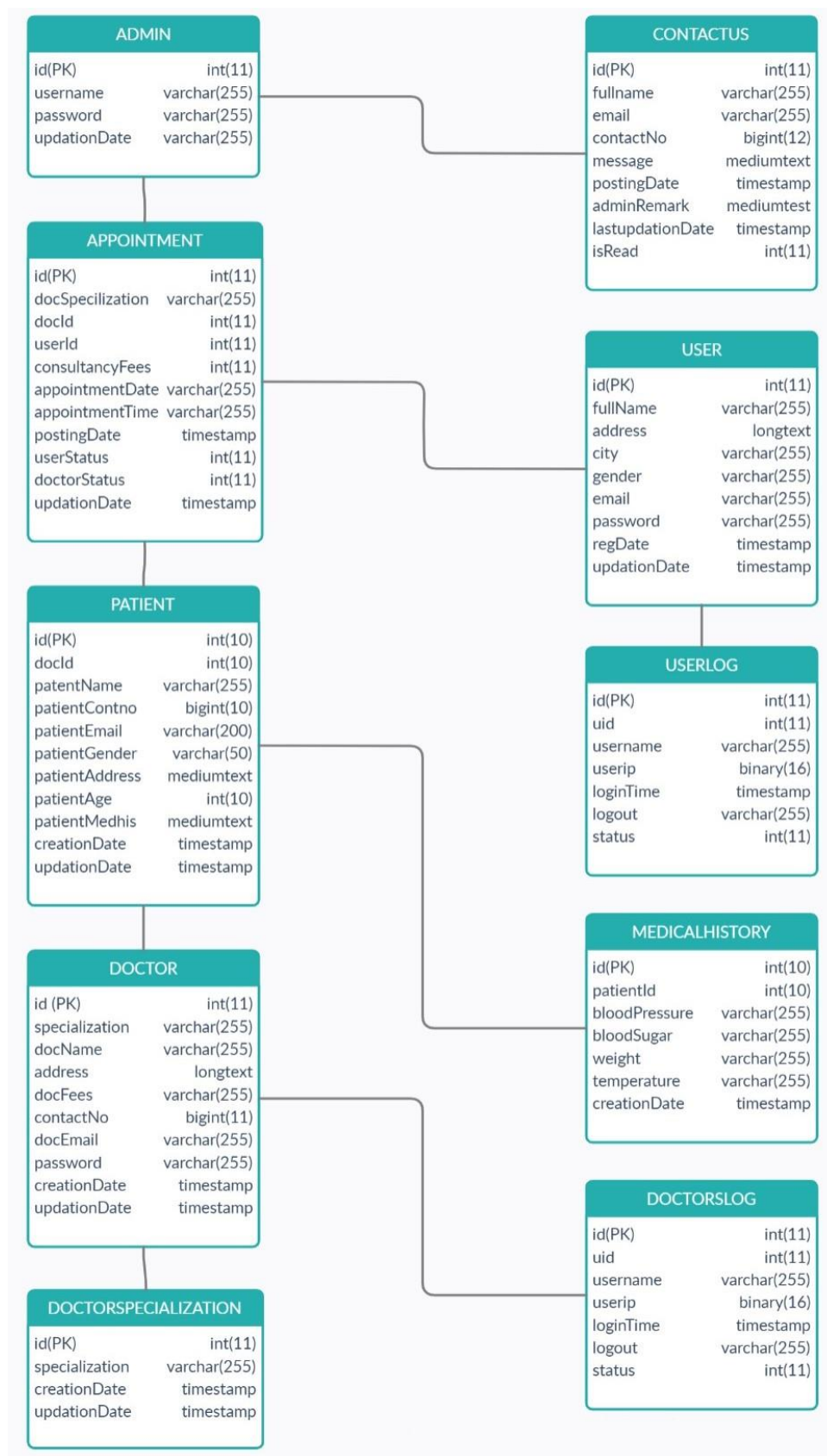


Figure 4.2. Schema Diagram for Hospital Management System

Chapter 5

Design and implementation

5.1 Front End

It contains a detailed description about each interface along with a screen shot of the interface.

5.1.1 Home Page:

Actor: Any users.



Fig.5.1. Home page.

Flow:

1. Any user can browse this page.

5.1.2 Admin Login Page:

Actor: Admin users

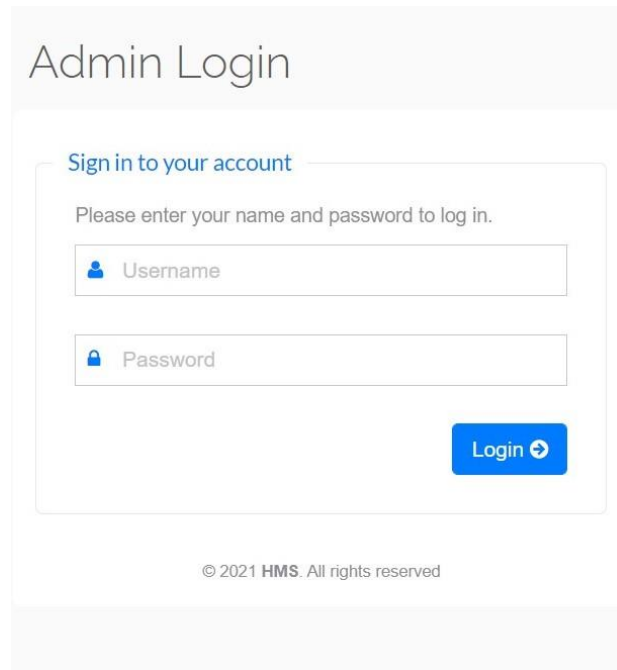
A mockup of an Admin Login page. The page has a light gray background. At the top, the title "Admin Login" is displayed in a large, dark gray font. Below the title, there is a white rectangular box containing the login form. Inside the box, the text "Sign in to your account" is written in a blue font. Below this, a message "Please enter your name and password to log in." is displayed in a small, gray font. There are two input fields: the first is labeled "Username" with a blue user icon to its left, and the second is labeled "Password" with a blue lock icon to its left. Both fields are white with a light gray border. To the right of the password field is a blue button with the text "Login" and a white right-pointing arrow. At the bottom of the white box, the copyright notice "© 2021 HMS. All rights reserved" is written in a small, gray font.

Fig.5.2. Admin login page.

Flow:

1. Only Admins can browse this page.
2. Admin user can control all patient appointments and all user's activities.

5.1.3 Admin user details page:

Actor: Admin users.

