Mini Project Report on

“Hospital Management System”

Submitted by

PD-14 Aniruddha Gangthade	ERP Number: 1032180331
PD-20 Gauri Godghase	ERP Number: 1032180427
PD-28 Aishwarya Gautam	ERP Number: 1032180710
PD-44 Amrita Ghosh	ERP Number: 1032181219

Under the Guidance of

Mrs. Pradnya Kulkarni

At

MIT-WPU
॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

TABLE OF CONTENTS

I Abstract

The purpose of the project entitled as HOSPITAL MANAGEMENT SYSTEM is to computerize the Front Office Management of Hospital to develop software which is user friendly simple, fast, and cost-effective. It deals with the collection of patient's information, diagnosis details, etc. Traditionally, it was done manually. The main function of the system is register and store patient details and doctor details and retrieve these details as and when required, and also to manipulate these details meaningfully. System input contains patient details, diagnosis details, while system output is to get these details on to the screen. The Hospital Management System can be entered using a username and password. It is accessible either by an administrator or doctor. Only they can add data into the database. The data can be retrieved easily. The data are well protected for personal use and makes the data processing very fast.

II LIST OF ABBREVIATIONS

HMS->Hospital Management System
SRS ->Software requirements specification
PC -> Personal Computer
HDD -> Hard Disc Drive
RAM ->Random Access Memory
IE ->Microsoft Internet Explorer
JAVA -> platform independence
SQL -> Structured query Language
ER-> Entity Relationship
UML ->Unified Modeling Language
IDE -> Integrated Development Environment
DDL -> Data Definition Language
DML ->Data Manipulation Language
DCL ->Data Control Language
TCL ->Transaction Control Language
JDBC ->Java Database Connectivity
IEEE-> Institute of Electrical and Electronics Engineers

III LIST OF FIGURES

IV LIST OF TABLES

1. Introduction

Hospital are the essential part of our lives providing best medical facilities to people suffering from various ailments. which may be due to change in climatic conditions, increased work-load, emotional trauma, stress etc. It is necessary for the hospitals to keep track of its day-to-day activities & records of its patients, doctors, nurses, ward boys and other staff personals that keep the hospital running smoothly & successfully.

But keeping track of all the activities and their records on paper is very cumbersome and error prone. It also is very inefficient and a time-consuming process. Observing the continuous increase in population and number of people visiting the hospital recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper.

Thus keeping the working of the manual system as the basis of our project, we have developed an automated version of the manual system, named as “Hospital Management System”.

The main aim of our project is to provide a paper-less hospital up to 90%. It also aims at providing low-cost reliable automation of the existing systems. The system also provides excellent security of data at every level of user-system interaction and also provides robust & reliable storage and backup facilities.

1.1 Motivation

- To meet a solution to manage medical records of patients across branches of a hospital in various districts
- To overcome existing problems occurring in recording of information.

1.2 Objectives of the system:

The project “Hospital management system” is aimed to develop to maintain the day-to-day state of admission/discharge of patients, list of doctors, reports generation, etc. It is designed to achieve the following objectives:

1. To computerize all details regarding patient details & hospital details.

2. It should be able to handle the test reports of patients conducted in the pathology lab of the hospital.

3. The information of the patients should be kept up to date and there record should be kept in the system for historical purposes.

2. Problem Definition

- The information is very difficult to retrieve and to find particular information like- to find out about the patient's history, the user has to go through various registers.
- Various changes to information, like patient details are difficult to make, as paper work is involved.
- Preparation of accurate and prompt report: - This becomes a difficult task as information is difficult to collect from various registers.
- So an attempt has been made to automate these tasks and minimize the workload as far as possible by a proper coordination between various modules and functionalities of a database.

2.1 Tools and Technologies Used

Hardware Requirements:

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application.

HARDWARE REQUIREMENTS FOR PRESENT PROJECT:

- PROCESSOR : Intel dual Core ,i3
- RAM : 1 GBHARD
- DISK : 80 GB

Software Requirements:

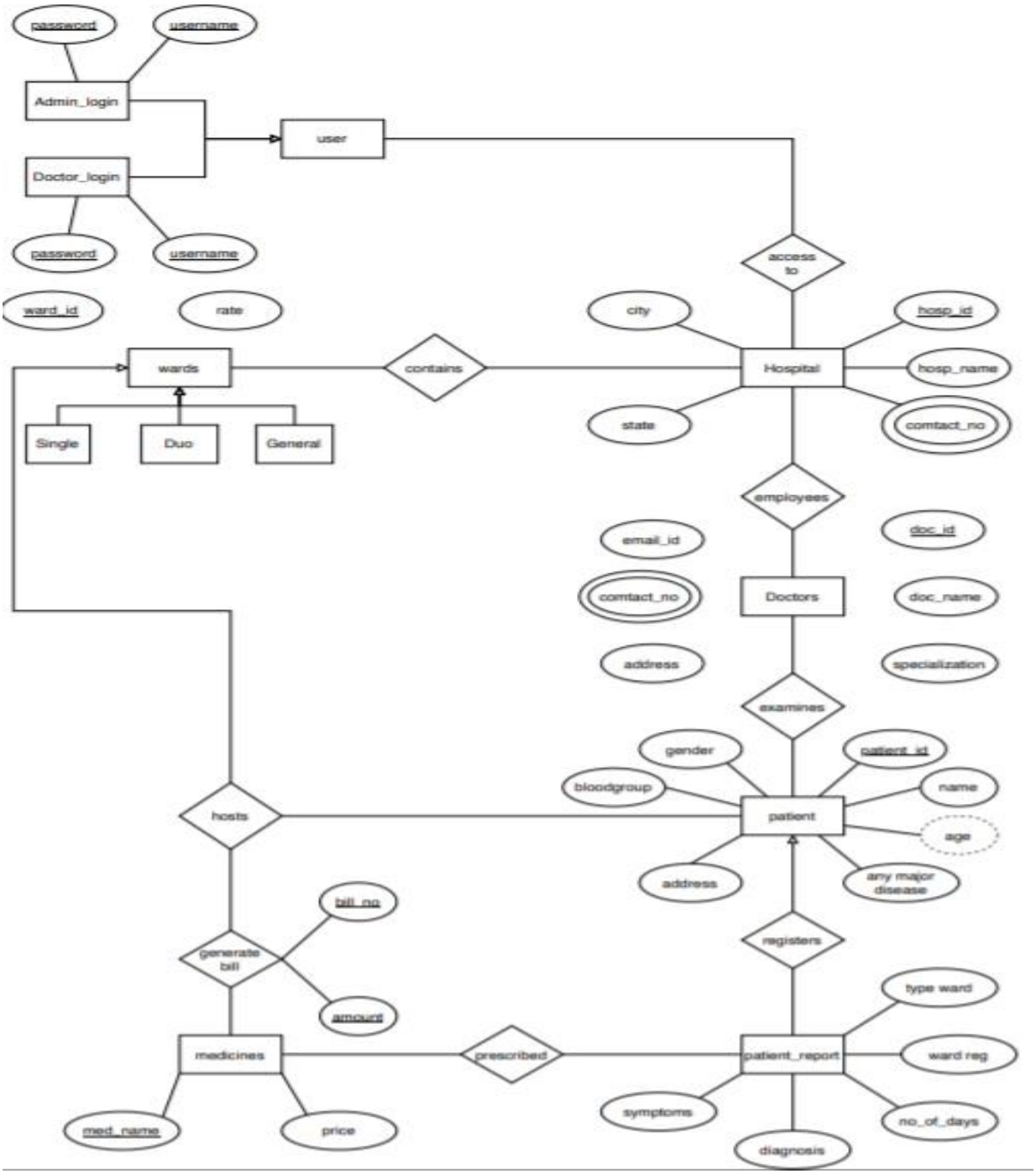
Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

SOFTWARE REQUIREMENTS FOR PRESENT PROJECT:

- OPERATING SYSTEM : Windows 7/ XP/8/10
- FRONT END : Java NetBeans/Eclipse IDE
- DATABASE : MySQL

