# Introduction

## Purpose of the System

Café Mocha App is a unique software solution designed to enhance operations at the coffee shop called Café Mocha. It is a compact and vibrant hub of activity that offers a variety of coffee types, a bakery, and soups. With its expected growth, the app becomes all the more vital for more efficient order-taking, billing, and menu-updating processes. The Café Mocha Application meets these needs by offering an inclusive digital platform that replaces the traditional manual processes, hence making everything smooth for the staff and the customers as well.

Among the major features are user authentication to allow only valid personnel to enter the system and manage orders and update menu items. It supports the addition, modification, and removal of orders with little hustle, besides the generation of bills automatically depending on the order items. It also has a help feature that drives it, guiding the user through the system to make the application very friendly for all staff. Through the implementation of the Café Mocha App, this coffee shop aims to ease up the running of the establishment, reduce errors, and enhance the quality of its service so that employees and customers can have a better time.

## 1Updateject Objectives

The creation of a standardized software program for Café order and menu administration is the main goal of the Café Mocha App. It's a system that lets you join in, add new orders, change orders, delete orders, and work with menu items. The system creates bills based on the orders placed in order to maintain accurate financial records.

## 1.3 Scope and Limitations

The scope of the project includes user authentication and user login management. Create an order, edit an order, and delete orders. Menu Management: Make a new menu and display the existing menu. Bill generation is developed on the basis of customer orders. Limitations: The system uses text files to save the data, which is practically not efficient when data scale increases as much as a DBMS can handle. The model is designed for single-user access at a time.

# System Design and Architecture

## Overview of System Architecture

The Café Mocha App is designed to streamline Café operations by integrating several key modules:The Café Mocha App is designed to streamline Café operations by integrating several key modules:

1. User Authentication: Offers a safe solution that offers validation of username and password.

2. Order Management: Responsible for creating as well as editing orders and deleting them and taking records of all the orders in a text file.

3. Menu Management: Enables easy change and frequent display of menu that keeps the menu alive and up to date.

4. Billing Module: Calculates the bills on the basis of orders prepared by the computer and does so in a very efficient manner.

5. Help Module: It offers instructions to the users; therefore, the app is simple for each member of the staff to navigate.

6. Data Storage: Sustain text files for storage of the user information and the orders in addition to the billing purposes in order to ensure that information is well arranged and easily retrievable.

This describes an architectural structure that is much more efficient, free of any break, and secure for the operation of the Café.

## 2.2 Flow Chart

**Following is the flow chart for the main functions of the system:**

* User Authentication: Verifying the credentials of the user.
* Main Menu: Options available to the authenticated user.
* Order Management: Add, modify, or remove orders
* Menu Management: Add new menu items and display the current menu.
* Bill Generation: Calculate and display bills based on orders.
* Exit: Exit the application.