Flow:

1. Admin user can browse this page.
2. Admin user can view all module here.

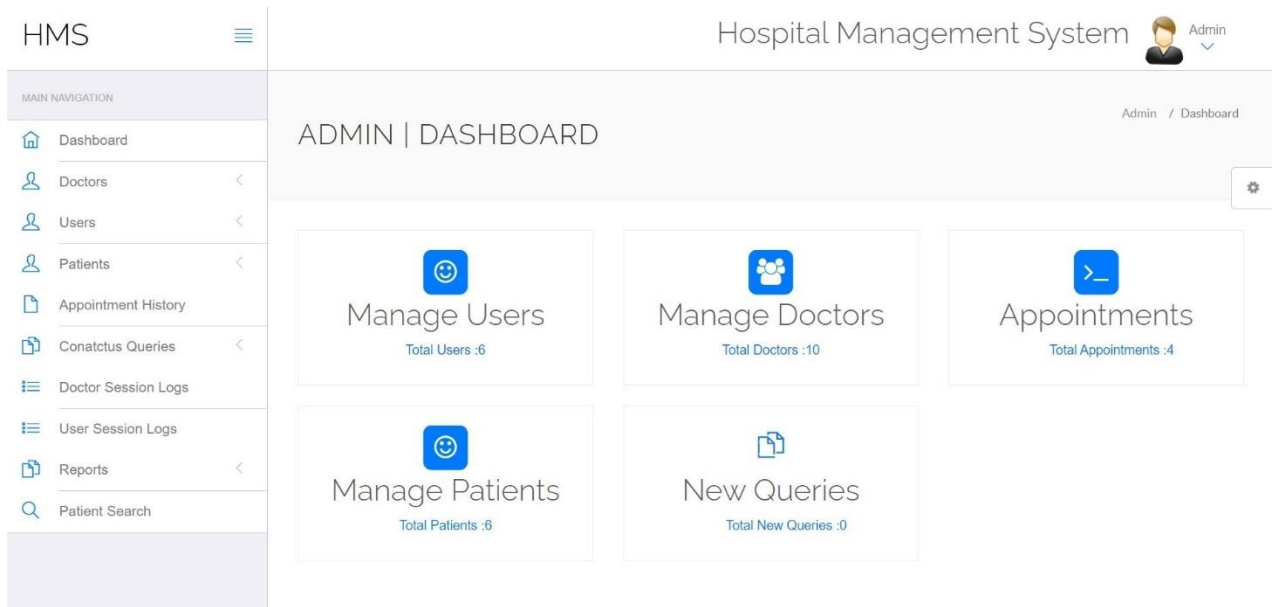


Fig.5.3. Admin user detail page.

5.1.4 Appointment History Page:

Actor: Admin users.

#	Doctor Name	Patient Name	Specialization	Consultancy Fee	Appointment Date / Time	Appointment Creation Date	Current Status	Action
1.	Aditya Patil	Vishal Kaliwal	Audiologist	800	2021-01-02 / 9:15 AM	2021-01-03 00:01:28	Cancel by Doctor	Canceled
2.	Pooja Kittur	Vibhav Nisarikar	Gynaecologist	600	2021-01-02 / 1:00 PM	2021-01-02 15:58:54	Active	No Action yet
3.	Adesh Mantur	Priyanka Priya	Cardiologist	900	2021-01-10 / 5:45 PM	2021-01-08 17:42:41	Active	No Action yet

© 2021 HMS. All rights reserved

Fig.5.4. Appointment history page.

Flow:

1. Admin users can browse this page.

2. Admin users can view all equipment history.

5.1.5 User login page:

Actor: User.

HMS | Patient Login

[Sign in to your account](#)

Please enter your name and password to log in.

[Forgot Password ?](#)

[Login](#)

Don't have an account yet? [Create an account](#)

© 2021 HMS. All rights reserved

Fig.5.5. Patient login page.

Flow:

1. User login is with their user email and password

Alternate Flow:

1. If the user email is wrong then it is asked to login again.
2. If the password is wrong then the user is asked to enter again.

5.1.6 User account create page:

Actor: User.

HMS | Patient Registration

Sign Up

Enter your personal details below:

Full Name

Address

City

Gender

☐ Female ☐ Male

Enter your account details below:

Email

Password

Password Again

☒ I agree

Already have an account? [Log-in](#)

Submit

© 2021 HMS. All rights reserved

Fig.5.6. User account create page.

Flow:

1. Password must be more than 8 characters.
2. Password and confirm password must be same.

Alternate Flow:

1. If the mandatory fields are not fill up then alert is shown.
2. If password is less than 8 characters or password and confirm password does not match an alert is shown.

5.1.7 User detail page:

Actor: User.

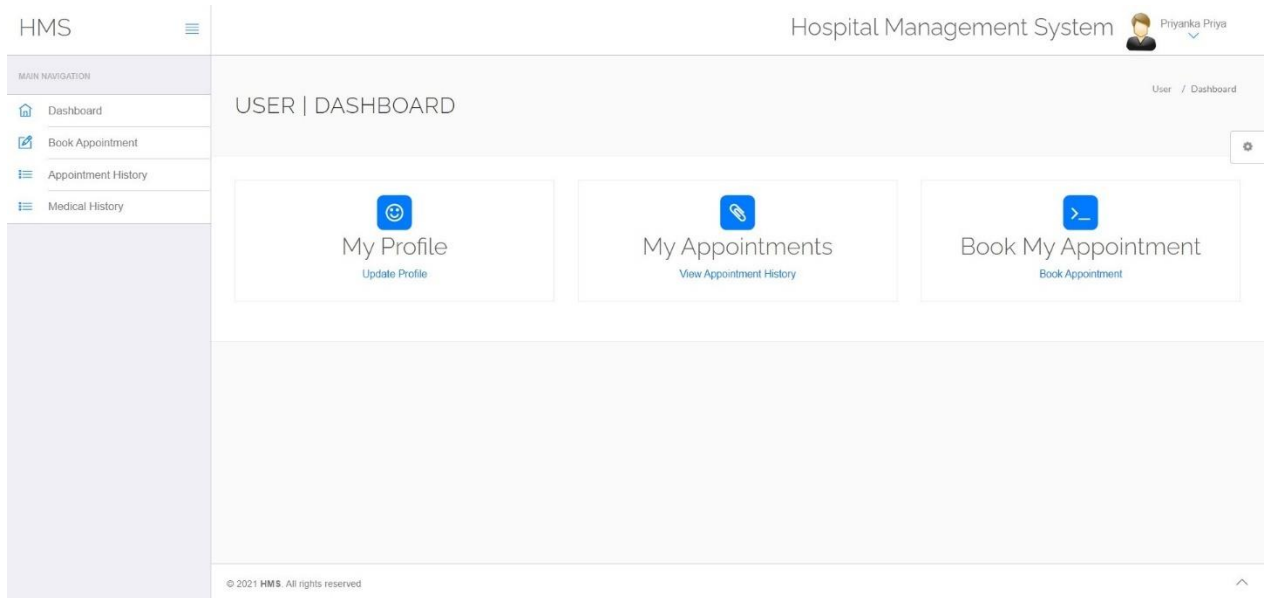


Fig.5.7. User detail page

Flow:

1. User must fill up all input fields.

Alternate Flow:

1. If all mandatory fields are not filled then an alert is shown.

5.1.8 Doctor login Page.

Actor: Doctor.