3.Database Design

3.1 Entity Relationship Diagram

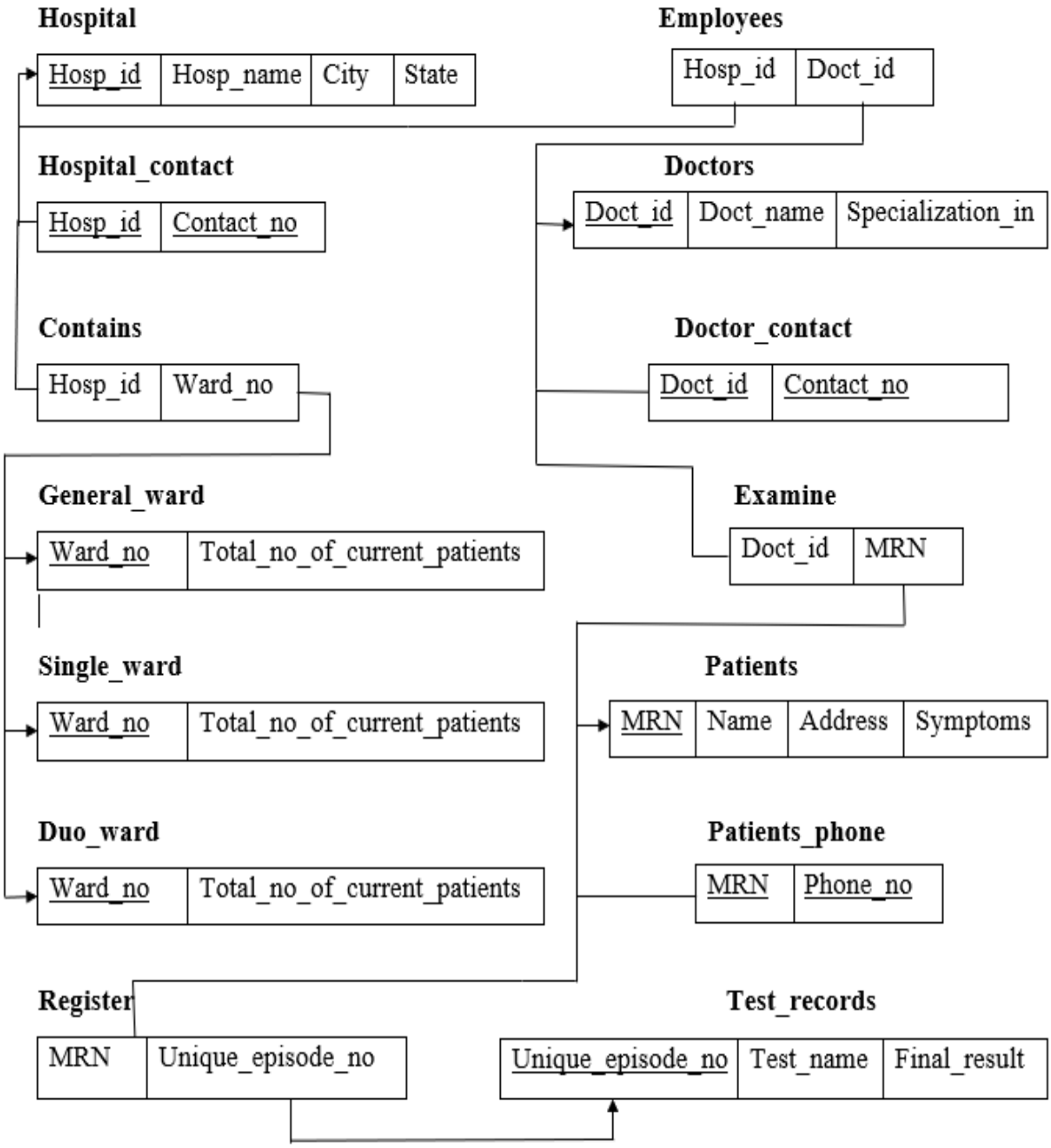


3. Database Schema

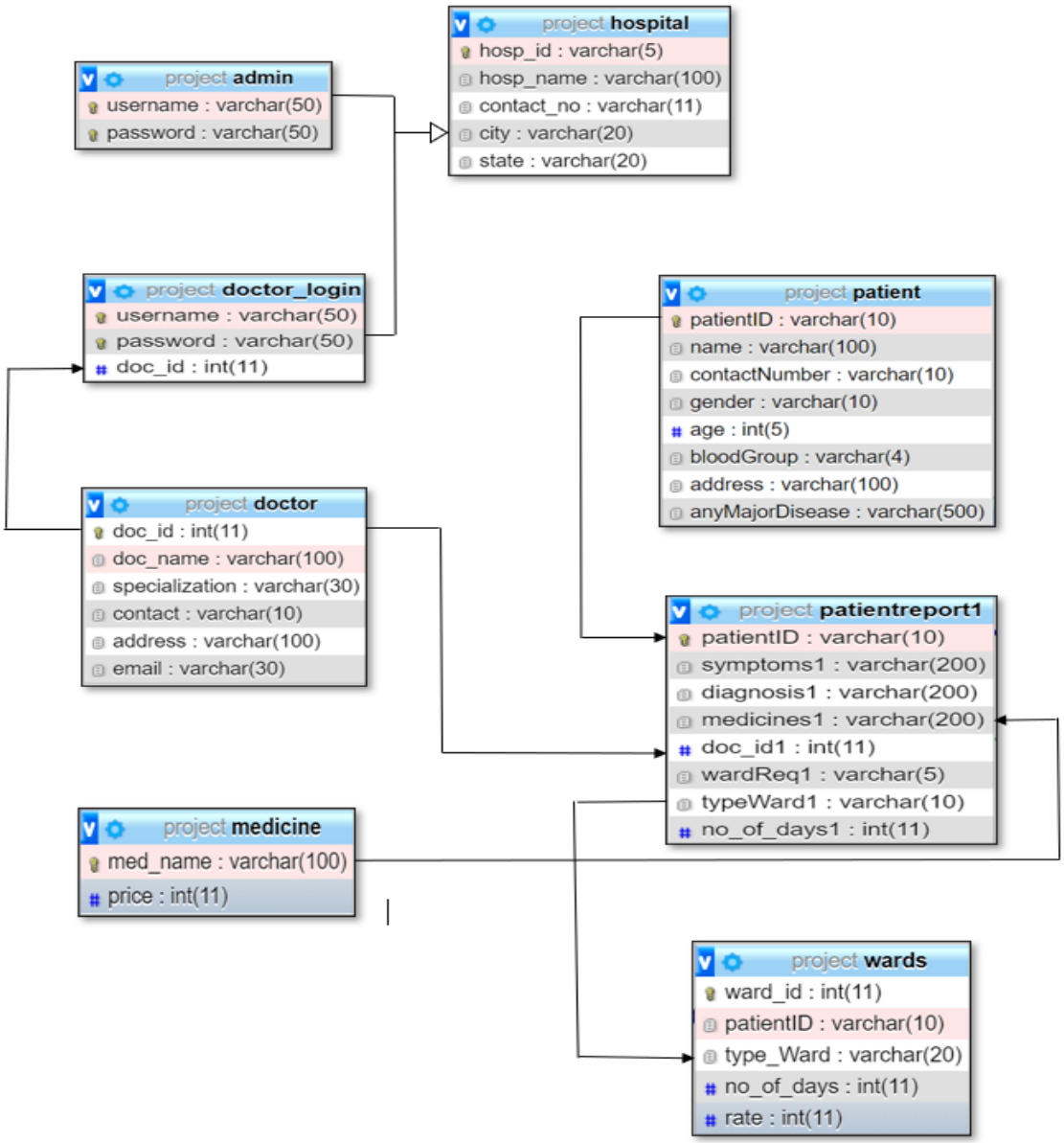
4.1 Relational Schemas

- Hospital (Hosp_id , Hosp_name , City , State)
  Hospital table has multi-valued attribute contact_no.
- Hospital_contact (Hosp_id , contact_no)
- General_ward (ward_id , rate, no_of_days, patientID)
- Single_ward (ward_id , rate, no_of_days, patientID)
- Duo_ward (ward_id , rate, no_of_days, patientID)
- Patients (patientID , Name , Address , Age , Gender, bloodGroup, anyMajorDisease)
  Patients table has multi-valued attribute set_of_phone.
- Patients_phone (patientID , set_of_phone)
- Patientreport(patientID, symptoms, diagnosis, medicines, doc_id, wardReq, typeWard, no_of_days)
- Doctors (doc_id , doct_name , specialization_in, email, address)
  Contact_no is a multi-valued attribute.
- Doctor_contact (doct_id , contact_no)
- Doctor_login(username, password)
- Medicine(med_name, price)

3.2 Relational Schema Diagram



4. Database Design using Schema Diagram



5. Database Normalization Till 3 NF

Hospital database (Primary key : hosp_id)

Hosp_id	Hosp_name	Contact_no	City	State
H100	Fortis Healthcare	8130000000 , 8527000000	Noida , Okhla Road	Delhi
H101	Fortis Multispeciality clinic	8917890000 , 7829100000 , 9675100000	Cunningham Road , Richmond Town , Nagarbhavi	Karnataka
H102	Fortis Medical Hospital	6783120000 , 7890300000 , 6729000000	Mulund , Mahim , Kalyan	Maharashtra

Patients database (Primary key : MRN)

MRN	Name	Phone_no	Address	Symptoms
P500	Jay Agarwal	9651423500 , 7895400000	A101 , Cannaught Place , Delhi	Cough , fever , mouth ulcers
P501	Rajib Soni	9276510000	64 , Janpath ,Delhi	Chest pain , Fatigue , Loss of appetite , Sweating
P502	Anwesha Singh	9783200000	MS Building , Karnataka	Blurred vision , Seizures , Difficulty speaking
P503	Riya Roy	8910927169 , 9840108869	Willow Towers , Kalyan	Abdominal cramps , Bleeding

6.1 1 NF Form

- All the attributes in a relation must have atomic domains.
- The values in an atomic domain must be indivisible units.

