

**NORTH SOUTH UNIVERSITY**

**Project Report on**

**Coffee Shop Management System**

**(Kashundi Cafe)**

Faculty: **Dr. Maktuba Mohid Binni (MMBA)**

Department of Engineering and Computer Science

Lab Instructor: **Md Abdullah Al Sayed**

Section: 14 Semester: Spring 2024

Course: CSE215 (Java Programming)

* Submitted By:

**Group -5**

|  |  |
| --- | --- |
| Name | ID |
| Sakib Ahmed Shishir | 2312546642 |
| Choitannyo Saha | 2312832642 |
| Farhin Ahmed Pranto | 2312734642 |

Abstract

This report presents the design and implementation of a Coffee Shop Management System developed using Java programming language. The primary objective of this system is to streamline the ordering process in a coffee shop by facilitating efficient interactions between customers and employees. In this system, employees are responsible for taking orders from customers and generating bill receipts. A significant feature of this system is the administrative module, which is exclusively accessible to the admin. The admin has the authority to add new employees to the system and configure their login credentials, including usernames and passwords. This centralisation of employee management enhances the security and functionality of the system, ensuring that only authorised personnel can process orders and access sensitive information. The system is designed to be user-friendly, ensuring that employees can easily manage orders and generate receipts with minimal training. Overall, the Coffee Shop Management System aims to improve service speed, accuracy, and customer satisfaction in coffee shop operations, leveraging the robust capabilities of Java for backend processing.

GitHub Link: https://github.com/Sakibcheez/Kashundi-Cafe-Management-System-Java.git

**Acknowledgement**

We are immensely grateful for the opportunity to express our heartfelt thanks to everyone who guided and supported us throughout the completion of our project.

Foremost, we extend our profound appreciation to our Lab instructor, **Abdullah Al Sayed,** whose invaluable guidance was crucial to the success of this project. Without his support and expertise, this project would not have reached fruition.

We would like to thank our course faculty, **Maktuba Mohid Binni** mam, especially.

Lastly, we are thankful to all our friends for their unwavering encouragement and support throughout this project. Their motivation played a significant role in the completion of our work.

# Coffee Shop Management System

## **Introduction**

The Coffee Shop Management System is designed to streamline the operations of a coffee shop, facilitating efficient order management and overall administration. This Java-based system allows an administrator to manage data, update information, and oversee employee activities. Employees can log in, take orders, generate bills, and interact with customers, ensuring a smooth and effective service experience.

### Background

The "Coffee Shop Management System" is an automated and OS-based application tailored to enhance the management processes of a coffee shop. It provides a comprehensive platform where the admin can control data, and customers can suggest changes to the coffee offerings. This system enables the administrator to add, update, and delete information efficiently, thus maintaining an organized and updated database.

### Objectives

* **Develop a Web-Based System**: To create a web-based application that effectively manages coffee information.
* **User Suggestions**: To facilitate customer feedback for potential changes in coffee offerings.
* **Support Rural Areas**: To aid rural communities by offering a platform that caters to their needs.
* **Efficiency**: To reduce processing time and improve service delivery.
* **Quality Products**: To ensure the provision of fresh and healthy coffee products.

### Purpose and Scope

#### Purpose

* **Efficient Record Handling**: To save time and costs by automating record-keeping processes.
* **Automated Processes**: To streamline various operational tasks within the coffee shop.
* **Quality Assurance**: To consistently provide fresh and high-quality coffee.
* **Revenue Generation**: To enhance profitability through improved management.

#### Scope

The Coffee Shop Management System aims to efficiently manage and run the coffee shop. It offers functionalities for employees to take orders, process payments, and manage customer interactions. The administrator can oversee all records, including employee data, products, customers, orders, and billing information, ensuring a holistic view of the shop’s operations.

#### Applicability

* **User-Friendly**: Designed to be intuitive and easy to use.
* **High Security**: Ensures secure handling of sensitive data.
* **Flexibility**: More adaptable compared to previous systems.
* **Comprehensive Operations**: Supports all essential operations like add, view, delete, and update.
* **Transparency**: Provides clear and transparent processes for users.
* **Support for Farmers and Rural Areas**: Enhances earnings and supports rural communities through improved service delivery.