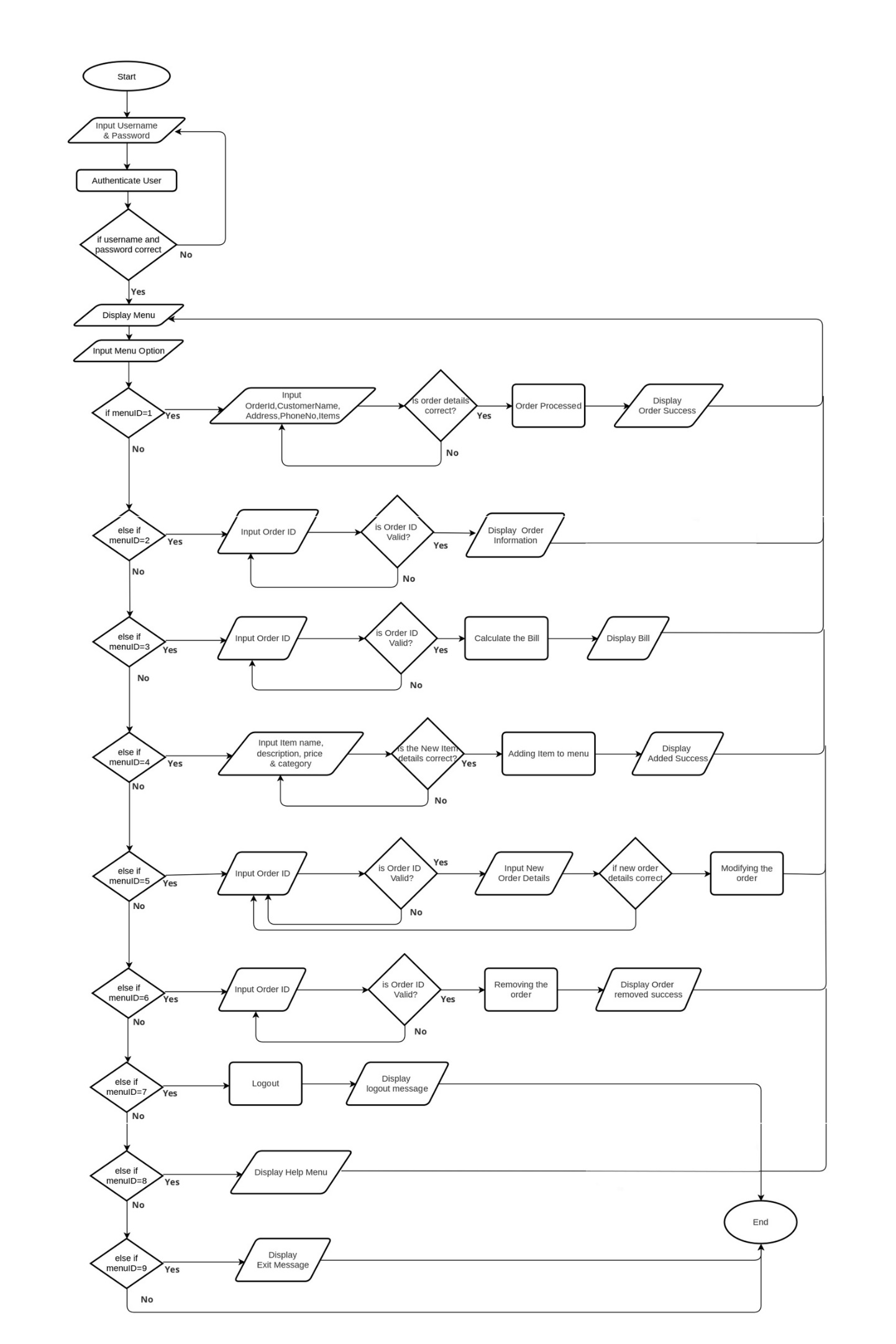


Figure 1 Flow Chart

Figure 2 Flow Chart

# Functional requirements analysis

* User Authentication and Security: Users should provide a valid username and password available in the user.txt file. The information entered should match; otherwise, the app prompts for re-entry of inputs.
* Order Management System: Users can place orders to be written at the end of the orders.txt file. Already existing orders should be modifiable to edit or delete further customer request tracking.
* Menu Management System : The system shall be able to add new items to the menu, which are created and updated in the menu.txt file. The present menu is shown to the user with all the details: name, description, price, category.
* Billing System : Generate a bill on an itemized basis by considering the items ordered along with their quantity, fetching prices from the menu.txt file. Create bills that are saved for future reference inside bills.txt.
* User Interface Requirements: The system must articulate the interface to the users through prompts and options for various actions. Help/Error messages pop out to guide the user in the different functionalities.

# Non-Functional Requirements

* Performance Requirements: The system has to be able to perform numerous read and write actions to the text files. The time taken to display each menu, the time taken to process orders, and the time taken to generate bills should be as short as possible to make the user experience pleasant.
* Security Requirements : Users’ identification details are to be only saved and can be only read by the application. Such messages should not contain information that can be either exploited to the disadvantage of the user or organization.
* Usability Requirements : The elements of the interface must be easily understandable, and it must either explain to the player what to do or give feedback. Handling of invalid inputs should be well implemented, giving hints to the users regarding the right information to input.
* Reliability and Maintenance : It should also be fast because users do not want to spend hours waiting for the system to load or wanting for the system to be up again after crashing. The application function and design must be easy to maintain as it is modular, so it will be easy to change the food menu or rectify some coding problems.

# 5. Software Development and Implementation

## 5. 1 Programming Language and Tools Used

The Café Mocha App is developed in Java. The application is purposely developed to be used in a console-based environment to ensure that it is easy to use.

## 5. 2 Implementation of Core Features

### 5. 2. 1 User Authentication Functionality

Method: *authenticateUser()*

Description: This method requires the user to enter the username and password as a means of identification. Afterwards, the program will check the input credentials against a list of stored credentials in a file known as user.txt. Once the two credentials coincide, the user is provided with the opportunity to access the given system. This functionality also makes it possible for only the authorized users to have permission to make alteration, or view any information within the application data.

A computer screen shot of code

Description automatically generated

Figure 3 Authenticate User

.

