



# **RAJALAKSHMI ENGINEERING COLLEGE**

**An AUTONOMOUS Institution  
Affiliated to ANNA UNIVERSITY, Chennai**

## **SMART BILLING SOFTWARE**

Submitted by:

**SANJAY B (221801045)  
SOORYA B (221801051)  
THOFIQ GANI M (221801057)**

## **AD19541 SOFTWARE ENGINEERING METHODOLOGY**

**Department of Artificial Intelligence and Data Science**

**Rajalakshmi Engineering College, Thandalam**

# **BONAFIDE CERTIFICATE**

Certified that this project report “**SMART BILLING SOFTWARE**” is the bonafide work of “**SANJAY B (221801045), SOORYA B (221801051), THOFIQ GANI M (221801057)** ”who carried out the project work under my supervision.

**Submitted for the Practical Examination held on** \_\_\_\_\_

**SIGNATURE**

**Dr. GNANASEKAR J M**  
**Head of the Department, Artificial intelligence**  
**and Data Science,Rajalakshmi Engineering**  
**College (Autonomous),Chennai-602105**

**SIGNATURE**

**Dr. MANORANJINI J**  
**Associate Professor, Artificial Intelligence and**  
**Data Science, Rajalakshmi Engineering College**  
**(Autonomous), Thandalam, Chennai-602105**

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

# **TABLE OF CONTENT**

<b>S.NO</b>	<b>CHAPTER</b>	<b>PAGE NUMBER</b>
<b>1.</b>	<b>INTRODUCTION</b>	
1.1	GENERAL	1
1.2	NEED FOR THE STUDY	2
1.3	OBJECTIVES OF THE STUDY	3
1.4	OVERVIEW OF THE PROJECT	4
<b>2.</b>	<b>REVIEW OF LITERATURE</b>	
2.1	INTRODUCTION	8
2.2	LITERATURE REVIEW	9
<b>3.</b>	<b>SYSTEM OVERVIEW</b>	
3.1	EXISTING SYSTEM	12
3.2	PROPOSED SYSTEM	13
3.3	FEASIBILITY STUDY	14
<b>4.</b>	<b>SYSTEM REQUIREMENTS</b>	
4.1	SOFTWARE REQUIREMENTS	16
4.2	HARDWARE REQUIREMENTS	17
<b>5.</b>	<b>SYSTEM DESIGN</b>	
5.1	SYSTEM ARCHITECTURE	18
5.2	MODULES DESCRIPTION	19

6.	<b>SOFTWARE TESTING</b>	
6.1	FUNCTION TESTING	27
6.2	TESTING OUTPUT	29
7.	<b>RESULTS AND DISCUSSION</b>	
7.1	RESULTS	32
7.2	OUTPUT	32
7.3	DISCUSSION	36
8.	<b>CONCLUSION AND FUTURE SCOPE</b>	
8.1	CONCLUSION	38
8.2	FUTURE SCOPE	38
9.	<b>REFERENCES</b>	40

## TABLE OF FIGURES

<b>S.N O</b>	<b>FIGURE</b>	<b>PAGE NUMBER</b>
1	SYSTEM FLOW DIAGRAM	18
2	CUSTOMER DETAILS MODULE	20
3	PRODUCT DETAILS MODULE	21
4	CART MANAGEMENT MODULE	23
5	PAYMENT SELECTION MODULE	24
6	BILL GENERATION MODULE	25
7	EXCEL SAVING MODULE	26
8	CART OPERATIONS	30
9	EXCEL SAVING FUNCTIONALITY	30
10	IMAGE IMPORT FUNCTIONALITY	30
11	TOTAL PRICE CALCULATION	31
12	USER INTERFACE	32
13	DETAILS ARE FILLED BY USER	33
14	QR-CODE IMPORT FROM COMPUTER	33
15	QR-CODE IS DISPLAYED IN BILLING SOFTWARE	34
16	PAYMENT OPTIONS ASKED FOR USER	34
17	ALL THE DETAILS STORED IN EXCEL	35

18	DETAILS ARE DISPLAYED IN EXCEL FILE	35
19	PRINT THE BILL UISNG PRINT BUTTON	36

## **ABSTRACT**

The purpose of this project is to implement a fast billing system with QR code payments and also data processing. Customers are also able to go through a purchase process successfully; they only have to scan a QR code which sends them to a payment page hence making it rather easy to pay and satisfying to customers. All the information about the transaction details such as the customer's information, the details of the product(s) sold, payment information is recorded in an Excel file for record-keeping and easy retrieval. Furthermore, the system creates periodic reports on financials as well as the frequency of method of payment and balances owed to offer indispensable information on future sales. To minimize reliance on the expensive credit sales, organizations rely on automated notifications of unpaid invoices so that follow-ups can be made on time to encourage early payments. This work involves integration of sophisticated payment channel and auto population of data house to address the challenges of billing and ineffective tracking of financial status of businesses. To reduce dependency on costly credit sales, the system incorporates automated reminders for unpaid invoices, ensuring timely follow-ups to encourage early payments. By integrating advanced payment channels and enabling automated data population, the project addresses challenges like billing inefficiencies and ineffective financial tracking.

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 GENERAL**

The purpose of this project is to develop a fast and efficient billing system designed to streamline payment processes and improve customer satisfaction. Leveraging modern technology such as QR code-based payments, the system ensures a seamless and hassle-free checkout experience. Customers simply scan a QR code to access a secure payment page, allowing them to complete transactions quickly and conveniently. This innovation eliminates the need for traditional payment methods, enhancing the overall user experience and encouraging repeat business.

All essential transaction data, including customer details, product information, payment status, and timestamps, are automatically recorded in an Excel file. This ensures accurate record-keeping and facilitates quick retrieval for audits, financial analysis, or customer service purposes. To provide deeper insights into business performance, the system generates periodic reports. These reports include metrics such as sales trends, payment method preferences, outstanding balances, and profitability analyses, equipping businesses with actionable data to make informed decisions.

To address the challenges of managing credit sales, the billing system integrates automated notifications for unpaid invoices. This feature allows organizations to follow up with customers promptly, reducing delays in payments and improving cash flow. By minimizing reliance on credit and promoting timely settlements, businesses can achieve greater financial stability.

Additionally, the system includes robust data processing capabilities, automating the population of transaction records and eliminating manual errors. The integration of sophisticated payment channels ensures compatibility with various platforms, catering to a diverse customer base. These features collectively address common



challenges in billing and financial tracking, such as inefficiency, missed follow-ups, and poor visibility into financial health.

In summary, this project delivers an all-in-one solution for businesses looking to modernize their billing processes. By combining advanced payment technology, comprehensive data recording, and insightful reporting, it empowers organizations to improve operational efficiency, boost customer satisfaction, and drive long-term growth.

## **1.2 NEED FOR THE STUDY**

In today's fast-paced and highly competitive business environment, the ability to process transactions quickly and accurately is essential for maintaining customer satisfaction and operational efficiency. Traditional billing systems often rely on manual data entry or outdated technologies, which can lead to errors, inefficiencies, and delays in payment processing. These issues not only frustrate customers but also hinder businesses from maintaining a healthy cash flow and achieving their financial goals.

The growing adoption of digital payment methods and customer expectations for seamless transactions necessitate a shift towards smarter, more integrated billing solutions. QR code technology offers a convenient and secure way for customers to make payments while reducing the complexities of cash or card-based transactions. Businesses that implement such modern solutions stand to enhance customer satisfaction and foster loyalty, giving them a competitive edge.

