SYMBIOSIS INSTITUTE OF TECHNOLOGY (SIT)
Constituent of Symbiosis International (Deemed University), Pune
(Established under Section 3 of the UGC Act of 1956 vide notification number F-9-12/2001-U-3 of the Government of india)
Re-Accredited by NAAC with 'A' Grade
॥वसुधैव कुटुम्बकम्॥

Software Requirement Specifications (SRS)

ON

Hospital Management System (HMS)

BACHELOR OF SCIENCE

IN

INFORMATION TECHNOLOGY

(2020-2021)

SUBMITTED BY

1. YOGESHWAR PAWADE– 19070124050(SYIT-2).
2. ALNOOR LALANI – 19070124036(SYIT-2).
3. HARSH PATEL– 19070124047(SYIT-2).

UNDER THE GUIDANCE OF

PROF. RHYTHM BHATIA.

CONTENTS

1. Introduction

1.1 Purpose.

- ✓ The Software is for the automation of Hospital Management.
- ✓ It maintains two levels of users.
  - (1) Administrator Level.
  - (2) User Level.
    - Doctor.
    - Pharmacist.
    - Receptionist.
- ✓ The Software includes Maintaining Patient details.
- ✓ Providing Prescription, Precaution's advice.
- ✓ Providing, maintaining, and managing all kinds of database of the hospital.
- ✓ Billing and Report generation.

1.2 Scope.

- ✓ This software is created by keeping in mind the requirements of the user and the stakeholder and all the things that a hospital management requires.
- ✓ The proposed software product is the **Hospital Management System (HMS)**. The system will be used to get the information from the patients and storing that data for future usages.
- ✓
- ✓ The current system uses MySQL for creating a Database and using the current given Database provided to the software.
- ✓ The intentions of the system are to reduce over-time pay and increase the number of patients that can be treated accurately.
- ✓ Requirement statements in this document are both functional and non-functional.

1.3 Overview.

- ✓ This Software Requirement Specification (SRS) is the requirement work product that formally specifies Hospital Management System (HMS).
- ✓ The objective of this document therefore is to formally describe the system's requirements including functional and non-functional

requirement and business rules and constraints. The detailed structure of this document is organized as follows:

- **Section 2** of this document provides an overview of HMS and include a general description of the product, user characteristics, general constraints, and any assumptions for this system.
- **Section 3** presents the detailed requirements of the system.
- **Section 7** presents the non-functional requirements of the system.

2. General Description

2.1 Product Perspective.

✓ This Hospital Management System is a self-contained system that manages activities of the hospital as patient Information, Pharmacist information and other facilities. Various stakeholders are involved in the hospital patient info system.

2.2 Product Features.

✓ The features of the system are as follows:

(1) Receptionist:

- Can create patient, update, delete with the information provided by the patient.
- Can create channel between doctor and patient.
- Can create login id for any user such as doctor, pharmacist.

(2) Doctor:

- Can update, insert, and view his/her information in the system.
- Can view channels created by receptionist to view his appointments.

(3) Pharmacist:

- Can view prescription created by doctor, create inventory and generate bill of the patient based on the prescription.

- Can create an item to add things to stock of medical.
- Can generate the report of whole sales done by him.

✓ Following users can login to system and can perform their function and logout easily but the registration can be just done by the receptionist.

✓ System also saves all the Database created and are inter linked with each other so that everyone can coordinate without moving from their chairs.

2.3 Design and Implementation Constraints.

✓ **Database:**
The system shall use the MySQL Database, which is a open source and free.

✓ **Operating System:**
The Development environment shall be Windows.

✓ **Web-Based:**
The system shall be a Web-based application.

2.4 Assumption and dependencies.

✓ It is assumed that the Hospital have enough trained staff to take care of the system and can maintain the system very well.

✓ It is assumed that the receptionist of the Hospital is well qualified to use the system properly as all the rights are given to receptionist as whole Database is under control of receptionist.

3. Function Requirement

3.1 Description.

✓ **Registration:**

- Login id and password is defined for receptionist. For logging in as doctor and pharmacist a login id and password has to be created by Admin by registering for a doctor, pharmacist and receptionist.

- Registration of a patient is also done by receptionist.
- Receptionist can create a appointment of patient which will be displayed on doctors desk.

✓ Functions according to users:

(1) **Receptionist:**

- Create Patient → update (Name, phone number, address), delete, clear, save (all this information is saved in a Database).
- Create Appointment → doctor name, patient name, room no., date.
- View Doctor → Can view available doctors and their info.
- Logout.

(2) **Doctor:**

- Doctor → update, delete, clear, save. Details that are asked are: Name, specialization, qualification, room no.
- View Appointment → can view channels created by receptionist which lets him know about his appointments. After selecting a channel doctor can prescribe the patient by clicking prescription button and can save the prescription.
- View Doctors → Can view available doctors and their info.
- Logout.

(3) **Pharmacist:**

- View Prescription → Can view prescription created and saved by doctor. It also has a inventory button to generate the bill of the patient.
- Create Item → Can create stock for medical.
- View Doctor → Can view available doctors.
- SalesReport → Will get whole report of sales including the money paid and the balance.
- Logout.

(4) **Admin:**

- Create User → Can create login credentials for new user of system (doctor, pharmacist, receptionist).
- View Doctors → Can view available doctors and their info.
- Logout.

3.2 Technical issues.

- ✓ **Database:**
  The system shall use the MySQL Database, which is a open source and free. Without which entries in Database will not work.
- ✓ **Operating System:**
  The Development environment shall be Windows.
- ✓ **Web-Based:**
  The system shall be a Web-based application.

3.3 Front End (Includes Designing UI).

- ✓ Login.
- ✓ Create Appointment.
- ✓ Create Item.
- ✓ Doctor.
- ✓ Patient.
- ✓ Inventory.
- ✓ Main Frame.
- ✓ Prescription.
- ✓ Sales Report.
- ✓ User Creation.
- ✓ View Appointment.
- ✓ View Doctor.
- ✓ View Prescription.

3.4 Back End (Includes Internal Connections).

- ✓ Back End includes all the connections made between different JFrames, JButtons and Databases which is done using Java Swing
- ✓ Adding, deleting, updating Database using frontend and displaying database live in JTables.

4. Interface Requirements

4.1 User Interface.

- ✓ The software provides good graphical interface for the user any administrator can operate on the system, performing the required task such as create, update, viewing the details of the book.
- ✓ Allows user to view quick report like patient details/prescriptions/bills etc in between particular time.
- ✓ Stock verification and search facilities based on different criteria.

4.2 Hardware Interface.

- ✓ Operating system: Windows.
- ✓ Hard Disk: 50GB.
- ✓ RAM: 512MB.
- ✓ Processor: i3 6^{th} generation x32-bit and above.

4.3 Software Interface.

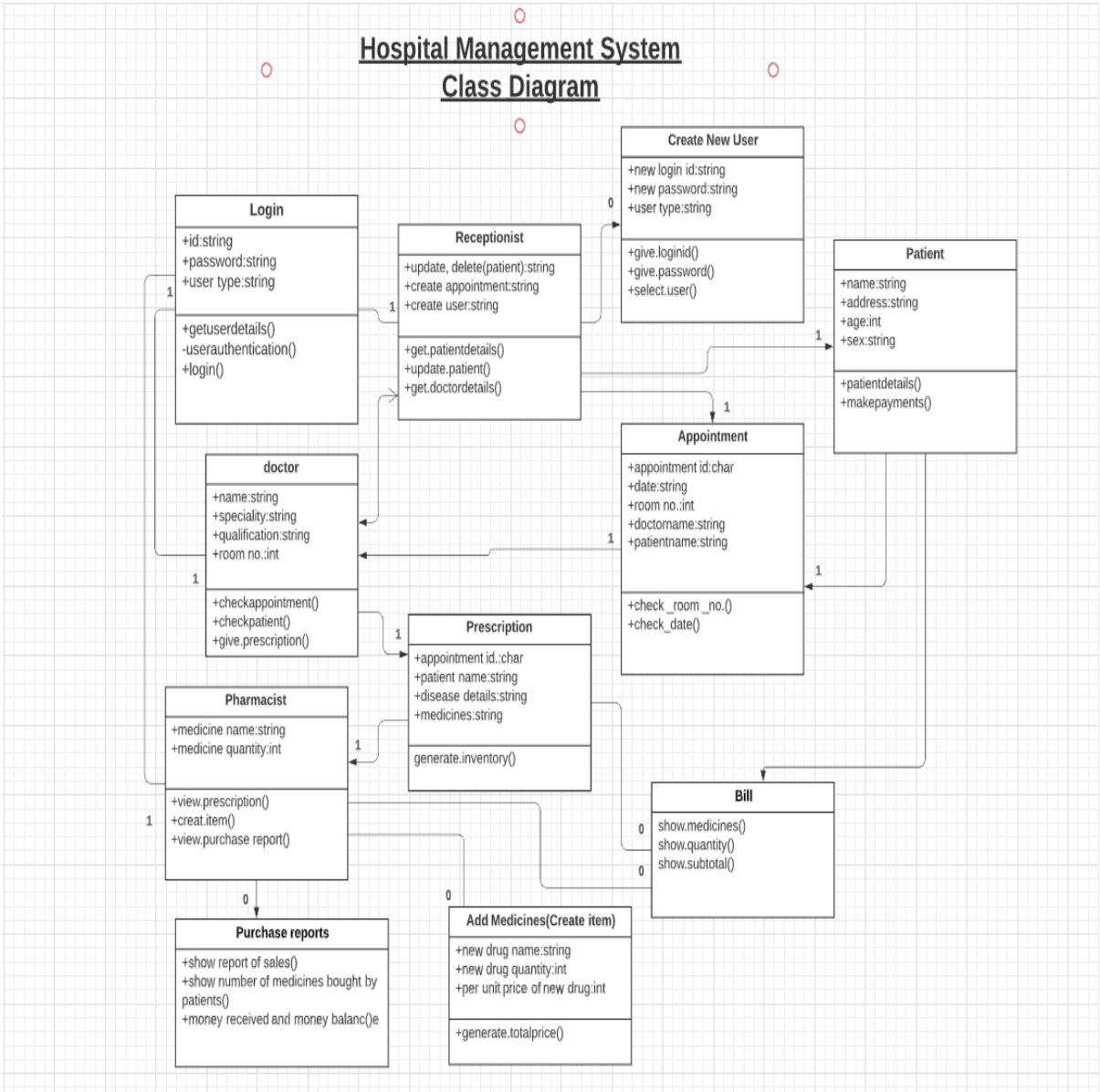
- ✓ Java language.
- ✓ Net beans IDE 7.0.1.
- ✓ MySQL server 2005 and above.

