

**GOVERNMENT SHAHID AKBAR ALI SCIENCE & TECHNOLOGY COLLEGE**

**BALIADANGI, THAKURGAON**

**AFFILIATED BY**

**HAJEE MOHAMMAD DANESH SCIENCE & TECHNOLOGY UNIVERSITY,**

**DINAJPUR**



**PROJECT NAME: RESTAURANT BILLING SYSTEM**

**Course Title: Application Development Sessional**

**Department: Computer Science & Engineering**

**Level -2 Semester- II**

**Course Code: CSE 252**

**Supervised by:**

Polash Singha, Lecturer

Department of Computer Science & Engineering

Govt. Shahid Akbar Ali Science & Technology College, Baliadangi, Thakurgaon.

**Presented By:**

Partha Sarathi Roy

Student ID: 2120104

Session: 2021

Md.Alamgir Hossain

Student ID: 2120108

Session: 2021

## **ABSTRACT**

This paper presents the development of a Restaurant billing system using Java and Java Swing for the graphical user interface (GUI). The system is designed to automate the billing process, allowing the user to select Restaurant items, calculate prices, and generate receipts. The application incorporates object-oriented programming principles, making it modular and easy to extend. Java Swing provides a user-friendly interface for seamless interaction. The system supports features like item selection, real-time price calculation. This solution aims to enhance efficiency and reduce manual errors in Restaurant operations.

## TABLE OF CONTENTS

|  |    |
|--|----|
| <b>CHAPTER 1: INTRODUCTION</b>                       | 1  |
| 1.1 Background of the Restaurant Billing System      | 1  |
| 1.2 Problem Statement                                | 1  |
| 1.3 Objectives of the System                         | 1  |
| 1.4 Scope of the System                              | 1  |
| 1.5 Challenges Previous Systems                      | 2  |
| 1.6 Challenges                                       | 2  |
| <b>CHAPTER 2: SYSTEM DESIGN AND IMPLEMENTATION</b>   | 3  |
| 2.1 Functional Requirements                          | 3  |
| 2.2 Use Case Diagrams                                | 3  |
| 2.3 Programming Language and Tools                   | 4  |
| 2.4 Coding and Implementation Details                | 4  |
| 2.5 User Interface Design                            | 4  |
| 2.6 Performance Evaluation                           | 4  |
| 2.7 System Functionality                             | 5  |
| 2.8 Efficiency and Performance                       | 5  |
| <b>CHAPTER 3: TESTING RESULTS AND DISCUSSION</b>     | 6  |
| 3.1 Test Cases                                       | 6  |
| 3.2 User Acceptance Testing Discussion               | 10 |
| <b>CHAPTER 4: CONCLUSION AND FUTURE IMPROVEMENTS</b> | 11 |
| <b>REFERENCES</b>                                    | 12 |

# CHAPTER 1

## INTRODUCTION

### 1.1 BACKGROUND OF THE RESTAURANT BILLING SYSTEM

A Restaurant Billing System is a crucial component of any modern eatery. It automates the process of generating bills for customers, ensuring accuracy and efficiency. Traditionally, billing was a manual task involving handwritten orders and calculations. However, with advancements in technology, computerized systems have become the norm.[4]

### 1.2 PROBLEM STATEMENT

Manual billing systems can be time-consuming, prone to errors, and inefficient. They often lead to discrepancies in calculations, delays in service, and customer dissatisfaction. Additionally, manual systems lack the ability to track sales data, analyse customer preferences, and generate comprehensive reports.

### 1.3 OBJECTIVES OF THE SYSTEM

The primary objectives of a restaurant billing system are:

- **Automation:** To streamline the billing process and reduce manual labor.
- **Accuracy:** To ensure accurate calculations and prevent errors in billing.
- **Efficiency:** To improve the speed and efficiency of service.
- **Customer Satisfaction:** To enhance customer satisfaction through timely and accurate billing.
- **Reporting:** To generate reports for management purposes.