Another critical factor driving the need for this study is the increasing demand for accurate and automated financial tracking. Many businesses struggle with maintaining comprehensive records of transactions, which affects their ability to analyze performance, forecast sales, and make data-driven decisions. Integrating automated data population and reporting tools into a billing system eliminates manual errors, saves time, and provides actionable insights into key financial metrics.

Moreover, delayed payments and outstanding invoices are persistent challenges that businesses face, particularly in credit sales. These can negatively impact cash flow and overall financial health. By incorporating features such as automated reminders for unpaid invoices, businesses can improve follow-up efficiency and encourage timely payments, reducing dependence on expensive credit options.

This study also addresses the need for businesses to adapt to evolving payment preferences, offering support for diverse payment channels and ensuring compatibility with modern financial systems. A robust billing system with these capabilities not only simplifies the payment process but also provides a foundation for long-term financial sustainability and growth.

In conclusion, the need for this study arises from the demand for a streamlined, technology-driven billing solution that enhances transaction efficiency, strengthens financial management, and ensures customer satisfaction. By bridging the gap between traditional and digital payment systems, this project delivers a practical and scalable solution for modern businesses.

### **1.3 OBJECTIVES OF THE STUDY**

The primary objective of this study is to design and implement an efficient billing system that leverages modern technologies to streamline payment processes and enhance business operations. The specific objectives include:

#### **Develop a QR Code-Based Payment System:**

To create a secure and user-friendly QR code payment mechanism that enables customers to complete transactions quickly and conveniently, minimizing wait times and improving their experience.

#### **Automate Transaction Data Management:**

To record and organize transaction details, including customer information, product details, and payment summaries, in an Excel file for accurate record-keeping and easy retrieval.

**Generate Periodic Financial Reports:**

To design a system that can generate comprehensive reports, providing insights into sales trends, payment preferences, outstanding balances, and overall financial performance.

**Enhance Payment Tracking and Follow-Ups:**

To integrate automated notifications for unpaid invoices, enabling businesses to follow up with customers efficiently and encouraging timely payments to improve cash flow.

**Ensure Compatibility with Modern Payment Channels:**

To develop a system compatible with various payment methods and platforms, catering to the diverse needs of customers and fostering adaptability to future technological advancements.

**Improve Operational Efficiency and Reduce Errors:**

To minimize manual interventions in billing and data entry, ensuring accurate and error-free processing of transaction data and enhancing the overall operational efficiency of the system.

**Support Decision-Making Through Data Analytics:**

To enable businesses to utilize transaction data for analytics, forecasting sales, and making data-driven decisions that promote financial stability and growth.

By achieving these objectives, the study aims to provide a robust, scalable, and customer-centric billing solution that addresses the current challenges in payment processing and financial tracking while ensuring long-term business success.

**1.4 OVERVIEW OF THE PROJECT**

This project focuses on developing a fast, efficient, and user-friendly billing system designed to meet the needs of modern businesses and their customers. The system leverages cutting-edge technologies, such as QR code-based payments, to simplify

the payment process, enhance customer satisfaction, and streamline business operations.

The core functionality of the project is centered around enabling seamless transactions where customers can scan a QR code to access a secure payment page. This eliminates the need for traditional payment methods, reduces manual intervention, and ensures a smooth checkout experience. Additionally, the system records all transaction details—customer information, product data, and payment specifics—in an Excel file for accurate and organized record-keeping.

To assist businesses in managing their financial data, the project includes a feature to generate periodic financial reports. These reports provide insights into key metrics, such as sales trends, payment methods, outstanding balances, and overdue invoices. Such data empowers businesses to make informed decisions and improve their operational strategies.

Another significant aspect of the project is the integration of automated notifications for unpaid invoices. This feature helps businesses follow up with customers promptly, encouraging timely payments and reducing reliance on credit sales. By addressing payment delays, the system ensures a healthier cash flow for businesses.

The system is also equipped with advanced data processing capabilities, ensuring error-free automation of transactional data and minimizing manual workload. Its compatibility with multiple payment channels and adaptability to future technologies make it a robust solution for modern billing challenges.

Overall, this project is designed to offer businesses a comprehensive and scalable billing solution. It not only addresses inefficiencies in traditional billing systems but also equips organizations with tools for financial tracking, data-driven decision-making, and enhanced customer service, fostering long-term success and growth.

## **WORKFLOW**

The workflow involves entering customer and product details, generating a bill with a QR code, processing payment by scanning the code or choosing another method, and

saving transaction details in an Excel file for easy record-keeping.

## **Data Input**

- **Computer:** The system starts with a computer interface where customer and product details are entered.
- **Customer Details:** Information about the customer, such as name, contact information, and address, is input into the system.
- **Product Details:** Details of the product(s) being purchased, including quantity, price, and item specifics, are also entered.

## **· Bill Generation**

- **Bill Details:** The system compiles customer and product details to generate a bill.
- **Generated Bill:** The final bill is created, detailing the items purchased, their costs, and the total amount due.

## **· QR Code Creation**

- **QR Code:** A QR code is generated for the bill, enabling customers to pay by scanning it with their mobile device.

## **· Payment Processing**

- **Payment Method:** The customer chooses their preferred payment method, such as cash or online payment.
- **Online Payment:** If the customer selects online payment, they scan the QR code, which redirects them to a secure payment page to complete the transaction.

## **· Transaction Recording**

- **Purchase Details:** All details of the transaction, including the customer's information, items purchased, and payment method, are recorded.

- **Excel File (XLS):** The transaction data is stored in an Excel file, facilitating organized record-keeping for future retrieval and analysis.

## **CHAPTER 2**

### **REVIEW OF LITERATURE**

#### **2.1 INTRODUCTION**

The last few decades have seen significant advancements in billing systems, driven by rapid technological progress that has transformed how transactions are processed, managed, and secured. Technology has enabled the development of efficient, automated billing systems that not only streamline payment processing but also enhance the customer experience and improve operational efficiency. Research highlights the increasing importance of integrating digital payment platforms, particularly QR code payments, which simplify the billing process and offer a fast, user-friendly payment option. Studies show that QR codes enhance security and convenience, allowing customers to pay by simply scanning a code, reducing the reliance on traditional cash or card transactions.

The literature on billing systems also emphasizes the importance of automated data processing and record-keeping. Automation helps to ensure that transaction details, such as customer information, product data, and payment records, are accurately recorded and organized in digital storage, commonly Excel or database systems. This minimizes human error, facilitates quick retrieval of data, and supports data analysis efforts. The use of structured storage systems is particularly significant in helping businesses manage large volumes of transaction data efficiently. In addition, periodic financial reports generated from automated data provide insights into business performance, trends in customer behavior, and payment patterns, allowing organizations to make data-driven decisions.

Another growing area of focus in billing literature is the importance of payment tracking and follow-ups, especially in managing credit sales. Research suggests that automated reminders for unpaid invoices are essential for improving cash flow, as they encourage customers to make timely payments and reduce the strain on a business's finances. Automated notifications serve as digital records, contributing to accountability and establishing trust with clients. Studies indicate that these reminders

can significantly reduce the amount of time spent on manual follow-ups, freeing staff to focus on other tasks.

Data security is another critical aspect discussed in the literature, as billing systems often handle sensitive customer information, including payment details. With increasing concerns over data privacy, studies recommend implementing data encryption and secure storage methods to protect personal and financial information. MySQL databases are frequently highlighted for their scalability and robust encryption capabilities, making them a preferred choice for billing systems that require secure data handling.

