**BILLING SYSTEM**

# A MINI PROJECT REPORT

***Submitted by***

***Kowsi K [22CSEB12]***

***Lohitha R [22CSEB13]***

***Naga Mrithula BM [22CSEB18]***

***Yogitha A [22CSEB33]***

***in partial fulfilment for the award of the degree***

**of**

**BACHELOR OF ENGINEERING**

**IN**

# COMPUTER SCIENCE AND ENGINEERING



**VELAMMAL COLLEGE OF ENGINEERING AND TECHNOLOGY**

**Autonomous**

**NOVEMBER 2023**

**BONAFIDE CERTIFICATE**

Certified that this Mini Project report ‘**Billing System**’ is the bonafide work of Kowsi K (22CSEB12), Lohitha R (22CSEB13), Naga Mrithula BM (22CSEB18), Yogitha A (22CSEB33) of **III** **Semester B.E Computer Science and Engineering** who carried out the project work under my supervision.

SIGNATURE SIGNATURE

Dr.R.DEEPALAKSHMI Ms.J.SHANTHALAKSHMI REVATHY

HEAD OF THE DEPARTMENT ASSISTANT PROFESSOR

COMPUTER SCIENCE & ENGG COMPUTER SCIENCE & ENGG

VELAMMAL COLLEGE OF ENGG & TECH VELAMMAL COLLEGE OF ENGG & TECH

MADURAI. MADURAI.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGE NO** |
| 1. | ABSTRACT | 4 |
| 2. | INTRODUCTION | 5 |
| 3. | MODULE DESCRIPTION | 6 |
| 4. | SOURCE CODE | 8 |
| 5. | APPENDICES (Output Screen) | 12 |
| 6. | CONCLUSION | 13 |
| 7. | REFERENCES | 13 |

**ABSTRACT**

In a world where businesses are growing, and we need things to be done fast and well, using digital tools, especially in billing, is super important. Going digital isn’t just about making business easier; it’s a big change that helps us save time and work better. Automated systems, especially in billing, are great because they make things easier, save time, and reduce mistakes. We’re done with the old way of manually doing billing and keeping records; businesses now like a digital world where billing info is right there when we need it, just a few clicks away. This change matches the modern need for getting business insights instantly and whenever we want.

In this context, the Billing System emerges as a simple yet impactful Java-based application, strategically designed to aid businesses in efficiently managing their billing processes. Created with the intent of simplifying billing tasks for small to medium-sized enterprises, this Java-based system offers a user-friendly interface dedicated to the creation and management of invoices. The project places emphasis on automating fundamental billing functions while ensuring accessibility for users with limited technical expertise. Boasting features such as real-time updates and a transparent transaction display, it represents an intuitive and organized approach to automated billing processes, harmonizing with the overarching trend of digital transformation in various aspects of our lives.

**INTRODUCTION**

In the ever-evolving landscape of contemporary business, the role of effective billing systems has transcended mere financial transactions. Even small enterprises require a reliable billing system to streamline financial operations. Our Billing System addresses this need by offering a lightweight, Java-based application that facilitates the creation, management, and tracking of invoices.

The methodology of the Billing System project deploys an online approach to streamline transaction management for businesses. Recognizing the limitations of traditional methods involving manual processes and paperwork, the system employs a digital platform to reduce paper usage and save time. Through the implementation of online billing, businesses can simplify their operations, minimize errors, and enhance the overall efficiency of the billing process. Users, in this case, can register online, input basic details, update information as needed, and interact with the system seamlessly. The project is designed to cater to the needs of businesses seeking a modern and efficient billing solution.

