1. INTRODUCTION

Hospitals are an extremely vital part of any society. They are, in essence, a complex structure of different groups of people performing various tasks to ensure its efficient working. Yet there is a lack of software that could economically ensure that all of the different departments of can function efficiently. The application provides an interface and functionality for every part of the hospitals multi-faceted environment. From storing patients records to making and modifying appointments this system facilitates better functioning of the Hospital..

1.1 OVERVIEW OF THE SYSTEM

The system provides a user interface to all end users, i.e front desk staff, doctors, nurses and pharmacists, involved in the management of the hospital. From the front desk managing the appointments to the doctors viewing patient records and writing prescriptions to the pharmacy maintaining its bills and inventory list, this system aims to centralize the management through one software and database. With immediate updation and user intuitive interfaces, the system provides a means to better organise the management process. It will aim to reduce patient wait times, billing and discharge process and also avoid confusion by maintaining the appointments on the centralized database. It will also provide the inventory list and maintain information about the medical and non-medical staff that is present in the hospital to ensure the hospital has an emergency medical response team. The system also maintains the finances of the hospital via the billing interface of the pharmacy and the hospital. Insurance claims are also stored for use in tax reports and claim verification.The system provides all the patients with a unique ID and stores all of their information. It provides a appointment interface for the front desk employees and the doctors to view the same. The pharmacy and the hospital can view their inventory and have their own billing interfaces.Visualization of key demographics is displayed.

1.2 PROBLEM STATEMENT

The application provides an interface and functionality for every part of the hospitals multi-faceted environment. From storing patients records to making and modifying appointments thissystem facilitates better functioning of all the departments of the Hospital.

2.SYSTEM ANALYSIS

2.1 EXISTING SYSTEM

Existing management systems in the healthcare sector provide numerous features to the end user. They provide features like clinical trial management and test data security, they also maintain the faculty and staff's information (licenses, insurances and practice permits) ensuring the physicians information is upto date [1]. Other systems provide OPD management, administration management, pharmacy information and also patient records management, all of which is available as an online web platform [2]. One system also provides cloud platform and patient centric workflow. They also provide options for storing medical documents [3].

2.1.1 DRAWBACKS OF EXISTING SYSTEM

The existing systems have some drawbacks that need to be highlighted. The systems do not provide different access levels for doctors, nurses, patients and the reception. An option to approve and reject appointments and switch timings and reschedule appointments is unavailable. Although a comprehensive system has been built they lack certain key features that Hridaya HMS provides. A solid emergency response page is unavailable. A way to monitor the end users movement in and out of the hospital is absent. No insightful data features are displayed in the existing systems to improve response and help keep the end user aware of the resources and expenditure of these resources and keep tab on the overall direction of the hospitals businesses.

2.2 PROPOSED SYSTEM

The system provides a user interface to all end users, i.e front desk staff, doctors, nurses and pharmacists, involved in the management of the hospital. From the front desk managing the appointments to the doctors viewing patient records and writing prescriptions to the pharmacy maintaining its bills and inventory list, this system aims to centralize the management through one software and database. With immediate updation and user intuitive interfaces, the system provides a means to better organise the management process. It will aim to reduce patient wait times, billing and discharge process and also avoid confusion by maintaining the appointments on the centralized database.

It will also provide the inventory list and maintain information about the medical and non-medical staff that is present in the hospital to ensure the hospital has an emergency medical response team.The system also maintains the finances of the hospital via the billing interface of the pharmacy and the hospital. Insurance claims are also stored for use in tax reports and claim verification.The system provides all the patients with a unique ID and stores all of their information. It provides a appointment interface for the front desk employees and the doctors to view the same. The pharmacy and the hospital can view their inventory and have their own billing interfaces. Visualization of key demographics is displayed.

2.2.1 ADVANTAGES OF PROPOSED SYSTEM

- A system that can replace the manual hospital management software.
- A database which stores patient details along with details of hospital staff.
- Reliable appointment making facility.
- Efficient handling of large amounts of data (Pharmacy and appointment).
- Doctor, Nurse, Front desk and pharmacy have a login.
- An easy to understand user friendly software.
- Attractive user interfaces to navigate through the system for the users.
- Visualization of stored data to gain meaningful insights.
- Emergency team readiness monitor.
- Facility to keep track of finances of the hospital

3. SYSTEM REQUIREMENTS

3.1 REQUIREMENT SPECIFICATION

3.1.1 Functional Requirements

● **Assigning an ID to the patients -** The HMS enables the staff in the front desk to provide a unique ID for each patient and then add them to the record sheet of the patient. The patients can utilize the ID throughout their hospital stay.

Check Out

● **Deleting Patient ID -** The staff in the administration section of the ward can delete the patient ID from the system when the patient's checkout from the hospital.

● **Adding to beds available list -**The Staff in the administration section of the ward can put the bed empty in the list of beds-available

Report Generation

● **Information of the Patient -** The Hospital Management System generates a report on every patient regarding various information like patients name, Phone number, bed number, the doctor's name whom its assigns, ward name, and more.

● **Availability of the Bed -** The Hospital Management system also helps in generating reports on the availability of the bed regarding the information like bed number unoccupied or occupied, ward name, and more.

Database

● **Mandatory Patient Information -** Every patient has some necessary data like phone number, their first and last name, personal health number, postal code, country, address, city, 'patient's ID number, etc.

● **Updating information of the Patient -** The hospital management system enables users to update the information of the patient as described in the mandatory information included.

3.2.1 Non Functional Requiremnts

Security

- Any modifications like insert, delete, update, etc. for the database can be synchronized quickly and executed only by the ward administrator.
- The staff in the front desk can view any data in the Hospital Management system, add new patients record to the HMS but they don't have any rights alter any data in it.
- The administrator can view as well as alter any information in the Hospital Management System.

Performance :

- Response Time - The system provides acknowledgment in just one second **.**

Maintainability:

- Back-Up -The system offers the efficiency for data back up.
- Errors - The system will track every mistake as well as keep a log of it.

Reliability :

- Availability - The system is available all the time.

3.2 HARDWARE REQUIREMENTS

- **System Memory:** About 1GB or higher for application excluding database.
- **RAM:** About 1GB for optimal performance.

3.3 SOFTWARE REQUIREMENTS

- **Platform:** Windows 8 or higher for stable and consistent performace.
- **Database :** MySQL (PHPMyAdmin)
- **Environment :** A working Java environment (JRE 7 or higher)

3.4 SYSTEM MODEL

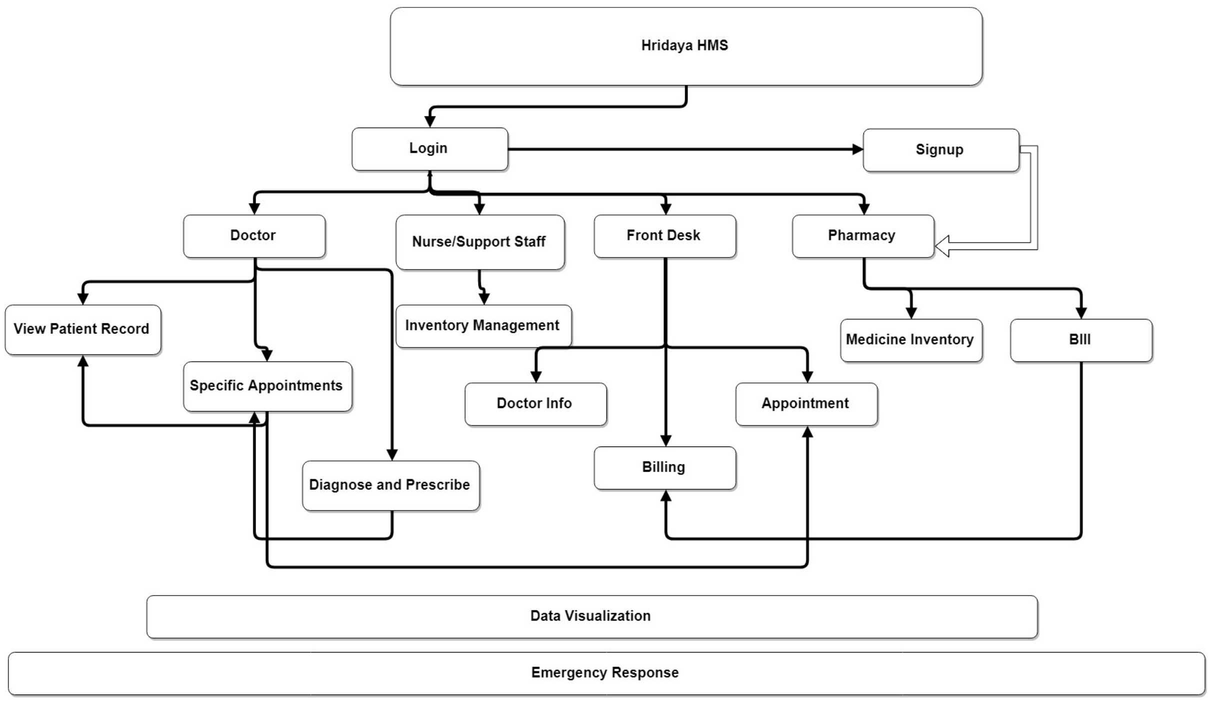


Fig. 3.4: Block Diagram

Login : The end user has to first login. The login interface is common to all types of end users. If the user doesn’t exist, signup facility is available

Doctor: The doctor can view the patients records. They can also view the appointments given to them specifically and modify this list. They can also prescribe and diagnose the patients that are available only to them(through appointments).

Front Desk: The front desk staff has the global appointment list and are the frist ones to add to this list. They also are responsible for the billing section. They have the facility to view all of the doctors information and credentials.

Nurse : The nurse end user is responsible for the inventory management (Beds, Operation theatre availability) along with the patients assigned to them.

Pharmacy: The pharmacy maintains the list of medicine inventory and also supplies and/or sells to the patient. This directly interacts with the billing available at the front desk.

Data Visualization: Data visualization features are available at almost every step of the way to different types of end users.

Emergency: The emergency team comprises of hospital staff that are currently logged on. If this number drops too low the end users are alerted immediately and a response is expected.

4 . DESIGN SPECIFICATION

4.1 SYSTEM ARCHITECTURE

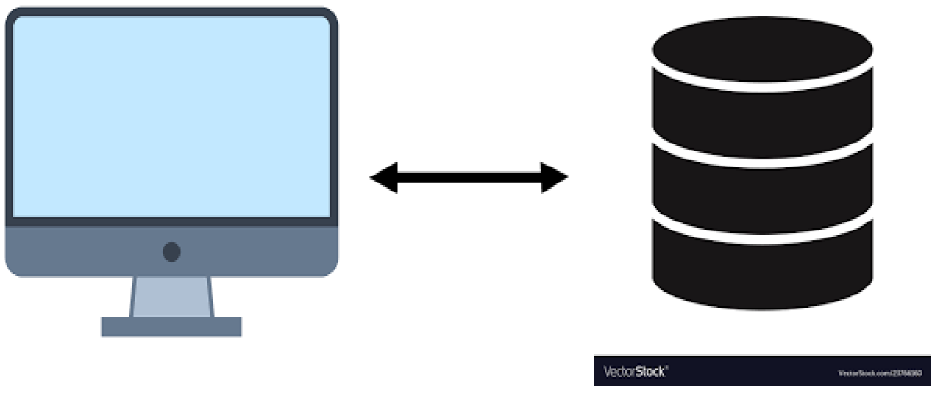


Fig. 4.1: 2tier Architecture

The proposed project has a 2-tier architecture as shown in Figure 1.1. The data is fetched from the database and given to the standalone application and data is written from the standalone application back to the database.

4.1.1 SUBSYSTEMS

Login: This system has four logins that is doctor, nurse, front-desk and administrator. All the end user types have different responsibilities and handle different operations of the hospital.

Registration: The registration process is for unregistered staff in the hospital. The registration process is overseen by the administrator (Related to the hiring process). The end users have to provide their details and login with their unique ID’s given to them by the system

Report Generation: The system provides reports of various data-reports at different stages in the processes. Data visualization is also provided along with most of these reports. The inventory management and medicine stocks for example have these reports and visualisation features.

Doctor :

a) Cancel/Reschedule appointment: The doctor can cancel the appointments or reschedule it making it a two-way process between the doctor and front desk.

b) Diagnose: The doctor can diagnose the patient after consultation and store information about the patient’s condition and history along with notes or comments in the database.
c) Prescribe: The doctor can prescribe medicines and these records are also stored in the database. These prescriptions can be identified by the patient ID and related to the diagnoses that the doctor performs.
d) Operate: The doctor can choose to admit the patient and operate on them. In this case the data of the room availability and the operation theatre status is also checked from the database to make an appointment for the same. The responsibility shifts to the nurse at this stage.

Nurse :

a) Inventory management: The nurse handles all of the inventory coming in and out of the hospital, including the medical inventory and the surgical inventory.

b) Room Maintenance: Each nurse is allotted a room for maintenance. They are responsible for checking which patient requires what care and also to maintain information about the same.

c) Operation Assistance: The nurses help the doctors/surgeons perform operations. Th information about the operations is also stored in the database. The responsibility of checking whether the operation theatre is free and equipped for the operation at hand falls on the nurse.

Front-Desk:

a) Make/Reschedule appointments: The front-desk employees start the appointment process and handle this front. They also handle rescheduling the appointment and cancellations.

b) Collect Patient info: The front desk employees collect the patient’s information and ensure all the data fields are intact and more or less accurate.

c) Billing: The front desk employees can access all the appointment tables to generate bills for the patient. The data is for patient’s consultation, operation and room / stay charges.