## System Analysis

### Existing System

* **Tedious Operations**: The previous system involved cumbersome processes.
* **Slow Data Processing**: Inefficiencies in data handling slowed down operations.
* **User Unfriendliness**: The earlier system was not easy to use.
* **Limited Orders and Benefits**: Restricted order capacity and minimal benefits.

### Fact-Finding Requirements

1. **Operations to be Performed**:
   * **Update**: Modify existing information.
   * **Delete**: Remove outdated or unnecessary data.
   * **Insert**: Add new records and information.
   * **View**: Access and review data.

**Proposed System:-**

• This system is user-friendly.

• It provides high security.

• It contains all operations like add, update, delete, view etc.

• It is an automated system.

• This system helps to place orders for coffee.

**Requirement Analysis**

I will use fact-finding techniques in my project, i.e., Coffee Shop Management. I gather information for my project according to the client's requirements, so the client can handle this system easily. It provides high security. Coffee Shop Management is an automated system that allows customers to order coffee easily and at any time.

**Hardware Requirements:-**

* Processor: Intel / AMD, which supports JDK Applications
* Hard Disk: 2 GB
* RAM: 2GB

**Software Requirement:**

Design Constraint:

• Front End: - Microsoft .net framework with Microsoft Visual Studio 2015

• Language: - English Requirement User:

• Operating system: - Any Operating System

• Browser: - Any Browser, jdk file

IDE: Apache NetBeans, Eclipse, IntelliJ Idea

JDK version: 20 or higher

**Code Contains:**

**Package Name:** cafeguii.app

.java File: CoffeShopCalculator.java

Class name: CoffeShopCalculator

Constructor name: CoffeShopCalculator()

Method name: calculateTax(double subtotal)

**Package Name:** cageguii.gui

.java File: Dashboard.java

Class name: Dashboard, ProductPeanel, YourClassName, JPanel,GroupLayout, SpinnerNumberModel

Constructor name : Dashboard(), YourClassName(), Font(String name, int style, int size), SpinnerNumberModel(int value, Comparable minimum, Comparablemaximum, int stepSize)

Method name: init(),setImage(),qtyIsZero(int qty),reset(),inintComponents(),ProductPanel(),

setBorder(Border border),setFont(Font font),Kashundi\_Cafe(),setTime(),run(),dudate(),setText(String text),setHorizontalAlignment(int alignment),addComponent(Component comp, Object constraints)

,addPreferredGap(LayoutStyle.ComponentPlacement type),actionPerformed(ActionEvent evt),

jCheckBox1ActionPerformed(java.awt.event.ActionEvent evt)

btnTotalActionPerformed(ActionEvent evt)

btnReceiptActionPerformed(ActionEvent evt)

btnResetActionPerformed(ActionEvent evt)

btnExitActionPerformed(ActionEvent evt)

**Package Name:** cafeguii.images2

(All images of food for dashboard)

**Package Name:** cafeguii.menu

.java File: CoffeShopItem.java

MenuItem.java

Class Name(CoffeShopItem.java) : CoffeeShopItem

Constructor Names(CoffeShopItem.java): CoffeeShopItem(String name, double price, String imagePath, JLabel label)

Interface Name: MenuItem

Method Names : (MenuItem.java): getName()getPrice()

Method Names (CoffeShopItem.java): getName()

getPrice(),getIcon(),getLabel(),createImageIcon(String path)

**Package Name:** Homepage

.java File: AdminPanel.java

EmployeeDashboard.java

EmployeeUtils.java

KashundiCafe.java(Main Executable Run file of the project)

LoginPanel.java

Class Names (AdminPanel.java): AdminPanel

Method Names (AdminPanel.java): showEmployees()

*Action Listener Methods*

actionPerformed(ActionEvent e) for btnAddEmployee

actionPerformed(ActionEvent e) for btnShowEmployees

actionPerformed(ActionEvent e) for btnback

actionPerformed(ActionEvent e) for btnmenu

actionPerformed(ActionEvent e) for btnsell

actionPerformed(ActionEvent e) for btnincome

Class Name (EmployeeDashboard.java): EmployeeDashboard

Constructor Names (EmployeeDashboard.java): EmployeeDashboard()

Class Name (EmployeeUtils.java): EmployeeUtils

Method Names (EmployeeUtils.java): loadEmployees(Map<String, String>

employees, String filename)

saveEmployees(Map<String, String>

employees, String filename)

