POS System (Point of sale System)

Team Members:

Piyush Kant

Mihir Gorasiya

Dhruv Patel

Naman Sharma

Motivation:

Our aim was to develop a user friendly and simple system in java, using object-oriented programming (OOP) concepts. the main motivation is to develop seamless user experience system with bill generation capabilities for business. We are adding some user validation like if user is cashier or manager and on the base of that user have different privileges and we added payment simulation just like real money payment. User will enter the amount of receipt and program will tell how much change needs to return to the customer.

Description of Design and implementation of the system:

This project is console based – menu driven application. On the launch of the application, a console will open, and, on every step, there will be a menu of options of action, from which users can select their action to perform tasks in application. For example, on register page, there should be two options sign up and login, application will show menu ”Enter 1 to sign up, Enter 2 to login”.

<u>Implementation</u>

To implement POS system, we have created different classes for different components like product, employee, receipt, and payment and added different functionality in different child classes like cashier, manager for employee and

card payment or cash payment for payment. We have stored all data in non-primitive iterable data types instead of database because this is very small application. We used different OOP concept to reuse code and reduce repeat code.

Data model:

To store data in this application, we used non-primitive iterable data types instead of database. Because this application is small, and it has less data available to store.

To develop this application, we need to store data about users (Manager and Cashier) and all product data (product id, product name, category, price). For this data types like array were better option for us.

System Component and Function:

Activity sequence:

At the beginning of the application, user will see the menu with options like clock in, clock out, get current employee, punch product and show all receipt but to get current employee, punch product user needs to clock in.

To clock in user will press 1 in first menu, and user will have option to enter employee id. If employee exist with specific employee id, user will login, or it will show user not exists.

After clock in user can perform any actions.

To clock out, user just need select clock out from menu and program will clock out user.

To punch product, user will select punch product option from menu.
In this state, application will show list of products available with price and category. Then user can enter product id to add in cart. If user punch any id that does not exists, punch product will end, and application will show receipt and payment method selection menu.

Outstanding Issues:

We faced issue before starting the project, to select the topic which can cover all topic of object-oriented programming. Then we also were confused where to start the project and how to design it. At the end, we also faced issues in integrate interface and polymorphism to migrate with functionality.

Result of a system evaluation:

The current system almost the same as we imagined at the beginning. There are some minor features are missing from the project, which is modify available items to punch (add or remove item), add discount for some products (multiple item purchase discount), membership card discount, etc. other than that, our system is same as our initial goal, and we can expand this project in the future by adding new functionalities.

What have we learned?

We learned many things during the project. First of all, we learned about planning, how to organize different task and set the priority to different tasks. We also learned many things related to programming. We revised all OOP concepts and learned how to merge all concept to gather in one project. We also learned version control system GitHub. How to manage code on GitHub, resolve

conflict, create pull request, merge pull request and manage code from different branches. Debug code was one of the most important topic we learned during the project.