The hospital and patients table contain multi-valued attributes for the column City and contact_no in hospital table and for the columns Phone_no and Symptoms in the patients table.

So after 1 NF on them the table of hospital becomes :

Hospital database (Primary key : Hosp_id , City)

Hosp_id	Hosp_name	City	State
H100	Fortis Healthcare	Noida	Delhi
H100	Fortis Healthcare	Okhla Road	Delhi
H101	Fortis Multispeciality clinic	Cunningham Road	Karnataka
H101	Fortis Multispeciality clinic	Richmond Town	Karnataka
H101	Fortis Multispeciality clinic	Nagarbhavi	Karnataka
H102	Fortis Medical Hospital	Mulund	Maharashtra
H102	Fortis Medical Hospital	Mahim	Maharashtra
H102	Fortis Medical Hospital	Kalyan	Maharashtra

Primary Key : Hosp_id , contact_no

Hosp_id	Contact_no
H100	8130000000
H100	8527000000
H101	8917890000
H101	7829100000
H101	9675100000
H102	6783120000
H102	7890300000
H102	6729000000

After 1 NF on the patients table it becomes :

Patients database (Primary key : MRN , Symptoms)

MRN	Name	Address	Symptoms
P500	Jay Agarwal	A101 , Cannaught Place , Delhi	Cough
P500	Jay Agarwal	A101 , Cannaught Place , Delhi	fever
P500	Jay Agarwal	A101 , Cannaught Place , Delhi	mouth ulcers
P501	Rajib Soni	64 , Janpath ,Delhi	Chest pain
P501	Rajib Soni	65 , Janpath ,Delhi	Fatigue
P501	Rajib Soni	66 , Janpath ,Delhi	Loss of appetite
P501	Rajib Soni	67 , Janpath ,Delhi	Sweating
P502	Anwesha Singh	MS Building , Karnataka	Blurred vision
P502	Anwesha Singh	MS Building , Karnataka	Seizures
P502	Anwesha Singh	MS Building , Karnataka	Difficulty Speaking
P503	Riya Roy	Willow Towers , Kalyan	Abdominal cramps
P503	Riya Roy	Willow Towers , Kalyan	Bleeding

Primary Key : MRN , Phone_no

MRN	Phone_no
P500	9651423500
P500	7895400000
P501	9276510000
P502	9783200000
P503	9840108869
P503	8910927169

6.2 2 NF FORM

- For a relation to be in Second Normal Form, it must be in First Normal form and every non-primary-key attribute is fully functionally dependent on the primary key.
- The normalization of 1NF relations to 2NF involves the removal of partial dependencies. If a partial dependency exists, we remove the function dependent attributes from the relation by placing them in a new relation along with a copy of their determinant.

In hospital table, Prime attributes are Hosp_id and city.
Non-prime attributes are contact_no , state and Hosp_name.
Hosp_id → Hosp_name (Partial dependency)
City → State (Partial dependency)

After 2 NF on hospital table :

Hospital database

Primary key : Hosp_id , City

Hosp_id	City
H100	Noida
H100	Okhla Road
H101	Cunningham Road
H101	Richmond Town
H101	Nagarbhavi
H102	Mulund
H102	Mahim
H102	Kalyan

Primary Key : Hosp_id , contact_no

Hosp_id	Contact_no
H100	8130000000
H100	8527000000
H101	8917890000
H101	7829100000
H101	9675100000
H102	6783120000
H102	7890300000
H102	6729000000

Primary key : Hosp_id

Hosp_id	Hosp_name
H100	Fortis Healthcare
H101	Fortis Multispeciality clinic
H102	Fortis Medical Hospital

Primary key : City

City	State
Noida	Delhi
Okhla Road	Delhi
Cunningham Road	Karnataka
Richmond Town	Karnataka
Nagarbhavi	Karnataka
Mulund	Maharashtra
Mahim	Maharashtra
Kalyan	Maharashtra

6.3.3 NF FORM

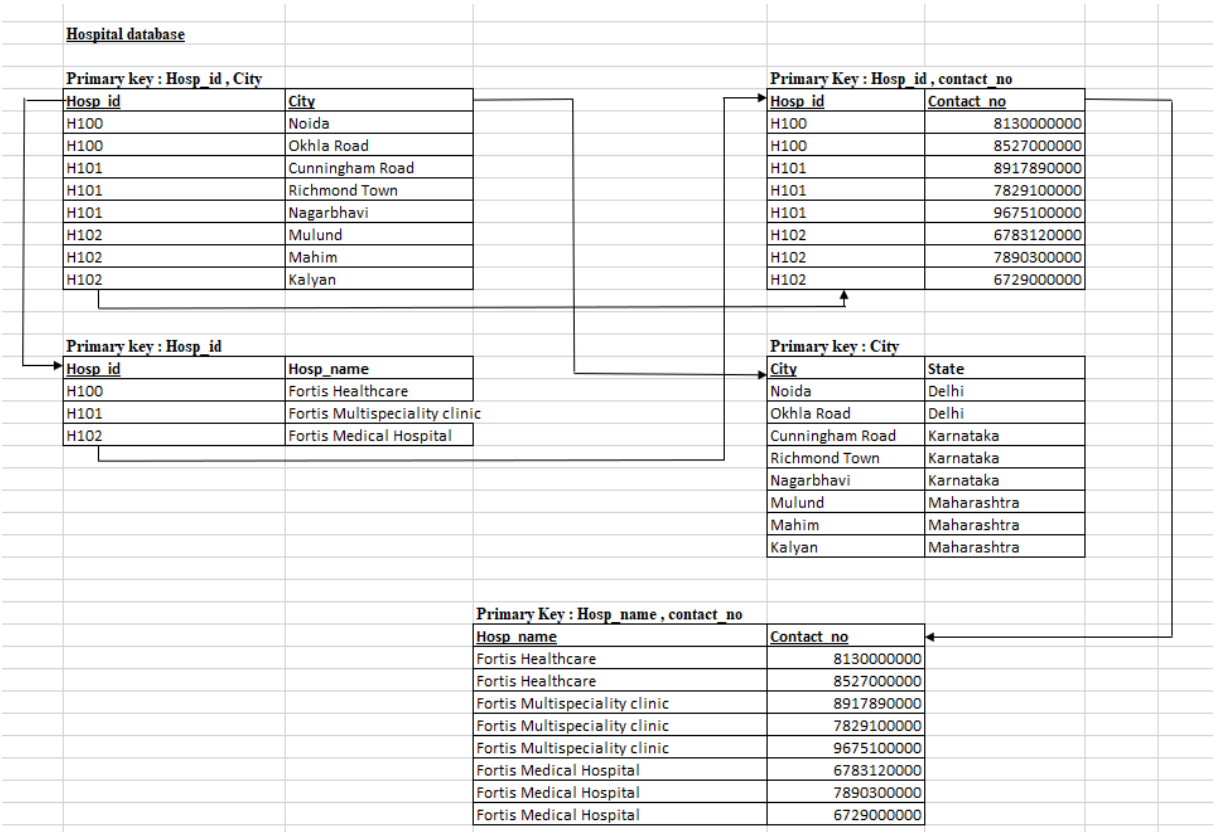
For a relation to be in Third Normal Form, it must be in Second Normal form and the following must satisfy −

- No non-prime attribute is transitively dependent on prime key attribute.
- For any non-trivial functional dependency, X → A, then either
  - X is a super key or
  - A is prime attribute.

In hospital table ,

Hosp_id → Hosp_name
Hosp_name → State (Transitive dependency)
Hosp_name → Contact_no (Transitive dependency)

Applying 3 NF to remove the transitive dependency on the hospital table :



Hospital database

Primary key : Hosp_id , City

Hosp_id	City
H100	Noida
H100	Okhla Road
H101	Cunningham Road
H101	Richmond Town
H101	Nagarbhavi
H102	Mulund
H102	Mahim
H102	Kalyan

Primary Key : Hosp_id , contact_no

Hosp_id	Contact_no
H100	8130000000
H100	8527000000
H101	8917890000
H101	7829100000
H101	9675100000
H102	6783120000
H102	7890300000
H102	6729000000

Primary key : Hosp_id

Hosp_id	Hosp_name
H100	Fortis Healthcare
H101	Fortis Multispeciality clinic
H102	Fortis Medical Hospital

Primary key : City

City	State
Noida	Delhi
Okhla Road	Delhi
Cunningham Road	Karnataka
Richmond Town	Karnataka
Nagarbhavi	Karnataka
Mulund	Maharashtra
Mahim	Maharashtra
Kalyan	Maharashtra

Primary Key : Hosp_name , contact_no

Hosp_name	Contact_no
Fortis Healthcare	8130000000
Fortis Healthcare	8527000000
Fortis Multispeciality clinic	8917890000
Fortis Multispeciality clinic	7829100000
Fortis Multispeciality clinic	9675100000
Fortis Medical Hospital	6783120000
Fortis Medical Hospital	7890300000
Fortis Medical Hospital	6729000000

6. DDL

Data Definition Language (DDL) refers to the CREATE, ALTER and DROP statements.