4.4 Communication Interface.

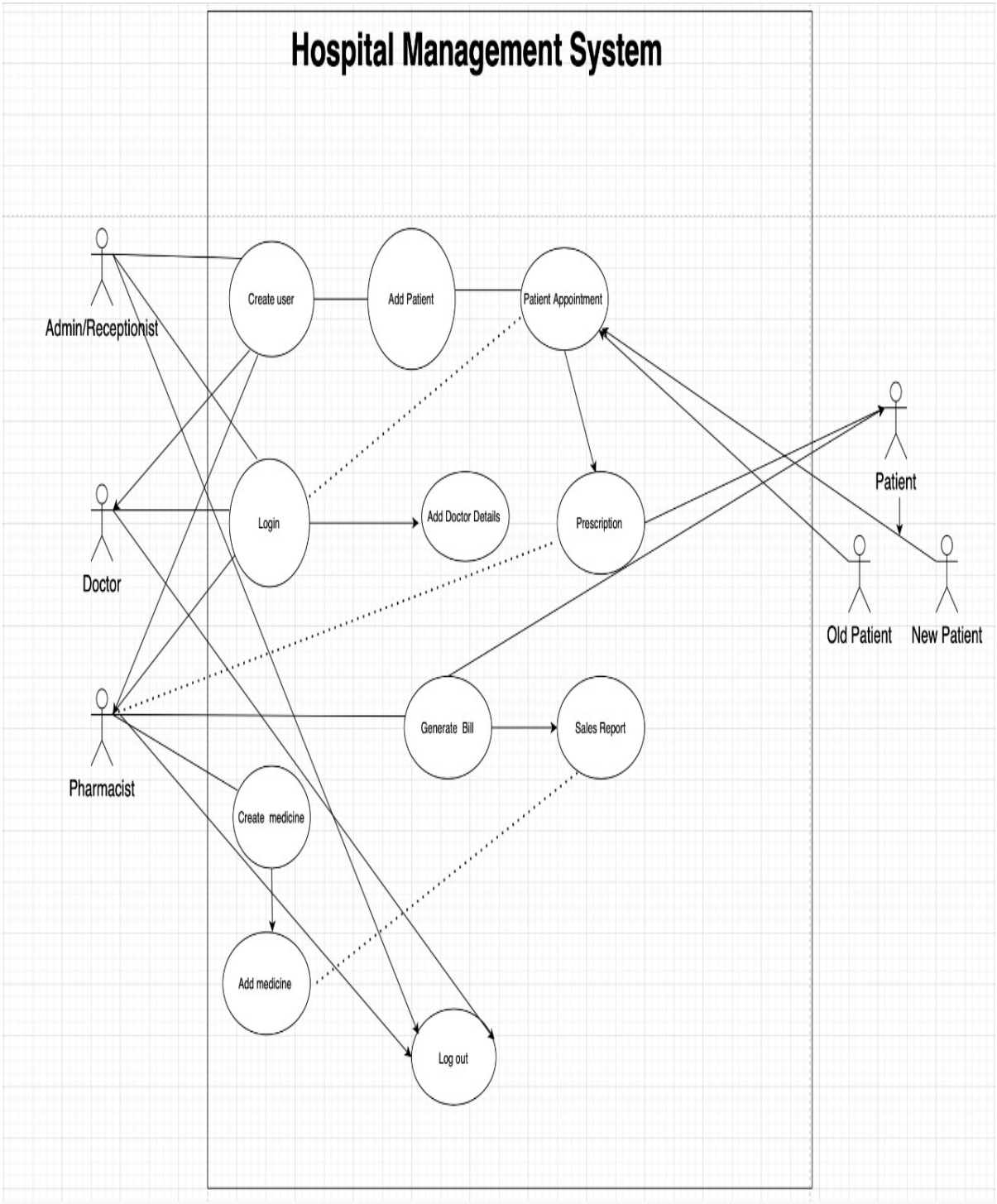
- ✓ Windows.

5. Software Design

5.1 Class Diagram.



5.2 Use Case Diagram.



6. Non function Requirement

6.1 Performance.

- ✓ **Response Time**:
  The system shall give response in 1 second after checking the patient's information.
- ✓ **Capacity**:
  The system must support 100 people at a time.
- ✓ **User Interface**:
  The user-interface shall respond within 5 seconds.
- ✓ **Conformity**:
  The system conforms to the Microsoft Accessibility.

6.2 Security.

- ✓ **Patient Identification:** The system requires the patient to identify himself/herself using patient ID and PHN.
- ✓ **Logon ID:** Any user who uses the system shall have a Logon ID and Password.
- ✓ **Modification:** Any modification (insert, delete, update) for the Database shall be synchronized and only by the administrator in the ward.
- ✓ **Front Desk Staff Rights:** Front Desk staff (Receptionist) shall be able to view all information, add new patients but shall not be able to modify any information in it.
- ✓ **Administrator's Rights:** Administrator shall be able to view and modify all information in Database.

6.3 Reliability.

- ✓ How general the form generation language is Simplicity vs. functionality of the form language = speed up form development but does not limit functional.

6.4 Availability.

- ✓ The system shall be available all the time.

6.5 Safety.

- ✓ Humans are error-prone, but the negative effects of common errors should be limited. E.g., Users should realize that a given command will delete data and be asked to confirm their intent or have the option to undo.

6.6 Software Quality.

- ✓ Good quality of the framework = produces robust, bug free software which contains all necessary requirements Customer satisfaction.

6.7 Reusability.

- ✓ Is part of the code going to be used elsewhere = produces simple and independent code modules that can be reused.

6.8 Maintainability.

- ✓ **Back Up:**
  The system shall provide the capability to back-up the Data.
- ✓ **Errors:**
  The system shall keep log of all the errors and Data.

7. Future Scope

- ✓ The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity. So, we can increase the capacity of our Database.
- ✓ In this system we can update the pharmacy system and make it transparent to patient so that they can get easy access
- ✓ We can also take out some rights from receptionist and create a new user as administrator and give those rights to administrator. Rights such as: creating user id and password for a new user is done by administrator in an original hospital system but here we have given those rights to receptionist.

8. Test Cases

1. Login Page.

Title: Login page – Authenticate login successfully on HMS

Description: A registered user should be able to successfully login.

