**CHAPTER 1**

**INTRODUCTION**

**1.1 About the Project:**

The "Sri Mens Wear Billing System" is a robust Java-based application tailored to streamline billing and stock management processes within a men's wear store. The system features two pivotal user roles—Admin and Staff—each equipped with specialized functionalities to enhance operational efficiency.

Admin privileges encompass managing the checkout list, generating bills using stock IDs, overseeing the current stock list, adding or deleting stock items, and facilitating the creation of Staff IDs. Additionally, Admin has the capability to edit and maintain accurate product details, ensuring a dynamic and up-to-date inventory.

Staff responsibilities are centered around billing operations, viewing the stock list, actively managing stock by adding or deleting items, and receiving real-time notifications for products reaching zero quantity.

Utilizing the Java programming language and leveraging the MySQL database management system, the system prioritizes seamless data handling and retrieval. The user interfaces for both Admin and Staff are intuitively designed to provide a user-friendly experience.

An added feature to the system is the "Create Staff ID" module, empowering the Admin to systematically generate Staff IDs and manage essential details for each staff member, further enhanci*ng* staff administration.

## CHAPTER 2

## SYSTEM STUDY

**2.1 Existing System:**

The current Sri Men's Wear system is semi-automated, managing purchase and sales operations using a software package with modules like Billing, Stockist,and Out of Stock. However, critical tasks such as Adding stock, stock notifications, creating purchase orders, and processing payment requests are handled manually. Physical processes, including collecting materials from couriers, distributing them, and maintaining stock levels, also rely on manual efforts.

**2.2 Disadvantages of Existing System**

* Poor stock management can disrupt supply chains, affect product availability, and impact customer satisfaction by not providing real-time insights into inventory levels.
* Poor staff ID and password management can result in confusion and security risks.
* Manual payment processing can be time-consuming and prone to errors. Handling transactions without automated systems may result in delays, increased workload, and a higher risk of inaccuracies in financial records.

**2.3 Proposed System**

the Sri Mens Wear Billing System focus on key areas, including advanced stock management, streamlined staff details management, and detailed checkout processes. The system aims to offer intelligent stock control, allowing for efficient tracking and forecasting. Simultaneously, it provides improved management of staff details, ensuring a systematic approach. The checkout process is designed for detailed transaction tracking, enhancing the overall customer experience. These targeted improvements aim to bring efficiency, organization, and a better shopping experience to the men's wear store.

**2.4 Advantages of Proposed System**

* Sri Mens Wear Billing System provide a streamlined approach. Admins can effortlessly add new products, remove outdated items, and make necessary edits, ensuring a dynamic and well-maintained inventory for efficient store management.
* Incorporates an intelligent forecasting algorithm to optimize inventory levels, minimizing overstock and out-of-stock situations.
* Strengthens user authentication mechanisms, ensuring secure access to the system.
* The improved staff details management in the proposed Sri Mens Wear Billing System enables seamless tracking of essential information like names, contacts, usernames, and passwords. This feature ensures a secure and organized approach to managing staff data, enhancing overall efficiency in store administration.

**2.5 Problem Definition and Description**

stock and staff management, as well as the checkout process. The stock management inefficiencies, such as the lack of a streamlined system for adding, deleting, and editing products, hinder the store's ability to maintain an accurate and up-to-date inventory. Additionally, the shortcomings in staff details management, including inadequate tracking of essential information like names, contacts, and login credentials, create organizational challenges. Moreover, the checkout process lacks detailed tracking, potentially affecting the customer experience. Addressing these issues is paramount to enhancing the overall efficiency, organization, and customer satisfaction within the Sri Mens Wear store.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 Packages Selected**

**Front End:** Java Swing(JavaFX)

**Back End:** MySQL

### 3.2 Resources Required

### 3.2.1Hardware Resources

* + - * **Processor :** Intel® Code™ i3-6006U CPU @ 2.00GHz 2.00GHz
      * **Memory :** 2 GB
      * **Disk space :** 1.5 GB of free disk space
      * **Screen Resolution** : 1024 X 768

### 3.2.1Software Resources

* + - * **Operating System** : Windows 10 / Linux
      * **Software**: :JAVA(jdk 14)
      * **DataBase :** Xampp 8.2.12 version (MySQL)

### 3.3 Data Flow Diagram

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

### Symbols Used in DFD

|  |  |
| --- | --- |
| **Symbols** | **Purpose** |
|  | External entities |
|  | Process: A transaction of information that resides within the bounds of the system to be module. |
|  | Dataflows |
|  | Database: A repository of data that is to be stored for use by one or more processes, may be as simple as buffer of queue or as a relational database. |

LEVEL 0:



# Figure 3.3.1 DFD Level 0

# LEVEL 1

# 

# 

# 

# Figure 3.3.2 Leval 1-Admin

# Level 2:

# 

# Figure 3.3.3 Level2-staff

## CHAPTER 4

## SYSTEM DESIGN

### 4.1 Architectural Design

### 

### Figure 4.1.1 Architecture

### 4.2 I/O Form Design

### Billing Form :

### 

### Fig 4.2.1 Bill I/O

### Purpose of the Form:

### The Billing Form streamlines transactions in the "Billing" module, allowing biller to select to input product details, apply discounts, and generate detailed invoices seamlessly.

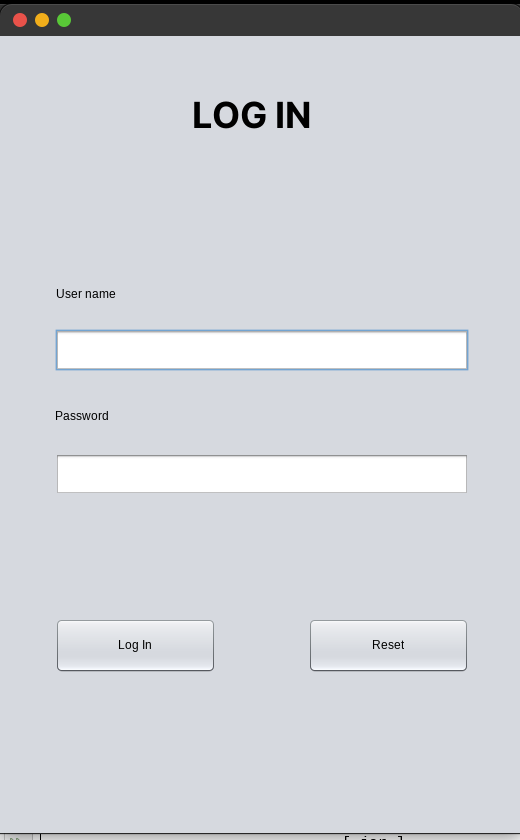
### Add/Delete/Edit Product

### Figure 4.2.2 add/delete I/O

### Purpose of the Form:

The Add/Delete/Edit form plays a pivotal role in maintaining an accurate and up-to-date product stock in the Sri Mens Wear Billing System. These functionalities empower the system administrator to adapt to changes in the product lineup, eliminate outdated items, and ensure the precision of the stock database for seamless billing and inventory management.

**Login Form**



### Figure 4.2.3 login I/O

### Purpose of the Form:

### The "Login" form serves as a secure gateway, allowing authorized administrators and Staff to access the Sri Men's Wear Billing System. It provides a crucial layer of authentication, ensuring data privacy and system security by verifying user credentials before granting access to the application.

### 4.3 Tables

**TABLE NAME:** LOGIN

**PYMERY KEY :** userName

**DESCRIPTION:** To store the users detiles

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Width** | **Constraints** |
| userName | Varchar | 10 | Primary Key |
| passWord | Varchar | 10 | Not Null |
| Staff Name | Varchar | 10 | Not Null |
| Staff Contact | Int | 10 | Not Null |

Table 4.3.1 login

**TABLE NAME:** Product List

**PYMERY KEY :** Product id(P\_ID)

**DESCRIPTION:** To store Product detiles

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Width** | **Constraints** |
| P\_ID | Int | 20 | Primary Key |
| p\_name | Varchar2 | 70 | Not Null |
| p\_qty | Number | - | Not Null |
| p\_price | Number | - | Not Null |

Table 4.3.2 Stock list

**TABLE NAME:** CHECKOUT

**PYMERY KEY :** Number(No)

**DESCRIPTION:** To store checkout detiles

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Width** | **Constraints** |
| No | Varchar2 | 20 | Primary Key |
| no\_product | Varchar2 | 70 | Not Null |
| order\_date | Number | - | Not Null |
| order\_time | Number | - | Not Null |
| amount | Varchar2 | 2 | Not Null |
| pay\_type | Varchar2 | 10 | Not Null |

Table 4.3.3 checkout

### 4.4 Entity Relationship Diagram

An entity–relationship model is the result of using a systematic process to describe and define a subject area of business data. It does not define business process; only visualize business data. The data is represented as components (entities) that are linked with each other by relationships that express the dependencies and requirements between them, such as: one building may be divided into zero or more apartments, but one apartment can only be located in one building. The three schema approach to software engineering uses three levels of ER models that may be developed.

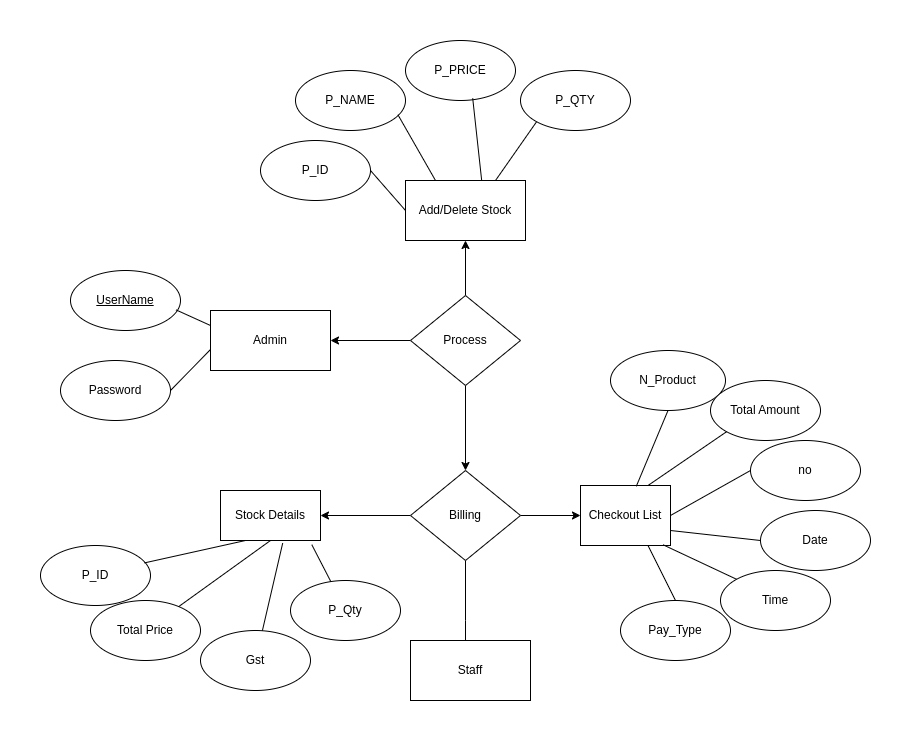


Figure 4.4.1 ER-diagram

## CHAPTER 5

## SYSTEM DEVELOPMENT

### 5.1 Functional Documentation:

### The Sri Mens Wear Billing System is a Java-based application designed to optimize the billing, stock management, and staff administration processes within a men's wear store. The system caters to two primary user roles: Admin and Staff. Admin holds authority over critical functions such as managing the checkout list, generating bills using stock IDs, overseeing the stock list, and creating Staff IDs. Staff members are responsible for billing operations, stock management, and receiving notifications for out-of-stock items.