4.2 DATABASE DESIGN

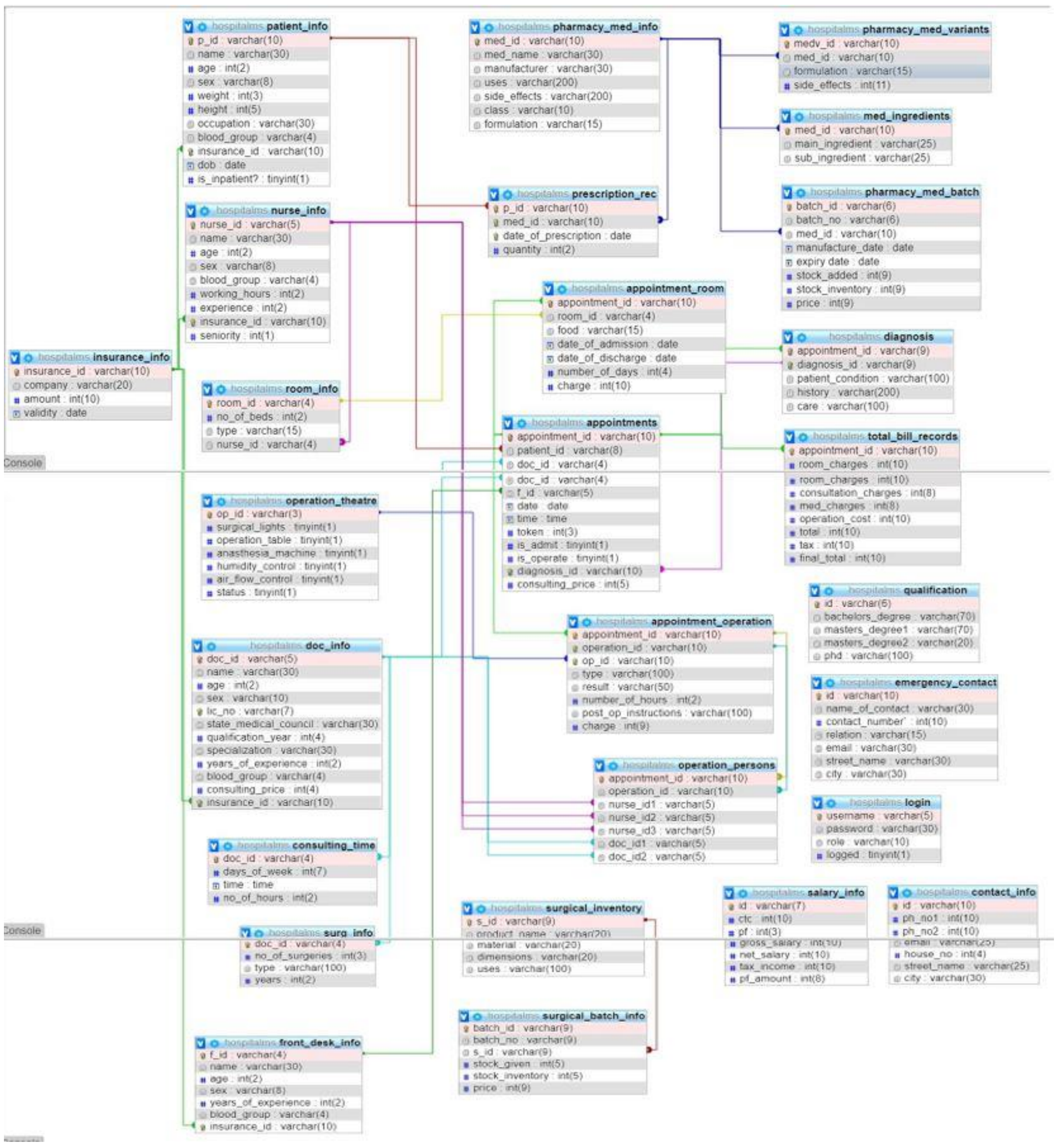


Fig. 4.2: Database Design

4.2.1 TABLE DESIGN

appointments
Table comments: Contains all of the appointments coming through the hospital. Core table

Column	Type	Null	Comments
appointment_id *(Primary)*	varchar(10)	No	The unique key that identifies the appointments.
patient_id	varchar(8)	No	The patient ID refers to the patient that made the appointment,
doc_id	varchar(4)	No	Doc ID refers to the doctor that the patient is scheduled to see.
f_id	varchar(5)	No	The front desk employee that initiated the appointment.
Date	date	No	Date of appointment.
Time	time	No	Time of the scheduled appointment
is_admit	tinyint(1)	No	Whether patient has to be admitted.
is_operate	tinyint(1)	No	Whether an operation is required.
diagnosis_id	varchar(10)	No	The diagnosis of the doctor is filled out here.
consulting_price	varchar(5)	No	The price for consultation.
Paid	tinyint(1)	No	Whether the bill has been paid
Cancelled	tinyint(1)	No	Whether the bill has been cancelled

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	appointment_id	7	A	No	
diagnosis_id	BTREE	Yes	No	diagnosis_id	7	A	No	

appointment_operation
Table comments: Contains information about the procedures in the hospital.

Column	Type	Null	Comments
appointment_id *(Primary)*	varchar(10)	No	Unique appointment identifier.
operation_id	varchar(10)	No	The operation ID identifies the operation uniquely.
op_id	varchar(10)	No	The operation identifier which is unique
Type	varchar(100)	No	The type of operation being performed.
Result	varchar(50)	No	The result of the operation.
number_of_hours	int(2)	No	The number of hours the operation procedure took place.
Charge	int(9)	No	The price of the operation.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	appointment_id	1	A	No	
op_id	BTREE	Yes	No	op_id	1	A	No	
operation_id	BTREE	Yes	No	operation_id	1	A	No	

appointment_room
Table comments: Contains the appointment info of inpatients in rooms.

Column	Type	Null	Comments
appointment_id *(Primary)*	varchar(10)	No	Unique appointment ID that identifies the appointment and hence all the other info.
room_id	varchar(4)	No	The room that is designated to the patient.
Food	varchar(15)	No	Food choice of the inpatient.
date_of_admission	date	No	Date of admission of the patient.
date_of_discharge	date	No	Date of discharge of the patient.
number_of_days	int(4)	No	Number of days that the patient is admitted.
Charge	int(10)	No	The price of the stay.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	appointment_id	1	A	No	

consulting_time
Table comments: Contains information about the doctors consultation timings

Column	**Type**	**Null**	**Comments**
doc_id *(Primary)*	varchar(4)	No	The unique identifier for doctors.
days_of_week	int(7)	No	The days of week the dcotor comes in. From 1-7
Time	time	No	time of arrival of the doctor.
no_of_hours	**varchar(2)**	**No**	**The number of hours the doctor accepts consultations.**

Indexes

Keyname	**Type**	**Unique**	**Packed**	**Column**	**Cardinality**	**Collation**	**Null**	**Comment**
PRIMARY	BTREE	Yes	No	doc_id	2	A	No	

contact_info
Table comments: Contains all of the contact info of the employees and the patients.

Column	**Type**	**Null**	**Default**	**Links to**	**Comments**	**MIME**
id *(Primary)*	varchar(10)	No			The unique ID identifying the user.	
ph_no1	varchar(10)	No			Phone number of work.	
ph_no2	varchar(10)	Yes	*NULL*		Phone number home.	
email	varchar(25)	No			Email of the user	
house_no	int(4)	No			House number of the user.	
street_name	varchar(25)	No			Street of residence of the user.	
City	varchar(30)	No			City of residence	

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	5	A	No	

diagnosis
Table comments: Contains the diagnosis of the patient with comments.

Column	Type	Null	Comments
appointment_id *(Primary)*	varchar(9)	No	Unique appointment identifier.
diagnosis_id	varchar(9)	No	The ID of the diagnosis records.
patient_condition	varchar(100)	No	Condition of the patient/the disease.
History	varchar(200)	No	Brief history of the patients condition.
Care	varchar(100)	No	brief notes about the care to be taken by patient.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	appointment_id	3	A	No	
diagnosis_id	BTREE	Yes	No	diagnosis_id	3	A	No	

doc_info
Table comments: Table contains all of the information regarding the doctor. Basic information along with insurance foreign key is also present.

Column	Type	Null	Comments
doc_id *(Primary)*	varchar(5)	No	Unique id assigned to the dcotors to identify them in the system. Starts with 'd'.
Name	varchar(30)	No	Name of the doctor.
Age	varchar(2)	No	Age of the doctor
Sex	varchar(10)	No	Gender of the doctor
lic_no	varchar(7)	No	Unique license number of the doctor awarded by the state medical council
state_medical_council	varchar(30)	No	The state medical council which awarded the license to practice.
qualification_year	varchar(4)	No	The year of qualification for practice.
Specialization	varchar(30)	No	major specialization of the doctor.
years_of_experience	varchar(2)	No	The number of years of practice/Experience.
blood_group	varchar(4)	No	The blood group of the doctor.
consulting_price	varchar(4)	No	The consulting price of the appointment.
insurance_id	varchar(10)	No	The insurance ID of the doctor(Health).
Employed	varchar(2)	No	Whether the doctor is employed or not.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	doc_id	2	A	No	
lic_no	BTREE	Yes	No	lic_no	2	A	No	
insurance_id	BTREE	Yes	No	insurance_id	2	A	No	

emergency_contact
Table comments: Contains all of the emergency contact info of users and patients.

Column	Type	Null	Comments
id *(Primary)*	varchar(10)	No	Unique ID of the users and patients.
name_of_contact	varchar(30)	No	Name of the emergency contact.
contact_number	varchar(10)	No	Contact number of the emergency contact.
Relation	varchar(15)	No	Relation to the user (ID)
Email	varchar(30)	No	Email of the emergency contact
street_name	varchar(30)	No	street name of the emergency contact
City	varchar(30)	No	City of residence of the emergency contact

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	4	A	No	

front_desk_info
Table comments: Contains basic info of a front desk employee.

Column	Type	Null	Comments
f_id *(Primary)*	varchar(4)	No	The unique ID given to the front desk.
Name	varchar(30)	No	name of the employee
Age	varchar(2)	No	Age of the employee
Sex	varchar(8)	No	Gender of the employee
years_of_experience	varchar(2)	No	The years of experiencee.
blood_group	varchar(4)	No	Blood group type of employee
insurance_id	varchar(10)	No	Unique insurance id of the front desk employee.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	f_id	1	A	No	
insurance_id	BTREE	Yes	No	insurance_id	1	A	No	

insurance_info
Table comments: Contains insurance information of all the users and patients

Column	Type	Null	Comments
insurance_id *(Primary)*	varchar(10)	No	The unique insurance ID of the user
company	varchar(20)	No	Company providing the insurance.
Amount	int(10)	No	The insured amount specified in the insurance.
Validity	date	No	The date till which the insurance is valid.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	insurance_id	4	A	No	

login
Table comments: Table contains the usernames and passwords which will be use

Column	Type	Null	Comments
username *(Primary)*	varchar(5)	No	The unique ID's assigned to doctors,nurses,front desk employees and the patients.
password	varchar(100)	No	Password.
Role	varchar(10)	No	The role of the user logging in. Based on username.
Logged	tinyint(1)	No	Whether the user is currently logged on or not.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	username	5	A	No	

log_times

Table comments: Maintains login logout times and dates of all end users

Column	Type	Null	Comments
id *(Primary)*	varchar(10)	No	The id's of end users
date_login *(Primary)*	date	No	The date of login
login_time *(Primary)*	time	No	The login time of user
logout_time	time	Yes	The logout time of user.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	6	A	No	
				date_login	10	A	No	
				login_time	32	A	No	

med_ingredients

Table comments: Contains the composition of the medicines available

Column	Type	Null	Comments
med_id *(Primary)*	varchar(10)	No	The unique identifier for medicines.
main_ingredient	varchar(25)	No	Main composite ingredient
sub_ingredient	varchar(25)	Yes	The sub ingredient of the medicine

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	med_id	0	A	No	

nurse_info
Table comments: Contains basic information about the nurses in the hospital.

Column	Type	Null	Comments
nurse_id *(Primary)*	varchar(5)	No	Unique identifier for nurses. Starts with 'n'.
Name	varchar(30)	No	Name of the nurse.
Age	varchar(2)	No	Age of the nurse.
Sex	varchar(8)	No	Gender of the nurse.
blood_group	varchar(4)	No	Blood group of the nurse.
working_hours	varchar(2)	No	The number of hours put in at work everyday.
experience	varchar(2)	No	Number of years of experience.
insurance_id	varchar(10)	No	Unique insurance ID of the nurse.
seniority	varchar(1)	No	A scale of 0-2 which describes the seniority of the nurse.
employed	varchar(2)	No	Whether the nurse is employed or not

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	nurse_id	1	A	No	
insurance_id	BTREE	Yes	No	insurance_id	1	A	No	

operation_persons
Table comments: Contains the staff that is performing the procedure.

Column	Type	Null	Comments
appointment_id *(Primary)*	varchar(10)	No	The unique identifier for appointments with operations.
operation_id	varchar(10)	No	The id of the operation appointment.
nurse_id1	varchar(5)	No	The nurse helping in the appointment.
doc_id1	varchar(5)	No	Surgeon performing the procedure.
doc_id2	varchar(5)	Yes	Optional surgeon on procedure.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	appointment_id	0	A	No	

operation_theatre
Table comments: Contains information about operation theater readiness.

Column	Type	Null	Comments
op_id *(Primary)*	varchar(3)	No	Unique ID identifying the operation theater.
surgical_lights	tinyint(1)	No	Whether the operation theater has surgical lights and is working.
operation_table	tinyint(1)	No	Whether the operation table is ready.
anasthesia_machine	tinyint(1)	No	Whether the anasthesia machine is ready.
humidity_control	tinyint(1)	No	Whether the humidity control is working.
air_flow_control	tinyint(1)	No	Whether the air flow control is working or not.
Status	tinyint(1)	No	Whether the operation theater is ready or not. Based on above boolean values.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	op_id	0	A	No	

patient_info

Table comments: Contains the required information about the patient.

Column	Type	Null	Comments
p_id *(Primary)*	varchar(10)	No	Unique ID assigned to evry patient to identify them in the system.
Name	varchar(30)	No	Name of the Patient
Age	int(2)	No	Age of the Patient
Sex	varchar(8)	No	Gender of the patient.
Weight	int(3)	No	The weight of the patient in kilograms.
Height	int(5)	No	Height of the Patient in inches
occupation	varchar(30)	No	Occupation of the patient to recognize lifestyle.
blood_group	varchar(4)	No	Blood group of the Patient
insurance_id	varchar(10)	No	The insurance of the patient. Can be 'no' incase patient has no insurance.
Dob	date	No	The date of birth of the employee.
is_inpatient	tinyint(1)	No	Whether the patient is resident in the hospital.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	p_id (9)	1	A	No	
insurance_id	BTREE	Yes	No	insurance_id	1	A	No	

pharmacy_med_batch
Table comments: Contains the medicine batches information and stock.