### 5.2.2 Order Processing and Management

Method: *createNewOrder(), modifyOrder(), removeOrder()*

Description:

*createNewOrder*(): This function is meant to keep records about a certain customer’s order comprising of the customer information and the particular items ordered. Each new order is assigned a unique order ID so that it can be tracked and managed individually. When all the information is received, this function writes into `orders.txt` all necessary details for future references or auditing purposes as a permanent transaction record.  
  
*modifyOrder*(): This function allows users to change details of an existing order. It first reads through all current orders from the file called `orders.txt’ for old information retrieval purposes. After picking an order you want to change, this function lets you modify relevant details like altering items or quantities. After all modifications are done, new order details are saved back in orders by being written into ‘orders.txt’, ensuring that this data reflects exactly what is there today.  
  
*removeOrder*(): This function gives one power to delete any given order in the system. It reads existing data about such orders with respect to what file ‘orders.txt’ contains before letting him know which one should be deleted according to the unique order ID assigned therein. The selected command deletes that particular one upon making a confirmatory head and updates the list in the file consequently.

A computer screen shot of a program code

Description automatically generated

Figure 4Add New Order

A computer screen shot of text

Description automatically generatedA computer screen shot of a program code

Description automatically generated

Figure 5Modify Order

Figure 6 Remove Order

### 5.2.3 Add/Modify Menu Item

Approach: *addMenuItem*()

Description: This form will allow the staff (admin) to add new items to the menu. Each menu item will be described by name, description, price, and category. The information of a new menu item will be logged in menu.txt.

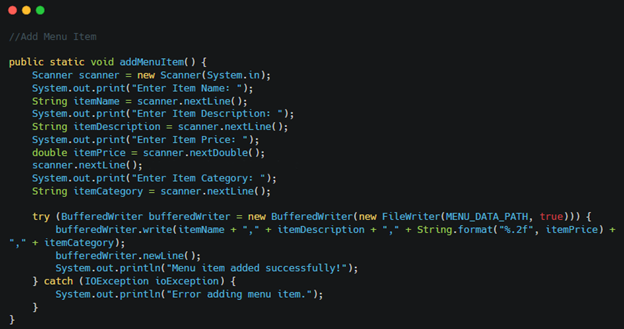


Figure 7 Add Menu Item

5.2.4 Bill Calculation and File Storing

Method: *generateBill*()

Description: This program calculates the total cost of an order regarding items and their amount, reads the item prices from menu.txt, and saves the generated bill details in bills.txt—just for keeping a record.



Figure 8 Generate Bill

## Use of Arrays and Structures

Arrays are applicable in the handling of data that involves menu items and the information of orders to give a simple and efficient method of handling data within the application. To this end, an array is applicable in storing the items that are put in an order to ensure easy management of such information.

## 5.4 Control Structures & Decision Making

There are several control structures used throughout the application, like loops (for, while) and decision-making constructs (if, switch-case) to manage user inputs and the flow of the program. These structures help to keep the user experience optimal by introducing him to the different menus and the different systems that are in the menu.

## 5. 5 File Handling Techniques

File handling can be considered one of the major sub-processes within the framework of the Café Mocha App. To read from or write into files, the application employs `BufferedReader` and `BufferedWriter` classes, respectively. This enables the users’ data and orders to be stored and easily retrieved when the application is opened; lists of menu items and bills are also stored for further use.

Figure 9 File Handling

# 6. Testing and Validation

## 6. 1 Testing Strategy and Methodology

The testing strategy here involves the coming up of the test cases for the various functionalities of the system. This approach involves testing of a program’s correctness by a tester who thinks through different test cases through different input cases and the manner in which the user interacts with the program.

## 6. 2 Test Plan Overview

The test plan lists several scenarios that include almost all aspects of the system functionalities, such as user authentication, order, menu, and bill. Each of the scenarios is intended for the input of proper as well as improper formats that will check the ability of the system to handle errors.

Test Plan

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case ID | Test Case Name | Description | Expected Result |
| TC1 | User Authentication | Test the login with correct credentials. | Displays “Login Successful !” and shows the Main Menu. |
| TC2 | Add New Order | Test adding a new order with valid details | Order Successfully added and message displayed |
| TC3 | Show Order Information | Test displaying order information with order ID | Displays order information that is connected to order ID |
| TC4 | Generate Bill | Test generating a bill for given order ID | Displays the correct total costs, and the bill is recorded. |
| TC5 | Add Menu Item | Test adding new items to the menu with the details | Displays item successfully added to menu |
| TC6 | Modify Order | Test modifying and existing order’s details | The order was successfully modified. |
| TC7 | Remove Order | Test removing orders from the system. | The order is successfully removed. |
| TC8 | Logout | Test Logout from the system | Displays Successfully Logout message and return to login |
| TC9 | Help Menu | Test displaying the help menu | Displayed help menu successfully |
| TC10 | Exit | Test existing the application | Exit successfully from the application. |

