**Project Name: Upcycle Products**

**Project Member:**

**Yogesh Mahajan** 220343120055

**Ajay Auti** 220343120018

**Chinmay Alandkar** 220343120009

**Govind Kulkarni** 220343120051

**Abstract:**

There are lots of people who want their old things convert into new one but they didn’t get the right shop or person which will do their work properly. So Upcycle is a platform which is going to help to find the right person and provide a platform. This will connect user with the shop and shop will provide customize products as user want

* Admin

The admin will have control over the entire system. There is no registration for him but he should have username and password to login into the system to provide security. He can view everyone's profiles associated with this project.

* Shop Management

Shops can be accessible to their portal only through their credentials. They also have access to add products and remove products which are upcycle products. They will able to register and login themselves by their own.

* End User management

User can be accessible to their portal only through their login credentials. They also have access to register and login them by themselves. They can be able to view the products which were added by the shops.

* Technologies: Java, Spring Boot, Hibernate, MySQL, React Js

**Implementation Technologies:**

1. **Spring Framework:**

Spring Framework is a Java platform that provides comprehensive infrastructure support for developing Java applications. Spring handles the infrastructure so you can focus on your application.

Spring enables you to build applications from “plain old Java objects” (POJOs) and to apply enterprise services non-invasively to POJOs. This capability applies to the Java SE programming model and to full and partial Java EE.

**1.1 Features of Spring Framework:**

**1. Lightweight**

Spring is modular lightweight framework which allows you to selectively use any of its modules on the top of Spring Core.

**2. Inversion of Control (IOC)**

This is another top feature of Spring Framework where application dependencies are satisfied by the framework itself. Framework creates the object in runtime and satisfies application dependencies.

**3. Aspect Oriented Programming (AOP)**

Aspect Oriented Programming (AOP) is very popular in programming world and in Spring it is well implemented. Developer can use Aspect Oriented Programming (AOP feature of Spring to develop application in which business logic is separated from system services.

**4. Container**

Spring provides its own container for managing the bean lifecycle.

**5. MVC Framework**

Spring MVC Framework is used for developing MVC based web applications.

**6. Transaction Management**

Spring framework provides generic Transaction Management layer which can be used with or without J2EE(JEE) environment.

**7. JDBC Exception Handling**

Spring provides its own abstraction of JDBC exception which further simplifies the exception handling in program.

**1.2 Advantages of Spring Framework:**

**1. Solving difficulties of Enterprise application development**

Spring is solving the difficulties of development of complex applications, it provides Spring Core, Spring IoC and Spring AOP for integrating various components of business applications.

**2. Support Enterprise application development through POJOs**

Spring supports development of Enterprise application development using the POJO classes which removes the need of importing heavy Enterprise container during development. This makes application testing much easier.

**3. Easy integration other frameworks**

Spring designed to be used with all other frameworks of Java, you can use ORM, Struts, Hibernate and other frameworks of Java together. Spring framework do not impose any restriction on the frameworks to be used together.

**4. Application Testing**

Spring Container can be used to develop and run test cases outside enterprise container which makes testing much easier.

**5. Modularity**

Spring framework is modular framework and it comes with many modules such as Spring MVC, Spring ORM, Spring JDBC, Spring Transactions etc. which can use as per application requirement in modular fashion.

**6. Spring Transaction Management**

Spring Transaction Management interface is very flexible it can configure to use local transactions in small application which can be scaled to JTA for global transactions.

1. **Hibernate Framework**

Hibernate is used to overcome the limitations of JDBC like:

1. JDBC code is dependent database software being used. And if we change the database in middle of the project then it cost more and it includes the changes of logic.
2. JDBC code is not portable code across the multiple database software.
3. While working with JDBC, there is no support object-level relationship.

To overcome this we use hibernate framework.

Hibernate is a framework which provides some **abstraction layer**, meaning that the programmer does not have to worry about the implementations, Hibernate does the implementations for you internally like **Establishing a connection with the database, writing query to perform CRUD operations etc**.  
 It is a **java framework** which is used to develop persistence logic. Persistence logic means to store and process the data for long use. More precisely Hibernate is an open-source, non-invasive, light-weight java ORM(Object-relational mapping) framework to develop objects which are independent of the database software and make independent persistence logic in all JAVA, JEE.

**Framework** means it is special install-able software that provides an abstraction layer on one or more technologies like JDBC, Servlet, etc to simplify or reduce the complexity for the development process.

Functionalities supported by Hibernate Framework:

1. Supports auto DDL operations
2. Supports auto primary key generation
3. Supports hibernate query language(HQL)
4. Exception handling is not mandatory
5. Supports cache memory
6. It’s a ORM tool means supports object relational mapping

**2.1 ORM**

An ORM tool simplifies the data creation, data manipulation and data access. It is a programming technique that maps the object to the data stored in the database.



The ORM tool internally uses the JDBC API to interact with the database.

**3.** **MySQL**

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

**Features of MySQL:**

* **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

* **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment.

* **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything.

* **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

* **MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

1. **React:**

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. ‘V’ denotes the view in MVC. ReactJs is an open-source, component-based front end library responsible only for the view layer of the application. It is maintained by Facebook.