The image shows a login page for a Hospital Management System (HMS) for doctors. The page has a light gray header with the text "HMS | Doctor Login". Below the header is a white box with a blue border. Inside the box, the text "Sign in to your account" is followed by "Please enter your name and password to log in." There are two input fields: "Username" with a blue person icon and "Password" with a blue lock icon. Below the password field is a link "Forgot Password ?". A blue "Login" button with a white right arrow is at the bottom right of the box. At the bottom of the page, there is a copyright notice: "© 2021 HMS. All rights reserved".

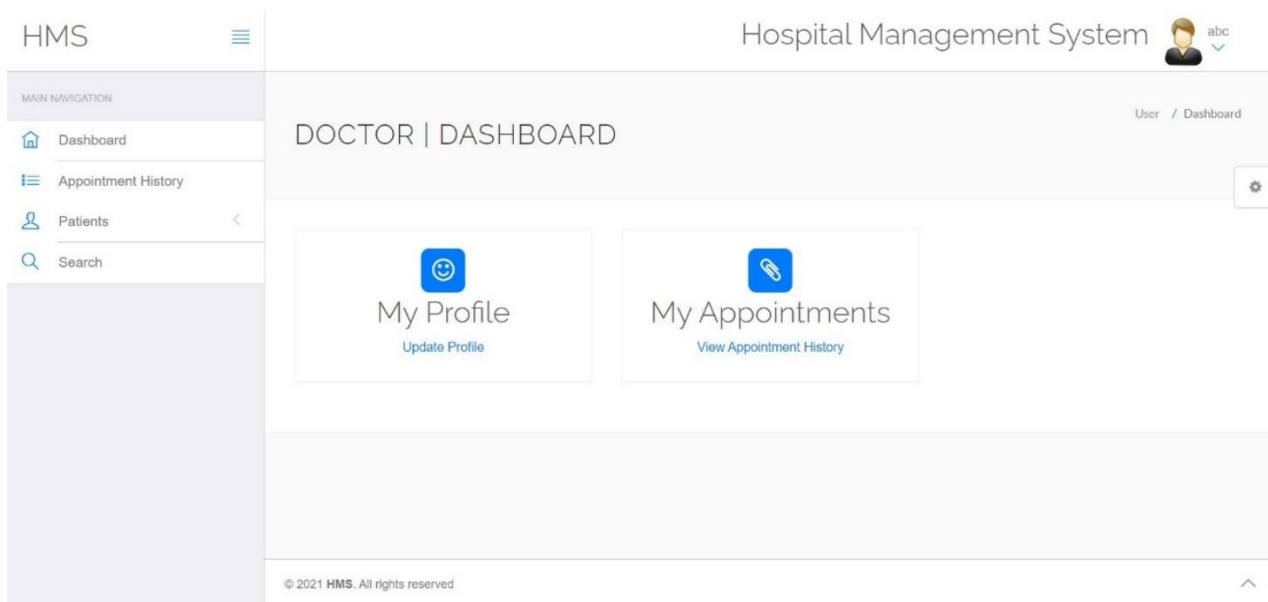
Fig.5.8. Doctor login page.

Flow:

1. Doctor logs in with the email and password.

5.1.9 Doctor detail page.

Actor: Doctor.



The image shows the doctor dashboard page of the Hospital Management System (HMS). The page has a light gray header with the text "HMS" and a hamburger menu icon. The main header area contains "Hospital Management System" and a user profile icon with the text "abc". Below the header is a white box with a blue border. Inside the box, the text "DOCTOR | DASHBOARD" is followed by "User / Dashboard". There are two main sections: "My Profile" with a blue smiley face icon and "Update Profile" link, and "My Appointments" with a blue calendar icon and "View Appointment History" link. At the bottom of the page, there is a copyright notice: "© 2021 HMS. All rights reserved".

Fig.5.9. Doctor detail page.

Flow:

1. Doctors can browse this page.

5.2 Back End

5.2.1 Creation and insertion of tables:

Admin

```
CREATE TABLE `admin` (
  `id` int(11) NOT NULL,
  `username` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `updatationDate` varchar(255) NOT NULL
)

INSERT INTO `admin` (`id`, `username`, `password`, `updatationDate`) VALUES
(1, 'Suchit', 'Suchit@12345', '01-01-2021 11:42:05 AM'),
(2, 'Shreyas', 'Shreyas@12345', '01-01-2021 11:43:05 AM');

ALTER TABLE `admin`

ADD PRIMARY KEY (`id`);
```

id	username	password	updatationDate
1	Suchit	Suchit@12345	01-01-2021 11:42:05 AM
2	Shreyas	Shreyas@12345	01-01-2021 11:43:05 AM

Appointment

```
CREATE TABLE `appointment` (
  `id` int(11) NOT NULL,
  `doctorSpecialization` varchar(255) DEFAULT NULL,
  `doctorId` int(11) DEFAULT NULL,
  `userId` int(11) DEFAULT NULL,
```

```

`consultancyFees` int(11) DEFAULT NULL,
`appointmentDate` varchar(255) DEFAULT NULL,
`appointmentTime` varchar(255) DEFAULT NULL,
`postingDate` timestamp NULL DEFAULT current_timestamp(),
`userStatus` int(11) DEFAULT NULL,
`doctorStatus` int(11) DEFAULT NULL,
`updationDate` timestamp NULL DEFAULT NULL ON UPDATE
current_timestamp()
)

INSERT INTO `appointment` (`id`, `doctorSpecialization`, `doctorId`, `userId`,
`consultancyFees`, `appointmentDate`, `appointmentTime`, `postingDate`, `userStatus`,
`doctorStatus`, `updationDate`) VALUES

(3, 'Audiologist', 2, 6, 800, '2021-01-02', '9:15 AM', '2021-01-02 18:31:28', 1, 0, '0000-
00-00 00:00:00'),

(4, 'Gynaecologist', 5, 5, 600, '2021-01-02', '1:00 PM', '2021-01-02 10:28:54', 1, 1,
'0000-00-00 00:00:00'),

(5, 'Dermatologist', 9, 7, 700, '2021-01-03', '5:30 PM', '2021-01-03 18:41:34', 1, 0,
'2021-01-03 18:48:30');

ALTER TABLE `appointment`

ADD PRIMARY KEY (`id`);

