CHAPTER 1

INTRODUCTION

HOSPITAL MANAGEMENT SYSTEM is an application used to register and control the details of doctor, receptionist and patient. It is a platform that provides the receptionist to manage and maintain the details of patient . It also allows the doctor to give the prescription for the patient and admin have access to every function.

1.1 Purpose

The development of this system contains the following activities, which try to automate the entire process keeping in the view of database integration approach. This system maintains the doctor, receptionist and patient information along with their details . This system is user friendly which provides application with various controls provided by system user interface. Authentication is provided for this application only registered users that is doctor and receptionist and admin can access. Doctor , receptionist and patient details is stored in centralized database which can be maintained by the system. This system allows the receptionist to manage the patient record systematically.

1.2 Scope

The objective of this application is to develop a system that effectively manages all the data related to the hospital. The purpose is to maintain a centralized database of all doctor. Receptionist and patient related information. The goal is to support various functions and processes necessary to manage the data efficiently.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

Software Requirement Specification specifies the requirements required to run the given desktop application. The detailed explanation of each type of requirement is given below.

2.1 Functional Requirement

- The admin can login to the system using the username and password.
- Admin have access to every functionality of the system.
- Receptionist should be able to login using his username and password.
- Receptionist have the access to add new patient details ,view patient details, update or delete patient details and full history of the patient.
- Doctor should be able to login using his username and password.
- Doctor have the access to give diagnosis to the patient and view the full history of the patient.
- Admin have the access to add new department , add new doctor, add new receptionist and add new patient details.
- Admin have the access to update or delete department details, doctor details, receptionist details.

2.2 Hardware Requirement Specification

- PROCESSOR: Intel®core™ 2 Duo
- SPEED: 2.10GHz
- RAM: 2.00 GB Minimum
- SPACE ON DISC:20GB Minimum

2.3 Software Requirement Specification

- Apache NetBeans IDE 12.0
- Xampp
- MySql
- Programming Languages: Java

2.3.1 XAMPP

XAMPP is a free and open source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, Maria DB database, and interpreters for scripts written in the PHP and Perl programming languages. XAMPP stands for Cross-Platform (X), Apache (A), Maria DB (M), PHP (P) and Perl (P). It is a simple, lightweight Apache distribution that makes it extremely easy for developers to create a local web server for testing and deployment purposes. Everything needed to set up a web server – server application (Apache), database (Maria DB), and scripting language (PHP) – is included in an extractable file. XAMPP is also cross-platform, which means it works equally well on Linux, Mac and Windows. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server extremely easy.

2.3.2 MYSQL

MySQL is an Oracle-backed open source relational database management system based on the Structured Query language (SQL). MySQL runs on virtually all platforms , including Linux , Unix and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and online publishing. MySQL is an important component of an open source enterprise stack called LAMP.

2.3.2 APACHE NETBEANS IDE 21.0

NetBeans is an integrated development environment for Java. NetBeans allows applications to be developed from a set of modular software components called modules. NetBeans runs on Windows, macOS, Linux and Solaris. NetBeans IDE lets you quickly and easily develop Java desktop, mobile, and web applications, as well as HTML5 applications with HTML, JavaScript, and CSS. The IDE also provides a great set of tools for PHP and C/C++ developers.

CHAPTER 3

SYSTEM DESIGN

Software design is the process by which an agent creates specification of software artifact, intended to accomplish goals, using the set of primitive components and subject to constraints.

3.1 ER Model

An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS and entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. The Figure 3.1 shows the ER diagram of the Hospital Management System.

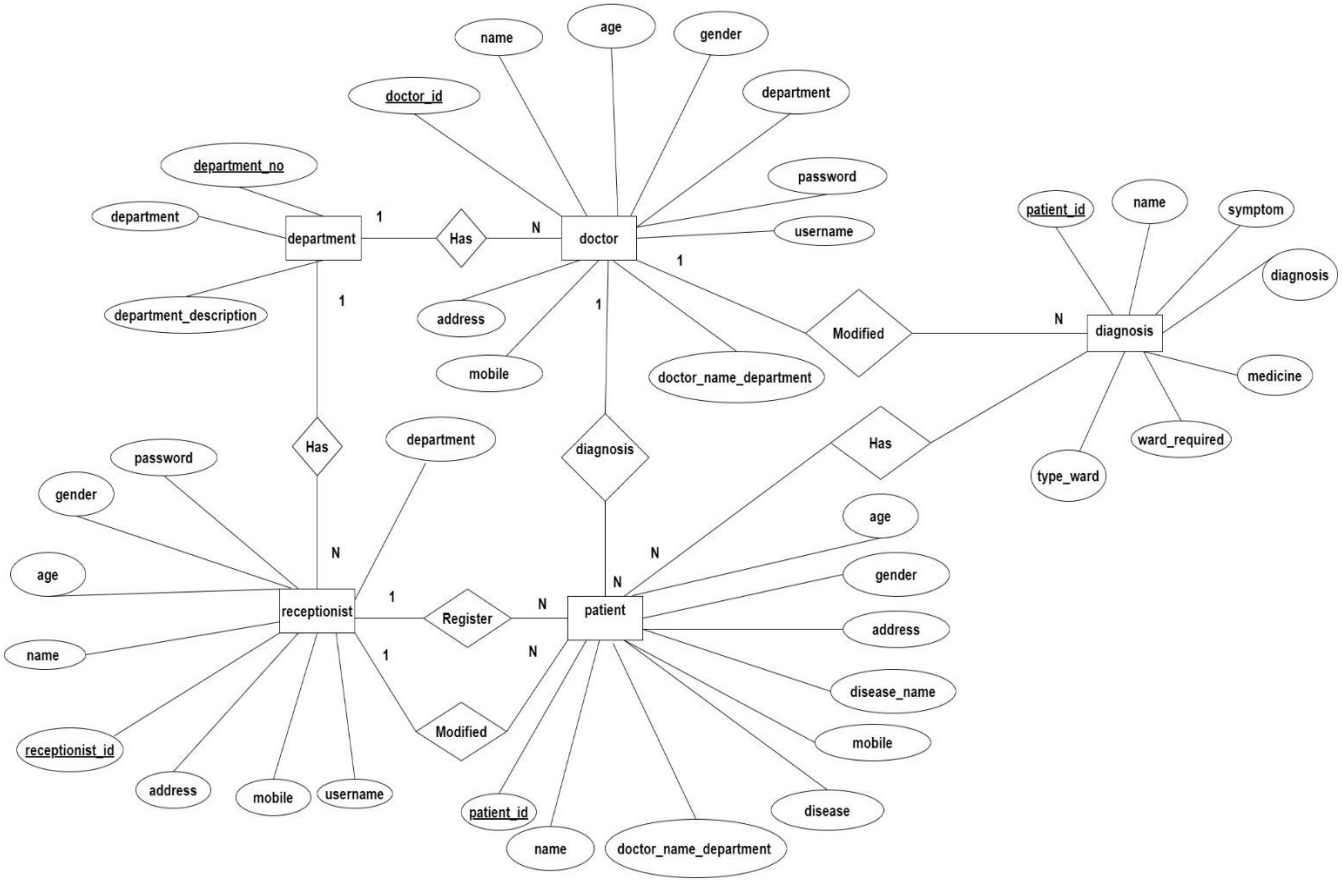

Figure 3.1: ER diagram for Hospital Management System.

3.2 Schema Diagram

A Database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated .