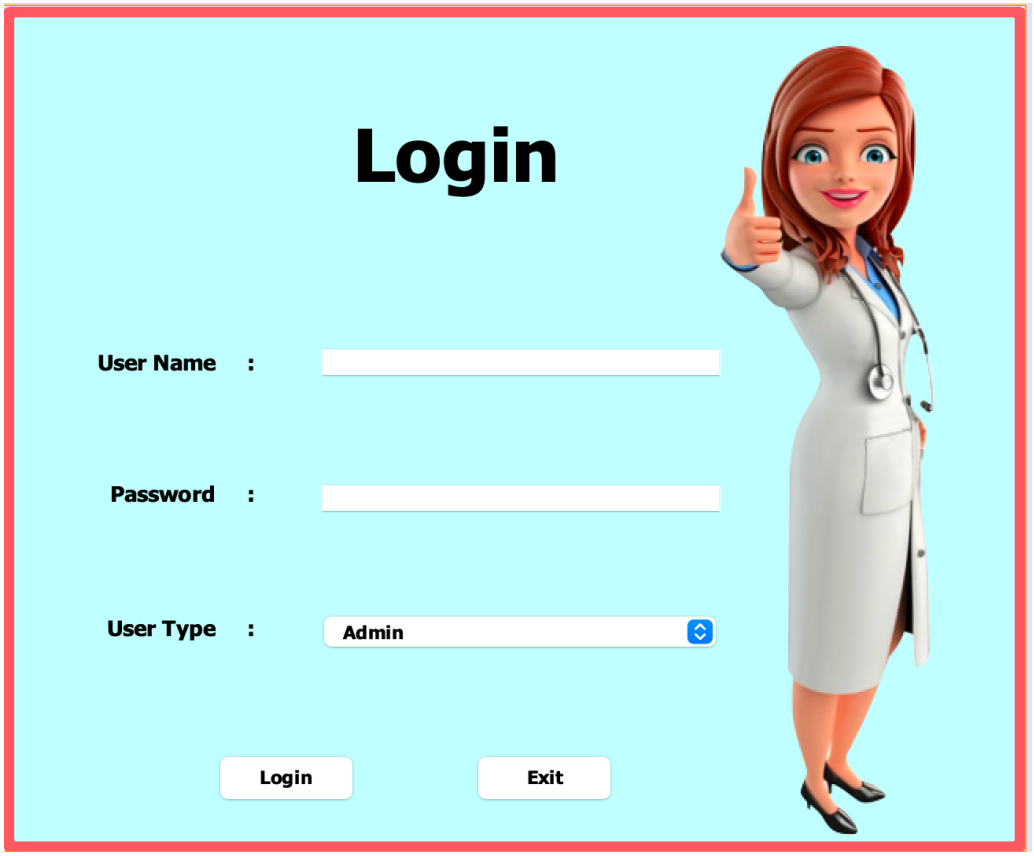
Precondition: The user must already be registered with an login ID and password by admin.

Assumption: Supported UI and database is being used.

Test Steps:

i. In the 'user name' field, enter the registered user name.
ii. In the 'password' field, enter the registered password.
iii. In the 'user type – admin, receptionist, doctor, pharmacist' field, enter the registered user type.
iv. CLICK 'Login'.
v. CLICK 'Exit'.

Login JFrame:



2. Main Frame.

Title: Admin Main Frame

Description: To display the functions available for the admin.

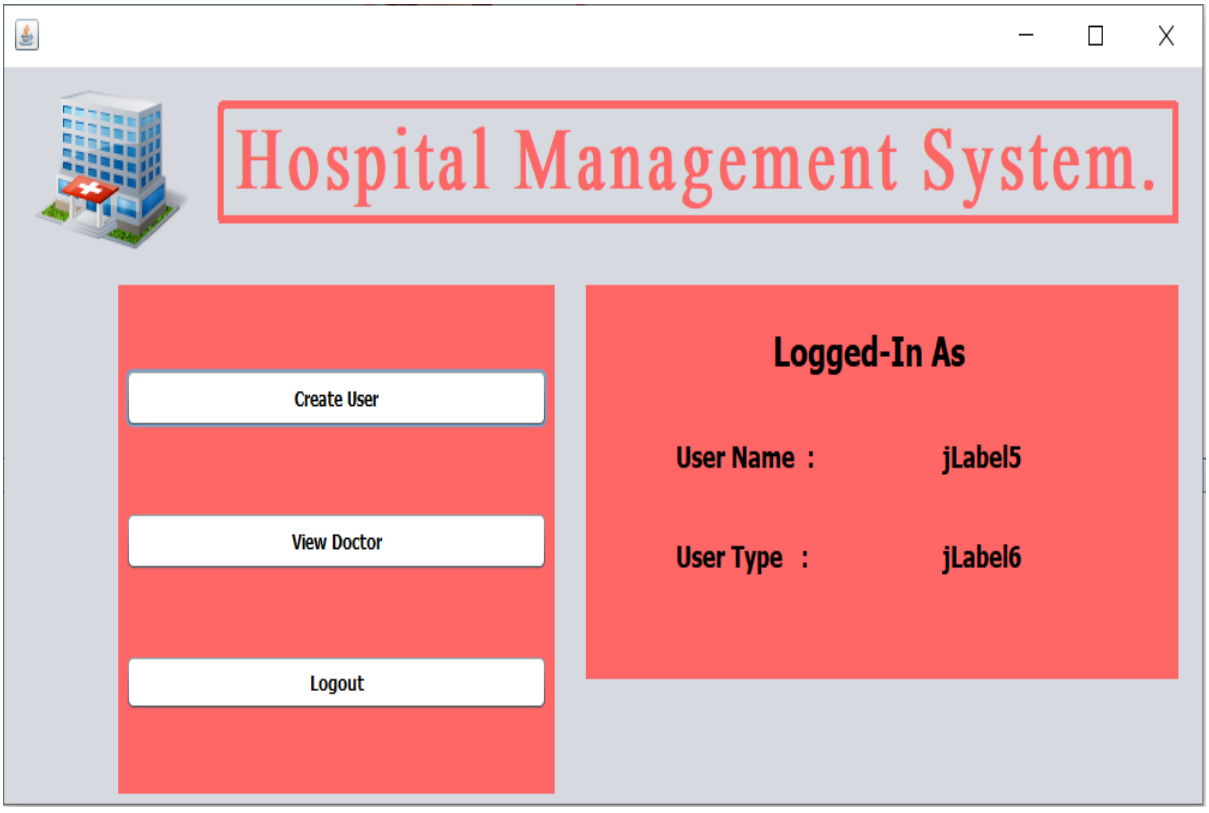
Precondition: Admin should login with the correct user ID, password and user type to get access to the admin functions.

Assumption: Supported UI and database is being used.

Test Steps:

i. In the 'Create user' function admin can create a new user for the hospital.
ii. In the 'View doctor' function admin can view all the available doctors with their information.
iii. CLICK 'Log Out'. User will be logged out from the system.
iv. In the Logged-in as JLabel user can see his/her's user name and user type.

Main Frame JFrame:



3. Main Frame 1.

Title: Receptionist Main Frame

Description: To display the functions available for the receptionist.

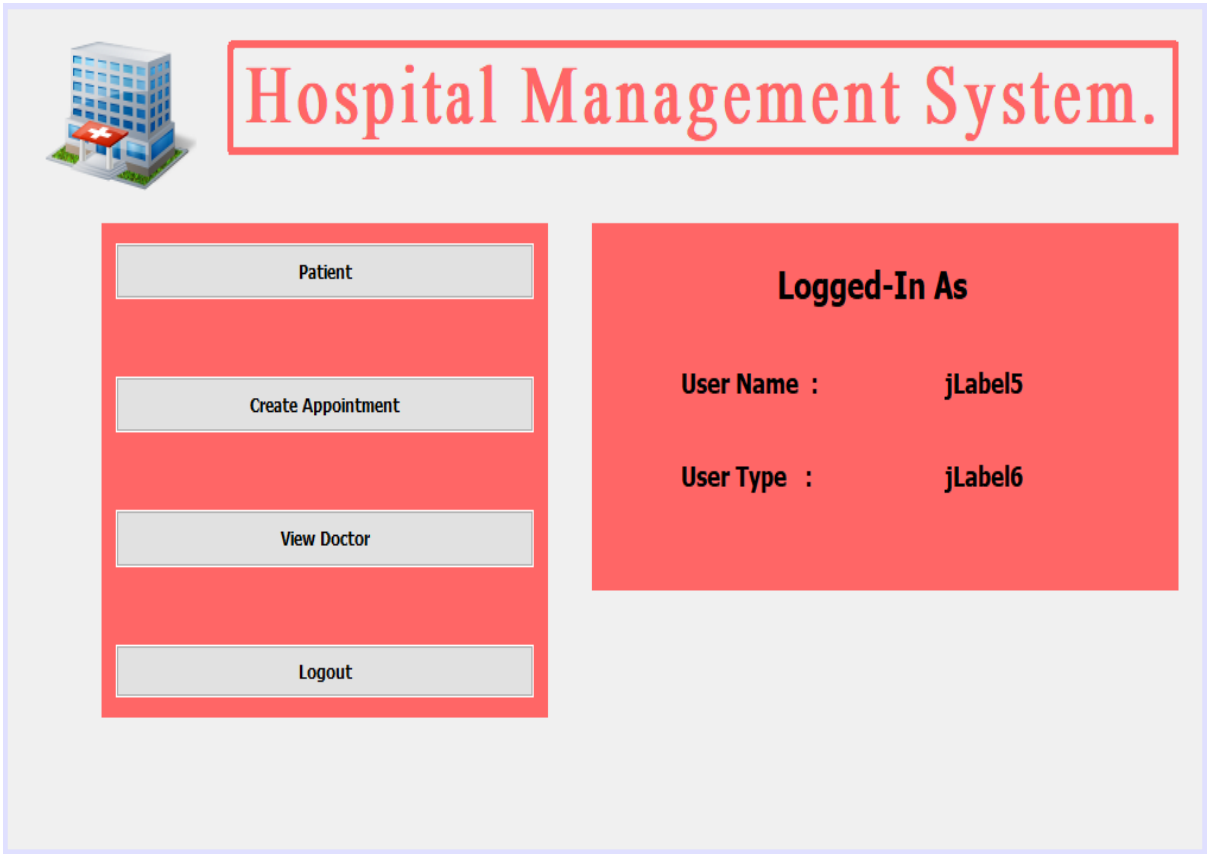
Precondition: Receptionist should login with the correct user ID, password and user type to get access to the receptionist functions.