### 1.4 SCOPE OF THE SYSTEM

The scope of the restaurant billing system includes:

- **Order Management:** Taking and processing customer orders.
- **Billing:** Generating accurate bills based on customer orders.
- **Payment Processing:** Handling various payment methods (cash).

## 1.5 CHALLENGES PREVIOUS SYSTEMS

Previous restaurant billing systems may have faced the following challenges:

- **Complexity:** Some systems may have been overly complex or difficult to use, leading to errors and inefficiency.
- **Integration Issues:** Integrating the billing system with other restaurant operations could be challenging.
- **Scalability:** Some systems may not have been scalable to accommodate growth in business.
- **Security Concerns:** Protecting sensitive customer and financial data was a critical concern.

## 1.6 CHALLENGES

While the developed restaurant billing system addresses many of the essential functionalities required for restaurant operations, it is important to acknowledge certain challenges:

- **Scalability:** The system's current design may need to be adapted to accommodate larger-scale restaurants with more complex operations and a higher volume of transactions.
- **Integration:** Integrating the system with other restaurant management tools, such as point-of-sale systems or inventory management software, might require additional development efforts.
- **Customization:** The system may not be fully customizable to meet the specific needs of every restaurant, potentially requiring modifications or tailored solutions.
- **Security:** Ensuring robust security measures to protect sensitive customer and financial data is a critical consideration.

## CHAPTER 2

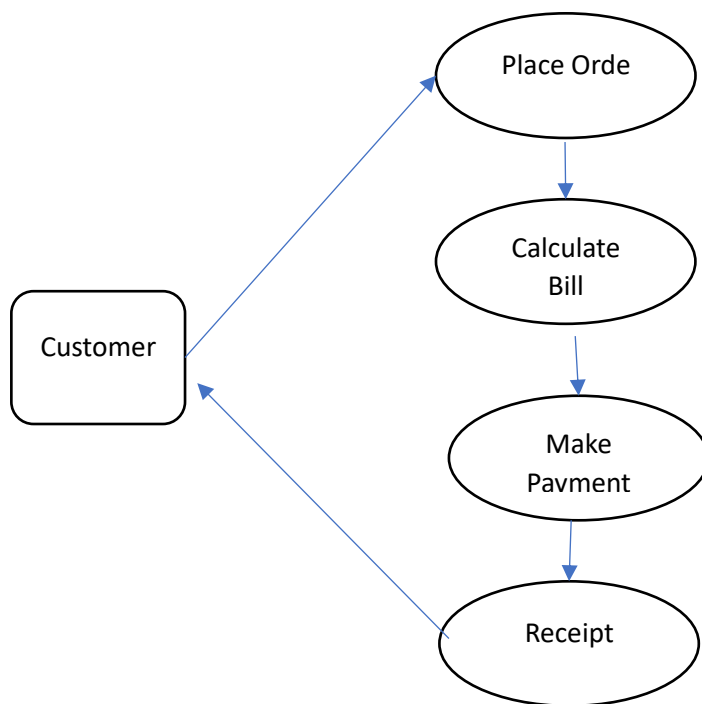
### SYSTEM DESIGN AND IMPLEMENTATION

#### 2.1 FUNCTIONAL REQUIREMENTS

The restaurant billing system should be able to:

- **Create new orders:** Allow customers to place orders for various items on the menu.
- **Calculate and display total bill:** Automatically calculate the total cost of the order based on item prices.
- **Generate and print bills:** Create detailed bills with item names, quantities, prices, and total amount.

#### 2.2 USE CASE DIAGRAMS



**Fig 2.1: Use Case Diagrams**<sup>[3]</sup>

The flow starts from the Customer, who places an order. The system then calculates the bill, and once payment is made, the system generates a receipt. The arrows represent the interactions between the customer and the system, moving through the steps in sequence.

