JAVA PROJECT REPORT

(Project Term January-May 2023)

BANK MANGMENT SYSTEM

Submitted by:

NAME	ROLL NO.	Registration No.
Ranjan Alok	40	12113780
Ashish Kumar	42	12114307
Rahul Singh	38	12113266

Project Group Number: …….

Course Code: CSE310

Under the Guidance of
Dr. Ranjith Kumar A
Assistant Professor

School of Computer Science and Engineering
Phagwara, Punjab

DECLARATION

We hereby declare that the project work entitled ("Online Banking System") is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science and Engineering from Lovely Professional University, Phagwara, under the guidance of Dr. Ranjith Kumar A, during January to April 2023. All the information furnished in this capstone project report is based on our own intensive work and is genuine.

Project Group Number: ……

Name of Student 1: Ashish Kumar
Registration Number: 12114307

Name of Student 2: Ranjan Alok
Registration Number: 12113780

Name of Student 3: Rahul Singh
Registration Number: 12113266

Ashish Kumar
Date:

Ranjan Alok
Date:

Rahul Singh
Date:

Table of Content

INTRODUCTION

Online Banking (Internet Banking or E-banking) allows customers of a financial institution to contact financial transactions on a secured website operated by the institution, which can be are tail bank, virtual bank, credit union or building society.

A bank management system is a software application that is designed to assist banks in efficiently managing their day-to-day operations. This report aims to provide an overview of the development of a bank management system using Java programming language.

Java is a widely used programming language that provides a high degree of flexibility, security, and Portability. In developing this system, the Java programming language was used to create a user-friendly and efficient system for bank management.

The bank management system has several modules, including customer management, account management, transaction management, and employee management. These modules provide a range of functionalities, such as account opening, account maintenance, transaction processing, and reporting.

One of the key features of this system is its ability to handle multiple users and provide different levels of access to the system based on their roles. This ensures that only authorized personnel have access to sensitive information and operations.

The bank management system also includes several security features, such as encryption of data, authentication of users, and validation of transactions. These features help to prevent fraudulent activities and ensure the security of customer data.

Overall, the bank management system developed using Java is a robust and efficient system that can help banks to streamline their operations and provide better services to their customers. It is a scalable solution that can be customized to meet the specific needs of different banks, making it a valuable tool for the banking industry.

SOFTWARE REQUIREMENTS:

The major so requirements of the project are as flow

Operating system:Windows Xp

HARDWARE REQUIREMENTS:

The hardware requirements that map towards the software are

RAM: 256M

Processor: Intel

Mouse

Keyboard

REQUIREMENTS:

It is the process of determining user expectations for anew or modified product. These features

are called requirements, must be quantifiable, relevant and detailed.

There are mainly two types of requirements:

i. FUNCTIONAL REQUIREMENTS:

- New Entry()

The customer can add new entries. So that he/she can create a new account to

store money and follow further transactions.

- display()

The customer uses this in order to verify the total amount available in his/her

account in order to do further transactions or not.

- deposit( )

The customer uses this inorder to add a specific amount to his/her bank account.

So this option shows them the previous amount and final amount after deposit.

- withdraw( )

The customer uses this to withdraw a specific amount from his/her account. So

this option shows them the previous amount in the account and total amount

after withdraw along with some conditions.

ii. NON-FUNCTIONAL REQUIREMENTS:

EXCEPTION HANDLING:

The exception handling in java is one of the powerful mechanism to handle the runtime
errors so that normal flow of the application can be maintained.

a. Try:

Java try block is used to enclose the code that might throw an exception. It must be used within
the method. Java try block must be followed by either catch block

b. Catch:

Java catch block is used to handle the Exception. It must be used after the try block only.
You can use multiple catch block with a single try.

FILES:

The java.io package contains nearly every class you might ever need to perform input and

output (I/O) in Java. All these streams represent an input source and an output destination. The

stream in the java.io package supports many data

INPUT STREAM READER:

An InputStreamReader is a bridge from byte streams to character streams: It reads bytes and

translates them into characters according to a specified character encoding.