Assumption: Supported UI and database is being used.

Test Steps:

i. In the 'Patient' function receptionist can register a patient.
ii. In the 'Create appointment' function receptionist can create an appointment.
iii. In the 'View doctor' function receptionist can view all the available doctors with their information.
iv. CLICK 'Log Out'. User will be logged out from the system.
v. In the Logged-in as JLabel user can see his/her's user name and user type.

Main Frame1 JFrame:



4. Main Frame 2.

Title: Doctor Main Frame

Description: To display the functions available for the doctor.

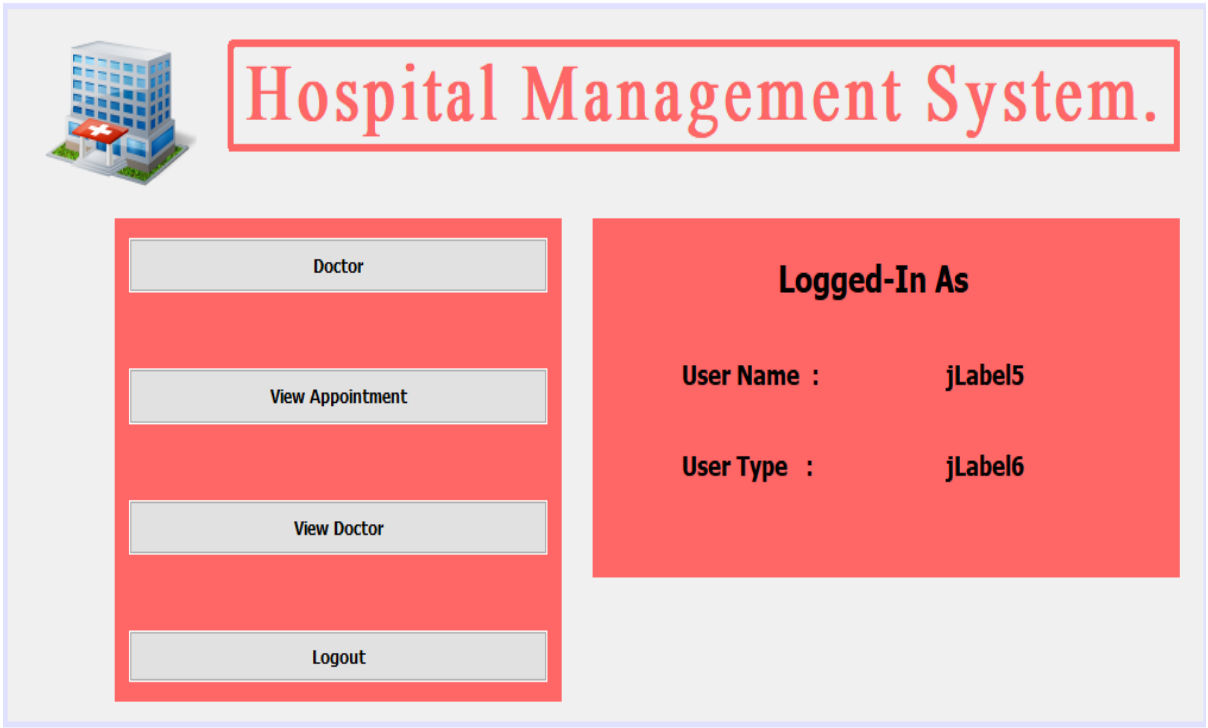
Precondition: Doctor should login with the correct user ID, password and user type to get access to the doctor functions.

Assumption: Supported UI and database is being used.

Test Steps:

i. In the 'Doctor' function, doctor can update/add his/her information in the database.
ii. In the 'View appointment' function, doctor can view the appointment which was created by the receptionist.
iii. In the 'View doctor' function receptionist can view all the available doctors with their information.
iv. CLICK 'Log Out'. User will be logged out from the system.
v. In the Logged-in as JLabel user can see his/her's user name and user type.

Main Frame2 JFrame:



5. Main Frame 3.

Title: Pharmacist Main Frame

Description: To display the functions available for the pharmacist.

Precondition: Pharmacist should login with the correct user ID, password and user type to get access to the pharmacist functions.

Assumption: Supported UI and database is being used.

Test Steps:

i. In the 'View prescription' function, pharmacist can view the prescription which was created by the doctor.
ii. In the 'Create Item' function, pharmacist can add the item which came into the pharmacy.
iii. In the 'View doctor' function receptionist can view all the available doctors with their information.
iv. CLICK 'Log Out'. User will be logged out from the system.
v. In the Logged-in as JLabel user can see his/her's user name and user type.

Main Frame3 JFrame:

6. Create User.

Title: User Creation

Description: Admin can create new user for the hospital, create the user name, password and the user type of that new user which can be used for the log-in.

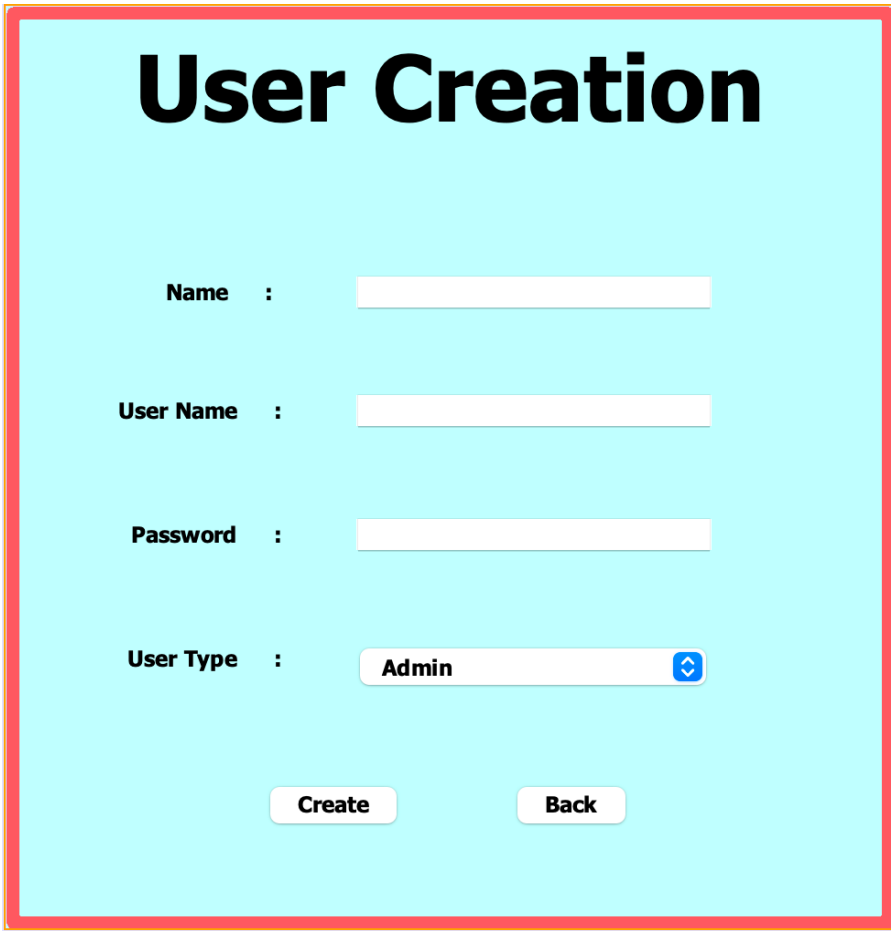
Precondition: User should provide his/her name correctly.

Assumption: Supported UI and database is being used.