In summary, the existing literature provides a foundation for developing a billing system that combines QR code payments, automated data processing, and secure storage practices. This project builds upon these findings with the goal of addressing challenges in efficiency, security, and user experience within billing systems, delivering a comprehensive solution that meets the needs of modern businesses and their customers.

## 2.1 LITERATURE REVIEW

S. No	Author Name	Paper Title	Description	Journal	Year
1	Sharma, R., Verma, S., & Singh, P..	Automated Billing System with QR Code Integration	Design and implementation of an automated billing system with QR code payments for fast and secure transactions.	International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)	2021



2	Kumar, A., & Jain, M.	Data Automation in Billing Systems for Small Businesses	Focus on automated data entry and storage in billing systems to reduce errors and improve data management efficiency.	Journal of Information Technology and Engineering (JITE)	2020
3	Patel, N., Desai, R., & Shah, T.	Enhancing Customer Experience with Automated Billing Notification	Study on the impact of automated payment reminders and notifications on customer satisfaction and cash flow.	International Journal of Innovations in Engineering and Technology (IJIET)	2019
4	Choudhary, L., & Gupta, P.	Data Security and Encryption in Digital Billing Systems	Analysis of data privacy techniques, including encryption and MySQL database security, in billing System.	Journal of Modern Data Security and Management (JMDSM)	2023

**Table no 1 Literature Review**

The literature review table provides Recent studies have explored various advancements in billing systems, focusing on automation, data security, and customer experience. Sharma, Verma, and Singh (2021) presented a billing system integrated with QR code technology, emphasizing the benefits of fast, secure transactions, and customer convenience. Kumar and Jain (2020) discussed the importance of automated

data entry and storage, highlighting how automation reduces errors and improves efficiency, particularly for small businesses. Patel, Desai, and Shah (2019) investigated the impact of automated billing notifications on customer satisfaction and business cash flow, finding that reminders for unpaid invoices lead to prompt payments and increased customer trust. Choudhary and Gupta (2023) focused on data security in billing systems, analyzing techniques such as encryption and MySQL database usage to protect sensitive customer data. Together, these studies provide a comprehensive foundation for developing an efficient and secure billing system that leverages automation, secure data handling, and customer-centric features.

## CHAPTER 3

### SYSTEM OVERVIEW

#### 3.1 EXISTING SYSTEM

The **Smart Billing Software** in many businesses are designed to handle essential transactions, record data, and provide a structured payment process. However, these systems vary in functionality and technological sophistication, with some still relying on manual data entry and limited automation, which can lead to inefficiencies and a higher likelihood of human error. Traditional billing processes often involve creating invoices manually, which can be time-consuming and prone to mistakes, especially when handling a large volume of transactions.

With the rise of digital technology, many businesses have shifted to automated billing systems that integrate basic digital tools for record-keeping and payment tracking. Such systems may utilize spreadsheets or basic databases to store customer and transaction data. In most cases, these systems allow businesses to store transaction details, calculate totals, and track outstanding balances, which has improved accuracy and retrieval of data.

Digital payment options, like credit card processing, have also become more common, but not all systems offer the level of flexibility and convenience that modern customers expect. Many billing systems lack integrated QR code functionality, which is increasingly popular due to its ease of use and customer-friendly nature. Additionally, many existing systems do not include automated reminders for unpaid invoices, requiring businesses to manually follow up with customers for payments, which can be time-consuming and inconsistent.

Data security is another challenge in current billing systems, particularly for businesses that handle sensitive financial and personal information. Many existing systems lack robust encryption and data protection measures, making them vulnerable to breaches and data theft.

### **3.2 PROPOSED SYSTEM**

The proposed billing system is designed to address existing limitations by incorporating advanced features that enhance efficiency, accuracy, customer convenience, and data security. It will automate data entry and storage, reducing manual errors and ensuring that transaction details are consistently recorded in an organized digital database, such as Excel or MySQL. The system will also integrate QR code technology for payments, allowing customers to easily scan a unique QR code on their bill and complete transactions quickly and securely, minimizing the need for traditional cash or card methods. To further streamline operations, automated notifications will remind customers of unpaid invoices via email or SMS, improving cash flow and reducing manual follow-ups. Data security will be a top priority, with encryption safeguarding sensitive customer information, particularly payment details, and ensuring that data remains protected against unauthorized access. Additionally, the system will include a reporting and analysis feature, automatically generating periodic reports on sales, payment trends, and outstanding balances, allowing businesses to make data-driven decisions and optimize performance. These features collectively aim to provide a more efficient, secure, and user-friendly billing process.

Furthermore, the system will be designed with scalability in mind, allowing it to grow alongside the business. As transaction volumes increase, the system's infrastructure, particularly the database, will be able to handle larger datasets without compromising performance. This will ensure that businesses can continue to rely on the billing system even as their operations expand. Additionally, the system will be customizable, allowing businesses to tailor the features to suit their specific needs. Whether it's adjusting invoice templates, modifying payment gateways, or configuring the notification system, the flexibility of the billing system ensures it aligns with the unique requirements of different industries. With a focus on automation, security, and ease of use, the system will ultimately empower businesses to improve operational efficiency, enhance customer satisfaction, and achieve better financial management.

### **3.3 FEASIBILITY STUDY**

#### **Technical Feasibility**

The proposed billing system is technically feasible using available technologies. Python, along with its libraries such as pandas for data handling, PyMySQL or SQLAlchemy for database interaction, qrcode for QR code generation, and smtplib for automated email notifications, are well-suited for building the system. These tools are widely supported, reliable, and capable of handling the requirements of the system. The integration of QR codes for payment processing is easily achievable, as Python's qrcode library can generate dynamic QR codes. Data encryption for securing customer information can be implemented using libraries like cryptography to ensure compliance with privacy regulations.

#### **Operational Feasibility**

Operationally, the proposed billing system is highly feasible. The automation of tasks such as data entry, invoice generation, and payment reminders will significantly reduce manual work, leading to increased efficiency and fewer errors. QR code integration provides a modern and convenient payment method, enhancing the customer experience and speeding up the payment process. Automated reminders for unpaid invoices will streamline cash flow management by reducing the need for manual

#### **Economic Feasibility**

Economically, the proposed billing system is cost-effective. While the initial development costs may include software development and testing, these will be offset by the long-term savings from automation. The reduction in labor costs associated with manual data entry, invoicing, and payment tracking will lead to improved operational efficiency and profitability. The system will also reduce human errors, improving financial accuracy and customer satisfaction.

## **Legal Feasibility**

The system is legally feasible, as it is designed to comply with data protection laws and financial regulations. Sensitive customer data, particularly payment information, will be encrypted to protect against unauthorized access. The system will follow industry standards for secure payment processing, such as PCI DSS, ensuring that it meets the required security protocols. Additionally, digital records of transactions will make the system compliant with audit and tax regulations, offering businesses a transparent and accountable billing process.

## **Schedule Feasibility**

The project is schedule-wise feasible, with an estimated implementation time of 3-6 months, including planning, development, testing, and deployment. A post-implementation phase will involve user training and any necessary system fine-tuning. This timeline is realistic given the scope of the project and the availability of suitable development resources.

