

**MINI PROJECT REPORT**

**CAFE MANAGEMENT SYSTEM**

*SUBJECT:*

CORE JAVA PROGRAMMING

*TOPIC:*

CAFE MANAGEMENT SYSTEM

*FACULTY CO-ORDINATOR:*

PROF. BABITA PATEL

*MADE BY:*

AISHWARYA SINGH (IU2241230272)

ZEAL SHAH (IU2241230253)

* **Introduction:**

The Cafe Management System is a Java application designed to streamline and simplify the process of managing a cafe's inventory and transactions. Developed using NetBeans IDE and Swing library, the system provides an intuitive user interface for customers to select food items, adjust quantities, and make purchases, while also offering administrative functions for generating receipts, resetting selections, and exiting the program.

* **Abstract:**

The Cafe Management System is a Java-based application developed using NetBeans IDE and Swing library to facilitate efficient management of cafe operations. This report outlines the features, implementation, and benefits of the system.

The system offers a user-friendly interface for customers to select food items, adjust quantities, and make purchases, with real-time updates of selected items and total cost. Administrative functions include receipt generation, reset options, and program termination.

Implemented through event handling and GUI design principles, the system provides a seamless user experience and streamlines cafe operations. Future enhancements may include database integration, user authentication, and additional features for inventory management and sales reporting. Overall, the Cafe Management System exemplifies the versatility and effectiveness of Java programming in the hospitality industry.

* **Java Libraries Used:**

1. **`java.awt`-** provides classes for creating and managing graphical user interfaces.
2. **`java.awt.print.PrinterException`-** handles exceptions related to printing.
3. **`java.text.SimpleDateFormat**`- formats dates.
4. **`java.util.Date`-** gets the current date and time.
5. **`javax.swing.ImageIcon**`- displays images in Swing components.
6. **`javax.swing.JOptionPane`-** provides a simple mechanism for displaying dialog boxes for user interaction.

* **Work Flow of the Program:**

Here's a basic workflow for our cafe management system program:

1. Start the Program:

- Launch the program either by running the executable JAR file or directly from the NetBeans IDE.

2. Main Interface Display:

- Upon starting the program, the main interface (JFrame form) appears.

- The interface displays a template with food items, buttons to adjust quantities, and a purchase button.

- The right side of the interface shows the selected items and their total cost.

- At the bottom, there are four buttons: "Total," "Receipt," "Reset," and "Exit."

3. Selecting Food Items:

- Users can browse through the food items displayed.

- They can click on the buttons associated with each item to increase or decrease the quantity.

4. Calculating Total Cost:

- As users adjust quantities, the program dynamically updates the total cost displayed on the right side of the interface.

5. Viewing Total Quantity and Price:

- Users can click the "Total" button to view the total quantity and price of all selected items.

6. Generating Receipt:

- Upon clicking the "Receipt" button, the program generates a receipt.

- The receipt will include details such as item names, quantities, individual prices, and the total cost.

7. Resetting Selections:

- If users want to start over, they can click the "Reset" button.

- This action clears all selections and resets the total cost to zero.

8. Exiting the Program:

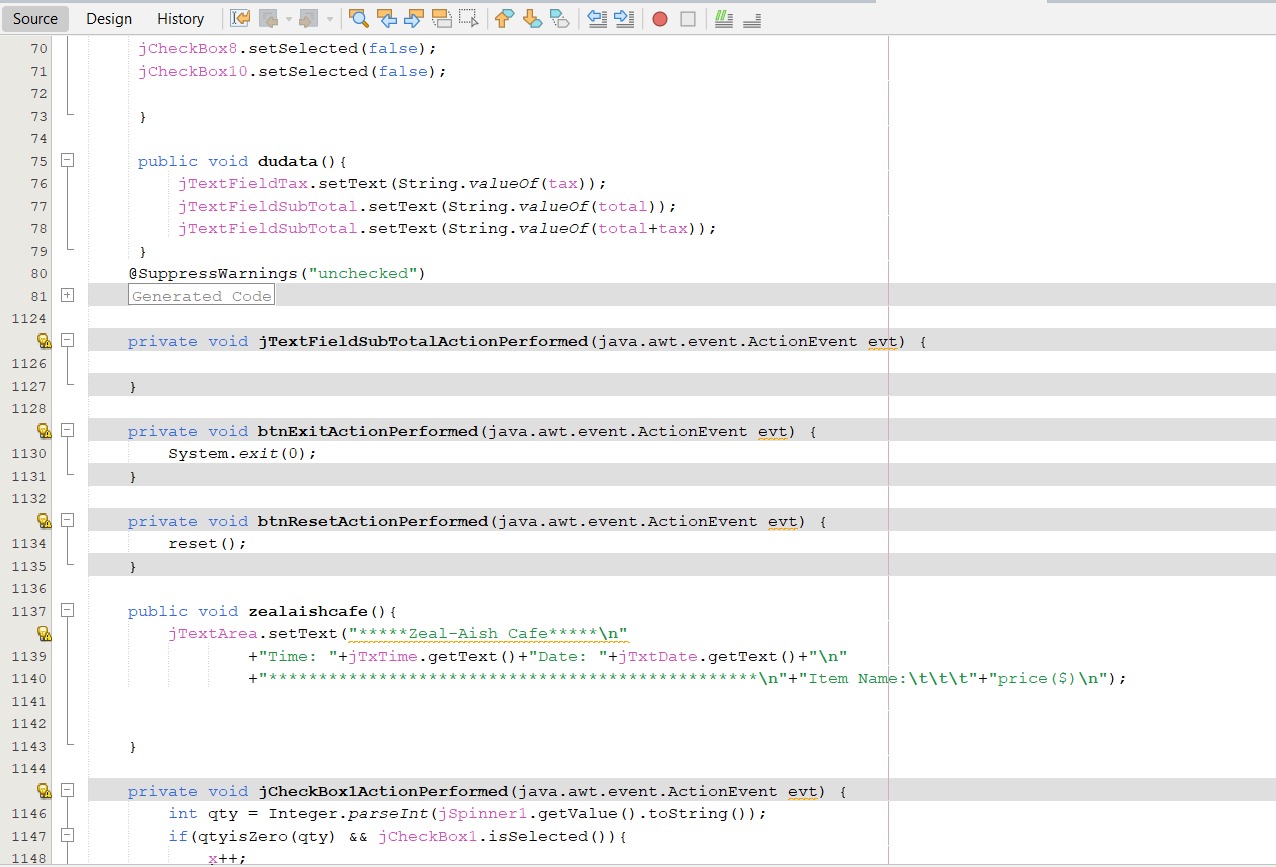
- Finally, users can choose to exit the program by clicking the "Exit" button.

