



**Tribhuvan University**  
**Faculty of Humanities and Social Sciences**

**A PROJECT REPORT**  
**ON**  
**TICKBILL**

**Submitted to**  
**Department of Computer Application**  
**Aadim National College**

*In partial fulfilment of the requirements for the Bachelor in Computer Application*

**Submitted By**

**Hridaya Prajapati**

Symbol No: 105902098

**Sushmeeta Shrestha**

Symbol No: 105902119

March 2025

Under the Supervision of  
**Mr. Saroj Bhandari**



**Tribhuvan University**  
**Faculty of Humanities and Social Sciences**  
**Aadim National College**

## **SUPERVISOR'S RECOMMENDATION**

I hereby recommend that this project prepared under my supervision by **Hridaya Prajapati (105902098)** and **Sushmeeta Shrestha (105902119)** entitled “**TickBill**” in partial fulfilment of the requirements of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

.....

**SIGNATURE**

**Mr. Saroj Bhandari**

**Supervisor**

**Aadim National College**

**Chabahil, Chuchepati, Kathmandu**



**Tribhuvan University**  
**Faculty of Humanities and Social Sciences**  
**Aadim National College**

## LETTER OF APPROVAL

This is to certify that this project prepared by **Hridaya Prajapati (105902098)** and **Sushmeeta Shrestha (105902119)** entitled “**TickBill**” in partial fulfilment of the requirements for the degree of Bachelor of Computer Application has been evaluated in our opinion. It is satisfactory in scope and quality as a project for the required degree.

<p>.....</p> <p><b>Mr. Saroj Bhandari</b>  <b>Supervisor</b>  Aadim National College  Chabahil, Chuchepati, Kathmandu</p>	<p>.....</p> <p><b>Mr. Shankar Prasad Sharma</b>  <b>Principal/Campus Chief</b>  Aadim National College  Chabahil, Chuchepati, Kathmandu</p>
<p>.....</p> <p><b>Internal Examiner</b></p>	<p>.....</p> <p><b>External Examiner</b></p>

## ABSTRACT

In growing world of IT maintaining an application which offer both ticketing and billing feature is essential. TickBill is an open-source, Java-based ticketing, billing, and inventory management system for startups and small businesses. It offers an inexpensive and customizable alternative to proprietary and rigidly structured Point of Sale (POS) systems that can be very expensive. There are no subscription fees, and the source code is fully accessible, making businesses independently make changes according to their requirements. TickBill provides essential features such as real-time inventory tracking, role-based access control, a friendly user interface, and an import/export feature to manage data effectively.

**Keywords:** *POS System, Open-source, RBAC, Swing with Java*

## ACKNOWLEDGEMENT

We would like to express our sincere gratitude to everyone who contributed to the development of the "TickBill" project. First and foremost, we would like to thank our project supervisor, **Mr. Saroj Bhandari** Sir for his invaluable guidance, support, and encouragement throughout this project. His expertise and insights were instrumental in shaping the direction and scope of this project.

Additionally, we extend our appreciation to our Coordinator **Er. Prem Subedi** Sir for providing the resources and support necessary to complete this project. The access to facilities and technology greatly facilitated our work.

Finally, we are thankful to our families and friends for their unwavering support and encouragement throughout this journey. Their belief in our abilities kept us motivated and focused.

This project would not have been possible without the contributions of all these individuals and organizations. Thank you for your support.

# TABLE OF CONTENTS

SUPERVISOR’S RECOMMENDATION .....	i
LETTER OF APPROVAL .....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENT .....	iv
TABLE OF CONTENTS.....	v
LIST OF ABBREVIATIONS.....	vii
LIST OF FIGURES .....	viii
LIST OF TABLES .....	ix
CHAPTER 1: INTRODUCTION .....	1
1.1 Introduction.....	1
1.2 Problem Statement.....	2
1.3 Objectives .....	2
1.4 Scope and Limitation .....	2
1.5 Report Organization.....	3
CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW .....	5
2.1 Background Study.....	5
2.2 Literature Review.....	5
CHAPTER 3: SYSTEM ANALYSIS AND DESIGN .....	7
3.1 System Analysis.....	7
3.1.1 Requirement Analysis.....	8
3.1.2 Feasibility Analysis.....	11
3.1.3 Data Modeling (ER-Diagram) .....	13
3.1.4 Process Modeling (DFD).....	13
3.2 System Design .....	14
3.2.1 Architectural Design .....	14
3.2.2 Database Schema Design.....	15

3.2.3 Interface Design .....	15
CHAPTER 4: IMPLEMENTATION AND TESTING .....	20
4.1 Implementation .....	20
4.1.1 Tools Used .....	20
4.1.2 Implementation Details of Modules.....	21
4.2 Testing.....	23
4.2.1 Test Cases for Unit Testing.....	24
4.2.2 Test Cases for System Testing.....	28
4.2.3 Integration Testing .....	29
4.2.4 System Testing.....	29
CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS .....	30
5.1 Lesson Learnt / Outcome .....	30
5.2 Conclusion .....	30
5.3 Future Recommendations .....	31
REFERENCES .....	32

## LIST OF ABBREVIATIONS

API	Application Programming Interface
AWT	Abstract Window Toolkit
CASE	Computer Aided Software Engineering
CSV	Comma Separated Values
DFD	Data Flow Diagram
ER	Entity Relationship
IDE	Integrated Development Environment
JDBC	Java Database Connectivity
JFC	Java Foundation Classes
MySQL	My Structured Query Language
RBAC	Role Based Access Control
POS	Point of Sale
UI	User Interface
UX	User Experience



## LIST OF FIGURES

Figure 3.1 Waterfall Model used in Development of TickBill.....	7
Figure 3.2 Use Case Diagram of TickBill .....	8
Figure 3.3 Project Gantt Chart of TickBill .....	12
Figure 3.4 ER Diagram of TickBill .....	13
Figure 3.5 Level 0 DFD of TickBill .....	13
Figure 3.6 Level 1 DFD of TickBill .....	14
Figure 3.7 Architecture Diagram of TickBill .....	14
Figure 3.8 Database Schema of TickBill .....	15
Figure 3.9 Login Panel of TickBill.....	15
Figure 3.10 Inventory Panel of TickBill .....	16
Figure 3.11 Sales Panel of TickBill .....	16
Figure 3.12 User Panel of TickBill.....	17
Figure 3.13 Import/Export Panel of TickBill.....	17
Figure 3.14 Main Panel of TickBill .....	18
Figure 3.15 Invoice Panel of TickBill.....	18
Figure 3.16 Per User Invoice Panel of TickBill.....	19
Figure 3.17 Report Panel of TickBill.....	19

## LIST OF TABLES

Table 3.1 Use Case Scenario for generating invoice .....	8
Table 3.2 Use Case Scenario for managing invoices.....	9
Table 3.3 Use Case Scenario for managing staffs .....	9
Table 3.4 Use Case Scenario for managing reports .....	9
Table 3.5 Use Case Scenario for managing inventory.....	10
Table 3.6 Use Case Scenario for managing data .....	10
Table 3.7 Use Case Scenario for Login .....	10
Table 4.1 Test Cases for User Login & Registration.....	24
Table 4.2 Test Cases for Invoice Generation.....	25
Table 4.3 Test Case for Admin Login .....	25
Table 4.4 Test Cases for User Management .....	26
Table 4.5 Test Case for Product Management.....	27
Table 4.6 Test Case for Invoice Management .....	27
Table 4.7 Test Case for Report Generation .....	27
Table 4.8 Test Case for Import/Export data.....	28
Table 4.9 Test Cases for System User Interface and Navigation .....	28
Table 4.10 Test Cases for System Functionality .....	28

# CHAPTER 1: INTRODUCTION

## 1.1 Introduction

In today's competitive business world, startups and small businesses are looking out for cheap work solutions to manage their work and keep up with efficiency. One of the key areas where businesses need efficiency is in ticketing, billing, and inventory management. A Point of Sale (POS) system is a comprehensive solution comprising both hardware and software components, aiming to facilitate seamless transaction processing and optimize sales operations, particularly in retail environments [1]. While proprietary POS systems are robust, they come with high costs in the form of SaaS subscriptions, hence limiting their usage to smaller enterprises. Most of the systems also lack flexibility or provide no room for modifications which might not be able to accommodate the needs of the business.