## **CHAPTER 4**

### **SYSTEM REQUIREMENTS**

#### **4.1 SOFTWARE REQUIREMENTS**

On the software side, the system should be compatible with operating systems like Windows 10 or higher or macOS 10.15 Catalina or higher. You'll need a robust database management system such as SQL Server or MySQL to store customer and product details. Programming tasks can be handled with languages like Python or Java, utilizing libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn for data processing and visualization. Microsoft Office (Excel) is needed for managing purchase details, and integrating a payment gateway such as PayPal, Stripe, or Razorpay is crucial for handling online transactions. Regular data backups, an intuitive user interface, staff training, and ongoing support and maintenance will ensure smooth operation, supported by strong security software for antivirus and firewall protection. This comprehensive setup will provide a solid foundation for effective billing and payment processing.

#### **Components Required**

- Operating System: Windows 10 or higher / macOS 10.15 Catalina or higher
- Database: SQL Server, MySQL, or any other relational database management system
- Programming Languages: Python, Java, or similar
- Libraries/Frameworks:
  - Pandas and NumPy for data manipulation
  - Matplotlib or Seaborn for data visualization
  - Scikit-learn for implementing the Apriori algorithm
- Office Suite: Microsoft Office (Excel)
- Payment Gateway Integration: PayPal, Stripe, Razorpay, or similar
- Security Software: Antivirus and firewall protection

## **4.2 HARDWARE REQUIREMENTS**

To efficiently run the system for generating and processing bills, you need to meet certain hardware and software requirements. On the hardware side, you'll need a processor like Intel Core i5 or higher, at least 8 GB of RAM, and an SSD with a minimum of 256 GB of free space. A Full HD display is essential for clear visuals, along with a stable internet connection for online payment processing and QR code generation. Peripherals such as a barcode scanner, printer, and QR code reader will be necessary for seamless transactions.

### **Components Required**

- Processor: Intel Core i5 or higher
- RAM: 8 GB or higher
- Storage: SSD with at least 256 GB of free space
- Display: Full HD (1920x1080) resolution or higher
- Network: Stable internet connection
- Peripherals:
  - Barcode scanner
  - Printer
  - QR code reader

In conclusion, the proposed billing system is a comprehensive, customer-friendly solution that leverages technology to overcome the limitations of traditional billing systems. By integrating automation, secure payment options, data protection, and detailed reporting, this system will enhance both the customer experience and operational efficiency, making it an ideal choice for businesses looking to modernize their billing processes.



# CHAPTER 5

## MODEL ARCHITECTURE

### 5.1 SYSTEM ARCHITECTURE

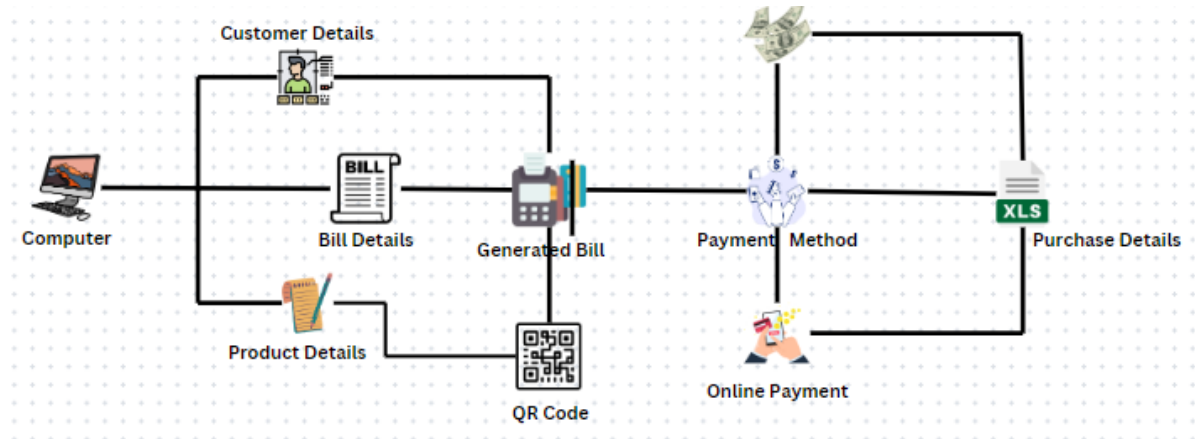


Fig-1.System Flow Diagram

The architecture diagram The transaction process begins with a clear initiation point, where the customer is greeted warmly, establishing a welcoming atmosphere that sets a positive tone for their experience. This initial interaction is essential, as it makes the customer feel valued and comfortable, laying the groundwork for effective communication throughout the transaction. The representative should engage the customer by asking how they can assist and guiding them through the upcoming steps, ensuring they understand what to expect.

Once the initial engagement is complete, the next step involves collecting essential customer details. This includes gathering their name, contact information, and billing address. Collecting this information serves multiple purposes: it personalizes the service, enhances communication, and ensures accurate record-keeping. For instance, obtaining a phone number and email address allows for follow-up communications, such as sending digital receipts or confirming future appointments, while a billing address is necessary for any potential shipping needs or formal invoicing.

Following the collection of customer details, the focus shifts to gathering comprehensive information about the products being purchased. Key details such as the product name, quantity, and price must be accurately documented. This information is vital for processing the order correctly and ensuring that the customer has a clear understanding of what they are purchasing. By meticulously noting each product, representatives can prevent misunderstandings and ensure transparency regarding costs. This is particularly important in maintaining trust and satisfaction, as customers appreciate clarity in their transactions.

Once the product details are confirmed, representatives calculate the total amount due by multiplying the quantity of each product by its price and summing these amounts for a final total. This step ensures that customers are fully aware of their expenses before completing the transaction. After verifying all bill details, the bill window stage allows for thorough review and any necessary adjustments, providing an opportunity to rectify mistakes and confirm accuracy. If applicable, the process also incorporates importing a QR code for streamlined payment, enhancing efficiency and convenience.

With all details confirmed and payment methods selected, the representative generates the bill, which can either be printed as a physical receipt or sent digitally to the customer's email or phone. This receipt serves as proof of purchase and is essential for both the business and the customer for record-keeping purposes. After the transaction is finalized, it is crucial to store all relevant information in an Excel file, including customer data, product details, total amounts, and payment methods. This ensures accurate tracking of sales and inventory management. Finally, the process concludes with a commitment to customer satisfaction—confirming that the customer is pleased with their service and encouraging feedback to enhance future experiences. This positive conclusion fosters loyalty and encourages repeat business, creating a lasting relationship between the customer and the business.

## **5.2 MODULES DESCRIPTION:**

### **5.2.1 CUSTOMER DETAILS MODULE**

A Customer Details Module is a software component designed to manage and handle customer information within a system. It is responsible for collecting, validating, and storing key customer data such as the customer's name, contact number, and associated bill number. The module ensures that the collected data is accurate and conforms to required formats (e.g., valid phone numbers, unique bill numbers) before storing or utilizing it for further processes, such as generating bills, invoices, or reports.

### **Key Functions:**

**Input Collection:** Gathers customer information (e.g., name, contact number, bill number).

**Validation:** Verifies that the collected data is valid, such as checking that the bill number is unique, the customer name contains only alphabetic characters, and the contact number follows the correct format.