The Billing System project Is driven by the following objectives:

* **Efficient Transaction Management:** The primary goal is to design a system that streamlines and enhances the efficiency of transaction management for businesses. By leveraging digital solutions, the project aims to simplify the complex process of itemizing transactions, calculating total amounts, and generating detailed invoices.
* **Reduction of Human Effort:** A key objective is to minimize manual labor, efforts, and potential errors associated with traditional billing methods. The system seeks to automate repetitive tasks, allowing businesses to focus on core operations rather than extensive manual record-keeping.
* **Real-time Updates and Transparency:** The project aims to provide real-time updates on ongoing transactions. This feature ensures transparency in the billing process, allowing users to have immediate feedback on the evolving bill and promoting accountability.
* **User-friendly Interface:** With a user-centric approach, the Billing System focuses on creating an intuitive graphical user interface (GUI) using Java Swing. This user-friendly design facilitates easy interaction, enabling users to add items, specify quantities, and input prices seamlessly.
* **Adaptability and Reset Functionality:** Acknowledging the iterative nature of transactions, the system incorporates a ‘Reset Bill’ feature. This allows users to start afresh without restarting the entire application, providing adaptability for multiple transactions within a single session.
* **Total Amount Calculation:** The project emphasizes accurate calculation of the total amount for added items, promoting precision in financial transactions. This ensures that businesses can rely on the system for accurate and error-free billing.

**MODULE DESCRIPTION**

* **ITEM MANAGEMENT MODULE:**

**Components:** *addItem()*, *generateBill()*, *resetBill()*

The Item Management Module is the backbone of the Billing System, orchestrating essential functions for seamless item processing. Within this module, the *addItem()*, *generateBill()*, and *resetBill()* functions collectively govern the intricate dance of adding items, calculating total amounts, and updating the bill display in real-time. When a user inputs item details, such as name, quantity, and price, the system dynamically calculates the total amount, offering users immediate feedback on their ongoing transaction. The module ensures accuracy and transparency, contributing significantly to the user experience by providing a robust foundation for managing transactions efficiently.

* **USER INTERFACE MODULE:**

**Components:** *JTextField*, *JButton*, *JTextArea*, *JPanel*, *JScrollPane*, *SwingUtilities*

At the intersection of aesthetics and functionality lies the User Interface Module, a pivotal component in the Billing System. Leveraging Java Swing components, this module crafts an engaging and user-friendly interface. Elements such as *JTextField*, *JButton*, *JTextArea*, *JPanel*, and *JScrollPane* come together to create a visually appealing platform. The thoughtful incorporation of these components ensures a cohesive and intuitive design, enhancing user interaction. Through a carefully designed graphical layout, this module fosters a positive user experience, making item management and transaction processing both accessible and visually satisfying.

* **TRANSACTION RECORDING MODULE:**

**Components:** *itemDetails*, *totalItemAmount*

In the intricate dance of capturing and processing user input, the Transaction Recording Module takes center stage. Within its realm, the *itemDetails* variable holds the formatted strings for each item, creating a detailed record of transaction specifics. Simultaneously, the *totalItemAmount* function calculates the cumulative amount for a transaction, contributing to the precision and efficiency of the billing process. This module becomes the custodian of transaction details, ensuring accuracy and immediacy in the reflection of ongoing transactions in the bill display.

* **USER INTERACTION MODULE:**

**Components:** *ActionListener*

The User Interaction Module is the responsive core of the Billing System, interpreting user actions and orchestrating system behaviour. By implementing the ActionListener interface, this module governs the behaviour of critical components such as the "Add Item" and "Generate Bill" buttons. The user's engagement triggers functions like *addItem()* and *generateBill()*, fostering an interactive and dynamic user experience. This module, through its responsiveness, adds a layer of sophistication to the system, making user interactions fluid and intuitive.

* **INITIALIZATION AND MAIN MODULE:**

**Components:** *main()*, *BillingSystem()*

As the curtain rises on the Billing System, the Initialization and Main Module takes the spotlight, setting the stage for the entire application. The *main()* method serves as the conductor, orchestrating the initiation of the system on the event dispatch thread using *SwingUtilities.invokeLater()*. It is within the *BillingSystem()* constructor that the system is initialized, configuring GUI components, defining the structure of the application window, and ensuring its visibility. This module, in its foundational role, provides the structure and order necessary for the Billing System to come to life, ensuring a seamless and visually coherent initiation for users.

These modules collectively form the foundational elements of our Billing System, providing the necessary structure and functionality for item management, user interaction, and system initialization.