DDL or Data Definition Language actually consists of the SQL commands that can be used to define the database schema. DDL allows to add / modify / delete the logical structures which contain the data or which allow users to access / maintain the data (databases, tables, keys, views...). DDL is about "metadata".

•CREATE TABLE - creates a new database table

• ALTER TABLE - alters (changes) a database table

• DROP TABLE - deletes a database table

• TRUNCATE - cleans all data

• RENAME- renames a table name

Sample of DDL commands used in our database :

```
create table admin
(username varchar(50),
password varchar(50));

/* adding primary keys */
ALTER TABLE `admin` ADD PRIMARY KEY( `username`, `password`);


create table doctor_login
(username varchar(50) not null,
 password varchar(50) not null,
 primary key(username,password));


create table hospital
(hosp_id varchar(5) primary key,
 hosp_name varchar(100),
 contact_no varchar(11),
 city varchar(20),
 state varchar(20));
```

```
/* patient entry in patient table*/
create table patient
(patientID varchar(10) primary key,
 name varchar(100) not null,
 contactNumber varchar(10),
 gender varchar(10),
 age int,
 bloodGroup varchar(4),
 address varchar(100),
 anyMajorDisease varchar(500) default null);

alter table `patient` modify age varchar(5) check(age>0);


/*doctor table*/
create table doctor
(doc_id integer primary key,
 doc_name varchar(100) not null,
 specialization varchar(30) not null ,
 contact varchar(10) ,
 address varchar(100),
 email varchar(30));


create table medicine
(med_name varchar(100) primary key,
price integer);

/* patient diagnosis information */
create table patientreport
(patientID varchar(10),
 symptoms varchar(200) not null,
 diagnosis varchar(200) not null,
 medicines varchar(200),
 doc_id integer,
 wardReq varchar(5),
 typeWard varchar(10),
 no_of_days integer,
 foreign key(patientID) references patient(patientID),
 foreign key(doc_id) references doctor(doc_id),
 foreign key(medicines) references medicine(med_name));

/*ward details*/
create table wards
(ward_id integer primary key auto_increment,
 patientID varchar(10),
 type_Ward varchar(20) not null,
 no_of_days integer,
 rate int,
 foreign key(patientID) references patient(patientID)
 on delete cascade);
```

7. DML

Data Manipulation Language (DML) refers to the INSERT, UPDATE and DELETE statements.

DML allows to add / modify / delete data itself.

- Insertion of new tuples into a given relation
- Deletion of tuples from a given relation.
- Updation of values in some tuples in a given relation

```
/* insert data into admin */
INSERT INTO `admin`(`username`, `password`) VALUES ("hms","admin");


/* username and password for doctor */
insert into doctor_login values
("doc1","doc1"),
("doc2","doc2"),
("doc3","doc3"),
("doc4","doc4"),
("doc5","doc5"),
("doc6","doc6"),
("doc7","doc7"),
("doc8","doc8"),
("doc9","doc9"),
("doc10","doc10");

SELECT * FROM `doctor_login` ;


insert into hospital values
('H100','Fortis Healthcare','9012345671','Noida','Delhi'),
('H101','Fortis Multispeciality Clinic','9012345129','Nagarbhavi','Karnataka'),
('H102','Fortis Medical Hospital','9012314555','Kalyan','Maharashtra');

SELECT * FROM `hospital`;


insert into doctor values
(1,"Akash","ortho","891234516","UP","asd@gmail.com"),
(2,"Abc","gynae","12345678","UP","qwerty@gmail.com");

SELECT * FROM doctor;


/* medicine names*/
insert into medicine values
("abc",800),
("xyz",900),
("pqr",1000);
```

8. DCL

Data Control Language(DCL) is used to control privileges in Database. To perform any operation in the database, such as for creating tables, sequences or views, a user needs privileges.

In DCL we have two commands,

- Grant : Used to provide any user access privileges or other priviliges for the database.
- Revoke : Used to take back permissions from any user.

```
/* DCL COMMANDS -- CREATE USER AND GRANT */

/* creation of user */
create user "admin"@"localhost" identified by "";

/* grant all privileges to user */
grant all on *.* to "admin"@"localhost";
```

9. Triggers

A MySQL trigger is a database object that is associated with a table. It will be activated when a defined action is executed for the table. The trigger can be executed when you run one of the following MySQL statements on the table **INSERT**, **UPDATE** and **DELETE** and it can be invoked before or after the event.

Triggers can work our work easier.

In this project we have taken a after insert trigger on Patientreport table which will automatically insert the data in the wards table after inserting into Patientreport table.

```
/* Trigger which automatically makes an entry in wards table after patient diagnosis */

delimiter $
create trigger after_insert_in_patientreport
after insert on patientreport
for each row
BEGIN
if new.wardReq="yes" THEN
if new.typeWard="General" THEN
insert into wards(patientID,type_Ward,no_of_days,rate)
values (new.patientID,new.typeWard,new.no_of_days,5000);
elseif new.typeWard="Single" THEN
insert into wards(patientID,type_Ward,no_of_days,rate)
values (new.patientID,new.typeWard,new.no_of_days,8000);
elseif new.typeWard="Duo" THEN
insert into wards(patientID,type_Ward,no_of_days,rate)
values (new.patientID,new.typeWard,new.no_of_days,10000);
end if;
end if;
end $
```

10. PL/SQL Procedure

PL/SQL subprograms are named PL/SQL blocks that can be invoked with a set of parameters. PL/SQL provides two kinds of subprograms −

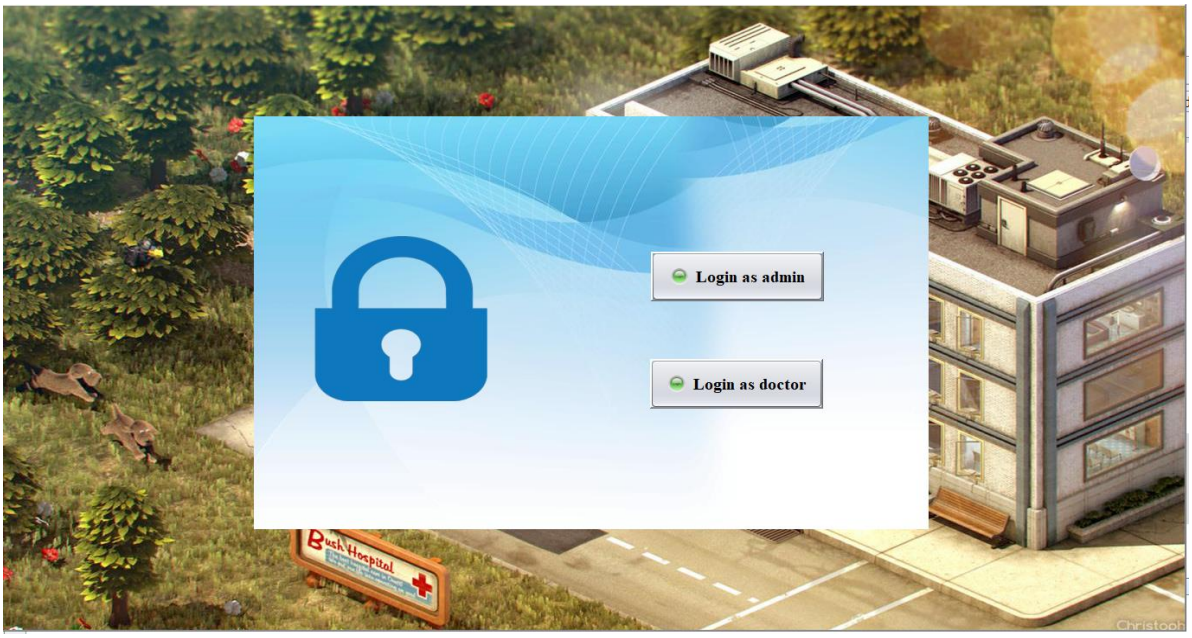
- **Functions** − These subprograms return a single value; mainly used to compute and return a value.
- **Procedures** − These subprograms do not return a value directly; mainly used to perform an action.