```

id	doctorSpecialization	doctorId	userId	consultancyFees	appointmentDate	appointmentTime	postingDate	userStatus	doctorStatus	updationDate
3	Audiologist	2	6	800	2021-01-02	9:15 AM	2021-01-03 00:01:28	1	0	0000-00-00 00:00:00
4	Gynaecologist	5	5	600	2021-01-02	1:00 PM	2021-01-02 15:58:54	1	1	0000-00-00 00:00:00
5	Dermatologist	9	7	700	2021-01-03	5:30 PM	2021-01-04 00:11:34	1	0	2021-01-04 00:18:30

Doctor

```

CREATE TABLE `doctors` (
`id` int(11) NOT NULL,
`specilization` varchar(255) DEFAULT NULL,
`doctorName` varchar(255) DEFAULT NULL,
`address` longtext DEFAULT NULL,

```

```
`docFees` varchar(255) DEFAULT NULL,  
`contactno` bigint(11) DEFAULT NULL,  
`docEmail` varchar(255) DEFAULT NULL,  
`password` varchar(255) DEFAULT NULL,  
`creationDate` timestamp NULL DEFAULT current_timestamp(),  
`updationDate` timestamp NULL DEFAULT NULL ON UPDATE  
current_timestamp()  
)  
  
INSERT INTO `doctors` (`id`, `specilization`, `doctorName`, `address`, `docFees`,  
`contactno`, `docEmail`, `password`, `creationDate`, `updationDate`) VALUES  
  
(1, 'Cardiologist', 'Adesh Mantur', 'Bijapur', '900', 9945574163,  
'adeshmantur12@gmail.com', 'Am@12345', '2021-01-01 06:00:00', '2021-01-01  
06:00:00'),  
  
(2, 'Audiologist', 'Aditya Patil', 'Koppal', '800', 7892246734,  
'adityapatil23@gmail.com', 'Ap@12345', '2021-01-01 07:00:00', '2021-01-01  
07:00:00'),  
  
(3, 'Dermatologist', 'Anup Kabberalli', 'Tumkur', '900', 8494809175,  
'anupkabberhalli90@gmail.com', 'Ak@12345', '2021-01-01 08:00:00', '2021-01-01  
08:00:00'),  
  
(4, 'Dentist', 'Mayur Hegde', 'Bagalkot', '700', 7022808747,  
'mayurhegde34@gmail.com', 'Mh@12345', '2021-01-01 09:00:00', '2021-01-01  
09:00:00'),  
  
(5, 'Pulmonologist', 'Pooja Kittur', 'Mysore', '700', 7349131435,  
'poojakittur78@gmail.com', 'Pk@12345', '2021-01-01 10:00:00', '2021-01-01  
10:00:00'),  
  
(6, 'Oncologist', 'Priyanka Sonu', 'Dharwad', '800', 9480641514,  
'priyankasonu89@gmail.com', 'Ps@12345', '2021-01-01 11:00:00', '2021-01-01  
11:00:00'),  
  
(7, 'Demo test', 'Shreyas Kularni ', 'Hubballi', '900', 9876543210, 'test@demo.com',  
'f925916e2754e5e03f75dd58a5733251', '2021-01-01 08:08:58', '2021-06-23 18:17:25'),  
  
(8, 'Gynaecologist', 'Vaibhav Khashyap', 'Hubballi', '600', 6361736455,  
'vaibhavkhashyap56@gmail.com', 'Vk@12345', '2021-01-01 12:00:00', '2021-01-01  
12:00:00'),
```

(9, 'Paediatrician', 'Venkatesh Mamdapur', 'Belgaum', '600', 7204761700, 'venkateshmamdaour67@gmail.com', 'Vm@12345', '2021-01-01 13:00:00', '2021-01-01 13:00:00'),

(10, 'Obstetrician', 'Vrushabh Turamari', 'Bangalore', '600', 9019818653, 'vrushabhturamari45@gmail.com', 'Vt@12345', '2021-01-01 14:00:00', '2021-01-01 14:00:00');

ALTER TABLE `doctors`

ADD PRIMARY KEY (`id`);

id	specilization	doctorName	address	docFees	contactno	docEmail	password	creationDate	updateDate
1	Cardiologist	Adesh Mantur	Bijapur	900	9945574163	adeshmantur12@gmail.com	Am@12345	2021-01-01 11:30:00	2021-01-01 11:30:00
2	Audiologist	Aditya Patil	Koppal	800	7892246734	adityapatil23@gmail.com	Ap@12345	2021-01-01 12:30:00	2021-01-01 12:30:00
3	Dermatologist	Anup Kabberalli	Tumkur	900	8494809175	anupkabberhalli90@gmail.com	Ak@12345	2021-01-01 13:30:00	2021-01-01 13:30:00
4	Dentist	Mayur Hegde	Bagalkot	700	7022808747	mayurhegde34@gmail.com	Mh@12345	2021-01-01 14:30:00	2021-01-01 14:30:00
5	Pulmonologist	Pooja Kittur	Mysore	700	7349131435	poojakittur78@gmail.com	Pk@12345	2021-01-01 15:30:00	2021-01-01 15:30:00
6	Oncologist	Priyanka Sonu	Dharwad	800	9480641514	priyankasonu89@gmail.com	Ps@12345	2021-01-01 16:30:00	2021-01-01 16:30:00
8	Gynaecologist	Vaibhav Khashyap	Hubballi	600	6361736455	vaibhavkhashyap56@gmail.com	Vk@12345	2021-01-01 17:30:00	2021-01-01 17:30:00
9	Paediatrician	Venkatesh Mamdapur	Belgaum	600	7204761700	venkateshmamdaour67@gmail.com	Vm@12345	2021-01-01 18:30:00	2021-01-01 18:30:00
10	Obstetrician	Vrushabh Turamari	Bangalore	600	9019818653	vrushabhturamari45@gmail.com	Vt@12345	2021-01-01 19:30:00	2021-01-01 19:30:00

Doctor log

CREATE TABLE `doctorslog` (

`id` int(11) NOT NULL,

`uid` int(11) DEFAULT NULL,

`username` varchar(255) DEFAULT NULL,

`userip` binary(16) DEFAULT NULL,

`loginTime` timestamp NULL DEFAULT current_timestamp(),

`logout` varchar(255) DEFAULT NULL,

`status` int(11) DEFAULT NULL

)

ALTER TABLE `doctorslog`

ADD PRIMARY KEY (`id`);

id	uid	username	userip	loginTime	logout	status
----	-----	----------	--------	-----------	--------	--------

