**Project Name: ToysHouse**

**Project Member:**

**Sandesh Javheri 220343120041**

**Sonal Andge 220343120010**

**Prasad Bansode 220343120082**

**Onkar Shete 220343120100**

**Abstract:**

From thousand of years humans are doing buying and selling the needy things, in old era buying and selling mechanism works on the physical mode where user must go on the shop to buy any essential things, but in 21st century where we are living in modern era human discovered internet and after that world gets connected digitally.So human discovered a new concept known as E-Commerce, where user can buy and sell their things on the internet. So most of the website sell users a new product, but then what about used product ,so the concept of refurbished thing comes. So for the families who are not capable to buy new product or if user wants to buy used product in low price they can buy the product from refurbished website. So we will devlope the website ToysHouse. By using this website needy user can buy the refurbished toys in affordable price for their children. Also if any user wants to sell their childs old toy then they can also sell their product on this website. There will be transparency in the system as there is no third party gets involved. So this website is focused on children whose familes are not capable of buying new things. Also project has capability to add new features in it.

**Implementation Technologies:**

1. **SpringBoot:**
2. Spring Boot is a project that is built on the top of the Spring Framework. It provides an easier and faster way to set up, configure, and run both simple and web-based applications.
3. It is a Spring module that provides the **RAD (Rapid Application Development)** feature to the Spring Framework. It is used to create a stand-alone Spring-based application that you can just run because it needs minimal Spring configuration.
   1. **Features of SpringBoot :**

## **1.Web Development**

It is well suited Spring module for web application development. We can easily create a self-contained HTTP server using embedded Tomcat, Jetty or Undertow. We can use the spring-boot- starter-web module to start and running application quickly.

## **2. Admin Support**

Spring Boot provides the facility to enable admin related features for the application. It is used to access and manage application remotely. We can enable it by simply using spring.application.admin.enabled property.

## **3. Externalized Configuration**

Spring Boot allows us to externalize our configuration so that we can work with the same application in different environments. Application use YAML files to externalize configuration..

## **4**. **Type-safe Configuration**

Strong type-safe configuration is provided to govern and validate the configuration of application. Application configuration is always a crucial task which should be type-safe. We can also use annotation provided by this library.

## **5**. **YAML Support**

It provides convenient way for specifying hierarchical configuration. It is a superset of JSON. The SpringApplication class automatically support YAML. It is successful alternative of properties.

## **6**. **Security**

Spring Boot applications are spring bases web applications. So, it is secure by default with basic authentication on all HTTP endpoints. A rich set of Endpoints are available for develop a secure Spring Boot application.

**2.1** **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).

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

**Hardware:**

1. Intel i3 processor 10th generation or later / AMD Ryzen 200 2nd generation or later

2. 4 GB ddr3 ram.