TickBill is an open-source, Java-based ticketing and billing management system. It allows startups to monitor sales, track down their inventories, and develop better customer interaction at low costs, as opposed to expensive proprietary POS systems.

TickBill gives small businesses a way to extend the system for their specific needs without ongoing costs since they have open access to the source code. This open-source option is free, not only of cost but also of vendor lock-in, giving businesses complete control over their data and business processes. It will effectively enable small businesses to manage their operations for the better customer experience, effective inventory management, and strong sales tracking. Besides that, the system also allows import/export functionality for seamless data management. The aim of TickBill is to change how startups manage their billing processes and engage with customers, providing a ready-to-use open-source solution to meet the emerging demands of small businesses.

## 1.2 Problem Statement

The high cost and limited functionality of proprietary POS systems create quite a headache for small businesses and startups. Due to the absence of an affordable and flexible alternative, the following problems have occurred in small businesses:

- Most proprietary POS systems charge high subscription fees either on a monthly or yearly basis. These cause a significant financial strain on startups and low-budget small businesses operating on a low budget.
- There are many proprietary systems where access to most of the key functionalities is behind paywalls or complex configurations, which depicts that there is lack of transparency in the system.
- It is challenging for a startup to track and manage its inventory using proprietary software. This leads to losses from overstocking or understocking of inventory.

## 1.3 Objectives

The main objectives of this project are:

- To develop a full CRUD-enabled system for managing ticketing, billing, and inventory, allowing staff to efficiently create, read, update, and delete data, and enabling users to access their tickets and bills.
- To implement role-based access control to ensure only authorized staff and managers can access sales, stock, and customer information.
- To enable real-time tracking of inventory and transactions to streamline sales operations and maintain accurate stock records.

## 1.4 Scope and Limitation

**Scope:** TickBill is a free, flexible, and user-friendly POS system designed for the particular needs of a small business or startup. This is done by offering an affordable alternative to expensive SaaS POS products through zero subscription fees without compromising similar functionalities. TickBill includes fully featured ticketing, billing, and inventory management that enable a business to organize its operations smoothly and keep records accurately. It applies Role-Based Access Control, ensuring that sensitive information within the system is granted only to authorized personnel-for example, managers and administrators. TickBill features a very intuitive interface and is easy to use even by people with minimum technical background.

**Limitations:**

The system also has some limitations which are specified below:

- TickBill doesn't support online payment feature.
- TickBill is desktop-based application and designed for offline use, and thus features which require real-time data synchronization.

## **1.5 Report Organization**

### **Chapter 1: Introduction**

This chapter introduces the project, showing its significance and relevance within the field. The problem statement has been defined, indicating the issue the project is particularly targeting. The objectives are clearly stated in order to identify the results the project is supposed to arrive at. The scope and limitations have also been included to show the boundaries of the project. At the end, it gives an overview of the report structure for the readers' sake.

### **Chapter 2: Background Study and Literature Review**

This chapter lays the foundation through a background study that elaborates on the necessary theories, concepts, and terminologies used in this domain. Second, it flows into an extensive literature review that amalgamates prior research, studies, and projects carried out related to the topic. The review provides a base and context for the present undertaking.

### **Chapter 3: System Analysis and Design**

This chapter covers system analysis, detailing project requirements as functional and non-functional. Functional requirements are illustrated using use case diagrams or lists, while non-functional ones address technical, operational, economic, and schedule feasibilities. Data modelling includes the ER diagram, and process modelling uses Data Flow Diagrams (DFD) to represent system processes. It further elaborates into design, covering system architecture, database schema, interface design with UI/UX elements, and physical DFDs for implementation details.

#### **Chapter 4: Implementation and Testing**

This chapter includes info about implementation phase, discussing the tools utilized such as CASE tools, programming languages, and database platforms. It provides a detailed account of module implementation, describing procedures and functions developed for each system module. The chapter continues with testing, beginning with unit testing to validate individual component functionalities, followed by system testing to ensure overall system reliability and performance.

#### **Chapter 5: Conclusion and Future Recommendations**

This chapter reflects on the project's outcomes and lessons learned. It presents a conclusive summary of the project's achievements, highlighting its contributions to the field. Future recommendations are provided to suggest potential enhancements or areas for further research, ensuring the project's continuous improvement and relevance.

## **CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW**

### **2.1 Background Study**

Most small businesses and startups are still having a hard time managing their sales, inventory, and customer interactions effectively. Most of them use manual methods or expensive proprietary POS systems, each with its limitation. Manual processes are error-prone, inefficient, and slow, while proprietary systems are usually highly subscription-based, not very flexible, and vendor-locked; therefore, they are out of reach for any business on a tight budget [1][2].

Today's modern POS systems have evolved to integrate advanced features like real-time inventory tracking, billing, and sales analytics. Such systems help streamline processes to enhance customer satisfaction and improve the overall operational efficiency of the firm. Despite their advantages, the costs of such proprietary POS solutions remain inaccessible to most small businesses, not to mention rigid structures [1].

The practical alternative is open-source POS systems, which are cost-effective, flexible, and customizable according to specific needs. Such systems avoid recurring costs and put the business in full control of the data and functionality. TickBill is a Java-based, open-source POS system built to meet such demands. It provides an economical, easy-to-use solution with features such as real-time inventory management and role-based access control, bridging the gaps in both manual methods and proprietary systems.

### **2.2 Literature Review**

Research on POS systems indicates that such systems play a critical role in facilitating business processes that involve smooth customer interaction and inventory management. The literature has also established that a POS system has gradually transformed from a simple transaction processing tool to an advanced platform with data analytics, cloud computing, and real-time inventory tracking.

Asrani et al. give an overview of the basic functionality of a POS system, which includes transaction recording, sales reporting, and inventory management. Such systems are crucial for enabling enterprises to maintain operational efficiency in a competitive retail environment. It is also pointed out that while the system provides substantial benefits, its

adoption is often restricted by expensive and rigid solutions, especially for small businesses and startups [1].

Kulandairaj explores the POS system's psychological and very practical influence on consumer behavior. This research highlights how the interface with the POS system can strengthen customer satisfaction and loyalty by ensuring fast and error-free billing. On the other hand, many small businesses cannot use these systems because of their complexity, lack of customization to suit peculiar business needs [2].

The literature also repeatedly identifies a need for affordable and flexible solutions that meet the needs of small businesses. It is suggested that such systems should be user-friendly and reduce dependence on technical expertise, while also offering basic functions such as inventory tracking, sales reporting, and the ability to process transactions securely. In this way, a POS system would enable a startup or small enterprise to hold its own in the marketplace while keeping operational costs as low as possible.

The literature identifies a current need for creative POS solutions that are affordable, customizable, and scalable. This is most important in the development of systems like TickBill, as these systems address specific technological gaps and provide small businesses with a system better suited to their needs.