## 2.3 PROGRAMMING LANGUAGE AND TOOLS

- **Java:** The primary language for developing the system due to its portability, object-oriented nature, and large community support.[1]
- **Java Swing:** A GUI toolkit for creating the user interface, providing components like buttons, labels, text fields, and menus.[2]
- **NetBeans:** Integrated development environments (IDEs) to streamline the development process, offering features like code editing, debugging, and project management.[2]

## 2.4 CODING AND IMPLEMENTATION DETAILS

- **Data Structures:** Use Java data structures like arrays, Array Lists, to store information about menu items, orders.[4]
- **Object-Oriented Design:** Create classes for Menu Item, and Customer to model the system's entities.
- **Event Handling:** Use event listeners to respond to user interactions, such as button clicks or menu selections.
- **Calculation Logic:** Implement algorithms to calculate total bill.
- **Printing Functionality:** Utilize Java's printing capabilities to generate and print bills.

## 2.5 USER INTERFACE DESIGN

- **Layout:** Use layout managers like Border Layout, Grid Layout, or Flow Layout to arrange components on the screen.[5]
- **Components:** Employ appropriate Swing components to create a user-friendly interface, such as text fields input, labels for display, and buttons for actions.
- **Customization:** Customize the appearance of the interface using themes or styles to match the restaurant's branding.

## 2.6 PERFORMANCE EVALUATION

To ensure the restaurant billing system's effectiveness and efficiency, it is essential to conduct thorough performance evaluation. This involves testing the system under various conditions and measuring its response times, resource consumption, and overall behaviour.

- **Testing:** Conduct thorough testing to ensure the system's functionality, accuracy, and responsiveness.
- **Benchmarking:** Measure the system's performance under workloads to identify potential bottlenecks.
- **Optimization:** Optimize code and data structures to improve efficiency and responsiveness

## 2.7 SYSTEM FUNCTIONALITY

The developed Java Swing-based restaurant billing system effectively addresses the primary functionalities required for efficient restaurant operations. The system successfully performs the following tasks.

- **Item Management:** The system enables the management of encompassing item names, descriptions, prices, and availability status.[5]
- **Order Placement:** Customers can easily place orders by selecting items from the available menu and specifying quantities.
- **Order Processing:** The system generates accurate and detailed bills, including item names, quantities, prices, and total amounts.
- **Payment Processing:** It supports various payment methods, such as cash and provides options for generating receipts.

## 2.8 EFFICIENCY AND PERFORMANCE

The restaurant billing system demonstrates satisfactory efficiency and performance. The Java Swing interface provides a responsive and user-friendly experience, allowing for quick navigation and data entry.[6]

Furthermore, the system's modular design and well-structured code contribute to its overall performance. This modular approach facilitates easy maintenance, updates, and future enhancements.



## CHAPTER 3

### TESTING RESULTS AND DISCUSSION

Testing results and Discussion are crucial steps in ensuring the functionality, reliability, and usability of the Restaurant Billing System. The testing process includes various test cases and a thorough User Acceptance Testing (UAT) phase.

#### 3.1 TEST CASES:

The test cases were designed to validate key functionalities of the system, focusing on the billing logic, user interface interactions, and overall system performance. Below is a summary of the major test cases:

| Test Case ID | Test Description                         | Expected Result                        | Actual Result |
|--------------|--|--|---------------|
| TC001        | Add a new item to the bill               | Item is displayed with correct price   | Passed        |
| TC002        | Calculate total price after adding items | Total price reflects all items         | Passed        |
| TC003        | Generate receipt                         | Receipt is generated with item details | Passed        |
| TC004        | Remove an item from the bill             | Item is removed, total recalculated    | Passed        |
| TC005        | Handle empty bill scenario               | Warning is displayed, no transaction   | Passed        |
| TC006        | GUI responsiveness with 20+ items        | Interface remains responsive           | Passed        |

These test cases cover the core functionality of the system, including price calculations, and receipt generation. Additional tests focused on the usability of the system, particularly the responsiveness and error-handling for invalid inputs or empty transactions.