Doctor Specialization

```
CREATE TABLE `doctorspecilization` (  
  `id` int(11) NOT NULL,  
  `specilization` varchar(255) DEFAULT NULL,  
  `creationDate` timestamp NULL DEFAULT current_timestamp(),  
  `updationDate` timestamp NULL DEFAULT NULL ON UPDATE  
  current_timestamp()  
)  
  
INSERT INTO `doctorspecilization` (`id`, `specilization`, `creationDate`,  
  `updationDate`) VALUES  
  
(1, 'Audiologist', '2021-01-01 01:00:00', '2021-01-01 01:00:00'),  
(2, 'Cardiologist', '2021-01-01 02:00:00', '2021-01-01 02:00:00'),  
(3, 'Dentist', '2021-01-01 03:00:00', '2021-01-01 03:00:00'),  
(4, 'Dermatologist', '2021-01-01 04:00:00', '2021-01-01 04:00:00'),  
(5, 'ENT Specialist', '2021-01-01 05:00:00', '2021-01-01 05:00:00'),  
(6, 'General Physician', '2021-01-01 06:00:00', '2021-01-01 06:00:00'),  
(7, 'Genral Surgeon', '2021-01-01 07:00:00', '2021-01-01 07:00:00'),  
(8, 'Gynaecologist', '2021-01-01 08:00:00', '2021-01-01 08:00:00'),  
(9, 'Demo test', '2016-12-28 07:37:39', '0000-00-00 00:00:00'),  
(10, 'Obstetrician', '2021-01-01 09:00:00', '2021-01-01 09:00:00'),  
(11, 'Oncologist', '2021-01-01 10:00:00', '2021-01-01 10:00:00'),  
(12, 'Pulmonologist', '2021-01-01 11:00:00', '2021-01-01 11:00:00');  
  
ALTER TABLE `doctorspecilization`  
  
  ADD PRIMARY KEY (`id`);
```

id	specilization	creationDate	updationDate
1	Audiologist	2021-01-01 06:30:00	2021-01-01 06:30:00
2	Cardiologist	2021-01-01 07:30:00	2021-01-01 07:30:00
3	Dentist	2021-01-01 08:30:00	2021-01-01 08:30:00
4	Dermatologist	2021-01-01 09:30:00	2021-01-01 09:30:00
5	ENT Specialist	2021-01-01 10:30:00	2021-01-01 10:30:00
6	General Physician	2021-01-01 11:30:00	2021-01-01 11:30:00
7	Genral Surgeon	2021-01-01 12:30:00	2021-01-01 12:30:00
8	Gynaecologist	2021-01-01 13:30:00	2021-01-01 13:30:00
9	Demo test	2016-12-28 13:07:39	0000-00-00 00:00:00
10	Obstetrician	2021-01-01 14:30:00	2021-01-01 14:30:00
11	Oncologist	2021-01-01 15:30:00	2021-01-01 15:30:00
12	Pulmonologist	2021-01-01 16:30:00	2021-01-01 16:30:00

Contact us

```
CREATE TABLE `contactus` (
  `id` int(11) NOT NULL,
  `fullname` varchar(255) DEFAULT NULL,
  `email` varchar(255) DEFAULT NULL,
  `contactno` bigint(12) DEFAULT NULL,
  `message` mediumtext DEFAULT NULL,
  `PostingDate` timestamp NULL DEFAULT current_timestamp(),
  `AdminRemark` mediumtext DEFAULT NULL,
  `LastupdationDate` timestamp NULL DEFAULT NULL ON UPDATE
current_timestamp(),
  `IsRead` int(11) DEFAULT NULL
)
```

```
INSERT INTO `contactus` (`id`, `fullname`, `email`, `contactno`, `message`,
`PostingDate`, `AdminRemark`, `LastupdationDate`, `IsRead`) VALUES
```

```
(1, 'Rahika Irkal', 'radhikaairkal12@gmail.com', 7899042235, 'In need of an ENT
specialist, is he/she available', '2021-01-02 10:00:00', 'Available', '2021-01-02
11:00:00', 1),
```


(2, 'Rohit Amberkar', 'rohitamberkar23@gmail.com', 9980072227, 'In need of an Pulmonologist, is he/she available', '2019-06-30 11:00:00', 'Available', '2021-01-02 13:00:00', 1),

(3, 'Gourish Bagalkoti', 'gourishbagalkoti@gmail.com', 8296799747, 'In need of an Oncologist, is he/she available', '2019-11-10 18:53:48', NULL, NULL, NULL);

ALTER TABLE `contactus`

ADD PRIMARY KEY (`id`);

id	fullName	email	contactno	message	PostingDate	AdminRemark	LastupdateDate	IsRead
1	Rahika Irkal	radhikairkal12@gmail.com	7899042235	In need of an ENT specialist, is he/she available	2021-01-02 15:30:00	Available	2021-01-02 16:30:00	1
2	Rohit Amberkar	rohitamberkar23@gmail.com	9980072227	In need of an Pulmonologist, is he/she available	2019-06-30 16:30:00	Available	2021-01-02 18:30:00	1
3	Gourish Bagalkoti	gourishbagalkoti@gmail.com	8296799747	In need of an Oncologist, is he/she available	2019-11-11 00:23:48	NULL	NULL	NULL

Medical history

CREATE TABLE `tblmedicalhistory` (

`ID` int(10) NOT NULL,

`PatientID` int(10) DEFAULT NULL,

`BloodPressure` varchar(200) DEFAULT NULL,

`BloodSugar` varchar(200) NOT NULL,

`Weight` varchar(100) DEFAULT NULL,

`Temperature` varchar(200) DEFAULT NULL,

`MedicalPres` mediumtext DEFAULT NULL,

`CreationDate` timestamp NOT NULL DEFAULT current_timestamp() ON UPDATE current_timestamp()

)

INSERT INTO `tblmedicalhistory` (`ID`, `PatientID`, `BloodPressure`, `BloodSugar`, `Weight`, `Temperature`, `MedicalPres`, `CreationDate`) VALUES

(2, 1, '120/185', '80/120', '61 Kg', '101 deg', NULL, '2021-01-02 15:00:00'),

(3, 2, '90/120', '92/190', '76 kg', '99 deg', NULL, '2021-01-02 16:00:00'),

(4, 3, '125/200', '86/120', '60 kg', '98 deg', NULL, '2021-01-02 16:00:00'),

(5, 4, '96/120', '98/120', '66 kg', '102 deg', NULL, '2021-01-02 17:00:00'),

(6, 5, '90/120', '97/120', '67 kg', '98 deg', NULL, '2021-01-02 18:00:00'),

```
(7, 6, '80/120', '100/120', '65 kg', '98.6 deg', NULL, '2021-01-02 19:00:00');
```

```
ALTER TABLE `tblmedicalhistory`
```

```
ADD PRIMARY KEY (`ID`);
```