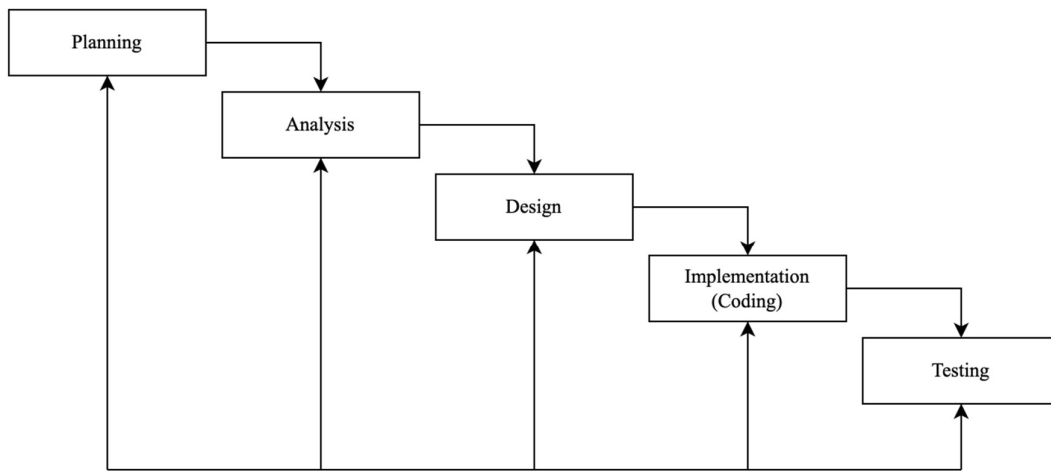


## CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

### 3.1 System Analysis

For the development of TickBill, we've used waterfall model as the development model for our system as this model is very easy to understand than other models. It is easy to manage and arrange tasks. Each phase must be completed before the new phase starts, so there is no overlapping in the phases.

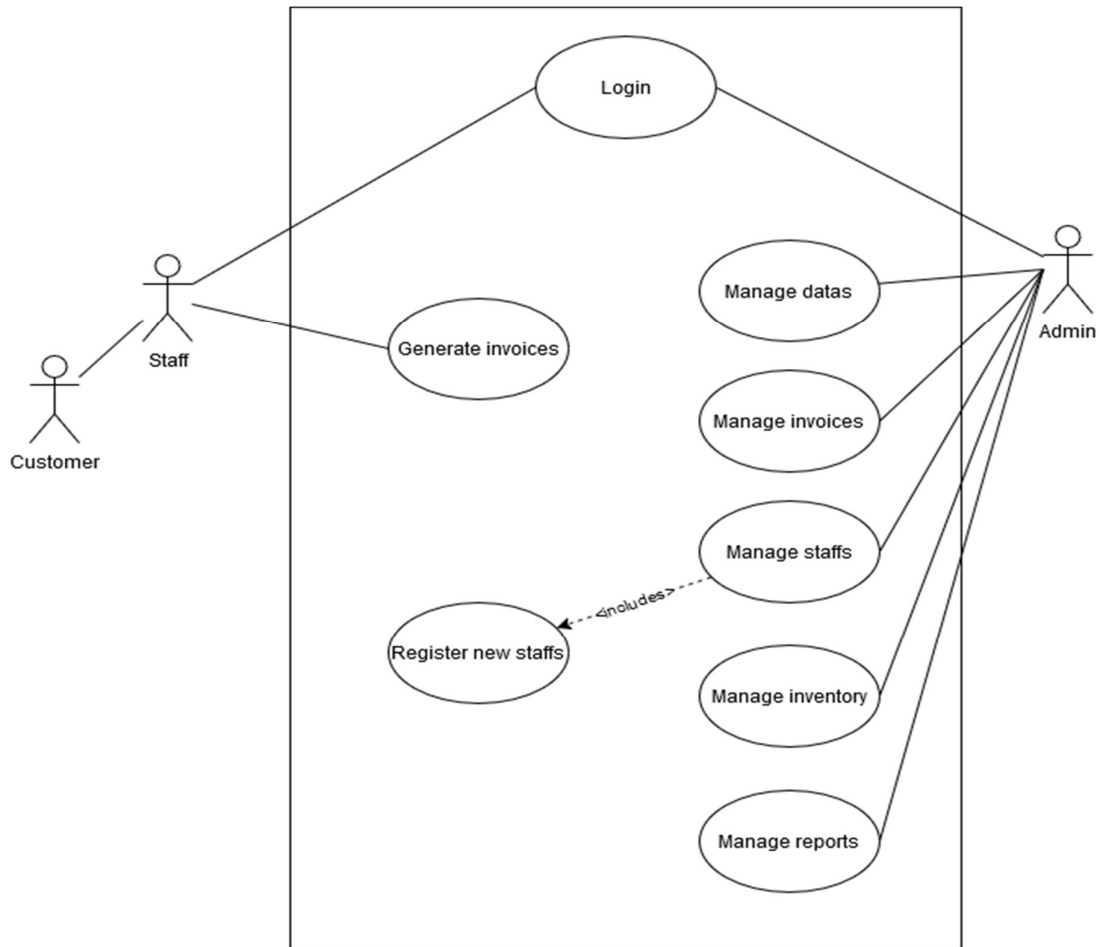
The following illustration is a representation of different phases of the waterfall model:



**Figure 3.1 Waterfall Model used in Development of TickBill**

### 3.1.1 Requirement Analysis

#### i. Functional Requirements



**Figure 3.2 Use Case Diagram of TickBill**

#### Use Case Description:

**Table 3.1 Use Case Scenario for generating invoice**

Use-Case Identifier	UC1: Generate Invoices
Primary Actor	Staff
Secondary Actor	Customer
Description	Staff inputs sales information and generates an invoice
Pre-Condition	User must have an Employee or Admin role
Success Scenario	Successfully generates an invoice and registers sales data
Failure Scenario	Fails to generate an invoice; sales data are not registered

**Table 3.2 Use Case Scenario for managing invoices**

Use-Case Identifier	UC2: Manage Invoices
Primary Actor	Admin
Secondary Actor	X
Description	Admin manages past invoices, including updating and removing them
Pre-Condition	User must have an Admin role
Success Scenario	Successfully updates or deletes invoices
Failure Scenario	Fails to update or delete invoices

**Table 3.3 Use Case Scenario for managing staffs**

Use-Case Identifier	UC3: Manage Staff
Primary Actor	Admin
Secondary Actor	X
Description	Admin manages staff details, including creation, deletion, or updates
Pre-Condition	User must have an Admin role
Success Scenario	Successfully creates, edits, or deletes staff information
Failure Scenario	Fails to create, update, or delete staff information

**Table 3.4 Use Case Scenario for managing reports**

Use-Case Identifier	UC4: Manage Reports
Primary Actor	Admin
Secondary Actor	X
Description	Admin can view or download reports based on existing invoices
Pre-Condition	User must have an Admin role and existing invoices
Success Scenario	Admin successfully views or downloads the report
Failure Scenario	Admin is unable to view or download the report

**Table 3.5 Use Case Scenario for managing inventory**

Use-Case Identifier	UC5: Manage Inventory
Primary Actor	Admin
Secondary Actor	X
Description	Admin manages inventory, including product information and stock/quantity
Pre-Condition	User must have an Admin role
Success Scenario	Successfully creates, updates, or deletes products
Failure Scenario	Fails to create, update, or delete products

**Table 3.6 Use Case Scenario for managing data**

Use-Case Identifier	UC5: Manage data
Primary Actor	Admin
Secondary Actor	X
Description	Admin manages the data including importing and exporting data for backup and restoration.
Pre-Condition	User must have an Admin role
Success Scenario	Successfully create or restore a backup.
Failure Scenario	Failed to create or restore a backup, and rollback to last working state.

**Table 3.7 Use Case Scenario for Login**

Use-Case Identifier	UC6: Login
Primary Actor	Admin, Staff
Secondary Actor	X
Description	Users log in with correct credentials to gain access
Pre-Condition	User must exist in the database with a valid credentials and role
Success Scenario	Successful login with appropriate privileges
Failure Scenario	Failure to login

## **ii. Non-Functional Requirements**

### **1. Security**

- There is a secure login for all users with varying levels of access determined by their roles.
- Sensitive information, including details of payments and personal data, is to be encrypted to ensure against unauthorized access.
- Only administrators have full authority to change system settings and data.

### **2. Maintainability**

- The system should provide automated backup options which can help in making regular data saving and restore possible.
- Minimize system downtime with effective recovery procedures so that this system may remain operational at all times.