React uses a declarative paradigm that makes it easier to reason about your application and aims to be both efficient and flexible. It designs simple views for each state in your application, and React will efficiently update and render just the right component when your data changes. The declarative view makes your code more predictable and easier to debug.

A React application is made of multiple components, each responsible for rendering a small, reusable piece of HTML. Components can be nested within other components to allow complex applications to be built out of simple building blocks. A component may also maintain an internal state.

1. **Hardware and Software Requirements (Minimum):**

**Hardware:**

1. Intel i5 processor 10th generation

2. 4 GB RAM

3. Windows 10 Home edition.

4. 200 GB Data HDD Space

5. Data Connection 500 kbps

**Software:**

1. Eclipse 4.7 Oxygen
2. MySQL 5.7 with Workbench 8.0
3. Google Chrome version 93.0
4. Apache Tomcat Server 8.5
5. Maven Dependencies
6. **ER Diagram:**

Figure 1: ER Diagram

1. **Table Structures:**

**Table 1. Address**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** |  | **Key** |  |
| Id | int | No | PRI | auto-increment |
| City | varchar(255) | Yes |  |  |
| Country | varchar(255) | Yes |  |  |
| State | varchar(255) | Yes |  |  |
| Zip | varchar(255) | Yes |  |  |

**Table 2. Admin**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** |  | **Key** |  |
| userid | varchar(255) | No | PRI |  |
| pwd | varchar(255) | Yes |  |  |
| uname | varchar(255) | Yes |  |  |

**Table 3. Customer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** |  | **Key** |  |
| Id | int | No | PRI | auto-increment |
| City | varchar(255) | Yes |  |  |
| Created\_timestamp |  | Yes |  |  |
| Gender | varchar(255) | Yes |  |  |
| Name | varchar(255) | Yes |  |  |
| Phone | varchar(255) | Yes |  |  |
| userid | varchar(255) | Yes |  |  |

**Table 4. hibernate\_sequence**

|  |  |
| --- | --- |
| **Column** | **Type** |
| next\_val | bigint |

**Table 5. orderdetails**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** |  | **Key** |  |
| id | int | No | PRI | auto-increment |
| qty | int | Yes |  |  |
| ordered | int | Yes |  |  |
| productid | int | Yes |  |  |

**Table 6. orders**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** |  | **Key** |  |
| orderid | int | No | PRI | auto-increment |
| orderdate | datetime(6) | Yes |  |  |
| addressed | int | Yes |  |  |
| |  |  | | --- | --- | | customerId |  | | int | Yes |  |  |
| paymentId | int | Yes |  |  |

**Table 7. payments**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** |  | **Key** |  |
| id | int | No | PRI | auto-increment |
| amount | int | Yes |  |  |
| cardno | varchar(255) | Yes |  |  |
| nameoncard | varchar(255) | Yes |  |  |
| paymentdate | datetime(6) | Yes |  |  |

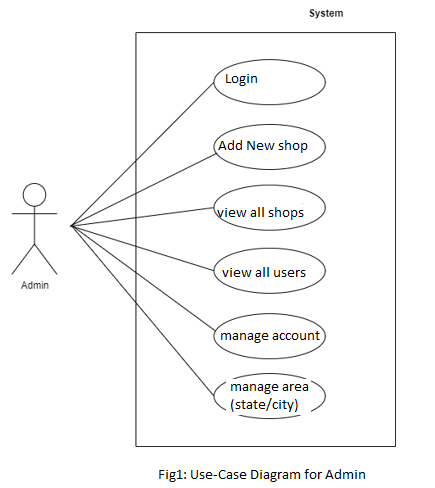
**Table 8. product**

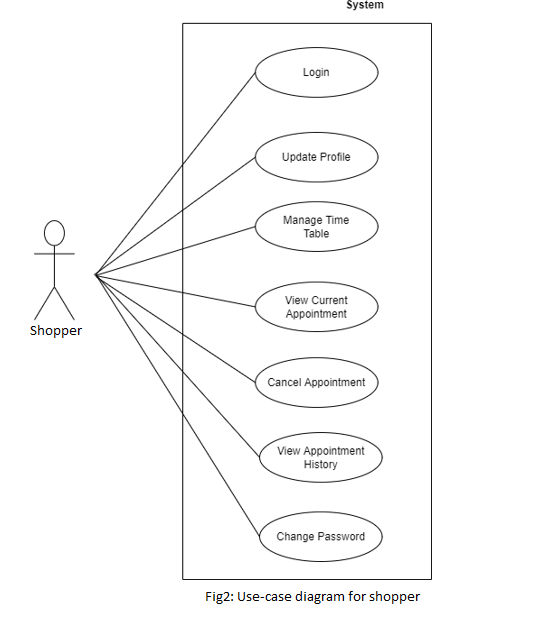
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** |  | **Key** |  |
| prodid | int | No | PRI |  |
| brand | varchar(255) | Yes |  |  |
| created\_timestamp | datetime(6) | Yes |  |  |
| pcat | varchar(255) | Yes |  |  |
| photo | varchar(255) | Yes |  |  |
| pname | varchar(255) | Yes |  |  |
| price | int | Yes |  |  |
| subcat | varchar(255) | Yes |  |  |
| sellerid | int | Yes |  |  |