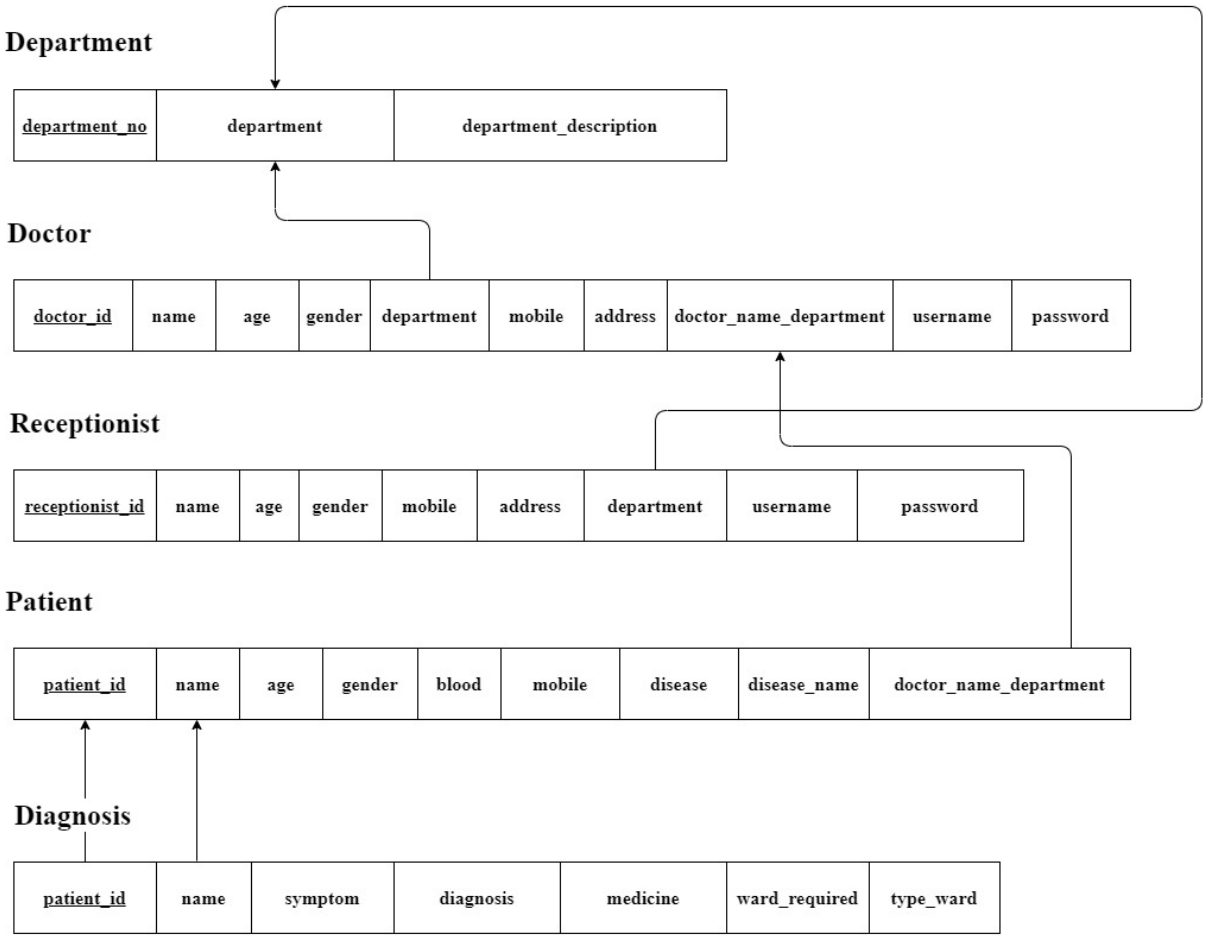

Figure 3.2: Schema diagram for Hospital Management System.

The above Figure 3.2 shows the database schema that explains relationships of different tables in the database. The tables Department table and Doctor table are interconnected through 'department' as a linking attribute. The tables Department table and Receptionist table are interconnected through 'department' as a linking attribute. The tables Doctor table and Patient table are interconnected through 'doctor_name_department' as a linking attribute. The tables Patient table and Diagnosis table are interconnected through 'patient_id & name.' as a linking attribute. In table Department 'department_no' is the primary key. In table Doctor 'doctor_id' is the primary key. In table Receptionist 'receptionist_id' is the primary key. In table Patient 'patient_id' is the primary key. In the table Diagnosis 'Patient_id' is the primary key.

3.3 Table Description

A table is a named original data base data set that is organized by rows and columns. The relational table is a fundamental relational data base concept because tables are the primary form of a data storage. Columns form the table's structure and rows form the content.

In Table 3.4 Department, department_no, department, department_description are the three attributes where department_no is the unique no for the department of varchar datatype where it is a primary key and department is the unique name for the department of varchar data type. department_description datatype is text.

Table 3.1: Department

Attributes	Data type	Constraints	Description
department_no	varchar	Primary key	Unique no for the department
department	varchar		Unique name for the department
department_description	text		Discription about the department

In Table 3.5 Doctor_Details, doctor_Id, name, age, gender, department, mobile, address, doctor_name_department, username, password are ten attributes. Where doctor_Id, name, gender, department, mobile, doctor_name_department, username, password are of datatype varchar and age is of datatype int and address is of datatype text. Where doctor _id is the primary key.

Table 3.2: Doctor_Details

Attributes	Data type	Constraints	Description
doctor_id	varchar	Primary key	Unique id for the doctor
name	varchar		Name of the doctor
age	int		Age of the doctor
gender	varchar		Gender of the doctor
department	varchar		Department of the doctor
mobile	varchar		Mobile no of the doctor
address	text		Address of the doctor
doctor_name_department	varchar		Name and department of the doctor

username	varchar		User name of the doctor
password	varchar		Password of the doctor

In Table 3.6 Receptionist_Details, receptionist_Id, name, age, gender, mobile, address, username, password, department are the nine attributes. Where receptionist_Id, name, gender, mobile, username, password, department are of datatype varchar and age is of datatype int and address is of datatype text. Where receptionist_id is primary key.

Table 3.3: Receptionist_Details

Attributes	Data type	Constraints	Description
receptionist_id	varchar	Primary key	Unique id for the receptionist
name	varchar		Name of the receptionist
age	int		Age of the receptionist
gender	varchar		Gender of the receptionist
mobile	varchar		Mobile no of the receptionist
address	text		Address of the receptionist
username	varchar		Username of the receptionist
password	varchar		Password of the receptionist
department	varchar		Department of the receptionist

In Table 3.7 Patient_Details, patient_id, name, age, gender, blood, mobile, disease, disease_name,doctor_name_department are the nine attributes. Department_Id, name, gender, blood, mobile, disease, disease_name, disease_name_department are of datatype varchar and age is of datatype int.Where patient_id is primary key.

Table 3.4: Patient_Details

Attributes	Data type	Constraints	Description
patient_id	varchar	Primary key	Unique id for the patient

name	varchar		Name of the patient
age	int		Age of the patient
gender	varchar		Gender of the patient
blood	varchar		Blood group of the patient
mobile	varchar		Mobile no of the patient
disease	varchar		Disease of the patient
disease_name	varchar		Disease of the patient
disease_name_department	varchar		Name and department of the patient

In Table 3.8 Diagnosis , patient_id, name, symptom, diagnosis, medicine, ward_required, type_ward are the seven attributes. patient_id, name, symptom, diagnosis, medicine, ward_required, type_ward are of datatype varchar. where patient_id is primary key.

Table3.5: Diagnosis

Attributes	Data type	Constraints	Description
patient_id	varchar	Primary key	Unique id given for the patient
name	varchar		Name of the patient
Symptom	varchar		Symptom of the patient
diagnosis	varchar		Diagnosis of the patient
medicine	varchar		Medicne of the patient
ward_required	varchar		Ward required for the patient
type_ward	varchar		Which type of ward required for the patient general/single/duo

CHAPTER 4

IMPLEMENTATION

Implementation is defined as specific set of activities designed to put into practice an activity or program of known dimensions. Implementation processes are purposeful and are described in sufficient details such that independent can detect the presence and strength of the "specific set of activities" related to implementation.

4.1 Details of the Language

Our project is implemented using Java programming. The reason we chose Java is because it is simple to comprehend than other programming languages. Since it is an easy and powerful language, it has been widely used for creating Desktop-based applications that requires utmost functionality with minimal coding. In addition, Java. Net based web applications are extremely secure as compared to applications of various other programming languages.

4.1.1 Java

Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers.

Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform.

The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java

technologies under the GNU General Public License. Oracle offers its own HotSpot Java Virtual Machine, however the official reference implementation is the OpenJDK JVM which is free open source software and used by most developers and is the default JVM for almost all Linux distribution.