**SOURCE CODE**

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class BillingSystem extends JFrame {

JTextField itemNameField, quantityField, priceField;

JTextArea billArea;

double totalAmount = 0;

public BillingSystem() {

setTitle("Billing System");

setSize(400, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

itemNameField = new JTextField(20);

quantityField = new JTextField(10);

priceField = new JTextField(10);

billArea = new JTextArea(10, 30);

billArea.setEditable(false);

JButton addItemButton = new JButton("Add Item");

JButton generateBillButton = new JButton("Generate Bill");

addItemButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

addItem();

}

});

generateBillButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

generateBill();

}

});

JPanel panel = new JPanel();

panel.add(new JLabel("Item Name: "));

panel.add(itemNameField);

panel.add(new JLabel("Quantity: "));

panel.add(quantityField);

panel.add(new JLabel("Price: "));

panel.add(priceField);

panel.add(addItemButton);

panel.add(generateBillButton);

JScrollPane scrollPane = new JScrollPane(billArea);

add(panel, "North");

add(scrollPane);

setVisible(true);

}

private void addItem() {

String itemName = itemNameField.getText();

int quantity = Integer.parseInt(quantityField.getText());

double price = Double.parseDouble(priceField.getText());

double totalItemAmount = quantity \* price;

totalAmount += totalItemAmount;

String itemDetails = String.format("%s - %d x %.2f = %.2f\n", itemName, quantity, price, totalItemAmount);

billArea.append(itemDetails);

itemNameField.setText("");

quantityField.setText("");

priceField.setText("");

}

private void generateBill() {

JOptionPane.showMessageDialog(this, String.format("Total Amount: %.2f", totalAmount));

resetBill();

}

private void resetBill() {

totalAmount = 0;

billArea.setText("");

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

new BillingSystem();

}

});