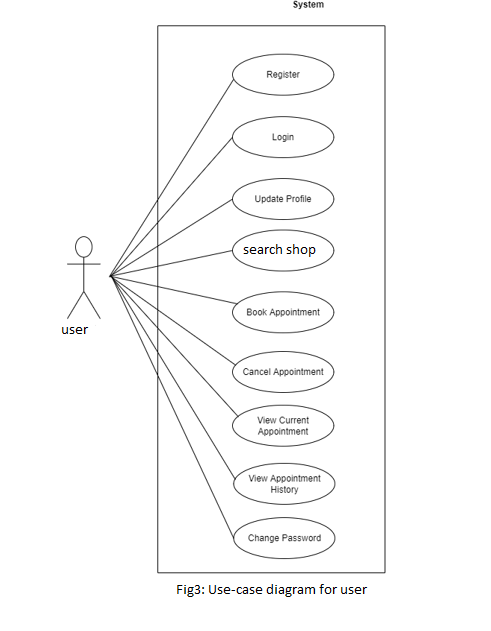
**Table 7. sellers**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Column** | | **Type** | |  | | **Key** | |  |
| id | int | | No | | PRI | | auto-increment | |
| city | varchar(255) | | Yes | |  | |  | |
| created\_timestamp | datetime(6) | | Yes | |  | |  | |
| name | varchar(255) | | Yes | |  | |  | |
| phone | varchar(255) | | Yes | |  | |  | |
| pwd | varchar(255) | | Yes | |  | |  | |
| userid | varchar(255) | | Yes | |  | |  | |