## 6.3 Test Case

|  |  |
| --- | --- |
| Test case ID | TC1 |
| Test case name | Check Login |
| Scenario | Test the login with correct credentials. |
| Test Data | Username: “admin”  Password: “password” |
| Expected Result | Displays “Login Successful!” and shows the Main Menu. |
| Actual Result | Same as expected |
| Screenshot |  |
| Conclusion | If the username and password are correct, login will be done. If they are incorrect, it displays an invalid message. |

|  |  |
| --- | --- |
| Test case ID | TC2 |
| Test case name | Check Add New Order |
| Scenario | Test adding a new order with valid details |
| Test Data | 1, John Doe, Address Road Country, 0702906039. Tea:2 |
| Expected Result | Order Successfully added and message displayed |
| Actual Result | Order Added and returned to the Main Menu |
| Screenshot |  |
| Conclusion | The order will be added to the system; if there are any incorrect details, the process will stop and show an error message. |

|  |  |
| --- | --- |
| Test case ID | TC3 |
| Test case name | Check Show Order Information |
| Scenario | Test displaying order information |
| Test Data | - |
| Expected Result | Displays the order information |
| Actual Result | Same as expected |
| Screenshot |  |
| Conclusion | Show all orders that are presented in the system. |

|  |  |
| --- | --- |
| Test case ID | TC4 |
| Test case name | Check Generate Bill |
| Scenario | Test generating a bill for given order ID |
| Test Data | 1 |
| Expected Result | Displays the order bill according to the order ID. |
| Actual Result | Same as expected |
| Screenshot |  |
| Conclusion | Generates the order total if the order id is matching with the existing order; otherwise, it shows the error message. |

|  |  |
| --- | --- |
| Test case ID | TC5 |
| Test case name | Check Add Menu Item |
| Scenario | Test adding new items to the menu with the details |
| Test Data | Donut, Sweety Dweety Donuts, 6.00, Sweets |
| Expected Result | Displays Item added successfully and added to the menu |
| Actual Result | Displays item added successfully |
| Screenshot |  |
| Conclusion | A new item will be added to the menu if there are any incorrect details. error message will display |

|  |  |
| --- | --- |
| Test case ID | TC6 |
| Test case name | Check Modify Order |
| Scenario | Test modifying and existing order’s details |
| Test Data | Tea:2;Pizza:1 |
| Expected Result | Displays item modified successfully |
| Actual Result | Same as expected |
| Screenshot |  |
| Conclusion | The order will modify after adding a new item to the order; if it's not a valid item, the error will display. |

|  |  |
| --- | --- |
| Test case ID | TC7 |
| Test case name | Check Remove Order |
| Scenario | Test removing orders from the system. |
| Test Data | 1 |
| Expected Result | Displays Item deleted successfully |
| Actual Result | Same as expected |
| Screenshot |  |
| Conclusion | Order will remove and show the removing message. If order ID is invalid, an error message will display. |

|  |  |
| --- | --- |
| Test case ID | TC8 |
| Test case name | Check Logout |
| Scenario | Test Logout from the system |
| Test Data | - |
| Expected Result | Displays “Successfully logged out. Have a nice day!” |
| Actual Result | Same as expected |
| Screenshot |  |
| Conclusion | The system will be logged out and returning to the authentication |

|  |  |
| --- | --- |
| Test case ID | TC9 |
| Test case name | Check Help |
| Scenario | Test displaying the help menu |
| Test Data | - |
| Expected Result | Displays the help menu |
| Actual Result | Same as expected |
| Screenshot |  |
| Conclusion | System will show the help area |
| Test case ID | TC10 |
| Test case name | Check Exit |
| Scenario | Test existing the application |
| Test Data | - |
| Expected Result | Displays an exit message and exits from the application. |
| Actual Result | Same as expected |
| Screenshot |  |
| Conclusion | System will close |

# 7 Café Mocha App User Guide

## 7.1 Overview of the System

The Café Mocha App is an overall management application for Café Mocha Store, enabling users to operate menu items, process orders efficiently, and calculate bills. The system will have all the necessary capabilities for the administration to manage the processes: adding or removing users and customers, updating menu items, and generating bills from customer orders

## 7.2 Navigating the Menu

The application menu is relatively simple and easy to use, hence very easy to explore and master all functionalities of the app. The following main menu will be displayed after user login:

* Login: Log in to the system using valid credentials.
* Place New Order: Enables staff to enter customer orders—new items ordered, their quantities, and special instructions.
* Order Information Display: Displays all the current orders in the system.
* Modify Order: This provides an interface to edit an already existing order.
* Delete Order: Allows the user to delete any order posted by indicating the Order ID.
* Add Menu Item: Add new menu items by entering an item name, price, and description.
* Display Menu: Lists the availability of all menu items in the Café.
* Generate Bill: Calculates the total bill for a given order, item-wise and quantity-wise.
* Help: Provides help on how to use the system.
* Exit: Exits the application.
* Users do this by entering the appropriate number on the menu to perform the intended action.