4.1.2 SQL

SQL (Structured Query Language) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, i.e., data incorporating relations among entities and variables. SQL offers two main advantages over older read–write APIs such as ISAM or VSAM. Firstly, it introduced the concept of accessing many records with one single command. Secondly, it eliminates the need to specify how to reach a record, e.g., with or without an index. Originally based upon relational algebra and tuple relational calculus, SQL consists of many types of statements,[8] which may be informally classed as sublanguages, commonly: a data query language (DQL), a data definition language (DDL), a data control language (DCL), and a data manipulation language (DML). The scope of SQL includes data query, data manipulation (insert, update and delete), data definition (schema creation and modification), and data access control. Although SQL is essentially a declarative language (4GL), it also includes procedural elements.

SQL was one of the first commercial languages to utilize Edgar F. Codd's relational model. The model was described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks". Despite not entirely adhering to the relational model as described by Codd, it became the most widely used database language.

SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standardization (ISO) in 1987. Since then, the standard has been revised to include a larger set of features. Despite the existence of standards, most SQL code requires at least some changes before being ported to different database system.

CHAPTER 5

SCREENSHOTS

The following screenshots includes database table structures and front-end view of a Hospital Management System. The database used for Hospital Management System is XAMPP-MySQL. The below table contains five tables used in Hospital Management System.

5.1 Screenshots of Table Created

The overall database contains mainly five tables as shown it the figure namely Department, Doctor_Details, Receptionist_Details, Patient_Details, Diagnosis. The description of each table is given below.

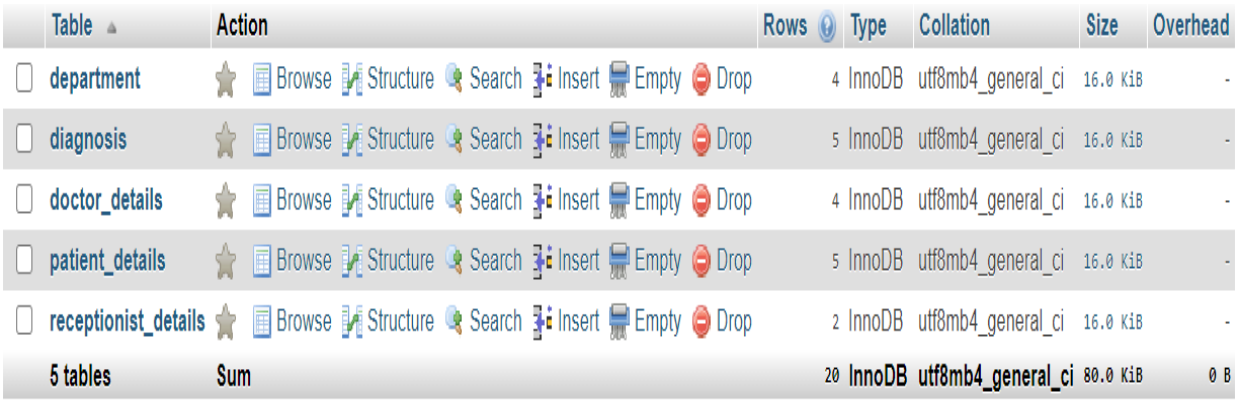

Table	Action	Rows	Type	Collation	Size	Overhead
department	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
diagnosis	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
doctor_details	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
patient_details	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8mb4_general_ci	16.0 KiB	-
receptionist_details	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
5 tables	Sum	20	InnoDB	utf8mb4_general_ci	80.0 KiB	0 B

Figure 5.1: Database table

In Figure 5.2 Department, department_no, department, department_description are the three attributes where department_no is the unique no for the department of varchar datatype where it is a primary key and department is the unique name for the department of varchar data type. department_description datatype is text.

department_no 1	department	department_description
DP01	ORTHO	orthopedics is a medical specialty that focuses on...
DP02	CARDIOLOGY	cardiology is the medical speciality dealing with ...
DP03	PEDIATRICS	pediatrics is the branch of medicine that involves...
DP04	NEUROLOGY	a medical specialty dealing with disorders of the ...

Figure 5.2: Department table

In Figure 5.3 Doctor_Details, doctor_Id, name, age, gender, department, mobile, address, doctor_name_department, username, password are ten attributes. Where doctor_Id, name,

gender, department, mobile, doctor_name_department, username, password are of datatype varchar and age is of datatype int and address is of datatype text. Where doctor _id is the primary key.

doctor_id ▲ 1	name	age	gender	department	mobile	address	doctor_name_department	username	password
D0001	AMAL	25	MALE	ORTHO	7510592019	Analiparayil(h),Chamathachal(po),Kannur	Dr Amal-ortho	amal	1234
D0002	ADON	28	MALE	CARDIOLOGY	8547211474	Nirappel(h),Chamathachal(po),Kannur	Dr Adon-cardiology	adon	1234
D0003	ANU	30	FEMALE	PEDIATRICS	8281555703	Mariyil(h),Chamathachal(po),Kochi	Dr Anu-Pediatrics	anu	1234
D0004	JIS	35	MALE	NEUROLOGY	8590951000	Kalayil(h),Payyavoor(po),Kannur	Dr Jis-Neurology	jis	1234

Figure 5.3: Doctor_Details table

In Figure 5.4 Receptionist_Details, receptionist_Id, name, age, gender, mobile, address, username, password, department are the nine attributes. Where receptionist_Id, name, gender, mobile, username, password, department are of datatype varchar and age is of datatype int and address is of datatype text. Where receptionist_id is primary key.

receptionist_id	name	age	gender	mobile	address	username	password ▼ 1	department
R0001	ANUPAMA	27	MALE	9562543115	Cheruvil(h),Thaliparamba(po),Kasargod	anupama	1234	ORTHO
R0002	JOSE	24	MALE	6734567890	Tharayil(h),Ulikkal(po),Kottayam	jose	1234	CARDIOLOGY

Figure 5.4: Receptionist_Details table

In Figure 5.5 Patient_Details, patient_id, name, age, gender, blood, mobile, disease, disease_name,doctor_name_department are the nine attributes. patient_id, name, gender, blood, mobile, disease, disease_name, disease_name_department are of datatype varchar and age is of datatype int. Where patient_id is primary key.

patient_id ▲ 1	name	age	gender	blood	mobile	disease	disease_name	doctor_name_department
P0001	RAJAN	50	MALE	O+	8765432109	No		Dr Amal-ortho
P0002	MEENA	70	FEMALE	A+	7689876543	Yes	Heart attack	Dr Adon-cardiology
P0003	JOSEPH	65	FEMALE	AB-	6578905674	No		Dr Jis-Neurology
P0004	AKHIL	8	MALE	O+	7869504321	No		Dr Anu-Pediatrics
P0005	LEELA	80	FEMALE	AB-	6789567890	No		Dr Amal-ortho

Figure 5.5: Patient_Details table

In Figure 5.6 Diagnosis , patient_id, name, symptom, diagnosis, medicine, ward_required, type_ward are the seven attributes. patient_id, name, symptom, diagnosis, medicine, ward_required, type_ward are of datatype varchar. where patient_id is primary key.

patient_id	symptom	name	diagnosis	medicine	ward_required	type_ward
P0001	Joint Pain	RAJAN	X-ray	Aspirin	NO	
P0002	Heart Pain	MEENA	ECG	Beta Blockers	Yes	General
P0003	Loss Of Balance	JOSEPH	CT Scan	Sodium Valproate	Yes	Single
P0004	Cough	AKHIL	X-ray	Antibiotic	NO	
P0005	Shoulder Pain	LEELA	X-ray	Aspirin	NO	

Figure 5.6: Diagnosis table

5.2 Front-end Screenshots

Front End was done using Java. The screenshots of the font end is given below.

The Figure 5.7 shows the front end of the Login page.

Figure 5.7: Front End Of Hospital Management System Login Page.

The Figure 5.8 shows the front end of Hospital Management System Admin Login. Where the admin can logging in to the admin panel.