Test Steps:

i. In the ‘name’ field, enter the correct name of the new user.
ii. In the ‘user name’ field, enter the user name of the new user.
iii. In the ‘password’ field, enter the password of the new user.
iv. In the ‘user type – admin, receptionist, doctor, pharmacist’ field, enter the new user type.
v. CLICK ‘Create’.
vi. CLICK ‘Back’.

User Creation JFrame:



7. Patient.

Title: Patient Registration

Description: Receptionist can register the new patient.

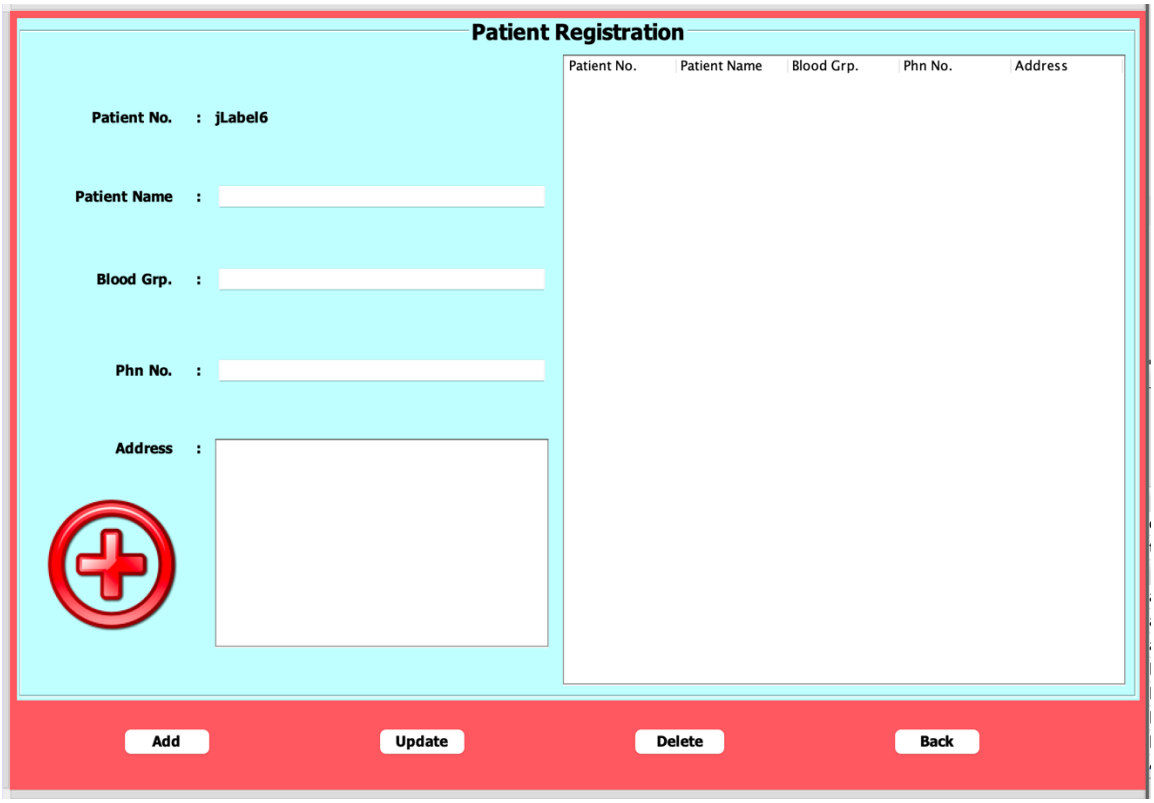
Precondition: Patient should give the correct details to the receptionist.

Assumption: Supported UI and database is being used.

Test Steps:

i. In the ‘Patient no.’ field patient number will be automatically generated as per the sequence.
ii. In the ‘Patient name’ field patient’s name has to be entered.
iii. In the ‘Blood Grp’ field patient’s blood group has to be entered.
iv. In the ‘Phn No.’ field patient’s phone number has to be entered.
v. In the ‘Address’ field patient’s address has to be entered.
vi. CLICK ‘Add’. To Add this entered details in the database.
vii. CLICK ‘Update’. To update any of the previously entered details in the database.
viii. CLICK ‘Delete’. To delete the registered patient from the database.
ix. CLICK ‘Back’.

Patient Registration JFrame:



8. Create Appointment.

Title: Create Appointment.

Description: Receptionist should be able to create an appointment.

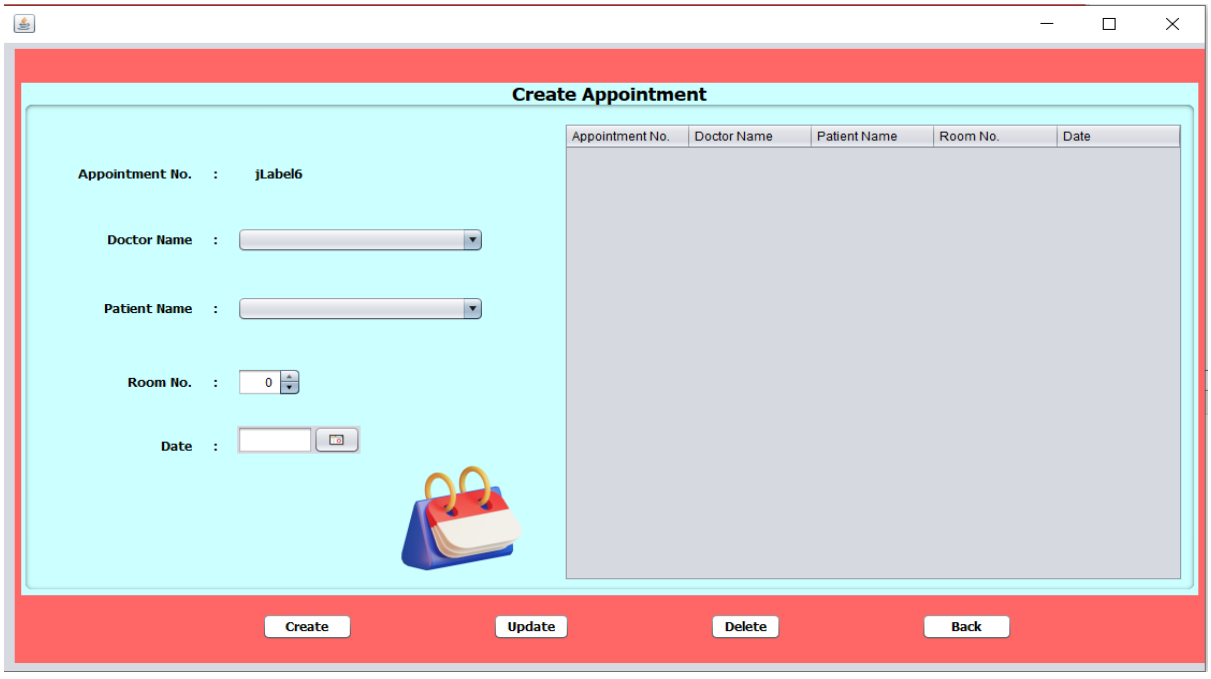
Precondition: Doctor name and Patient name database should be available

Assumption: Supported UI and database is being used.

Test Steps:

i. In the 'Appointment No.' field appointment number will be automatically be generated according to the sequence.
ii. In the 'Doctor name' combo box, names will be there for all the available doctor. Drop-down list would be displayed.
iii. In the 'Patient name' combo box, names will be available for all the registered patients. Drop-down list would be displayed.
iv. In the 'Room No.' JSpinner, receptionist would be able to allocate the room for the appointment.
v. In the 'Date' JDateChooser, receptionist should be able to choose the date of the appointment.
vi. CLICK 'Create'. To Add this entered details in the database.
vii. CLICK 'Update'. To update any of the previously entered details in the database.
viii. CLICK 'Delete'. To delete the registered patient from the database.
ix. CLICK 'Back'.

Create an appointment JFrame:



9. Doctor.

Title: Doctor

Description: Doctor should be able to see/update/add his/her information on the database.

Precondition: Doctor name should login with his/her correct ID and password to get the access to all the functions of the doctors.

Assumption: Supported UI and database is being used.

Test Steps:

i. In the 'Doctor No.' field, doctor number should be generated automatically as per the sequence.
ii. In the 'Doctor name' field, doctor has to provide his/her correct name.
iii. In the 'Specialization' field, doctor has to provide his/her correct specialization.
iv. In the 'Qualification' field, doctor has to provide his/her correct qualification.
v. In the 'Phone No.' field, doctor has to provide his/her correct phone number.
vi. In the 'Room No.' JSpinner, doctor has to select the room number.
vii. CLICK 'Add'. To Add this entered details in the database.
viii. CLICK 'Update'. To update any of the previously entered details in the database.
ix. CLICK 'Delete'. To delete the registered patient from the database.
x. CLICK 'Back'.

