ATM SIMULATION SYSTEM

Project Report

Submitted by

P.VINEETHA AP22110010580

K.INDU MEGHANA AP22110010540

A.MAHITHA AP22110010576

Under the Supervision of

Mr. LAKSHMI NARAYANA KOLLA
Lecturer
Department of Computer Science and Engineering
SRM University-AP

In partial fulfilment for the requirements of the project

BACHELOR OF TECHNOLOGY IN

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRM UNIVERSITY-AP

NEERUKONDA

MANAGALAGIRI - 522503

ANDHRA PRADESH, INDIA

APRIL-2025

TABLE OF CONTENTS

ABSTRACT

The ATM Simulation System is a desktop application in Java that is used to mimic the basic functions of an Automated Teller Machine (ATM). It has been created with Java and AWT for its graphical user interface, and coupled with MySQL via JDBC for its backend data storage. The purpose of this project is to enable users to experience an interactive and real-world-like banking environment. The system supports basic banking activities like account login, balance check, cash withdrawal, and cash deposit. All transactions are processed securely and updated in the MySQL database in real time, maintaining data consistency and integrity.

IntelliJ IDEA was utilized as the development platform, improving productivity with intelligent coding suggestions and debugging capabilities. The application shows good knowledge of object-oriented programming, event-driven GUI programming, and relational database integration. The project is a realistic implementation of basic Java concepts, JDBC-based database connectivity, and AWT-based GUI programming and hence a realistic simulation of ATM operations for training and demonstration purposes.

INTRODUCTION

In today's digital age, banking operations have been more automated and customers get instant, safe, and convenient access to their banking accounts via means like ATMs. The current project, named "ATM Simulation System", is a desktop application that is designed using Java, JDBC, MySQL, and AWT, and it intends to mimic the main operations of an actual ATM in a testable and friendly environment.

Java is a platform-independent and object-oriented language was used for the backend functionality and logic. AWT (Abstract Window Toolkit) was employed to develop the graphical user interface (GUI), offering a responsive and interactive user interface. The application uses JDBC (Java Database Connectivity) to connect with a MySQL database, allowing for real-time data storage and retrieval of user account details and transaction history.

This project not only enhances Java programming and GUI knowledge but also gives practical exposure to database connection and system integration. It acts as a basis for understanding the nature of finance software systems and sets the platform for creating sophisticated banking applications.

OBJECTIVES

The main objectives of the ATM Simulation System project are:

1. **To simulate real-world ATM operations** such as balance inquiry, cash withdrawal, cash deposit, and PIN change.

2. **To develop a user-friendly graphical interface** using Java AWT that allows smooth interaction for users.

3. **To implement secure user authentication** by validating account numbers and PINs before allowing access to banking functions.

4. **To integrate a backend MySQL database** using JDBC to store and manage user account details and transaction history.

5. **To apply core Java concepts** including object-oriented programming, exception handling, and event-driven programming.

6. **To demonstrate database connectivity and manipulation** through SQL queries executed from within a Java application.

7. **To create a reliable and efficient system** capable of handling multiple transactions with accurate data processing and validation.

SYSTEM DESIGN

The system design of the ATM Simulation System is structured into three main components: User Interface (UI), Business Logic, and Database Layer.

1. User Interface (Frontend) – Java AWT

The GUI is developed using AWT components like Frame, Label, Button, TextField, and PasswordField.
The main screens include:

Login screen for entering account number and PIN.
Dashboard with options: Balance Inquiry, Deposit, Withdraw.

2. Business Logic (Backend) – Core Java

The backend handles all transaction logic and user input processing. Functions include:

- Verifying user credentials.
- Performing balance checks and updates.
- Logging transactions.
- Validating input to prevent invalid or malicious operations.

Object-oriented principles are used to create modular, reusable code (e.g., separate classes for `Account`, `TransactionHandler`, etc.).

3. Database Layer – MySQL with JDBC

MySQL database is used to store user account data.
Tables include:

  - accounts – storing account number, user name, PIN, and balance.