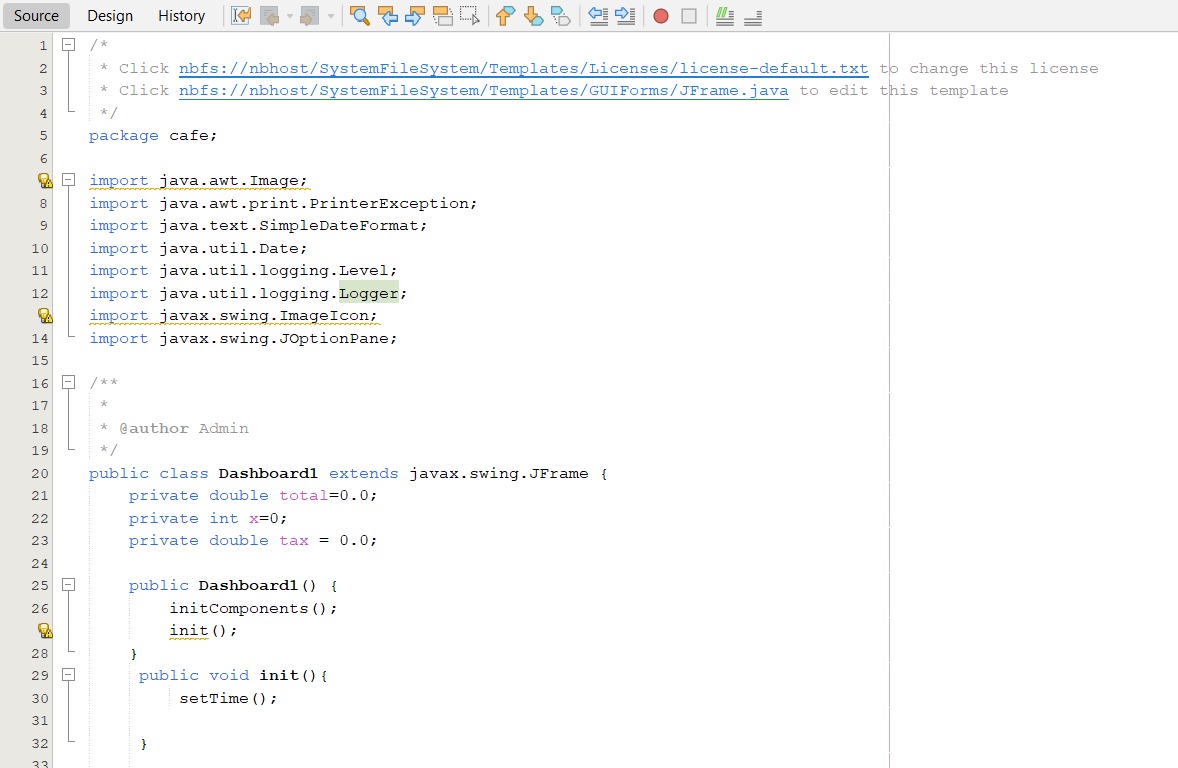
- This action closes the JFrame and terminates the program.