### The Key Modules Of Sri Men’s Wear Bill System Modules

* + - Billing Module
    - Stock list Module
    - Checkout List Module
    - Add/Delete Product Module
    - OutofStock List Module
    - Create Staff ID

Billing Module:

1. This module allows both Admin and Staff to generate bills using stock IDs. It incorporates a user-friendly interface for quick and efficient billing processes.
2. Staff members can initiate bills during customer transactions, while Admin can also access this functionality for various purposes.

Stockist Module:

* Users, both Admin and Staff, can access this module to view the current stock list. The interface presents detailed information about available products, quantities, and relevant details.
* This module aids in proactive inventory management by providing real-time insights into stock levels.

Checkout List Module:

* This module showcases detailed information about completed checkouts. It serves as a reference for Admin and Staff to review and cross-verify transactions.
* The intuitive display includes information such as items purchased, quantities, and total amounts.

Add/Delete Stock Module:

* Admin can utilize this module to add new products to the stock. The interface prompts the user to input essential details such as product name, category, and quantity.
* Admin holds the capability to remove outdated or no longer available products from the stock. The module prompts the user to select the product to be deleted.
* This functionality maintains the accuracy of the stock database.

Out of Stock Module:

* This module provides a list of products with zero quantity, indicating items that are currently out of stock.

Create Staff ID:

* Admin exclusively accesses this module to create Staff IDs. The interface collects essential details for each staff member, including their name and role.
* The module ensures a systematic and secure approach to staff administration within the system.

### **5.2 Special Features of the Language**

### **1.Java :**

### Java boasts several special features that contribute to its widespread use and popularity. It is renowned for its platform independence, allowing code to run on any device with a Java Virtual Machine (JVM). Following an object-oriented paradigm, Java emphasizes modularity and code reusability. Its strong typing enhances code reliability, while automatic memory management through garbage collection simplifies memory handling for developers. The language offers a rich standard library (Java API) covering diverse functionalities, supports multi-threading for concurrent execution, and excels in exception handling, promoting stable applications. Security features, including applet sandboxing, make Java suitable for internet-based deployments. The language's documentation is extensive, aiding developers in understanding and utilizing its features, and the active community provides ample resources for problem-solving and continuous learning, contributing to Java's longevity and versatility in application development.

### **2.MySQL:** MySQL, an open-source relational database management system, stands out with its scalability, ACID compliance, robust performance optimization, support for replication, diverse data types, triggers, and stored procedures. Security features, including user authentication and SSL/TLS support, contribute to a reliable and secure database environment. The active community provides ample resources, and MySQL's compatibility across various platforms and programming languages further solidifies its position as a versatile and widely adopted RDBMS solution.

3. Java Editor:

### IntelliJ IDEA and NetBeans are integrated development environments (IDEs) widely used for Java projects. IntelliJ IDEA, developed by JetBrains, offers a rich set of features, including advanced code assistance, intelligent code completion, and a powerful debugging toolset. Its intuitive interface enhances developer productivity, and it supports various frameworks and technologies. On the other hand, NetBeans, an open-source IDE, is known for its ease of use and strong support for Java development. It provides seamless integration with Maven, excellent GUI design tools, and a modular architecture. Both IDEs offer strong Java support, with IntelliJ IDEA emphasizing productivity and NetBeans focusing on simplicity and extensibility, catering to diverse preferences within the Java development community.

### **4.Xampp MySQL Server :**

### XAMPP's MySQL Server, part of the XAMPP web server solution, simplifies the setup of a local development environment by bundling a pre-configured MySQL database management system with Apache, PHP, and Perl. With a user-friendly interface, it facilitates database administration, table creation, query execution, and user permission management for local testing and development of MySQL-dependent applications. The version number (e.g., 4.X) denotes specific releases, indicating potential improvements and updates to the bundled MySQL server, making it a convenient tool for web developers to prototype and test database-driven applications locally before deployment.

## **CHAPTER 6**

## **TESTING**

### **6.1 Types of Testing Done**

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, designing and coding.

TESTING OBJECTIVES

1. Testing is process of executing a program with the intent of finding an error.

2. A good test case design is one that has a probability of finding an as yet undiscovered error.

3. A successful test is one that uncovers an as yet undiscovered error.

These above objectives imply a dramatic change in view port.

1. Unit test

2. Integration test

3. Validation Test

4. System Test

### **1 Unit Testing**

### The first test in the development process is the unit test. The source code is normally divided into modules, which in turn are divided into smaller units called units. These units have specific behavior. The test done on these units of code is called unit test*.* Unit test depends upon the language on which the project is developed. Unit test*s* ensure that each unique path of the project performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### **2 Integration Test**

### Integration tests verify the interaction between different units or components of a software system. Unlike unit tests, which test individual units in isolation, integration tests focus on testing the interfaces and interactions between these units. Integration tests ensure that the integrated components work together as expected and that they can communicate and exchange data correctly..

### **3 Validation Test**

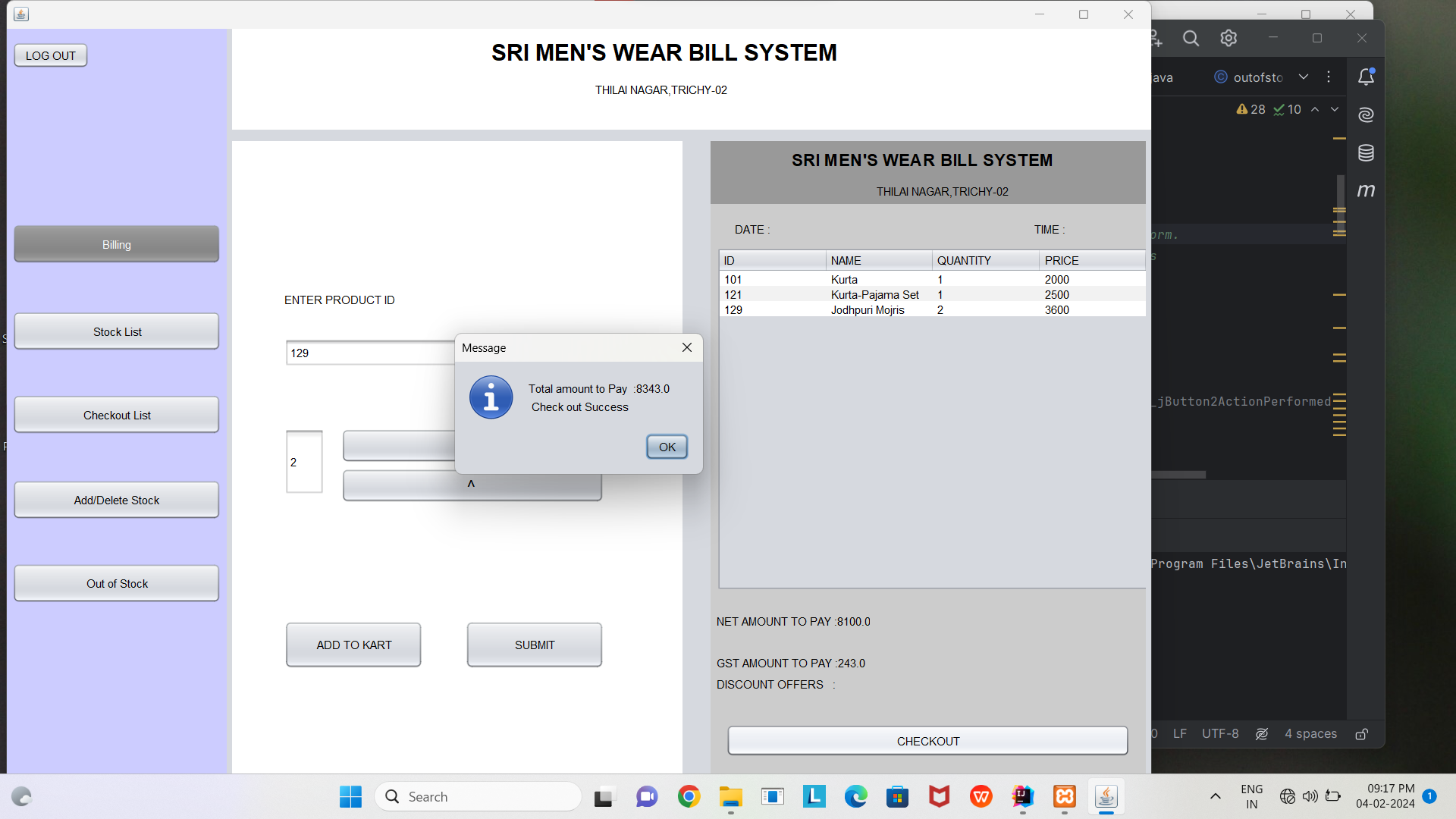
### Validation tests, also known as acceptance tests or user acceptance tests (UAT), are conducted to validate whether a software system meets the specified requirements and satisfies the needs of the end-users. These tests are typically performed by stakeholders, clients, or end-users to determine if the software meets their expectations and if it fulfills the intended purpose. Validation tests focus on the overall functionality, usability, and performance of the software from a user's perspective

### **4 System Test**

### System tests evaluate the behavior of a complete and integrated software system. Unlike integration tests that focus on the interaction between individual components, system tests assess the entire system's functionality, performance, and compliance with requirements. System tests validate the system as a whole, including its interfaces with external systems, databases, and user interfaces. These tests ensure that the software system meets all specified requirements and functions correctly in its intended environment.