Class Names (KashundiCafe.java): KashundiCafe

LoginPanel

AdminPanel

EmployeeDashboard

EmployeeUtils

Constructors Names (KashundiCafe.java): KashundiCafe()

LoginPanel(KashundiCafe mainApp)

AdminPanel(KashundiCafe mainApp)

EmployeeDashboard()

Method Names (KashundiCafe.java): main(String[] args)

initialize()

showAdminPanel()

showEmployeeDashboard()

getEmployees()

saveEmployees()

loadEmployees(Map<String, String>

employees, String filename) in

EmployeeUtils

saveEmployees(Map<String, String>

employees,String filename) in

EmployeeUtils

Class Name (LoginPanel.java):

LoginPanelConstructor Names (LoginPanel.java): LoginPanel(KashundiCafe mainApp)

Method Names (LoginPanel.java): login()

getlogin()

**Package Name:** Data File

FileReader.java --------abstract classs

Sales\_History.java -------public class extend JFrame

-constructor

Sales\_History()

private void displayFileContent()

TextFileReader.java-------public class extend FileReader

Total\_income.java--------public class extends JFrame

-constructor

public Total\_income()

private void displayFileContent()

**All .java Classes are:**

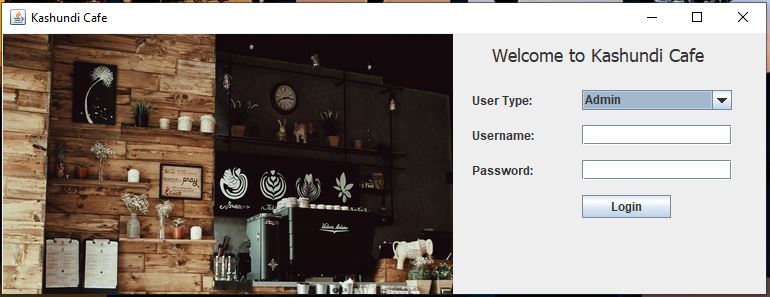
* AdminPanel.java
* EmployeeDashboard.java
* EmployeeUtils.java
* KashundiCafe.java (Main Executable Run file of the project)
* LoginPanel.java
* FileReader.java
* Sales\_History.java
* TextFileReader.java
* Total\_income.java
* CoffeShopCalculator.java
* Dashboard.java
* CoffeShopItem.java
* MenuItem.java

**Project Features and Short Descriptions:**

|  |  |  |
| --- | --- | --- |
| Features | Description | Developed  By |
| **Login Panel**   1. Welcome Label 2. User Type Selection 3. Username Input 4. Password Input 5. Login Button 6. Login Process 7. Helper Method(getlogin) | The *Login Panel* class is a Swing GUI for the Kashundi Cafe application, enabling user login as either Admin or Employee. It features fields for username and password, a dropdown for user type selection, and a login button. Upon submission, it validates credentials and navigates to the appropriate dashboard, displaying error messages for invalid entries. | Farhin Ahmed Pranto |
| **Admin Panel**   1. Admin Panel Label 2. Add Employee 3. Show Employee 4. Remove Employee 5. Logout 6. Menu Navigation 7. Dynamin Employee | The *Admin Panel* class is a Swing GUI for the Kashundi Cafe application, providing administrative functionalities. It allows admins to add new employees, view the employee list, and remove employees. The interface includes fields for entering employee details and buttons for adding employees, displaying the employee list, logging out, and accessing the employee dashboard. | Farhin Ahmed Pranto |
| **Employee Utils**   1. Load Employees from file 2. Save Employees to file | *Employee Utils* provides methods for loading and saving employee data. *loadEmployees* reads username-password pairs from a file into a map. *saveEmployees* writes username- password pairs from a map to a file. Both methods handle file I/O and ensure data integrity. Errors during file operations are reported. This utility assists in managing employee information for the Kashundi Cafe  application. | Choitannyo Saha |
| **Employee Dashboard**   1. Employee Dashboard Label 2. Integration with External Dashboard 3. Unsupported   ‘dispose()’method | The *Employee Dashboard* class represents the user interface for employees in the Kashundi Cafe application. It displays a dashboard with relevant information and options. The layout is set manually using absolute positioning. Additional components can be added as required. Currently, it disposes of the panel, awaiting further implementation. | Choitannyo Saha |
| **Kashundi Cafe**   1. The main executable class | *Kashundi Cafe* manages employee information. It includes a login system with admin and employee roles, an admin panel  for managing employees and an employee | Choitannyo Saha |