## **3.1.2 Feasibility Analysis**

Feasibility study is a very significant process that confirms the practicality of a project by ensuring all factors are considered before the development of a project. In this perspective, for TickBill, this study assesses the technical, operational, and economic aspects, showing clearly whether the project is viable and could be sustainable.

### **i. Operational Feasibility**

Integrating TickBill into regular business operations can be easily achieved. Since this is open source, there is no legal or operational barrier to its usage, hence it can be feasible for many industries.

### **ii. Economic Feasibility**

Since TickBill is open source in nature, it is free from all sorts of licensing and subscription costs. It is economically viable since the system is free, saving companies from the high prices of proprietary software.

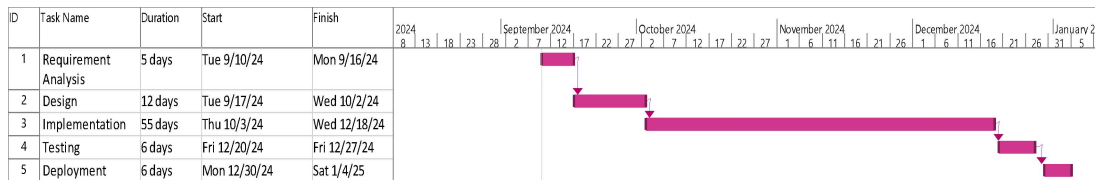
### **iii. Technical Feasibility**

The tools required for the development of TickBill are available and free of cost: Java, Swing, MySQL. The system can be deployed on multiple platforms, Windows, Linux, and macOS, hence it is technically viable.

#### iv. Schedule Feasibility

The project is feasible within a well-planned 3-months' timeline, while following the waterfall model for the system development. Here's the Gantt chart depicting the timeline for the project.

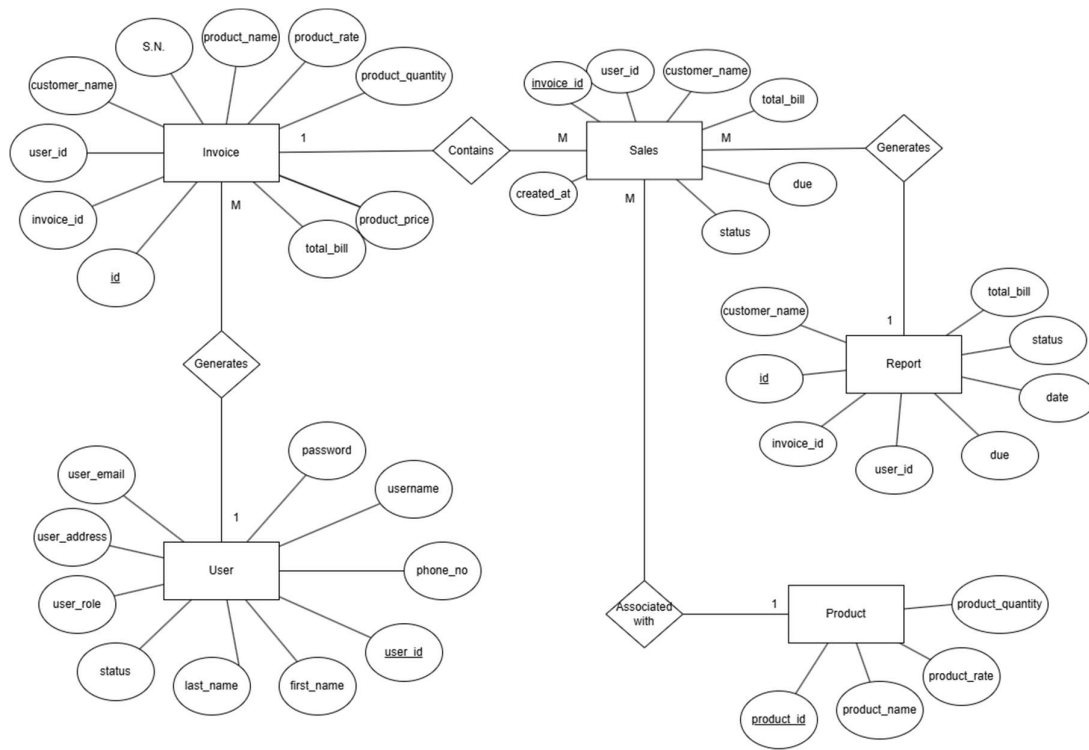
Task Name	Duration	Start	Finish
Requirement Analysis	5 days	Tue 9/10/24	Mon 9/16/24
Design	12 days	Tue 9/17/24	Wed 10/2/24
Implementation	50 days	Thu 10/10/24	Wed 12/18/24
Testing	7 days	Thu 12/20/24	Fri 12/27/24
Deployment	2 days	Mon 12/30/24	Tue 1/04/25



**Figure 3.3 Project Gantt Chart of TickBill**

### 3.1.3 Data Modeling (ER-Diagram)

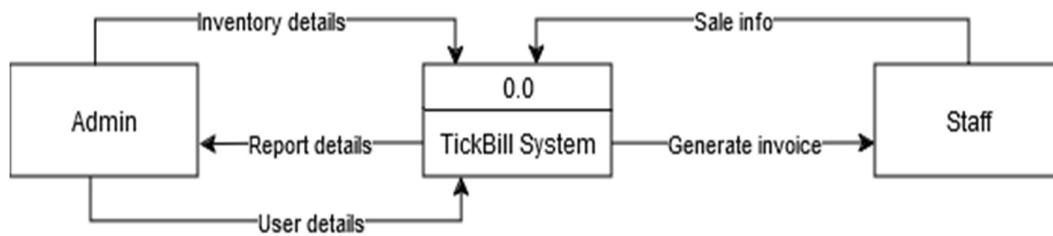
The ER Diagram of the TickBill is given below:



**Figure 3.4 ER Diagram of TickBill**

### 3.1.4 Process Modeling (DFD)

The Level 0 and Level 1 Data flow diagram of the TickBill are below:



**Figure 3.5 Level 0 DFD of TickBill**

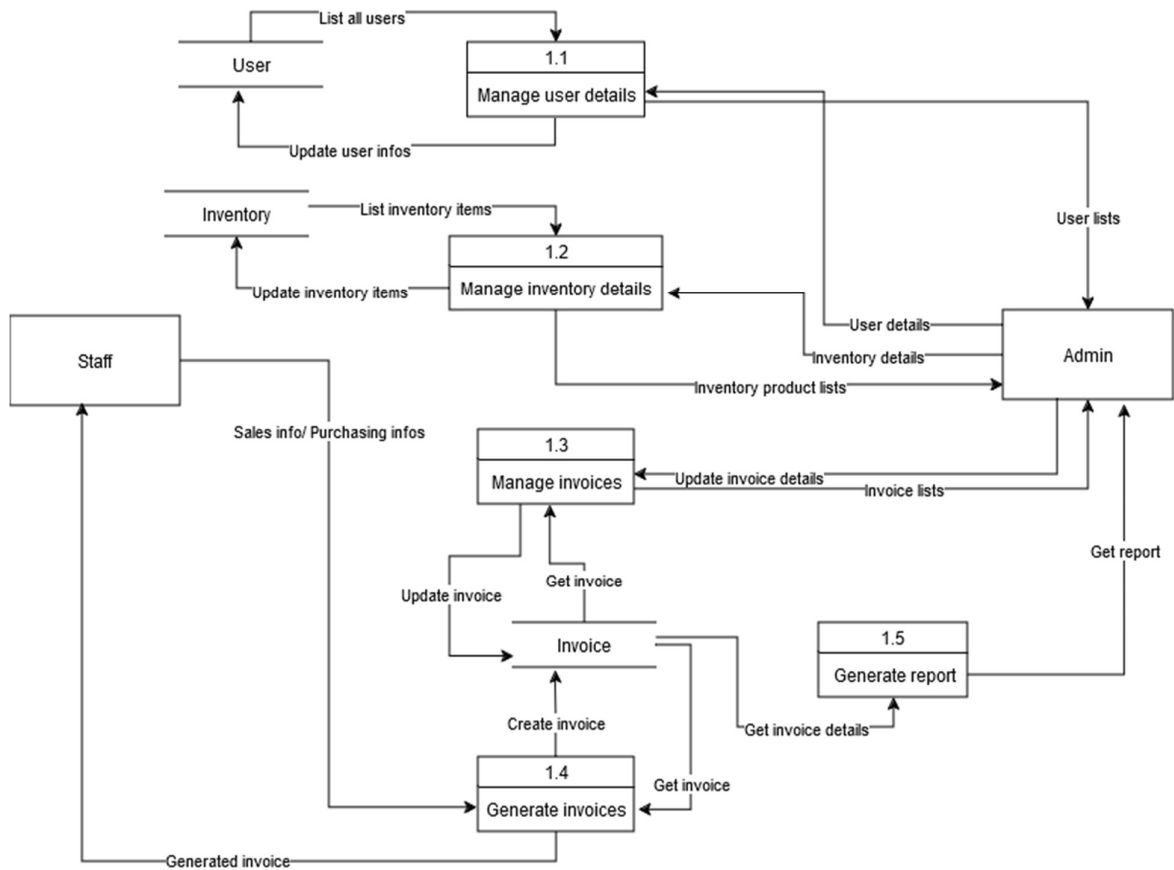


Figure 3.6 Level 1 DFD of TickBill

## 3.2 System Design

### 3.2.1 Architectural Design

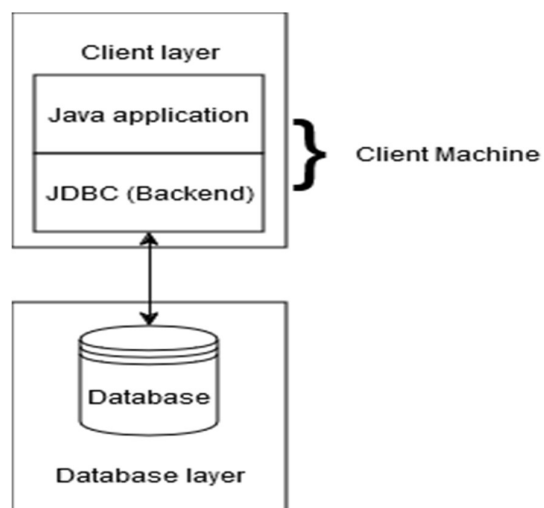


Figure 3.7 Architecture Diagram of TickBill



### 3.2.2 Database Schema Design

The database schema of the TickBill is below:

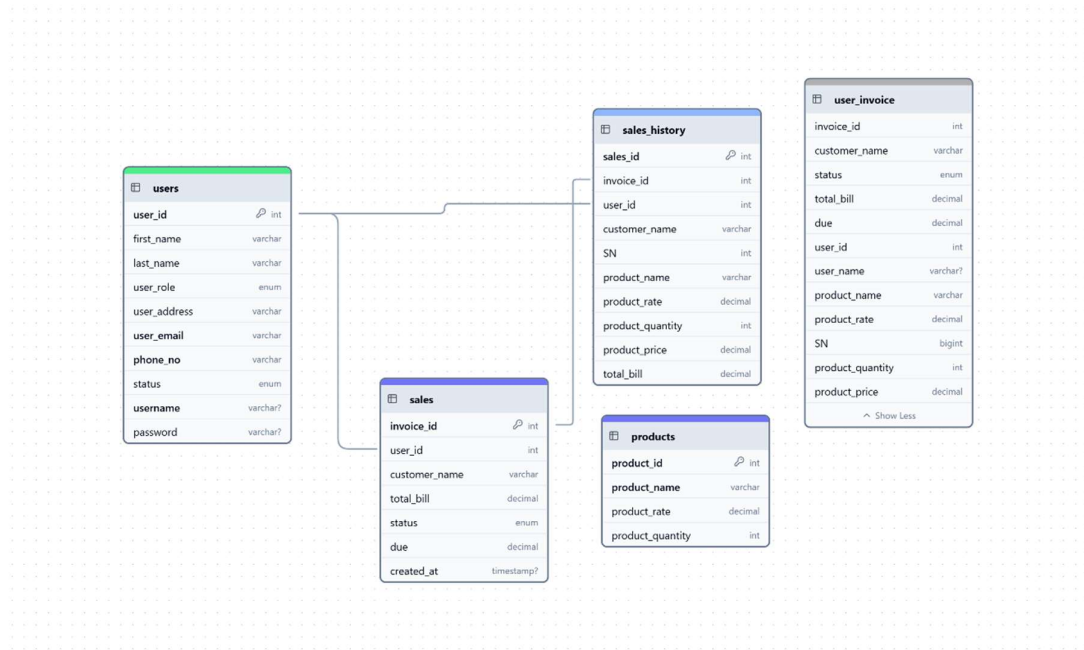


Figure 3.8 Database Schema of TickBill

### 3.2.3 Interface Design

The login panel features a light gray background with rounded corners. It contains the following elements:

- Username:** A text label followed by a white input field.
- Password:** A text label followed by a white input field.
- Port:** A text label followed by a white input field containing the value "3306".
- Login:** A rectangular button with a black border and the text "Login" in black.

Figure 3.9 Login Panel of TickBill

Name:   
Price:   
Quantity:

ID	Name	Price	Quantity

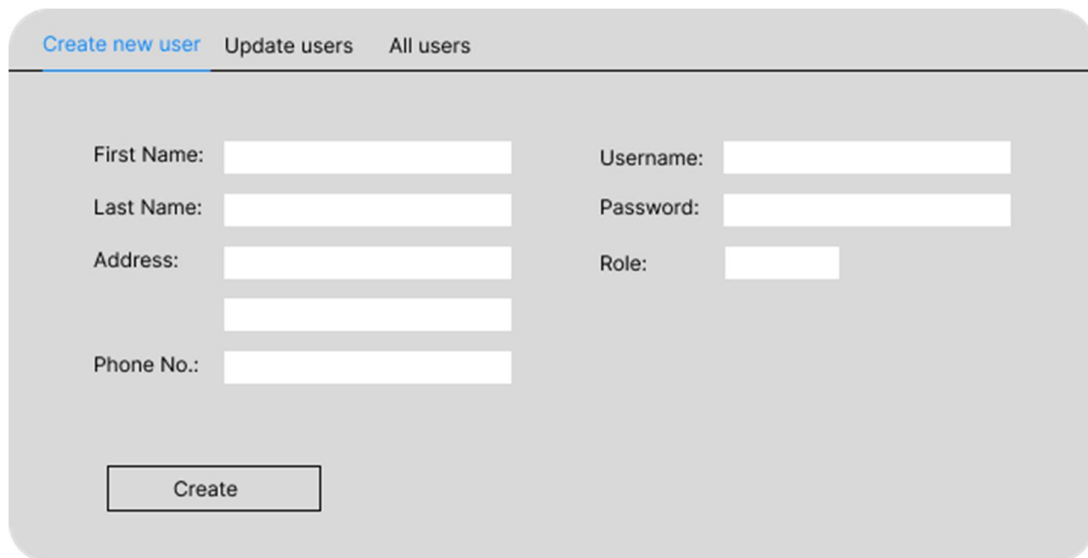
Figure 3.10 Inventory Panel of TickBill

Invoice no. XX  
Customer: 
Product: 
Quantity: 
Unit Price: 0.00
Total Price: 0.00

Invoice ID	Name	Quantity	Unit Price	Total Price

Total Amount:   
Paid Amount:   
Balance/Due:

Figure 3.11 Sales Panel of TickBill



The 'User Panel of TickBill' is a web interface for managing users. It features a top navigation bar with three tabs: 'Create new user' (highlighted in blue), 'Update users', and 'All users'. Below the tabs, the form is organized into two columns. The left column contains labels and input fields for 'First Name', 'Last Name', 'Address' (with two stacked input boxes), and 'Phone No.'. The right column contains labels and input fields for 'Username', 'Password' (with a single long input box), and 'Role' (with a single input box). At the bottom left of the form is a 'Create' button.

