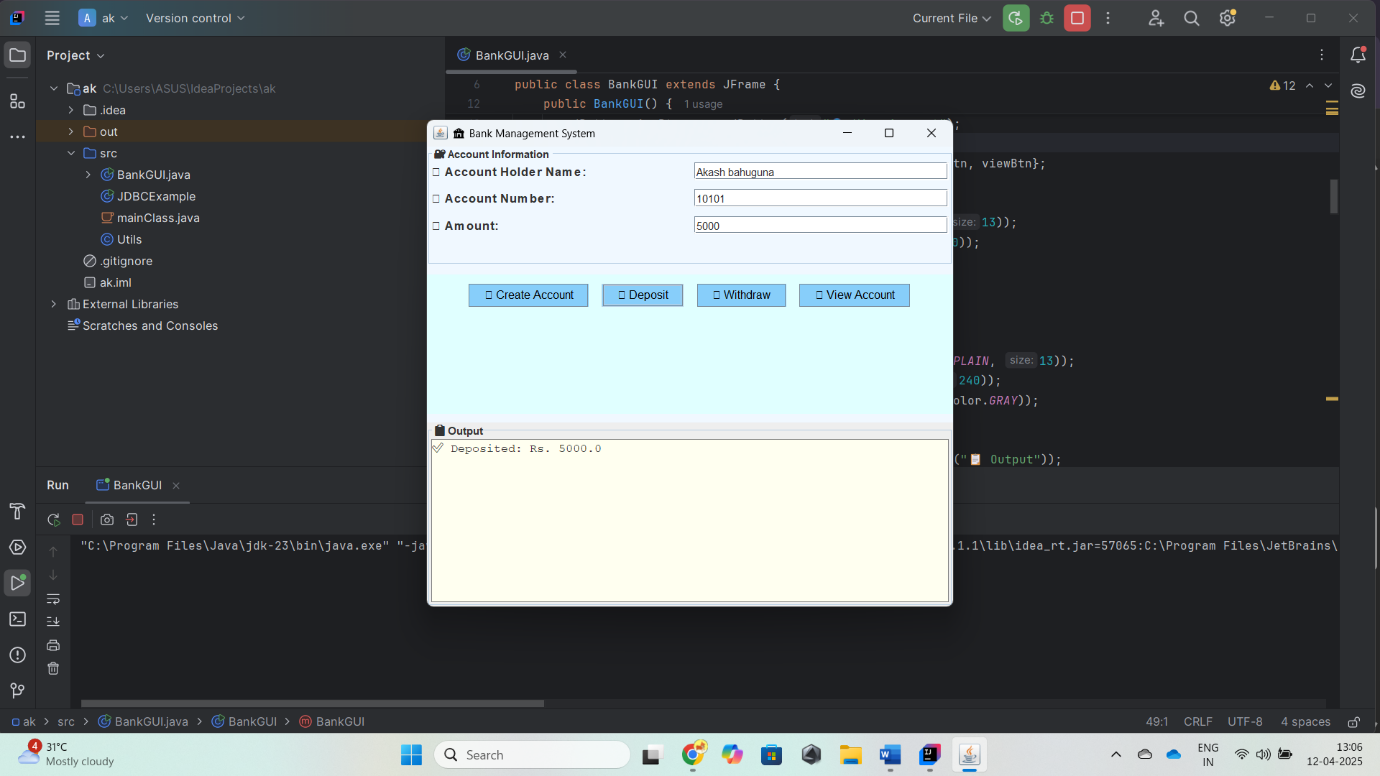
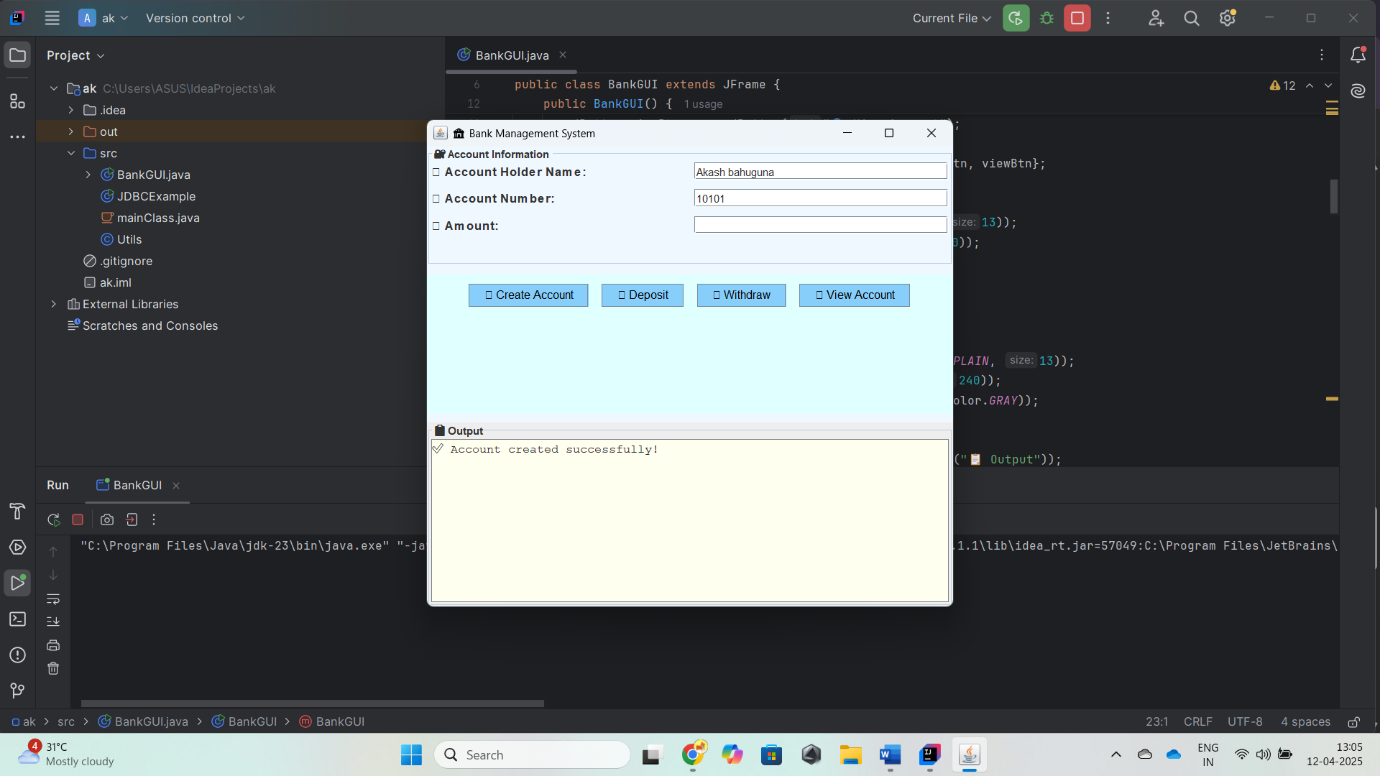
**Implementation**