Column	Type	Null	Comments
batch_id *(Primary)*	varchar(6)	No	The unique identifier to recognize the batches of medicine.
batch_no	varchar(6)	No	The batch number to identify seller side sale info..
med_id	varchar(10)	No	The medicine that is being delivered in batch.
manufacture_date	date	No	The date of the manufacture of the medicine.
expiry date	date	No	Date of expiry of the medicine in the batch.
stock_added	int(9)	No	The amount of medicine given/supplied.
stock_inventory	int(9)	No	The stock left in inventory.
Price	int(9)	No	The price of the medicine in the specified batch.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	batch_id	0	A	No	

pharmacy_med_info
Table comments: Contains all the information about the medicines available.

Column	Type	Null	Comments
med_id *(Primary)*	varchar(10)	No	Unique identifier for medicines.
med_name	varchar(30)	No	Name of the medicine identified by med_id.
manufacturer	varchar(30)	No	Company that manufactures the said medicines.
side_effects	varchar(200)	No	The possible side effects of the said medicine.
Class	varchar(10)	No	The class of the drug. Determines whether prescriptions re required and other cautionaries.
formulation	varchar(15)	No	The form factor of the said medicine.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	med_id	0	A	No	

pharmacy_med_variants
Table comments: Contains all of the variants/alternates of the said med_id.

Column	Type	Null	Comments
medv_id *(Primary)*	varchar(10)	No	The unique identifier for the medicine variants.
med_id	varchar(10)	No	The med_id for which the medv_id is the variant/alternative.
formulation	varchar(15)	No	Form factor of the said medicine.
side_effects	int(11)	No	possible side effects of the drug. Could be different from med_id side effects.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	medv_id	0	A	No	

prescription_rec
Table comments: Contains all of the prescription information to patient.

Column	Type	Null	Comments
p_id *(Primary)*	varchar(10)	No	The unique ID that identifies the patient.
med_id *(Primary)*	varchar(10)	No	The medicine ID that identifies the medicine.
date_of_prescription *(Primary)*	date	No	The date of prescription.
quantity	int(2)	No	The quantity prescribed.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	**BTREE**	**Yes**	**No**	**p_id**	**0**	**A**	**No**	
				med_id	**0**	**A**	**No**	
				date_of_prescription	**0**	**A**	**No**	

qualification
Table comments: Contains the information about qualification of the employee

Column	Type	Null	Comments
id *(Primary)*	varchar(6)	No	The ID that identifies user type.
bachelors_degree	varchar(70)	No	The bachelors degree of the employee
masters_degree1	varchar(70)	Yes	Masters degree of employee
masters_degree2	varchar(20)	Yes	Second masters degree.
phd	varchar(100)	Yes	Specialization of employee

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	4	A	No	

room_info

Table comments: Contains relevant room information (Basic)

Column	Type	Null	Comments
room_id *(Primary)*	varchar(4)	No	The room ID identofues the rooms uniquely.
no_of_beds	int(2)	No	The number of beds available in the room.
Type	varchar(15)	No	The type of sharing the room accomodates.
nurse_id	varchar(4)	No	The ID of the nurse incharge of this rooms maintenance.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	room_id	0	A	No	

salary_info

Table comments: Contains the cost to hospital through employees.

Column	Type	Null	Comments
id *(Primary)*	varchar(7)	No	The unique ID of the user (nurses,doctors,front desk).
Ctc	varchar(10)	No	The cost to company of the employee.
Pf	varchar(3)	No	The pf that is being given(in percentage)
gross_salary	varchar(10)	No	Gross salary of the employee
net_salary	varchar(10)	No	The net salary after calculating pf
tax_income	varchar(10)	No	income tax that is supposed to be paid.
pf_amount	varchar(8)	No	The amount of pf calculated from pf above.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	id	3	A	No	

surgical_batch_info
Table comments: Contains information of the delivery of surgical inventory.

Column	Type	Null	Comments
batch_id *(Primary)*	varchar(9)	No	Unique identifier that represents the batch ID.
batch_no	varchar(9)	No	The batch number with respect to the seller.
s_id	varchar(9)	No	The inventory that is being delivered.
stock_given	int(5)	No	The amount of stock delivered.
stock_inventory	int(5)	No	The amount in inventory.
Price	int(9)	No	Price of the batch being delivered.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	batch_id	0	A	No	

surgical_inventory
Table comments: Contains information about the surgical inventory available.

Column	Type	Null	Comments
s_id *(Primary)*	varchar(9)	No	The unique ID that identifues the surgical inventory.
product_name	varchar(20)	No	The name of the product identifies by s_id.
material	varchar(20)	No	The material of the given product(composition).
dimensions	varchar(20)	No	The dimensions of the product in string form.
Uses	varchar(100)	No	The uses of the inventory.

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	s_id	0	A	No	

surg_info
Table comments: Contains information about the doctors surgical expertise.

Column	Type	Null	Comments
doc_id *(Primary)*	varchar(4)	No	The unique doctor identifier.
no_of_surgeries	int(3)	No	Number fo surgeries performed.
Type	varchar(100)	No	Brief about the types of operations.
Years	int(2)	No	Years of performing surgery (experience)

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	doc_id	0	A	No	

total_bill_records
Table comments: Contains the financial bill information.

Column	Type	Null	Comments
appointment_id *(Primary)*	varchar(10)	No	The appointment ID for which the bill is being generated.
consultation_charges	varchar(8)	No	The consulation charges of the appointment ID.
Total	varchar(10)	No	The total bill amount.
Tax	varchar(10)	No	tx calculated on the total of the bill.
final_total	varchar(10)	No	The total bill amount after taxes.
Paid	varchar(10)	No	Either cash or card

Indexes

Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY	BTREE	Yes	No	appointment_id	1	A	No	

4.2.2 DATA FLOW DIAGRAM

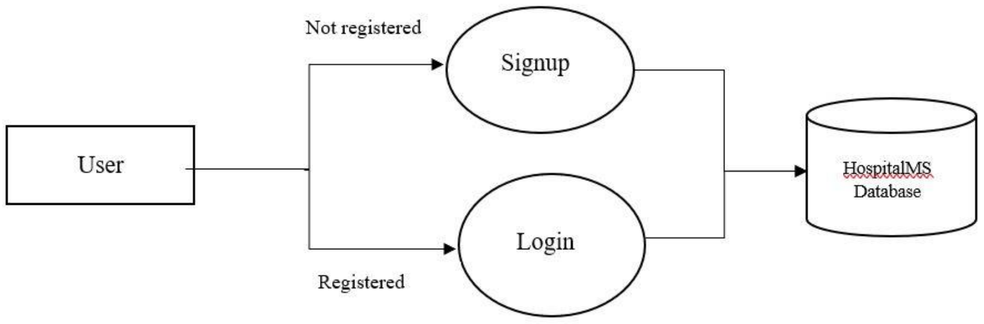


Fig 4.4.1: 0 DFD

The system first requires the user to login to perform any action at all. If the user is unregistered, a signup option (registration) is available.

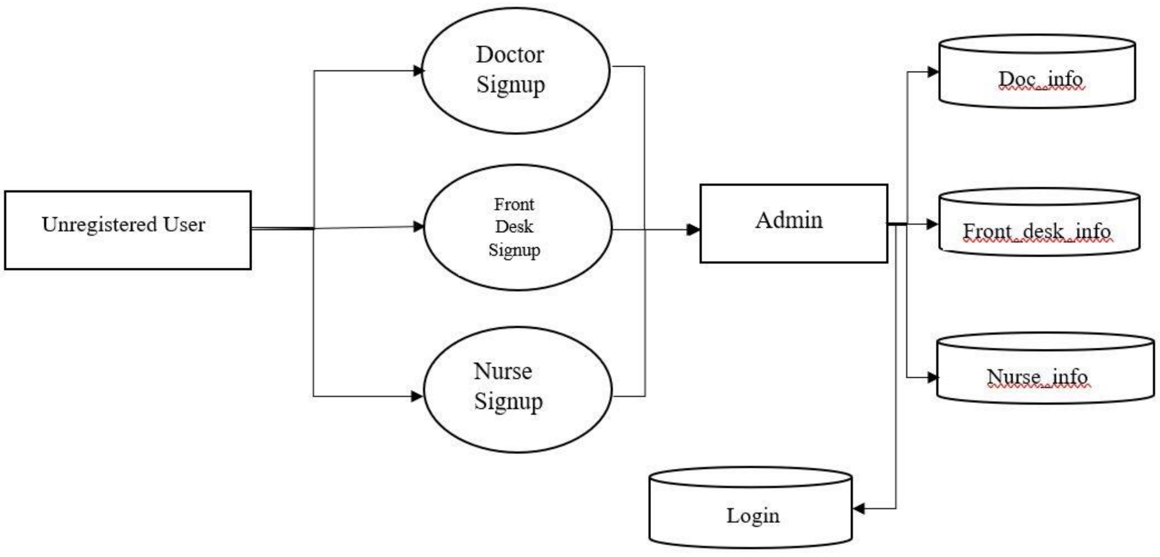
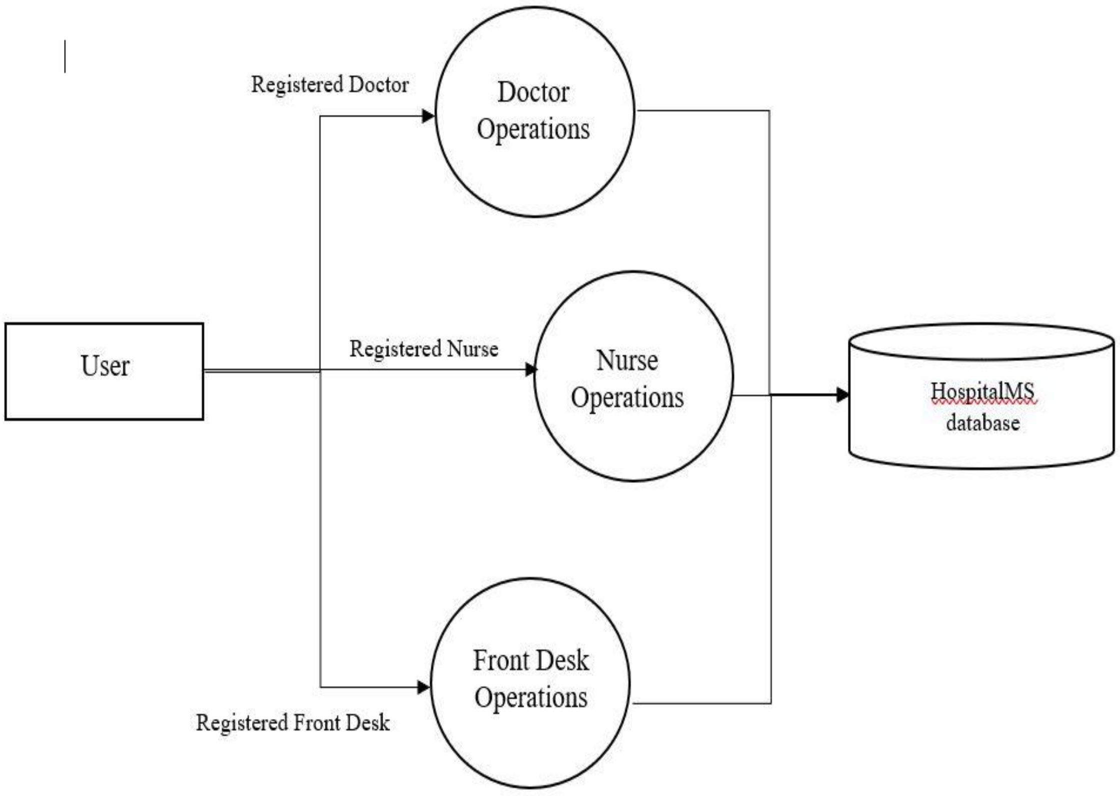


Fig 4.4:2 DFD Signup



The signup process consists of three different processes for three end user types, the doctor, nurse and the front desk employee. The respective data collection and validity is verified by the administrator who oversees this process.

Fig 4:4: 3 DFD Login

The system has a common login screen although there are different types of end users. Based on the login credentials the system automatically determines the user type and serves the end user the respective operation type and options.

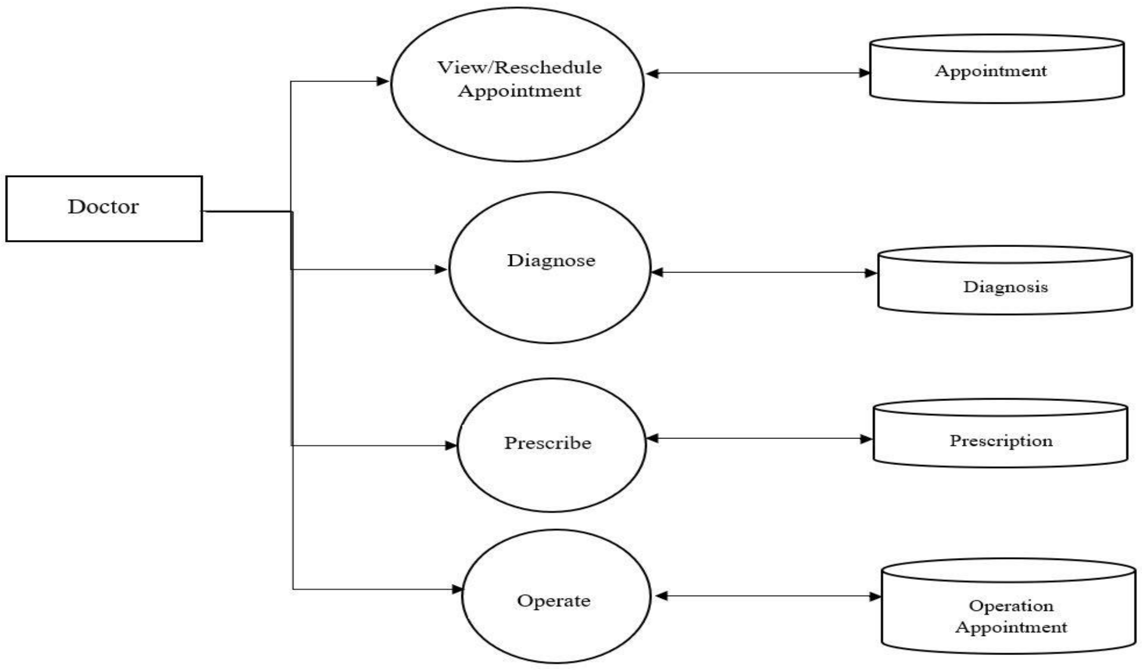


Fig 4.4.4DFD Doctor

The doctor can cancel the appointments or reschedule it making it a two-way process between the doctor and front desk. The doctor can diagnose the patient after consultation and store information about the patient's condition and history along with notes or comments in the database.