  - transactions – recording transaction type, amount, date, and account number.

JDBC is used for database connectivity, enabling real-time updates and queries from the Java application.

4. Tools and Technologies

- Language: Java
- Database: MySQL
- IDE: IntelliJ IDEA
- GUI: AWT (Abstract Window Toolkit)
- Connector: JDBC (Java Database Connectivity)

IMPLEMENTATION

The implementation of the ATM Simulation System was carried out using Java for the core logic and AWT for the graphical user interface. IntelliJ IDEA served as the development environment due to its support for Java and ease of integration with external libraries. The backend database was created using MySQL, and connectivity between the application and the

CODE:

Mysql Code

```
USE ATM;

CREATE TABLE users (
   id INT AUTO_INCREMENT PRIMARY KEY,
   username VARCHAR(50) NOT NULL,
   pin VARCHAR(4) NOT NULL,
   balance DECIMAL(10, 2) NOT NULL
)

INSERT INTO users (username, pin, balance) VALUES ('vineetha', '1234',
1000.00);
INSERT INTO users (username, pin, balance) VALUES ('indu', '4321',
2000.00);
INSERT INTO users (username, pin, balance) VALUES ('mahitha', '5678',
2000.00);
```

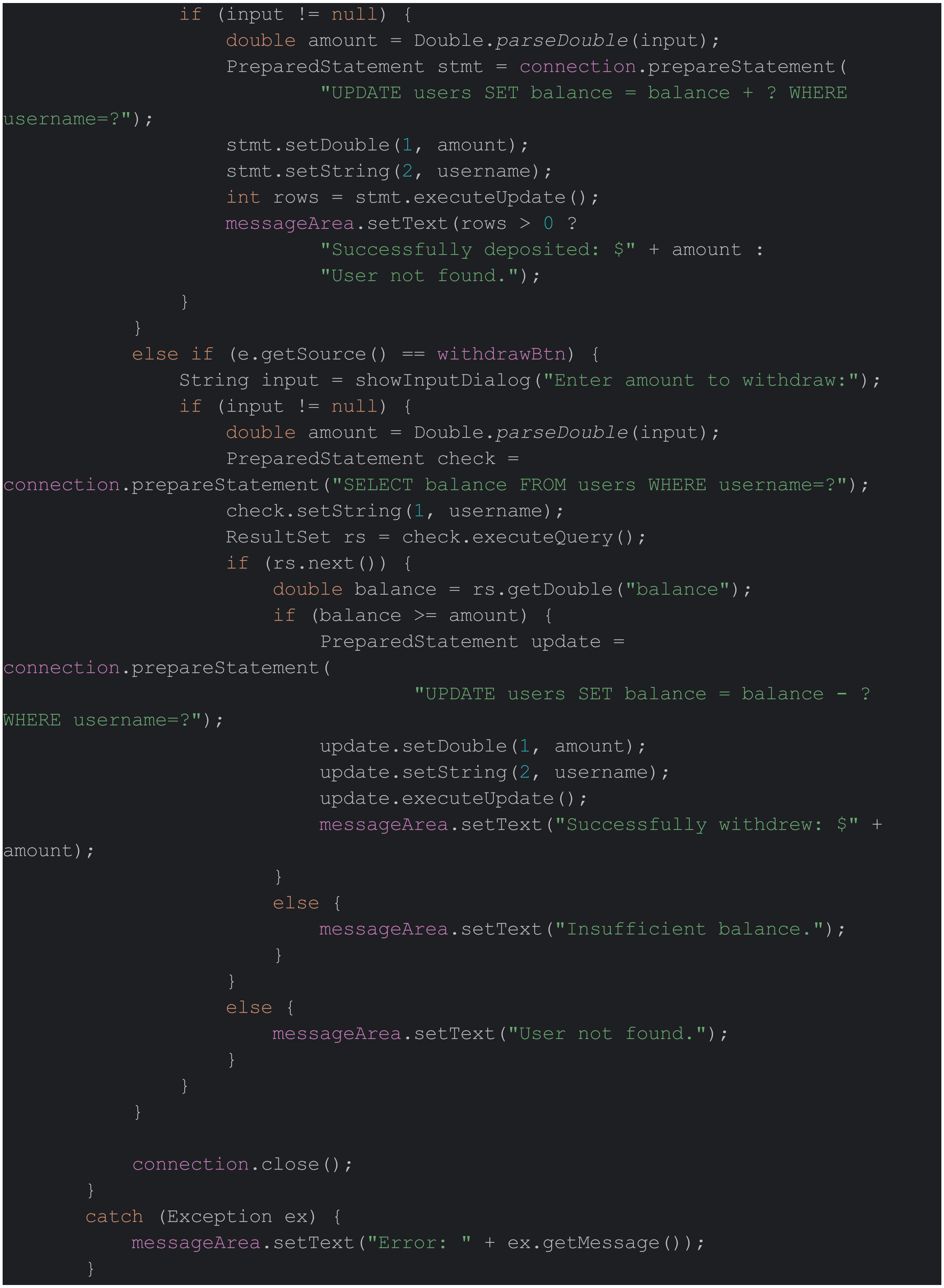
Connection Code

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

class ATM {
   public static void main(String[] args) {
       String url = "jdbc:mysql://127.0.0.1:3306/ATM";
       String user = "root";
       String password = "vineetha";

       try {
           // Load the MySQL JDBC driver
           Class.forName("com.mysql.cj.jdbc.Driver");
           // Establish the connection
           Connection conn = DriverManager.getConnection(url, user, password);
           System.out.println("Connection established successfully!");
       } catch (ClassNotFoundException e) {
           System.out.println("MySQL JDBC Driver not found.");
           e.printStackTrace();
