PES University, Bengaluru

(Established under Karnataka Act 16 of 2013)

Department of Computer Science & Engineering

Session: Jan - May 2022

UE19CS353 – Object Oriented Analysis and Design with Java

Theory ISA (Mini Project)

Report on

Pharmacy Inventory Management System

By:

B Pravena – PES2UG19CS076

Bharath Kumar S P – PES2UG19CS087

Bhuvantej R – PES2UG19CS092

6th Semester - B section

Table of Contents

Tables and Figures

1. Project Description

Pharmacy management system stores and manages Medicine, company and agent details.

●It helps in storing the data, organizes the entire system, controls the use of medication & improves customer satisfaction.

●Our project helps the business owner to control their stock, accounting and choose the right medication. With all the basic and advanced pharmacy features it helps in controlling all the business activities .

●Agents are able to sell the stock to different vendors and automatic bills get printed.

Link to Github repository -
https://github.com/bkumarsp/PharmacyInventoryManagementSystem

2. Analysis and Design Models

1) USE CASE DIAGRAM -:

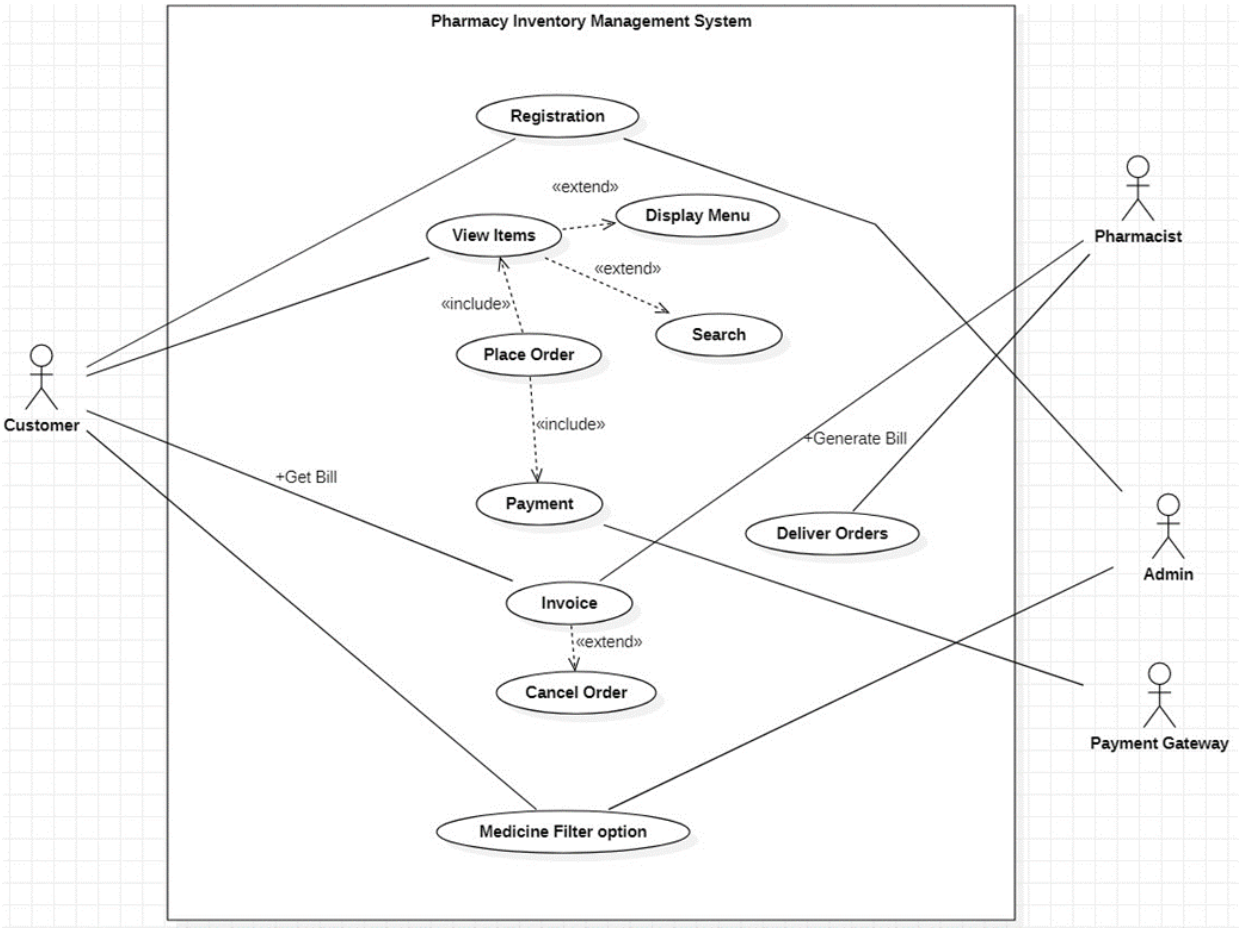

Fig1. Use Case Diagram

Name – Pharmacy Inventory Management System
Summary/Overview – Searching and Ordering medicines online
Actors – Customer, Pharmacist, Admin, Payment Gateway
Pre-conditions - Medicine must be available to place order

- Customer must have sufficient funds to do online payment

Description – Customer initially registers. They can then view items, look at the menu and search for desired medicines. They then place the order and make the payment. The payment gateway such as Google Pay, Net Banking, etc. The Pharmacist delivers the orders. He/ She can generate the bill/ invoice. The customer can cancel the order.
Exceptions - Insufficient funds, cannot identify account, power failure

2) Class Diagram -:

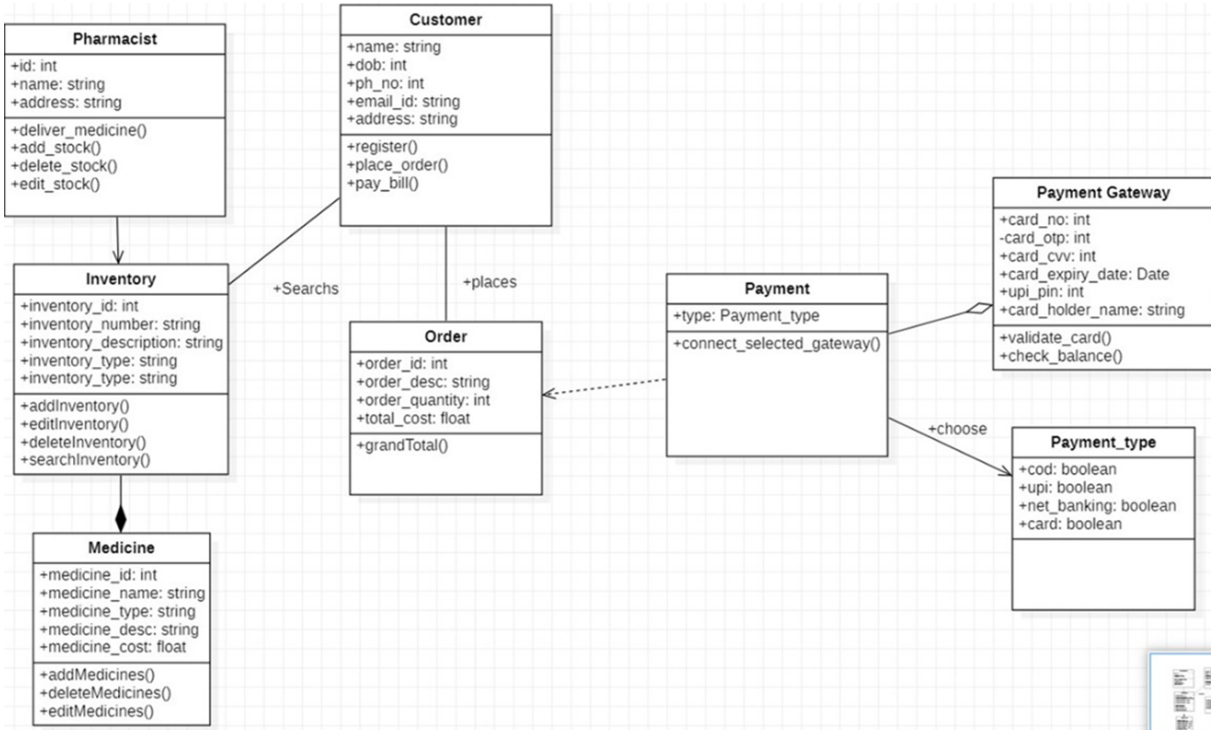

Fig2. Class Diagram

Specifications-:

Class **Customer** has their details such as name, date of birth, phone number, etc. These are accepted as data members. The functions that can be performed by them are register(), place_order(), pay_bill(), etc. Class **Inventory** had its unique id, number, description, etc. We can perform functions like add, edit, delete and search the inventory.

Class **Medicine** has composition relation with Class Inventory.

Class **Pharmacist** has data members as name, address, etc. They can deliver the medicine and also add, delete or edit the stock.

Class **Order** has id, quantity and the grandTotal() function to compute total cost.

Class **Payment** has attribute payment type which can be GPay, netbanking, etc. On knowing the type, we can connect to the selected one.

Class **Payment Gateway** is aggregated to class Payment. It takes in all the details of the card and checks if the card is valid and the balance.

Class **Payment Type** is extended from Payment and checks if UPI ID, netbanking, cash on delivery, etc.

3) State Diagram -:

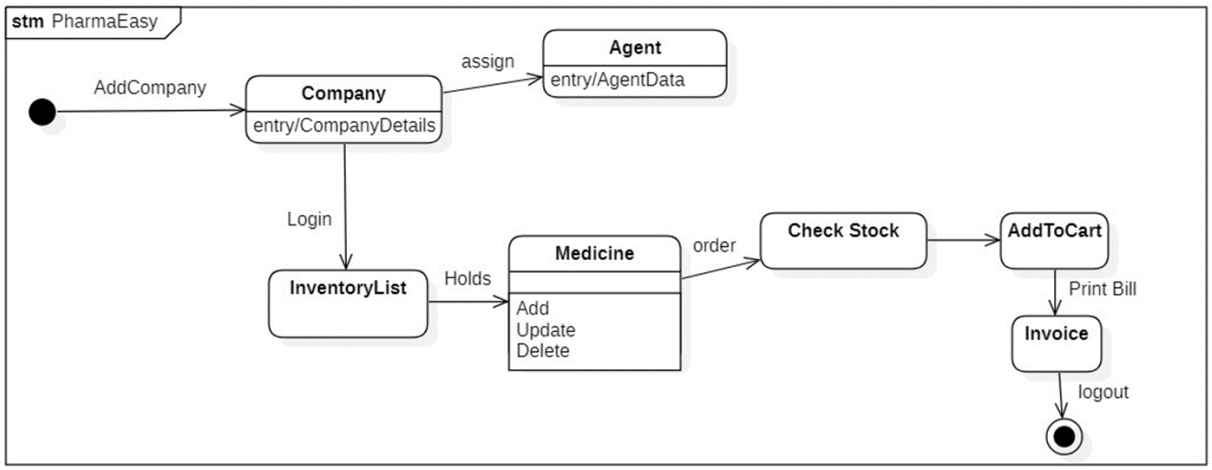

Fig3. State Diagram

4) Activity Diagram -:

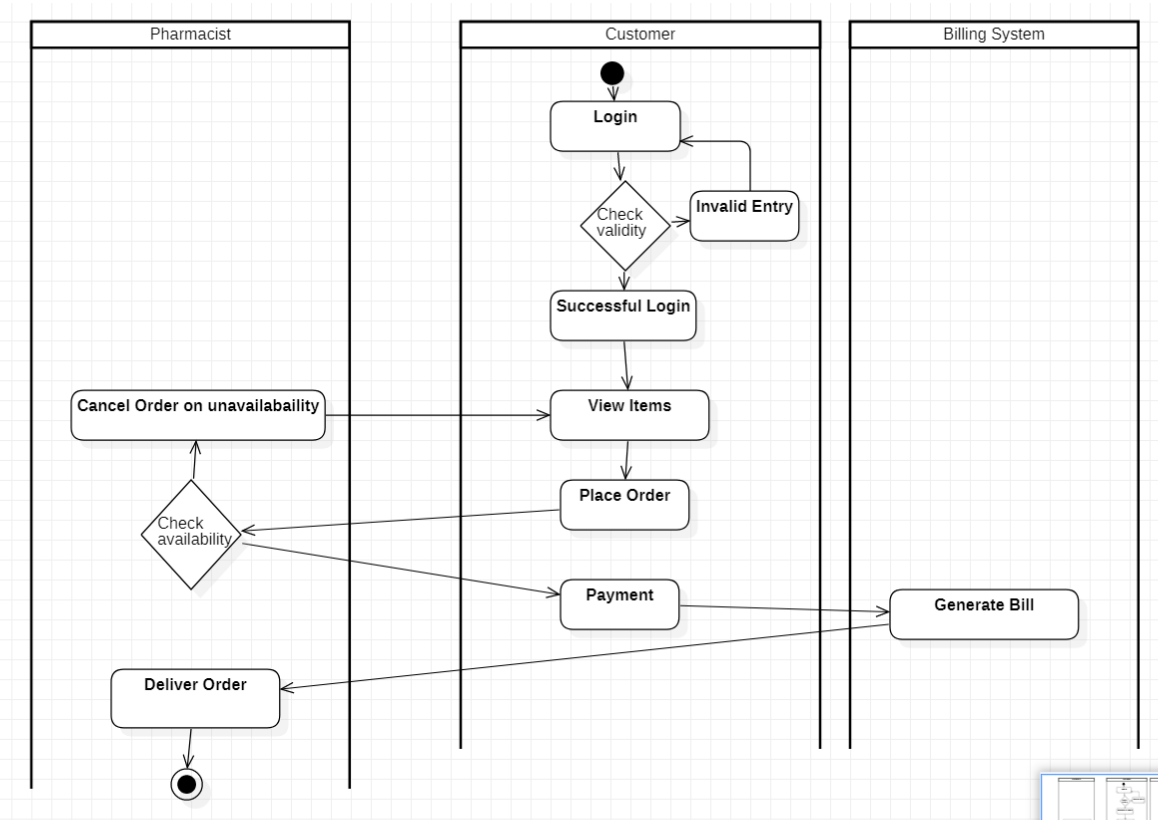

Fig4. Activity Diagram

3. Tools and Frameworks Used

IDE:NetBeans

NetBeans is an integrated development environment for Java. NetBeans allows applications to be developed from a set of modular software components called modules.

DATABASE: JavaDB

Java DB is a relational database management system (RDBMS). Data is stored and queried via SQL or JDBC.

FRAMEWORK: Swing

The Swing Application Framework is a Java specification for a simple application framework.

4. **Design Principles and Design Patterns Applied**

•Design Principle:- Single responsibility Principle(SRP)

–MedicineClass

–AgentClass

–CompanyClass

•Design Patterns:- Facade

–SellerClass

–Facade is a structural design pattern that provides a simplified interface to a complex system of classes, library or framework.

5. Application Screenshots

1) Inventory Page -:

Fig5. Inventory Page Screenshot

2) Company Page -:

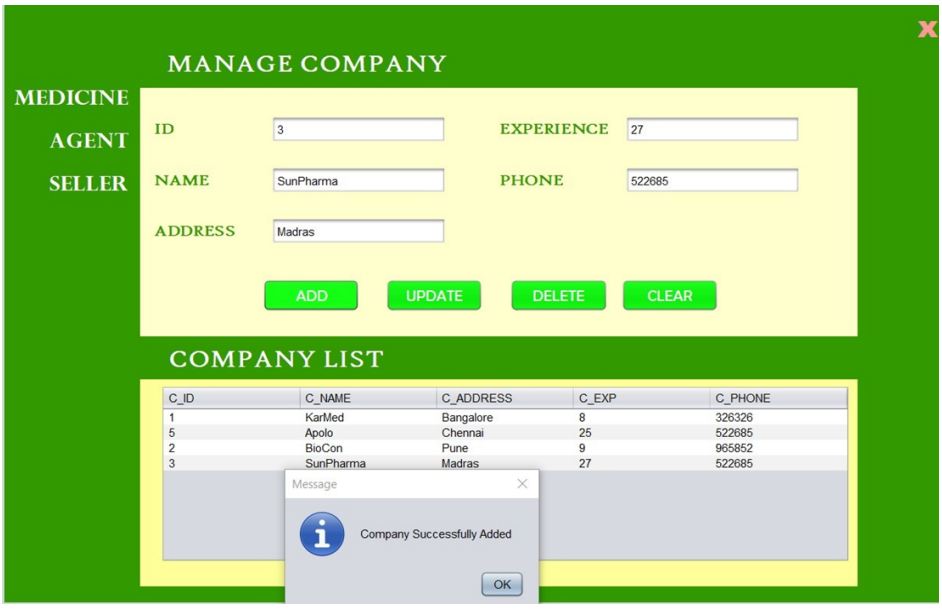

Fig6. Company Page Screenshot

3) Agent Page -:

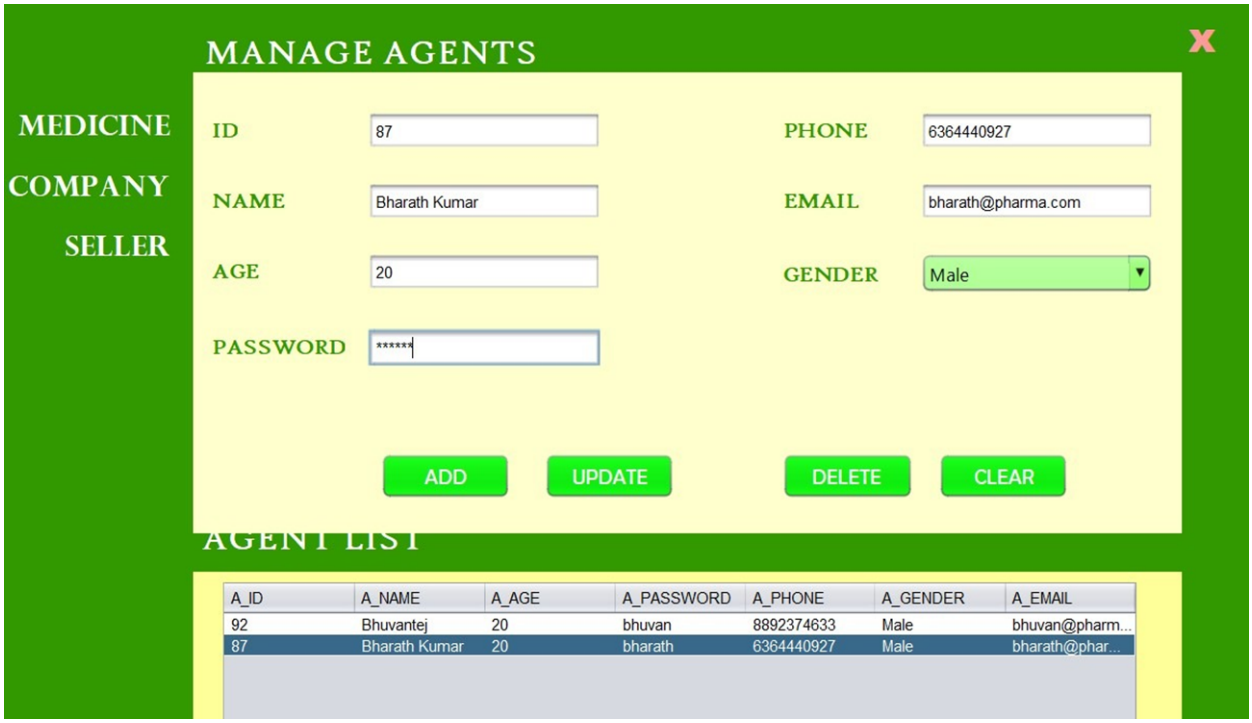

Fig7. Agent Page Screenshot

4) Seller Page -:

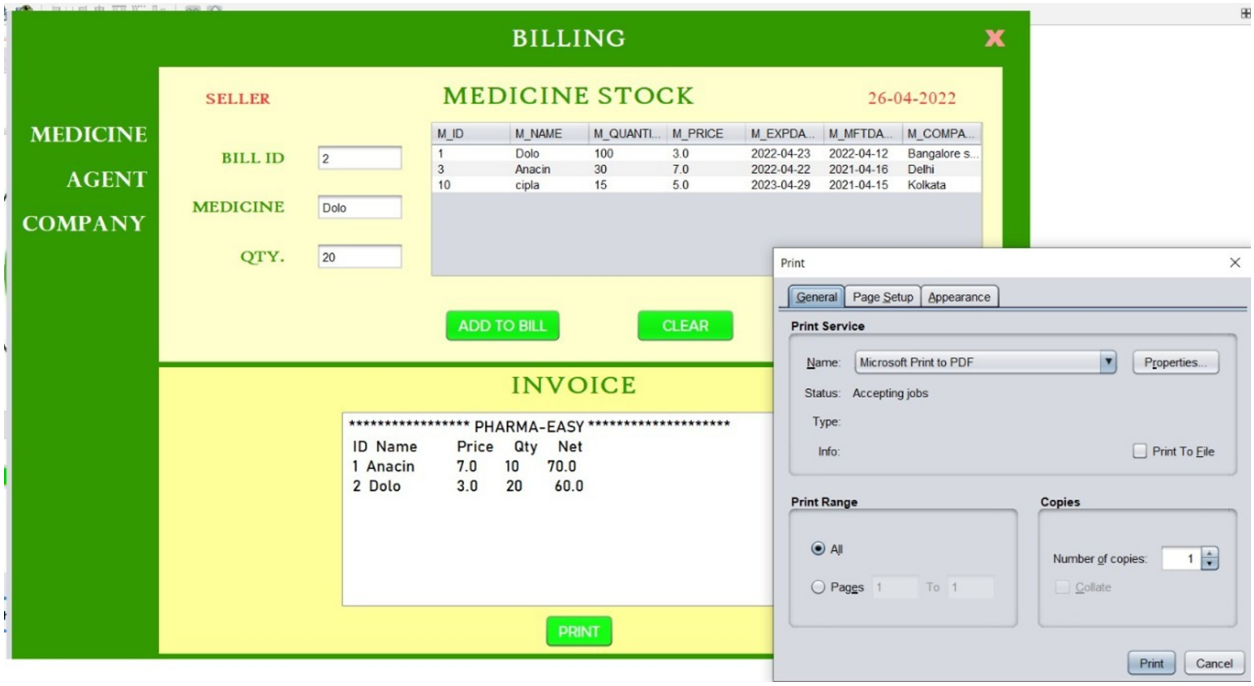

Fig8. Seller Page Screenshot

6. Team member contributions

B Pravena (PES2UG19CS076)	Bharath Kumar S P (PES2UG19CS087)	Bhuvantej R (PES2UG19CS092)
Medicine Class	Seller Class	Agent class
Company Class	Backend DB connection, Invoice	User table creations
DB queries for Inventory System	Bill generation and Medicine Queries	Database Query for Agent profile

Table1. Team Member Contributions

7. Conclusion

This project provides a user friendly environment for managing pharmacy inventory. This helps manage medicine, agents and company data. All the classes implement CRUD operations and follow Single Responsibility Principle. The **company class, Agent class** and the **Medicine class** forms the base of our project and links to DB server directly. The **Seller class** is where customers can order medicines from the inventory. The data is stored in Java DB where we can easily perform SQL operations. Once the order is placed, the invoice will be generated and provides the Print option.

8. References

- https://netbeans.apache.org/kb/docs/java/quickstart-gui.html
- https://www.guru99.com/use-case-diagrams-example.html
- https://creately.com/blog/diagrams/class-diagram-relationships/
- https://www.tutorialspoint.com/uml/uml_activity_diagram.htm
- https://www.geeksforgeeks.org/unified-modeling-language-uml-state-diagrams/