1. **UML Diagrams:**







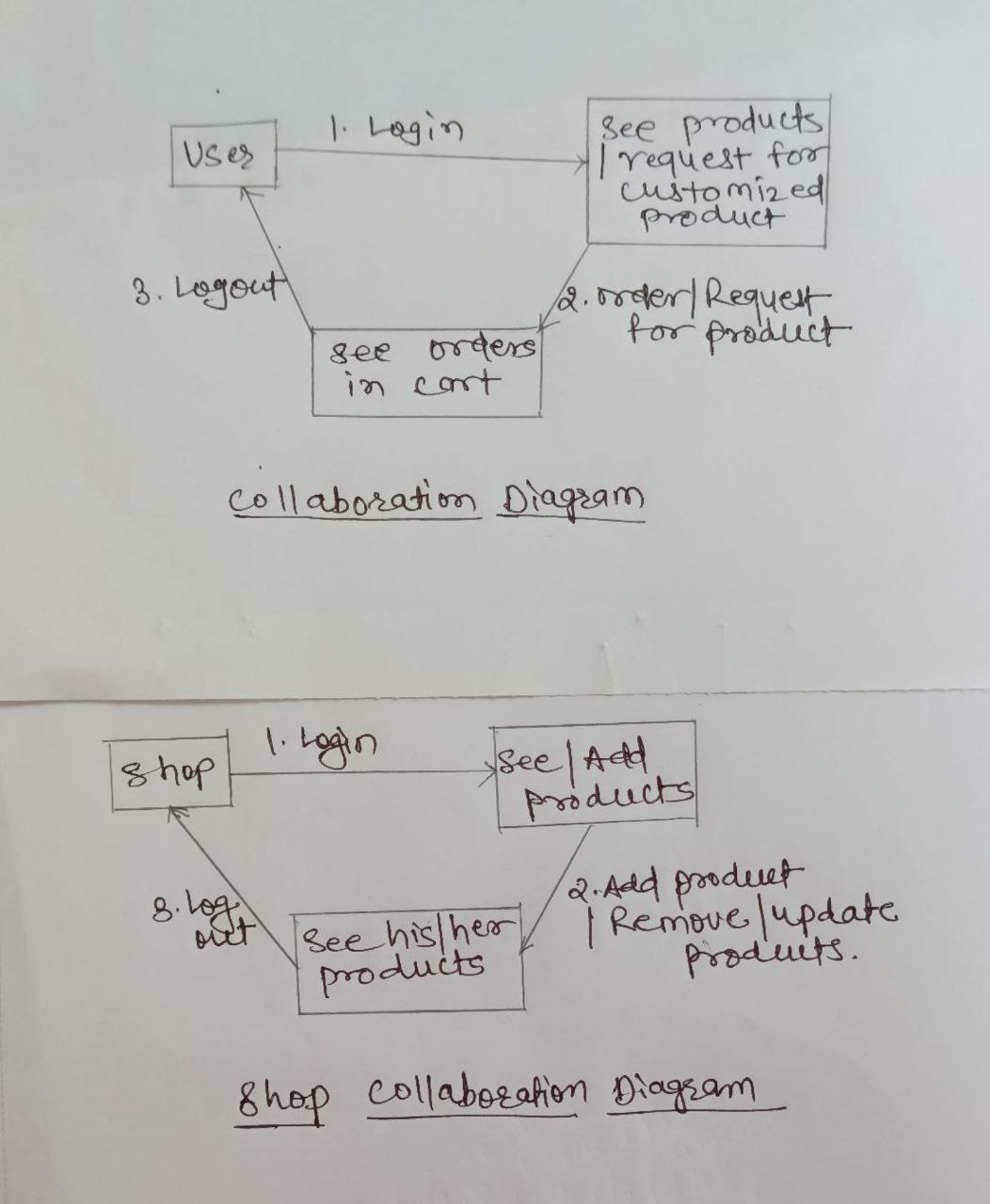
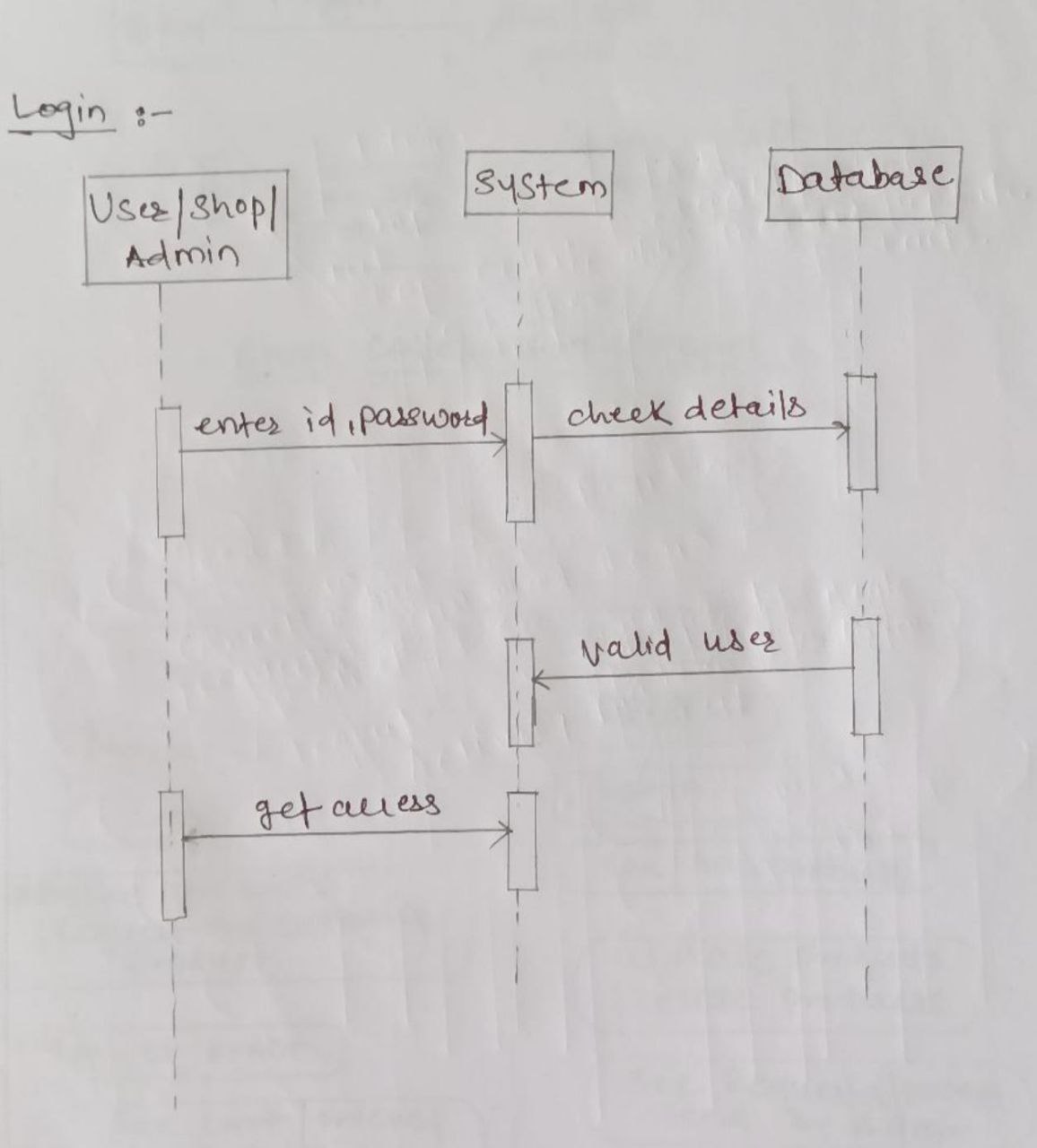