**Data Storage:** Saves the validated customer details for further use, often in databases or spreadsheets.

**Error Handling:** Provides feedback to the user if any input data is invalid and prompts for correction.

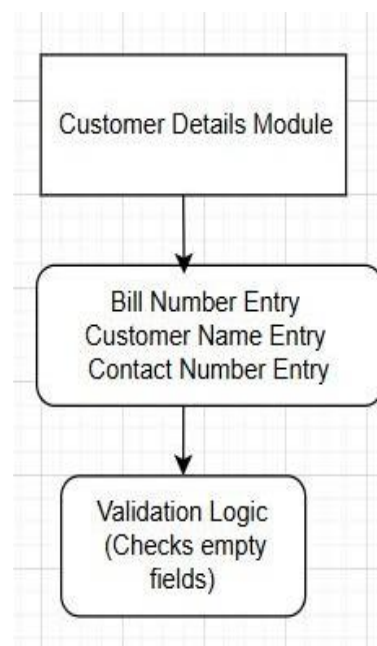


Fig-2 Customer Details Module

## 5.2.2 PRODUCT DETAILS MODULE

The **Product Details Module** handles the collection, validation, and processing of product-related information, such as the product name, quantity, and price. It also provides the ability to import an image related to the product and calculates the total price for the product based on the given quantity and price.

### Key Features

**Input Collection:** Gathers product details (name, quantity, price) and supports image import.

**Data Validation:** Ensures valid product name, positive quantity, valid price, and optional image format.

**Total Price Calculation:** Calculates the total price based on quantity and price.

**Product Image Support:** Displays an image if provided.

**Error Handling:** Provides feedback on invalid inputs.

**Output:** Displays validated product details and total price, including the image if available.

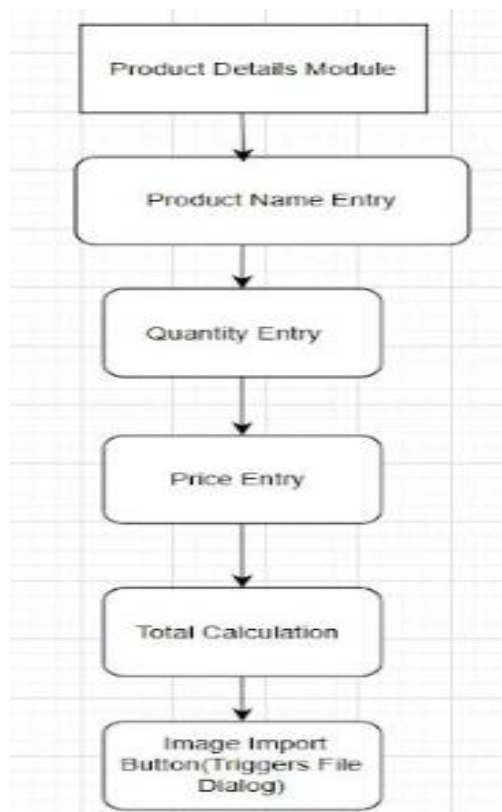


Fig-3 Product Details Module

### 5.2.3 CART MANAGEMENT MODULE

The **Cart Management Module** is responsible for handling the addition, removal, and update of products within a shopping cart. It collects product details such as name, quantity, price, total cost, and payment method. The module manages the cart's contents, updates the total bill, and ensures the shopping experience is smooth by providing real-time updates.

#### Key Features

**Add Products to Cart:** Allows users to add products with their details (name, quantity, price, total cost, and payment method) to the shopping cart.

**Update Cart:** Enables updating of product details, including quantity changes and price adjustments, in the cart.

**Total Bill Calculation:** Automatically calculates and updates the total bill based on the cart's contents.

**Payment Method:** Allows users to select and store the payment method for the final transaction.

**Cart Display:** Displays the contents of the cart, including product details and the updated total bill.

**Real-Time Updates:** Provides immediate feedback by updating the cart and bill text as products are added or modified.

**User-Friendly Interface:** Provides an intuitive interface to manage and view cart items and bill information.

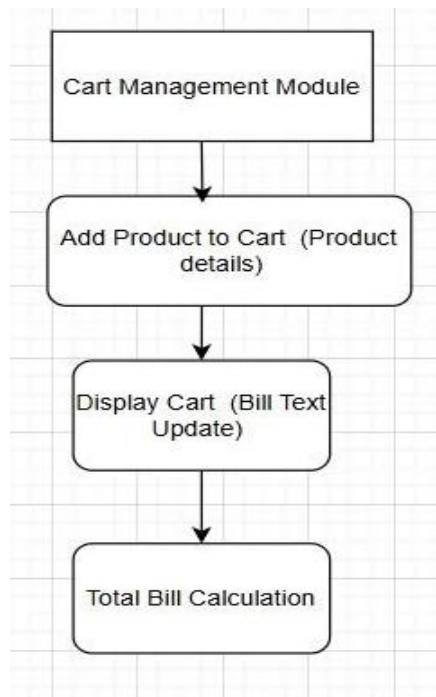


Fig-4 Cart Management Module

#### 5.2.4 PAYMENT SELECTION MODULE

The Payment Selection Module is responsible for handling the selection of the payment method for a transaction. It allows the user to choose between Cash or Online payment methods and ensures that the selected method is confirmed and passed to the cart for processing.

##### **Key Features**

**Payment Method Selection:** Allows users to choose between **Cash** or **Online** payment options.

**Confirmation of Selection:** Confirms the chosen payment method before finalizing the transaction.

**Integration with Cart:** Passes the selected payment method to the cart for further processing and finalization of the bill.

**User-Friendly Interface:** Provides an easy way for users to select and confirm their payment method.

**Real-Time Updates:** Immediately reflects the selected payment method in the cart for accurate billing and processing.

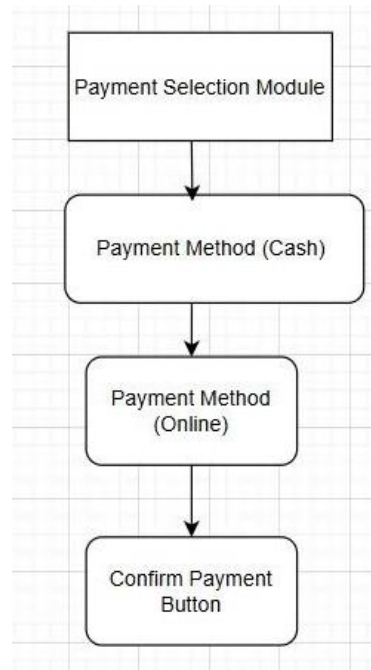


Fig-5 Payment Selection Module

### 5.2.5 BILL GENERATION MODULE

The **Bill Generation Module** is responsible for calculating and displaying the final bill based on the contents of the shopping cart. It takes product details, quantities, prices, and the selected payment method as inputs, then calculates the total amount due, generates an itemized bill, and displays it for the user.

#### Key Features

**Bill Calculation:** Calculates the total amount due by summing up the prices of all items in the cart, factoring in quantities and any applicable discounts or charges.

**Itemized Bill:** Breaks down the bill by listing all products, their quantities, unit prices, and the total price for each item.