3. Windows 10 Home edition or later.

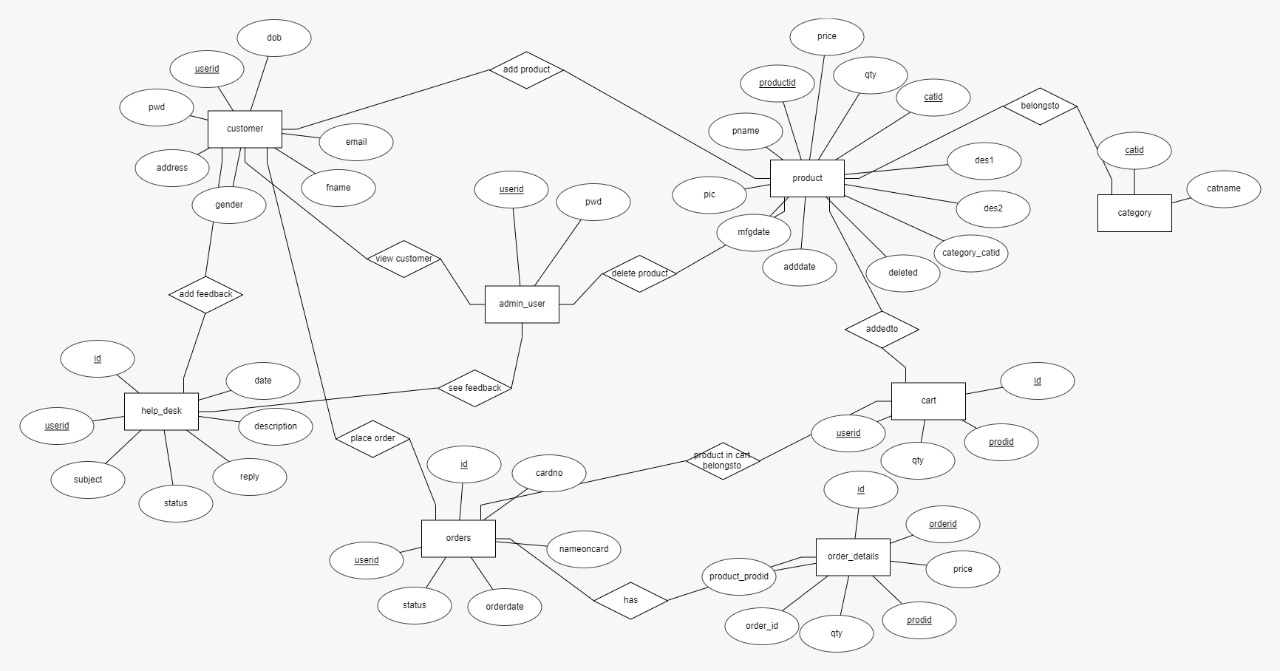
4. 200 GB Sata HDD Space

5. Data Connection 1mbps

**Software:**

1. Eclipse Oxygen
2. MySQL 5.7 with Workbench 8.0
3. Google Chrome version 105.0
4. Maven Dependencies
5. **ER Diagram:**

Figure 1: ER Diagram

****

**5.Table Structures:**

1. **Table name:Admin**

+--------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------+--------------+------+-----+---------+-------+

| userid | varchar(255) | NO | PRI| NULL | |

| dob | varchar(255) | YES | | NULL | |

| email | varchar(255) | YES | | NULL | |

| fname | varchar(255) | YES | | NULL | |

| gender | varchar(255) | YES | | NULL | |

| lname | varchar(255) | YES | | NULL | |

| pwd | varchar(255) | YES | | NULL | |

+--------+--------------+------+-----+---------+-------+

1. **Table name:Customer**

+--------+--------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+--------+--------------+------+-----+---------+-------+

| userid | varchar(255) | NO | PRI | NULL | |

| dob | varchar(255) | YES | | NULL | |

| email | varchar(255) | YES | | NULL | |

| fname | varchar(255) | YES | | NULL | |

| gender| varchar(255) | YES | | NULL | |

| lname | varchar(255) | YES | | NULL | |

| pwd | varchar(255) | YES | | NULL | |

+--------+--------------+------+-----+---------+-------+

1. **Table name:Product**

+----------------+--------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+----------------+--------------+------+-----+---------+----------------+

| prodid | int | NO | PRI | NULL | auto\_increment |

| adddate | varchar(255) | YES | | NULL | |

| catid | int | NO | MUL | NULL | |

| company | varchar(255) | YES | | NULL | |

| deleted | bit(1) | NO | | NULL | |

| mfgdate | varchar(255) | YES | | NULL | |

| pic | varchar(255) | YES | | NULL | |

| pname | varchar(255) | YES | | NULL | |

| price | float | NO | | NULL | |

| qty | int | NO | | NULL | |

| salt | varchar(255) | YES | | NULL | |

| category\_catid | int | YES | MUL | NULL | |

+----------------+--------------+------+-----+---------+----------------+

**4.Table Name:Category**

+---------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+---------+-------------+------+-----+---------+-------+

| catid | int | NO | PRI | NULL | |

| catname | varchar(50) | YES | | NULL | |

+---------+-------------+------+-----+---------+-------+

**5.Table Name:Cart**

+--------+--------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+--------+--------------+------+-----+---------+----------------+

| id | int | NO | PRI | NULL | auto\_increment |

| prodid | int | NO | MUL | NULL | |

| qty | int | NO | | NULL | |

| userid | varchar(255) | YES | MUL | NULL | |

+--------+--------------+------+-----+---------+----------------+

**6.Table Name:Orders**

+------------+--------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+------------+--------------+------+-----+---------+----------------+