**Sample Code Snippet:**

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import java.util.ArrayList;  
  
public class BankGUI extends JFrame {  
 private ArrayList<BankAccount> accounts = new ArrayList<>();  
  
 private JTextField nameField, numberField, amountField;  
 private JTextArea outputArea;  
  
 public BankGUI() {  
 setTitle("🏦 Bank Management System");  
 setSize(600, 550);  
 setDefaultCloseOperation(JFrame.*EXIT\_ON\_CLOSE*);  
 setLocationRelativeTo(null);  
 setLayout(new BorderLayout(10, 10));  
 getContentPane().setBackground(new Color(240, 248, 255));  
  
 JPanel topPanel = new JPanel(new GridLayout(4, 2, 10, 10));  
 topPanel.setBorder(BorderFactory.*createTitledBorder*("🔐 Account Information"));  
 topPanel.setBackground(new Color(240, 248, 255));  
  
 JLabel nameLabel = new JLabel("👤 Account Holder Name:");  
 nameLabel.setFont(new Font("Arial", Font.*BOLD*, 14));  
 topPanel.add(nameLabel);  
 nameField = new JTextField();  
 topPanel.add(nameField);  
  
 JLabel numberLabel = new JLabel("🔢 Account Number:");  
 numberLabel.setFont(new Font("Arial", Font.*BOLD*, 14));  
 topPanel.add(numberLabel);  
 numberField = new JTextField();  
 topPanel.add(numberField);  
  
 JLabel amountLabel = new JLabel("💵 Amount:");  
 amountLabel.setFont(new Font("Arial", Font.*BOLD*, 14));  
 topPanel.add(amountLabel);  
 amountField = new JTextField();  
 topPanel.add(amountField);  
  
 JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.*CENTER*, 15, 10));  
 buttonPanel.setBackground(new Color(224, 255, 255));  
  
 JButton createBtn = new JButton("➕ Create Account");  
 JButton depositBtn = new JButton("💰 Deposit");  
 JButton withdrawBtn = new JButton("💸 Withdraw");  
 JButton viewBtn = new JButton("🔍 View Account");  
  