## 7.3 Performing Common Tasks

* Login: The application runs and the user types in their username and password. In case the credentials are correct, it redirects to the main menu.
* Add New Order: Click Add New Order from the main menu. Fill in the information of the order, like the customer and the menu items chosen. Save to send the changes back into the system.
* Modify Order: Change the order by selecting "Modify Order" from the main menu, followed by entering the Order ID and changing the details accordingly. Remember to save the changes for them to take effect in the system.
* Deleting Order: A menu to indicate "Delete Order" will be shown, and it proceeds to ask for the Order ID; upon confirmation, it will delete the order from the system.
* Bill Generation: On selecting "Generate Bill" from the main menu, it will ask for the Order ID, calculate the total costs based on items ordered and their prices, and save it in the bills.txt.

# 8 Annotated Screenshots

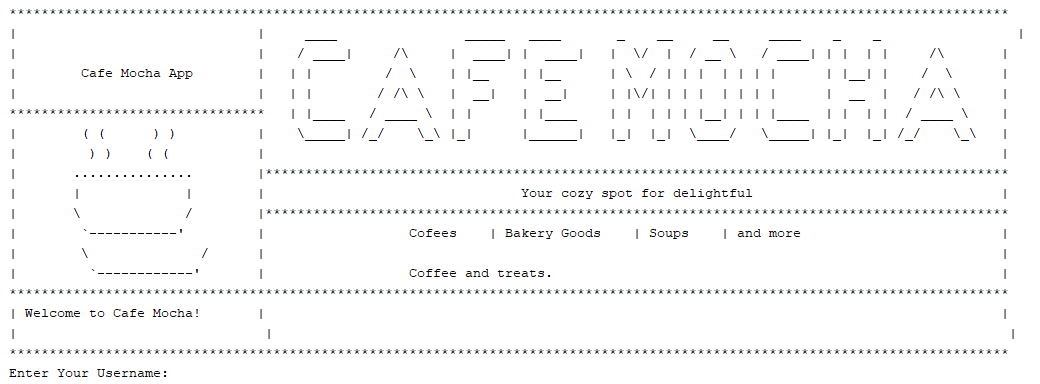


Figure 10 Entering Screen

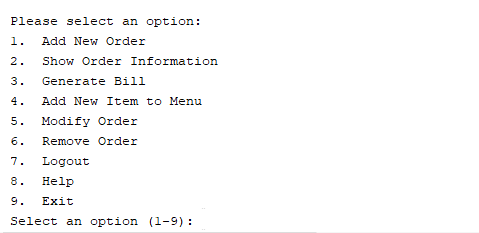


Figure 11 App Menu

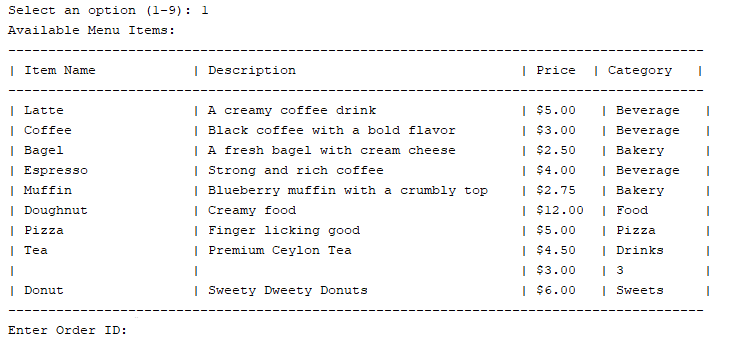


Figure 12 Food Menu

# References

Programiz. Java Methods. [online] programiz.com. Available at: https://www.programiz.com/java-programming/methods [Accessed 30th August 2024].

Tutorialspoint. File Handling in Java. [online] tutorialspoint.com. Available at:

https://www.tutorialspoint.com/file-handling-in-java [Accessed 30th August 2024].

W3schools (2024). Java Tutorial. [online] w3schools.com. Available at:

https://www.w3schools.com/java/ [Accessed 30th August 2024].