BUFFERED READER CLASS:

Buffered Reader class reads text from a character-input stream, buffering characters so as to

provide for the efficient reading of characters, arrays, and lines. Following are the important

points about Buffered Reader:

« The buffer size may be specified, or the default size may be used.

ABSTRACT:

The purpose of this project is in partial fulfilment of the requirements of customer using the

online banking for payment. The Design and development of this Bank Management system

provides a more secured approach in managing bank customer's information which strengthens

the relationships between banks and their customers by providing the right solutions that uses

a multilevel security to improve customer satisfaction. The programming language used to

develop this project is. Java.

The Domain "Banking System " keeps the day by day tally record as a complete banking. It

can keep the information of Account type, account opening form, Deposit, Withdrawal, and

Searching the transaction, Transaction report, Individual account opening form, Group

Account. The exciting part of this project is; it displays Transaction reports, Statistical

Summary of Account type and Interest Information.

SCOPE OF THE PROJECT:

The scope of a project titled "Bank Management System" would typically include designing, developing, and implementing a software system that can help banks manage their operations efficiently. The main objective of this project would be to provide a platform for managing all the core banking activities such as account management, transaction processing, loan management, customer relationship management, and reporting.

Here are some specific areas that would be covered under the scope of this project:

1.Account Management: The software system would allow the bank to manage their customer accounts, including account opening, closing, balance inquiries, and account transactions.

2. Transaction Processing: The system would enable the bank to process transactions such as deposits, withdrawals, transfers, and bill payments.

3. Loan Management: The software system would allow the bank to manage their loan portfolio, including loan application processing, loan disbursement, and loan repayment tracking.

4.Customer Relationship Management: The system would enable the bank to maintain a comprehensive customer database, manage customer interactions, and provide personalized services to customers.

5.Reporting: The software system would provide various reporting features, including account statements, transaction summaries, customer profiles, and financial statements.

6.Security: The system would have robust security measures in place to protect sensitive customer data, prevent fraud, and ensure regulatory compliance.

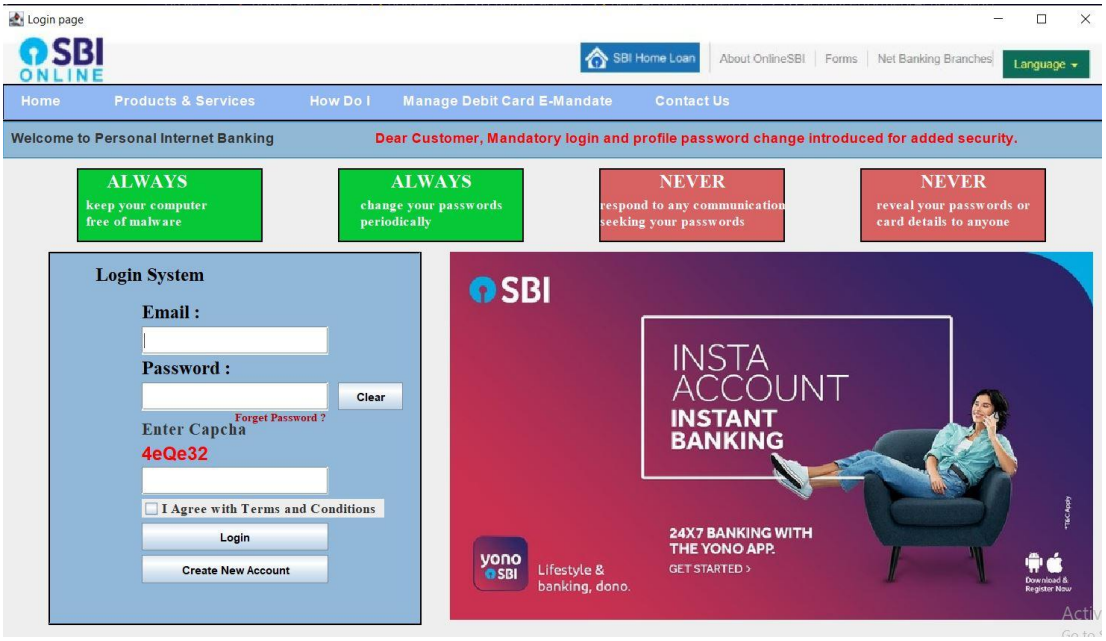
7.Integration: The software system would integrate with other banking systems and services, such as payment gateways, ATM networks, and mobile banking applications.