ID	PatientID	BloodPressure	BloodSugar	Weight	Temperature	MedicalPres	CreationDate
2	1	120/185	80/120	61 Kg	101 deg	NULL	2021-01-02 20:30:00
3	2	90/120	92/190	76 kg	99 deg	NULL	2021-01-02 21:30:00
4	3	125/200	86/120	60 kg	98 deg	NULL	2021-01-02 21:30:00
5	4	96/120	98/120	66 kg	102 deg	NULL	2021-01-02 22:30:00
6	5	90/120	97/120	67 kg	98 deg	NULL	2021-01-02 23:30:00
7	6	80/120	100/120	65 kg	98.6 deg	NULL	2021-01-03 00:30:00

Patients

```
CREATE TABLE `tblpatient` (
```

```
`ID` int(10) NOT NULL,
```

```
`Docid` int(10) DEFAULT NULL,
```

```
`PatientName` varchar(200) DEFAULT NULL,
```

```
`PatientContno` bigint(10) DEFAULT NULL,
```

```
`PatientEmail` varchar(200) DEFAULT NULL,
```

```
`PatientGender` varchar(50) DEFAULT NULL,
```

```
`PatientAdd` mediumtext DEFAULT NULL,
```

```
`PatientAge` int(10) DEFAULT NULL,
```

```
`PatientMedhis` mediumtext DEFAULT NULL,
```

```
`CreationDate` timestamp NULL DEFAULT current_timestamp(),
```

```
`UpdationDate` timestamp NULL DEFAULT NULL ON UPDATE  
current_timestamp()
```

```
)
```

```
INSERT INTO `tblpatient` (`ID`, `Docid`, `PatientName`, `PatientContno`,  
`PatientEmail`, `PatientGender`, `PatientAdd`, `PatientAge`, `PatientMedhis`,  
`CreationDate`, `UpdationDate`) VALUES
```

```
(1, 1, 'Priyanka Priya', 1234567890, 'test@gmail.com', 'Female', 'CSE KLEIT Hubballi',  
26, 'She is diabetic patient', '2021-01-03 12:00:00', '2021-01-03 12:00:00'),
```

(2, 2, 'Amit Hegde', 7899537060, 'amithegde12@gmail.com', 'Male', 'CSE KLEIT Hubballi', 20, 'He is Partially deaf', '2021-01-03 13:00:00', '2021-01-03 13:00:00'),

(3, 3, 'Ayush Saraf', 8310678754, 'ayushsaraf23@gmail.com', 'Male', 'CSE KLEIT Hubballi', 21, 'Has multiple problems', '2021-01-03 14:00:00', '2021-01-03 14:00:00'),

(4, 4, 'Tushar V', 9880225683, 'tushark34@gmail.com', 'Male', 'CSE KLEIT Hubballi', 20, 'Has gum issues', '2021-01-03 15:00:00', '2021-01-03 15:00:00'),

(5, 5, 'Vibhav Nisarikar', 8884467152, 'vibhavnisarikar45@gmail.com', 'Male', 'CSE KLEIT Hubballi', 21, 'Has pulmonary artery blockage', '2021-01-03 16:00:00', '2021-01-03 16:00:00'),

(6, 6, 'Vishal Kaliwal', 7337627302, 'vishalkaliwal56@gmail.com', 'Male', 'CSE KLEIT Hubballi', 20, 'Has to be looked by an Oncologist', '2021-01-03 17:00:00', '2021-01-03 17:00:00');

```
ALTER TABLE `tblpatient`
```

```
ADD PRIMARY KEY (`ID`);
```

ID	Docid	PatientName	PatientContno	PatientEmail	PatientGender	PatientAdd	PatientAge	PatientMedhis	CreationDate	UpdationDate
2	2	Amit Hegde	7899537060	amithegde12@gmail.com	Male	CSE KLEIT Hubballi	20	He is Partially deaf	2021-01-03 18:30:00	2021-01-03 18:30:00
3	3	Ayush Saraf	8310678754	ayushsaraf23@gmail.com	Male	CSE KLEIT Hubballi	21	Has multiple problems	2021-01-03 19:30:00	2021-01-03 19:30:00
4	4	Tushar V	9880225683	tushark34@gmail.com	Male	CSE KLEIT Hubballi	20	Has gum issues	2021-01-03 20:30:00	2021-01-03 20:30:00
5	5	Vibhav Nisarikar	8884467152	vibhavnisarikar45@gmail.com	Male	CSE KLEIT Hubballi	21	Has pulmonary artery blockage	2021-01-03 21:30:00	2021-01-03 21:30:00
6	6	Vishal Kaliwal	7337627302	vishalkaliwal56@gmail.com	Male	CSE KLEIT Hubballi	20	Has to be looked by an Oncologist	2021-01-03 22:30:00	2021-01-03 22:30:00

User log

```
CREATE TABLE `userlog` (
```

```
`id` int(11) NOT NULL,
```

```
`uid` int(11) DEFAULT NULL,
```

```
`username` varchar(255) DEFAULT NULL,
```

```
`userip` binary(16) DEFAULT NULL,
```

```
`loginTime` timestamp NULL DEFAULT current_timestamp(),
```

```
`logout` varchar(255) DEFAULT NULL,
```

```
`status` int(11) DEFAULT NULL
```

```
)
```

```
ALTER TABLE `userlog`
```

```
ADD PRIMARY KEY (`id`);
```

id	uid	username	userip	loginTime	logout	status
----	-----	----------	--------	-----------	--------	--------