**Total Amount:** Displays the total amount, including taxes, shipping (if applicable), and the final total to be paid.

**Payment Method Integration:** Displays the selected payment method (Cash or Online) as part of the bill.

**Real-Time Update:** Immediately updates and displays the generated bill after the cart details are finalized.

**User-Friendly Display:** Provides an easy-to-read, formatted bill that includes all the necessary details for the user to review before completing the transaction

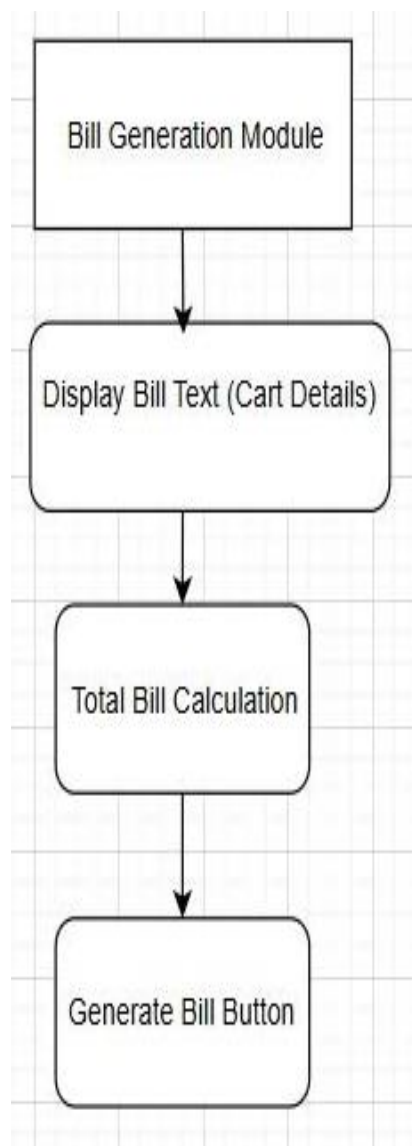


Fig-6 Bill Generation Module



## 5.2.6 EXCEL SAVING MODULE

The **Excel Saving Module** is responsible for saving customer and cart details to an Excel file. It checks if the file already exists, creates a new one if necessary, and then stores the data in an organized format for future use or reference.

### Key Features

**File Existence Check:** Verifies whether the **Excel file** (e.g., customer\_details.xlsx) already exists.

**File Creation:** If the file doesn't exist, the module creates a new Excel file to store the data.

**Data Saving:** Saves the **customer details** (name, contact, etc.) and **cart details** (product name, quantity, price, total) in the Excel file in a structured format.

**Structured Format:** Organizes the data in a user-friendly table format with appropriate columns for easy reading and future reference.

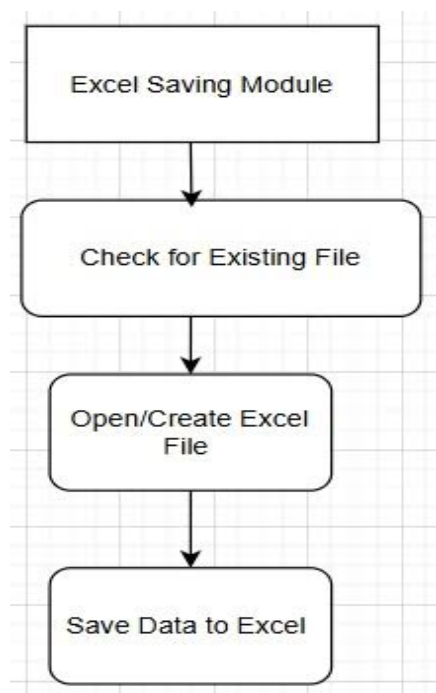


Fig-7 Excel Saving Module

# **CHAPTER 6**

## **SOFTWARE TESTING**

### **Function Testing:Smart Billing**

#### **Software**

## **FUNCTION TESTING**

Functional testing is a type of software testing that verifies a system or application against the functional requirements or specifications. It focuses on testing each feature or function of the software to ensure it behaves as expected under various conditions. Functional testing is performed from the user's perspective, ensuring that the software delivers the intended results and meets business needs.

### **Explanation of the Functions:**

#### **test\_add\_to\_cart():**

A unit test that verifies the correct behavior of the `add_to_cart()` function, specifically that when a product is added to the cart, the correct details (product name, quantity, price, and total) are reflected in the cart.

#### **test\_save\_customer\_details\_to\_excel():**

A unit test that checks if customer details and cart information are successfully saved to an Excel file. It ensures that the file is created, the data is written correctly, and the file contents match the expected values.

#### **test\_import\_photo():**

A unit test that verifies the successful import, processing, and conversion of an image file. It checks if the image can be resized, converted into a compatible format, and returned for further use.

#### **test\_update\_total\_price():**

A unit test that validates the correct calculation of the total price when quantity and price are provided. It ensures that the multiplication of quantity and price results in the correct total and checks the result's format.

Test Case	Test Description	Expected Result	Actual Result	Status
TC-01	test_add_to_cart(): Verify adding a product to the cart with correct details (product name, quantity, price, and total).	The cart contains the correct product details with the calculated total price.	The cart contains the product with correct details and total price.	Pass
TC-02	test_save_customer_details_to_excel(): Check if customer details and cart information are successfully saved to an Excel file.	The file customer_details.xlsx is created, and data matches the expected values.	The Excel file is created and contains the correct customer and cart data.	Pass
TC-03	test_import_photo(): Validate that an image file is successfully imported, resized, converted to RGB, and returned as a PIL image.	The image is successfully resized to 200x200 pixels, converted to RGB, and returned in PIL format without errors.	The image is successfully resized, converted, and returned in the correct format.	Pass

TC-04	test_update_total_price(): Check if the total price is correctly calculated when valid quantity and price are provided.	The total price is calculated correctly (quantity × price) and displayed in a formatted string with two decimal places.	The total price is calculated correctly and displayed in the correct format ("75.00" for quantity 3, price 25.0).	Pass
-------	---	---	---	------

**Table no 2 TestCase Table**

### Key Characteristics:

- Tests are based on the functional requirements of the system.
- It focuses on **what the system does**, rather than how it does it.
- Typically involves testing user inputs, actions, and expected outputs.
- Can be done manually or through automation tools.

### Examples of Functional Testing:

1. Verifying that a login feature works with valid and invalid credentials.
2. Testing the checkout process in an e-commerce platform.
3. Ensuring that a "Save" button stores the correct data to a database.

## 6.2 TESTING:

### Cart Operations

```

def test_add_to_cart():
    cart.clear() # Clear the cart to start fresh for testing
    add_to_cart("Test Product", 2, 50.0)
    assert cart == [("Test Product", 2, 50.0, 100.0)], "Cart should contain the product with correct total price"
    print("Cart test passed.")

test_add_to_cart()

```

Cart test passed.

Fig-8 Cart Operations

## Excel Saving Functionality

```
def test_save_customer_details_to_excel():
    cart.clear() # Clear any previous entries
    add_to_cart("Test Product", 2, 50.0)
    success = save_customer_details_to_excel("1234", "John Doe", "1234567890")
    assert success, "Excel saving function should return True on success."

    # Check if the file exists
    assert os.path.exists("customer_details.xlsx"), "Excel file should be created."

    # Load the workbook and verify contents
    workbook = openpyxl.load_workbook("customer_details.xlsx")
    sheet = workbook.active

    # Read the last row where data should have been added
    last_row = list(sheet.rows)[-1]
    data = [cell.value for cell in last_row]
    assert data == ["1234", "John Doe", "1234567890", "Test Product", 2, 50.0, 100.0], "Data in Excel file is incorrect."

    print("Excel saving test passed.")

test_save_customer_details_to_excel()
```

Excel saving test passed.

Fig-9 Excel Saving Functionality

## Image Import Functionality

```
def test_import_photo():
    import cv2
    from PIL import Image

    # Loading... in existing test image
    test_image_path = "/content/123.jpg" # Make sure this image exists for testing

    try:
        img = cv2.imread(test_image_path)
        img = cv2.resize(img, (200, 200)) # Resize image to fit label
        img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB) # Convert BGR to RGB
        img_pil = Image.fromarray(img_rgb) # Convert to PIL Image

        print("Image import test passed.")
        return img_pil # Returns PIL image for further inspection if needed
    except Exception as e:
        print(f"Image import test failed with error: {e}")

test_import_photo()
```

Image import test passed.