### **6.2 Test Data and Output**

**Billing Process**

Figure6.2.1 test-bill

**Add Stock**

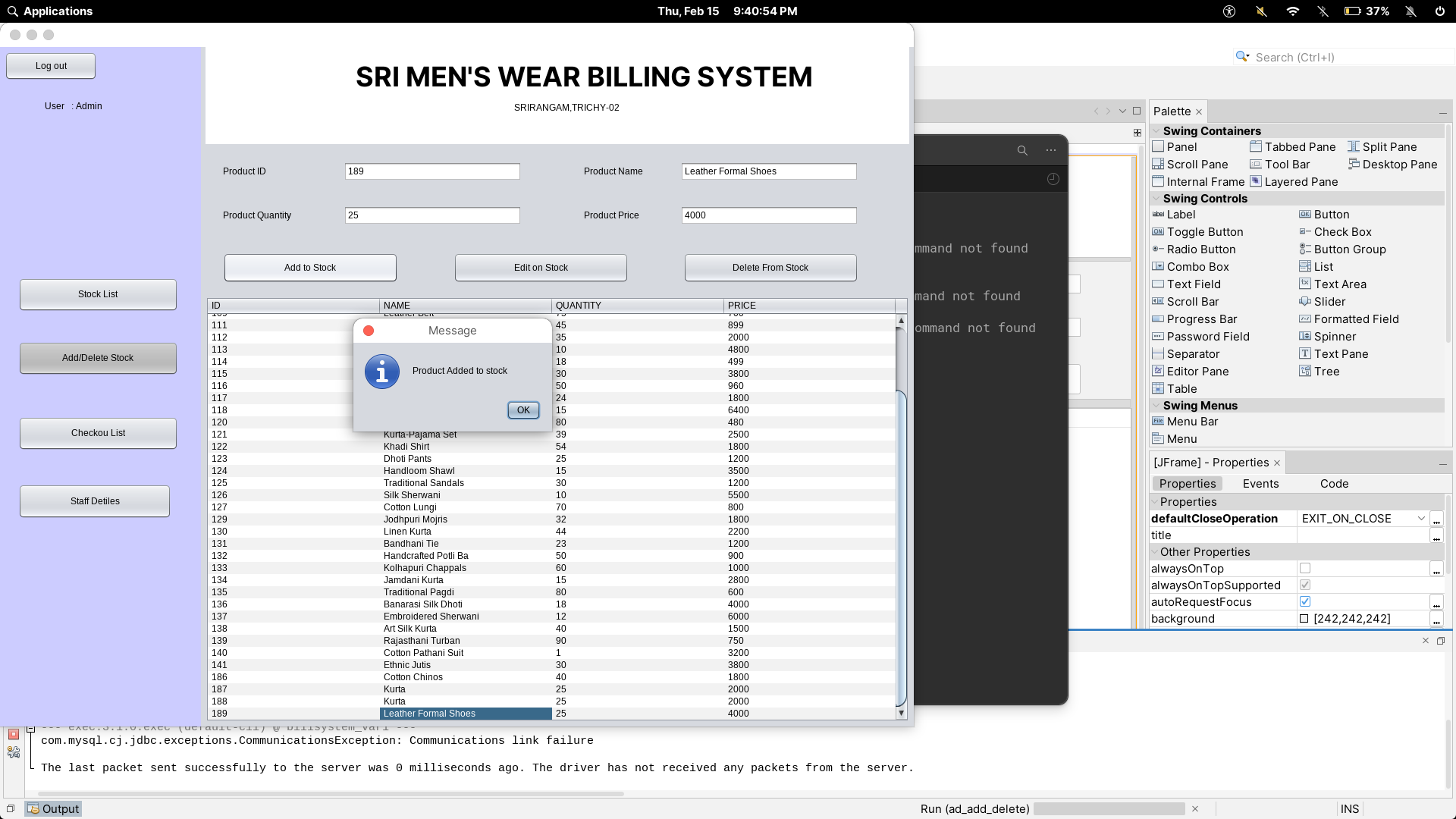
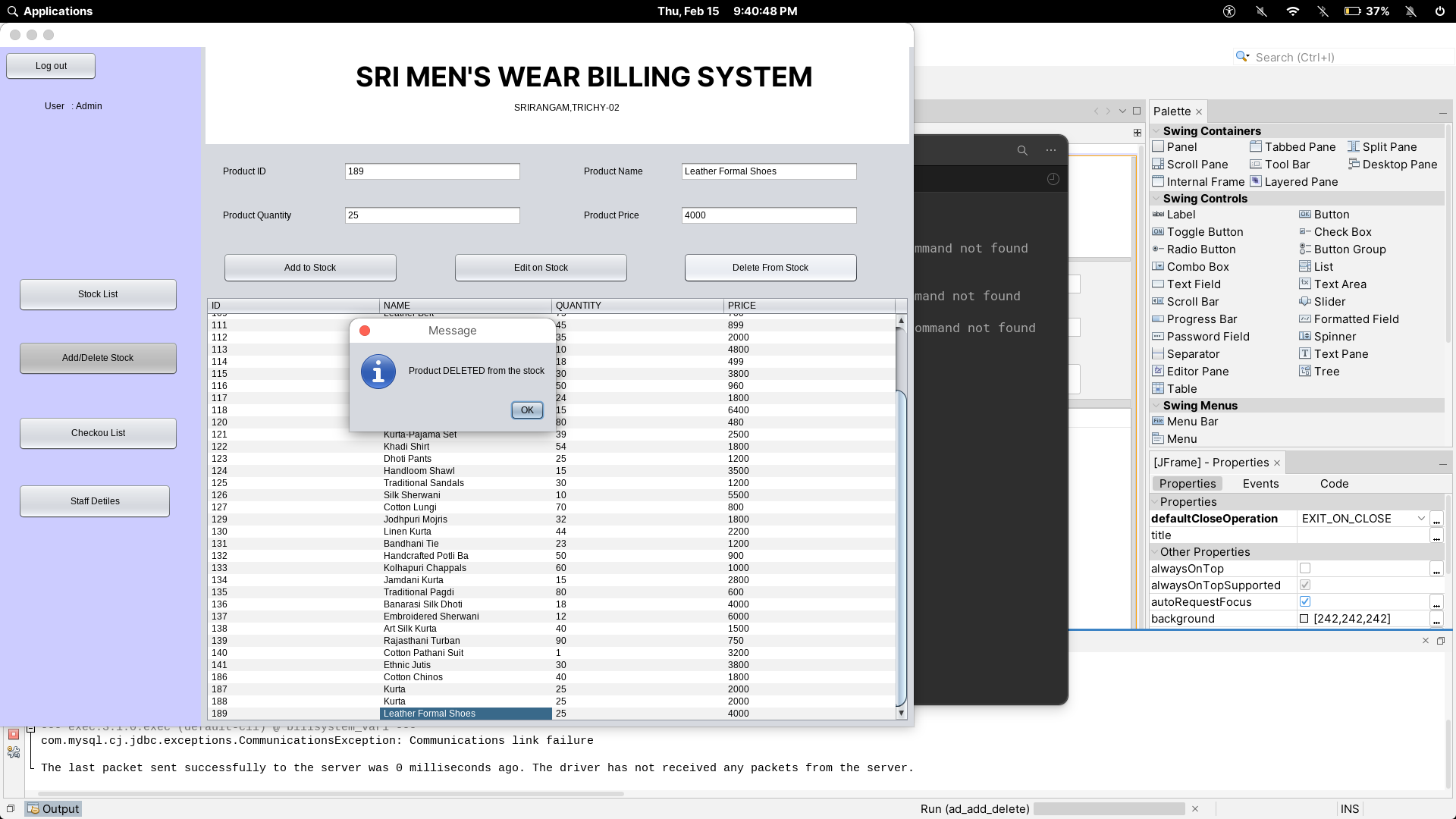
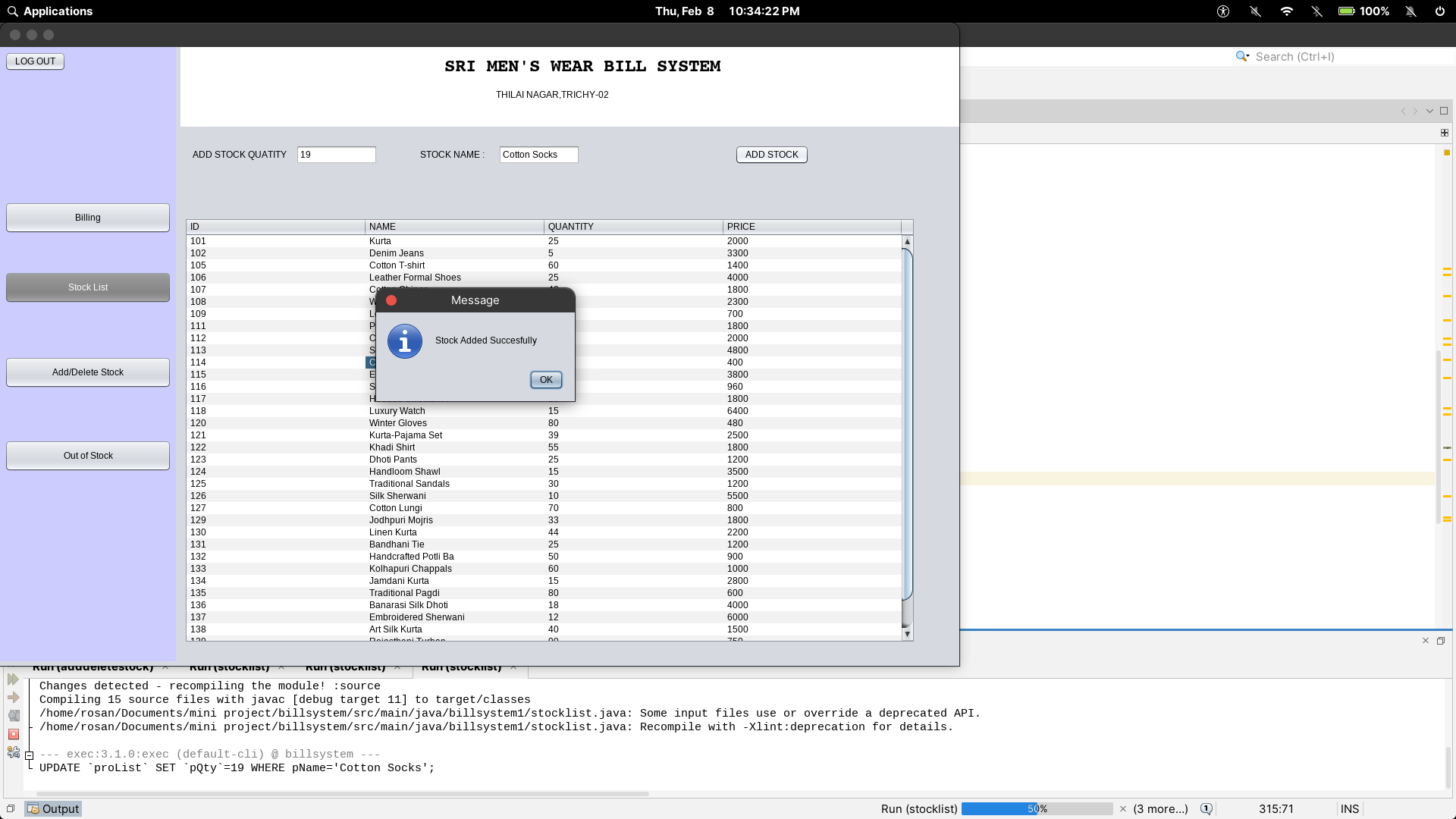


Figure 6.2.2 test-add stock

**Delete Stock**

Figure 6.2.3 test-del stock

Add Quantity

Figure 6.2.3 test-qty

**CHAPTER 7**

**USER MANUAL**

### 7.1 Hardware Requirements

* + - * **Processor :** Intel® Code™ i3-9006U CPU @ 2.00GHz 2.00GHz
      * **Memory : 4** GB
      * **Disk space : 500** GB of free disk space
      * **Screen Resolution : 1024 X 768**

### 7.2 Software Resources

* + - * **Operating System** : Windows 10 / Linux
      * **Software**: :JAVA(jdk 14)
      * **DataBase :** Xampp 8.2.12 version (MySQL)

### 7.3 Installation Procedure

**Step 1:** To setup the package, double click the billsystem.exe file

**Step 2:** Select Setup Language, window will open.

**Step 3:** Select the language during installation and click ok

**Step 4:** Next, the setup license window will open, Click I agree the license agreement and ok.

**Step 5:** Select the location to where the programs and files should be installed, using Browse option. By default C:\ is selected

**Step 6:** After select the location, click ok.

**Step 7:** Next, the installation begins and it take some time to complete the progress.

**Step 8:** After, the installation completed, a window will open with a prompt message, Successfully Installed and click finish.