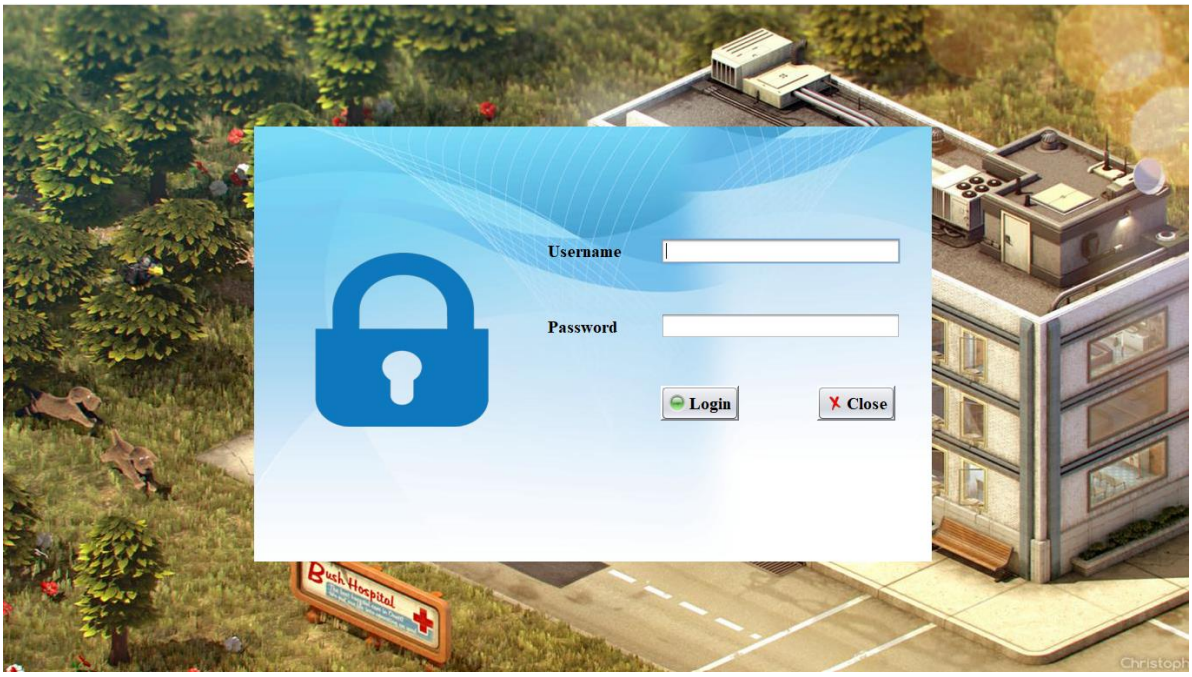
PL/SQL Procedures are used to calculate the bill of a patient in our project . It takes the in parameter patientID, which is used to output various details about the patient and calculate the bill of the patient based on the type of the ward of the patient, number of days stayed by him and medicines prescribed to him.

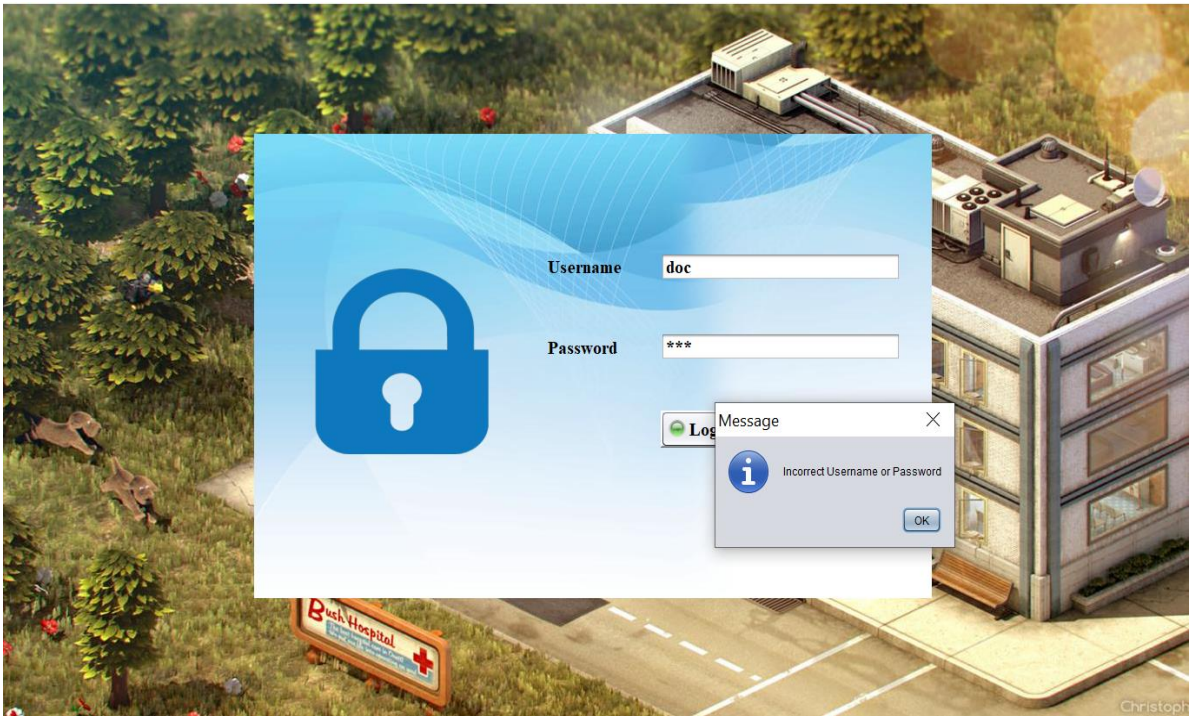
```
/* Procedure is used to calculate the bill of a patient */

delimiter $
CREATE PROCEDURE `bill_pay`(IN `pid` VARCHAR(10), OUT `nm` VARCHAR(100), OUT `ag` INT, OUT `gen` VARCHAR(10),
OUT `addr` VARCHAR(100), OUT `wt` VARCHAR(20), OUT `wc` INT, OUT `mc` INT, OUT `cost` INT)
BEGIN
select name into nm from patient where patientID=pid;
select age into ag from patient where patientID=pid;
select gender into gen from patient where patientID=pid;
select address into addr from patient where patientID=pid;
select type_Ward into wt from wards where patientID=pid;
select (rate*no_of_days) into wc from wards where patientID=pid;
select price into mc from medicine where med_name in (select medicines1 from patientreport1 where patientID=pid);
select coalesce((mc + wc),mc,wc,0) into cost;
END $
```