Doctor JFrame:

10. View Appointment.

Title: View Appointment

Description: Doctor should be able to view all the appointment created by receptionist and create prescription on particular Appointment number.

Precondition: A Appointment should be scheduled from before which will be visible here in jTable1.

Assumption: Supported UI and database is being used.

Test Steps:

i. 'jTable1' is used to show the appointments with appointment details .
ii. CLICK 'Prescription'. To create prescription for the selected Appointment No.
iii. CLICK 'Back'.

View Appointment JFrame:

11. View Prescription.

Title: View Prescription.

Description: Pharmacist should be able to view all the prescription created by doctor and create inventory on appointment number.

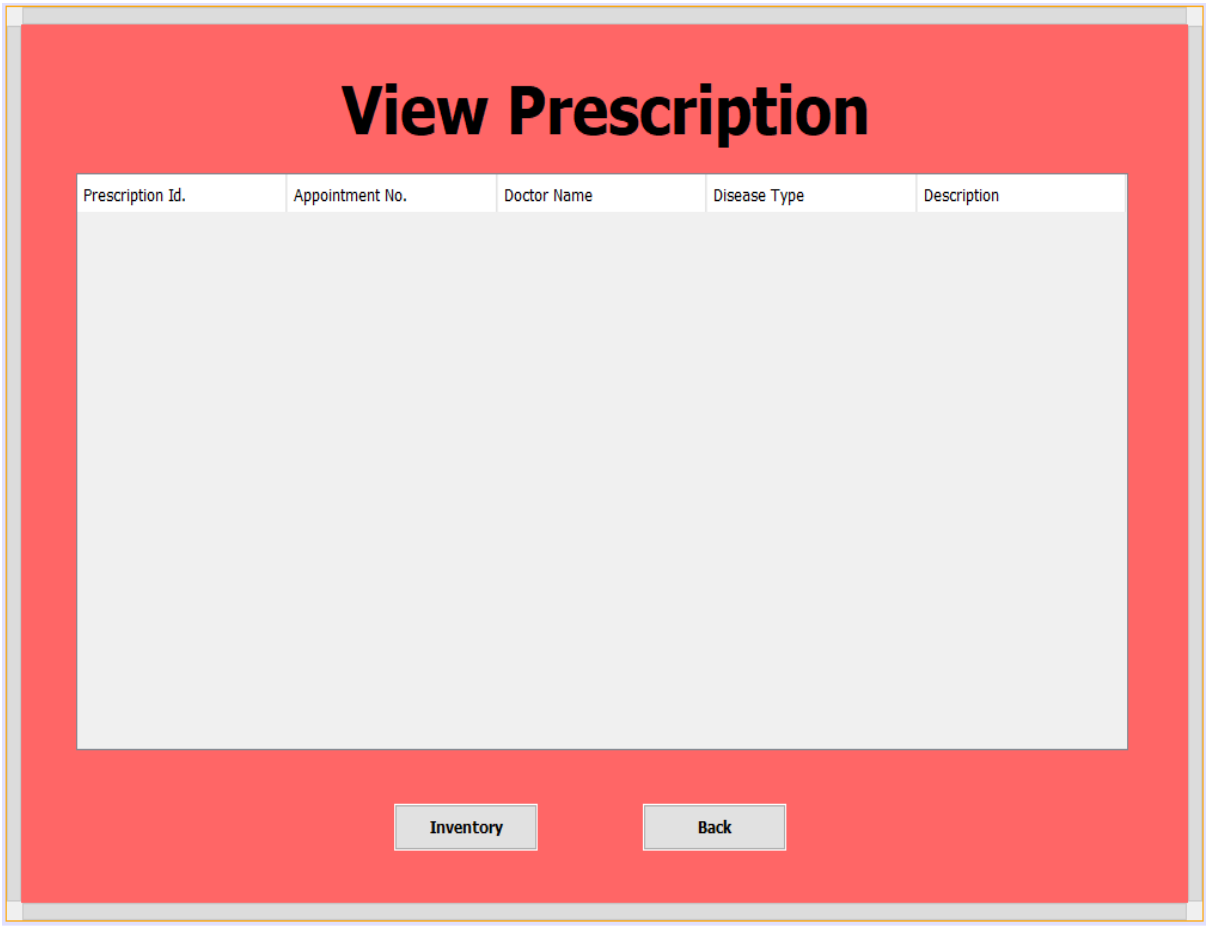
Precondition: A prescription should be created from before which will be visible here in jTable1.

Assumption: Supported UI and database is being used.

Test Steps:

i. ‘jTable1’ is used to show the prescription created with prescription details .
ii. CLICK ‘Inventory’. To create Inventory for the selected Prescription Id.
iii. CLICK ‘Back’.

View Prescription JFarme:



12. Create Item.

Title: Create Item.

Description: pharmacist should be able to create/add items in the stock which he receives.

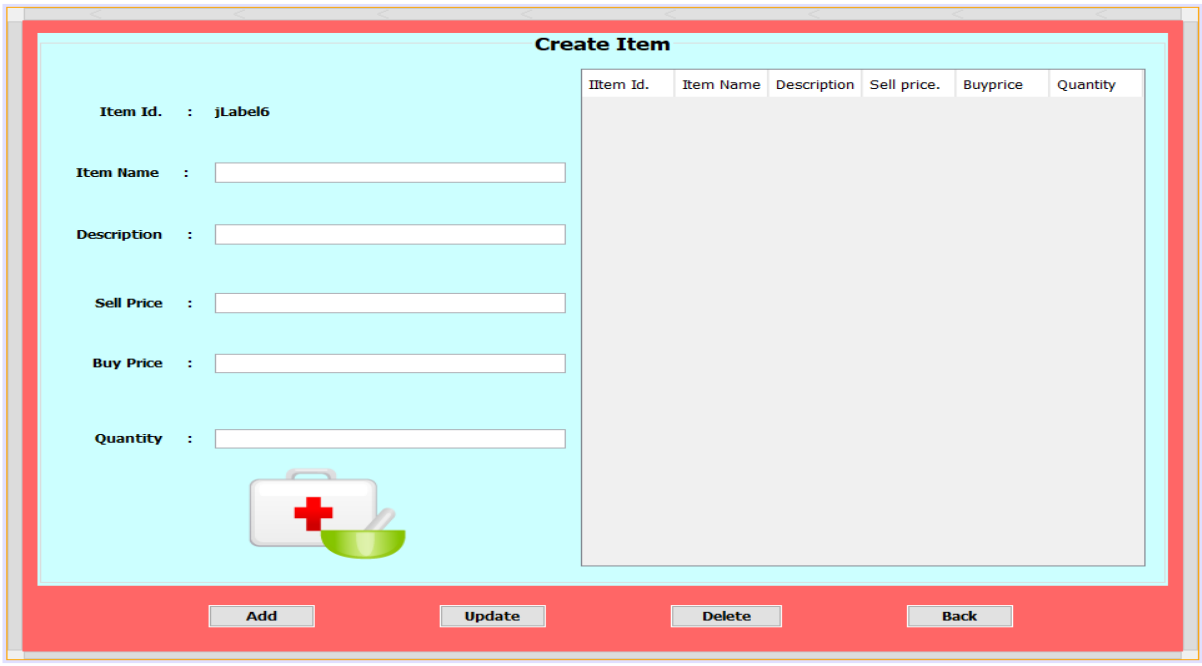
Precondition: Pharmacist should know all the drugs names, buy price and sell price.

Assumption: Supported UI and database is being used.

Test Steps:

i. In the 'Item Id.' field Item number will automatically be generated according to the sequence.
ii. In the 'Item name' field, Item name should be provided.
iii. In the 'Description' field, Description of the item should be provided.
iv. In the 'Sell Price' field, Sell Price of the item should be provided.
v. In the 'Buy Price' field, Buy price of the item should be provided.
vi. In the 'Quantity' field, Quantity of the item to be added should be provided.
vii. CLICK 'Add'. To Add this entered details in the database.
viii. CLICK 'Update'. To update any of the previously entered details in the database.
ix. CLICK 'Delete'. To delete the registered patient from the database.
x. CLICK 'Back'.

Create Item JFrame:



13. View Doctor.

Title: View Doctor

Description: Every user of the HMS should be able to view the available doctors in the database with their information.