**7.4 Sample I/O.**

**Login Form:**

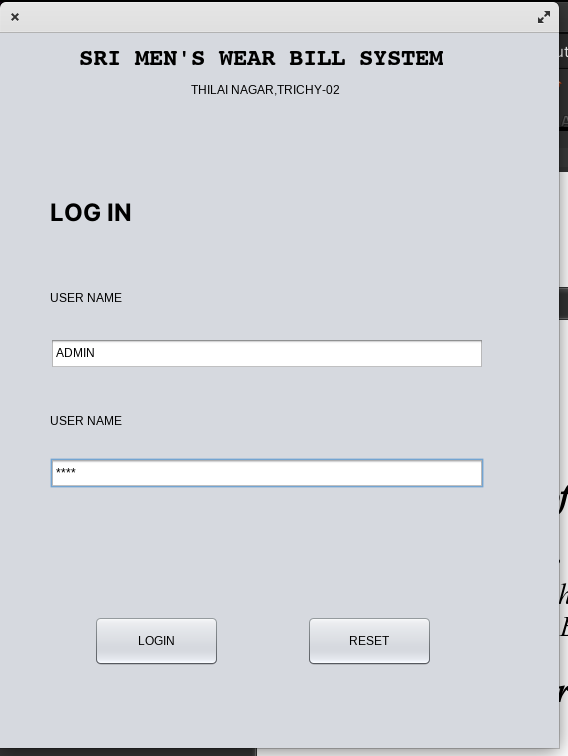
****

Figure 7.4.1 login

### Billing

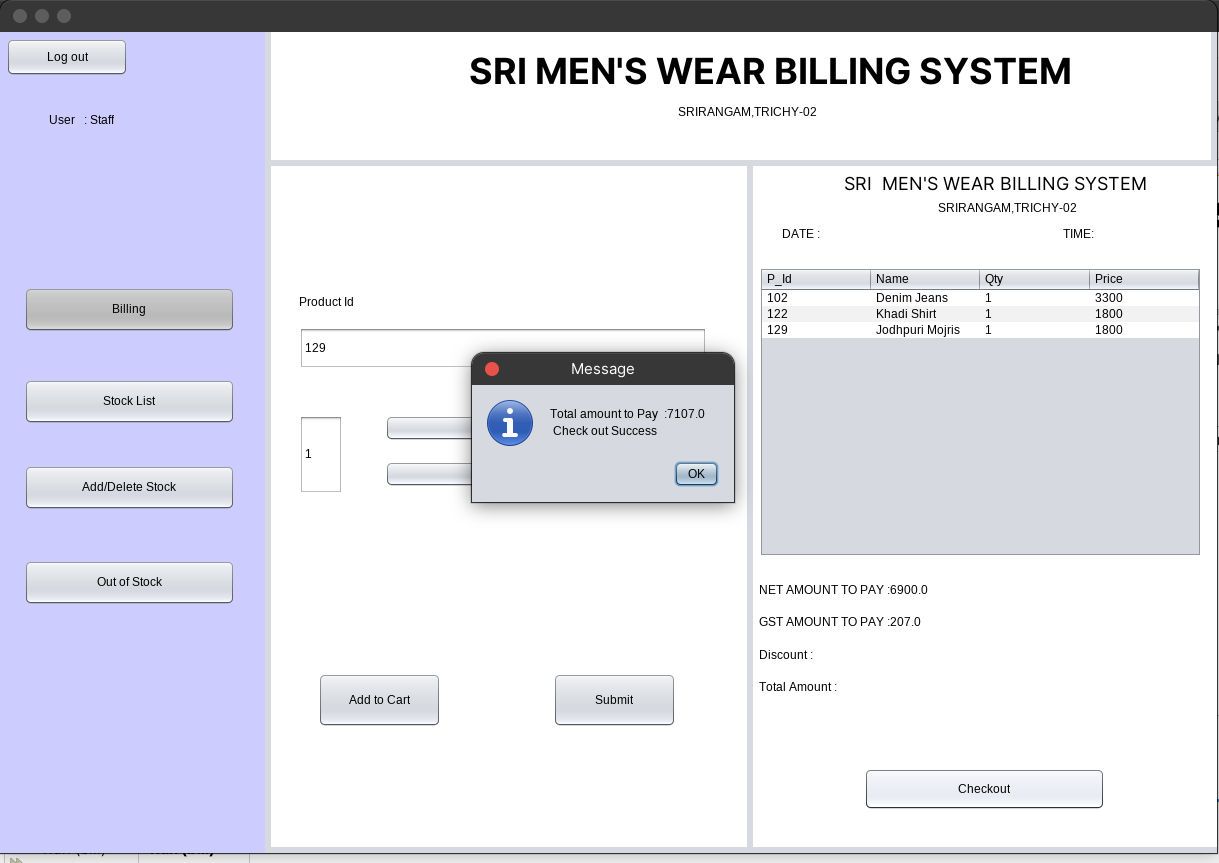
****

Figure 7.4.2 bill

**Add/Edit/Delete Stock Item:**

### 

Figure 7.4.3 add/delete/edit item

**Add Quantity**

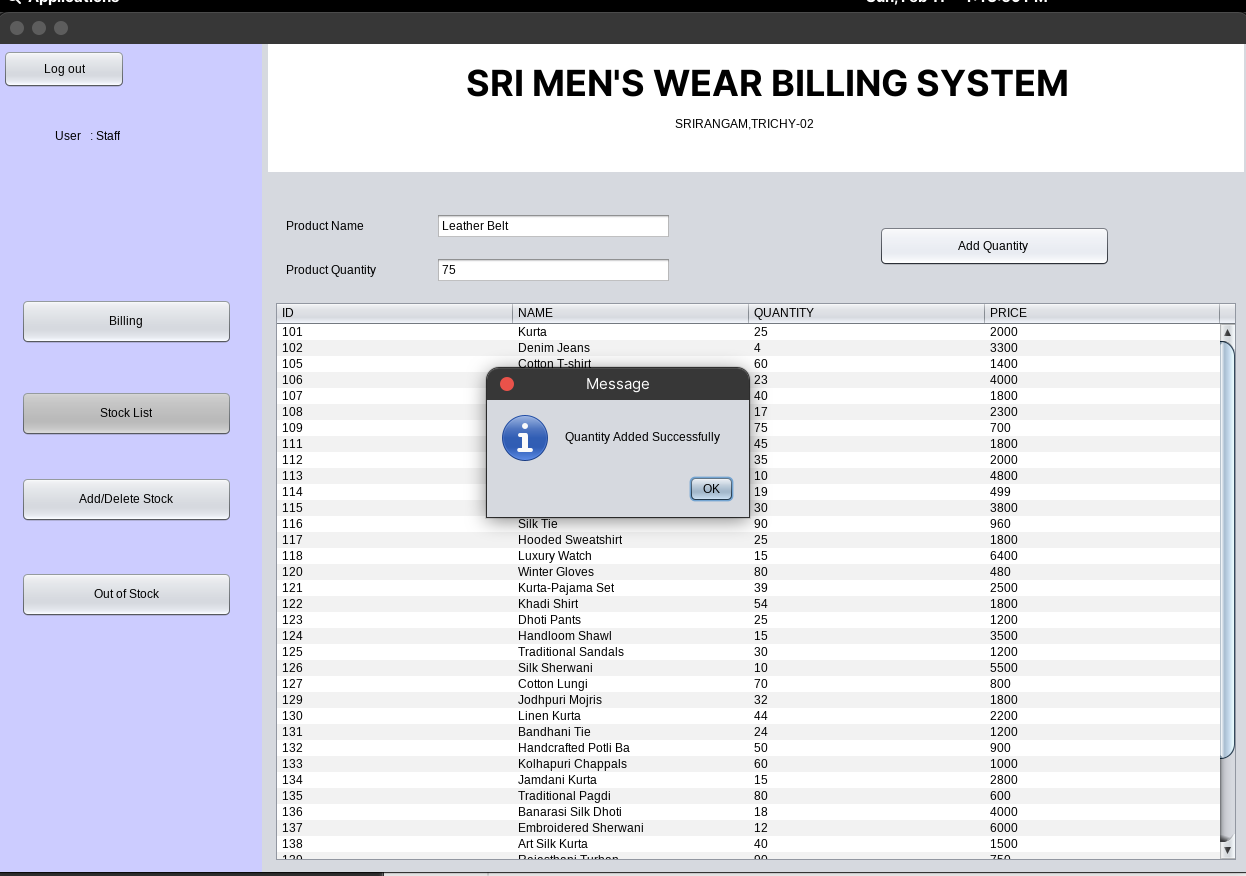
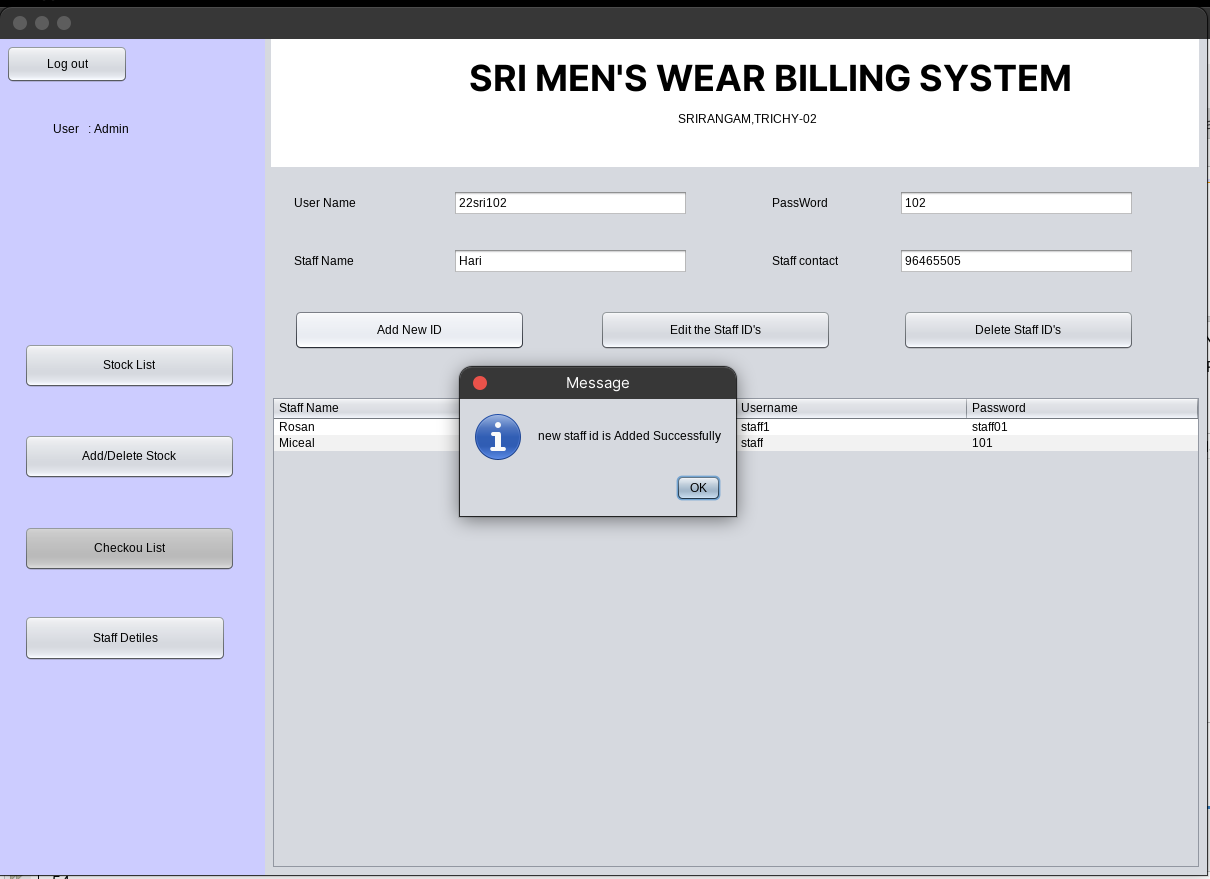


Figure 7.4.4 Add-qty

**Staff ID**

Figure 7.4.5 staff-Details

# CHAPTER 8

# CONCLUSION

### 8.1 Summary of the Project

### The Sri Men's Wear Billing System is a robust solution tailored for the seamless operation of a men's clothing store. It encompasses a user-friendly interface for product addition, quantity adjustments, and pricing details, simplifying the billing process. The system ensures accuracy in transactions, providing customers with precise and transparent billing information.