11. Frontend GUI Screenshot

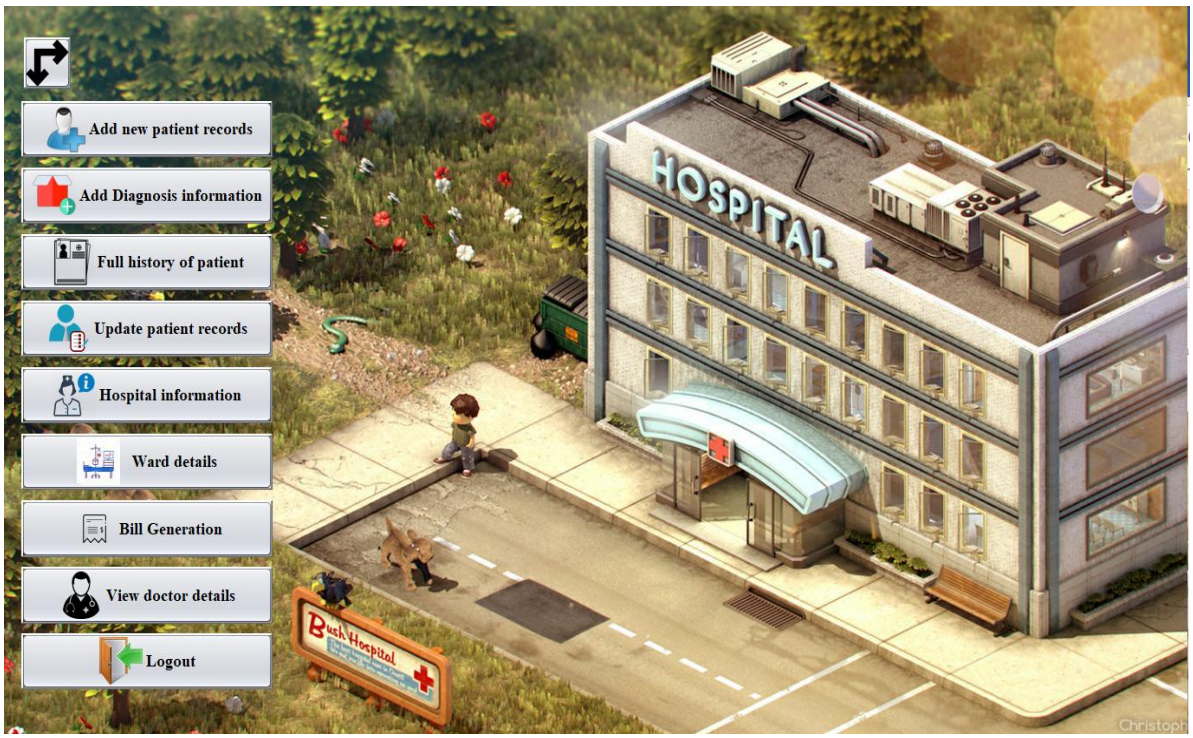
11.1 Login module



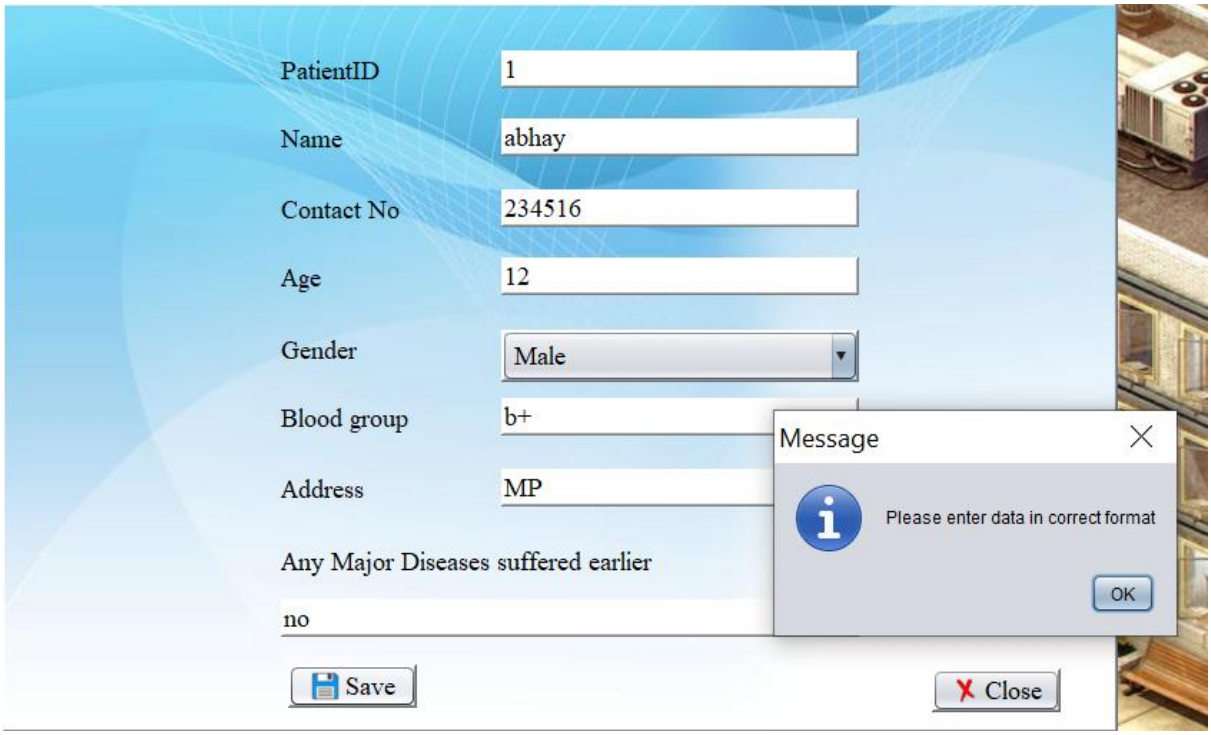


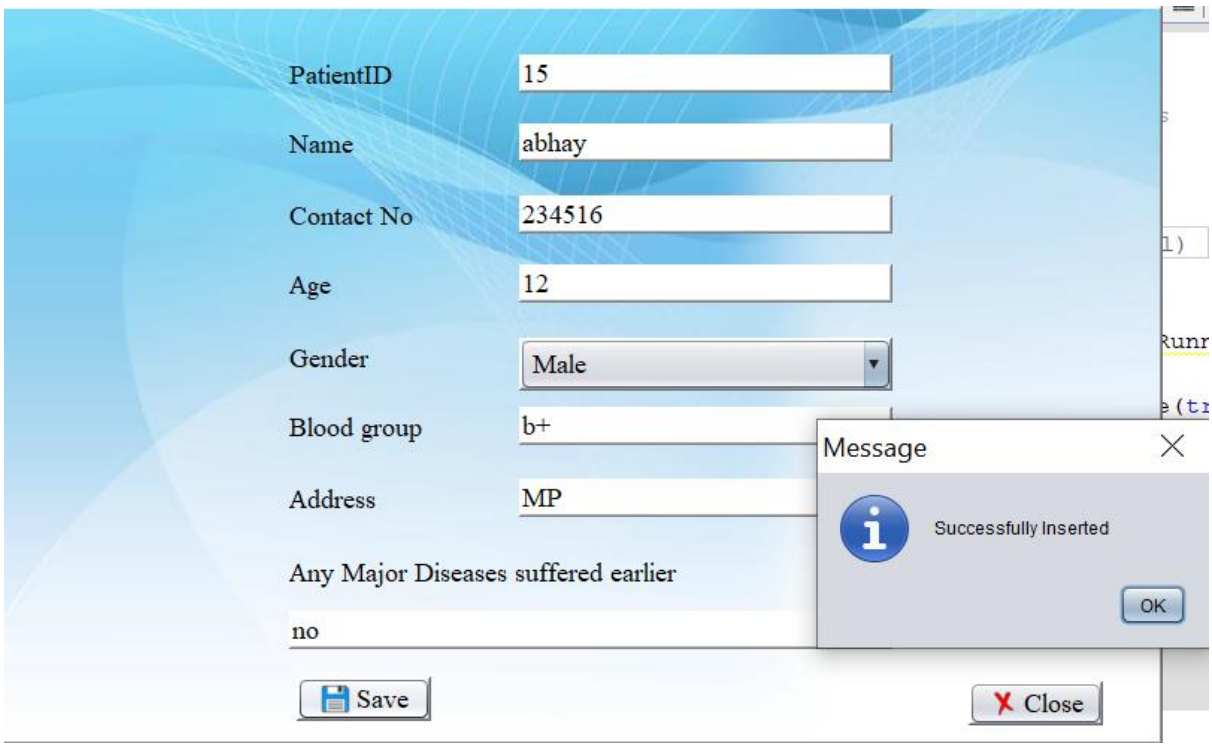


11.2 Home page

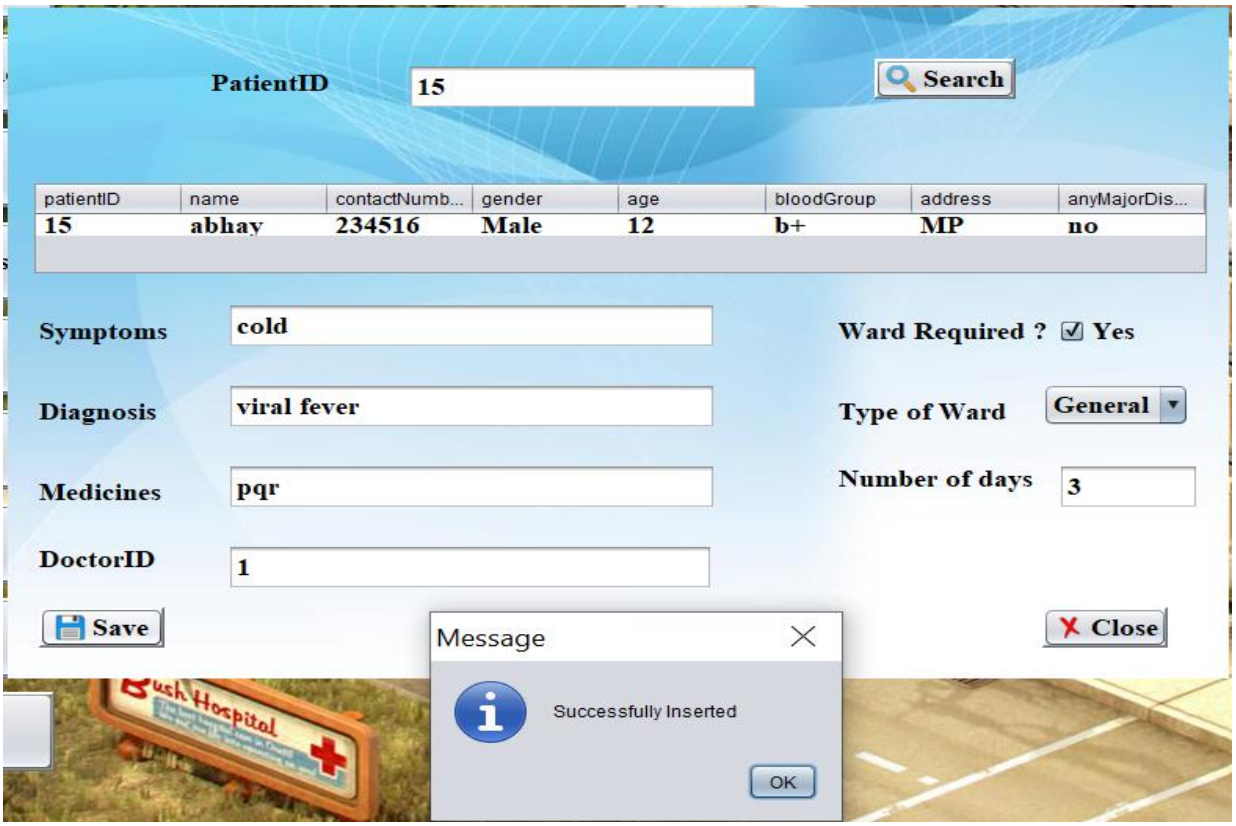


11.3 Add New Patient Record

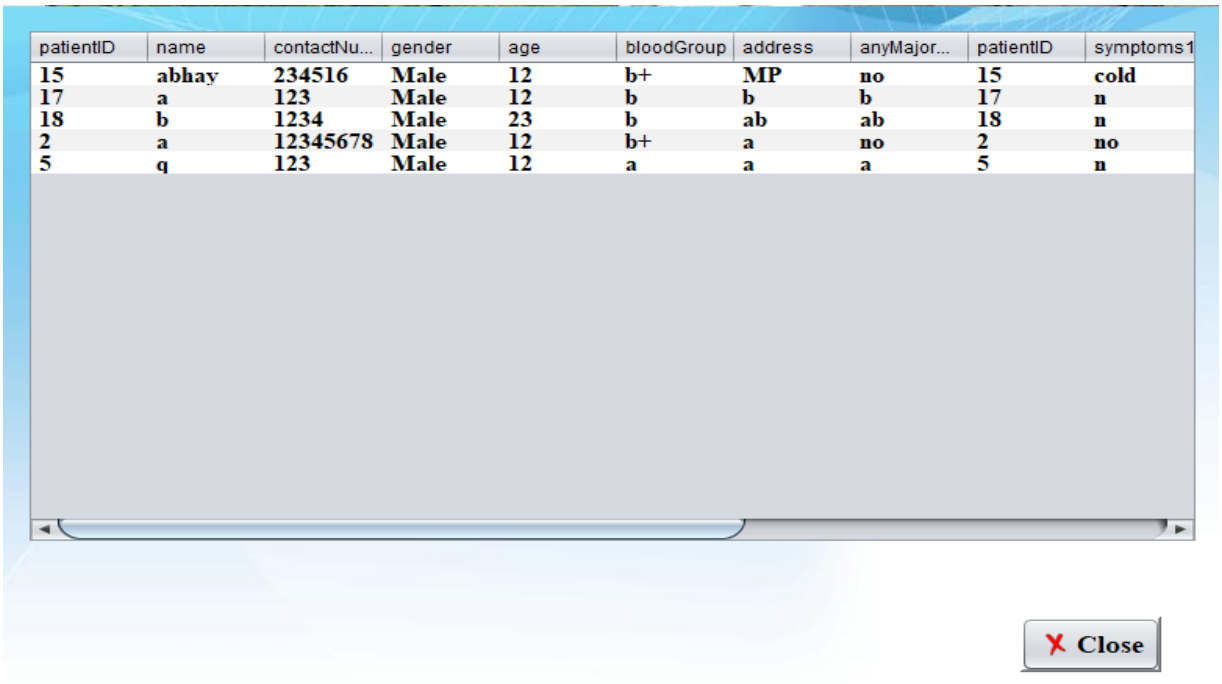




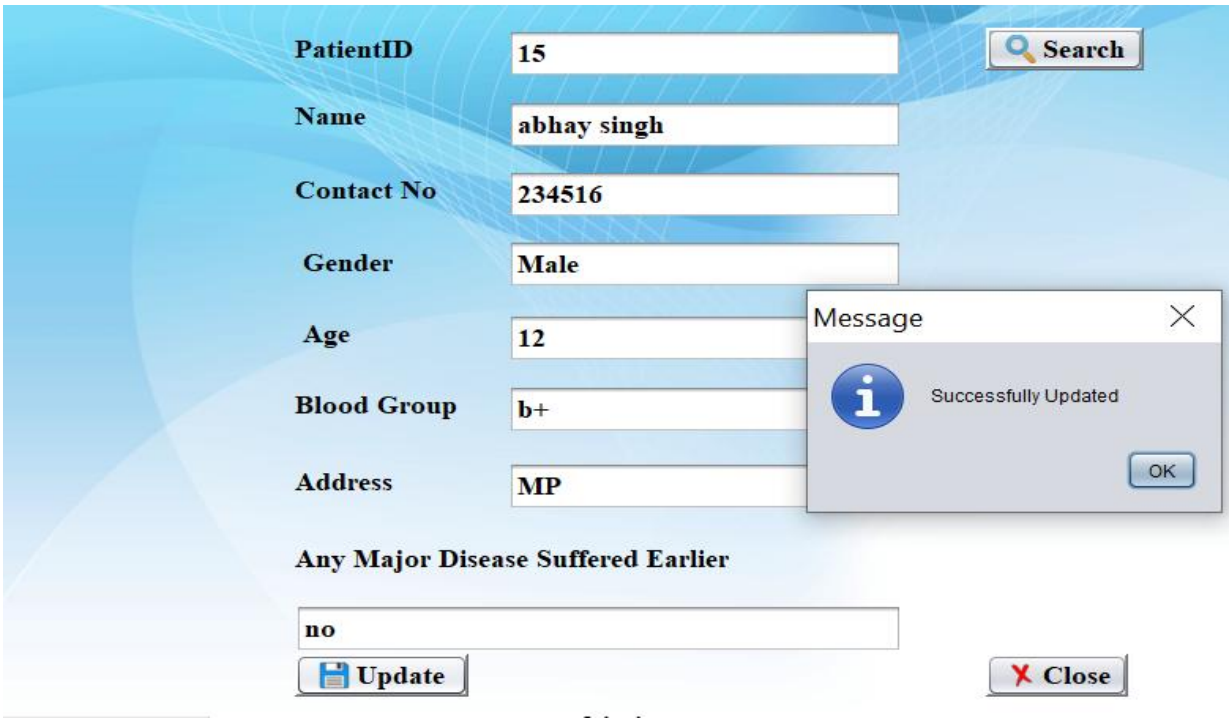
11.4 Add Diagnosis Information



11.5 Full history of patient

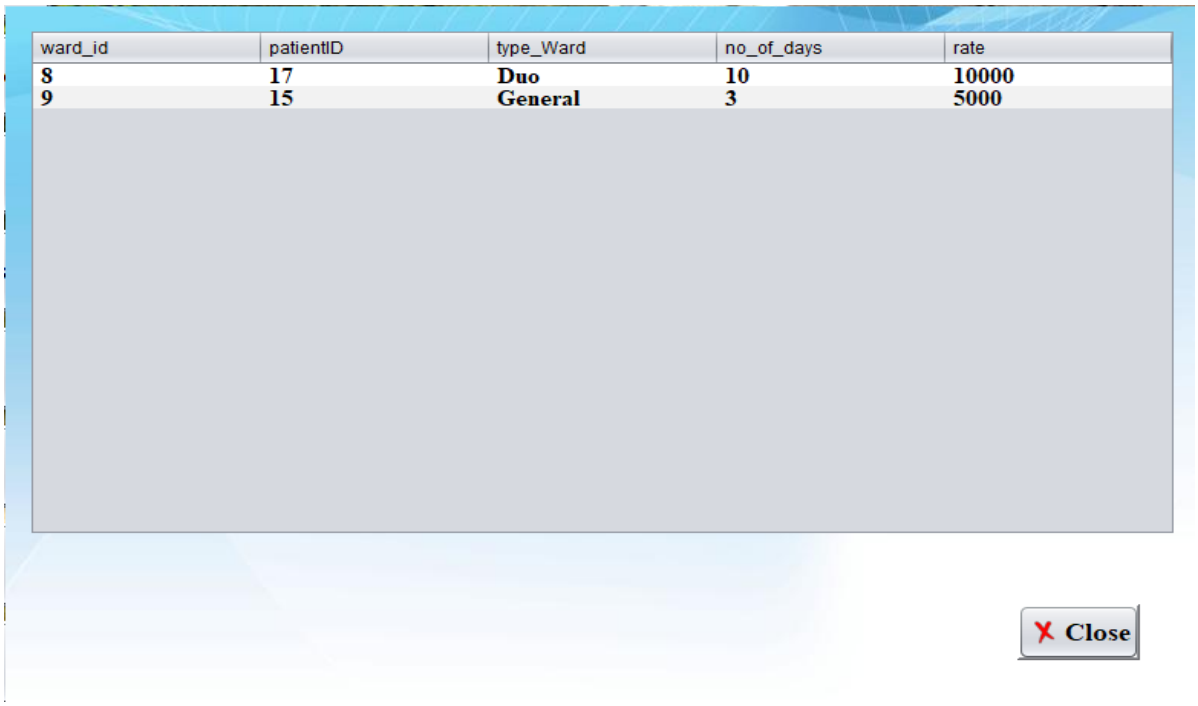


11.6 Update Patient Records



11.7 Hospital Information

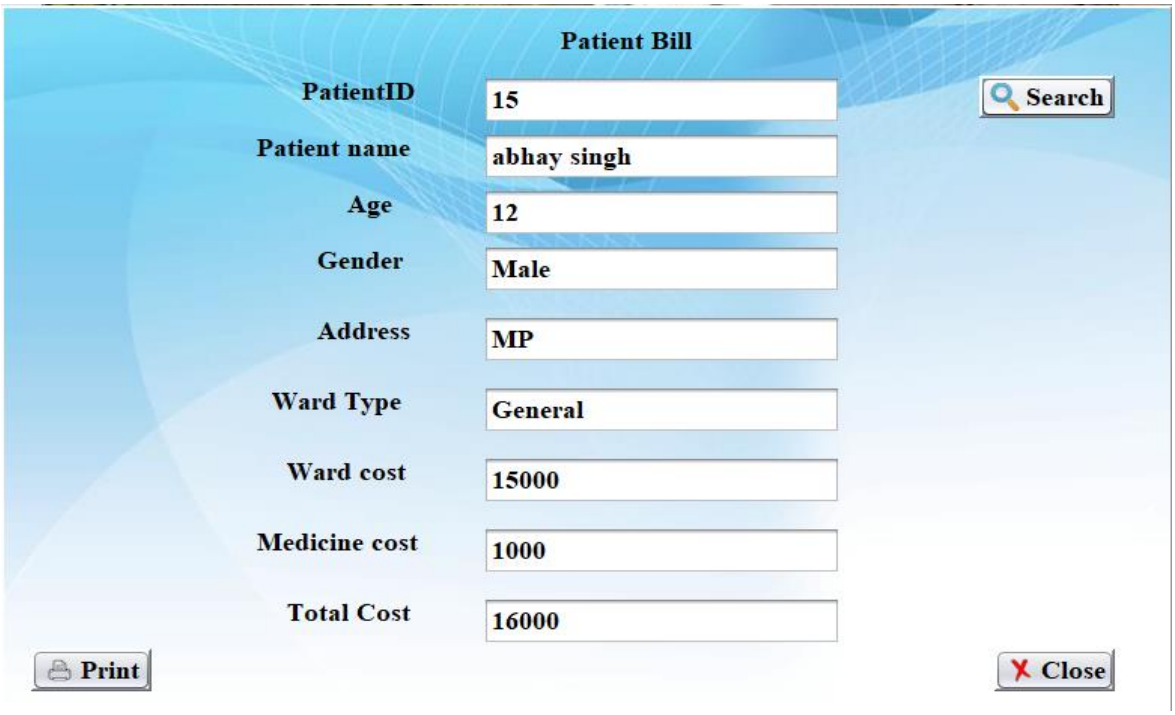
11.8 Ward Details



ward_id	patientID	type_Ward	no_of_days	rate
8	17	Duo	10	10000
9	15	General	3	5000

Close

11.9 Bill Generation



Patient Bill

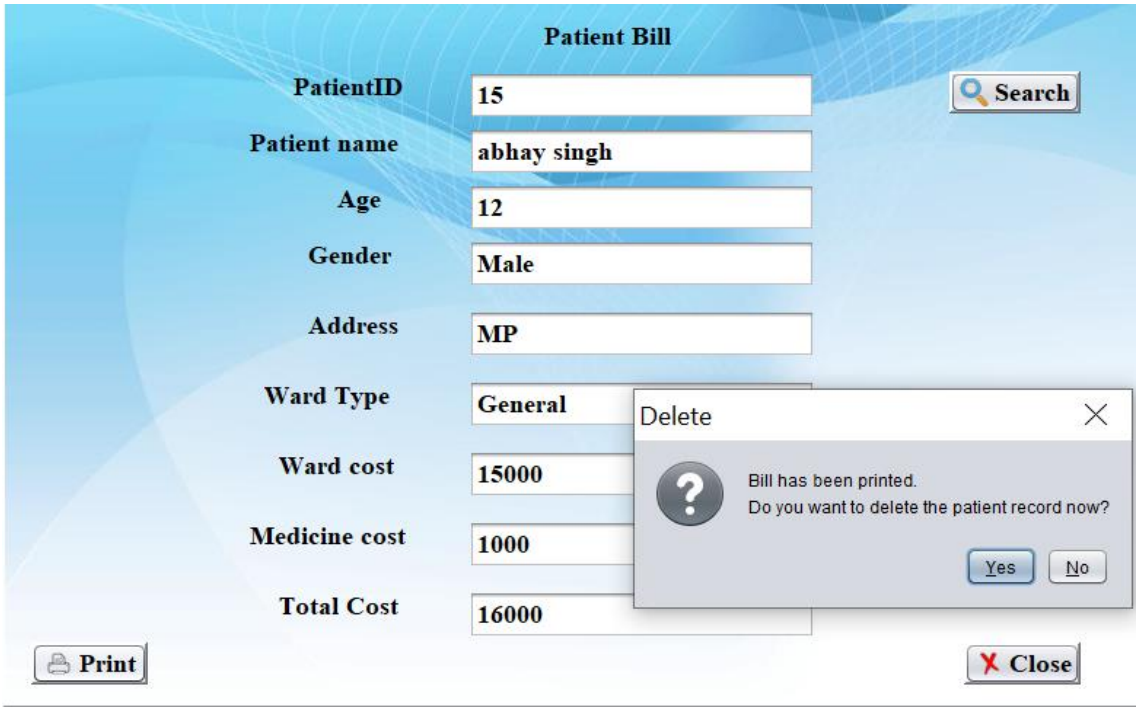
PatientID	15
Patient name	abhay singh
Age	12
Gender	Male
Address	MP
Ward Type	General