       } catch (SQLException e) {
           System.out.println("Connection failed: " + e.getMessage());
           e.printStackTrace();
       }
   }
}
```

GUI Code

```java
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

class ATMGui extends Frame implements ActionListener {
   TextField usernameField, pinField;
   TextArea messageArea;
   Button loginBtn, balanceBtn, depositBtn, withdrawBtn;
   Connection connection;

   public ATMGui() {
       setTitle("ATM Simulation");
       setSize(500, 300);
       setLayout(new BorderLayout(10, 10));
```

```
        setLocationRelativeTo(null);

        Panel inputPanel = new Panel(new FlowLayout(FlowLayout.LEFT, 10, 10));
        inputPanel.add(new Label("Username:"));
        usernameField = new TextField(10);
        inputPanel.add(usernameField);
        inputPanel.add(new Label("PIN:"));
        pinField = new TextField(6);
        pinField.setEchoChar('*');
        inputPanel.add(pinField);
        add(inputPanel, BorderLayout.NORTH);

        Panel buttonPanel = new Panel(new FlowLayout(FlowLayout.CENTER, 15,
10));
        loginBtn = new Button("Login");
        balanceBtn = new Button("Check Balance");
        depositBtn = new Button("Deposit");
        withdrawBtn = new Button("Withdraw");

        buttonPanel.add(loginBtn);
        buttonPanel.add(balanceBtn);
        buttonPanel.add(depositBtn);
        buttonPanel.add(withdrawBtn);

        messageArea = new TextArea("", 4, 40,
TextArea.SCROLLBARS_VERTICAL_ONLY);
        messageArea.setEditable(false);

        Panel messagePanel = new Panel(null);
        messagePanel.setPreferredSize(new Dimension(450, 110));

        Panel innerPanel = new Panel(new BorderLayout());
        innerPanel.setBounds(10, 5, 480, 150);
        innerPanel.setBackground(Color.WHITE);
        innerPanel.add(messageArea, BorderLayout.CENTER);

        messagePanel.add(innerPanel);