A key feature includes user authentication, enhancing security measures to safeguard sensitive data. Real-time updates on product availability contribute to effective inventory management, preventing overselling and stockouts. The system's reporting functionalities generate comprehensive insights into sales trends, popular products, and overall business performance.

### 8.2 Future Enhancements

* + - In future, this desktop application will develop as a Web based application where the customer can directly interacts with the system.
    - In future, the online payment facility will be added.
    - The customer can send the order request, quotation through online.
    - In Future, the Authentication will upgrade more effectively
    - The sale history will add in the next upgrade of this Project

**BIBLIOGRAPHY**

**Book References**

* Herbert Scheldt, “The Complete Reference Java 2.0”, Tata McGraw Hill, New Delhi, 2002.
* . Herbert Schildt,” Java: A Beginner‟s Guide McGraw-Hill Education New Delhi,
* Jason Gerner, Elizabeth Naramore , Morgan Owens and Matt Warden, “Professional LAMP - Using Linux , Apache, My SQL and PHP5Web development”, Wiley Publisher, 2006.

### Web References

### https://chat.openai.com/c/aeb4b95c-6efa-466c-996c-0a1e20be4f7

### <https://www.youtube.com/watch?v=EF3yvfmAJIs&list=PLBpH5WxSM4d1hX8CRJw_q838KNCM2DVJZ>

* + - **https://www.visual-paradigm.com/tutorials/data-flow-diagram-dfd.jsp**
    - [**https://app.lucidchart.com/documents#/documents?folder\_id=home**](https://app.lucidchart.com/documents#/documents?folder_id=home)
    - [**https://jaxenter.com/netbeans/the-top-10-netbeans-features-according-to-its-**](https://jaxenter.com/netbeans/the-top-10-netbeans-features-according-to-its-users)[**users**](https://jaxenter.com/netbeans/the-top-10-netbeans-features-according-to-its-users)
    - **https://www.javaguicodexample.com/installusejavanetbeans1.html**

APPENDIX-A

**APPENDIX A: SOURCE CODE**

**LOGIN MODULE**

package billsystem;

import java.sql.\*;

import javax.swing.JOptionPane;

public class login extends javax.swing.JFrame {

public login() {

initComponents();

}

private void initComponents() {

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jLabel3 = new javax.swing.JLabel();

name = new javax.swing.JTextField();

pass = new javax.swing.JPasswordField();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jLabel1.setFont(new java.awt.Font("Inter", 1, 36));

jLabel1.setText("LOG IN");

jLabel2.setText("User name");

jLabel3.setText("User name");

name.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

nameMouseClicked(evt);

} });

pass.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

passMouseClicked(evt);

}

});

jButton1.setText("Reset");

jButton1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton1ActionPerformed(evt);

}

});

jButton2.setText("Log In");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(192, 192, 192)

.addComponent(jLabel1))

.addGroup(layout.createSequentialGroup()

.addGap(55, 55, 55)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(jLabel2)

.addComponent(name)

.addComponent(pass)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 161, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 92, Short.MAX\_VALUE)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 161, javax.swing.GroupLayout.PREFERRED\_SIZE)))))

.addContainerGap(54, Short.MAX\_VALUE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(56, 56, 56)

.addComponent(jLabel3)

.addContainerGap(390, Short.MAX\_VALUE)))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(57, 57, 57)

.addComponent(jLabel1)

.addGap(192, 192, 192)

.addComponent(name, javax.swing.GroupLayout.PREFERRED\_SIZE, 42, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(38, 38, 38)

.addComponent(jLabel2)

.addGap(30, 30, 30)

.addComponent(pass, javax.swing.GroupLayout.PREFERRED\_SIZE, 42, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(123, 123, 123)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 55, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jButton2, javax.swing.GroupLayout.PREFERRED\_SIZE, 55, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(160, Short.MAX\_VALUE))

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(251, 251, 251)

.addComponent(jLabel3)

.addContainerGap(532, Short.MAX\_VALUE)))

);

pack();

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

String uname = name.getText(), upass = pass.getText();

if (uname.equals("admin") && upass.equals("admin")) {

new adstocklist().setVisible(true);

this.dispose();

} else {

String query = "select \* from login where userName ='" + uname + "';";

try {

Connection con;

con = DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem", "root", "");

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(query);

rs.next();

if (rs.getString("passWord").equals(upass)) {

new bill().setVisible(true);

this.dispose();

} else {

System.out.println("lose");

pass.setText("");

JOptionPane.showMessageDialog(this, "Enter Correct password");

}

} catch (SQLException e) {

System.out.println(e);

pass.setText("");

name.setText("");

JOptionPane.showMessageDialog(this, "Enter Correct username");

}

}

}

private void nameMouseClicked(java.awt.event.MouseEvent evt) {

name.setText("");

}

private void passMouseClicked(java.awt.event.MouseEvent evt) {

pass.setText("");

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

name.setText("");

pass.setText("");

}

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new login().setVisible(true);

}

});

}

private javax.swing.JButton jButton1;

private javax.swing.JButton jButton2;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JTextField name;

private javax.swing.JPasswordField pass;

}

**BILLING MODULE**

package billsystem;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.time.LocalDate;

import java.time.LocalTime;

import java.time.format.DateTimeFormatter;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

public class bill extends javax.swing.JFrame {

LocalTime currentTime = LocalTime.now();

LocalDate currentdate = LocalDate.now();

DateTimeFormatter myFormatObj1 = DateTimeFormatter.ofPattern("hh:mm a");

private String time = currentTime.format(myFormatObj1);

private String date = String.valueOf(currentdate);

public bill() {

initComponents();

}

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

jButton10 = new javax.swing.JButton();

jLabel12 = new javax.swing.JLabel();

jPanel2 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jPanel3 = new javax.swing.JPanel();

jLabel11 = new javax.swing.JLabel();

text = new javax.swing.JTextField();

qut = new javax.swing.JTextField();

jButton6 = new javax.swing.JButton();

jButton7 = new javax.swing.JButton();

jButton8 = new javax.swing.JButton();

jButton9 = new javax.swing.JButton();

jPanel4 = new javax.swing.JPanel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

jTable1 = new javax.swing.JTable();

netamount = new javax.swing.JLabel();

gstamount = new javax.swing.JLabel();

jLabel9 = new javax.swing.JLabel();

jLabel10 = new javax.swing.JLabel();

jButton1 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jButton2.setText("Billing");

jButton3.setText("Add/Delete Stock");

jButton4.setText("Stock List");

jButton5.setText("Out of Stock");

jButton10.setText("Log out");

jLabel11.setText("Product Id");

qut.setText("1");

jButton6.setText("-");

jButton7.setText("+");

jButton8.setText("Add to Cart");

jButton9.setText("Submit");

jPanel4.setBackground(new java.awt.Color(255, 255, 255));

jLabel3.setFont(new java.awt.Font("Inter", 0, 18)); // NOI18N

jLabel3.setText("SRI MEN'S WEAR BILLING SYSTEM");

jLabel4.setText("SRIRANGAM,TRICHY-02");

jLabel5.setText("DATE :");

jLabel6.setText("TIME:");

netamount.setText("Net Amount :");

gstamount.setText("GST Amount :");

jLabel9.setText("Total Amount :");

jLabel10.setText("Discount :");

jButton1.setText("Checkout");

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

System.out.println(count+amount+date+time);

JOptionPane.showMessageDialog(this,"Total amount to Pay :"+(amount+gst)+"\n Check out Success");

String formattedDate = LocalDate.parse(date, DateTimeFormatter.ofPattern("yyyy-MM-dd")).toString();

DateTimeFormatter myFormatObj1 = DateTimeFormatter.ofPattern("hh:mm:ss a");

String timenow = currentTime.format(myFormatObj1);

String query = "INSERT INTO checkout (n\_pro, amount, pay\_type, c\_date, c\_time) VALUES (?, ?, 'Cash', ?, ?)";

try (Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/billsystem", "root", "");

PreparedStatement ps = con.prepareStatement(query)) {

ps.setInt(1, count);

ps.setDouble(2, amount);

ps.setString(3, formattedDate); // Use the formatted date

ps.setString(4, timenow);

// Execute the query

ps.executeUpdate();

System.out.println("Data inserted successfully.");

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

i=0;

DefaultTableModel tb = (DefaultTableModel)jTable1.getModel();

tb.setNumRows(i);

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

new bill().setVisible(true);

this.dispose();

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

new stocklist().setVisible(true);

this.dispose();

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

new outofstock().setVisible(true);

this.dispose();

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

new adddeletestock().setVisible(true);

this.dispose();

}

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {

String id = String.valueOf(Integer.valueOf(text.getText()));

String query = "select \* from proList where pId = '"+id+"'";

Connection con;

try { con=DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem","root","");

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(query);

rs.next();

int pid = rs.getInt(1);

String name = rs.getString(2);

int qty =Integer.valueOf(qut.getText()) ;

int ruppe= qty \* Integer.valueOf(rs.getInt(4));

amount+=ruppe;

int q = rs.getInt(3)-qty;

System.out.println(q);

String query1 = "UPDATE `proList` SET `pQty`="+q+" WHERE pId="+pid+";";

PreparedStatement ps = con.prepareStatement(query1);

ps.execute();

DefaultTableModel tb = (DefaultTableModel)jTable1.getModel();

int j=0;

tb.setNumRows(i);

String tbData[]={String.valueOf(pid),name, String.valueOf(qty), String.valueOf(ruppe)};

tb.addRow(tbData);

j++;

i++;

count +=1;

}

catch (SQLException e) {

JOptionPane.showMessageDialog(this,"THIS ID DOES NOT EXIST");

System.out.println(e);

}

}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {

System.out.printf("in submit");

try {

gst = amount\*3/100;

netamount.setText("NET AMOUNT TO PAY :" + amount);

gstamount.setText("GST AMOUNT TO PAY :" + gst);

JOptionPane.showMessageDialog(this,"Submited to Check out");

}catch (Exception e){

System.out.println(e);

} }

private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {

new login().setVisible(true);

this.dispose();

}

public static void main(String args[]) {

new bill().setVisible(true);

}

int count ;float amount = 0;int i=0;

float gst = 0;

}

**STOCK LIST**

package billsystem;

import java.sql.\*;

import javax.swing.\*;

public class stocklist extends javax.swing.JFrame {

public stocklist() {

initComponents();

setintitable();

}

id = text.getText();

n=name.getText();

q=qty.getText();

p=price.getText();

String Query ="UPDATE `proList` SET `pId`='["+id+"]',`pName`='["+n+"]',`pQty`='["+q+"]',`pPrice`='["+p+"]' WHERE pId = "+id+";";

try {

Connection con;

con = DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem","root","");

PreparedStatement ps = con.prepareStatement(Query);

ps.execute();

JOptionPane.showMessageDialog(this,"Product Added to stock");

}

catch (SQLException e) {

System.out.println(e);

JOptionPane.showMessageDialog(this,"Fill the required INFO");

}

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

jButton10 = new javax.swing.JButton();

jLabel12 = new javax.swing.JLabel();

jPanel2 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jPanel3 = new javax.swing.JPanel();

jLabel3 = new javax.swing.JLabel();

qty = new javax.swing.JTextField();

name = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

jButton6 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

table = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(204, 204, 255));

jButton2.setText("Billing");

jButton2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton2ActionPerformed(evt);

}

});

jButton3.setText("Add/Delete Stock");

jButton3.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton3ActionPerformed(evt);

}

});

jButton4.setBackground(new java.awt.Color(153, 153, 153));

jButton4.setText("Stock List");

jButton4.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton4ActionPerformed(evt);

}

});

jButton5.setText("Out of Stock");

jButton5.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton5ActionPerformed(evt);

}

});

jButton10.setText("Log out");

jButton10.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

jButton10ActionPerformed(evt);