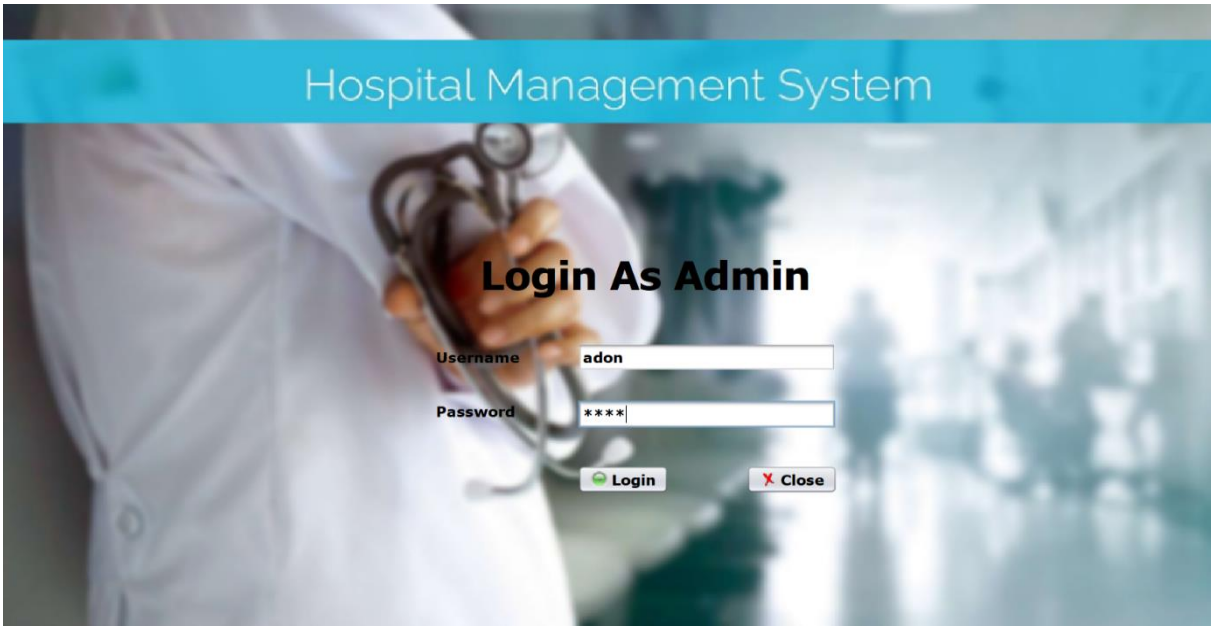

Figure 5.8: Front End Of Hospital Management System Admin Login.

The figure 5.9 shows the dashboard at the front end of the Hospital Management System where the admin can view all the functionality.

Figure 5.9: Front End Of Admin Panel.

The Figure 5.10 shows the the front end of the departmrnt details where admin can update or delete department details.

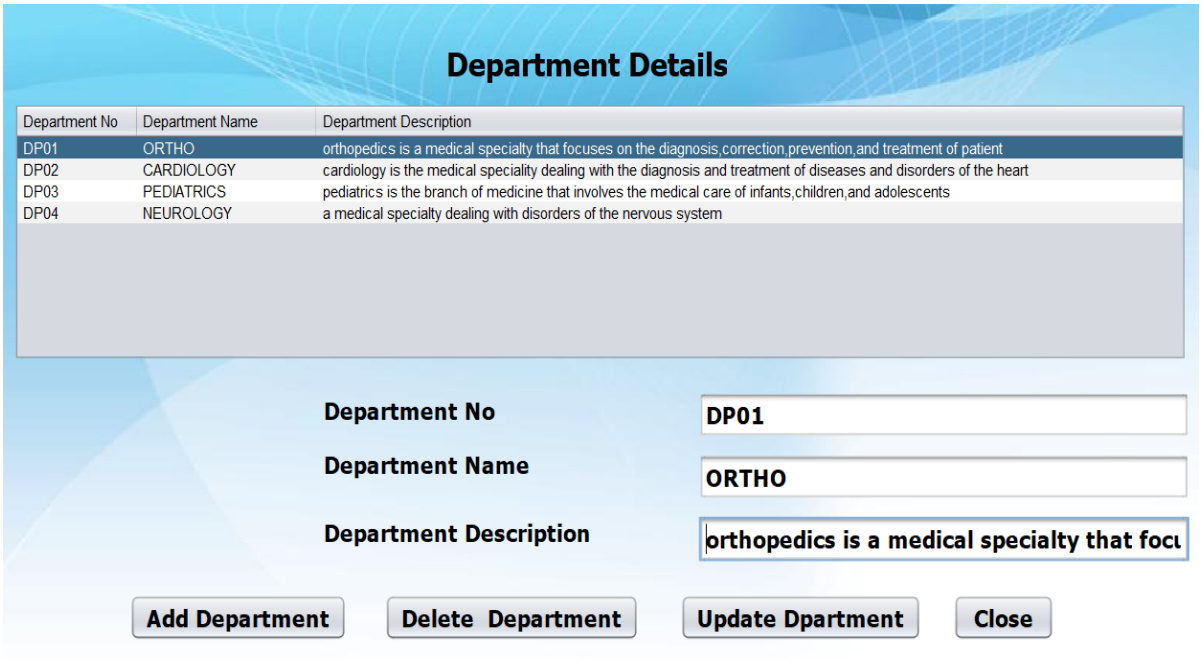

Figure 5.10: Front End Of Department Details Of Hospital Management System.

The Figure 5.11 shows the front end of the doctor registration page of Hospital Management System.

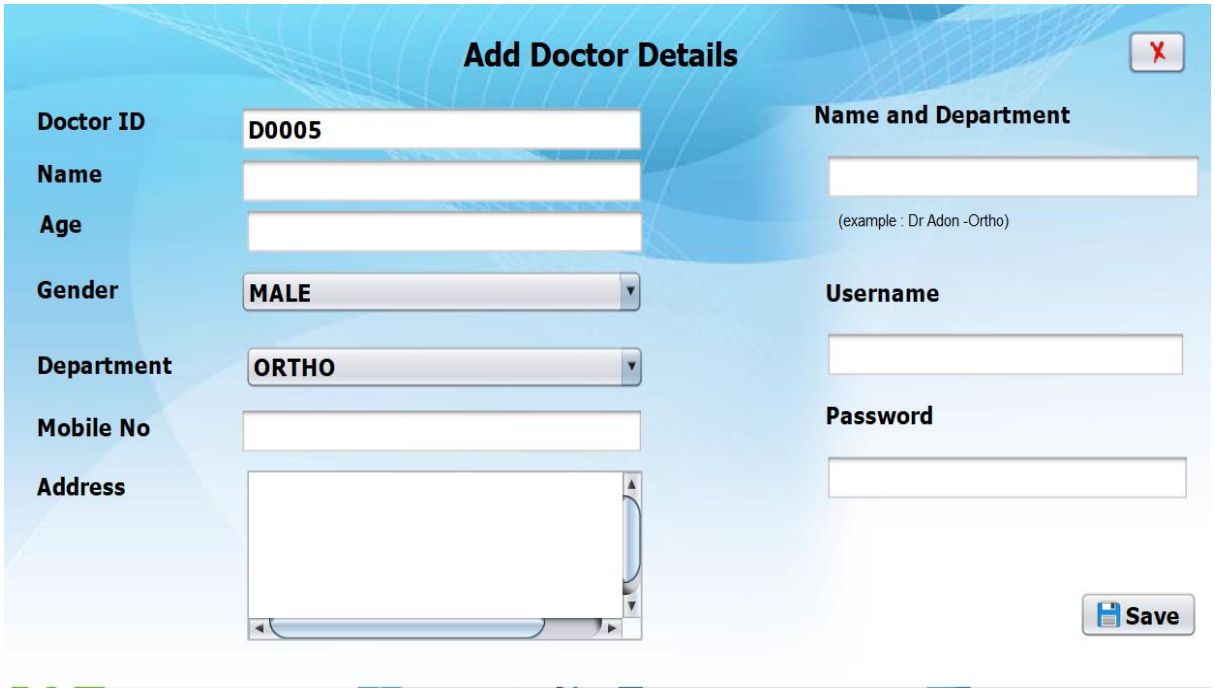

Figure 5.11: Front End Of Doctor Registration Page Of Hospital Management System.

The Figure 5.12 shows the front end of the doctor details of Hospital Management System.