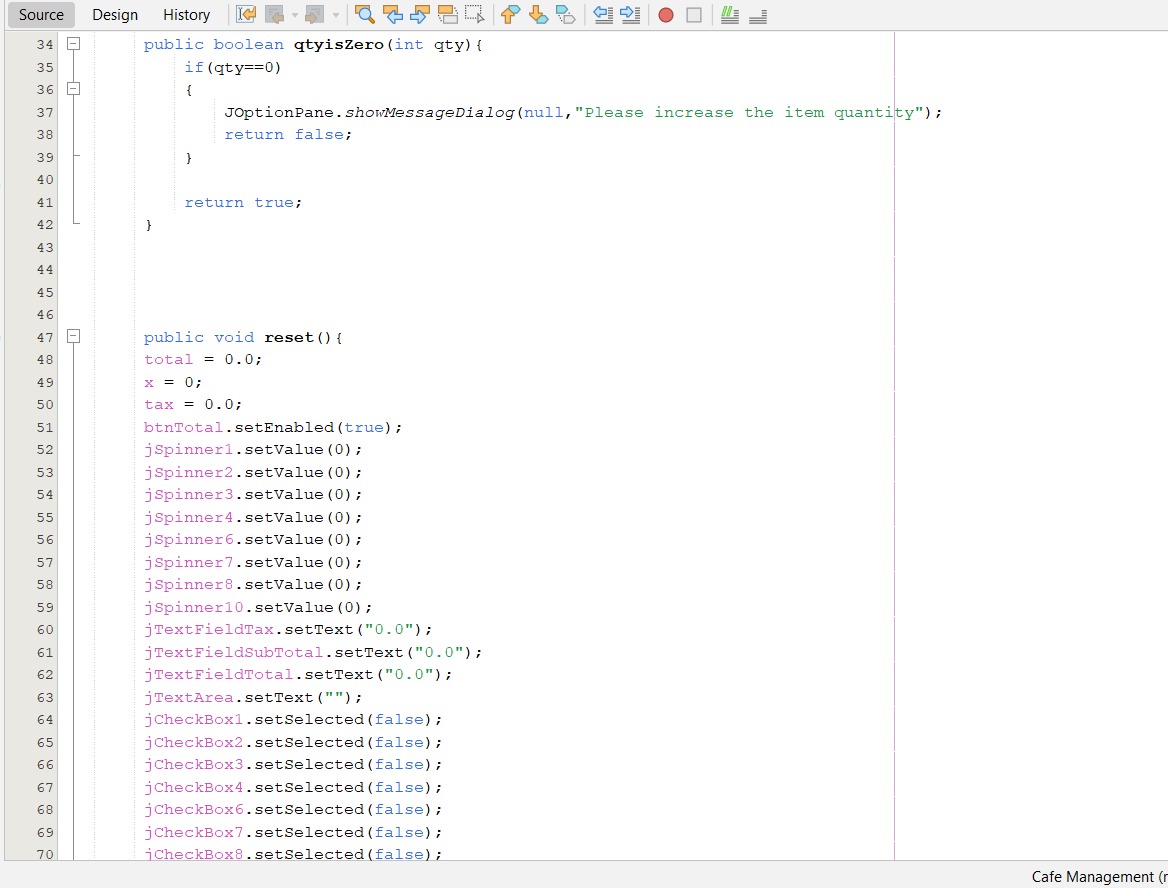
* **Implementation:**

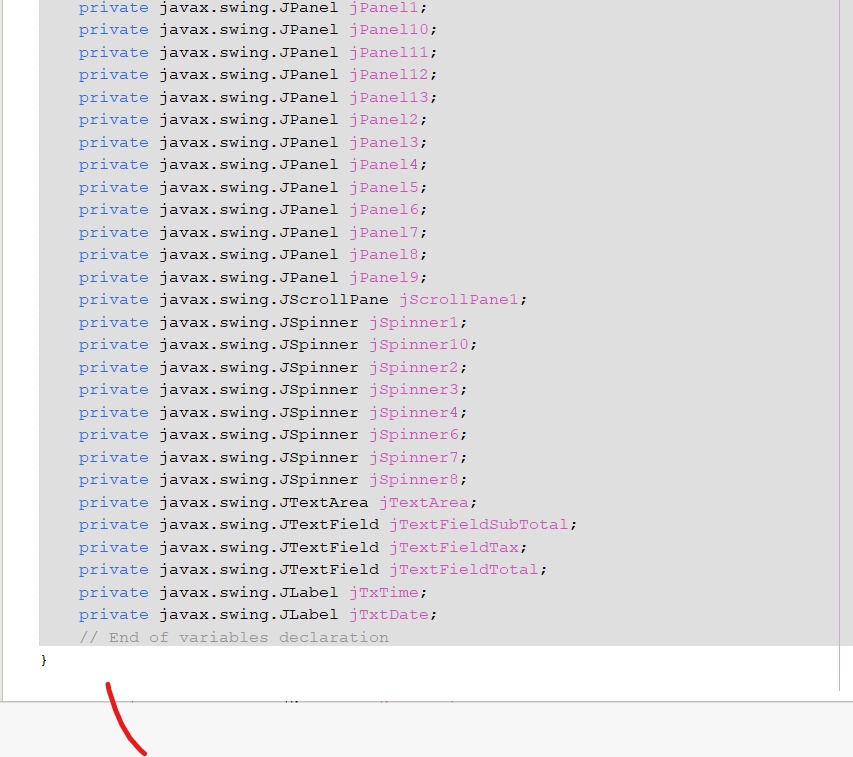
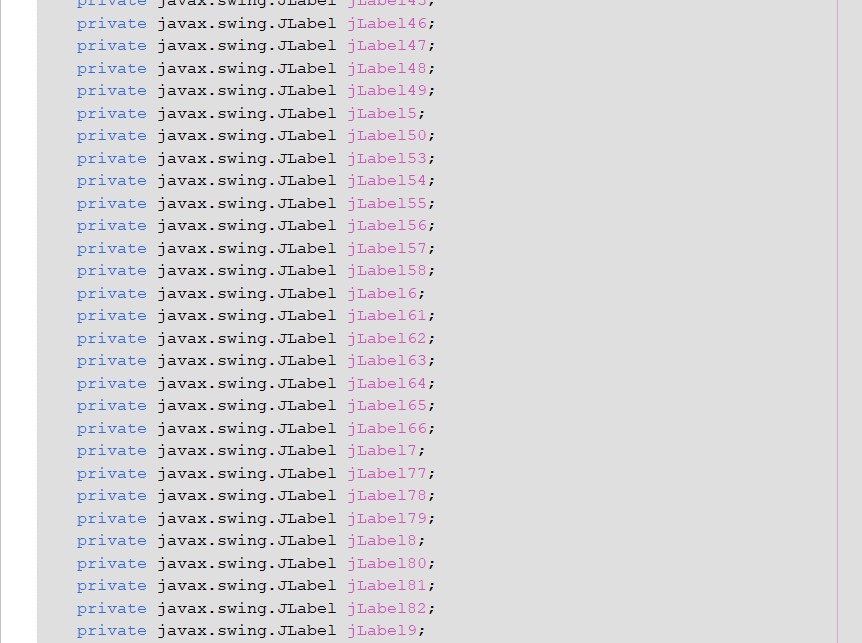
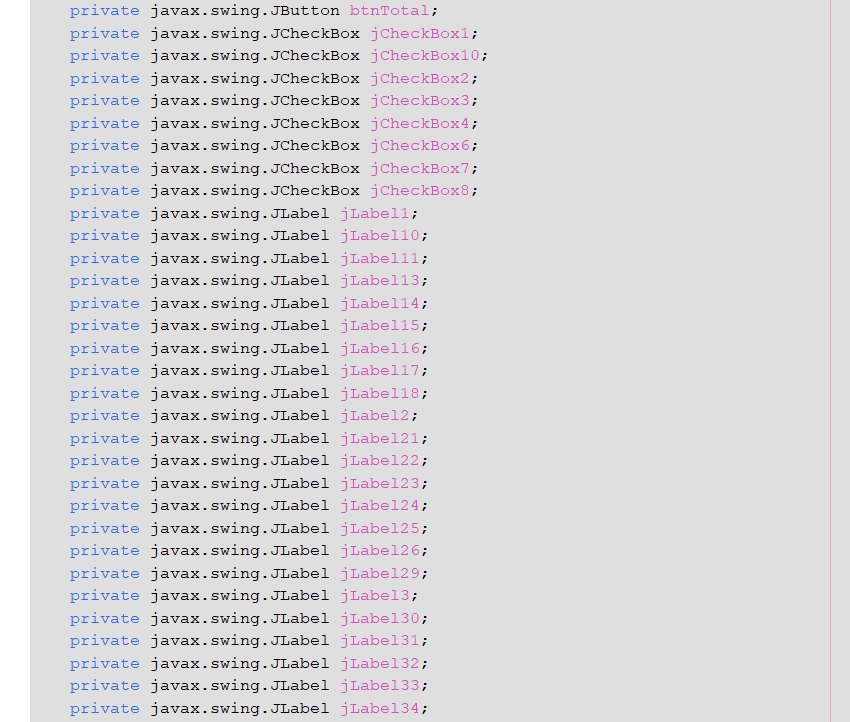