Create new user   Update users   All users

First Name:

Last Name:

Address:

Phone No.:

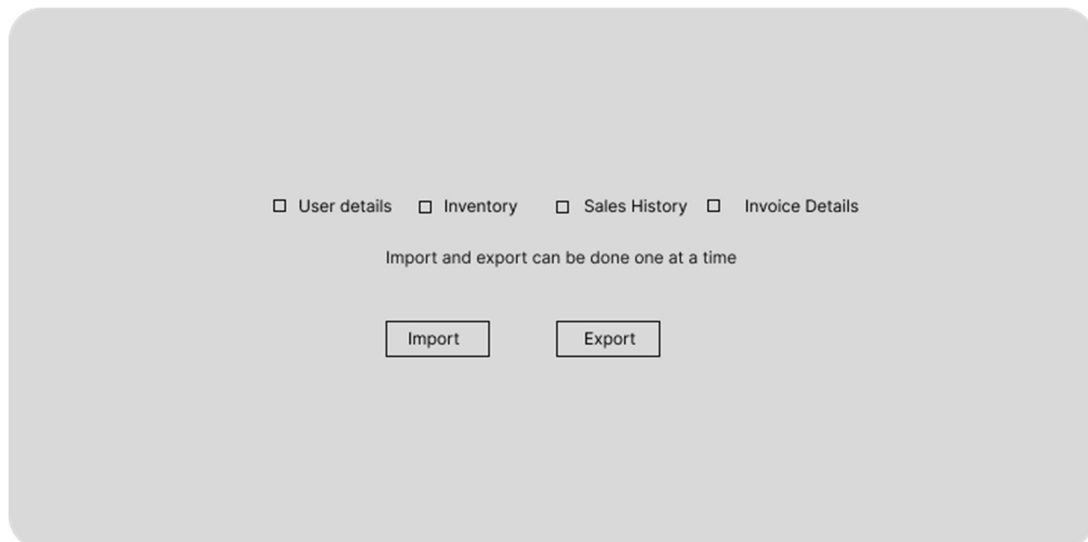
Username:

Password:

Role:

Create

**Figure 3.12 User Panel of TickBill**



The 'Import/Export Panel of TickBill' is a web interface for data management. It features a top navigation bar with four tabs: 'User details', 'Inventory', 'Sales History', and 'Invoice Details'. Below the tabs, a message states 'Import and export can be done one at a time'. At the bottom are two buttons: 'Import' and 'Export'.

☐ User details   ☐ Inventory   ☐ Sales History   ☐ Invoice Details

Import and export can be done one at a time

Import   Export

**Figure 3.13 Import/Export Panel of TickBill**

The diagram illustrates a menu-driven system interface. It features a vertical column of five menu options on the left: "Users", "Inventory", "Reports", "Sales", and "Import/Export". Each option is contained within a rectangular button. To the right of this column is a large, empty rectangular area representing the workspace for the selected function. In the top right corner of the interface, there is a "Terminate" button.

### Figure 3.14 Main Panel of TickBill

Invoice ID:

Customer Name:

Status:

Total bill:

Due amount:

Paid amount:

Search

Update

Delete

Invoice ID	Customer	Total bill	Status	Due amt	Billed by

### Figure 3.15 Invoice Panel of TickBill



## CHAPTER 4: IMPLEMENTATION AND TESTING

### 4.1 Implementation

#### 4.1.1 Tools Used

##### **Front-end Tools:**

**Swing:** Swing is a JFC library and an extension of the AWT. Java Swing offers much-improved functionality over AWT, new components, expanded components features, and excellent event handling with drag-and-drop support.

##### **Back-end Tools:**

**JDBC Driver:** JDBC is a Java API to connect and execute the query with the database. It is a specification from Sun Microsystems that provides a standard abstraction (API or Protocol) for Java applications to communicate with various databases. It provides the language with Java database connectivity standards. It is used to write programs required to access databases. JDBC, along with the database driver, can access databases and spreadsheets.

##### **Development Environment:**

**NetBeans IDE:** NetBeans IDE is a popular development environment for creating Java Swing applications which includes GUI builder with drag and drop Swing components. and the automatic code generation for the interaction with the Swing components. It also supports Layout manager including Free layout.

**IntelliJ IDEA IDE:** IntelliJ IDEA IDE is one of the most widely used IDE for Java development because it includes context-aware code completion suggests relevant options based on the current context, significantly reducing typing time and improving accuracy. It offers robust refactoring options, enabling easy code restructuring without breaking functionality.

##### **Database:**

**MySQL:** MySQL is a popular open-source relational database management system used to store and manage data. It provides a reliable and efficient solution for storing user profiles, posts, comments, and other relevant information. MySQL supports robust querying, indexing, and transactional capabilities.

### **Documentation Tools:**

**Microsoft Word:** Microsoft Word is a widely used word processing application used for creating comprehensive project documentation, including project plans, requirements, user manuals, and other project-related documents. It offers a range of formatting and editing tools to create professional documentation.

**Microsoft Project:** Microsoft Project is widely used project management software product assisting in developing a schedule, assigning resources to tasks, tracking progress and analyzing workloads. It also offers creating different figures like Gantt chart and timelines.

**Draw.io:** Draw.io is a diagramming tool used for creating visual representations of system architecture, flowcharts, and other diagrams. It is utilized to create diagrams illustrating the system's components, interactions, and data flows, aiding in the visualization and planning of the project.

**ChartDB.io:** ChartDB.io is a free and open-source diagram editors, which helps in visualizing and designing the database with a single query and help in obtaining relationship diagrams.

## **4.1.2 Implementation Details of Modules**

### **i. User Registration and Login Module**

Description:

This module enables users to register and log in to the TickBill platform. It ensures secure access by validating the user credentials and assigning the designated roles for authorized access.

Implementation:

1. **Registration Form:** A registration form collects user details such as name, email, password, contact information.
2. **Validation:** Regular expressions and String methods ensures that all the required fields in the registration form are filled, the email provided is in a valid format, and the password meets the specified standards, such as having a minimum length, and whether the user have the verified role or not while logging in.
3. **Login Form:** A login form collects user details such as username and password, and searches for the user in the database, and allowed to log in only if the user has verified tag and proper role, and matching input details.

## **ii. Admin Module**

### **Description:**

This module provides the administrator with the ability to manage the user information, inventory information, invoice information, and the backup and restoration of the data. This includes viewing, adding, updating and deleting the user details, inventory details, invoice details.

### **Implementation:**

1. **Manage user details:** Admins can create new users, view, update and delete existing user information. Updated data are fetched and updated runtime on the tabbed pane located at the side.
2. **Manage inventory details:** Admins can add new items, view, update and delete existing items in the inventory.
3. **Manage invoice details:** Admins can generate new invoice, view, update and delete existing invoices from the invoice details. The quantity in inventory is also deducted runtime respectively for that item ordered while generating invoice.
4. **View and generate reports:** Admin can view and generate reports of the desired time period finding out highest selling items and revenue.

## **iii. User Module**

### **Description:**

This module enables user to generate and view invoice details. This includes adding new invoice, and view existing invoices.

### **Implementation:**

1. **Create Invoice:** Creating invoice collects the order information such as customer name, items purchased, total bills paid and generate printable invoice which is then stored to the invoice details.
2. **View Invoice:** Shows all the invoices generated till now, along with the user who generated those invoices, and the items the customer purchased along with the total amount of bill paid.



#### iv. Backup and Restore Module

Description:

This module enables administrator to backup and restore existing data from a CSV file. This module also supports ROLLBACK and COMMIT which prevents database from any corruption or mismatched data enrolment.