The doctor can prescribe medicines and these records are also stored in the database. These prescriptions can be identified by the patient ID and related to the diagnoses that the doctor performs. The doctor can choose to admit the patient and operate on them. In this case the data of the room availability and the operation theatre status is also checked from the database to make an appointment for the same. The responsibility shifts to the nurse at this stage.

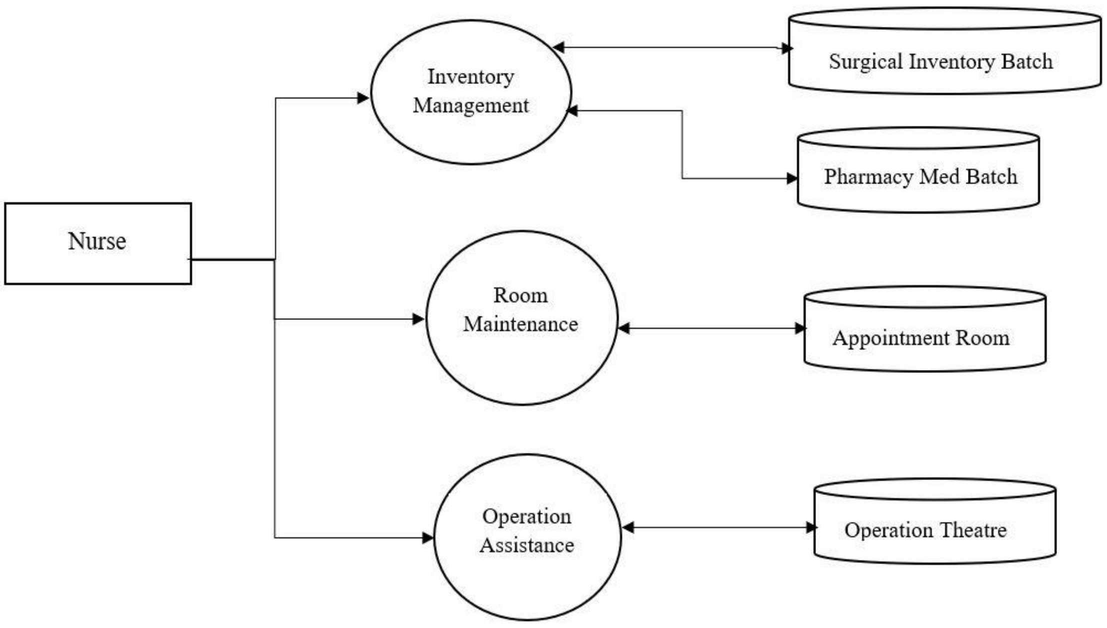


Fig 4.3: 1 DFD Nurse

The nurse handles all of the inventory coming in and out of the hospital, including the medical inventory and the surgical inventory. Each nurse is allotted a room for maintenance. They are responsible for checking which patient requires what care and also to maintain information about the same.

The nurses help the doctors/surgeons perform operations. The information about the operations is also stored in the database. The responsibility of checking whether the operation theatre is free and equipped for the operation at hand falls on the nurse.

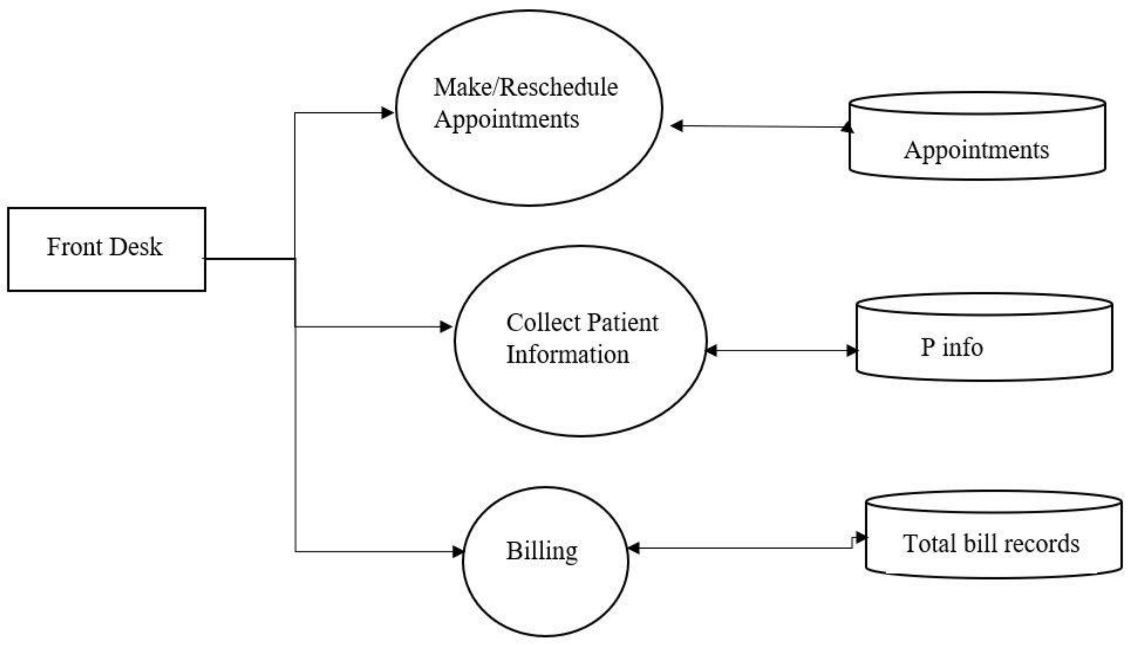


Fig 4.3: 1 DFD Front desk

The front-desk employees start the appointment process and handle this front. They also handle rescheduling the appointment and cancellations. The front desk employees collect the patient’s information and ensure all the data fields are intact and more or less accurate. The front desk employees can access all the appointment tables to generate bills for the patient. The data is for patient’s consultation, operation and room / stay charges.

4.3 USER INTERFACE DESIGN

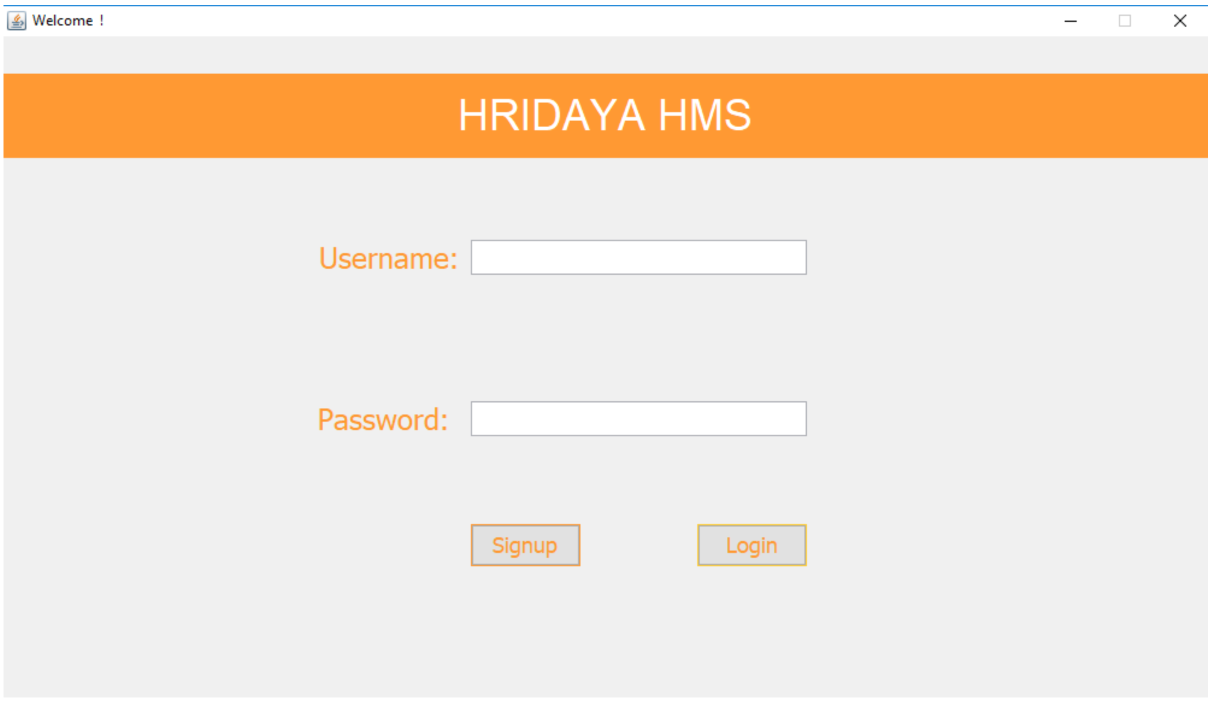
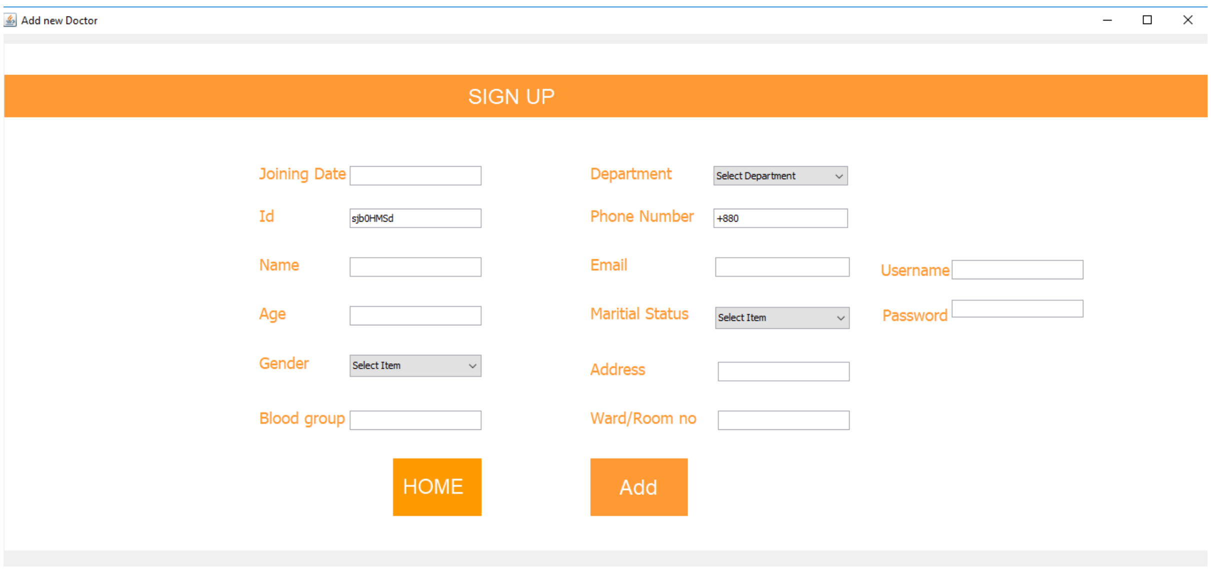


Fig 4.3.1: Login Screen

The user is supposed to enter their credentials, that is, their ID and password. Depending on the credentials, a user can be a Doctor, Nurse, Front-desk or an Admin and they be directed to their respective pages.



Joining Date		Department	Select Department		
Id	sjb0HMSd	Phone Number	+880		
Name		Email		Username	
Age		Maritial Status	Select Item	Password	
Gender	Select Item	Address			
Blood group		Ward/Room no			

Fig 4.3.2: Registration Screen

The registration process is for unregistered staff in the hospital. The registration process is overseen by the administrator (Related to the hiring process). The end users have to provide their details and login with their unique ID's given to them by the system.

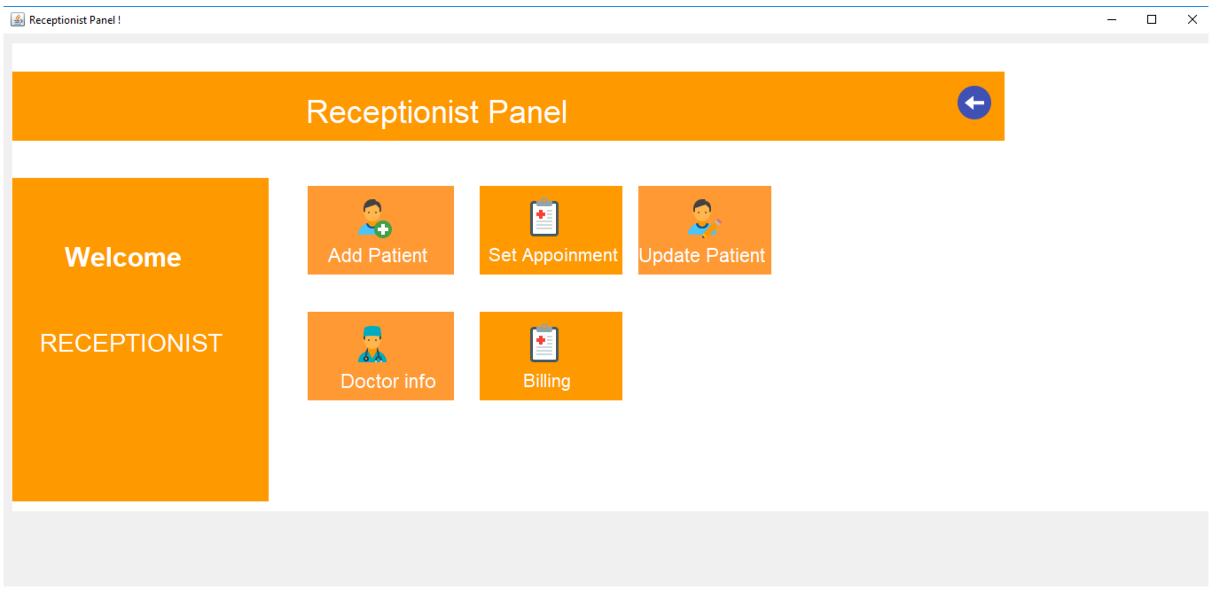


Fig 4.3.3: Receptionist Panel Screen

The front-desk employees start the appointment process and handle this front. They also handle rescheduling the appointment and cancellations. The front desk employees collect the patient’s information and ensure all the data fields are intact and more or less accurate. The front desk employees can access all the appointment tables to generate bills for the patient. The data is for patient’s consultation, operation and room / stay charges.