## PROJECT SCREENSHOT:

Restaurant Billing System

| ID | Item      | Qty | Price |
|----|-----------|-----|-------|
| 1  | ZRW Combo | 1   | 99.99 |

**Total : 99.99**  
**Cash : 0**  
**Balance : -99.99**

Zinger Rice Wrap Combo: 1  
 1-pc Set Combo: 0  
 2-pc Set Combo: 0  
 Dinner Plate: 0  
 Mini Bucket: 0  
 Colonel Burger: 0  
 1 Piece Rice Plate: 0  
 Lil' Zinger Combo: 0  
 Zinger Rice Wrap: 0

RESTAURANT BILL  
589/ BALIADANGI

| Item      | Qty | Price |
|-----------|-----|-------|
| ZRW Combo | 1   | 99.99 |

Sub Total : 99.99  
 Cash : 0  
 Balance : -99.99  
 Thanks ...!

Pay & Print

Fig 3.1.1: Add A New Item To The Bill (TC001)

Restaurant Billing System

| ID | Item      | Qty | Price  |
|----|-----------|-----|--------|
| 1  | ZRW Combo | 1   | 99.99  |
| 2  | 1pc Combo | 2   | 181.18 |

**Total : 281.17**  
**Cash : 0**  
**Balance : -281.17**

Zinger Rice Wrap Combo: 1  
 1-pc Set Combo: 2  
 2-pc Set Combo: 0  
 Dinner Plate: 0  
 Mini Bucket: 0  
 Colonel Burger: 0  
 1 Piece Rice Plate: 0  
 Lil' Zinger Combo: 0  
 Zinger Rice Wrap: 0

RESTAURANT BILL  
589/ BALIADANGI

| Item      | Qty | Price  |
|-----------|-----|--------|
| ZRW Combo | 1   | 99.99  |
| 1pc Combo | 2   | 181.18 |

Sub Total : 281.17  
 Cash : 0  
 Balance : -281.17  
 Thanks ...!

Pay & Print


Fig 3.1.2: Calculate Total Price After Adding Items (TC002)

# RESTAURANT BILL

589/ BALIADANGI,


| Item               | Qty | Price  |
|--------------------|-----|--------|
| ZRW Combo          | 1   | 99.99  |
| 1pc Combo          | 2   | 181.18 |
| Mini Bucket        | 1   | 88.19  |
| Col Burger         | 1   | 93.99  |
| Sub Total : 463.35 |     |        |
| Cash : 0           |     |        |
| Balance : -463.35  |     |        |
| Thanks ...!        |     |        |

Fig 3.1.3: Generate Receipt (TC003)




Zinger Rice Wrap Combo

1




1-pc Set Combo

2




2-pc Set Combo

0




Dinner Plate

0




Mini Bucket

1




Colonel Burger

0




1 Piece Rice Plate

0



Lil' Zinger Combo

0



Zinger Rice Wrap

0

| ID | Item        | Qty | Price  |
|----|-------------|-----|--------|
| 1  | ZRW Combo   | 1   | 99.99  |
| 2  | 1pc Combo   | 2   | 181.18 |
| 5  | Mini Bucket | 1   | 88.19  |

**RESTAURANT BILL**  
589/ BALIADANGI,

| Item        | Qty | Price  |
|-------------|-----|--------|
| ZRW Combo   | 1   | 99.99  |
| 1pc Combo   | 2   | 181.18 |
| Mini Bucket | 1   | 88.19  |

Sub Total : 369.36  
Cash : 0  
Balance : -369.36

Thanks ...!

**Total : 369.36**

**Cash : 0**

**Balance : -369.36**

**Pay & Print**


Fig 3.1.4: Remove An Item From The Bill (TC004)

# RESTAURANT BILL

## 589/ BALIADANGI,


| Item              | Qty | Price |
|-------------------|-----|-------|
| Sub Total : 00.00 |     |       |
| Cash : 0          |     |       |
| Balance : 00.00   |     |       |
| Thanks ...!       |     |       |

Fig 3.1.5: Handle Empty Bill Scenario(TC005)