Implementation:

1. Import/Restore data: Data can be imported only if the CSV are parsed properly by the CSV parser, else it will rollback to previous working state.
2. Export data: Data can be exported but only one at a time, meaning backing up user details, inventory details, invoice details but only one at a time because of the limitation of exporting file using FileChooser in Java Swing. Data exported have the column names as well for making it easier for the parser to import data later on in future.

#### 4.2 Testing

Software testing evaluates the applications and systems to identify variations between expected and actual performance, and verifies that it meets the requirements defined. Main goals of software testing are:

- **Verification:** Confirming the software functions correctly according to its design and specifications.
- **Validation:** Assurance that the software meets user needs and expectations effectively.
- **Defect Identification:** Identifying bugs, errors, and other defects that might cause programs to behave incorrectly or unexpectedly.
- **Quality Assurance:** Contribute to overall quality by ensuring standards and best practices are met.
- **Risk Mitigation:** Early determination and reduction of risks regarding security vulnerabilities, performance bottlenecks, or usability problems.

#### 4.2.1 Test Cases for Unit Testing

**Table 4.1 Test Cases for User Login & Registration**

SN	Test Case	Test Input	Expected Result	Outcome	Result
1	Sign up/Registration with valid details	First name: john Last name: doe Email: johndoe@email.com Username: john Password: John@123	New user is created, and data is stored in database with status unverified.	As expected	PASS
2	Registration with invalid details	Empty input	Dialog pops up saying first name is empty and data is not stored in database	As expected	PASS
3	Login with valid credentials	Username: john Password: John@123	If user is unverified, dialog with "User unverified" is shown. If user is verified, user is logged in successfully redirect to Main panel with only required buttons enabled.	As expected	PASS
4	Login with invalid credentials	Username: john Password: John@12345	Dialog pops up saying "Invalid username or password"	As expected	PASS

**Table 4.2 Test Cases for Invoice Generation**

SN	Test Case	Test Input	Expected Result	Output	Result
1	Generate invoice using valid inputs	Customer Name: testuser Enter required product with product quantity Add to Cart Pay & Print	Invoice is generated.	As expected	PASS
2	Generate invoice using invalid inputs	Empty customer name Add products and quantity Add to Cart	Dialog popup showing customer name is required	As expected	PASS

**Table 4.3 Test Case for Admin Login**

SN	Test Case	Test Input	Expected Result	Outcome	Result
1	Login with valid credentials	username: admin password: admin123	User is redirected to Main panel with access to all the pages.	As expected	PASS
2	Login with invalid credentials	username: admin password: admin12334	Dialog showing "Username or password invalid" is shown and user is not redirected to Main panel	As expected	PASS

**Table 4.4 Test Cases for User Management**

SN	Test Case	Test Input	Expected Result	Outcome	Result
1	Create new user with valid input	First name: Ram Last name: Kumar Address: Kathmandu Email: ram.kumar@gmail.com Phone no: 9876543345 Role: Employee Status: Verified	New user is created and stored in database	As expected	PASS
2	Create new user with already existing username	First name: Ram Last name: Shrestha Address: Kathmandu Email: ram.shrestha@gmail.com Phone no: 9876543335 Role: Employee Status: Verified	Dialog showing duplicate phone number is shown	As expected	PASS
3	Create user with invalid inputs	Email: j@j.com	Dialog showing invalid email is shown	As expected	PASS
4	Update existing user information	First name: Ram Last name: Kumar Address: Bhaktapur Email: ram.kumar@gmail.com Phone no: 9876543345 Role: Employee Status: Verified	Updated user information is shown and stored in database	As expected	PASS
5	Delete existing user	Input user ID of the existing user to delete	User is deleted and removed from database	As expected	PASS

**Table 4.5 Test Case for Product Management**

SN	Test Case	Test Input	Expected Result	Outcome	Result
1	Create new product	Product name: Lasssi Product Rate: 120 Product Quantity: 50	New product is created and stored in database	As expected	PASS
2	Create new product with invalid details	Product Rate: aa	Error dialog showing invalid column is shown, and data is not stored in database	As expected	PASS
3	Update existing product	Enter product name of existing product Click Search Update required field	Updated product details are shown and datas are updated in database	As expected	PASS
4	Delete existing product	Enter name of the product to delete	Product is deleted from database and table	As expected	PASS

**Table 4.6 Test Case for Invoice Management**

SN	Test Case	Test Input	Expected result	Outcome	Result
1	Update existing invoice	Search the invoice using invoice ID Update required fields	Updated invoice is shown and data in database are updated	As expected	PASS
2	Remove existing invoice	Input the invoice ID Click on Delete button	Invoice with input invoice ID is deleted in database and table	As expected	PASS

**Table 4.7 Test Case for Report Generation**

SN	Test Case	Test Input	Expected Outcome	Outcome	Result
1	Show existing data	Input the required date Click on Show	Data from user specified dates are shown	As expected	PASS
2	Generate report	Input the required date Click on Generate	Printable output as same as table is shown	As expected	PASS

**Table 4.8 Test Case for Import/Export data**

SN	Test Case	Test Input	Expected Outcome	Outcome	Result
1	Backup existing data	Click on the checkbox required to backup Selected the directory to save along with filename	Data are saved in the user selected directory in CSV format	As expected	PASS
2	Restore existing data	Click on the checkbox required to restore Select the user requested CSV file	Data are restored on top of existing data if the CSV file is parsed properly else it fails and rollback stating invalid CSV file	As expected	PASS

**4.2.2 Test Cases for System Testing****Table 4.9 Test Cases for System User Interface and Navigation**

SN	Test Case	Test Input	Expected Outcome	Outcome	Result
1	Navigation between pages	Clicking on navigation buttons	User navigates smoothly without any error	As expected	PASS
2	Terminate the session	Click on terminate button at top right corner	Session terminates and user is redirected to login page	As expected	PASS

**Table 4.10 Test Cases for System Functionality**

SN	Test Case	Test Input	Expected Outcome	Outcome	Result
1	Invoice generation	Provide valid input and generate invoice	A printable invoice is generated and data are stored in database	As expected	PASS
2	Create and manage users	Create new user with valid details and manage the information	User with provided details are created and stored in database	As expected	PASS
3	Create and manage inventory	Create new product with valid details and manage the information	Product with provided details are created and stored in database	As expected	PASS

#### **4.2.3 Integration Testing**

Integration testing is a testing process where two or more program units, previously tested individually, are combined and tested in various ways. This level of testing mainly deals with the interfaces between components. In our system, several modules and components have been integrated. For instance, the sales module is designed in such a way that it would work perfectly fine only if integrated with the login module. These integrations are tested to optimize performance under the maximum capacity of the system.

#### **4.2.4 System Testing**

System testing is where each of the individual pieces is built into one complete system and then tested to see that everything works. This type of test reveals if the pieces are compatible, work well with each other, and provide proper data back and forth through the system interfaces. In our situation, system testing was conducted with precision so that the system components work well with each other and work without fault like navigation between the pages.

## **CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS**

### **5.1 Lesson Learnt / Outcome**

The development of TickBill has been an educational experience, and it has reinforced the necessity for open-source software for small businesses. While developing this system, we have gained considerable knowledge in software development, database design, graphical user interface design, and combining ticketing, billing, and inventory management functionality. We came to know how important role-based access control (RBAC) is in sustaining security and what kind of adequate testing strategies must be adopted for enhancing system dependability.

Some of the key deliverables of the project are:

- Developing a cost-effective, open-source alternative to proprietary POS systems.
- Implementing real-time monitoring of inventory to prevent stock-out or overstocking.
- Enforcing role-based access control to guard sensitive business information.
- Developing an easy-to-use and user-friendly interface for seamless interaction.
- Providing support to businesses to export and import data to enhance data management and backup features.