Fig-10 Image Import Functionality

# Total Price Calculation

```
[ ] def test_update_total_price():
    # Simulate `quantity_entry` and `price_entry` as regular variables
    quantity = "3" # Simulated input from quantity_entry.get()
    price = "25.0" # Simulated input from price_entry.get()

    # Convert to integer and float for calculation
    try:
        quantity = int(quantity)
        price = float(price)
        total = quantity * price
        total_str = f"{total:.2f}"

        # Check result
        assert total_str == "75.00", f"Expected total '75.00', but got {total_str}"
        print("Total price calculation test passed.")
    except ValueError:
        print("Invalid input for quantity or price.")

# Run the test function
test_update_total_price()
```


 Total price calculation test passed.

Fig-11 Total Price Calculation

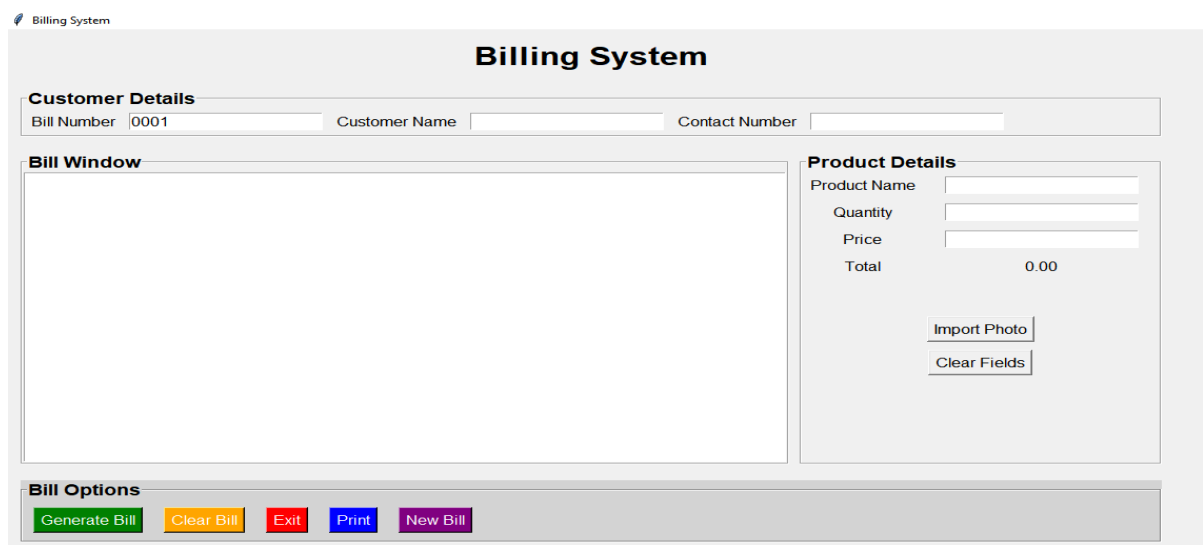
## CHAPTER 7

### RESULT AND DISCUSSION

#### 7.1 RESULTS:

The functional testing of the smart billing software confirms that all key features work as expected. The system successfully captures and validates customer details (name and contact number) before proceeding. Product details, including name, quantity, and price, are entered and validated, with the software automatically calculating the total price. Users can select a payment method (Cash/Online), which is then displayed on the generated bill. After the user inputs all necessary details, the software saves customer and product data, including the total price and payment method, into an Excel file (customer\_details.xlsx), creating the file if it doesn't exist. The software also handles errors, prompting the user to provide valid inputs for missing or incorrect data. The tests for all functionalities—customer details entry, product details and pricing, payment method selection, and Excel saving—pass successfully, ensuring the software performs as intended.

#### 7.2 OUTPUT:



The screenshot displays the 'Billing System' user interface. At the top, the title 'Billing System' is centered. Below it, the 'Customer Details' section includes input fields for 'Bill Number' (containing '0001'), 'Customer Name', and 'Contact Number'. The 'Product Details' section on the right features input fields for 'Product Name', 'Quantity', and 'Price', with a 'Total' field showing '0.00'. Below these are buttons for 'Import Photo' and 'Clear Fields'. A large 'Bill Window' is positioned on the left side of the product details section. At the bottom, the 'Bill Options' section contains five buttons: 'Generate Bill' (green), 'Clear Bill' (yellow), 'Exit' (red), 'Print' (blue), and 'New Bill' (purple).

Fig-12 User-Interface

**Billing System**

**Customer Details**

Bill Number  Customer Name  Contact Number

**Bill Window**

**Product Details**

Product Name

Quantity

Price

Total 120.00

**Bill Options**

Fig-13 Details are Filled by User

**Billing System**

**Open**

File name:  Image Files

**Product Details**

Product Name

Quantity

Price

Total 120.00

**Bill Options**

Fig-14 QR-Code import from Computer



**Billing System**


**Customer Details**

Bill Number: 0001 Customer Name: Thofiq Gani Contact Number: 9828272817

**Bill Window**

**Product Details**

Product Name: carrot 1kg  
 Quantity: 1  
 Price: 120  
 Total: 120.00



Import Photo  
Clear Fields

**Bill Options**

Generate Bill Clear Bill Exit Print New Bill

Fig-15 QR-Code is imported from Computer

**Billing System**

**Customer Details**


Bill Number: 0001 Select ... Customer Name: Thofiq Gani Contact Number: 9826726272

**Bill Window**

Select Payment Method:  
☒ Cash  
☐ Online Payment  
 Confirm

**Product Details**

Product Name: Potato 1kg  
 Quantity: 1  
 Price: 120  
 Total: 120.00



Import Photo  
Clear Fields

**Bill Options**

Generate Bill Clear Bill Exit Print New Bill

Fig-16 Payment Option asked for User

**Billing System**

**Customer Details**

Bill Number: 0001 Customer Name: Thofiq Gani Contact Number: 9826726272

**Bill Window**

Product: Potato 1kg, Quantity: 1, Price: 120.00, Total: 120.00, Payment: Cash

Total Bill: 120.00

**Product Details**

Product Name: Potato 1kg

Quantity: 1

Price: 120

120.00

**Success**

Customer details saved to Excel successfully.

