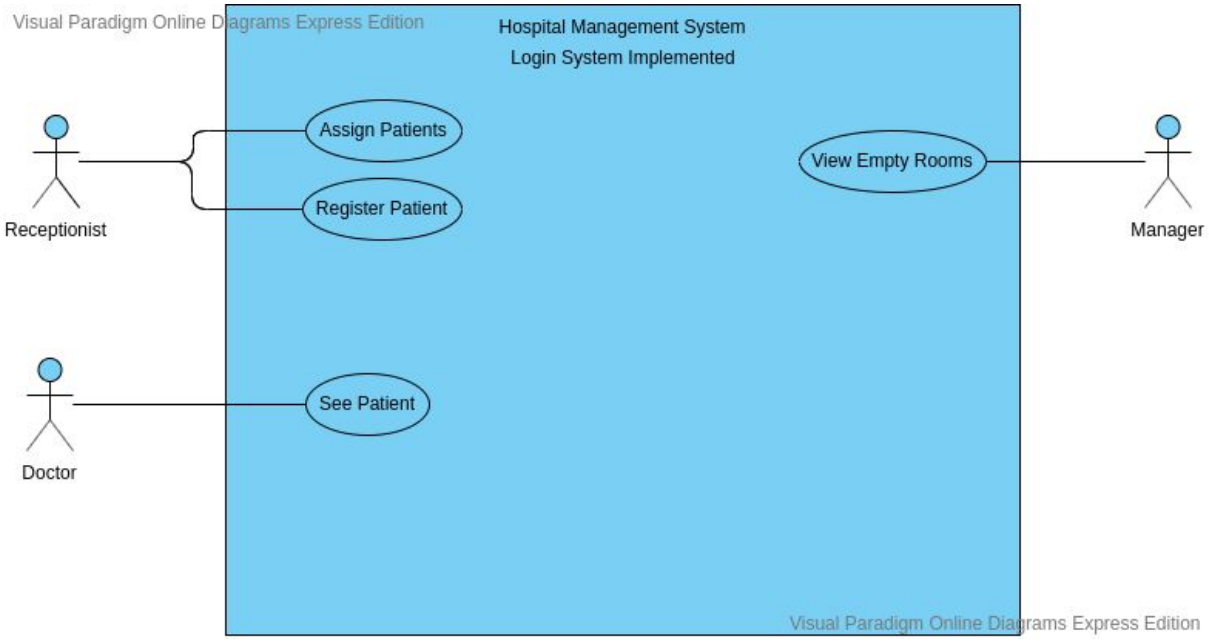
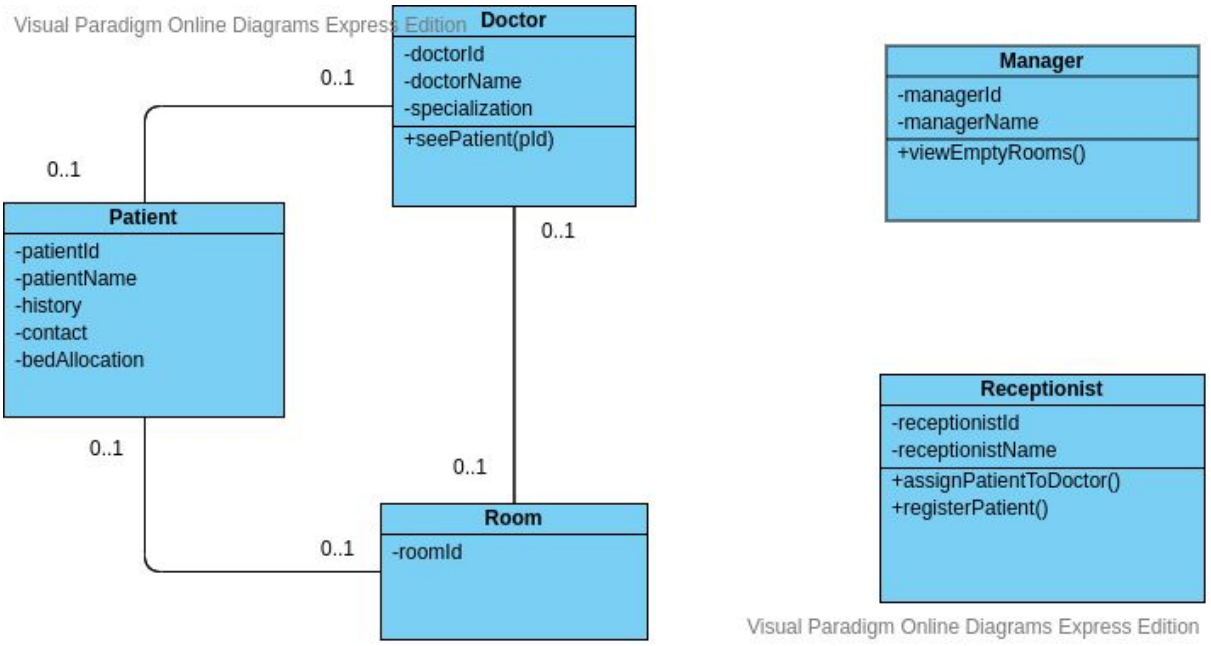
Design Document