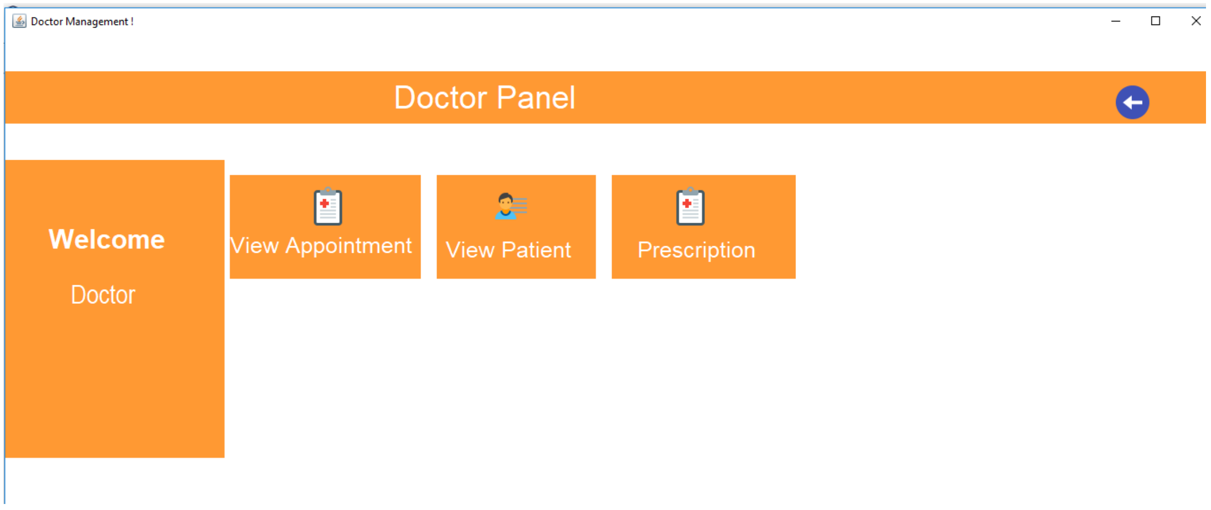


Fig 4.3.4: Doctor Panel Screen

The doctor can cancel the appointments or reschedule it making it a two-way process between the doctor and front desk. The doctor can diagnose the patient after consultation and store information about the patient’s condition and history along with notes or comments in the database.

The doctor can prescribe medicines and these records are also stored in the database. These prescriptions can be identified by the patient ID and related to the diagnoses that the doctor performs. The doctor can choose to admit the patient and operate on them. In this case the data of the room availability and the operation theatre status is also checked from the database to make an appointment for the same. The responsibility shifts to the nurse at this stage.

5. IMPLEMENTATION

5.1 SOURCE CODE

Admin page:

```
private javax.swing.JLabel lbl_manageempic;
private javax.swing.JPanel pnl_manage;public class Admin extends
javax.swing.JFrame {
String username,date,time;
Admin()

{
    initComponents();

 }
Admin(String text, String datetime, String time) {
            initComponents();
     username=text;
     date=datetime;
     this.time=time;

  }   private void initComponents() {
     jPanel1 = new javax.swing.JPanel();
     jPanel3 = new javax.swing.JPanel();
     jLabel10 = new javax.swing.JLabel();
     jLabel21 = new javax.swing.JLabel();
     lbl_logout = new javax.swing.JLabel();
     jPanel2 = new javax.swing.JPanel();
  jLabel1 = new javax.swing.JLabel();
     jLabel2 = new javax.swing.JLabel();

 adminName = new javax.swing.JLabel();
     jLabel4 = new javax.swing.JLabel();
     jLabel3 = new javax.swing.JLabel();
     jPanel11 = new javax.swing.JPanel();
     deleteReceptionbtn1 = new javax.swing.JLabel();
     deleteReceptionbtn3 = new javax.swing.JLabel();
     pnl_manage = new javax.swing.JPanel();
     lbl_manageemp = new javax.swing.JLabel();
     lbl_manageempic = new javax.swing.JLabel();

     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
     setUndecorated(true);
     addWindowListener(new java.awt.event.WindowAdapter() {
         public void windowClosed(java.awt.event.WindowEvent evt) {
             formWindowClosed(evt);
         }
     });

     jPanel1.setBackground(new java.awt.Color(255, 255, 255));

     jPanel3.setBackground(new java.awt.Color(255, 153, 51));

     jLabel10.setFont(new java.awt.Font("Arial", 0, 36)); // NOI18N
     jLabel10.setForeground(new java.awt.Color(255, 255, 255));
     jLabel10.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
     jLabel10.setText("ADMIN ");
```

```
        jLabel21.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                jLabel21MouseClicked(evt);
            }
        });

 lbl_logout.setIcon(new  javax.swing.ImageIcon(getClass().getResource("/icon/power
off.png"))); // NOI18N
        lbl_logout.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                lbl_logoutMouseClicked(evt);
            }
        });

        javax.swing.GroupLayout                    jPanel3Layout                    =                    new
javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                .addComponent(jLabel10, javax.swing.GroupLayout.PREFERRED_SIZE,
232, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(176, 176, 176)
                .addComponent(jLabel21)
                .addGap(203, 203, 203)
                .addComponent(lbl_logout,
javax.swing.GroupLayout.PREFERRED_SIZE,                                        67,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
 .addGap(22, 22, 22))
    );
    jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
        .addGroup(jPanel3Layout.createSequentialGroup()
          .addGap(11, 11, 11)

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
            .addComponent(lbl_logout,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addGroup(jPanel3Layout.createSequentialGroup()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
                .addComponent(jLabel10,
javax.swing.GroupLayout.PREFERRED_SIZE,                              59,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel21))
              .addGap(0, 0, Short.MAX_VALUE)))
          .addContainerGap())
    );

    jPanel2.setBackground(new java.awt.Color(255, 153, 51));

    jLabel1.setFont(new java.awt.Font("Arial", 1, 30)); // NOI18N
    jLabel1.setForeground(new java.awt.Color(255, 255, 255));
    jLabel1.setText("Welcome");
 jLabel2.setFont(new java.awt.Font("Tahoma", 0, 24)); // NOI18N
    jLabel2.setForeground(new java.awt.Color(255, 255, 255));

    adminName.setFont(new java.awt.Font("Arial", 0, 28)); // NOI18N
    adminName.setForeground(new java.awt.Color(255, 255, 255));

    jLabel4.setFont(new java.awt.Font("Arial Narrow", 0, 30)); // NOI18N
    jLabel4.setForeground(new java.awt.Color(255, 255, 255));
    jLabel4.setText("ADMIN");
    jLabel4.setToolTipText("");

    javax.swing.GroupLayout          jPanel2Layout          =          new
javax.swing.GroupLayout(jPanel2);
    jPanel2.setLayout(jPanel2Layout);
    jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
      .addGroup(jPanel2Layout.createSequentialGroup()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
          .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(74, 74, 74)

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
              .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(86, 86, 86)
                .addComponent(jLabel2))
              .addGroup(jPanel2Layout.createSequentialGroup()
            .addComponent(adminName)

    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel4,
    javax.swing.GroupLayout.PREFERRED_SIZE,                              80,
    javax.swing.GroupLayout.PREFERRED_SIZE))))
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(51, 51, 51)
                    .addComponent(jLabel1)))
            .addGap(53, 53, 53))
        );
        jPanel2Layout.setVerticalGroup(

    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
    G)
            .addGroup(jPanel2Layout.createSequentialGroup()

    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignme
    nt.LEADING)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(55, 55, 55)
                    .addComponent(jLabel1)
                    .addGap(35, 35, 35)
                    .addComponent(adminName)
                    .addGap(29, 29, 29))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
    jPanel2Layout.createSequentialGroup()
                    .addContainerGap()
                    .addComponent(jLabel4,
    javax.swing.GroupLayout.PREFERRED_SIZE,                              28,
    javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(18, 18, 18)))
```

```
 .addComponent(jLabel2)
          .addContainerGap(100, Short.MAX_VALUE))
     );

     jLabel3.setFont(new java.awt.Font("Arial Narrow", 0, 30)); // NOI18N
     jLabel3.setForeground(new java.awt.Color(255, 255, 255));
     jLabel3.setText("Doctor ");
     jLabel3.setToolTipText("");

     jPanel11.setBackground(new java.awt.Color(255, 153, 51));
     jPanel11.setPreferredSize(new java.awt.Dimension(150, 100));

     deleteReceptionbtn1.setFont(new java.awt.Font("Arial", 0, 25)); // NOI18N
     deleteReceptionbtn1.setForeground(new java.awt.Color(255, 255, 255));

deleteReceptionbtn1.setHorizontalAlignment(javax.swing.SwingConstants.CENTE
R);
     deleteReceptionbtn1.setText("Demographics");
     deleteReceptionbtn1.addMouseListener(new java.awt.event.MouseAdapter() {
       public void mouseClicked(java.awt.event.MouseEvent evt) {
         deleteReceptionbtn1MouseClicked(evt);
       }
     });

     deleteReceptionbtn3.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/report.png"))); // NOI18N
     deleteReceptionbtn3.addMouseListener(new java.awt.event.MouseAdapter() {
       public void mouseClicked(java.awt.event.MouseEvent evt) {
         deleteReceptionbtn3MouseClicked(evt);
       }
```

```
    });

        javax.swing.GroupLayout jPanel11Layout = new
javax.swing.GroupLayout(jPanel11);
        jPanel11.setLayout(jPanel11Layout);
        jPanel11Layout.setHorizontalGroup(

jPanel11Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
            .addComponent(deleteReceptionbtn1,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 242, Short.MAX_VALUE)
            .addGroup(jPanel11Layout.createSequentialGroup()
                .addGap(96, 96, 96)
                .addComponent(deleteReceptionbtn3)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
        jPanel11Layout.setVerticalGroup(

jPanel11Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADI
NG)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel11Layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(deleteReceptionbtn3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(deleteReceptionbtn1)
                .addContainerGap(24, Short.MAX_VALUE))
        );
```

```
 pnl_manage.setBackground(new java.awt.Color(255, 153, 51));
        pnl_manage.setPreferredSize(new java.awt.Dimension(150, 100));
        pnl_manage.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                pnl_manageMouseClicked(evt);
            }
        });

        lbl_manageemp.setFont(new java.awt.Font("Arial", 0, 25)); // NOI18N
        lbl_manageemp.setForeground(new java.awt.Color(255, 255, 255));
lbl_manageemp.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        lbl_manageemp.setText("Manage Employee");
        lbl_manageemp.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                lbl_manageempMouseClicked(evt);
            }
        });

        lbl_manageempic.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/icons8-add-48.png"))); //
NOI18N
        lbl_manageempic.addMouseListener(new java.awt.event.MouseAdapter() {
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                lbl_manageempicMouseClicked(evt);
            }
        });

        javax.swing.GroupLayout pnl_manageLayout = new
javax.swing.GroupLayout(pnl_manage);
```

```
 pnl_manage.setLayout(pnl_manageLayout);
        pnl_manageLayout.setHorizontalGroup(

pnl_manageLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
            .addComponent(lbl_manageemp,
javax.swing.GroupLayout.DEFAULT_SIZE, 225, Short.MAX_VALUE)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnl_manageLayout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                .addComponent(lbl_manageempic)
                .addGap(87, 87, 87))
        );
        pnl_manageLayout.setVerticalGroup(

pnl_manageLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
pnl_manageLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(lbl_manageempic)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(lbl_manageemp)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );


        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
        .addComponent(jPanel3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
              .addContainerGap()
              .addComponent(jLabel3,
javax.swing.GroupLayout.PREFERRED_SIZE, 80,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(jPanel1Layout.createSequentialGroup()
              .addGap(23, 23, 23)
              .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 70,
Short.MAX_VALUE)
              .addComponent(pnl_manage,
javax.swing.GroupLayout.PREFERRED_SIZE, 225,
javax.swing.GroupLayout.PREFERRED_SIZE)))
          .addGap(36, 36, 36)
          .addComponent(jPanel11, javax.swing.GroupLayout.PREFERRED_SIZE,
242, javax.swing.GroupLayout.PREFERRED_SIZE)
          .addGap(349, 349, 349))
    );
    jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADIN
G)
        .addGroup(jPanel1Layout.createSequentialGroup()
  .addContainerGap()
            .addComponent(jPanel3,  javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING)
               .addGroup(jPanel1Layout.createSequentialGroup()
                  .addGap(149, 149, 149)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignme
nt.LEADING, false)
                     .addComponent(jPanel11,
javax.swing.GroupLayout.DEFAULT_SIZE, 121, Short.MAX_VALUE)
                     .addComponent(pnl_manage,
javax.swing.GroupLayout.DEFAULT_SIZE, 121, Short.MAX_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
               .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,  78,
Short.MAX_VALUE)
                  .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addGap(52, 52, 52)))
            .addComponent(jLabel3,  javax.swing.GroupLayout.PREFERRED_SIZE,
28, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(67, 67, 67))
     );
```

```
 javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(layout.createSequentialGroup()
        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
    );

    pack();
  }// </editor-fold>

  private void jLabel21MouseClicked(java.awt.event.MouseEvent evt) {

  }

  private void deleteReceptionbtn1MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
}


    private void deleteReceptionbtn3MouseClicked(java.awt.event.MouseEvent evt) {
        // TODO add your handling code here:
    }


    private void lbl_manageempMouseClicked(java.awt.event.MouseEvent evt) {
         new AddEmp(username,date,time).setVisible(true);
        this.dispose();
    }


    private void lbl_manageempicMouseClicked(java.awt.event.MouseEvent evt) {
        new AddEmp(username,date,time).setVisible(true);
        this.dispose();
    }


    private void lbl_logoutMouseClicked(java.awt.event.MouseEvent evt) {
            String
connectString="jdbc:mysql://localhost:3307/hospitalms?zeroDateTimeBehavior=co
nvertToNull";
        Connection con = null;


        try {
            Class.forName("com.mysql.jdbc.Driver");
            con=DriverManager.getConnection(connectString,"root","");
             Statement st=con.createStatement();
           String qr="update login SET logged = '0' WHERE username =
'"+username+"'";
              st.executeUpdate(qr);
              SimpleDateFormat dateformat1 = new SimpleDateFormat("HH:mm:ss");
```

```
   String timeout=dateformat1.format(Calendar.getInstance().getTime());
           String qt="update log_times SET logout_time = '"+timeout+"' WHERE id
= '"+username+ "' AND date_login = '"+date+"' and login_time = '"+time+"'";
           st.executeUpdate(qt);
           con.close();
      } catch (ClassNotFoundException ex) {
         Logger.getLogger(Admin.class.getName()).log(Level.SEVERE, null, ex);
      } catch (SQLException ex) {
         Logger.getLogger(Admin.class.getName()).log(Level.SEVERE, null, ex);
      }
      finally
      {
         new Login().setVisible(true);
         this.dispose();
      }

   }

   private void pnl_manageMouseClicked(java.awt.event.MouseEvent evt) {
      new AddEmp(username,date,time).setVisible(true);
      this.dispose();
   }

   private void formWindowClosed(java.awt.event.WindowEvent evt) {

   }

   /**
    * @param args the command line arguments
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code
(optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look
and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Admin.class.getName()).log(java.util.logging.Lev
el.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Admin.class.getName()).log(java.util.logging.Lev
el.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Admin.class.getName()).log(java.util.logging.Lev
el.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Admin.class.getName()).log(java.util.logging.Lev
el.SEVERE, null, ex);
```

```
    }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Admin().setVisible(true);
            }
        });
    }
    private javax.swing.JLabel adminName;
    private javax.swing.JLabel deleteReceptionbtn1;
    private javax.swing.JLabel deleteReceptionbtn3;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel21;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel11;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel3;
    private javax.swing.JLabel lbl_logout;
    private javax.swing.JLabel lbl_manageemp;
    // End of variables declaration
}
```

