The entry point for the program in main.java is the Main class, which initializes a loop that allows users to navigate the system. Here they may choose to enter as an admin (who requires the password “qwer123” to proceed) or a customer (who needs to enter their current balance to proceed). Private access modifiers are used in every class to protect data and provide controlled access via getters and setters - for example to set a password - using encapsulation.

Admin Functionalities: A **TreeMap** is used to store the menu items sorted by price with the price as the key and the item as the value (Item is a class with name, price and category required to initialize) so that it easy to view items by sorted price. A two-level **PriorityQueue** is used to store orders, with VIP orders getting precedence so the orders are processed correctly. Variables sales and orders are initialized to 0 to keep track of the amount of money made and number of orders placed and incremented as sales/orders are placed. Everything in Admin class is declared static as there is only one Admin and the data in this class remains constant. There are functions to add, remove, view and update items to the menu, which traverse through the TreeMap values and match with item names to locate each item. There is an Order class which contains an **ArrayList** of items. There is a function to view all pending orders by traversing through the allVIP queue first and then the allNorm queue which displays requests if present. Order status can be updated by matching with the order id. Refunds are managed by calling the refund() function in the Customer class, which checks if the order is canceled, refunds the balance, removes the order from the customer’s purchase history and updates the status to refunded. A sales report with all past orders and the corresponding stats can be printed as well.

Customer Functionalities: A single dummy customer is initialized to demonstrate the program with ease. Customers can view the menu by ascending or descending order (by using the descendingMap() function) of price similar to the Admin as the Items are sorted automatically in a TreeSet. The search and filter functions work similarly by traversing through the menu and comparing with the respective parameters. They can also make requests, add, remove and edit the quantity of items using functions that alter the Order class. They can become VIPs by paying 500. The total cost of the order and its status can be viewed using a simple getter. The checkout() function in the Customer class checks if the balance is more than the total and deducts money accordingly (printing an error message otherwise), clears the cart and adds the order to the customer’s purchase history (which is an ArrayList of orders and can be viewed by them as well). The customer can also cancel the order, which updates its status to 3. Customers can also provide and view reviews item wise, which are stored as an ArrayList for each item.

Note while running: There is an infinite loop so you can run the portal as many times as you want as whichever user you want, and the data is saved each time. So if for example you want to remove an item from the menu, you need to run an iteration of the portal before and add it so it is removable.