}

});

jLabel4.setText("Product Quantity");

jButton6.setText("Add Quantity ");

},

new String [] {

"ID", "NAME", "QUANTITY", "PRICE"

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

new bill().setVisible(true);

this.dispose();

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

new stocklist().setVisible(true);

this.dispose();

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

new outofstock().setVisible(true);

this.dispose();

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

new adddeletestock().setVisible(true);

this.dispose();

}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

int q = Integer.valueOf(qty.getText());

String n = name.getText();

try{

String query="UPDATE `proList` SET `pQty`="+q+" WHERE pName='"+n+"';";

System.out.println(query);

Connection con;

con = DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem","root","");

PreparedStatement ps = con.prepareStatement(query);

ps.execute();

JOptionPane.showMessageDialog(this,"Quantity Added Successfully");

setintitable();

}

catch (Exception e){

System.out.println(e);

}

}

private void tableMouseClicked(java.awt.event.MouseEvent evt) {

DefaultTableModel model = (DefaultTableModel)table.getModel();

int myindex = table.getSelectedRow();

qty.setText(model.getValueAt(myindex, 2).toString());

name.setText(""+model.getValueAt(myindex, 1).toString());

}

private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {

new login().setVisible(true);

this.dispose();

}

public static void main(String args[]) {

new stocklist().setVisible(true);

}

private void setintitable() {

Connection con;

String query = "select \* from proList where pQty >0";

try {

con = DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem","root","");

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(query);

// table.setModel(Db.Utilils.resultSetToTableModel(rs));

while(rs.next()){

String id = String.valueOf(rs.getInt(1));

String pqty = String.valueOf(rs.getInt(3));

String pname = String.valueOf(rs.getString(2));

String price = String.valueOf(rs.getInt(4));

String tbData[]={id,pname,pqty,price};

DefaultTableModel tb = (DefaultTableModel)table.getModel();

tb.addRow(tbData);

}

catch (SQLException e) {

System.out.println(e);

}

}

**Add/Delete Stock MODULE**

package billsystem;

import java.sql.\*;

import java.sql.Statement;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

public class adddeletestock extends javax.swing.JFrame {

public adddeletestock() {

initComponents();

setintitable();

}

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

jButton9 = new javax.swing.JButton();

jLabel7 = new javax.swing.JLabel();

jPanel2 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jPanel3 = new javax.swing.JPanel();

jLabel3 = new javax.swing.JLabel();

qty = new javax.swing.JTextField();

name = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

text = new javax.swing.JTextField();

price = new javax.swing.JTextField();

jButton6 = new javax.swing.JButton();

jButton7 = new javax.swing.JButton();

jButton8 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

table = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jButton2.setText("Billing");

jButton3.setText("Add/Delete Stock");

jButton4.setText("Stock List");

jButton5.setText("Out of Stock");

jButton9.setText("Log out");

jLabel7.setText("User : Staff");

jLabel1.setText("SRI MEN'S WEAR BILLING SYSTEM"); jLabel2.setText("SRIRANGAM,TRICHY-02");

jLabel3.setText("Product Name");

jLabel4.setText("Product Quantity");

jLabel5.setText("Product ID");

jLabel6.setText("Product Price");

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

new bill().setVisible(true);

this.dispose();

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

new stocklist().setVisible(true);

this.dispose();

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

new outofstock().setVisible(true);

this.dispose();

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

new adddeletestock().setVisible(true);

this.dispose();

}

private void tableMouseClicked(java.awt.event.MouseEvent evt) {

DefaultTableModel model = (DefaultTableModel)table.getModel();

int myindex = table.getSelectedRow();

qty.setText(model.getValueAt(myindex, 2).toString());

name.setText(model.getValueAt(myindex, 1).toString());

text.setText(model.getValueAt(myindex, 0).toString());

price.setText(model.getValueAt(myindex, 3).toString());

}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

Connection con;

String n,q,p;

n=name.getText();

q=qty.getText();

p=price.getText();

String query = "insert into proList(pName,pQty,pPrice) values('"+n+"',"+q+","+p+")";

try {

con = DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem","root","");

PreparedStatement ps = con.prepareStatement(query);

ps.execute();

JOptionPane.showMessageDialog(this,"Product Added to stock");

}

catch (SQLException e) {

System.out.println(e);

JOptionPane.showMessageDialog(this,"Fill the required INFO");

} }

private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {

int id = Integer.valueOf(text.getText());

String query = "delete from prolist where pId="+id+";";

try {

Connection con; con=DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem","root","");

PreparedStatement ps = con.prepareStatement(query);

ps.execute();

JOptionPane.showMessageDialog(this,"Product DELETED from the stock");

}

catch (SQLException e) {

System.out.println(e);

JOptionPane.showMessageDialog(this,"Fill the required INFO");

} }

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {

new login().setVisible(true);

this.dispose();

}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

String n,q,p,id;

id = text.getText();

n=name.getText();

q=qty.getText();

p=price.getText();

String Query = "UPDATE `proList` SET `pId`=" + Integer.valueOf(id) + ", `pName`='" + n + "', `pQty`=" + Integer.valueOf(q) + ", `pPrice`=" + Integer.valueOf(p) + " WHERE `pId`=" + Integer.valueOf(id) + ";";

System.out.println(Query);

try {

Connection con; Con=DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem","root","");

PreparedStatement ps = con.prepareStatement(Query);

ps.execute();

JOptionPane.showMessageDialog(this,"The Detiles Updated Successfully");

}

catch (SQLException e) {

System.out.println(e);

JOptionPane.showMessageDialog(this,"Fill the required INFO");

}

}

public static void main(String args[]) {

new adddeletestock().setVisible(true);

}

private void setintitable() {

Connection con;

String query = "select \* from proList where pQty >0";

try {

con = DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem","root","");

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(query);

while(rs.next()){

String id = String.valueOf(rs.getInt(1));

String pqty = String.valueOf(rs.getInt(3));

String pname = String.valueOf(rs.getString(2));

String price = String.valueOf(rs.getInt(4));

String tbData[]={id,pname,pqty,price};

DefaultTableModel tb = (DefaultTableModel)table.getModel();

tb.addRow(tbData);

} catch (SQLException e) {

System.out.println(e);

}}}

**CHECKOUT LIST**

package billsystem;

import java.sql.\*;

import javax.swing.table.DefaultTableModel;

public class adcheckoutlist extends javax.swing.JFrame {

public adcheckoutlist() {

initComponents();

insertdata();

}

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jButton3 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

jButton1 = new javax.swing.JButton();

jButton2 = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

jPanel3 = new javax.swing.JPanel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

table = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jButton3.setText("Add/Delete Stock");

jButton4.setText("Stock List");

jButton5.setText("Checkou List");

jButton1.setText("Staff Detiles");

jButton2.setText("Log out");

jLabel3.setText("SRI MEN'S WEAR BILLING SYSTEM");

jLabel4.setText("SRIRANGAM,TRICHY-02");

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

new ad\_add\_delete().setVisible(true);

this.dispose();

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

new adstocklist().setVisible(true);

this.dispose();

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

new adcheckoutlist().setVisible(true);

this.dispose();

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

new staff().setVisible(true);

this.dispose();

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

new login().setVisible(true);

this.dispose();

}

public static void main(String args[]) {

new adcheckoutlist().setVisible(true);

}

private void insertdata(){

Connection con;

String query = "select \* from checkout";

try { con=DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem","root","");

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(query);

// table.setModel(Db.Utilils.resultSetToTableModel(rs));

while(rs.next()){

String id = String.valueOf(rs.getInt(1));

String date= rs.getString(2);

String time = rs.getString(3);

String npro = String.valueOf(rs.getInt(4));

String amount = String.valueOf(rs.getInt(5));

String type = rs.getString(6);

String tbData[]={id,date,time,npro,amount,type};

DefaultTableModel tb = (DefaultTableModel)table.getModel();

tb.addRow(tbData);

}

catch (SQLException e) {

System.out.println(e);

} }}

**STAFF DETAILS MODULE**

package billsystem;

import java.sql.\*;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

public class staff extends javax.swing.JFrame {

public staff() {

initComponents();

setintitable();

}

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jButton3 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

jButton1 = new javax.swing.JButton();

jLabel9 = new javax.swing.JLabel();

jButton9 = new javax.swing.JButton();

jPanel3 = new javax.swing.JPanel();

jLabel3 = new javax.swing.JLabel();

jLabel4 = new javax.swing.JLabel();

jPanel4 = new javax.swing.JPanel();

jLabel5 = new javax.swing.JLabel();

jLabel6 = new javax.swing.JLabel();

jLabel7 = new javax.swing.JLabel();

jLabel8 = new javax.swing.JLabel();

jButton6 = new javax.swing.JButton();

name = new javax.swing.JTextField();

sname = new javax.swing.JTextField();

pass = new javax.swing.JTextField();

num = new javax.swing.JTextField();

jButton7 = new javax.swing.JButton();

jButton8 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

table = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

button3.setText("Add/Delete Stock");

jButton4.setText("Stock List");

jButton5.setText("Checkou List");

jLabel9.setText("User : Admin");

jButton9.setText("Log out");

jLabel5.setText("User Name");

jLabel6.setText("PassWord");

jLabel7.setText("Staff Name");

jLabel8.setText("Staff contact");

jButton6.setText("Delete Staff ID's");

jButton7.setText("Add New ID");

jButton8.setText("Edit the Staff ID's");

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

new ad\_add\_delete().setVisible(true);

this.dispose();

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

new adstocklist().setVisible(true);

this.dispose();

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

new adcheckoutlist().setVisible(true);

this.dispose();

}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

String id= name.getText();

String p = pass.getText();

String query = "delete from login where userName='"+id+"' and passWord = '"+p+"';";

System.out.println(query);

try {

Connection con; con=DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem","root","");

PreparedStatement ps = con.prepareStatement(query);

ps.execute();

JOptionPane.showMessageDialog(this,"The Staff's ID is Deleted Successfully");

setintitable() ;

}

catch (SQLException e) {

System.out.println(e);

JOptionPane.showMessageDialog(this,"Fill the required INFO");

}

}

private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

Connection con;

String u,p,n,c;

u =name.getText();

p=pass.getText();

n=sname.getText();

c=num.getText();

String query = "insert into login(userName,passWord,staff\_name,contact) values('"+u+"','"+p+"','"+n+"','"+c+"')";

System.out.println("insert into login(userName,passWord,staff\_name,contact) values('"+u+"',"+p+","+n+""+c+")");

try { con=DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem","root","");

PreparedStatement ps = con.prepareStatement(query);

ps.execute();

JOptionPane.showMessageDialog(this,"new staff id is Added Successfully");

}

catch (SQLException e) {

System.out.println(e);

JOptionPane.showMessageDialog(this,"Fill the required INFO");

}

setintitable() ;

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

new staff().setVisible(true);

this.dispose();

}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {

new login().setVisible(true);

this.dispose();

}

private void tableMouseClicked(java.awt.event.MouseEvent evt) {

DefaultTableModel model = (DefaultTableModel)table.getModel();

int myindex = table.getSelectedRow();

name.setText(model.getValueAt(myindex, 2).toString());

pass.setText(model.getValueAt(myindex, 3).toString());

sname.setText(model.getValueAt(myindex, 0).toString());

num.setText(model.getValueAt(myindex, 1).toString());

}

public static void main(String args[]) {

new staff().setVisible(true);

}

private void setintitable() {

Connection con;

String query = "select \* from login";

try {

con=DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem","root","");

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(query);

// table.setModel(Db.Utilils.resultSetToTableModel(rs));

while(rs.next()){

String sname = rs.getString(4);