Users

```
CREATE TABLE `users` (
  `id` int(11) NOT NULL,
  `fullName` varchar(255) DEFAULT NULL,
  `address` longtext DEFAULT NULL,
  `city` varchar(255) DEFAULT NULL,
  `gender` varchar(255) DEFAULT NULL,
  `email` varchar(255) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  `regDate` timestamp NULL DEFAULT current_timestamp(),
  `updateDate` timestamp NULL DEFAULT NULL ON UPDATE
current_timestamp()
)
```

```
INSERT INTO `users` (`id`, `fullName`, `address`, `city`, `gender`, `email`,
`password`, `regDate`, `updateDate`) VALUES
```

```
(1, 'Priyanka Priya', 'CSE KLEIT Hubballi', 'Hubballi', 'Female', 'test@gmail.com',
'f925916e2754e5e03f75dd58a5733251', '2021-01-03 15:00:00', '0000-00-00 00:00:00'),
```

```
(2, 'Ankita Bangalore', 'CSE KLEIT Hubballi', 'Hubballi', 'Female',
'ankitabangalore01@gmail.com', 'Ab@12345', '2021-01-03 16:00:00', '2021-01-03
16:00:00'),
```

```
(3, 'Amit Hegde', 'CSE KLEIT Hubballi', 'Hubballi', 'Male', 'amithegde12@gmail.com',
'Ah@12345', '2021-01-03 17:00:00', '2021-01-03 17:00:00'),
```

```
(4, 'Tushar V', 'CSE KLEIT Hubballi', 'Hubballi', 'Male', 'tusharv34@gmail.com',
'Tv@12345', '2021-01-03 19:00:00', '2021-01-03 19:00:00'),
```

```
(5, 'Vibhav Nisarika', 'CSE KLEIT Hubballi', 'Hubballi', 'Male',
'vibhavnisarika45@gmail.com', 'Vn@12345', '2021-01-03 20:00:00', '2021-01-03
20:00:00'),
```

```
(6, 'Vishal Kaliwal', 'CSE KLEIT Hubballi', 'Hubballi', 'Male',
'vishalkaliwal56@gmail.com', 'Vk@12345', '2021-01-03 21:00:00', '2021-01-03
21:00:00');
```

```
ALTER TABLE `users`
```

```
ADD PRIMARY KEY (`id`),
```

id	fullName	address	city	gender	email	password	regDate	updateDate
2	Ankita Banglore	CSE KLEIT Hubballi	Hubballi	Female	ankitabanglore01@gmail.com	Ab@12345	2021-01-03 21:30:00	2021-01-03 21:30:00
3	Amit Hegde	CSE KLEIT Hubballi	Hubballi	Male	amithegde12@gmail.com	Ah@12345	2021-01-03 22:30:00	2021-01-03 22:30:00
4	Tushar V	CSE KLEIT Hubballi	Hubballi	Male	tusharv34@gmail.com	Tv@12345	2021-01-04 00:30:00	2021-01-04 00:30:00
5	Vibhav Nisarikar	CSE KLEIT Hubballi	Hubballi	Male	vibhavnisarikar45@gmail.com	Vn@12345	2021-01-04 01:30:00	2021-01-04 01:30:00
6	Vishal Kaliwal	CSE KLEIT Hubballi	Hubballi	Male	vishalkaliwal56@gmail.com	Vk@12345	2021-01-04 02:30:00	2021-01-04 02:30:00

CHAPTER 6

TESTING AND VALIDATION

6.1 Use case diagram

Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a "system" is something being developed or operated, such as a web site. The "actors" are people or entities operating under defined roles within the system.