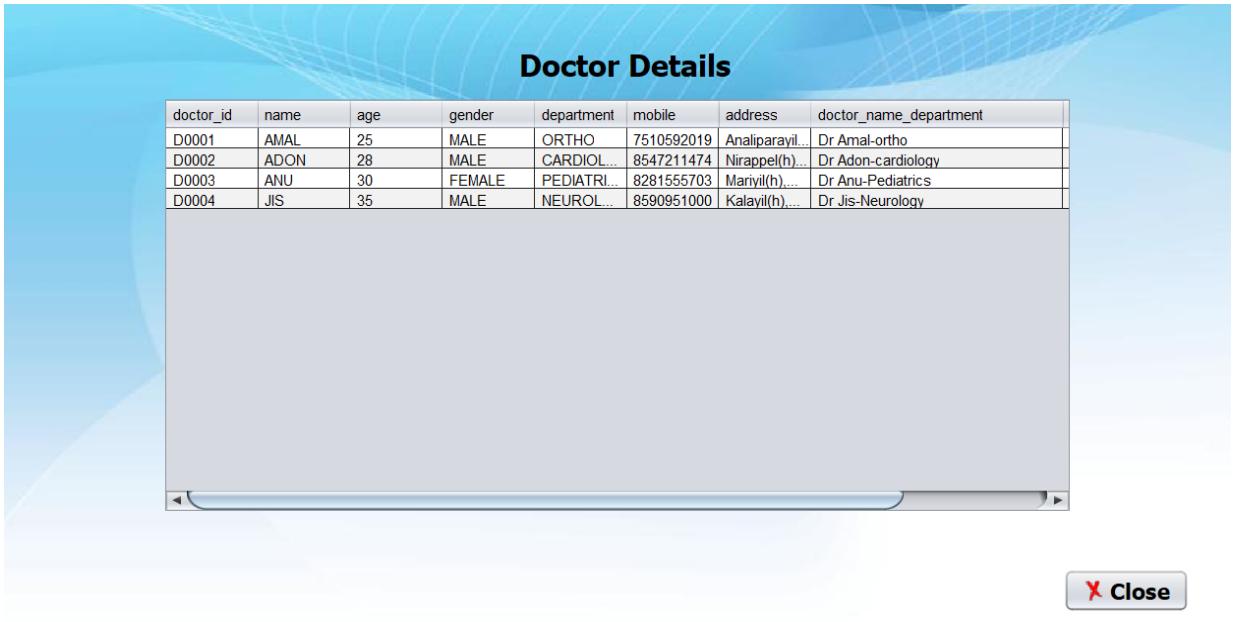

Figure 5.12: Front End Of The Doctor Details Of Hospital Management System.

The Figure 5.13 shows the front end of the update or delete doctor details where admin can update or delete doctor details.

Figure 5.13: Front End Of Update Or Delete Doctor Details Of Hospital Management System.

The Figure 5.14 shows the front end of receptionist registration page of Hospital Management System.

Figure 5.14: Front End Of Registration Form Receptionist Details.

The figure 5.15 shows the front end of the receptionist details of Hospital Management System.

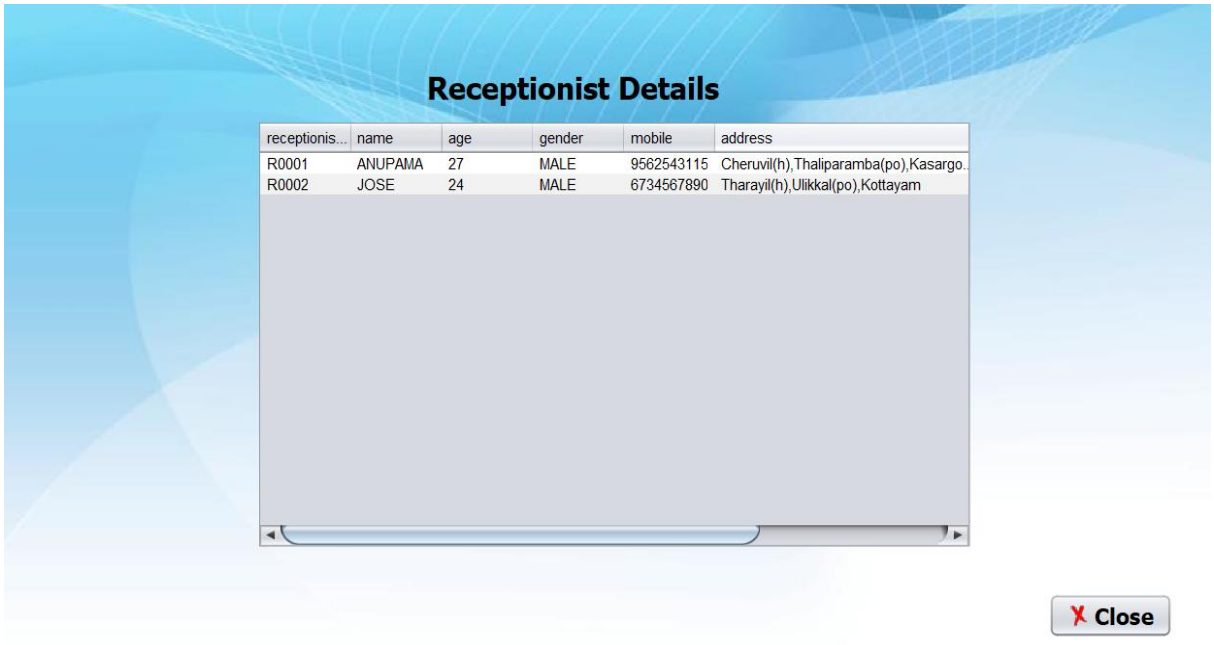

Figure 5.15: Front End Of Receptionist Details Of Hospital Management System.

The Figure 5.16 shows the front end of the update or delete receptionist details where admin can update or delete receptionist details.

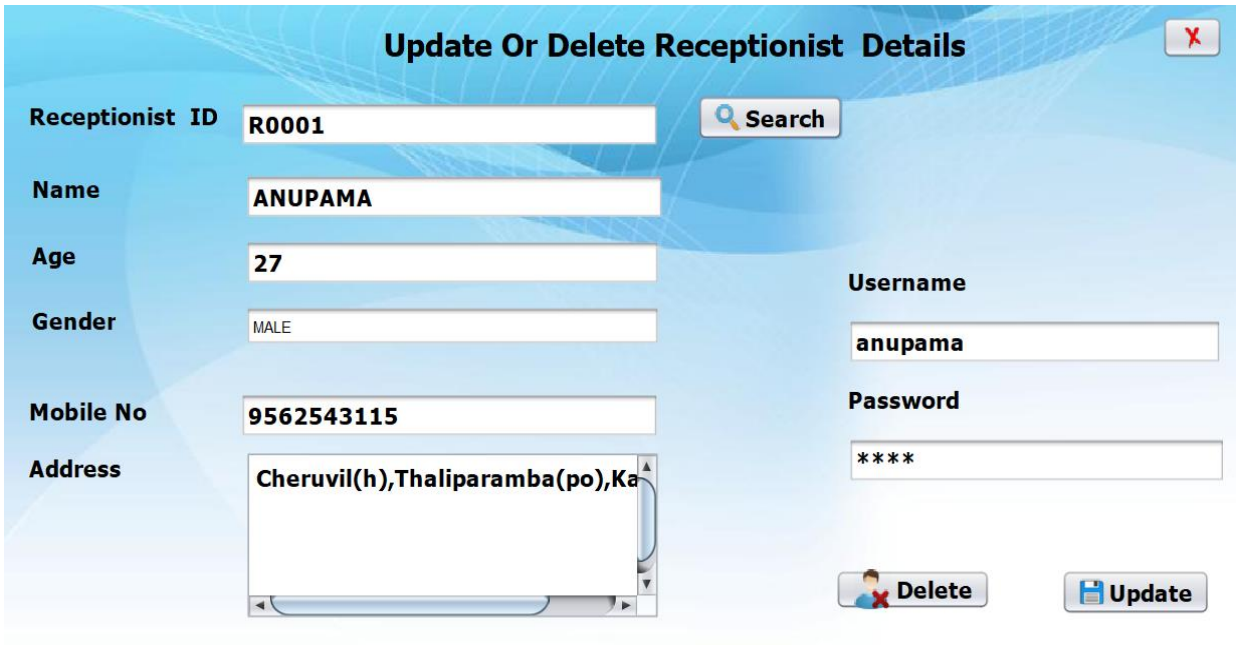

Figure 5.16: Front End Of Update Or Delete Receptionist Details .