 JButton[] buttons = {createBtn, depositBtn, withdrawBtn, viewBtn};  
 for (JButton btn : buttons) {  
 btn.setFocusPainted(false);  
 btn.setFont(new Font("Arial", Font.*PLAIN*, 13));  
 btn.setBackground(new Color(135, 206, 250));  
 btn.setForeground(Color.*BLACK*);  
 buttonPanel. Add(btn);  
 }  
  
 outputArea = new JTextArea(10, 40);  
 outputArea.setFont(new Font("Monospaced", Font.*PLAIN*, 13));  
 outputArea.setBackground(new Color(255, 255, 240));  
 outputArea.setBorder(BorderFactory.*createLineBorder*(Color.*GRAY*));  
 outputArea.setEditable(false);  
 JScrollPane scrollPane = new JScrollPane(outputArea);  
 scrollPane.setBorder(BorderFactory.*createTitledBorder*("📋 Output"));  
  
 add(topPanel, BorderLayout.*NORTH*);  
 add(buttonPanel, BorderLayout.*CENTER*);  
 add(scrollPane, BorderLayout.*SOUTH*);  
  
 createBtn.addActionListener(e -> createAccount());  
 depositBtn.addActionListener(e -> deposit());  
 withdrawBtn.addActionListener(e -> withdraw());  
 viewBtn.addActionListener(e -> viewAccount());  
  
 setVisible(true);  
 }  
  
 private void createAccount() {  
 String name = nameField.getText();  
 if (name.isEmpty() || numberField.getText().isEmpty()) {  
 outputArea.setText("❗ Please enter all details to create an account.\n");  
 return;  
 }  
  
 try {  
 int number = Integer.*parseInt*(numberField.getText());  
  
 for (BankAccount acc : accounts) {  
 if (acc.getAccountNumber() == number) {  
 outputArea.setText("⚠️ Account already exists!\n");  
 return;  
 }  
 }  
  
 accounts.add(new BankAccount(name, number));  
 outputArea.setText("✅ Account created successfully!\n");  
 } catch (NumberFormatException e) {  
 outputArea.setText("❌ Invalid account number. Please enter digits only.\n");  
 }  
 }  
  
 private void deposit() {  
 try {  
 int number = Integer.*parseInt*(numberField.getText());  
 double amt = Double.*parseDouble*(amountField.getText());  
  
 for (BankAccount acc : accounts) {  
 if (acc.getAccountNumber() == number) {  
 acc.deposit(amt);  
 outputArea.setText("✅ Deposited: Rs. " + amt + "\n");  
 return;  
 }  
 }  
 outputArea.setText("❌ Account not found!\n");  
 } catch (NumberFormatException e) {  
 outputArea.setText("❌ Please enter valid numbers in account number and amount fields.\n");  
 }  
 }  
  
 private void withdraw() {  
 try {  
 int number = Integer.*parseInt*(numberField.getText());  
 double amt = Double.*parseDouble*(amountField.getText());  
  
 for (BankAccount acc : accounts) {  
 if (acc.getAccountNumber() == number) {  
 if (acc.getBalance() >= amt) {  
 acc.withdraw(amt);  
 outputArea.setText("✅ Withdrawn: Rs. " + amt + "\n");  
 } else {  
 outputArea.setText("❗ Insufficient balance.\n");  
 }  
 return;  
 }  
 }  
 outputArea.setText("❌ Account not found!\n");  
 } catch (NumberFormatException e) {  
 outputArea.setText("❌ Please enter valid numbers in account number and amount fields.\n");  
 }  
 }  
  
 private void viewAccount() {  
 try {  
 int number = Integer.*parseInt*(numberField.getText());  
  
 for (BankAccount acc : accounts) {  
 if (acc.getAccountNumber() == number) {  
 outputArea.setText("👤 Account Holder: " + acc.getAccountHolderName() +  
 "\n🔢 Account Number: " + acc.getAccountNumber() +  
 "\n💰 Balance: Rs. " + acc.getBalance() + "\n");  
 return;  
 }  
 }  
 outputArea.setText("❌ Account not found!\n");  
 } catch (NumberFormatException e) {  
 outputArea.setText("❌ Invalid account number format.\n");  
 }  
 }  
  
 public static void main(String[] args) {  
 SwingUtilities.*invokeLater*(BankGUI::new);  
 }  
}  
  
class BankAccount {  
 private String accountHolderName;  
 private int accountNumber;  
 private double balance;  
  
 public BankAccount(String name, int number) {  
 this.accountHolderName = name;  
 this.accountNumber = number;  
 this.balance = 0.0;  
 }  
  
 public void deposit(double amount) {  
 balance += amount;  
 }  
  
 public void withdraw(double amount) {  
 if (amount <= balance) {  
 balance -= amount;  
 }  
 }  
  
 public String getAccountHolderName() {  
 return accountHolderName;  
 }  
  
 public int getAccountNumber() {  
 return accountNumber;  
 }  
  
 public double getBalance() {  
 return balance;  
 }  
}

**OUTPUT** A screenshot of a computer

AI-generated content may be incorrect.