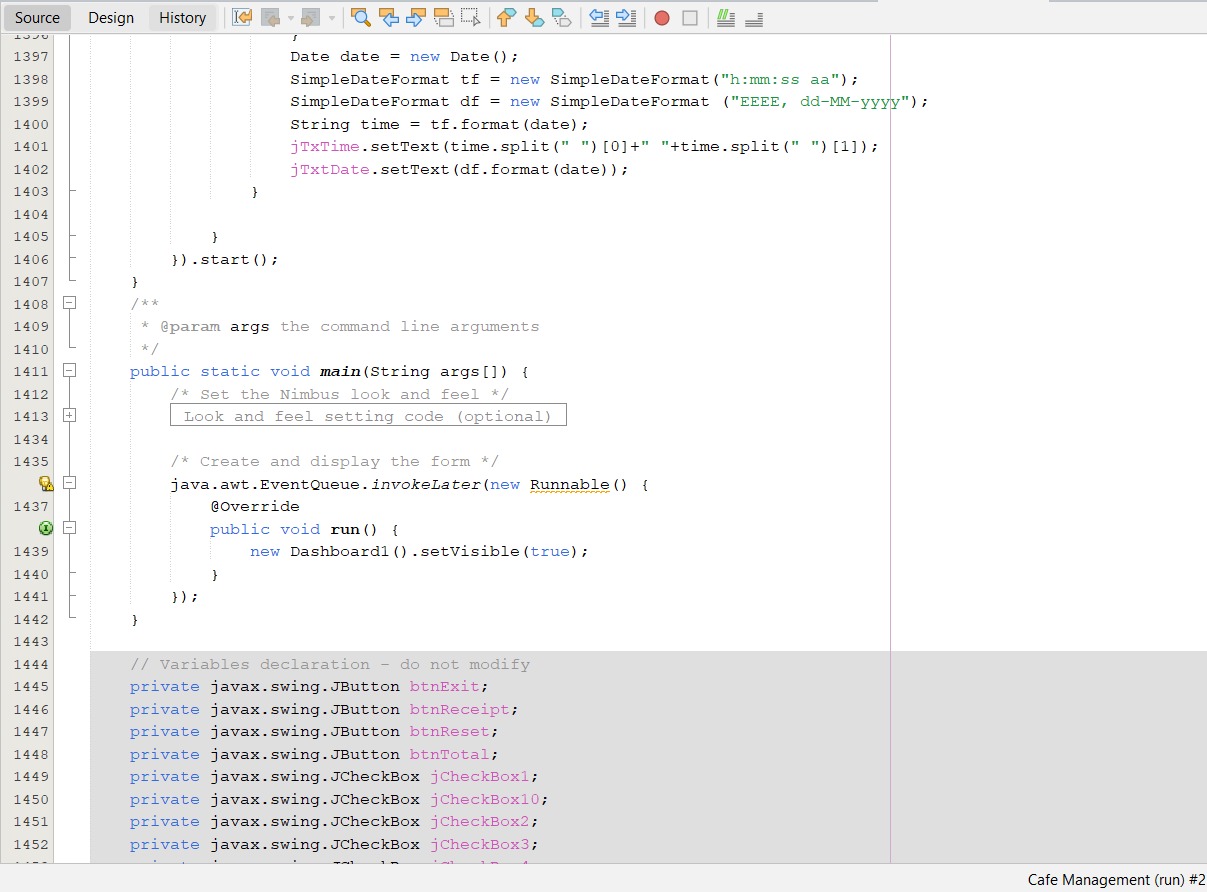
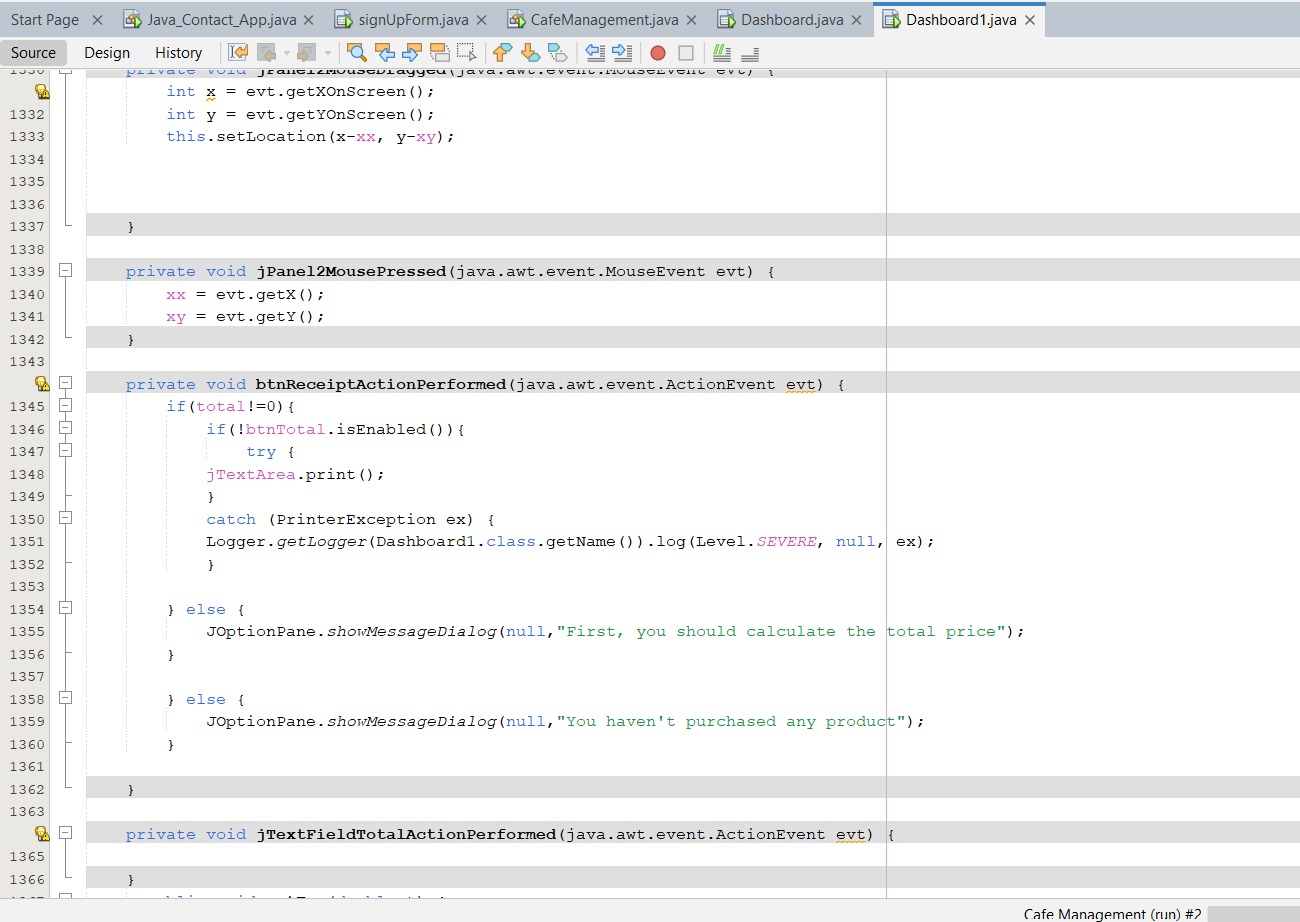
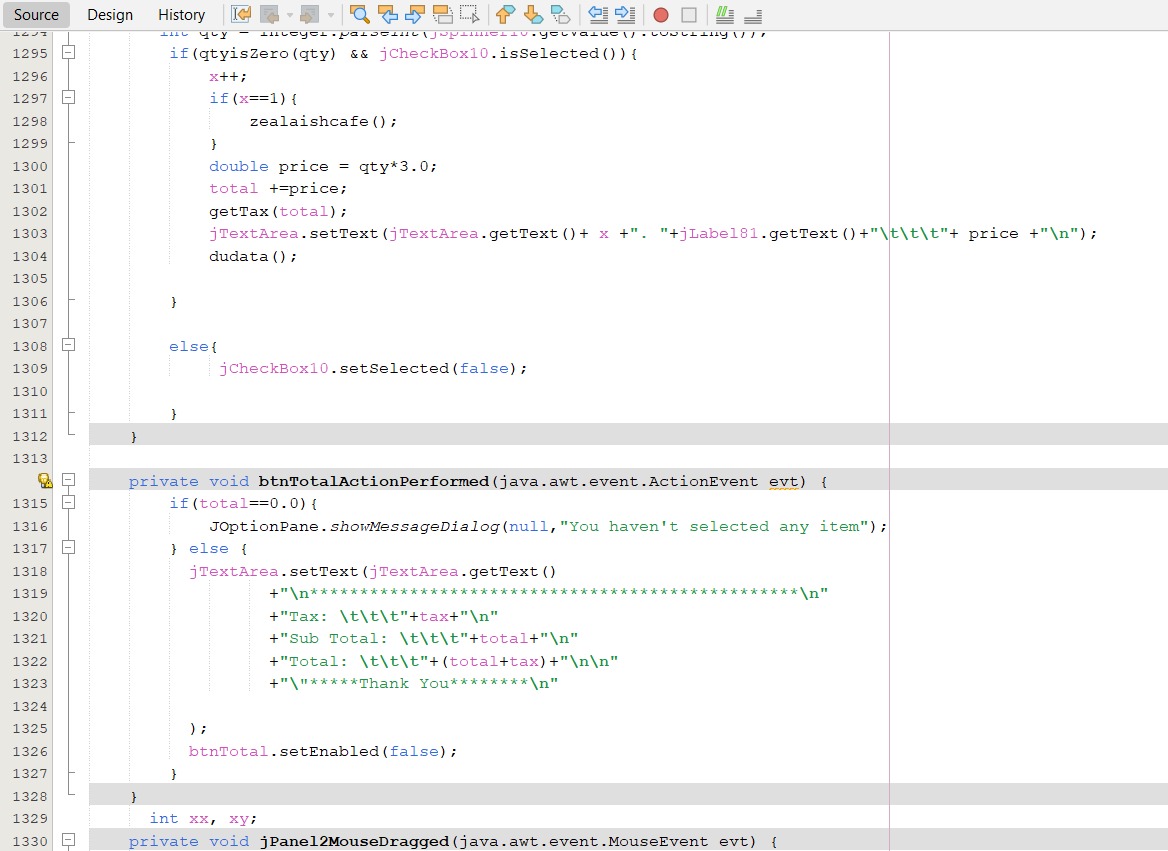
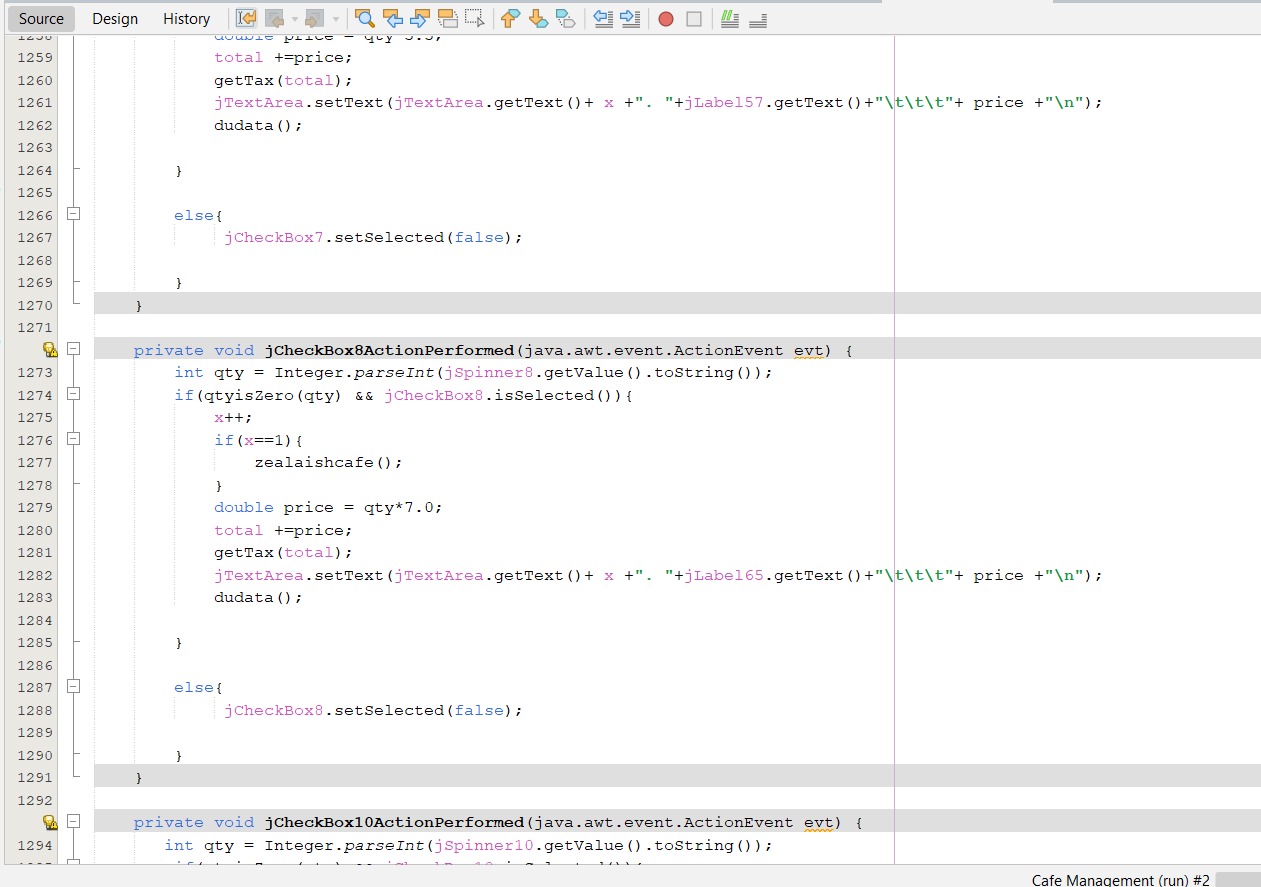
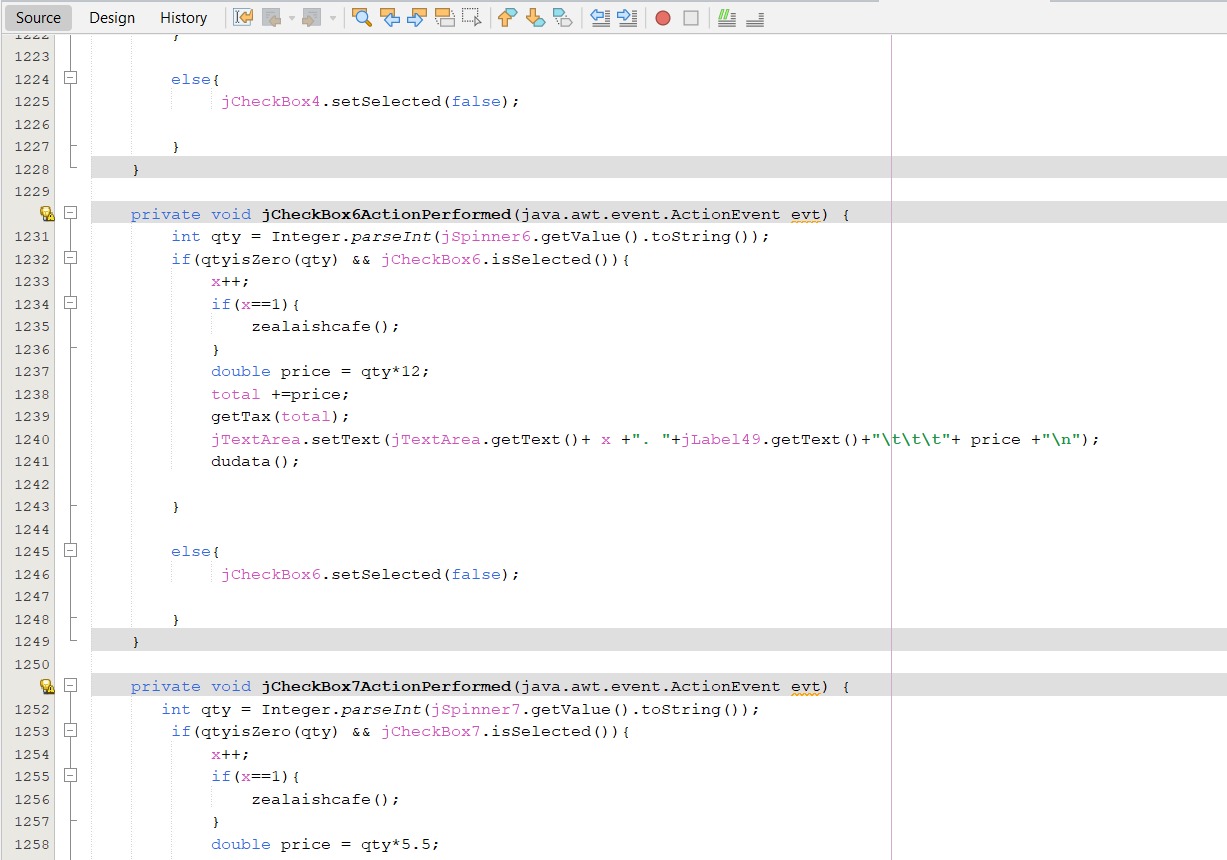