|  |  |  |
| --- | --- | --- |
|  | dashboard. The application uses a HashMap to store employee data and reads and writes to a file named "employees.txt". | , Sakib Ahmed Shishir, Farhin  Ahmed Pranto |
| **Coffee Shop Calculator**  1. Calculating with Tax | The *Coffee Shop Calculator* class manages calculations for a coffee shop's transactions. It initializes variables for total, items, and tax. The constructor sets default values. The *calculateTax* method computes tax based on a subtotal using a fixed rate. This utility assists in computing taxes for purchases in  the coffee shop application. | Farhin Ahmed Pranto |
| **Coffee Shop Item**   1. Attributes 2. Constructors 3. Getter Methods 4. Helper Methods | The *Coffee Shop Item* class represents an item in the cafe's menu. It stores the name, price, and icon of the item. The constructor initializes these attributes with the provided parameters. It also provides methods to retrieve the item's details, such as name,  price, icon, and associated label. | Sakib Ahmed Shishir |
| **Dashboard**   1. Design of Dashboard 2. The Billing System | *Dashboard* contains the design of the menu bar, purchase sign, and the design of the billing system and the printing feature which  will save as a PDF or we can print the slip from the printer. | Sakib Ahmed Shishir |
| **Data File**   1. Abstract class File Reader 2. Sell History 3. Total Income | In admin panel admin can see sell history and total income by clicking sell info and total income. The .txt file passes through an object file called Path. | Sakib Ahmed Shishir |

**Project Description Part:**

* 1. **Home Page**

It is the first look of Kashundi Café (Coffee Shop Management System) when we run the Programme. The interface shows a "Kashundi Cafe" window with a welcoming message: "Welcome to Kashundi Cafe." There is a dropdown menu labelled "User Type" with the option "Admin" and “Employee”. Below the "User Type" dropdown, there are fields for "Username" and "Password" to log in. There is a "Login" button below the password field. The right side of the interface displays a cosy and aesthetically pleasing coffee shop interior with wooden shelves, coffee equipment, and decorative items, providing a welcoming and inviting atmosphere.

**Difficulties:**

1. **Data File Rearrangement:** Rearranging data files can be time-consuming, especially if there's a lot of data to organise or if the file structure needs frequent changes.
2. **Data File Handling:** Managing data files within your Java project, including reading from and writing to them, can be challenging, particularly if there are complex data structures involved.
3. **Object Handling:** Calling objects to display data information can sometimes be tricky, especially if there are dependencies or if the data retrieval process is complex.
4. **GUI and Interface Design:** Designing a user-friendly graphical user interface (GUI) and implementing interface actions can pose challenges, especially when trying to ensure smooth interaction and intuitive navigation for users.

**Limitations:**

1. **Closing Panel Issue:** The limitation where closing a panel box results in the entire program closing is a significant usability issue. It restricts the user's ability to navigate within the application seamlessly and can lead to frustration.

In the Admin Panel Section, when the admin opens the total income or total sales, it closes that section only, but it closes the whole program. We tried a lot to fix the problem, but we could not. We tried when the admin showed the total income or sales information from the admin panel and closed the box, and then it returned to the admin panel section.

## **Conclusion**

The Coffee Shop Management System (CSMS) developed for Kashundi Cafe has successfully streamlined the ordering process, improving efficiency and customer satisfaction. This Java-based system allows administrators to manage employee data securely, ensuring only authorised personnel have access to sensitive information. The system minimises training requirements and enhances operational efficiency by providing an intuitive interface for employees to manage orders and generate receipts.

Throughout the project, we faced challenges such as data file rearrangement, GUI design complexities, and object handling. Despite these difficulties, the team managed to create a functional and user-friendly system. However, limitations remain, particularly with the panel closing issue in the Admin Panel, which affects the overall user experience. This is an area for future improvement.

The CSMS embodies our goal of leveraging technology to create a seamless and efficient management solution for coffee shops. It simplifies operations and supports rural communities by offering a platform that caters to their specific needs. Moving forward, continuous enhancements and addressing the identified limitations will ensure the system remains robust and effective, contributing to better service delivery and operational excellence in coffee shop management.