5.2 Screen Shots

Fig 5.1 Login Page

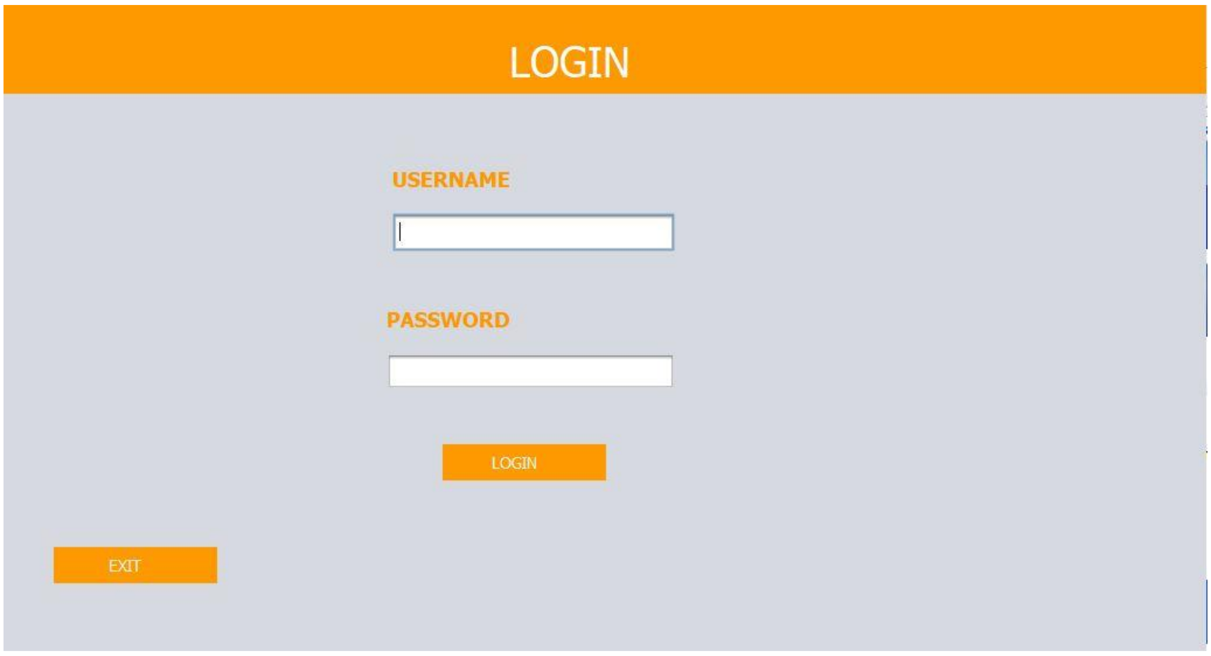


Fig 5.2 Admin Page

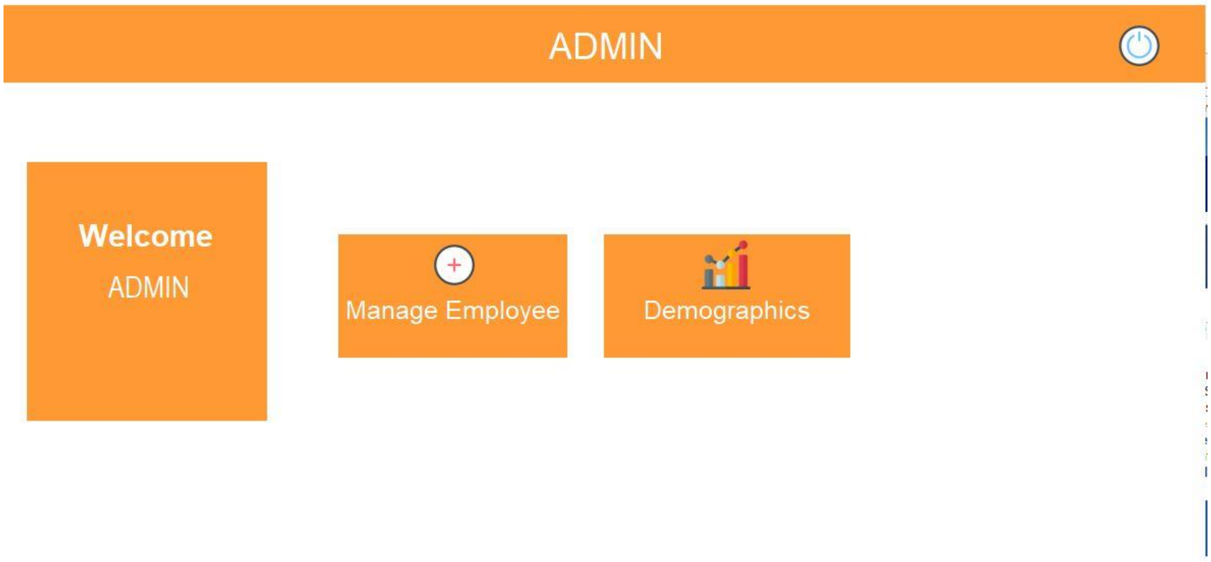
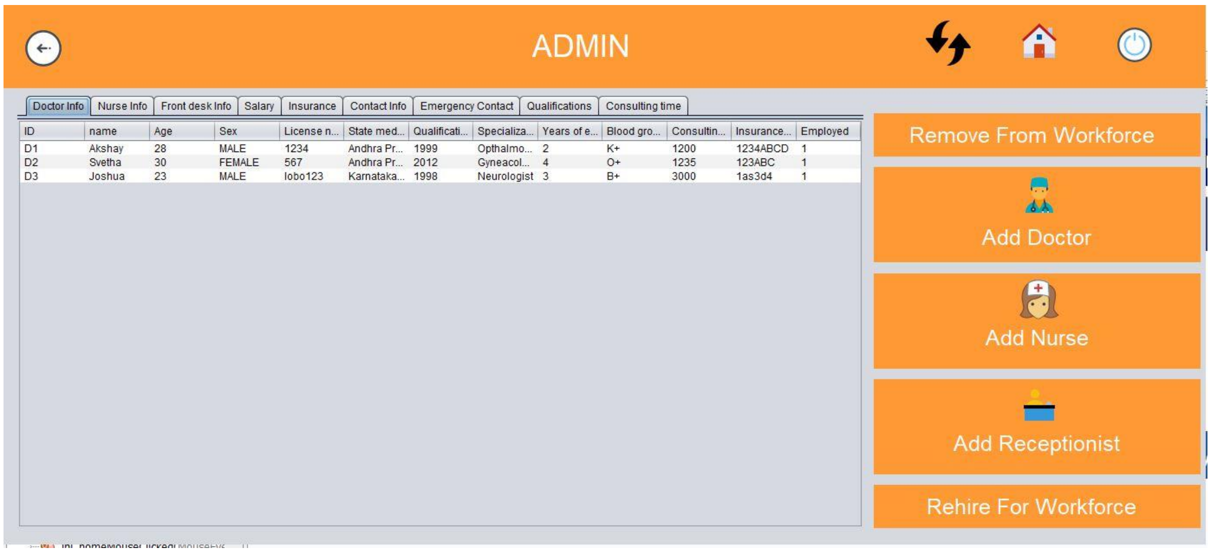
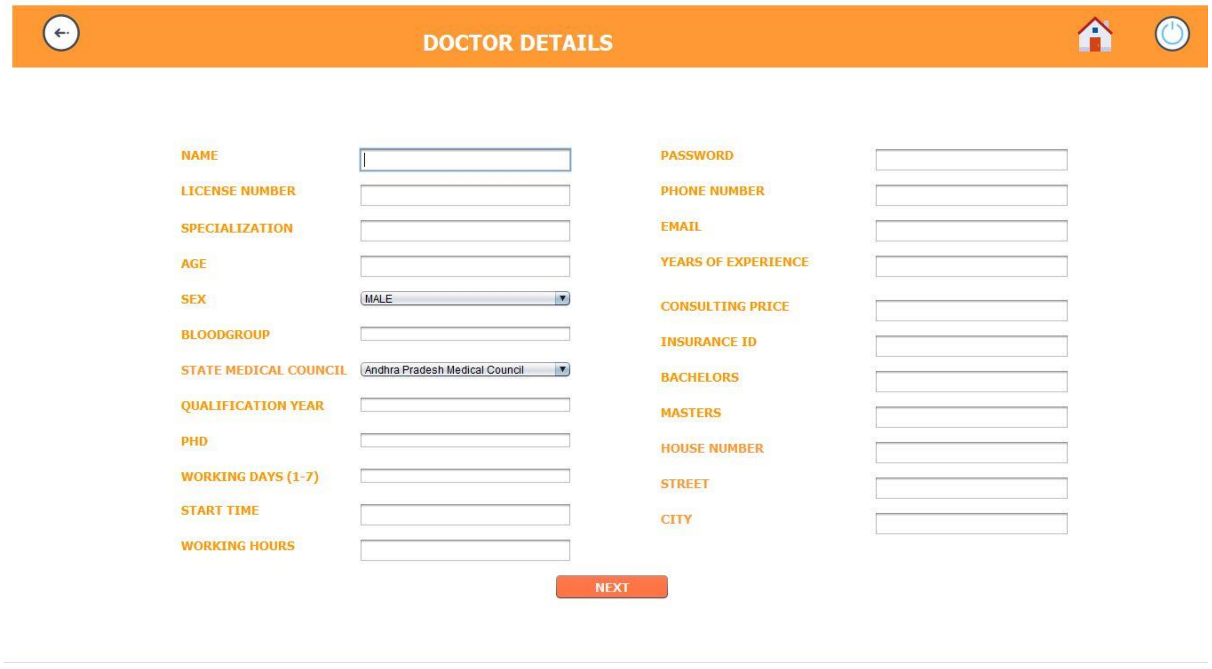


Fig 5.3 UserInterface of Admin



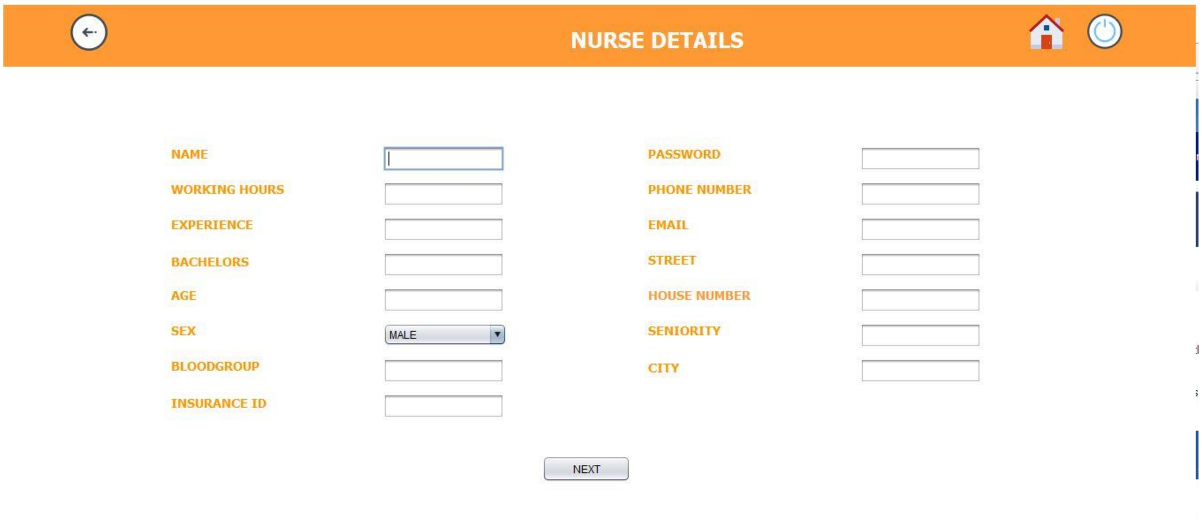
ID	name	Age	Sex	License n...	State med...	Qualificati...	Specializa...	Years of e...	Blood gro...	Consultin...	Insurance...	Employed
D1	Akshay	28	MALE	1234	Andhra Pr...	1999	Opthalmo...	2	K+	1200	1234ABCD	1
D2	Svetha	30	FEMALE	567	Andhra Pr...	2012	Gyneacol...	4	O+	1235	123ABC	1
D3	Joshua	23	MALE	lobo123	Karnataka...	1998	Neurologist	3	B+	3000	1as3d4	1