String num = rs.getString(5);

String name = rs.getString(2);

String pass = rs.getString(3);

String tbData[]={sname,num,name,pass};

DefaultTableModel tb = (DefaultTableModel)table.getModel();

tb.addRow(tbData);

} }

catch (SQLException e) {

System.out.println(e);

}}}

**OUT OF STOCK**

package billsystem;

import java.sql.\*;

import javax.swing.JOptionPane;

import javax.swing.table.DefaultTableModel;

public class outofstock extends javax.swing.JFrame {

public outofstock() {

initComponents();

setintitable();

}

private void initComponents() {

jPanel1 = new javax.swing.JPanel();

jButton2 = new javax.swing.JButton();

jButton3 = new javax.swing.JButton();

jButton4 = new javax.swing.JButton();

jButton5 = new javax.swing.JButton();

jButton10 = new javax.swing.JButton();

jLabel9 = new javax.swing.JLabel();

jPanel2 = new javax.swing.JPanel();

jLabel1 = new javax.swing.JLabel();

jLabel2 = new javax.swing.JLabel();

jPanel3 = new javax.swing.JPanel();

jLabel5 = new javax.swing.JLabel();

name = new javax.swing.JTextField();

jLabel4 = new javax.swing.JLabel();

qty = new javax.swing.JTextField();

jButton1 = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

table = new javax.swing.JTable();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

jPanel1.setBackground(new java.awt.Color(204, 204, 255));

jButton2.setText("Billing");

jButton3.setText("Add/Delete Stock");

jButton4.setText("Stock List");

jButton5.setBackground(new java.awt.Color(153, 153, 153));

jButton5.setText("Out of Stock");

jButton10.setText("Log out");

jLabel9.setBackground(new java.awt.Color(255, 255, 255));

jLabel9.setText("User : Staff");

jLabel5.setText("Product ID");

jLabel4.setText("Product Quantity");

jButton1.setText("Add Stock");

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

new bill().setVisible(true);

this.dispose();

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

new stocklist().setVisible(true);

this.dispose();

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

new outofstock().setVisible(true);

this.dispose();

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

new adddeletestock().setVisible(true);

this.dispose();

}

private void tableMouseClicked(java.awt.event.MouseEvent evt) {

DefaultTableModel model = (DefaultTableModel)table.getModel();

int myindex = table.getSelectedRow();

qty.setText(model.getValueAt(myindex, 2).toString());

name.setText(model.getValueAt(myindex, 0).toString());

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

int q = Integer.valueOf(qty.getText());

String n = name.getText();

try{

String query="UPDATE `proList` SET `pQty`="+q+" WHERE pName='"+n+"';";

System.out.println(query);

Connection con;

con = DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem","root","");

PreparedStatement ps = con.prepareStatement(query);

ps.execute();

JOptionPane.showMessageDialog(this,"The quantity is added");

setintitable();

}

catch (Exception e){

System.out.println(e);

} }

private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {

new login().setVisible(true);

this.dispose();

}

public static void main(String args[]) {

new outofstock().setVisible(true);

}

private void setintitable() {

Connection con;

String query = "select \* from proList where pQty=0";

try { con=DriverManager.getConnection("JDBC:MYSQL://localhost:3306/billsystem","root","");

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(query);

// table.setModel(Db.Utilils.resultSetToTableModel(rs));

DefaultTableModel tb = (DefaultTableModel)table.getModel();

tb.setNumRows(0);

while(rs.next()){

String id = String.valueOf(rs.getInt(1));

String pqty = String.valueOf(rs.getInt(3));

String pname = String.valueOf(rs.getString(2));

String price = String.valueOf(rs.getInt(4));

String tbData[]={id,pname,pqty,price};

tb.addRow(tbData);

}}

catch (SQLException e) {

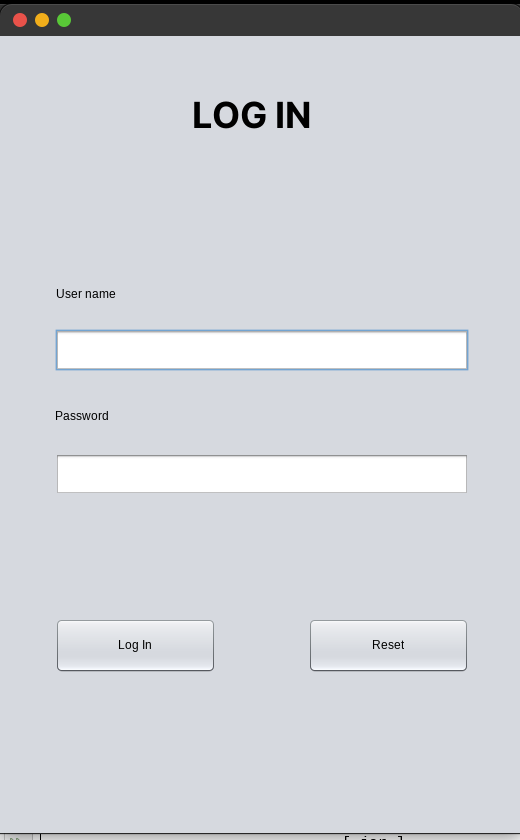
System.out.println(e);