Figure 3: Collaboration Diagram



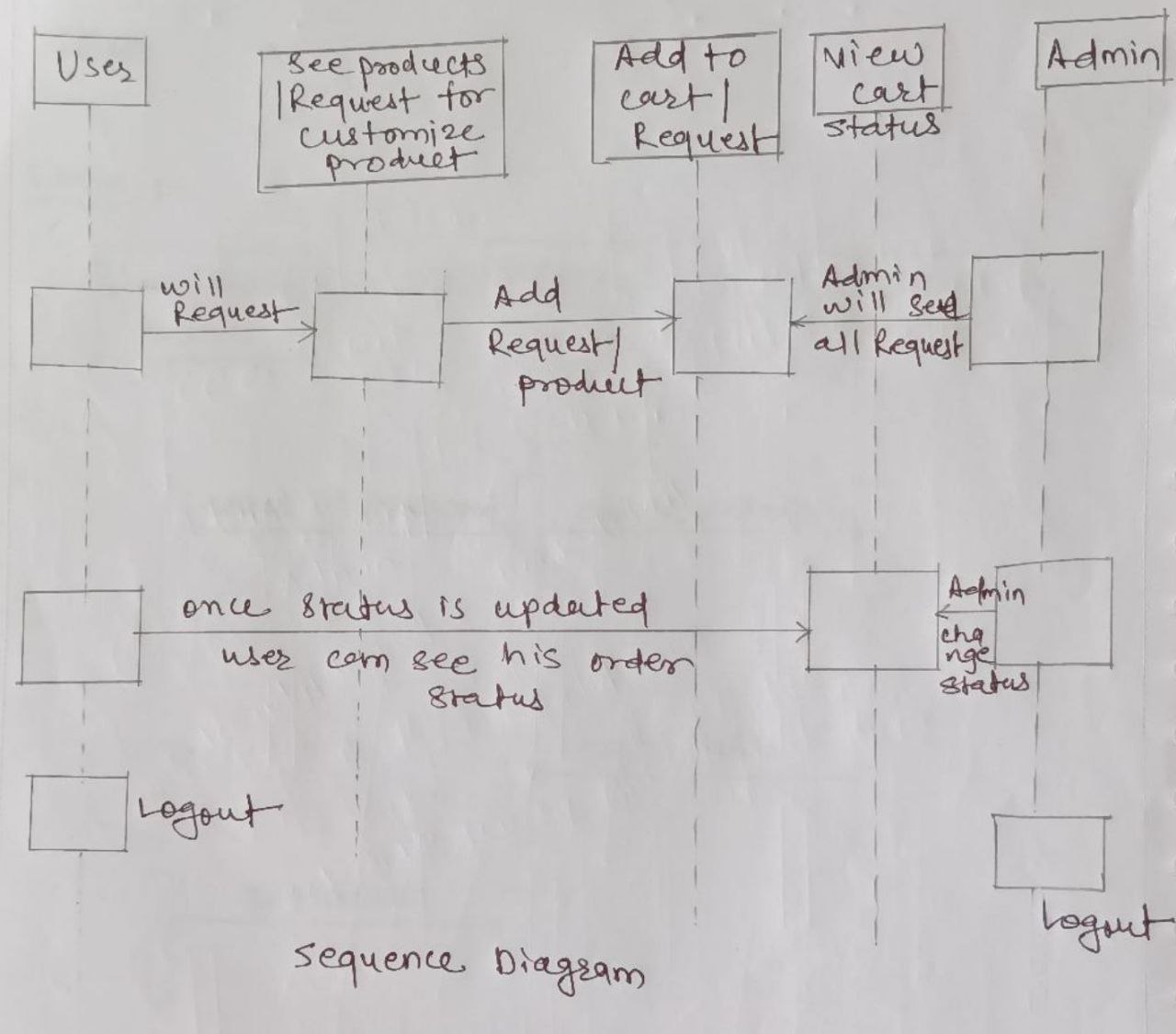


Figure 4: Sequence Diagram

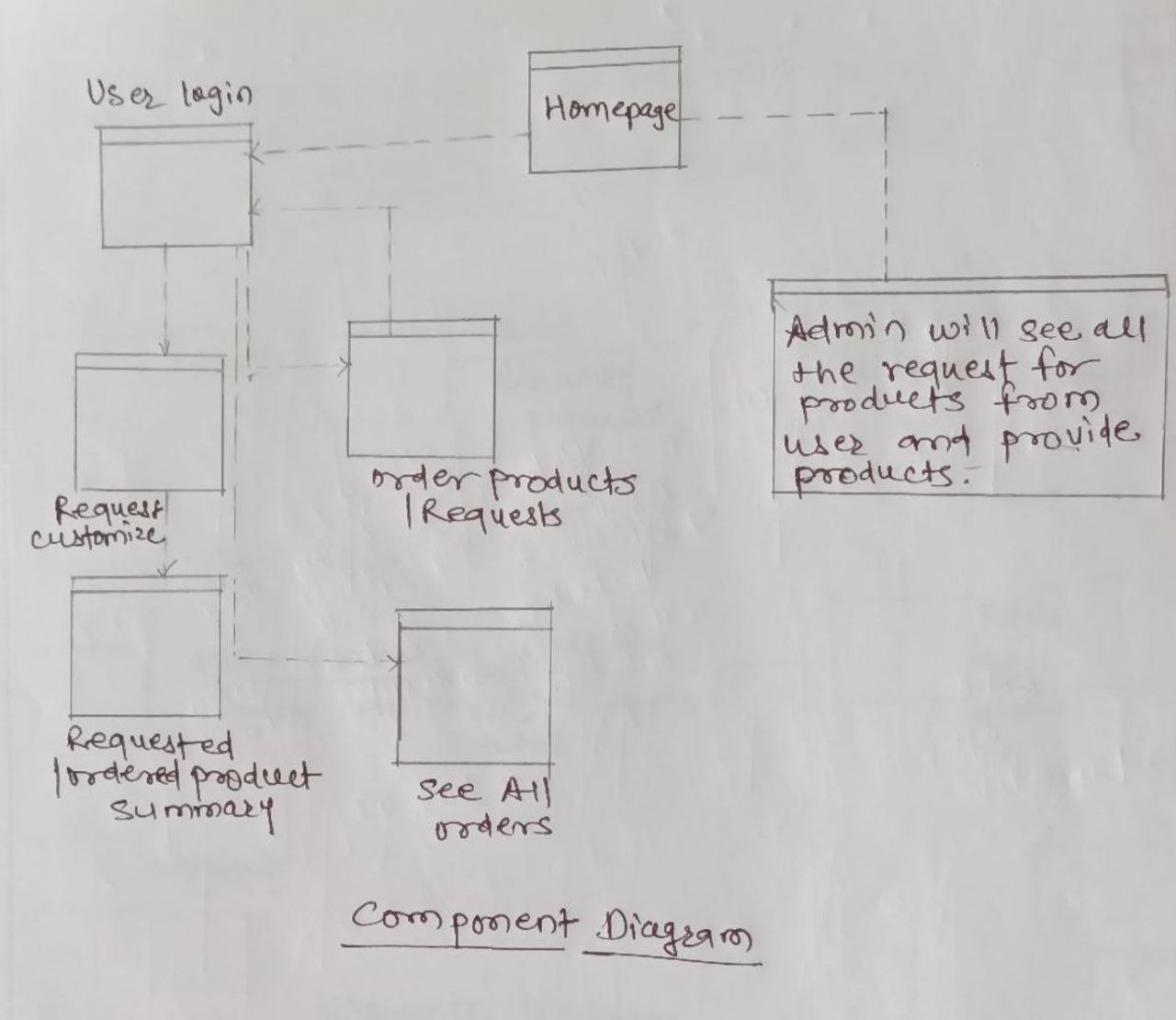


Figure 5: Component Diagram

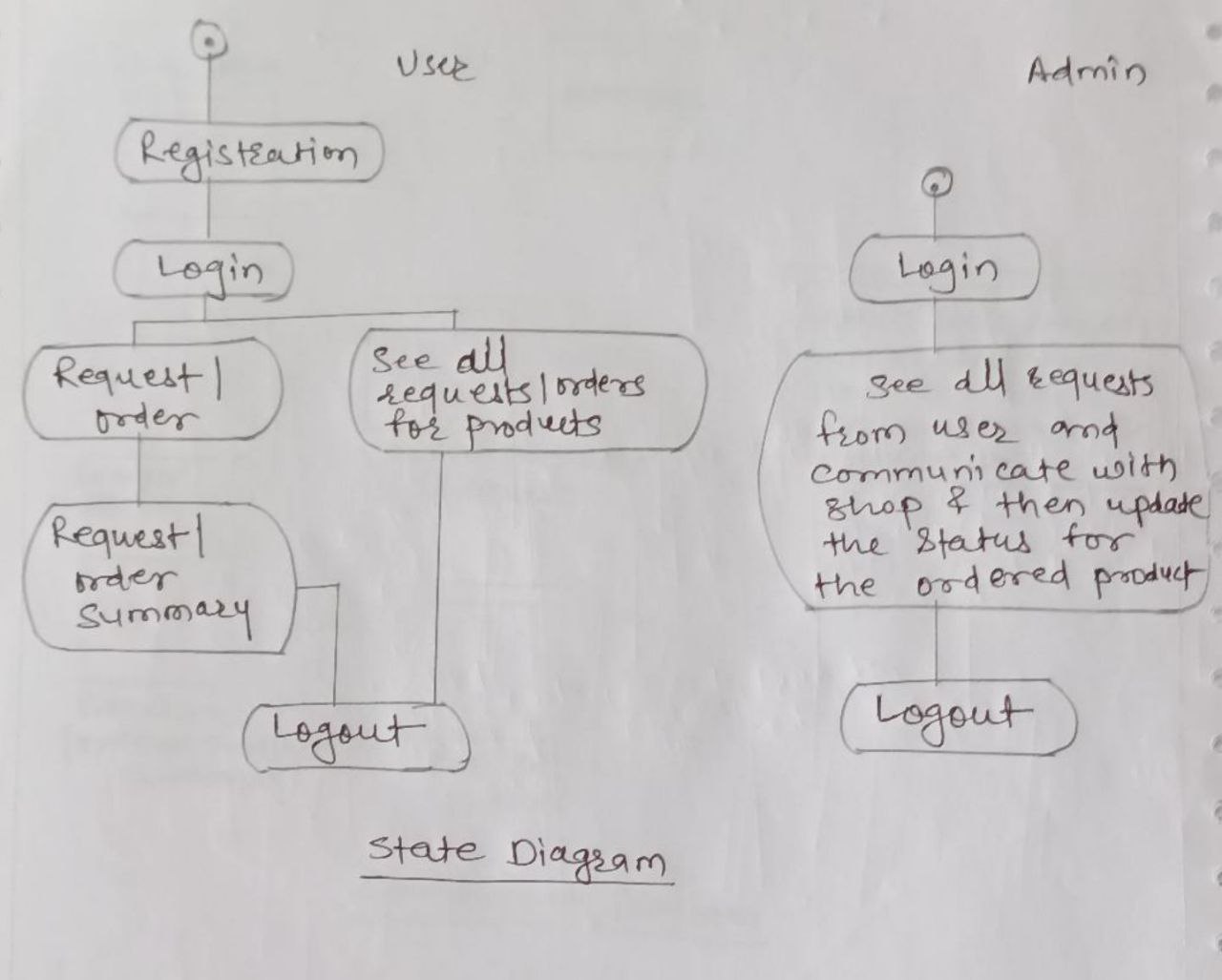


Figure 6: State Diagram

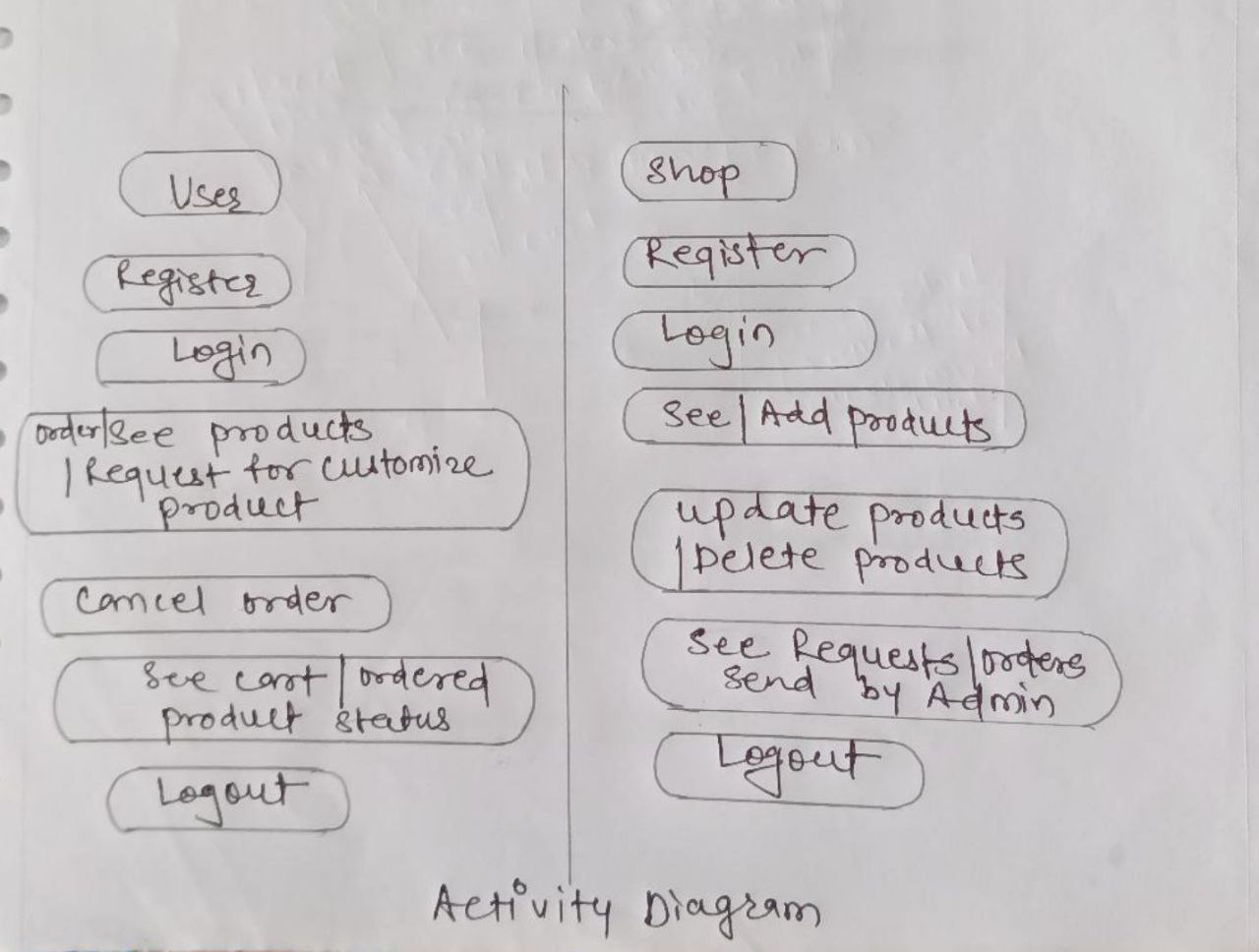


Figure 7: Activity Diagram

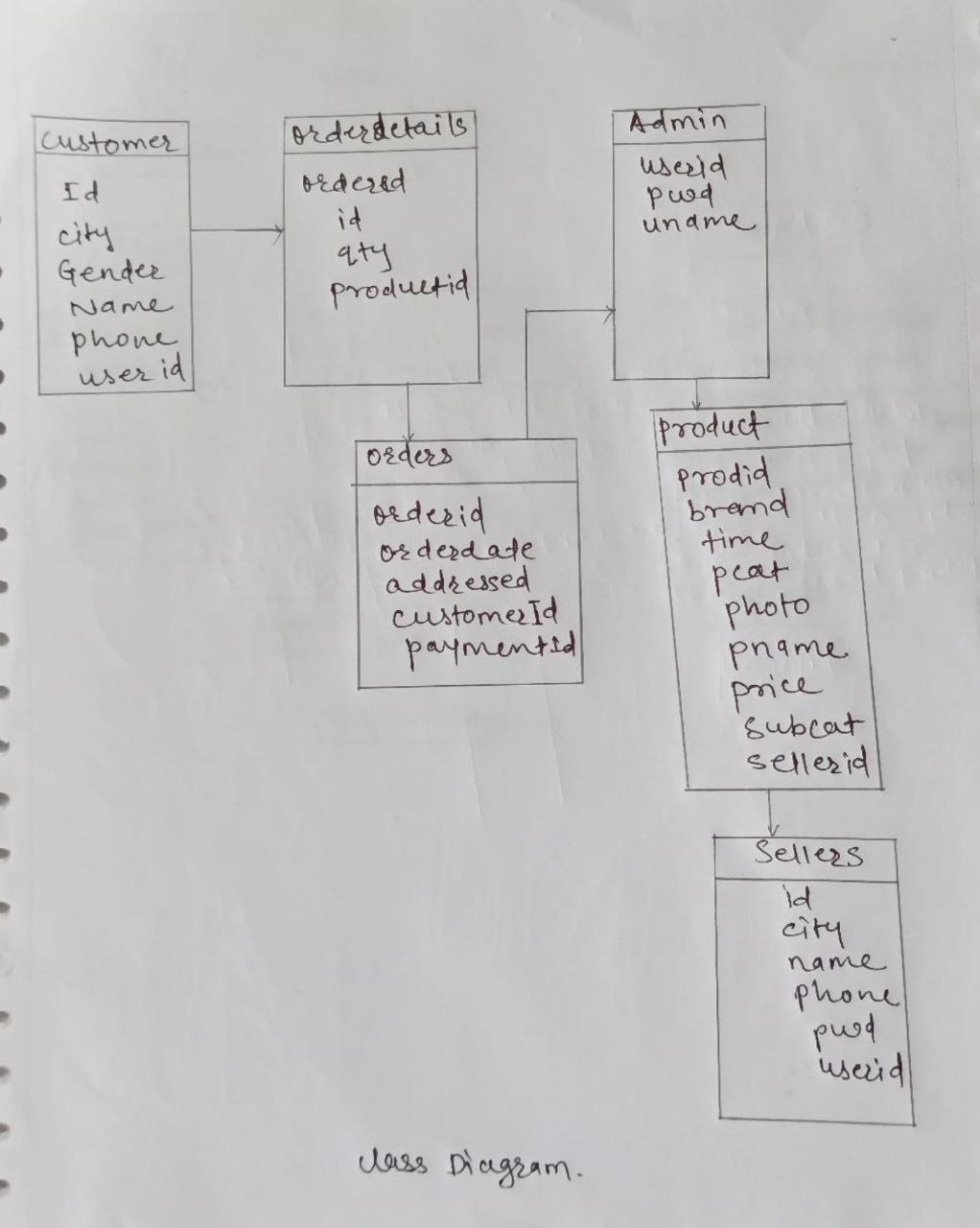


Figure 8: Class Diagram

1. **End to End Flow of Application:**

**User:**

* 1. User will login to the portal or will have to register if he is not a registered user.
  2. After registration User will login and Dashboard page will be displayed to him which will display products which are available to buy and option to get customize product.
  3. From that page User can click on the product to buy and proceed for checkout.
  4. User can also select the category for direct search of required product.
  5. User is also able to give feedback of the product which he had received.
  6. User will also able to request for customized product and also can see his cart. Only Admin is able to see users cart.

**Admin:**

1. Admin will login as Admin from the ‘**Admin login**’ page and will be able to see products ordered and requested for customization by the Users of a particular area.
2. Admin will notify to the shops for the particular product.
3. It is the job of Admin to assign appropriate contractor or service person who will deliver the requested product to user.
4. After products delivered admin will change the status of cart of users and check for the feedback and make changes for better service.

**Shop:**

1. Shopper will login to the portal or will have to register if he is not a registered user.
2. Shopper is able to add products according to categories provided.
3. Shopper will also able to edit added products and also have access to remove the products.
4. Shops will also able to see all request received from admin for different products and he will also able to add availability status of the products.
5. Future Scope of Project:
6. Verification of users.
7. Improvement in design and service
8. Mobile Application
9. UPI Payment
10. Rent equipment according to user need.

**Thank You!**