Ward cost	15000
Medicine cost	1000
Total Cost	16000

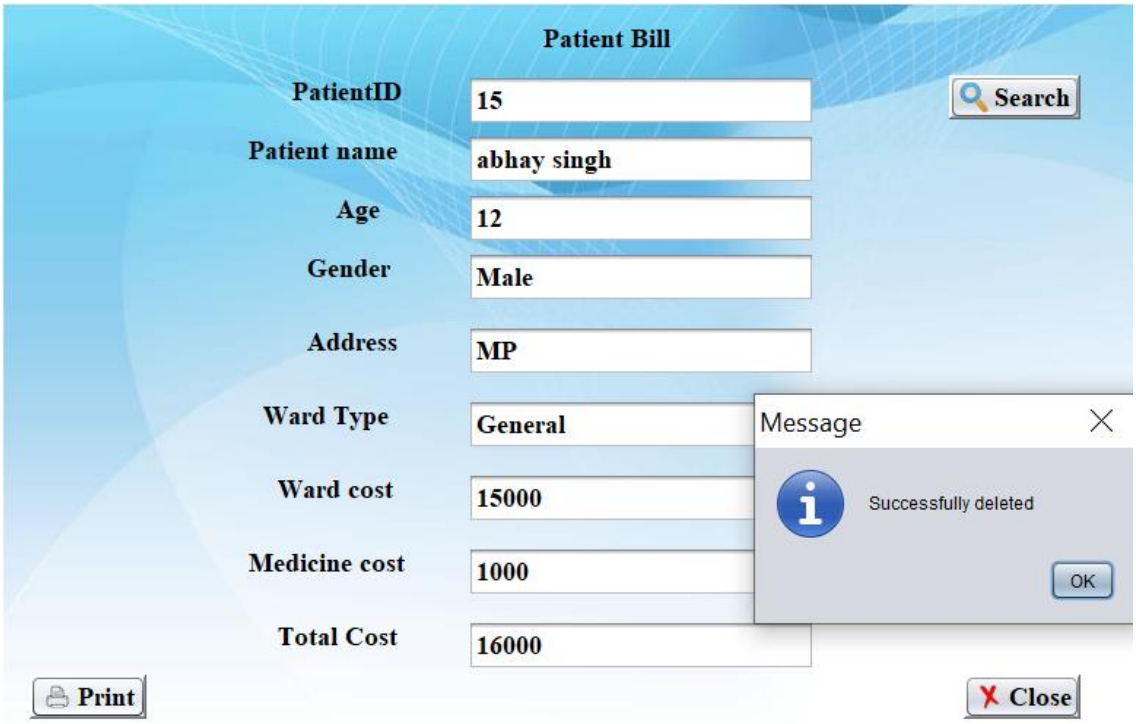
Search

Print

Close

11.10 Delete patient





12. Conclusion

The project Hospital Management System (HMS) is for computerizing the working in a hospital. It is a great improvement over the manual system. The computerization of the system has speed up the process. In the current system, the front office managing is very slow.

The hospital managing system was thoroughly checked and tested with dummy data and thus is found to be very reliable. The software takes care of all the requirements of an average hospital and is capable to provide easy and effective storage of information related to patients that come up to
the hospital.

It generates test reports and also provides the facility for searching the details of the patient. It also provides billing facility on the basis of patient's status whether it is an indoor or outdoor patient. The system also provides the facility of backup as per the requirement.

13. References

- Binstock. Andrew (May 20. 2015). "Java's 20 Years of Innovation" . Forbes. Archived from the original on March 14, 2016. Retrieved March 18. 2016.

- Herbert Scheldt. Java Complete Reference, Fifth Edition. Tata McGraw Hill Edition.

- Gosling. James: Joy. Bill: Stele. Guy: Bracha. Gilad. "The Java Language Specification, 2nd Edition". Archived from the original on August 5, 2011. Retrieved February 8. 2008.

- “MySQL 8.0 Release Notes". mysql.com. Retrieved 29 July 2019.

- https://en.wikipedia.org/wiki/NetBeans

- hitps://www. tutorial spoint.com/my sqlAndex.htm

- bitps//www. w3schools.com/java/java_intro.asp