Fig 5.4 Doctor Details



NAME		PASSWORD	
LICENSE NUMBER		PHONE NUMBER	
SPECIALIZATION		EMAIL	
AGE		YEARS OF EXPERIENCE	
SEX	MALE	CONSULTING PRICE	
BLOODGROUP		INSURANCE ID	
STATE MEDICAL COUNCIL	Andhra Pradesh Medical Council	BACHELORS	
QUALIFICATION YEAR		MASTERS	
PHD		HOUSE NUMBER	
WORKING DAYS (1-7)		STREET	
START TIME		CITY	
WORKING HOURS			

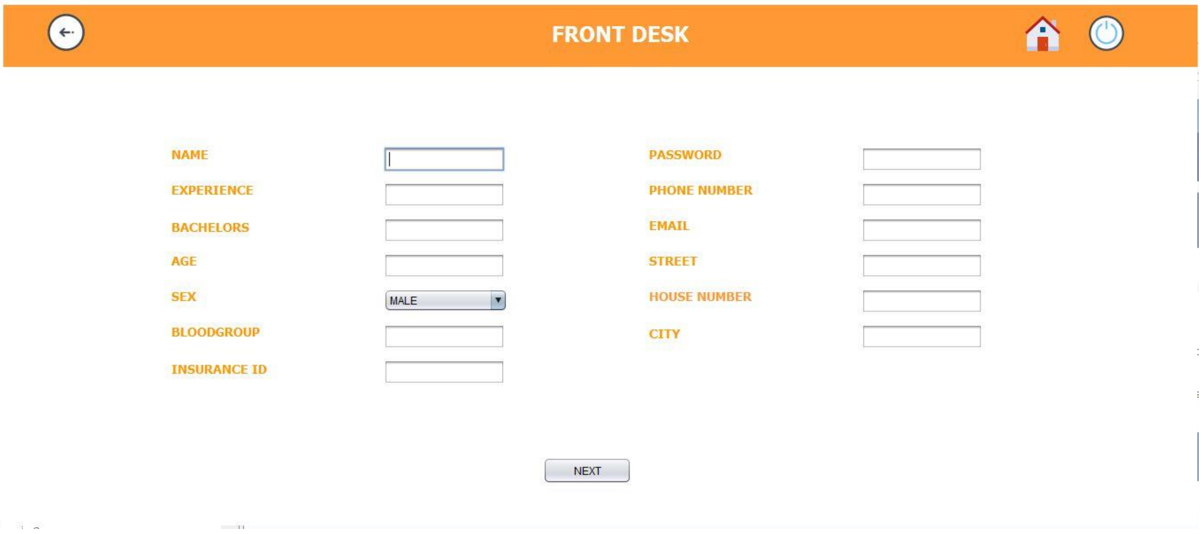
Fig 5.5 Nurse Details


NURSE DETAILS

NAME		PASSWORD	
WORKING HOURS		PHONE NUMBER	
EXPERIENCE		EMAIL	
BACHELORS		STREET	
AGE		HOUSE NUMBER	
SEX	MALE	SENIORITY	
BLOODGROUP		CITY	
INSURANCE ID			

NEXT

Fig 5.6 Front Desk Details


FRONT DESK

NAME		PASSWORD	
EXPERIENCE		PHONE NUMBER	
BACHELORS		EMAIL	
AGE		STREET	
SEX	MALE	HOUSE NUMBER	
BLOODGROUP		CITY	
INSURANCE ID			

NEXT

Fig 5.7 Demographics of Expenses

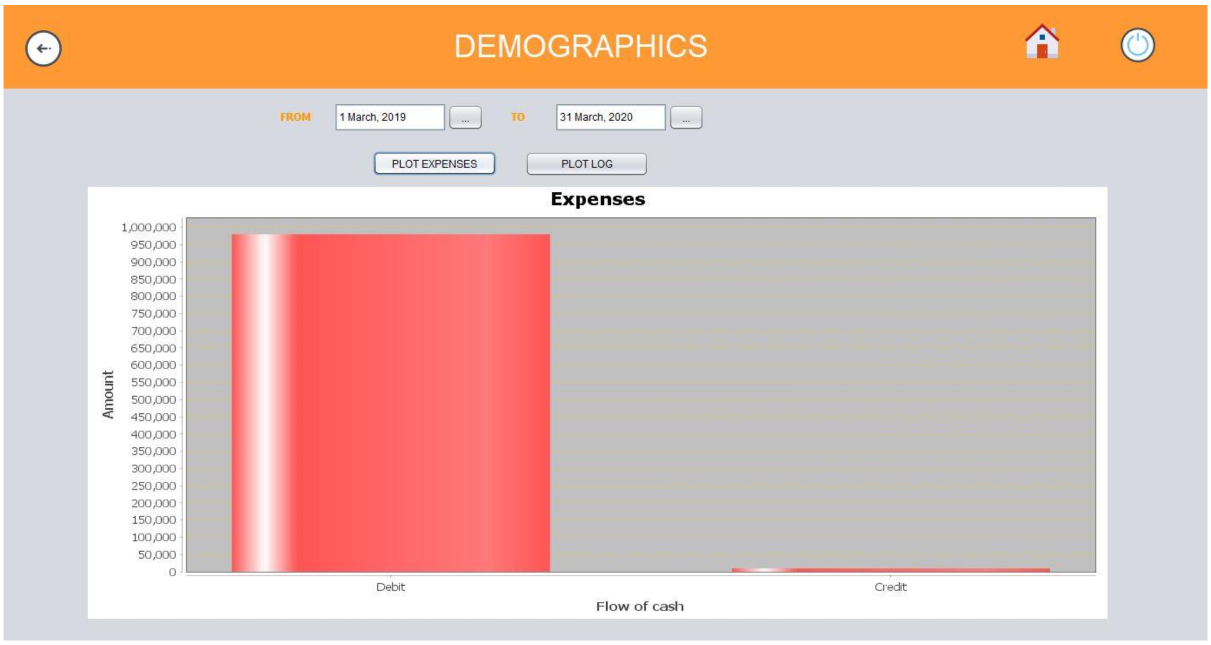


Fig 5.8 Demoegraphics of Log Times

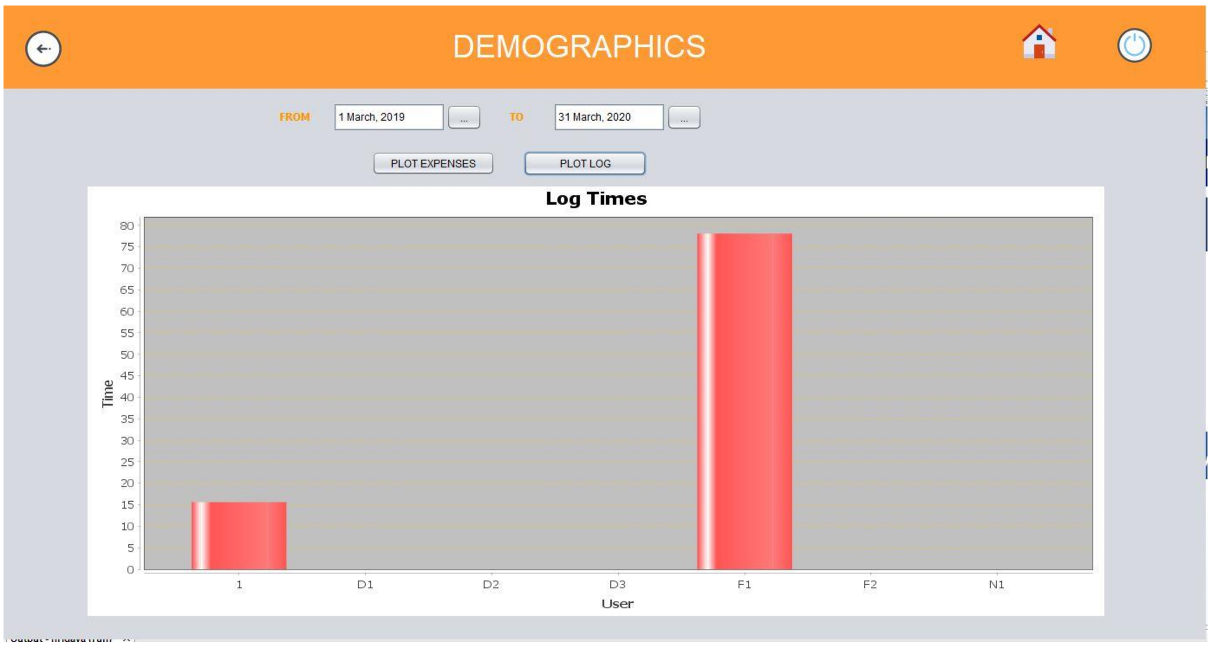


Fig 5.9 FrontDesk HomePage

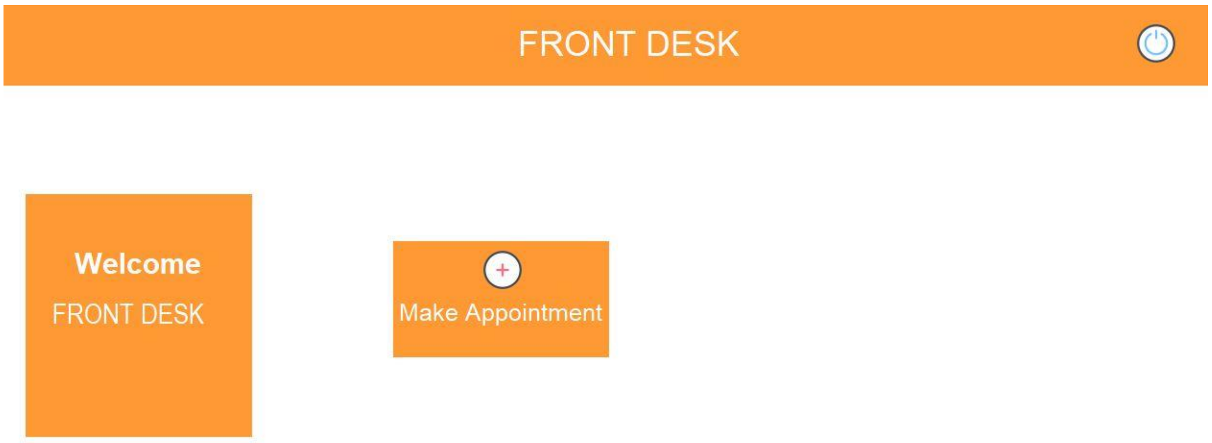
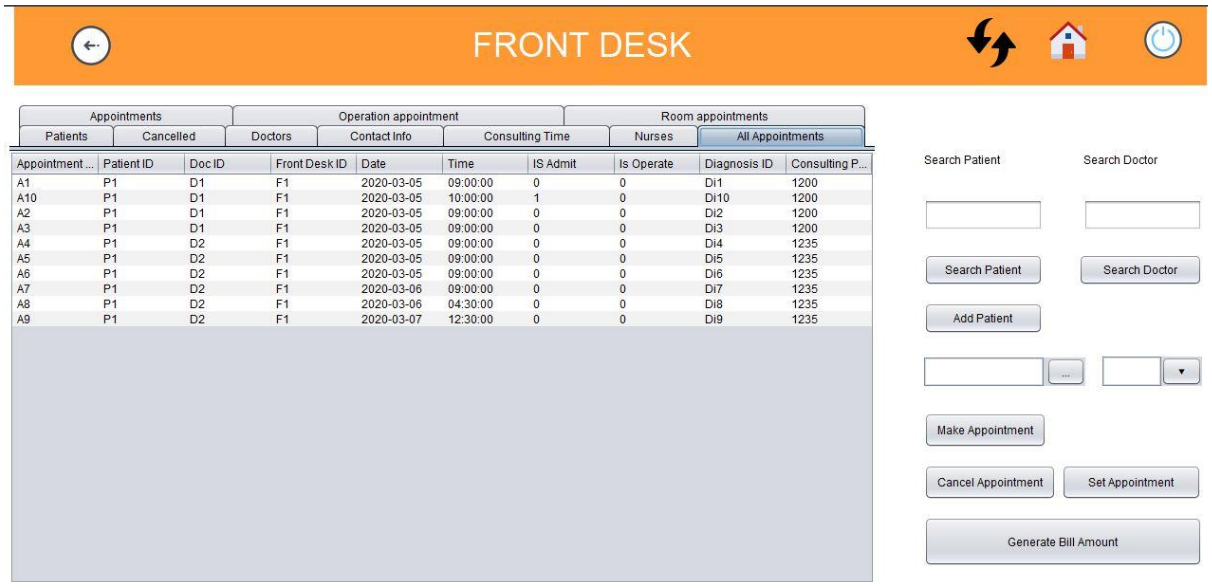
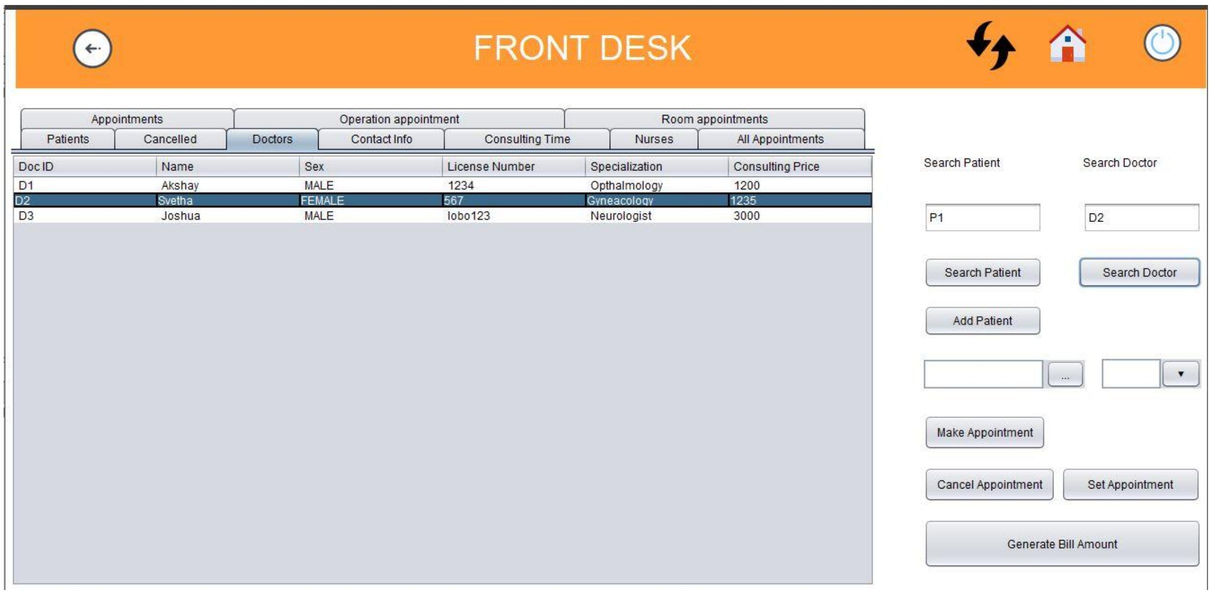