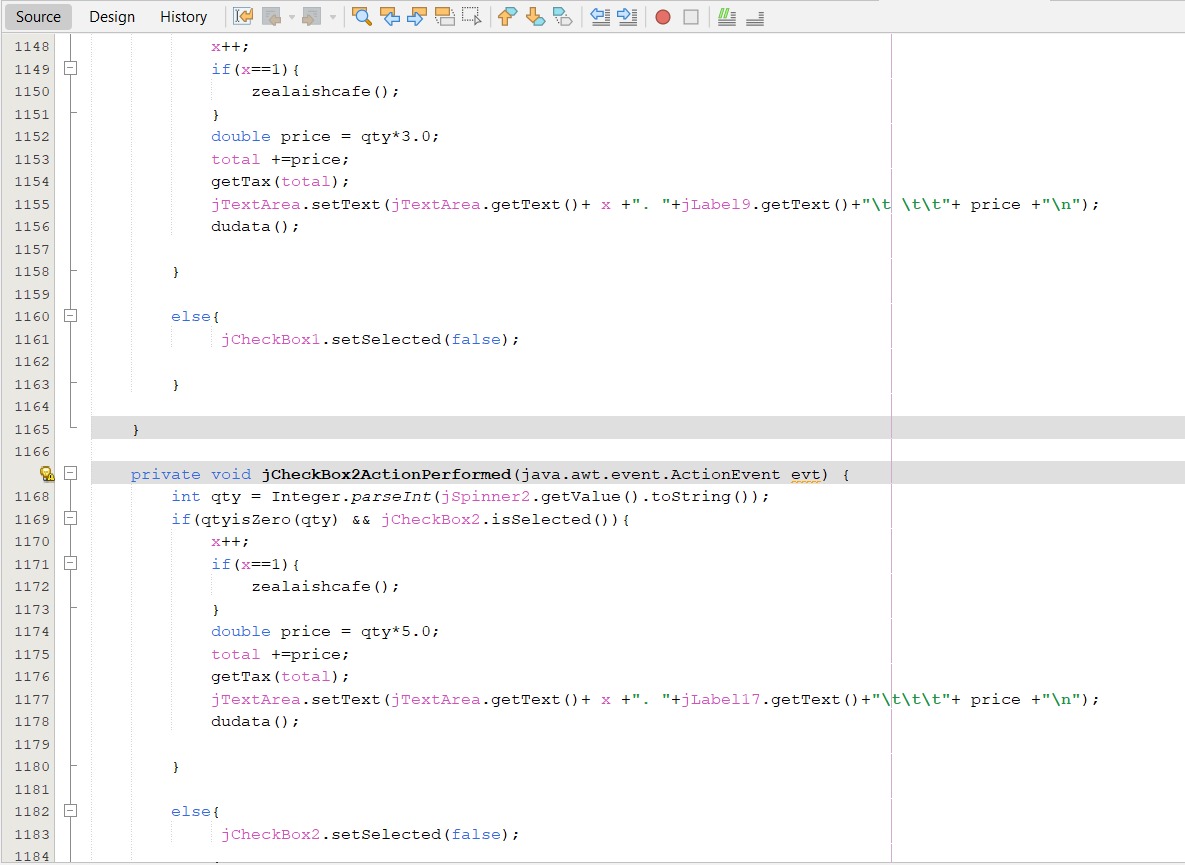
The system is implemented in Java using NetBeans IDE and the Swing library. Each feature is realized through a combination of event handling, data manipulation, and GUI design principles. Key components such as buttons, labels, text fields, and panels are strategically placed and styled to enhance usability and aesthetic appeal.