In summary, the scope of the "Bank Management System" project would cover all aspects of a modern banking system, providing an efficient and secure platform for banks to manage their operations and provide high-quality services to their customers.

Modeules

1. Login module

This module helps to make a login page so that the user can enter the user name and password to enter into the bank management system in java. Name the file as **homepage.java**



Three Button added in login module.

1. Login: - It will return new window page.

2. Clear: - It will erase text written in text area.

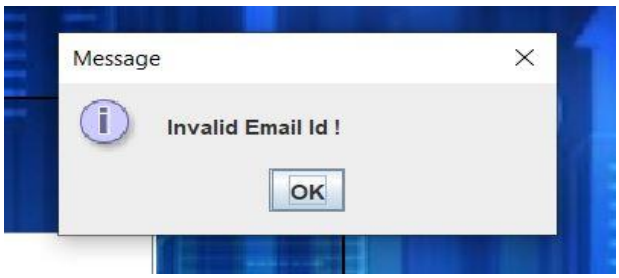
3. Create new Account: - New user entry.

4. Forget Password: - If a user forgets his/her password he can change it through this button.

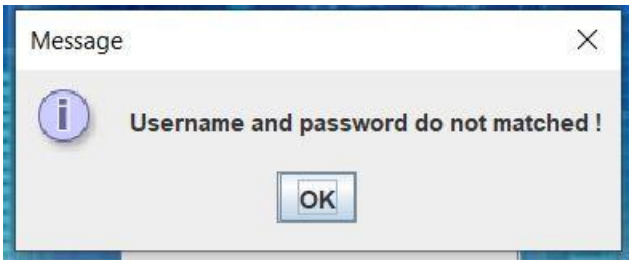
This page contains three text area:

1. Email: User enters the email Id that is registered in database of the bank.
2. Password: User enters the password that is registered in database of the bank.
3. Enter Capcha: This is required field of this page, user have to enter captcha as a identification between an human and AI.

When the entered user email didn't match with pattern of email, it will return the "Invalid email id".



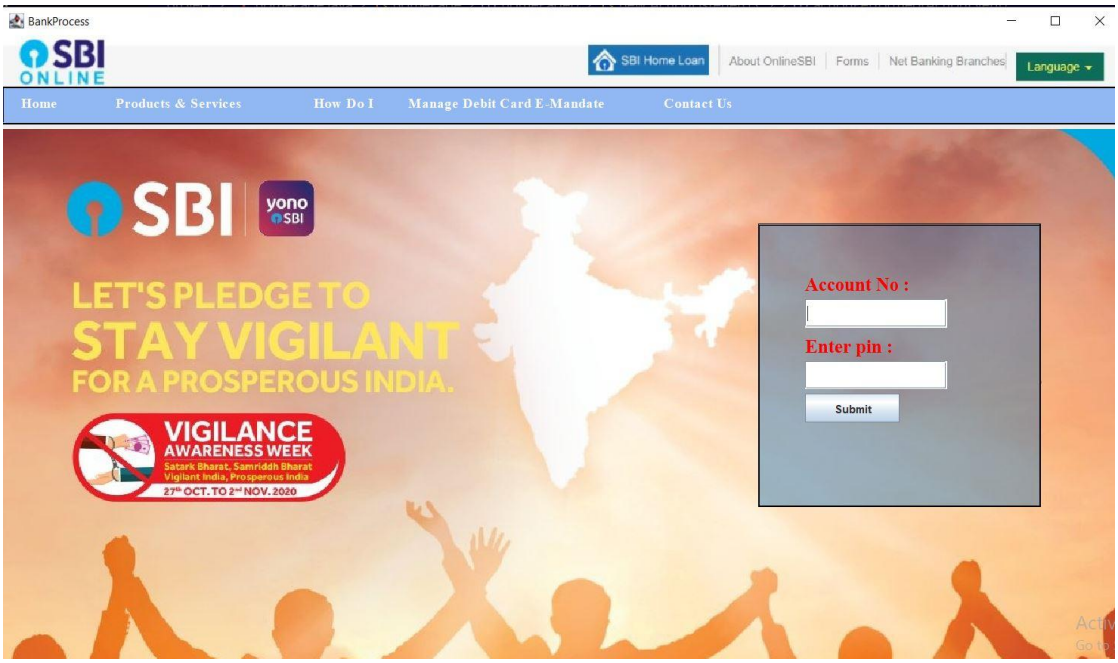
When the entered user name and password didn't match the existing records in the database, it will return the "Username and password did not match".