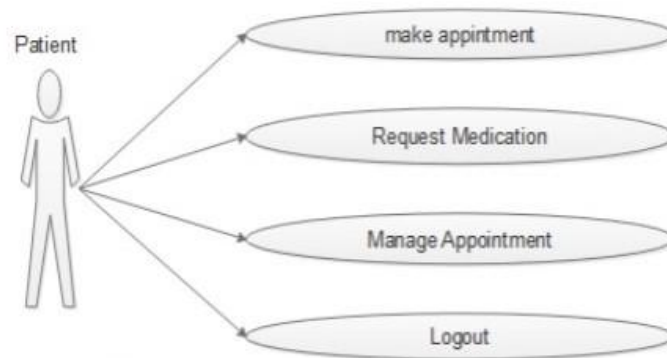


Fig 6.1. Use case diagram for Patient

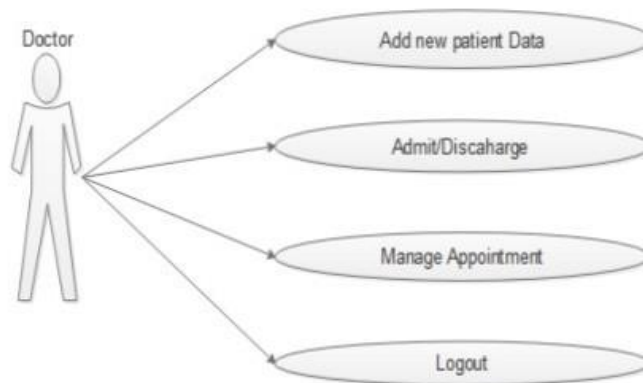


Fig 6.2 Use case diagram for Doctor

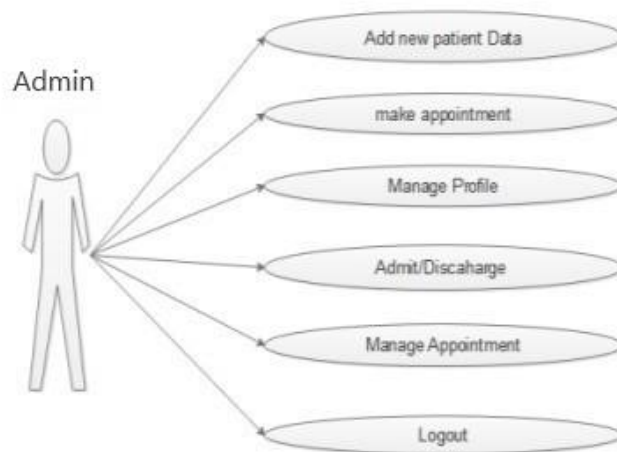


Fig 6.3 Use case diagram for Admin

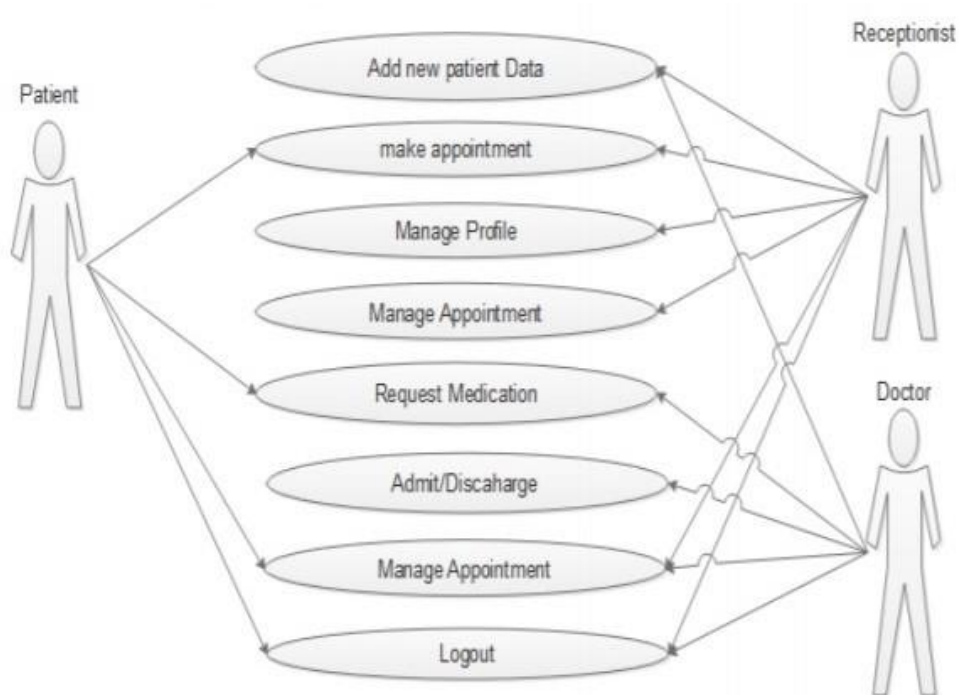


Fig 6.4. Use case diagram for Hospital Management System

6.2 Sequence diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams.

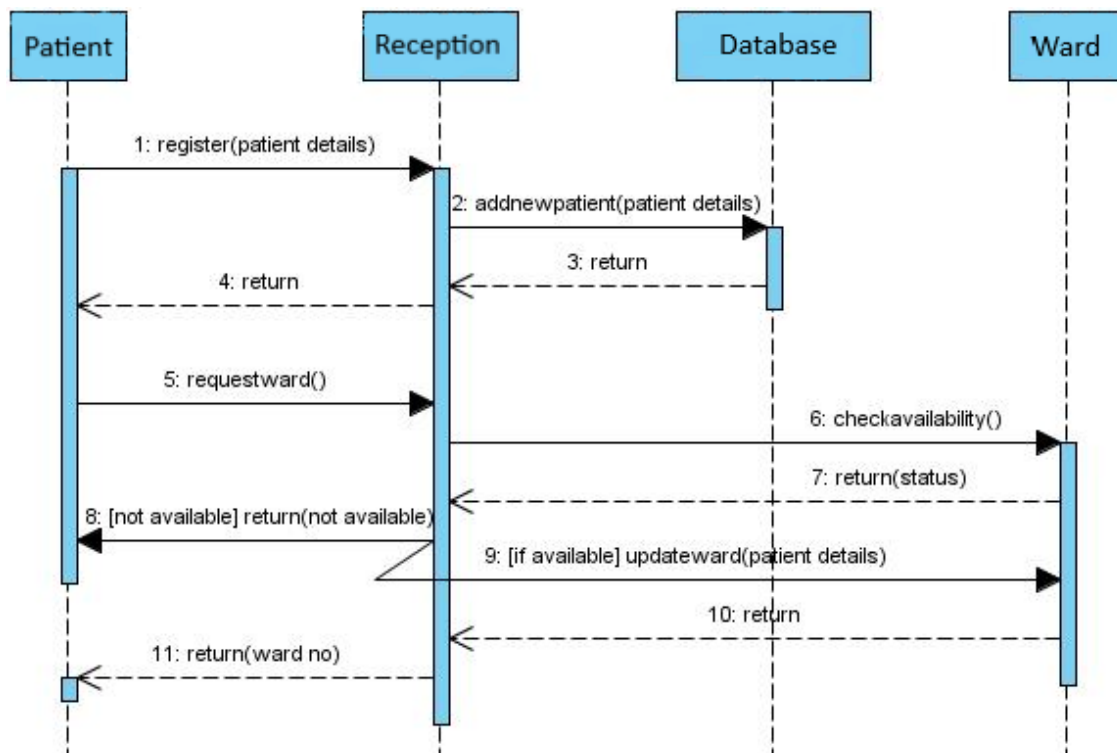


Fig 6.5. Sequence diagram for Hospital Management System

6.3 Result

The patient registers himself and logs into his account. He requests an appointment from the doctor with a particular specialty at desired date and time. The doctor logs in and checks if there are any appointment requests. If the patient wishes to, he can also cancel the appointment. If the appointment is confirmed he can visit the doctor on the approved time and date. The entire system is managed by the admin.

6.4 Summary

In this chapter we showcased the testing and validation process with use case diagram and Sequence diagram and with result obtained for the proposed project.

CHAPTER 7

ADVANTAGES AND DISADVANTAGES

7.1 Advantages

- This system helps to reduce the waiting time of the patient.
- User can select the appointment time according to his/her preference.
- The patient can view the doctors list on his requirement criteria; hence he can choose best for himself.
- The application is user friendly.
- Our system doesn't allow any person just to register himself as a doctor, as admin himself adds a new doctor.

7.2 Disadvantages.

- The system requires large database to store all doctors and patient information as well as the appointment.
- If patient requests for an appointment and if he cancels it then it will be a loss for the doctor.
- If the doctor does not see his appointment status whole day then the patient who has requested for that doctor will not come to know that the doctor is busy, and he will suffer.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

The main objective of developing the system is that the online appointment is getting popular day by day, as it empowers patient to make the appointment to the doctor, clinic or hospital. It makes a lot easier for the patient to just to go to their computer, access a website and select a doctor they wish and make an appointment then go to the hospital or clinic and visit the doctor as per the scheduled appointment instead of waiting in a line just to take an appointment for hours.

The patients can also involve in the health decisions that they have to make. They can also make an appointment to another doctor other than theirs, by nothing more than a click. The best part of it is that the patient can do this 24/7 from the comforts of their home.

8.2 Future Scope

There are also few features which can be intimated with the system to make it more flexible. Below list shows the future points that can be considered.

- Directly getting the images for city scan or x-ray from the connected devices.
- The patient record can be maintained along with the diagnostic reports.
- Billing of patients
- Blood bank information management.
- Producing ECG using connected device.
- Video Conferencing facility for remote areas for treatment.
- The listing of doctors who are nearer to the patient's location.

CHAPTER 9

BIBLIOGRAPHY

- [1] Deepak Thomas “Beginning PHP 4 Databases”, Wrox Press Ltd. Paperback-17,
- [2] Matt Doyle, “Beginning PHP 5.3, 2nd edition”
- [3] Luke Welling, Laura Thomson. Sams PHP and MySQL Web Development, 2nd edition
- [4] W. Jason Gilmore “Beginning PHP 5 and MySQL 5 from Novice to Professional SECOND EDITION”
- [5] Abraham Silberschatz, Henry F. Korth and S. Sudarshan “Sixth Edition Database System Concepts released”

WEB REFERENCES

- [1] [Server-Side Scripting](http://php.net/manual/en/index.php) <http://php.net/manual/en/index.php>
- [2] [HTML & CSS](https://www.w3schools.com/) <https://www.w3schools.com/>
- [3] [Bootstrap](http://getbootstrap.com/) <http://getbootstrap.com/>
- [4] <https://stackoverflow.com/>