### **5.2 Conclusion**

TickBill successfully overcomes the limitations of the traditional proprietary Point of Sale systems by offering an open-source, flexible, and economically viable solution. The system offers small businesses the ability to manage their sales, inventory, and customer management in an effective way without the burden of recurring subscription fees. The project not only achieves its initial objectives but also lays a solid foundation to further enhancements.

The robust utilization of Java Swing, MySQL, and JDBC enables the application to easily operate in offline mode, which is very suitable for small and medium enterprises. The module-based design makes customization easy, and companies can mold the system based on their needs. In addition, strict testing has validated the functionality, security, and performance of the system, providing a solid and scalable solution.



### **5.3 Future Recommendations**

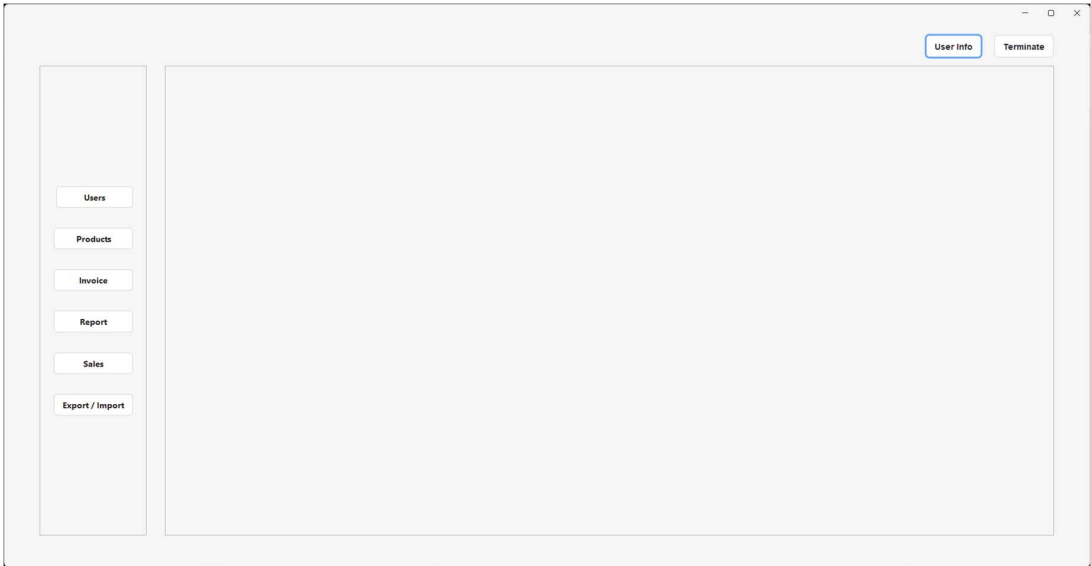
Although TickBill offers a good platform, there are various enhancements that can make it even more capable. Online payment integration, cloud-based synchronization, and mobile app support are some of the future features that can be added. More advanced reporting and analytics would give businesses greater insights, while multi-language support could increase its reach. Support for integration with accounting and CRM software, stronger security features, and increased scalability would make it even more useful.

By incorporating these improvements, ssssTickBill can evolve into a more sophisticated and powerful POS system while maintaining its open-source advantage to meet the growing demands of small businesses. Continuous development and fine-tuning will make it relevant and efficient in the long run in an evolving market.

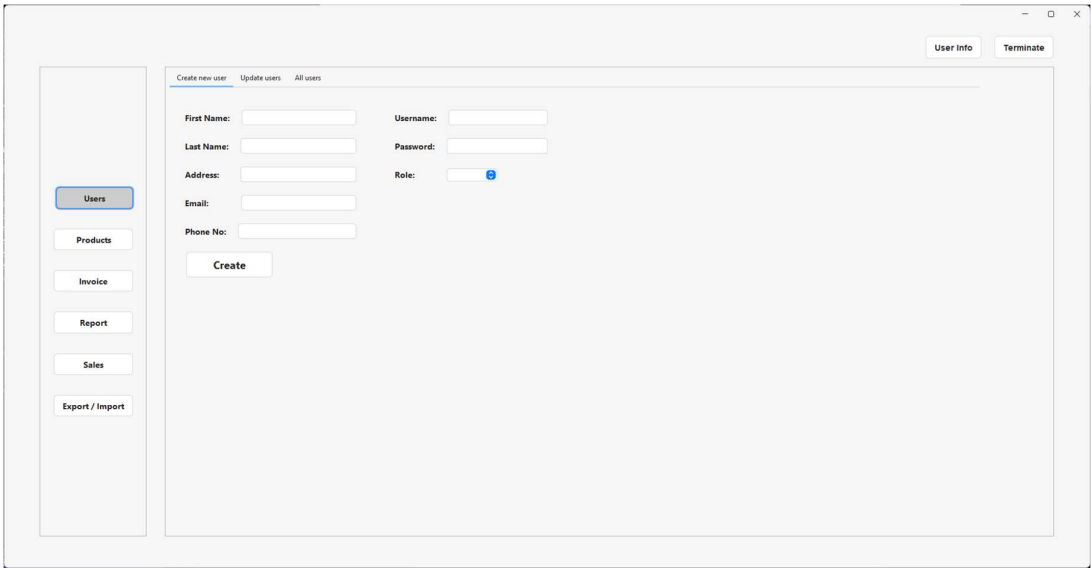
## REFERENCES

- [1] H. Asrani, S. Vishwakarma, D. Asrani, and Dr. K. Asrani. (Mar. 2024). *Point of Sale Systems*. B.N. College of Engineering and Technology, Lucknow, India: International Journal of Innovative Research in Computer Science & Technology (IJIRCST).
- [2] Kulandairaj, A. J. (Jan.-Feb. 2015). *The effectiveness of point of sales (POS) displays in the buying behaviour of consumers*. Loyola College. Chennai-34, Tamil Nadu, India: International Journal of Technical Research and Applications.

# APPENDICES



Main Panel



User Panel

Login

Username:

Password:

Login

Login Panel

User info

Terminate

Users

Products

Invoice

Report

Sales

Export / Import

Name:

Price:

Quantity:

Create

Search

Update

Delete

ID	Name	Price	Quantity
1	French Fries	19.99	150
2	Fried Chicken	29.5	60
3	Burger	15.0	189
4	Steak	45.0	8

Product Panel

Users

Products

Invoice

Report

Sales

Export / Import

INVOICE NO. 3

Customer:  Product:  Quantity:  Unit Price: 00.00 Total Price: 0.00

Invoice ID	Name	Quantity	Unit Price	Total Price
------------	------	----------	------------	-------------

Total Amount:

Paid Amount:

Balance / Due:

Add to Cart

Remove

Remove All

Pay & Print

## Sales Panel

Users

Products

Invoice

Report

Sales

Export / Import

Invoice ID:

Customer Name:

Status:

Total Bill:

Due Amount:

Paid Amount:

Search

Update

Delete

Invoice ID

Customer

Total bill

Status

Due Amount

Billed by (User ID)

1	test	\$15.50	Pending	\$15.50	1
2	test	200.00	Pending	200.00	1

## Invoice Panel

S.N.	Item	Rate	Qty	Total
1	Fried Chicken	29.50	2	59.00
2	Fried Chicken	29.50	2	59.00
3	Steak	45.00	2	90.00

Invoice ID

2

Customer Name:

asdf

Status:

Pending

Total Amount:

208.00

Paid Amount:

0.0

Due Amount:

208.00

Billed by (User ID) :

1

Per User Invoice Panel

Users

Products

Invoice

Report

Sales

Export / Import

User Info

Terminate

☐ User Details

☐ Inventory

☐ Sales History

☐ Invoice Details

Import and export can be done one at a time

Import

Export

Import / Export Panel

Users

Products

Invoice

Report

Sales

Export / Import

Date: 09-01-2025 - 09-01-2025

Show

Generate

Total revenue: 0.0

Loss 0.0

Highest selling product: No data available

SN	Customer	Total Amount	Paid Amount	Due Amount	Status
----	----------	--------------	-------------	------------	--------

Report Panel