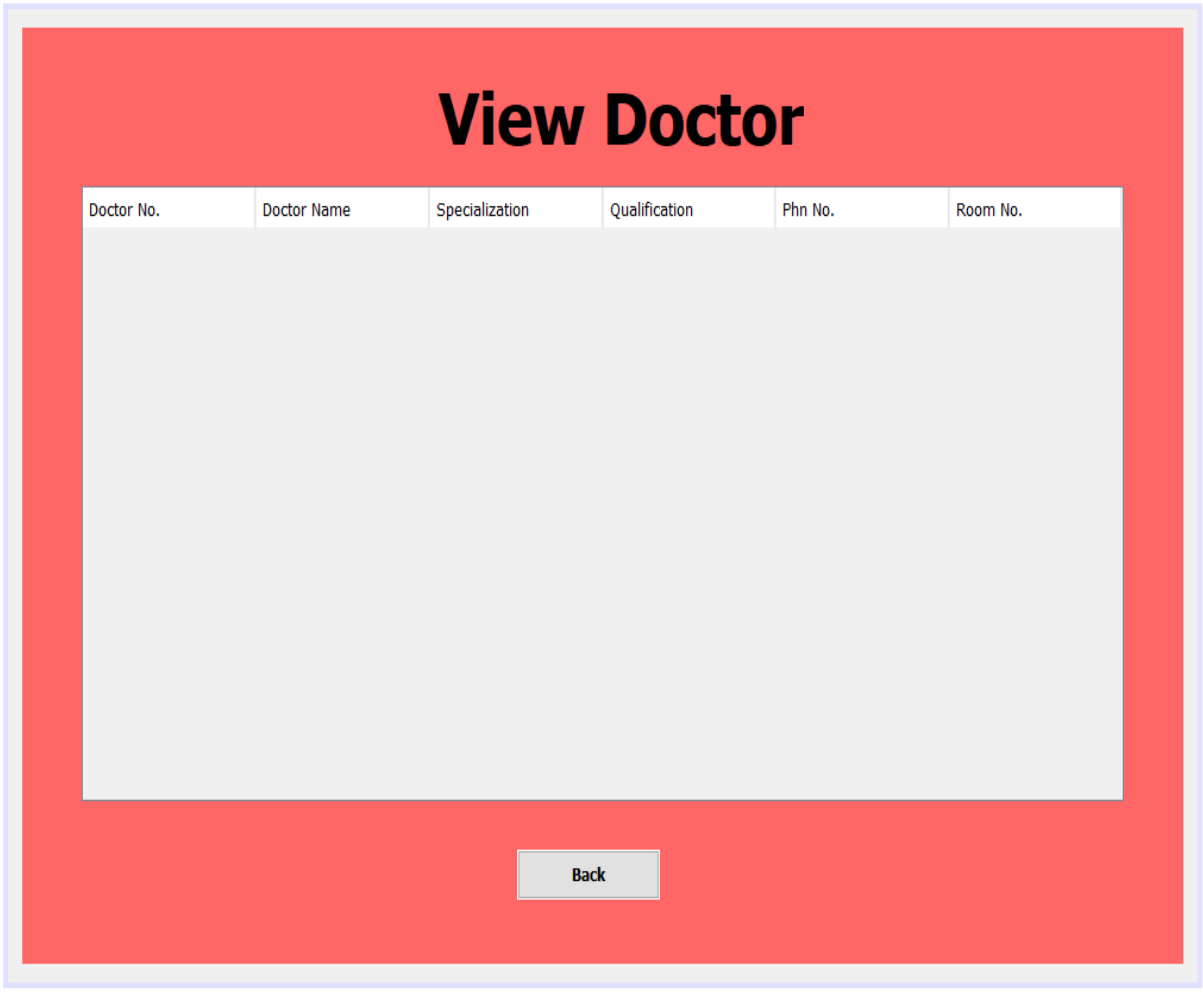
Precondition: Doctor should provide his/her's information in the doctor function provided to the doctor this information provided is used in the jTable1.

Assumption: Supported UI and database is being used.

Test Steps:

i. 'jTable1' is used to show all the doctors available in the database with their information .
ii. CLICK 'Back'.

View Doctor JFrame:



14. Prescription.

Title: Prescription

Description: Doctor can create prescription for any appointment number just by selection particular appointment number. Here he can provide medication based on the disease.

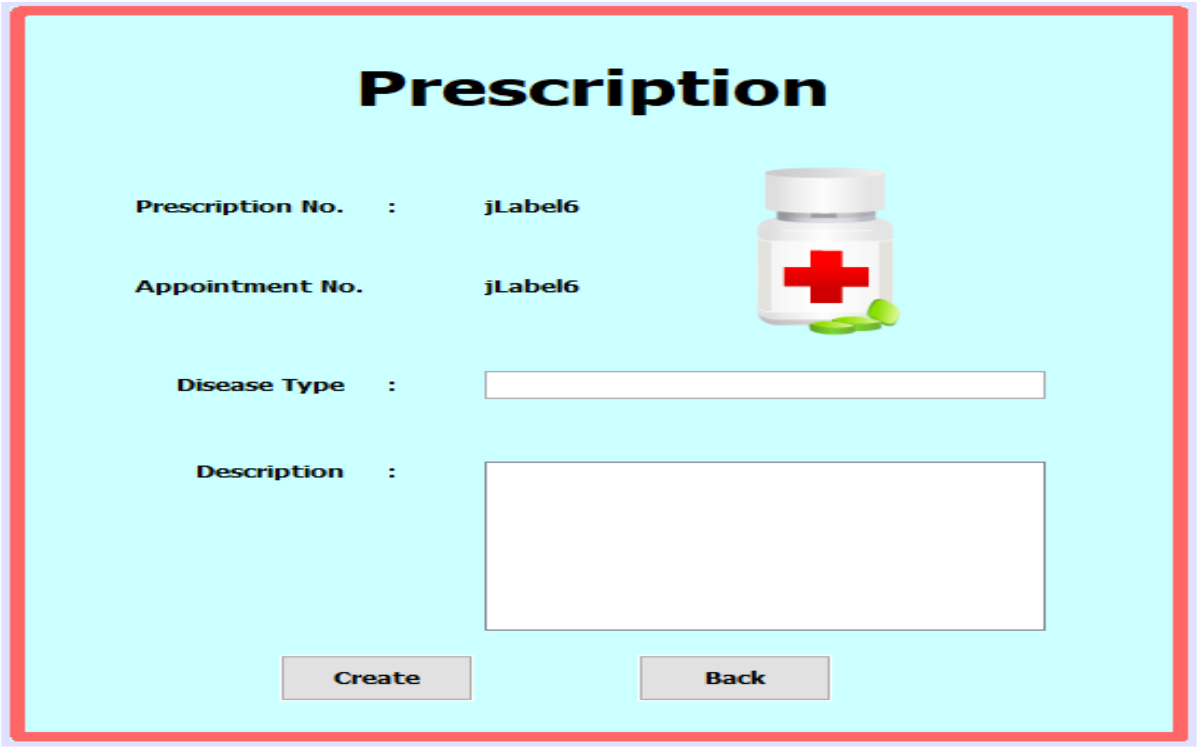
Precondition: A appointment has to be already created on which doctor can provide medication.

Assumption: Supported UI and database is being used.

Test Steps:

i. In the 'Prescription No.' field prescription number will be automatically generated as per the sequence.
ii. In the 'Appointment No.' field Appointment number will be automatically generated as per the selected appointment number whose prescription is been created.
iii. In the 'Disease Type' field, Doctor should provide the disease.
iv. In the 'Description' field, Doctor should provide the description of the disease and medication to the disease.
v. CLICK 'Create'. To Add this entered details in the database.
vi. CLICK 'Back'.

Prescription JFrame:



15. Inventory.

Title: Inventory

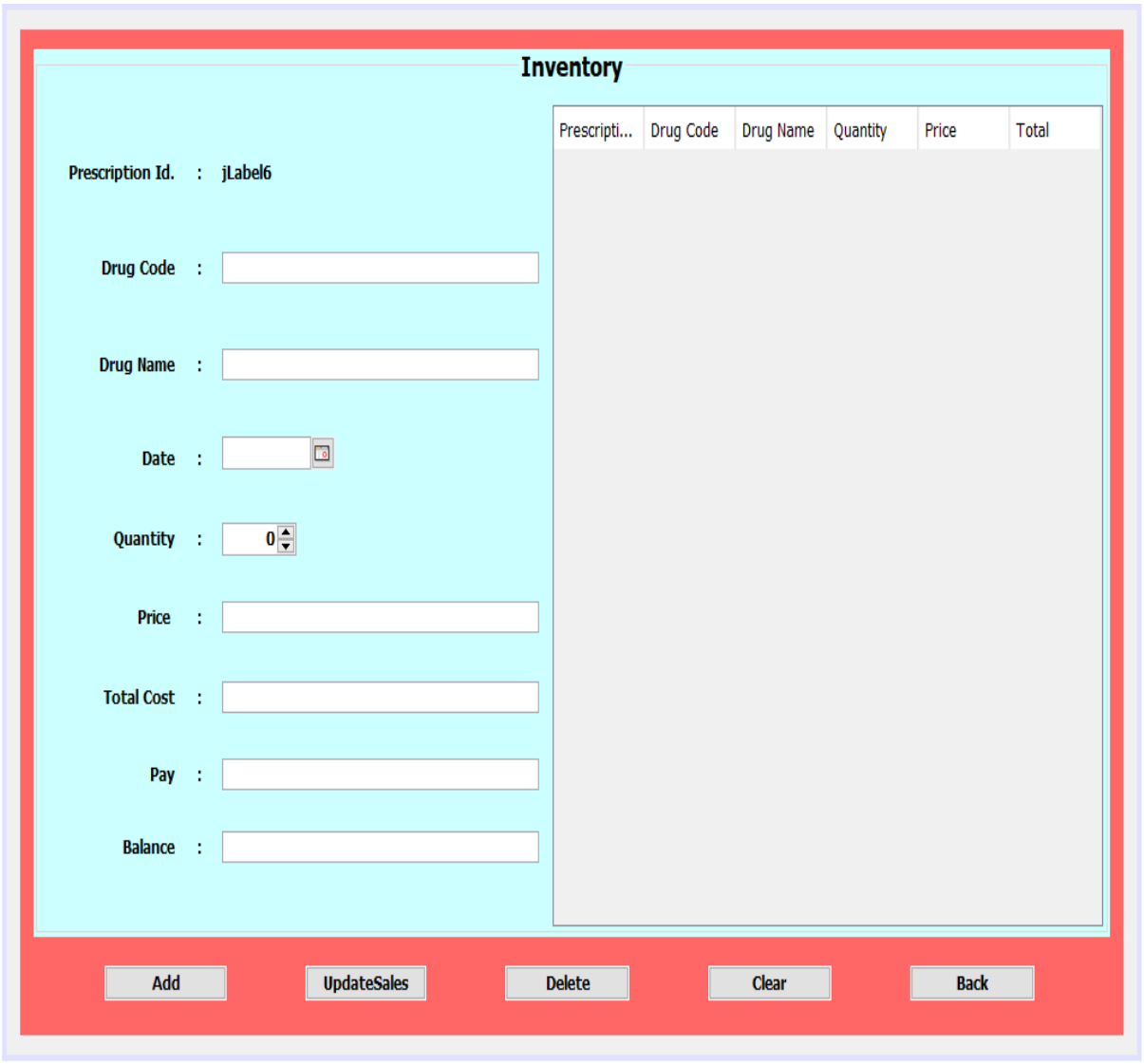
Description: Pharmacist should be able to create inventory/bill for selected prescription id.

Precondition: A prescription has to be already created whose bill can be generated here.

Assumption: Supported UI and database is being used.

Test Steps:

i. In the ‘Prescription Id.’ Field, prescription Id. will automatically be generated according to the selected prescription to generate Inventory.
ii. In the ‘Drug Code’ field, Item name should be provided.
iii. In the ‘Drug Name’ field, Description of the item should be provided.
iv. In the ‘Date’ JDateChooser, pharmacist should be able to choose the date of the bill.
v. In the ‘Quantity’ field, Quantity of the drug as per prescription should be provided.
vi. In the ‘Price’ field, Sell Price of the item should be provided.
vii. In the ‘Total Cost’ field, it will be automatically generated by the formula (Quantity * Price).
viii. In the ‘Pay’ field, pharmacist should enter the amount paid by the patient.
ix. In the ‘Balance’ field, it will be automatically generated by the formula (Pay-Total Cost). i.e amount that pharmacist has to give back to patient.
x. CLICK ‘Add’. To Add this entered details in the database.
xi. CLICK ‘Clear’. To clear all the text fields.
xii. CLICK ‘Delete’. To delete the registered patient from the database.
xiii. CLICK ‘Back’.
xiv. CLICK ‘UpdateSales’. Clicking this will update the information in SalesReport database which pharmacist can keep for his monthly or daily sales report.


Inventory
Prescription Id. : jLabel6
Drug Code :
Drug Name :
Date :
Quantity : 0
Price :
Total Cost :
Pay :
Balance :
Prescripti...
Drug Code
Drug Name
Quantity
Price
Total
Add
UpdateSales
Delete
Clear
Back

16. Sales Report.

Title: Sales Report

Description: Pharmacist can view all his Sales report here with all the details.

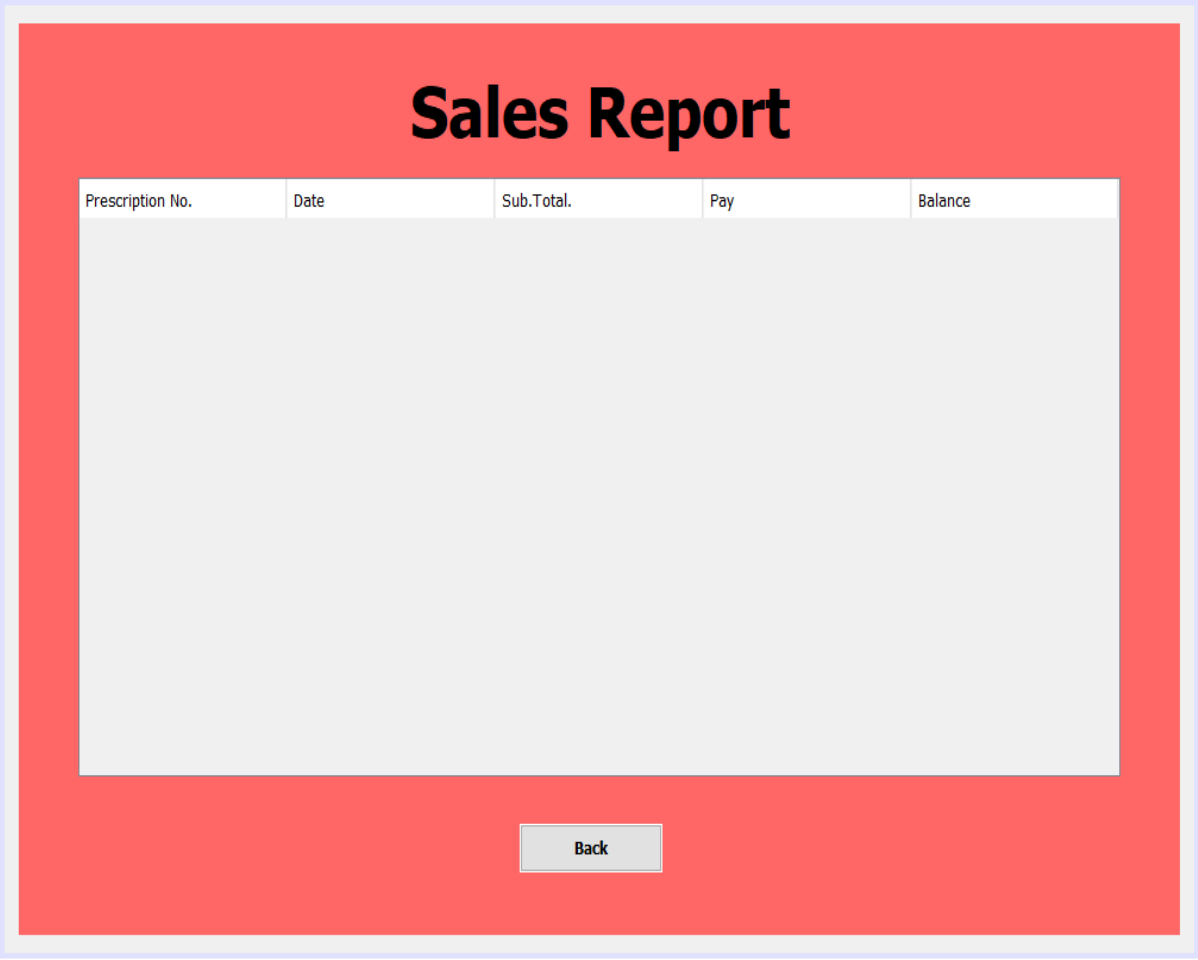
Precondition: Inventory has to be already created with correct details.

Assumption: Supported UI and database is being used.

Test Steps:

i. ‘jTable1’ is used to show all the SalesReport available in the database with information .
ii. CLICK ‘Back’.

Sales Report JFrame:



9. Conclusion

The project Hospital Management System (HMS) is for computerizing the working in a hospital. It is a great improvement over a manual system. The computerization of the system has speed up the process. In the current system, the front office managing is very slow. The software takes care of all information related to patients that come up to the hospital.

It also provides billing facilities based on patient's status whether it is indoor or outdoor patient. It also includes facilities like pharmacy system for the stock details of medicines in the pharmacy. Providing such enables the users to include more comments into the system. Also, it was our great experience to work on this project, we enjoyed it a lot and hope this project full fills all the needs of a hospital.

10. References

www.youtube.com

www.javatpoint.com

Trail: Creating a GUI with JFC/Swing (The Java™ Tutorials) (oracle.com)

www.geeksforgeeks.org

www.coursera.org

www.stackoverflow.com

draw.io