The Figure 5.17 shows the front end of the receptionist login page of Hospital Management System.

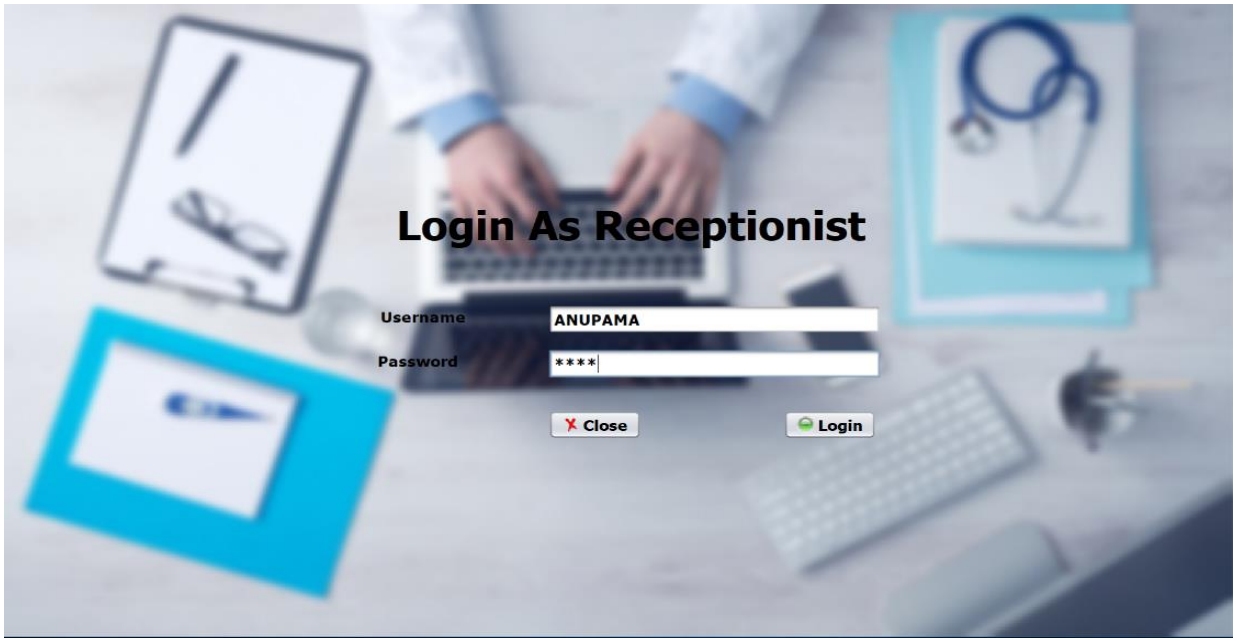

Figure 5.17: Front End Of The Receptionist Login Page Of Hospital Management System.

The Figure 5.18 shows the front end of the receptionist portal details of Hospital Management System.

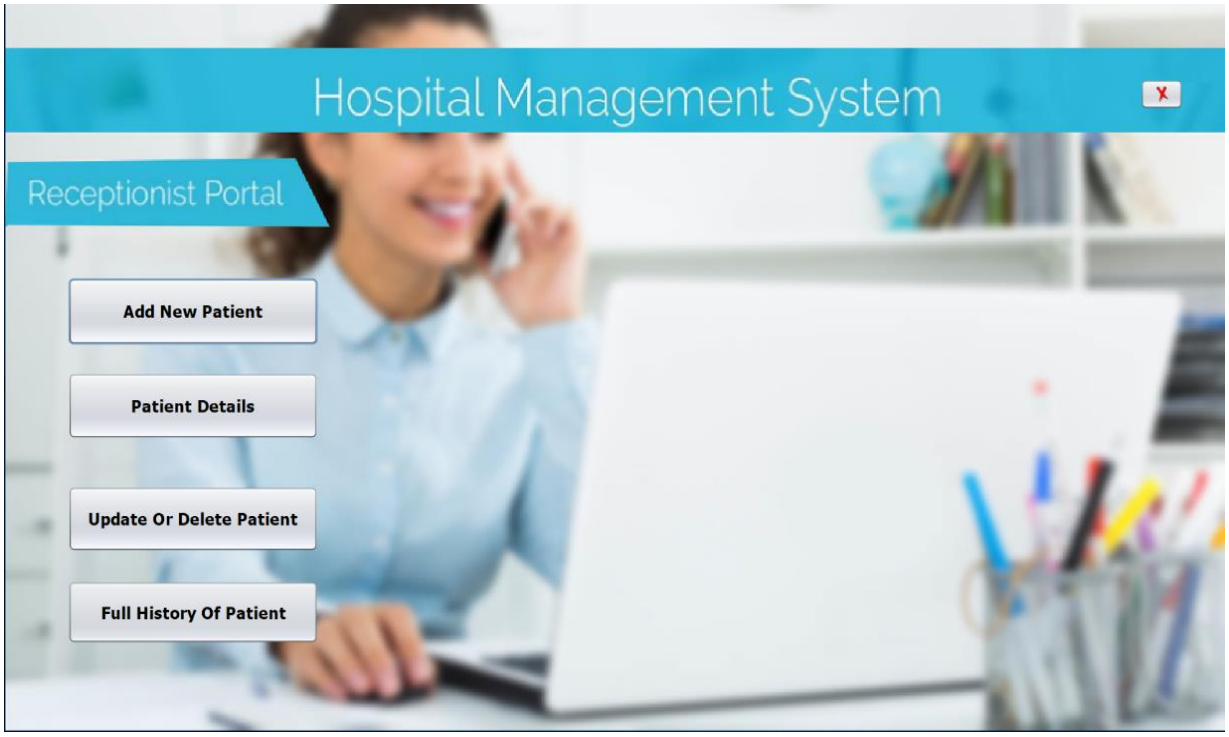

Figure 5.18 : Front End Of The Receptionist Portal Of Hospital Management System.

The Figure 5.19 shows the front end of the patient registration form of Hospital Management System.

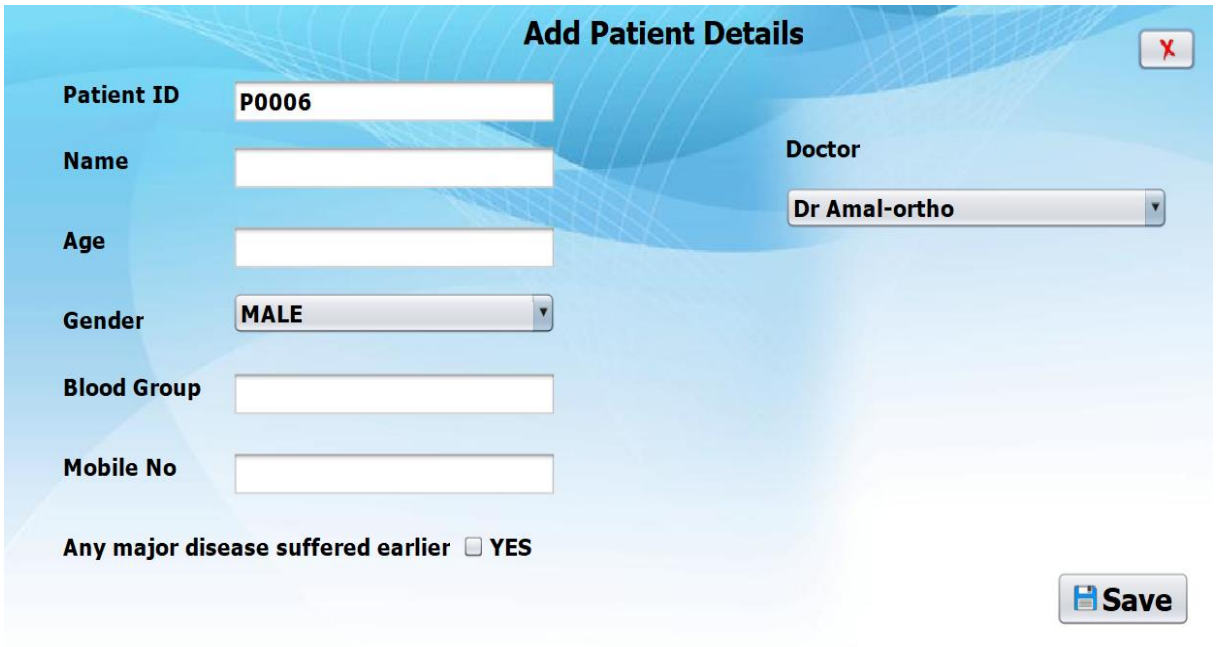

Figure 5.19 : Front End Of Patient Registration Form.