Fig 5.10 Front Desk Details



Appointment ...	Patient ID	Doc ID	Front Desk ID	Date	Time	IS Admit	Is Operate	Diagnosis ID	Consulting P...
A1	P1	D1	F1	2020-03-05	09:00:00	0	0	Di1	1200
A10	P1	D1	F1	2020-03-05	10:00:00	1	0	Di10	1200
A2	P1	D1	F1	2020-03-05	09:00:00	0	0	Di2	1200
A3	P1	D1	F1	2020-03-05	09:00:00	0	0	Di3	1200
A4	P1	D2	F1	2020-03-05	09:00:00	0	0	Di4	1235
A5	P1	D2	F1	2020-03-05	09:00:00	0	0	Di5	1235
A6	P1	D2	F1	2020-03-05	09:00:00	0	0	Di6	1235
A7	P1	D2	F1	2020-03-06	09:00:00	0	0	Di7	1235
A8	P1	D2	F1	2020-03-06	04:30:00	0	0	Di8	1235
A9	P1	D2	F1	2020-03-07	12:30:00	0	0	Di9	1235

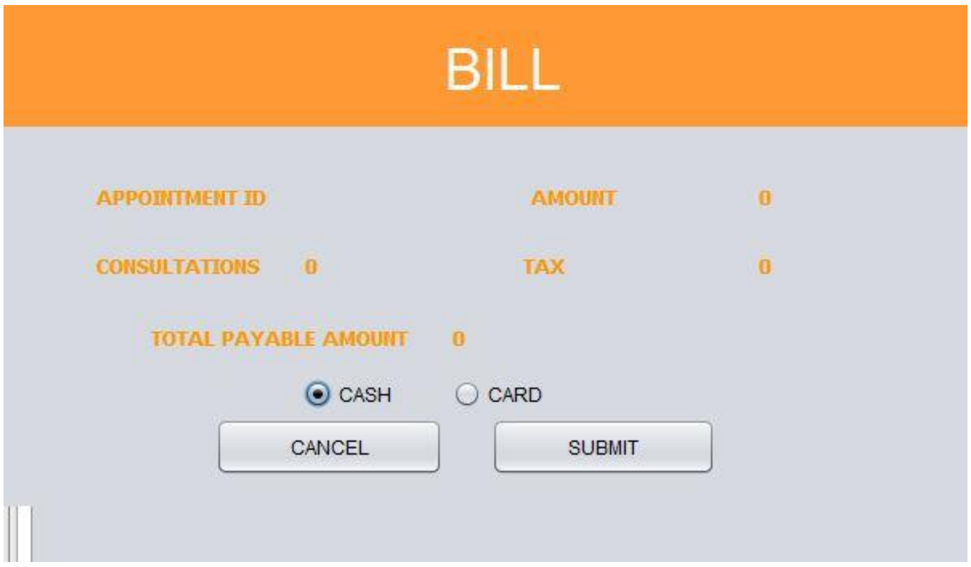
Fig 5.11 Front Desk UserInterface



Doc ID	Name	Sex	License Number	Specialization	Consulting Price
D1	Akshay	MALE	1234	Opthalmology	1200
D2	Svetha	FEMALE	567	Gyneacology	1235
D3	Joshua	MALE	lobo123	Neurologist	3000

Search Patient: P1
Search Doctor: D2

Fig 5.12 Billing Page



APPOINTMENT ID
AMOUNT 0
CONSULTATIONS 0
TAX 0
TOTAL PAYABLE AMOUNT 0
[✓ CASH] CARD

Fig 5.13 Doctor Panel

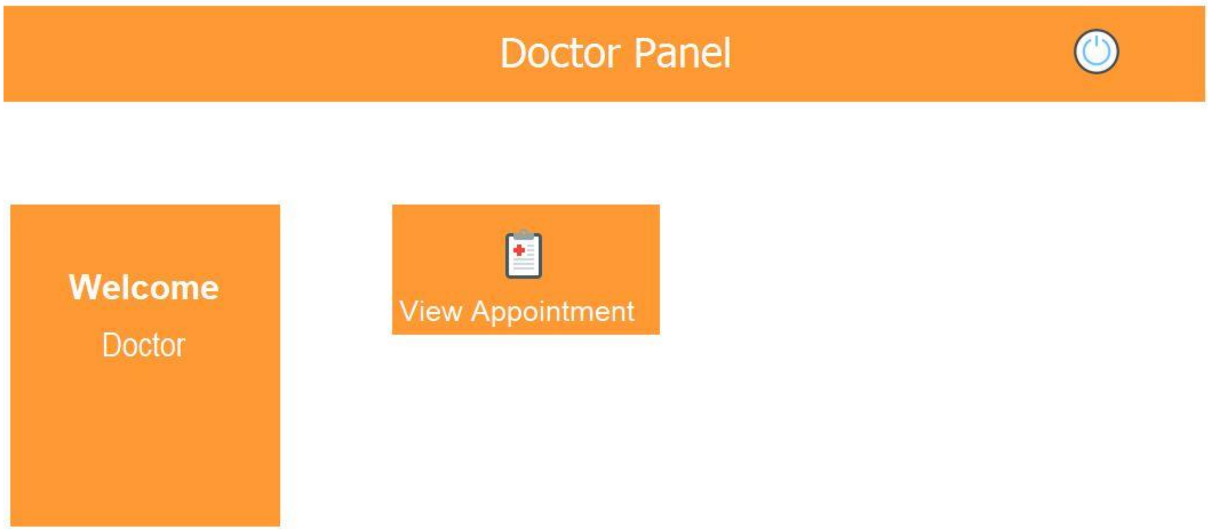
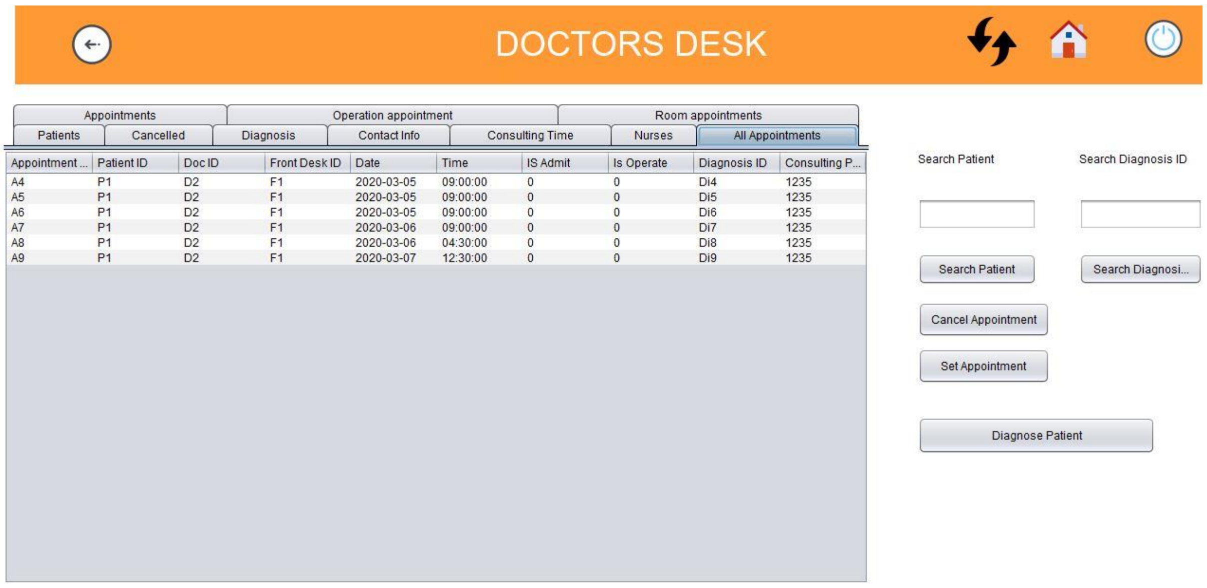
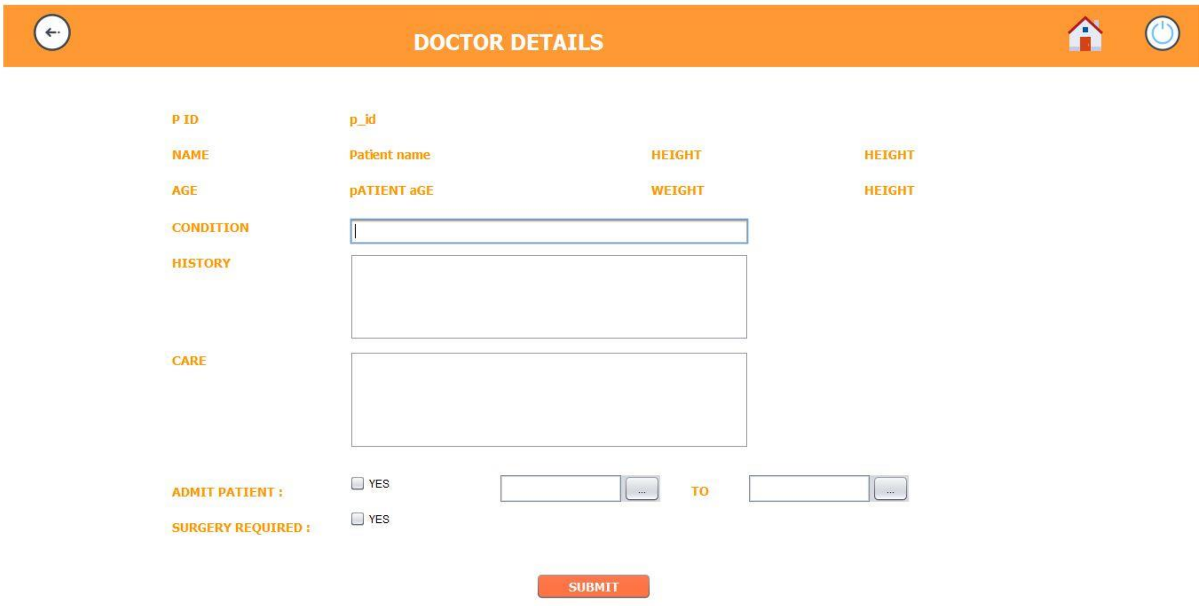


Fig 5.14 Doctor Desk



Appointment ...	Patient ID	Doc ID	Front Desk ID	Date	Time	IS Admit	Is Operate	Diagnosis ID	Consulting P...
A4	P1	D2	F1	2020-03-05	09:00:00	0	0	Di4	1235
A5	P1	D2	F1	2020-03-05	09:00:00	0	0	Di5	1235
A6	P1	D2	F1	2020-03-05	09:00:00	0	0	Di6	1235
A7	P1	D2	F1	2020-03-06	09:00:00	0	0	Di7	1235
A8	P1	D2	F1	2020-03-06	04:30:00	0	0	Di8	1235
A9	P1	D2	F1	2020-03-07	12:30:00	0	0	Di9	1235

Fig 5.15 Doctor Details



P ID	p_id		
NAME	Patient name	HEIGHT	HEIGHT
AGE	pATIENT aGE	WEIGHT	HEIGHT
CONDITION			
HISTORY			
CARE			
ADMIT PATIENT :	☐ YES		TO
SURGERY REQUIRED :	☐ YES		

6.CONCLUSION

Hospitals are an extremely vital part of any society. They are, in essence, a complex structure of different groups of people performing various tasks to ensure its efficient working. Yet there is a lack of software that could economically ensure that all of the different departments of can function efficiently. The application provides an interface and functionality for every part of the hospitals multi-faceted environment. From storing patients records to making and modifying appointments this system facilitates better functioning of the Hospital.

6.1 ADVANTAGES

- A system that can replace the manual hospital management software.
- A database which stores patient details along with details of hospital staff.
- Reliable appointment making facility.
- Efficient handling of large amounts of data (Pharmacy and appointment).
- Doctor, Nurse, Front desk and pharmacy have a login.
- An easy to understand user friendly software.
- Attractive user interfaces to navigate through the system for the users.
- Visualization of stored data to gain meaningful insights.
- Emergency team readiness monitor.
- Facility to keep track of finances of the hospital

6.2 DISADVANTAGES

- Use of redundant systems will result in a greater cost.
- It may be a bit confusing for unfamiliar users.
- There is an uncertain building department review and approval process.
- It requires high maintenance
- Database –The system use the MySQL Database ,which is an open source and free
- Operating System –The System environment shall be only in Windows OS
- Desktop Application-The System shall be a Desktop Application

6.3 FUTURE ENHANCEMENT

The proposed system is hospital management system.We can enhance this system by including more facilities like pharmacy system for the stock details of medicines in the pharmacy and to display the details of patients for individual doctor .Providing such features enable the users to include more comments into the system