        Panel centerPanel = new Panel(new BorderLayout());
        centerPanel.add(buttonPanel, BorderLayout.NORTH);
        centerPanel.add(messagePanel, BorderLayout.CENTER);
        add(centerPanel, BorderLayout.CENTER);


        loginBtn.addActionListener(this);
        balanceBtn.addActionListener(this);
        depositBtn.addActionListener(this);
        withdrawBtn.addActionListener(this);
        setVisible(true);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }
    public void actionPerformed(ActionEvent e) {
        String username = usernameField.getText();
        String pin = pinField.getText();

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection(

"jdbc:mysql://127.0.0.1:3306/ATM?useSSL=false&serverTimezone=UTC",
                    "root", "vineetha");

            if (e.getSource() == loginBtn) {
                PreparedStatement stmt = connection.prepareStatement("SELECT *
FROM users WHERE username=? AND pin=?");
                stmt.setString(1, username);
                stmt.setString(2, pin);
                ResultSet rs = stmt.executeQuery();
                if (rs.next()) {
                    messageArea.setText("Login successful! Welcome, " +
username);
                }
                else {
                    messageArea.setText("Invalid username or PIN.");
                }
            }
            else if (e.getSource() == balanceBtn) {
                PreparedStatement stmt = connection.prepareStatement("SELECT
balance FROM users WHERE username=?");
                stmt.setString(1, username);
                ResultSet rs = stmt.executeQuery();
                if (rs.next()) {
                    double balance = rs.getDouble("balance");
                    messageArea.setText("Your balance is: $" + balance);
                }
                else {
                    messageArea.setText("User not found.");
                }
            }
            else if (e.getSource() == depositBtn) {
                String input = showInputDialog("Enter amount to deposit:");
```



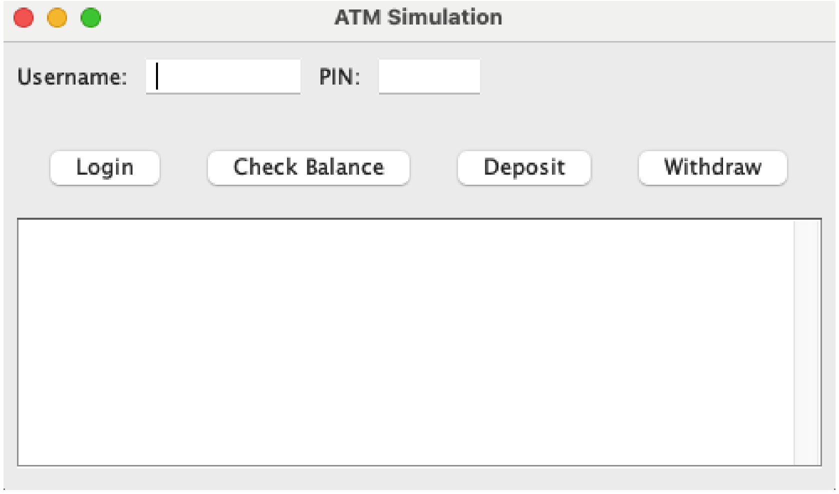
```
                if (input != null) {
                    double amount = Double.parseDouble(input);
                    PreparedStatement stmt = connection.prepareStatement(
                            "UPDATE users SET balance = balance + ? WHERE
username=?");
                    stmt.setDouble(1, amount);
                    stmt.setString(2, username);
                    int rows = stmt.executeUpdate();
                    messageArea.setText(rows > 0 ?
                            "Successfully deposited: $" + amount :
                            "User not found.");
                }
            }
            else if (e.getSource() == withdrawBtn) {
                String input = showInputDialog("Enter amount to withdraw:");
                if (input != null) {
                    double amount = Double.parseDouble(input);
                    PreparedStatement check =
connection.prepareStatement("SELECT balance FROM users WHERE username=?");
                    check.setString(1, username);
                    ResultSet rs = check.executeQuery();
                    if (rs.next()) {
                        double balance = rs.getDouble("balance");
                        if (balance >= amount) {
                            PreparedStatement update =
connection.prepareStatement(
                                    "UPDATE users SET balance = balance - ?
WHERE username=?");
                            update.setDouble(1, amount);
                            update.setString(2, username);
                            update.executeUpdate();
                            messageArea.setText("Successfully withdrew: $" +
amount);
                        }
                        else {
                            messageArea.setText("Insufficient balance.");
                        }
                    }
                    else {
                        messageArea.setText("User not found.");
                    }
                }
            }