* **Code:**





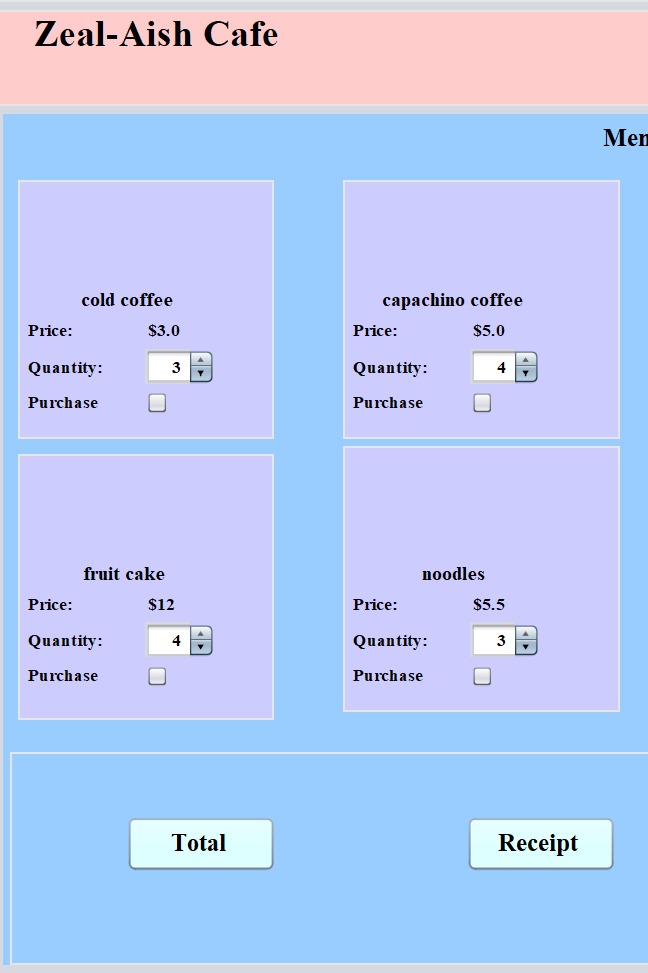




* **Output:**



* ADDING ITEMS

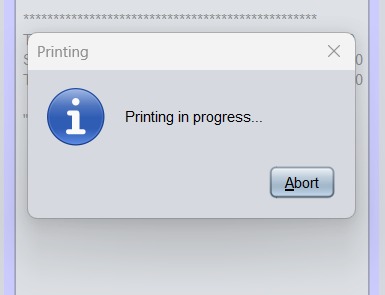
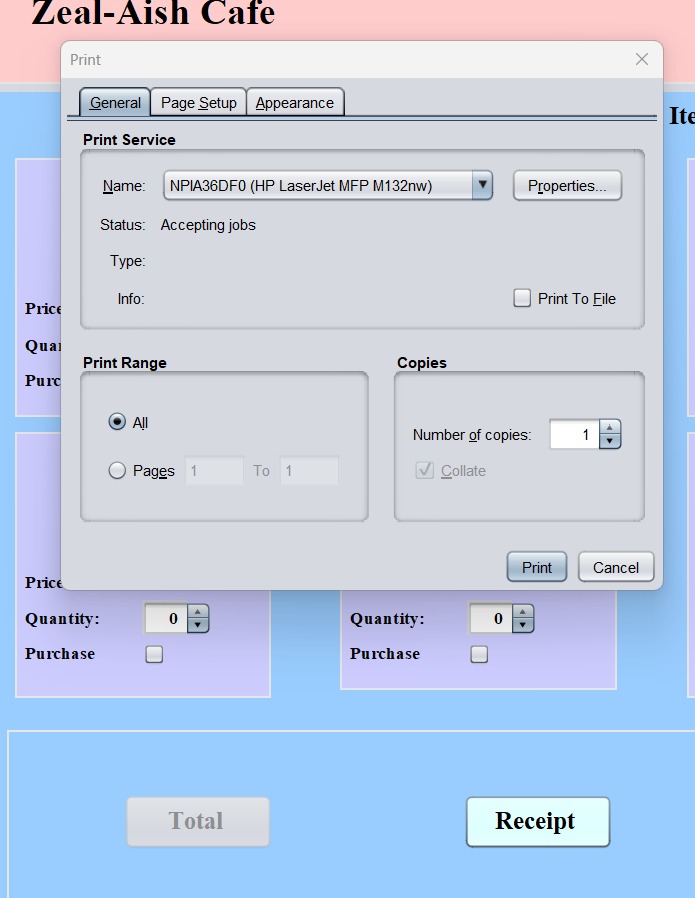