- Ram S IMT2017521

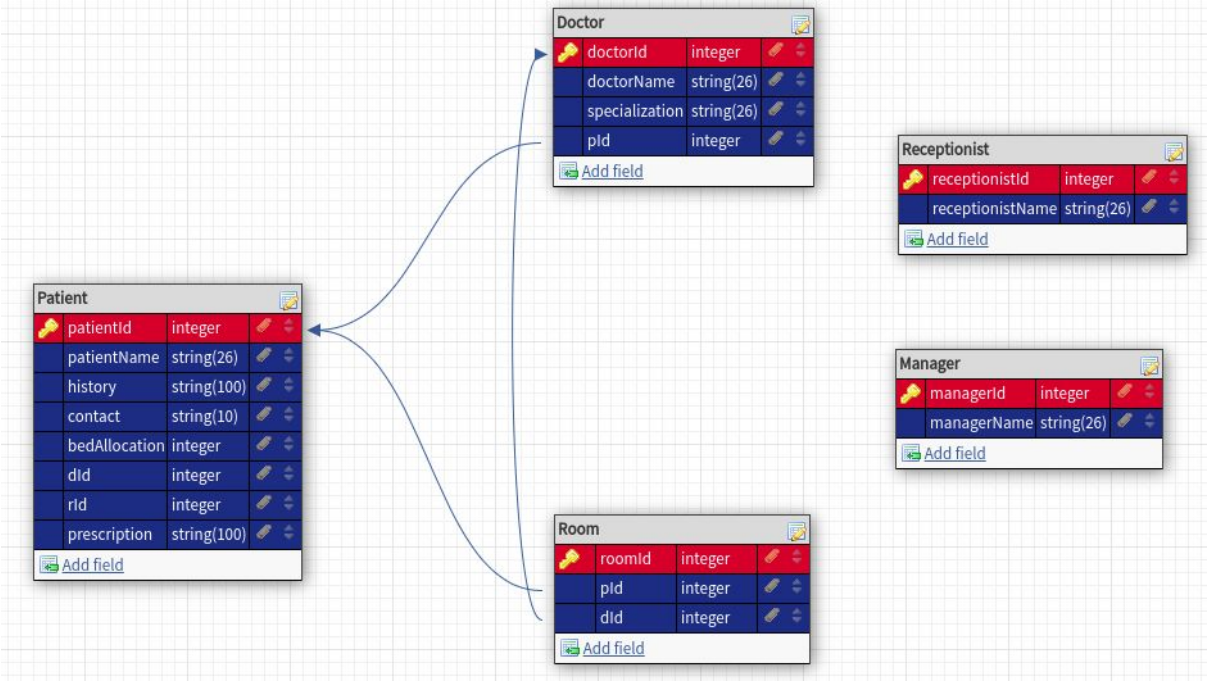
Use Case Diagram:



Class Diagram:



Database Design:



Output Screenshots:

Main Menu:

```
Welcome to the Hospital Management System
Loading class `com.mysql.jdbc.Driver'. This is deprecated.
Connecting to database...
Creating 2 Patients...
Please Enter your Login Details:
1 - Doctor
2 - Manager
3 - Receptionist
0 - Exit
```

Receptionist:

Assign Patient to Doctor:

```
Please Enter your Login Details:
1 - Doctor
2 - Manager
3 - Receptionist
0 - Exit
3
Enter Receptionist Id:
1
Enter Receptionist Name:
Receptionist1
Welcome Receptionist!
1 - Assign Patient to Doctor and allot room
2 - Register Patient
0 - Logout
1
Patient 1 has been assigned to Doctor4 at Room 91
```

Register Patient:

```
Welcome Receptionist!
1 - Assign Patient to Doctor and allot room
2 - Register Patient
0 - Logout
2
Enter Patient ID:
5
Enter Patient Name:
Patient5
Enter Patient History:
history5
Enter Patient Contact:
contact5
Enter Patient Bed Allocation:
1
Patient has been added to the Database as unassigned
Welcome Receptionist!
1 - Assign Patient to Doctor and allot room
2 - Register Patient
0 - Logout
```

Manager:

```
Please Enter your Login Details:
1 - Doctor
2 - Manager
3 - Receptionist
0 - Exit
2
Enter Manager Id:
1
Enter Manager Name:
Manager1
Welcome Manager!
1 - View Rooms Vacant
0 - Logout
1
List of Vacant Rooms are:
1
11
21
31
41
51
61
71
81
Welcome Manager!
1 - View Rooms Vacant
0 - Logout
```

Doctor:

```
Please Enter your Login Details:
1 - Doctor
2 - Manager
3 - Receptionist
0 - Exit
1
Enter Doctor Id:
4
Enter Doctor Name:
Doctor4
Welcome Doctor!
1 - Consult Patient
0 - Logout
1
Patient 1 has been consulted
Welcome Doctor!
1 - Consult Patient
0 - Logout
```

Note:

1) I have not used Java Beans (Classes) as I have implemented this program using JDBC programming. Since all my data processing was on the SQL side, I simply needed to generate the appropriate SQL statements for each function and execute them.
2) The Class Diagram I have provided is relevant only if I use Hibernate. I have included the code for my other classes as well, though not required, as I switched to JDBC halfway through the project.
3) I have provided a Relational Design (though not required) as a Relational Design helps make my JDBC application much easier to understand.

Instructions to run code:

1) If running for the first time, run ‘create.sql’ on MySQL to create the corresponding database.
2) Then run ‘ddl.sql’ to create all the tables required for the application.
3) In case you wish to delete all the tables to remake them later, run ‘drop.sql’
4) If you wish to only delete all the rows in all the tables, run ‘delete.sql’
5) After running ‘ddl.sql’, compile ‘App.java’ and ‘Dbhelper.java’ together. Now, run java with App (which has the main function) with the classpath containing JDBC jar files.
6) You should have an interface on the console terminal allowing for login to the application.