            connection.close();
        }
        catch (Exception ex) {
            messageArea.setText("Error: " + ex.getMessage());
        }
```

```
    }

    private String showInputDialog(String message) {
        return new InputDialog(this, message).getInput();
    }

    public static void main(String[] args) {
        new ATMGui();
    }
}
class InputDialog extends Dialog implements ActionListener {
    private TextField inputField;
    private String input;

    public InputDialog(Frame parent, String message) {
        super(parent, "Input", true);
        setLayout(new FlowLayout());
        add(new Label(message));
        inputField = new TextField(10);
        add(inputField);
        Button ok = new Button("OK");
        ok.addActionListener(this);
        add(ok);
        setSize(250, 100);
        setLocationRelativeTo(parent);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        input = inputField.getText();
        dispose();
    }

    public String getInput() {
        return input;
    }
}
```

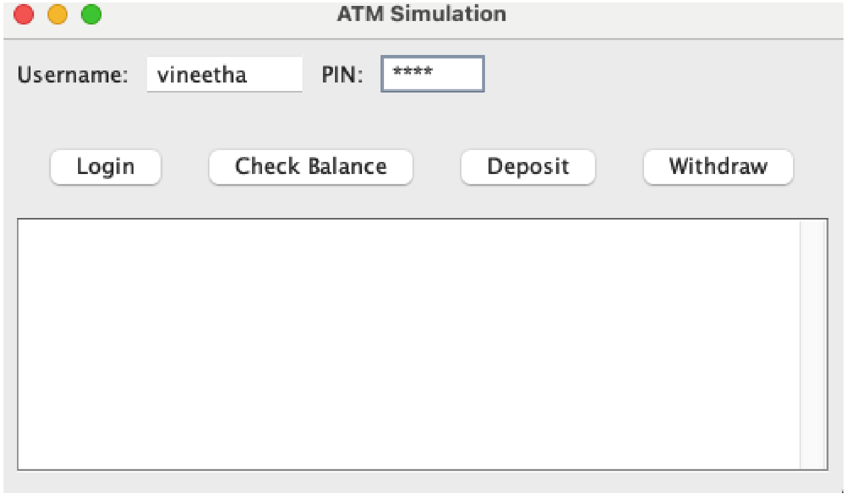
RESULTS AND ANALYSIS