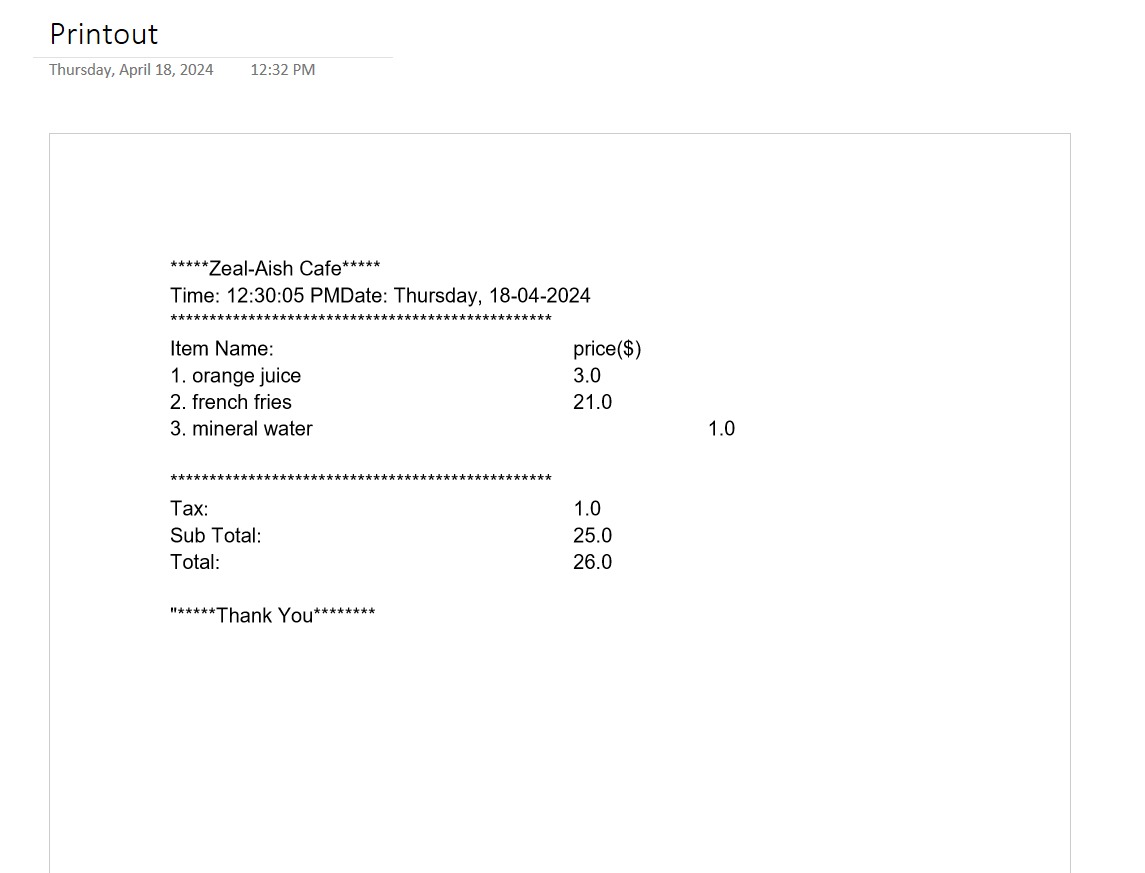
* CLICKING PURCHASE BUTTON



* TOTAL BUTTON:



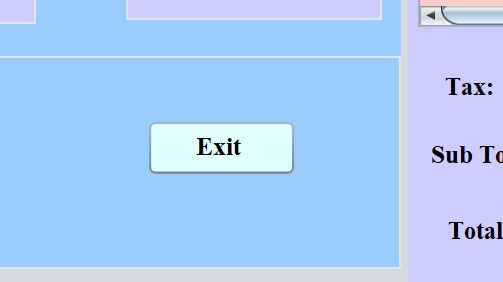
* RECEIPT BUTTON(ASKS TO PRINT RECIEPT): 
* RECEIPT PRINTED:



* RESETING TO DEFAULT SETTINGS:



* EXITING THE PANEL:



* **Importance:**

1. Efficiency Boost: Automates order processing, inventory tracking, and billing, saving time and reducing errors for cafe staff.

2. Customer Satisfaction: Enables faster service and accurate billing, leading to improved customer experience and loyalty.

3. Data Insights: Gathers valuable sales and customer preference data, aiding in informed business decisions and strategic planning.

4. Financial Management: Assists in budgeting, forecasting, and profitability analysis through streamlined sales and expense tracking.

5. Scalability and Professionalism: Supports future growth and demonstrates professionalism, enhancing competitiveness in the market.

* **Future Enhancements:**

- Integration with a database for storing and retrieving menu items, customer orders, and transaction history.

- Implementation of user authentication and authorization for secure access to administrative features.

- Addition of features such as inventory management, sales reporting, and customer loyalty programs to further enhance the system's capabilities.

* **Conclusion:**

The Cafe Management System offers a comprehensive solution for managing cafe operations efficiently. By leveraging Java's capabilities and Swing library, the system delivers a seamless user experience with intuitive controls and robust functionality. Whether for small- MMMMscale cafes or larger establishments, this system can streamline ordering processes, improve customer service, and simplify administrative tasks.

SIGNATURE\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_