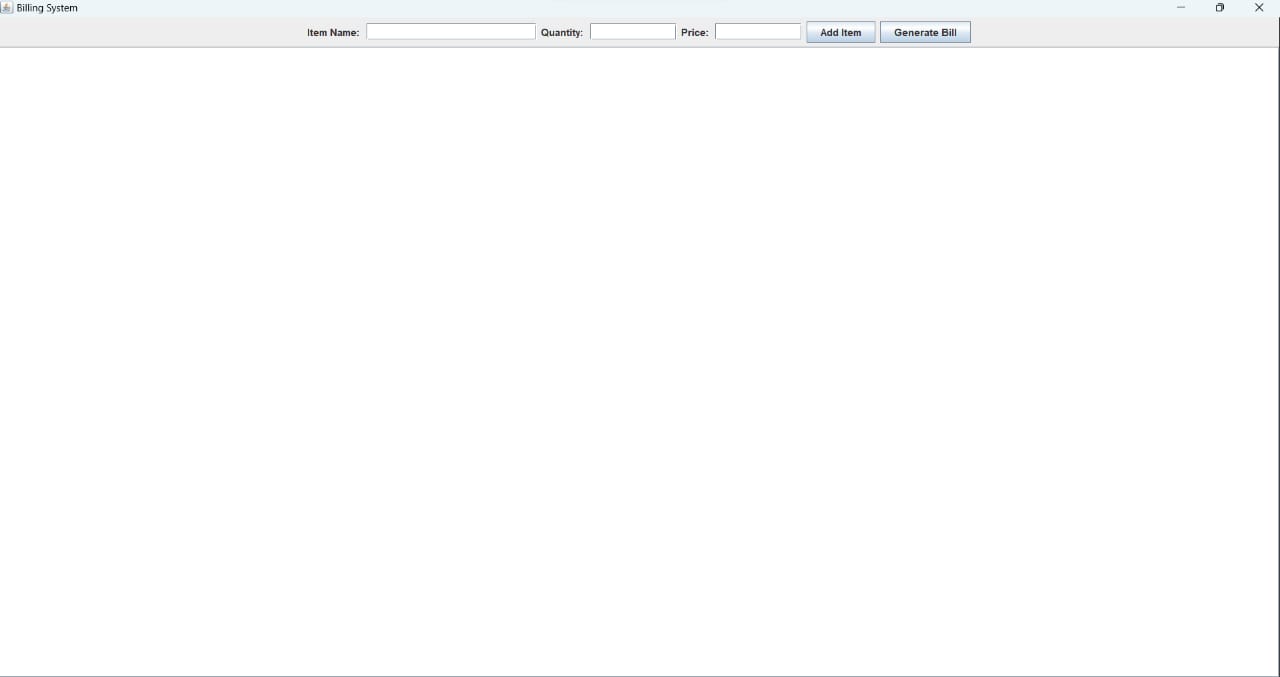
}

}

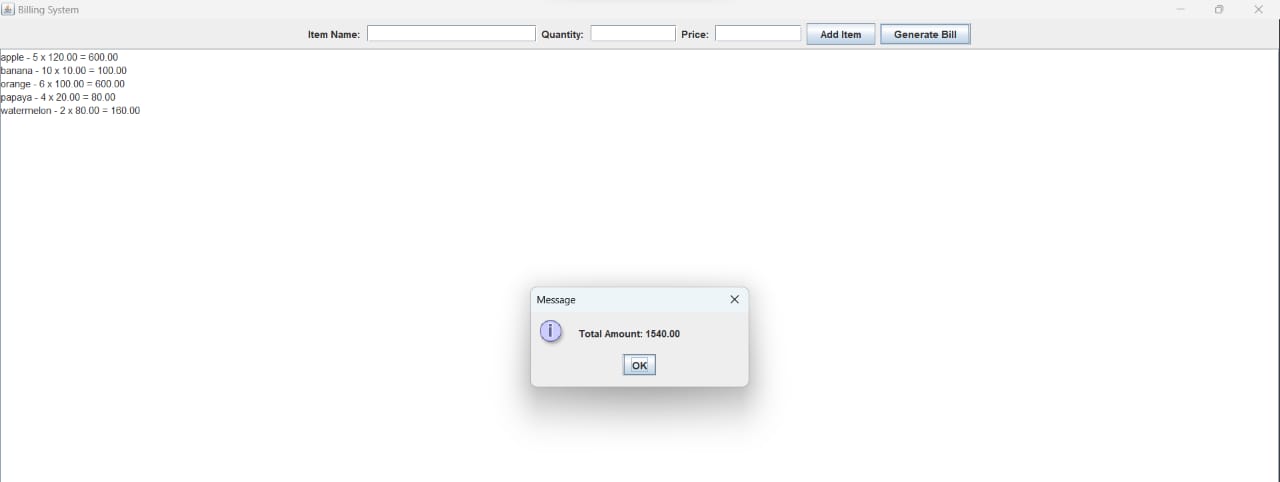
**APPENDIX**

**(Output Screen)**

**FRAME:**



**GENERATION OF BILL:**



**CONCLUSION**

In conclusion, the Billing System project serves as a practical and easy-to-use answer for businesses in search of a streamlined way to handle transactions. Through the integration of a thoughtfully crafted user interface and modules that support adding items, receiving real-time bill updates, and generating bills, the system effectively tackles the main issues linked with billing procedures. This project showcases the capability of Java applications in simplifying financial processes and promoting precision in handling transactions. As businesses increasingly adopt technology to enhance their operations, the Billing System offers a valuable resource for refining the billing workflow.

**REFERENCES**

**1.** [Kowsi2410/oops\_miniproject (github.com)](https://github.com/Kowsi2410/oops_miniproject)

**2.** [Why are Java project file hierarchies so deeply nested? - Quora](https://www.quora.com/Why-are-Java-project-file-hierarchies-so-deeply-nested)

**3.** [What is Framework in Java - Javatpoint](https://www.javatpoint.com/what-is-framework-in-java?shem=ssusba)

**4.** [17 Popular Java Frameworks for 2023: Pros, cons, and more · Raygun Blog](https://raygun.com/blog/popular-java-frameworks/)

**5.** [ChatGPT (openai.com)](https://chat.openai.com/auth/login)