The ATM Simulation System was successfully implemented and tested to perform the core functionalities of a real-world ATM.

Basic AWT interface:



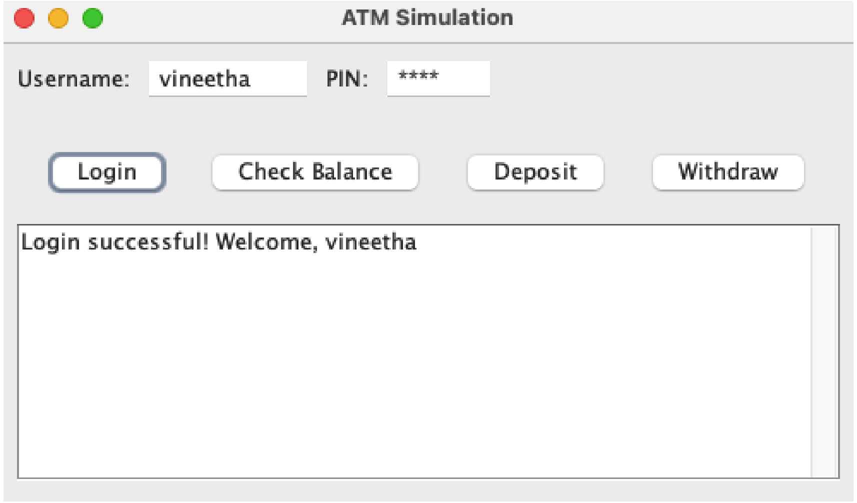
Credentials:



Username: vineetha

PIN: ****

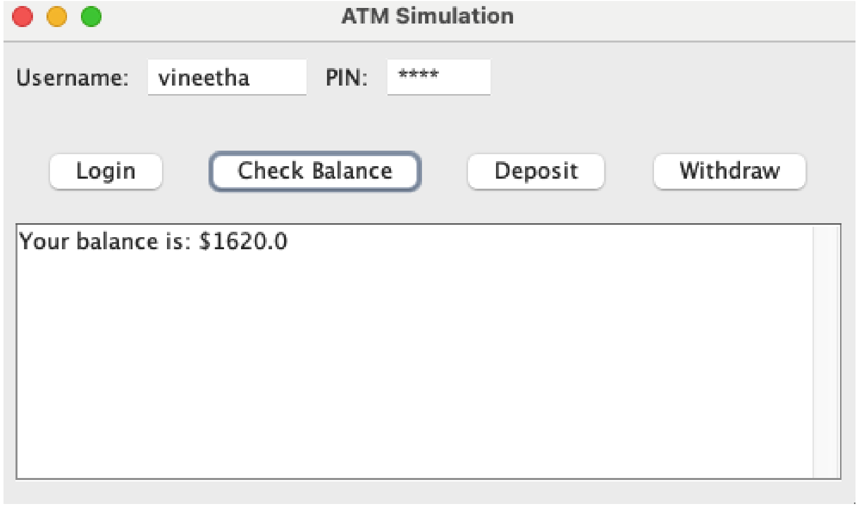
Login:



Username: vineetha PIN: ****

Login successful! Welcome, vineetha

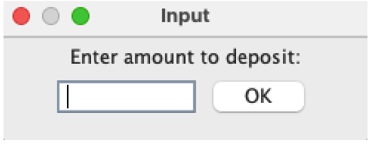
Balance Inquiry:



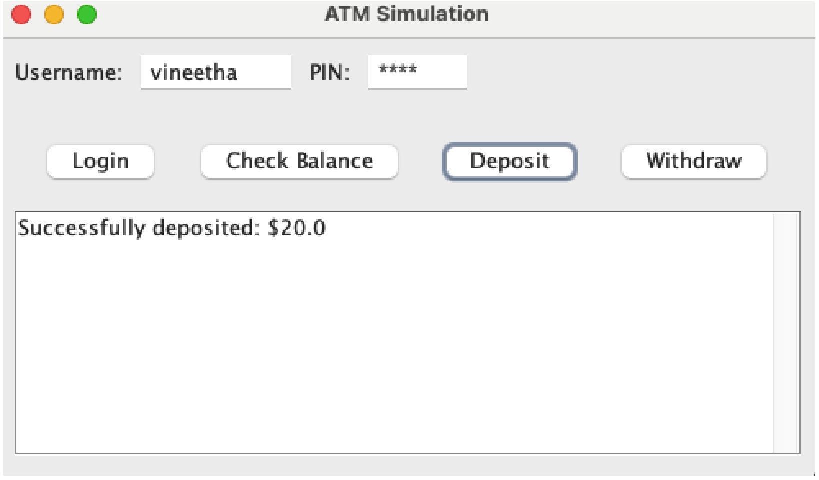
Username: vineetha PIN: ****

Your balance is: $1620.0

Deposit:



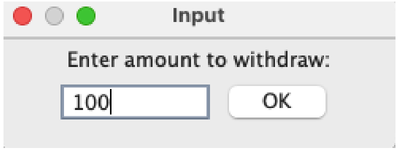
Enter amount to deposit:



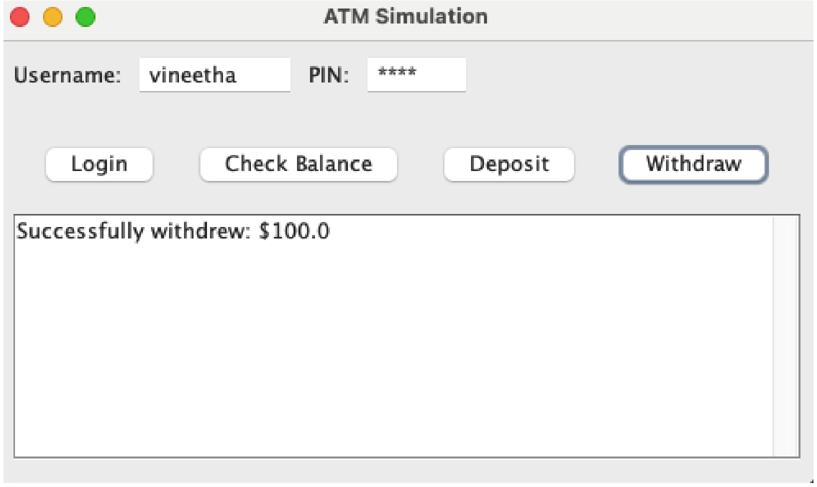
Username: vineetha PIN: ****

Successfully deposited: $20.0

Withdraw:



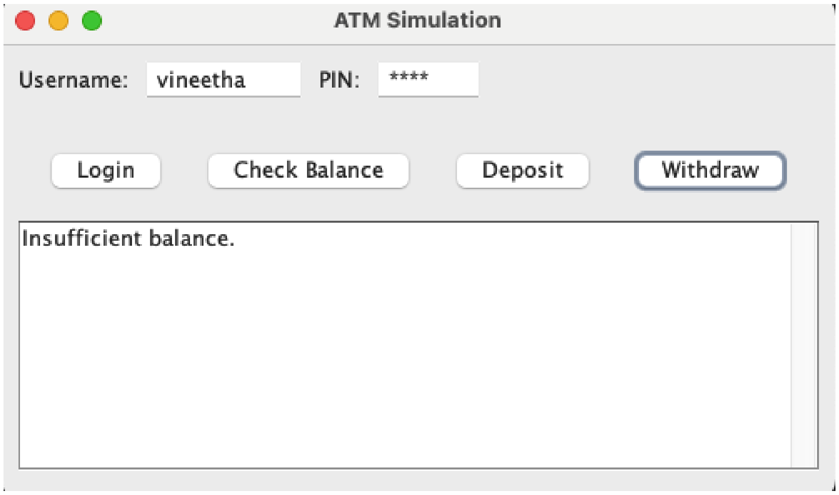
Enter amount to withdraw: 100



Username: vineetha PIN: ****

Successfully withdrew: $100.0

If withdrawal is out of limit:



Username: vineetha PIN: ****

```
Insufficient balance.
```

Overall, the implementation of this project not only reinforced concepts of Java programming and GUI development but also provided practical experience with database management and application design. It successfully simulates real-world ATM operations in a simplified desktop environment.

CONCLUSION

The ATM Simulation System uses Java, AWT, JDBC, and MySQL. The project exemplifies the applied usage of basic Java concepts, graphical user interface creation, and database connectivity. Users are able to execute the fundamental banking functions through this simulation.The system met all its specified goals, providing a secure and easy-to-use interface that communicates smoothly with a backend database. Every transaction is processed in real-time and updated correctly in the database, maintaining consistency and security.

Not only is this project an educational instrument to learn how ATM systems work, but also a starting point for further developments. With added development, functions such as multi-user support, receipt of transactions, and security enhancements could be included to increase the robustness of the system and bring it closer to commercially available ATM software.

In summary, the project has been an invaluable learning experience, solidifying programming skills, database management, and system design principles, while producing a working and interactive simulation of an ATM system.