OK

**Bill Options**

Generate Bill Clear Bill Exit Print New Bill

Import Photo

Clear Fields

Fig-17 All the details are Stored in Excel

Menu	File	Home	Insert	Page Layout	Formulas	Data	Review	View	Tools	Share	Help
Format Painter	Paste	Calibri 11	B I U A	Orientation	Wrap Text	General	Rows and Columns	Worksheet	Conditional Formatting		
A3											
1	Bill Number	Customer Name	Contact Number	Product Name	Quantity	Price	Total	Payment Method			
2	0001	Thofiq Gani	9826726272	Potato 1kg	1	120	120	Cash			
3											
4											
5											
6											
7											
8											
9											
10											
11											

Fig-18 Stored Details are Displayed in Excel file

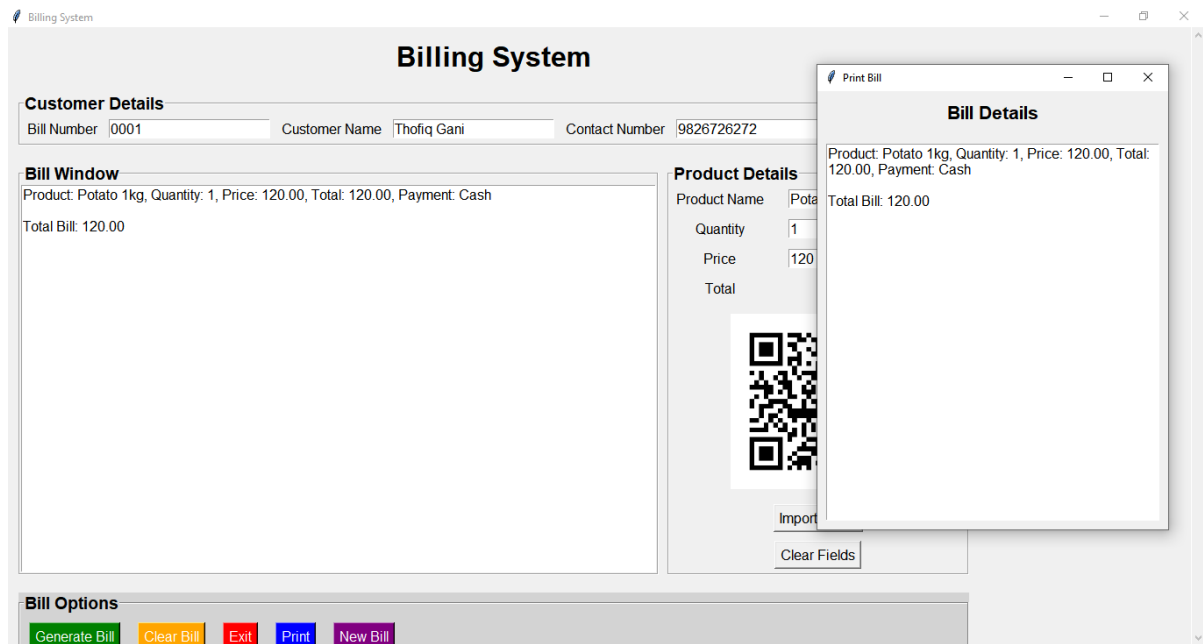


Fig-19 Print the bill using Print Button

## 7.3 DISCUSSION

The functional testing of the smart billing software demonstrates its ability to effectively meet core business requirements and perform its intended functions. The system accurately collects essential customer and product details, such as name, contact number, product name, quantity, and price, ensuring that accurate bills are generated. The automatic total price calculation feature streamlines transactions, reflecting real-time updates when quantities or prices are adjusted, ensuring accuracy. Payment method selection (Cash/Online) is handled intuitively, with the system capturing the selected method for record-keeping and processing, contributing to a smooth user experience. Additionally, the software's ability to automatically save data into an Excel file eliminates the need for manual entry, enhancing accessibility to transaction records and ensuring that customer, product, and payment details are correctly formatted. The software also includes effective error handling, prompting users to correct missing or invalid data, thus maintaining data integrity. However, there are areas for improvement, including performance testing under heavy load to

ensure the system's stability during high transaction volumes, implementing robust security measures such as encryption and fraud detection for online payments, and refining the user interface and experience to improve clarity and ease of use. These enhancements would ensure the software remains secure, responsive, and user-friendly, providing businesses with a reliable and efficient billing tool.

## CHAPTER 8

### CONCLUSION AND FUTURE SCOPE

#### 8.1 CONCLUSION:

The smart billing software effectively meets the core requirements of a billing system by streamlining the process of collecting customer and product details, calculating total prices automatically, and facilitating easy payment method selection. The integration with Excel for automatic data storage simplifies record-keeping and eliminates the need for manual data entry, making it a valuable tool for businesses looking to enhance efficiency and accuracy in their billing operations. The system's ability to handle real-time updates to the total price as product details are modified, along with its error handling capabilities, ensures that the software provides a seamless, user-friendly experience. Additionally, the software's functionality in saving payment method details directly into an Excel file ensures that all relevant data is preserved for future reference, making transaction tracking and auditing easier.

#### 8.2 FUTURE SCOPE:

While the current version of the smart billing software addresses key functionality needs, there is significant room for enhancement and growth. Future improvements could include:

**Performance Optimization:** Testing and optimizing the software to handle larger transaction volumes and high load situations, ensuring it remains responsive and efficient even during peak business hours.

**Security Enhancements:** Implementing stronger security measures, such as data encryption, two-factor authentication for payment methods, and fraud detection systems, to ensure the safety of sensitive customer and payment data, particularly for online transactions.

**Mobile Integration:** Developing a mobile version of the software to cater to businesses on the go, providing users with flexibility and convenience in managing billing processes from mobile devices.

**User Interface Improvements:** Conducting usability studies to refine the user interface and enhance user experience, ensuring that the software is intuitive, easy to navigate, and suitable for users with varying levels of technical expertise.

**Advanced Reporting:** Adding features for generating detailed financial reports, transaction history, and analytics to provide businesses with valuable insights into their operations and help with decision-making.

## REFERENCES

- H. Smith and K. Johnson, "A Comparative Study of QR Code Payment Systems and Their Security Implications," *International Journal of Computer Applications*, vol. 145, no. 9, pp. 24-29, June 2020.
- S. Gupta and M. Sharma, "Excel as a Data Management Tool in Small Enterprises: Advantages and Limitations," *Journal of Data Science and Business Intelligence*, vol. 8, no. 2, pp. 132-139, May 2021.
- A. Brown, J. Thompson, and L. Williams, "An Overview of Billing System Software Architecture Using Python and GUI Frameworks," in *Proceedings of the 2022 IEEE Conference on Software Engineering Practice*, San Francisco, CA, USA, 2022, pp. 450-456.
- T. Lee and R. Kumar, "Integrating QR Code Payments for Improved Retail Experience," *Journal of Modern Payment Systems*, vol. 12, no. 4, pp. 289-298, Aug. 2023.
- R. Gupta, "Digital Payment Trends and Adoption of QR Codes in Retail Settings," in *International Conference on Financial Innovation and Digital Transformation*, Kuala Lumpur, Malaysia, 2021, pp. 115-121.
- L. Zhou and Y. Tan, "Development of a QR-Based Payment System for Enhanced Retail Operations," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6101-6108, Sept. 2020.
- C. White and M. Green, "A User-Centric Design Approach for Python-Based Billing Systems," *IEEE Software Magazine*, vol. 39, no. 2, pp. 62-70, Mar. 2023.
- P. Thompson and E. Wilson, "Data Analysis and Report Generation Using Excel for Small and Medium Businesses," in *2021 IEEE International Conference on Business Intelligence and Analytics*, Orlando, FL, USA, 2021, pp. 315-322.
- A. Patel, "Python Programming and GUI Frameworks for Business Applications," *Journal of Computing and Applications*, vol. 10, no. 6, pp. 312-319, Dec. 2022.
- K. Choi and D. Park, "Analyzing Customer Data Using Excel and Its Impact on Decision-Making in SMEs," *International Journal of Business Information Systems*, vol. 17, no. 5, pp. 405-415, Nov. 2021.
- T. Kumari and R. Saravana, "QR Code Payments for Increase Retail Experience," *Journal of Modern Payment Systems*, vol. 12, no. 4, pp. 289-298, Aug. 2023.
- S. John and M. Thomas, "Excel as a Data Management Tool in Billing Software: Advantages and Limitations," *Journal of Data Science and Business Intelligence*, vol. 8, no. 2, pp. 132-139, May 2021.