| id | int | NO | PRI | NULL | auto\_increment |

| cardno | varchar(255) | YES | | NULL | |

| nameoncard | varchar(255) | YES | | NULL | |

| orderdate | datetime(6) | YES | | NULL | |

| status | varchar(255) | YES | | NULL | |

| userid | varchar(255) | YES | MUL | NULL | |

+------------+--------------+------+-----+---------+----------------+

**7.Table Name:Order\_Details**

+----------------+-------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+----------------+-------+------+-----+---------+----------------+

| id | int | NO | PRI | NULL | auto\_increment |

| orderid | int | NO | MUL | NULL | |

| price | float | NO | | NULL | |

| prodid | int | NO | | NULL | |

| qty | int | NO | | NULL | |

| order\_id | int | YES | MUL | NULL | |

| product\_prodid | int | YES | MUL | NULL | |

+----------------+-------+------+-----+---------+----------------+

**8.Table Name:Help\_desk**

+-------------+--------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-------------+--------------+------+-----+---------+----------------+

| id | int | NO | PRI | NULL | auto\_increment |

| date | datetime(6) | YES | | NULL | |

| description | varchar(255) | YES | | NULL | |

| reply | varchar(255) | YES | | NULL | |

| status | varchar(255) | YES | | NULL | |

| subject | varchar(255) | YES | | NULL | |

| userid | varchar(255) | YES | MUL| NULL | |

+-------------+--------------+------+-----+---------+----------------+

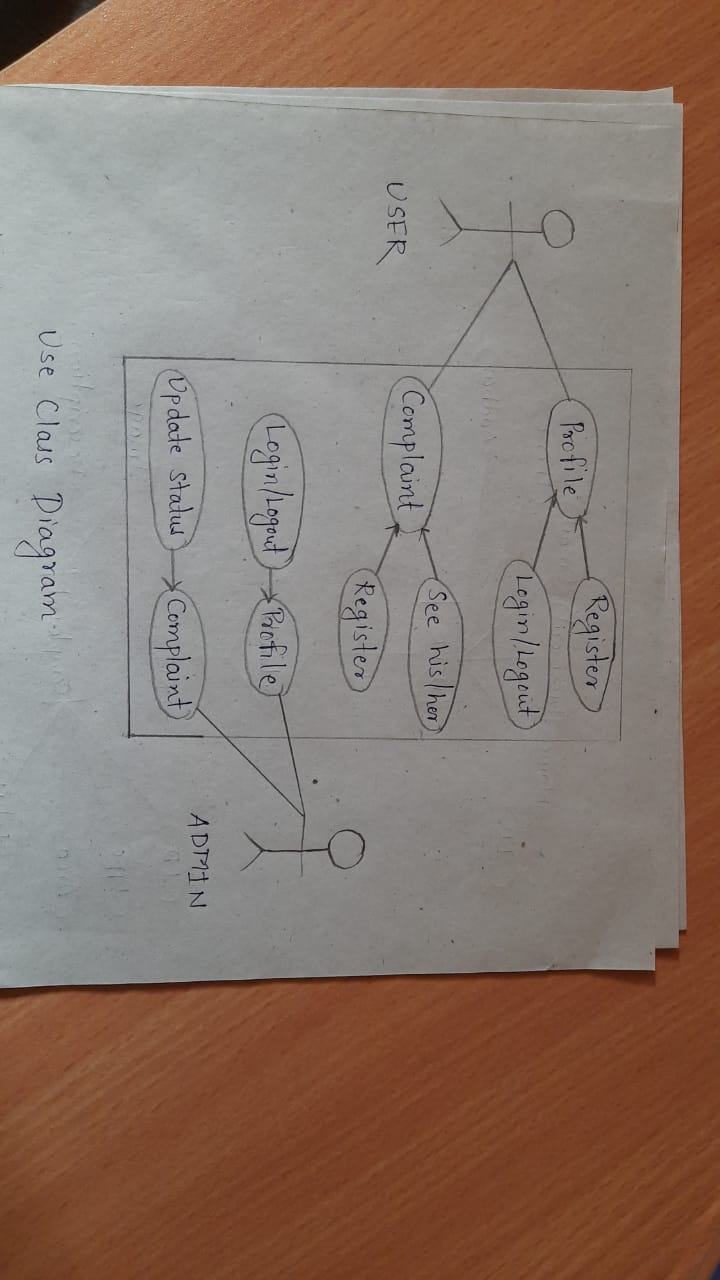
**6.UML Diagrams:**

Figure 3: Use Case

Figure Use Case

Figure 4: Collaboration Diagram

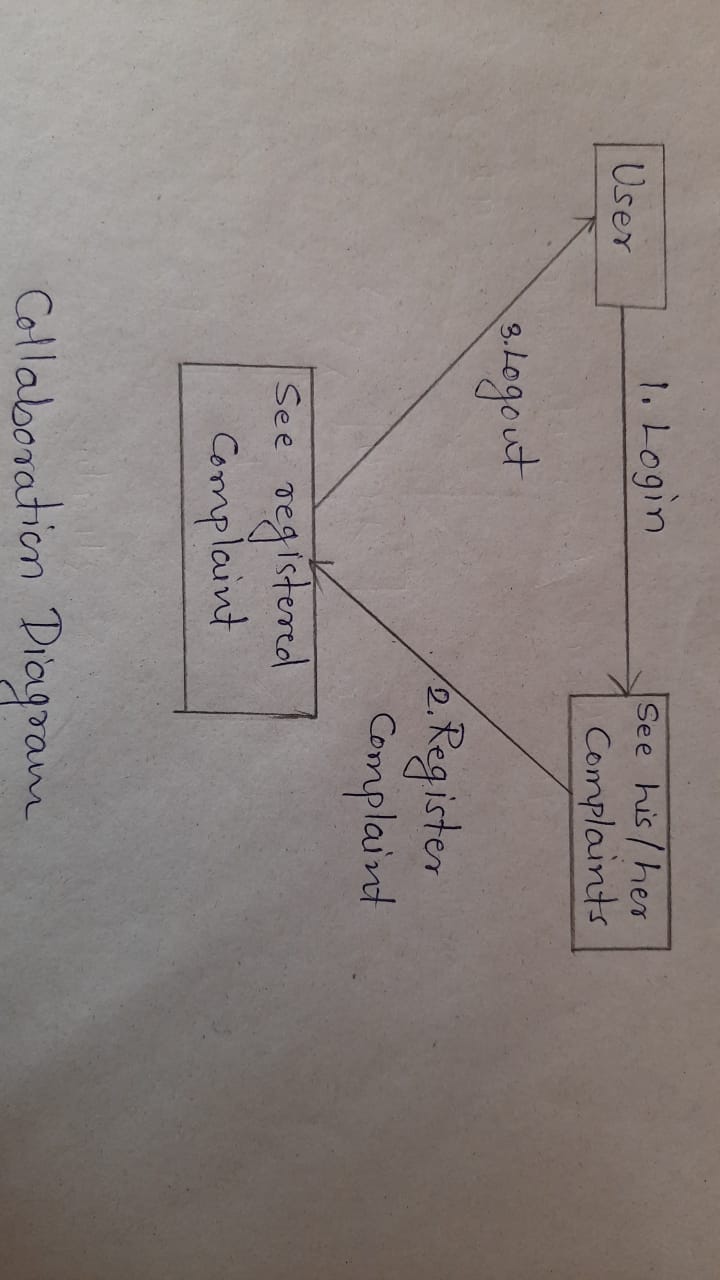


Figure Collabration Diagram

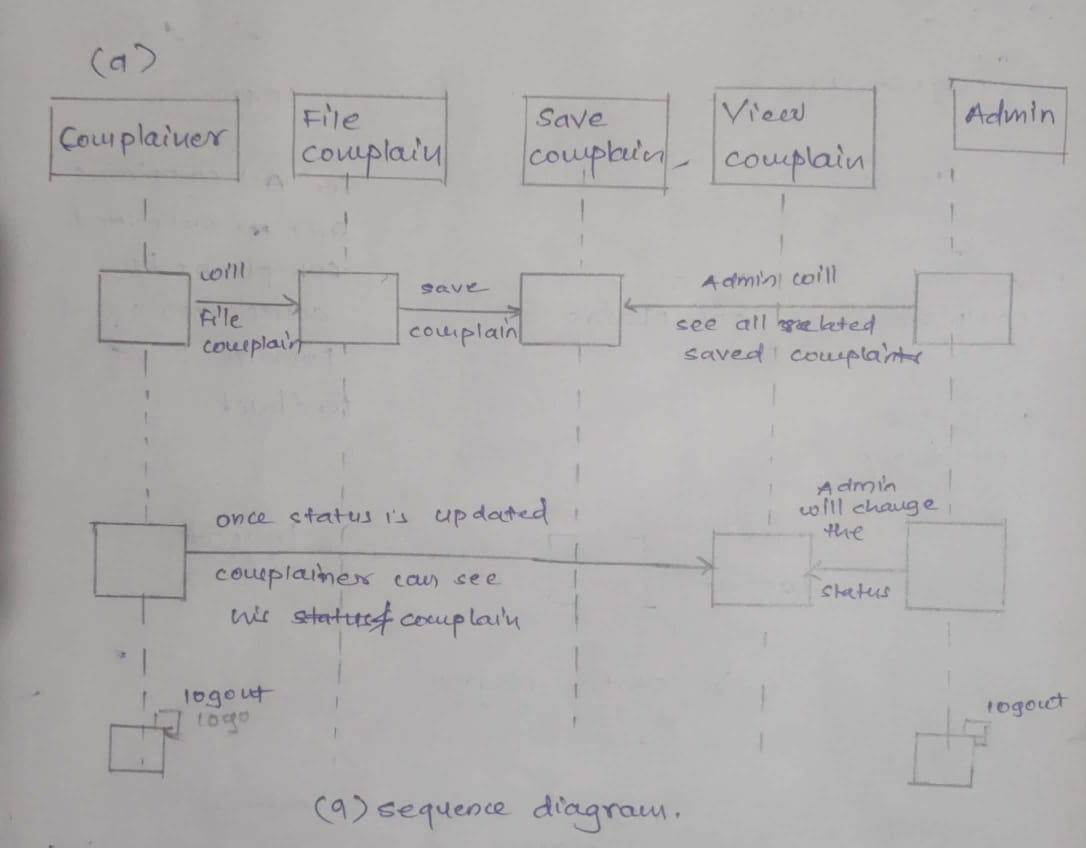


Figure 6: Sequence Diagram

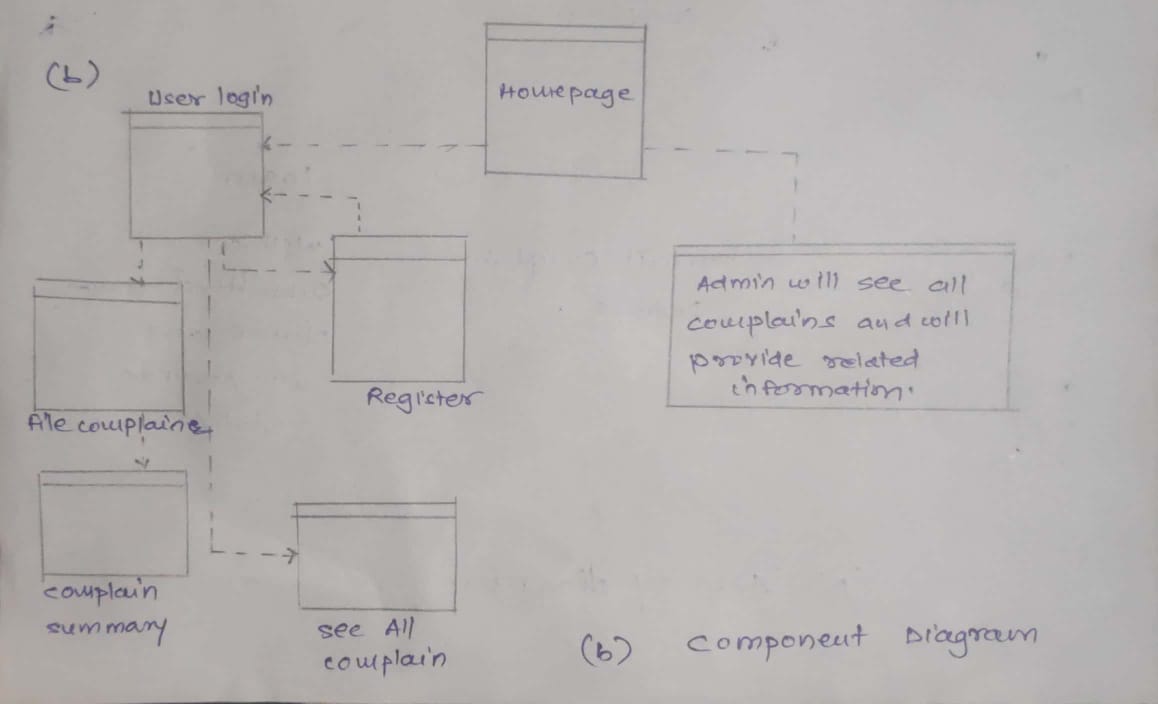