The Figure 5.20 shows the front end of the patient details of Hospital Management System.

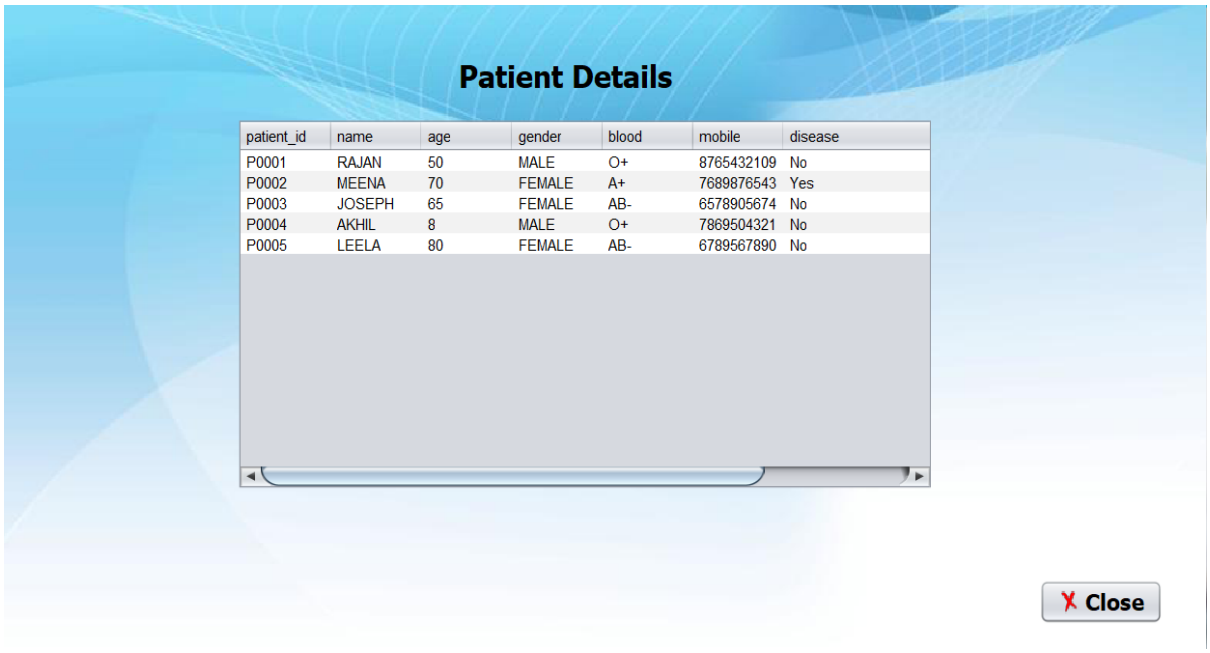

Figure 5.20: Front End Of Patient Details Of Hospital Management System.

The Figure 5.21 shows the front end of the update or delete patient details where admin can update or delete patient details.

Figure 5.21: Front End Of Update Or Delete Patient Details Of Hospital Management System.

The Figure 5.22 shows the front end of the doctor login page of Hospital Management System.

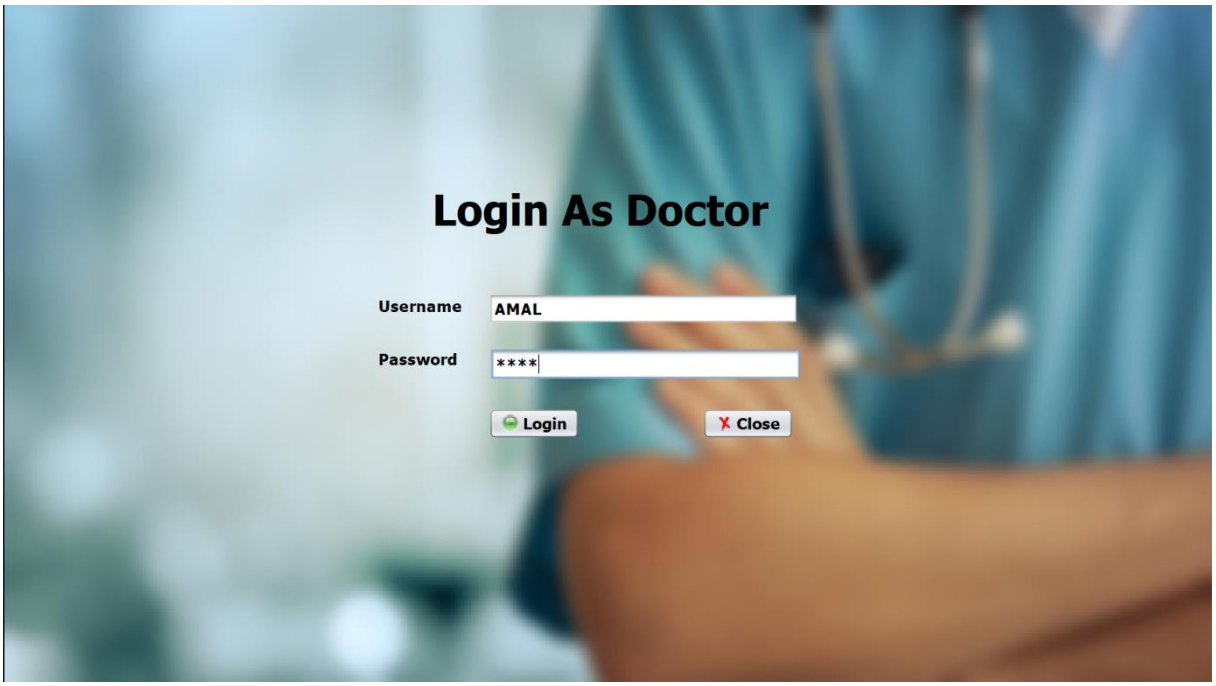

Figure 5.22: Front End Of Doctor Login Page Of Hospital Management System.

The Figure 5.23 shows the front end of the doctor portal details of Hospital Management System.