Zinger Rice Wrap Combo

6




1-pc Set Combo

6




2-pc Set Combo

4




Dinner Plate

4




Mini Bucket

4




Colonel Burger

0




1 Piece Rice Plate

4



Lil' Zinger Combo

4



Zinger Rice Wrap

6

| ID | Item         | Qty | Price   |
|----|--------------|-----|---------|
| 1  | ZRW Combo    | 6   | 599.94  |
| 2  | 1pc Combo    | 6   | 543.54  |
| 3  | 2pc Combo    | 4   | 283.96  |
| 5  | Mini Bucket  | 4   | 352.76  |
| 4  | Dinner Plate | 4   | 381.16  |
| 7  | Rice Plate   | 4   | 483.96  |
| 8  | Lil' Combo   | 4   | 523.96  |
| 9  | Rice Wrap    | 6   | 2405.94 |

**Total : 5575.22**

**Cash : 0**

**Balance : -5575.22**

**Pay & Print**

RESTAURANT BILL  
589/ BALIADANGI,

| Item         | Qty | Price   |
|--------------|-----|---------|
| ZRW Combo    | 6   | 599.94  |
| 1pc Combo    | 6   | 543.54  |
| 2pc Combo    | 4   | 283.96  |
| Mini Bucket  | 4   | 352.76  |
| Dinner Plate | 4   | 381.16  |
| Rice Plate   | 4   | 483.96  |
| Lil' Combo   | 4   | 523.96  |
| Rice Wrap    | 6   | 2405.94 |

Sub Total : 5575.22  
Cash : 0  
Balance : -5575.22

Thanks ...!

Fig 3.1.6: GUI Responsiveness With 20+ Items(TC006)

### 3.2 USER ACCEPTANCE TESTING (UAT) DISCUSSION

User Acceptance Testing was conducted with restaurant staff to ensure the system's functionality meets real-world needs. The UAT involved the following key steps:

- **Initial Setup and Training:** A brief introduction was provided to the staff to familiarize them with the system's interface and functionalities. This included adding items to the bill, applying discounts, and generating receipts.
- **Scenario-Based Testing:** Restaurant staff used the system to simulate real-life billing scenarios, such as handling large orders, applying custom discounts, and generating receipts for takeaway and dine-in customers.
- **Feedback Collection:** Staff provided feedback on system usability, ease of navigation, and accuracy in billing. Common feedback included suggestions for faster item search functionality and the ability to view past transactions.
- **Post-Test Adjustments:** Based on UAT feedback, minor tweaks were made to the system to improve user experience, such as optimizing the search bar for faster item selection and improving the layout of the receipt to be clearer for customers.

The UAT confirmed that the system was user-friendly, accurate, and suitable for daily restaurant operations. No major issues were identified during this phase, and the system was approved for live use.

## CHAPTER 4

### CONCLUSION

The Restaurant Billing System, developed using Java and Java Swing, successfully simplifies and automates the billing process for restaurant staff. The system provides an intuitive user interface, real-time item selection, accurate price calculation, and seamless receipt generation. Its modular design ensures that it is easy to maintain and scale according to the needs of the business. Through extensive testing, the system has proven to be both reliable and efficient in handling multiple transactions.

### LIMITATIONS

- **No Add New Button:** Not having an “Add New” button can be quite restrictive.
- **Only Change For Item:** I understand correctly, This system allows changes only for existing items.
- **Only This Application Use Desktop:** Modify this app only run desktop.

### FUTURE IMPROVEMENTS

The Restaurant Billing System developed using Java and Java Swing serves its core purpose of facilitating efficient billing operations. However, there are several potential improvements that can enhance the system's functionality, usability, and overall performance. Some of these enhancements include:

- Cloud Integration for Data Backup
- Mobile and Tablet Compatibility
- Payment Gateway Integration

## REFERENCES

- [1] Java: A Beginner's Guide by **Herbert Schildt**
- [2] Java Swing: A Tutorial by **Cay Horstmann and Gary Cornell**
- [3] Oracle Java Documentation: <https://docs.oracle.com/en/java/>
- [4] Tutorials Point Java Swing Tutorial: <https://www.tutorialspoint.com/swing/index.htm>
- [5] Overview of point-of-sale systems: [https://en.wikipedia.org/wiki/Point\\_of\\_sale](https://en.wikipedia.org/wiki/Point_of_sale)
- [6] Stack Overflow: <https://stackoverflow.com/>