Figure 7: Component Diagram

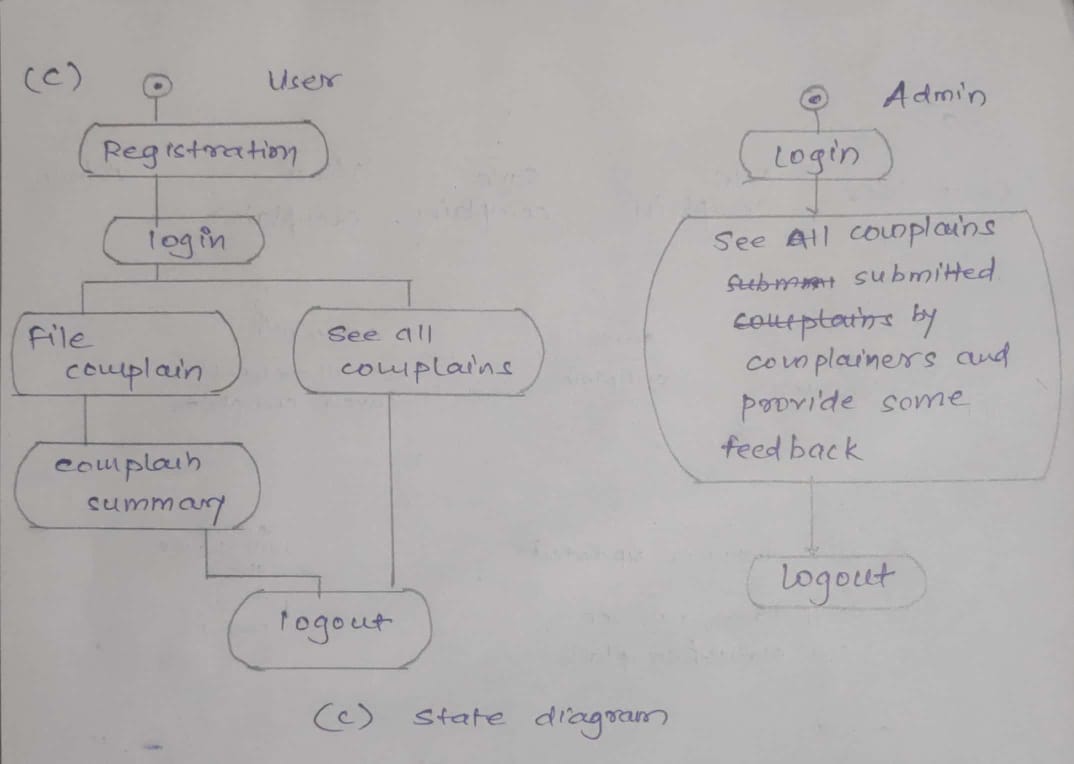


Figure 8: State Diagram

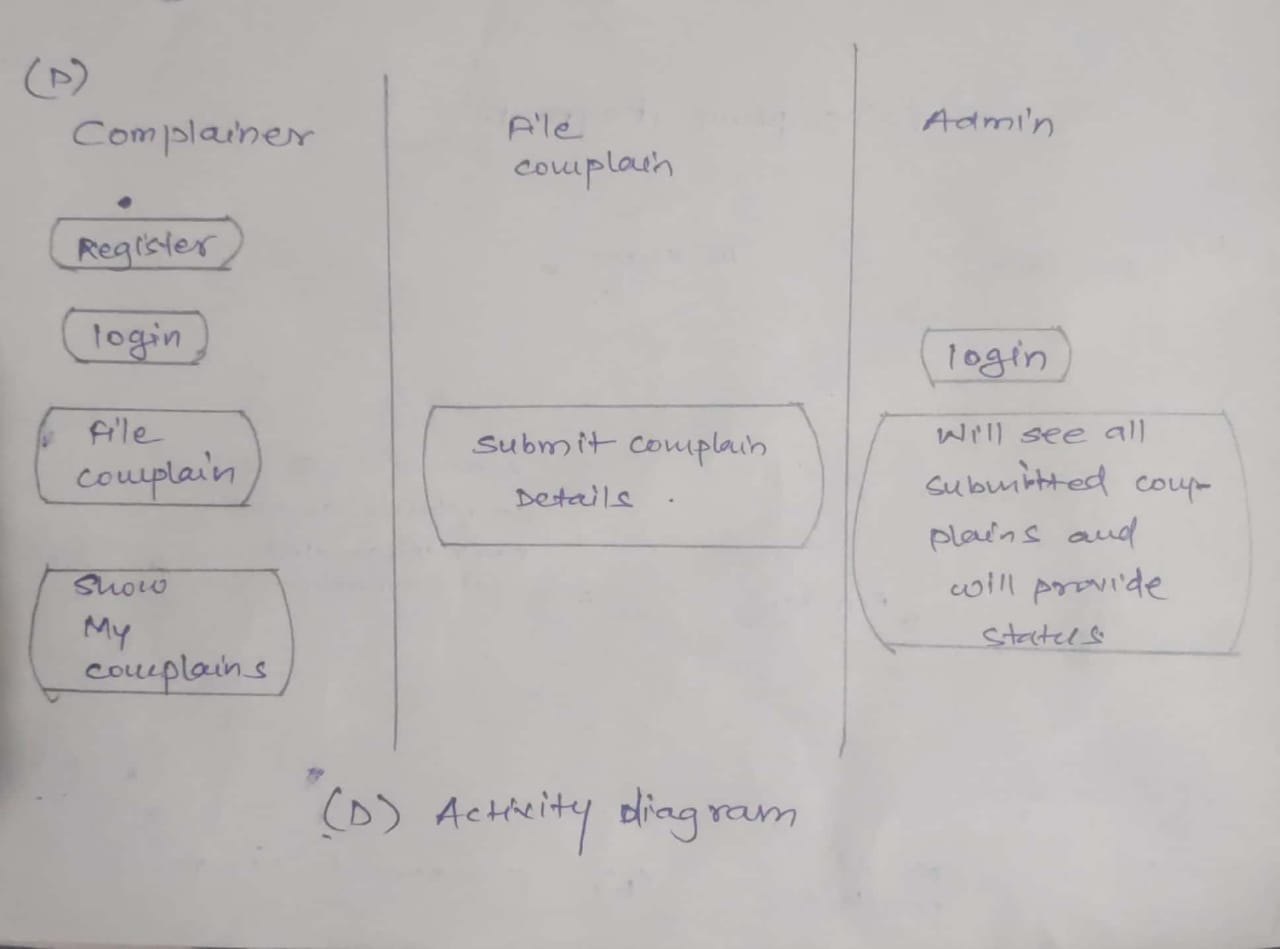


Figure 9: Activity Diagram

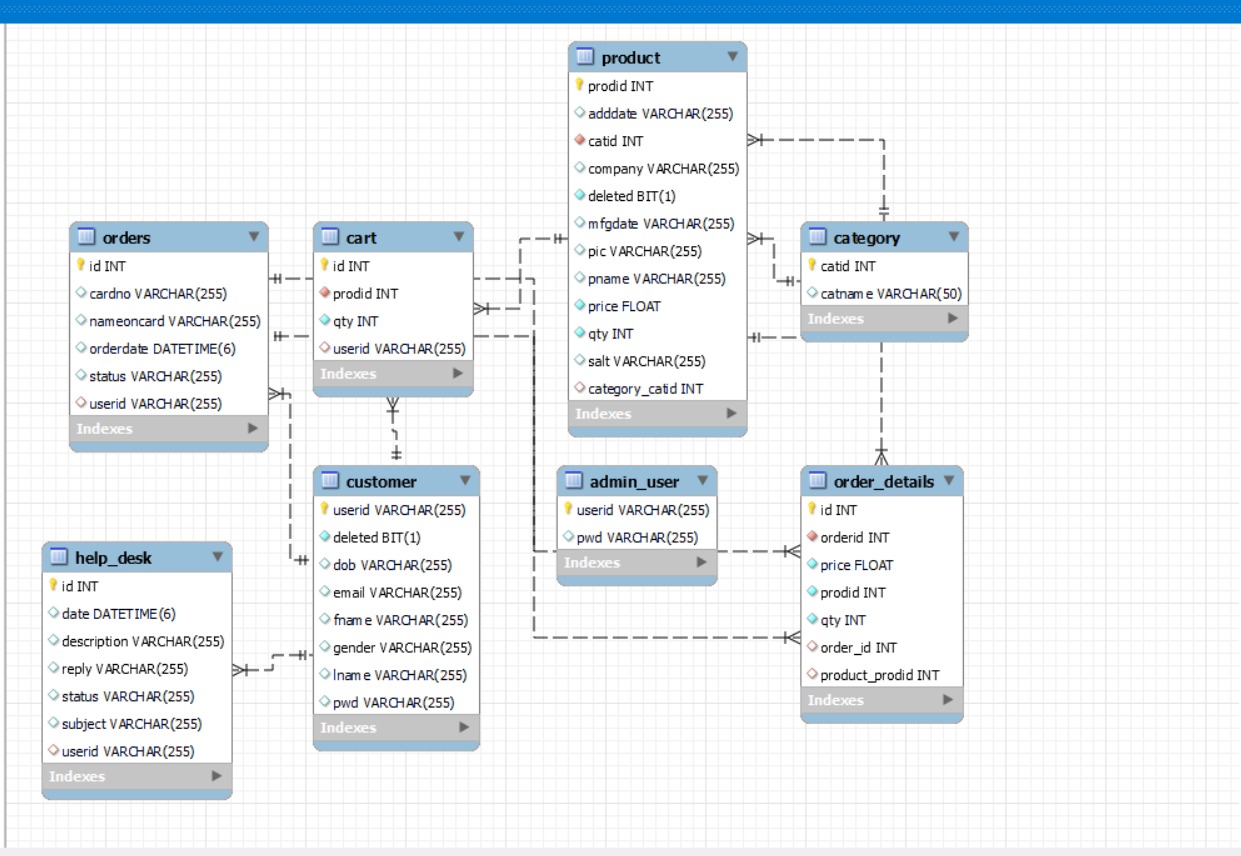


Figure 10: Class Diagram

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 the previous complains and its status if any.
  3. From that page can User can click on the ‘**file new complain’** button and reach the complaint details form page.
  4. In the complaint details page the User has to pick a category among the **Eight**predefinedcategories and brief about the problem with affected area (address) and image of the object or place.
  5. A ‘**summary report’** will be displayed on the Website showing all the details of the complaint.
  6. User will only be able to see his complaint after the complaint has been ‘**Received’**or either ‘**Resolved’** by the respective admin of the category chosen.

**Admin:**

1. Admin will login as Admin from the ‘**Admin login**’ page and will be able to see his share of Complains filed by the Users of a particular area.
2. Admin can Review the complaint and after understanding it Admin will ‘**Receive’** the complaint.
3. It is the job of Admin to assign appropriate contractor or service person to resolve the matter at the hand as soon as possible to avoid disturbance among the public.
4. After conforming about the completion/resolving of the problem, Admin will check the status of the problem as ‘**Resolved’**and head over to the next complaint if any.
5. **Future Scope of Project**
6. Verification of users.
7. Improvement in design.
8. Mobile Application
9. Integration of new function like charity events for poor children.
10. UPI Payment
11. Chat BOT
12. Rating System

**Thank You!**