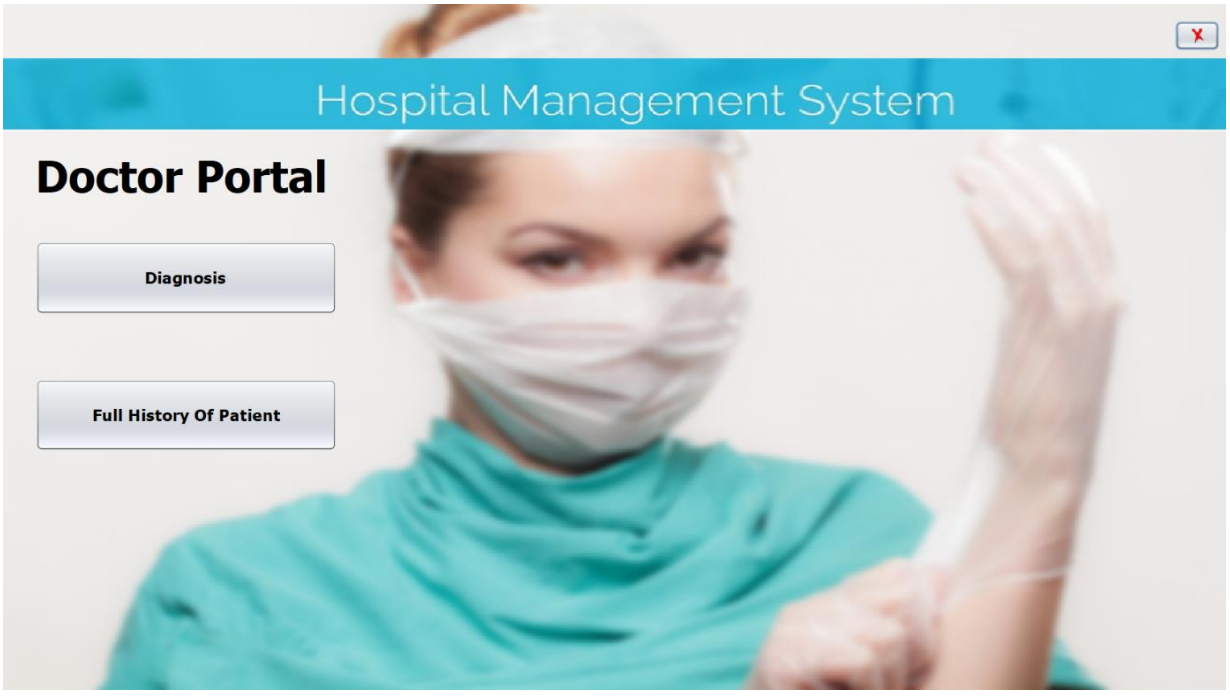

Figure 5.23: Front End Of Doctor Portal Of Hospital Management System.

The Figure 5.24 shows the front end of diagnosis of patient form.

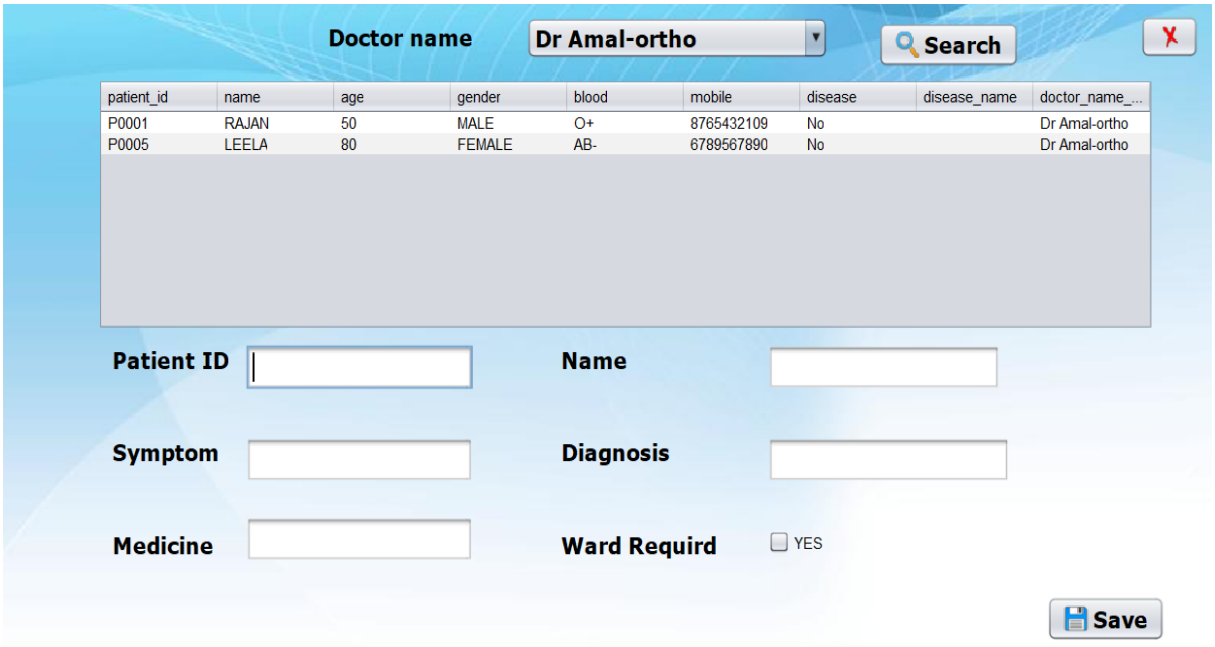

Figure 5.24: Front End Of Diagnosis Of Patient .

The Figure 5.25 shows the front end of the full history of patient details in the Hospital Management System.

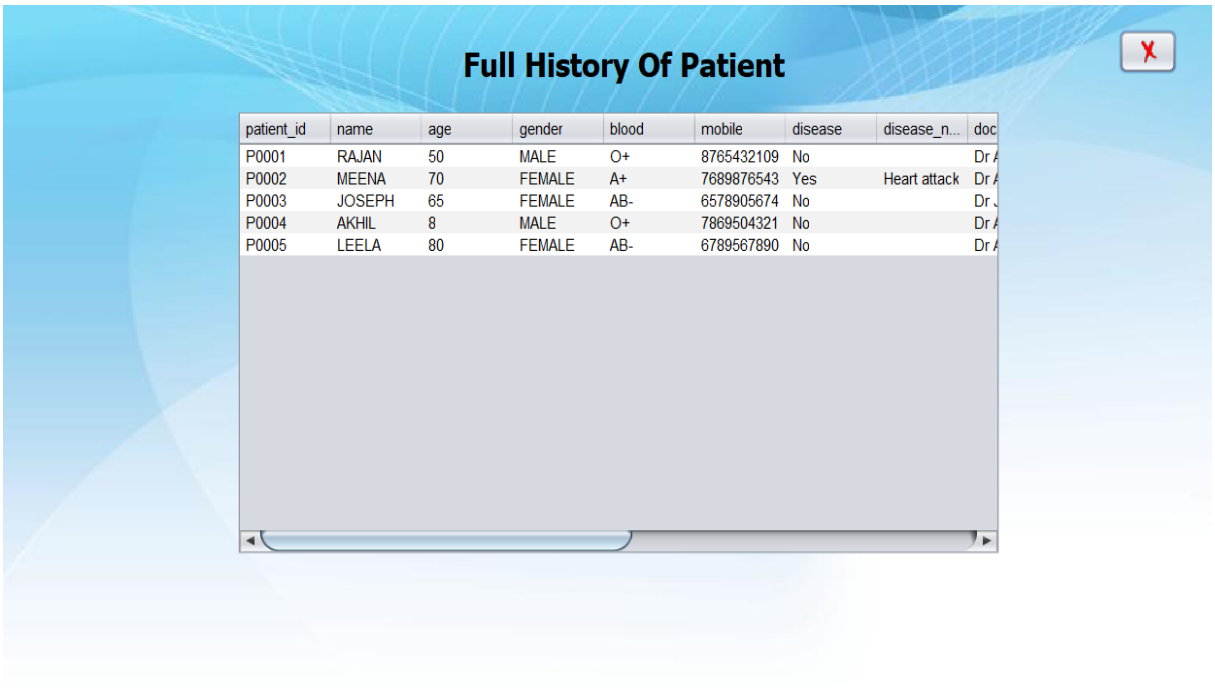

Figure 5.25: Front End Of Full History Of Patient Details In The Hospital Management System.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

Here the main objectives of this project, to create a desktop application by which admin can register the details of doctor, receptionist, patient and login to make changes in doctor details, receptionist details, patient details, department details and update or delate doctor details, receptionist details, patient details, department details.

6.2 Future Work

As a part of future enhancement, we can add pharmacy section for patient, which will help the patient to buy the medicine. Also provide blood bank section where any person can donate and receive blood. Hence willing to create an enhanced verion of existing system.

REFERENCES

[1] Ramez Elmasri and Shamkant B. Navathe "Database System Models language, Design and application programming"- Pearson, 7th Edition, 2017.

[2] T. harder, A. Reuter , "Database Systems for Non-Standard Application", Proc. Int Computing Symposium, 1983.

[3] Basics Of Java Programing: https://www.w3schools.com/java

[4] Java Database Connectivity: https://www.javatpoint.com/java-tutorial

[5] Java OOPs Concepts: https://www.tutorialspoint.com/java