}}

APPENDIX-B

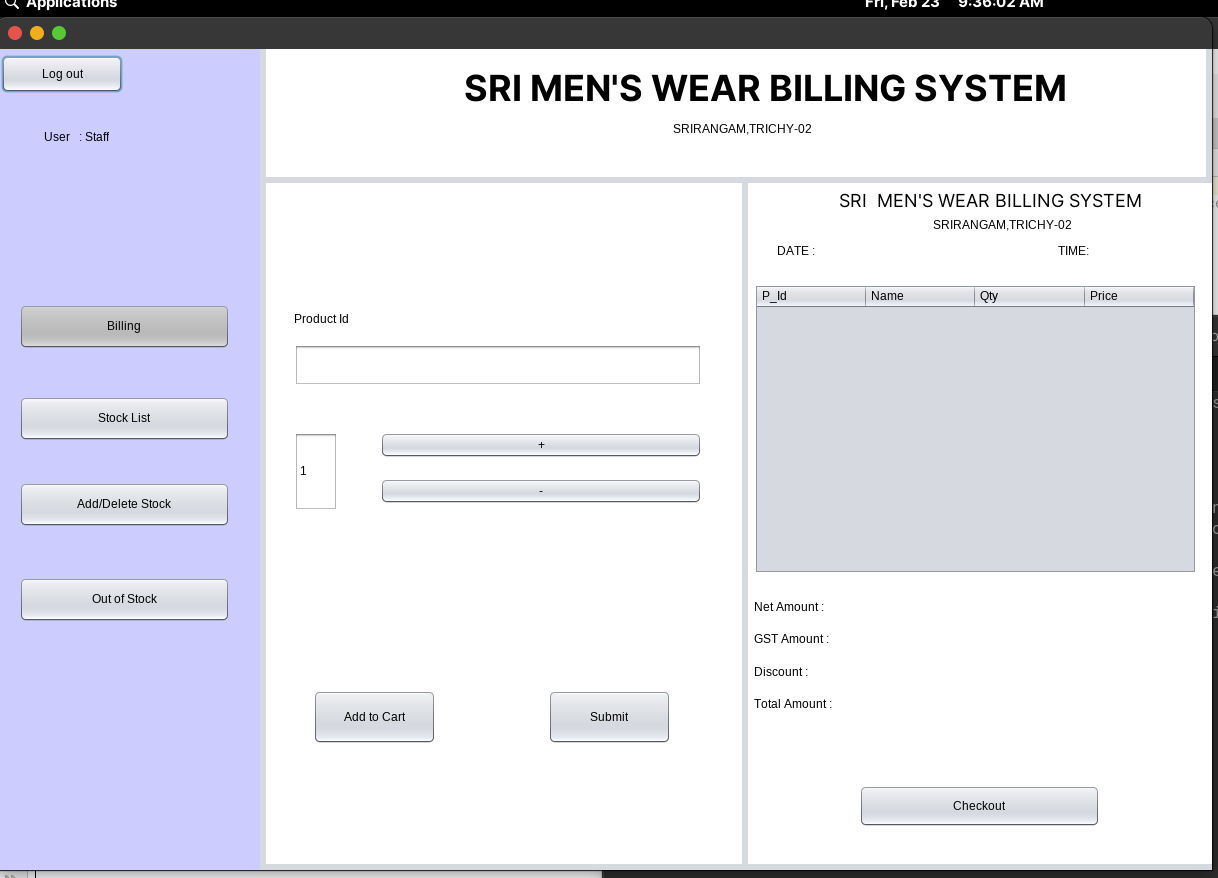
**APPENDIX B: SCREEN SHOTS**

**LOGIN**

****

*Figure B.1.1 Login*

***BILLING***

******

*Figure B.1.2 bill*

***STOCK LIST***

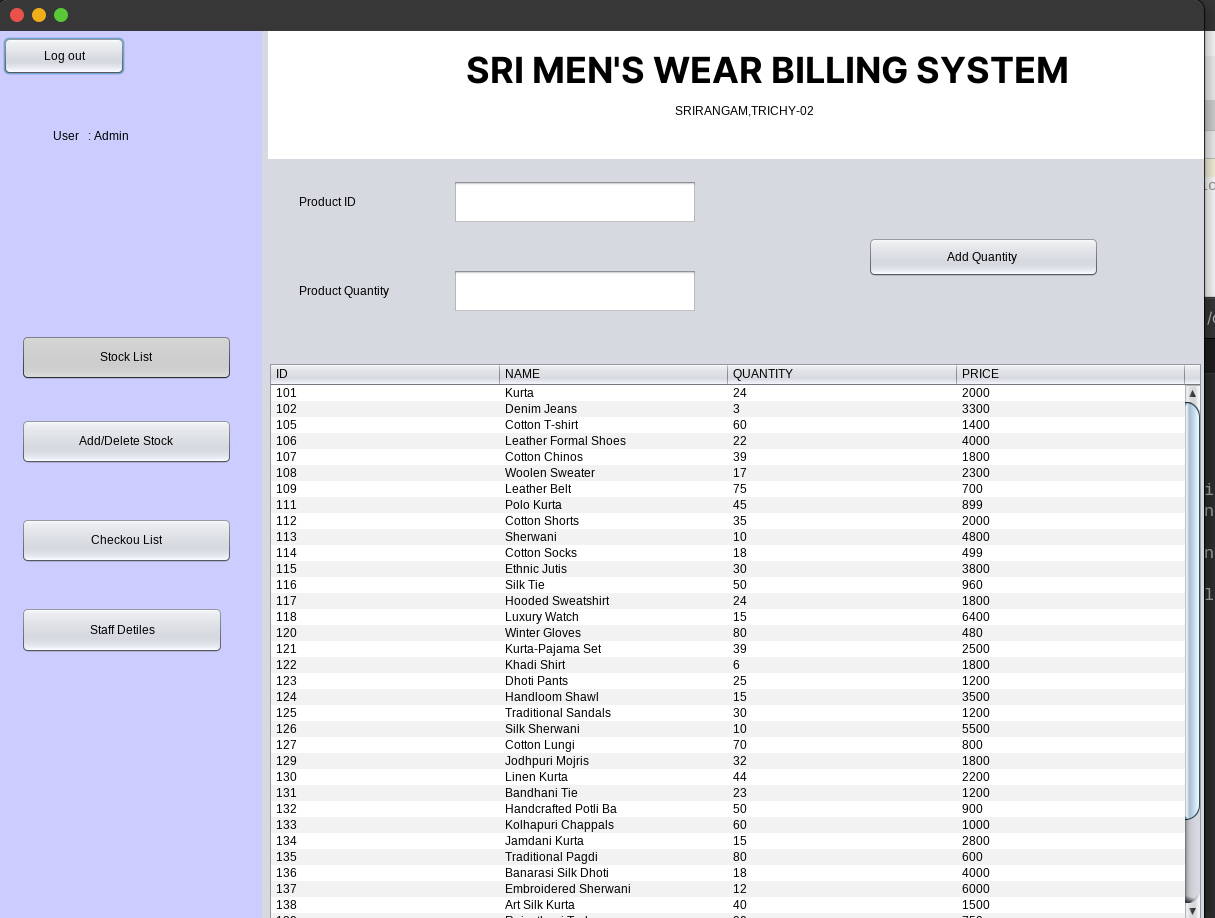
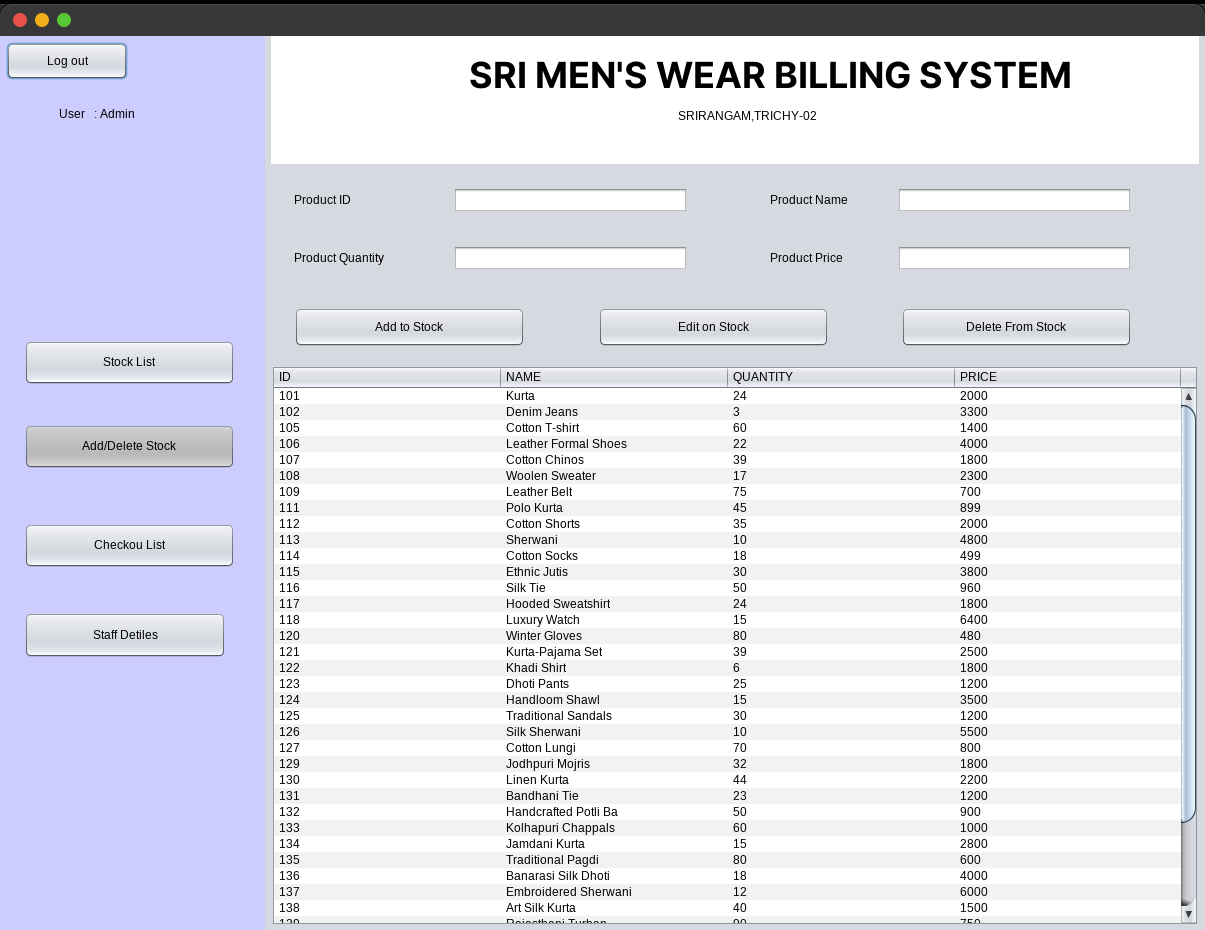
******

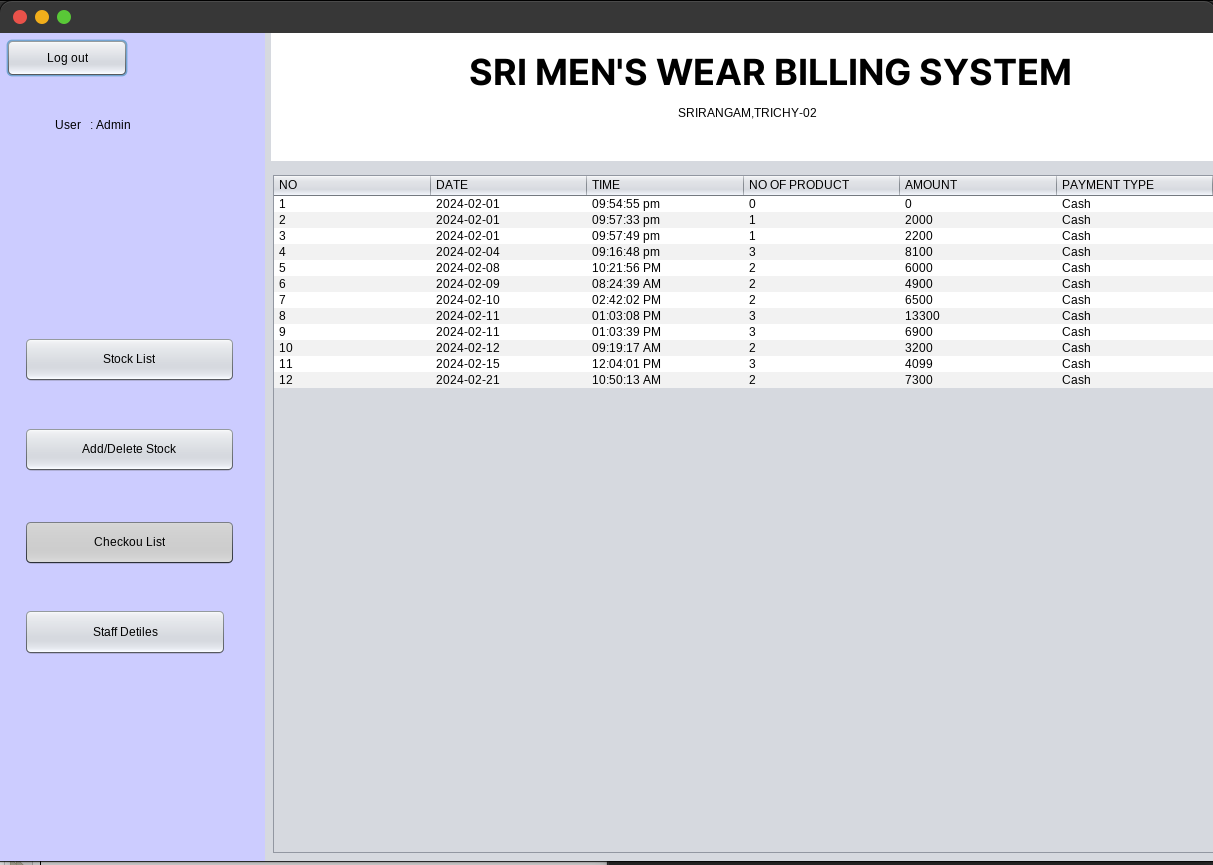
Figure B.1.3 Stock-List

**ADD/DELETE STOCK**

****

*Figure B.1.4 Add/Delete Stock*

***CHECKOUT LIST***

******

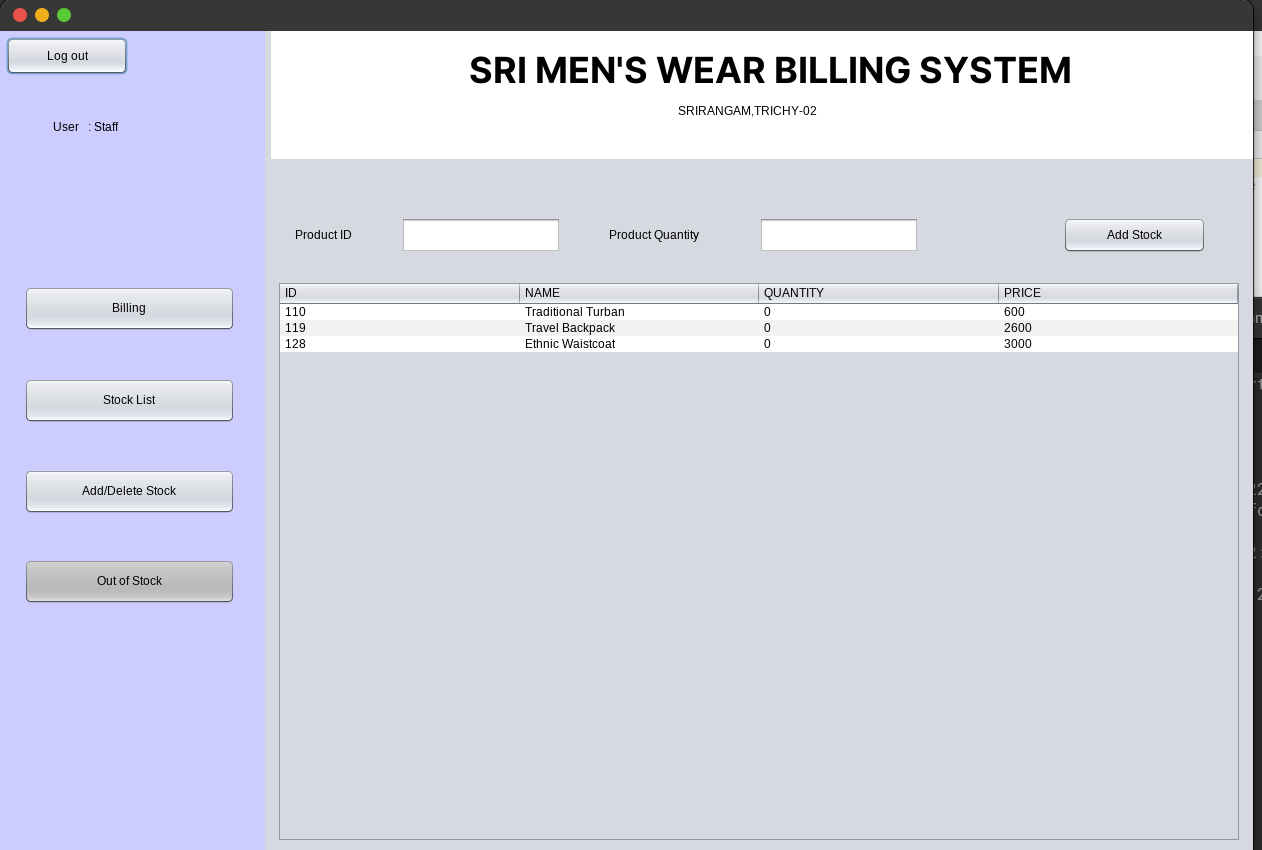
*Figure B.1.5 Checkout-List*

***STAFF DETAILS***

******

*Figure B.1.6 Staff-Details*

***OUT OF STOCK***



*Figure B.1.7 Out Of Stock*