Whenever user enter the correct Email id, Password and captcha user is directed to **BankProcess.java** module where real authentic work has to be done.

2.Bank Process – Module:

In this module real validation and authentication work is done on textfeilds Account Number and Pin.



Text fields in this page are:

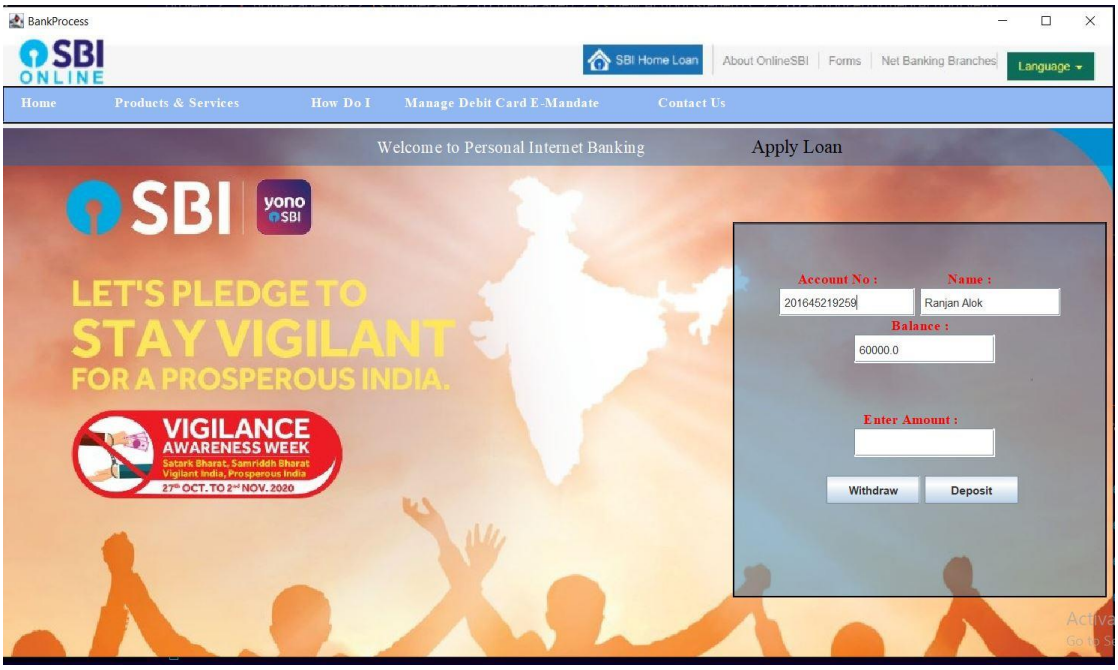
1. Account No: User need to enter account number provided by the bank.
2. Enter Pin: User need to add pin of there account which is same as their ATM pin.

This page contains 1 Button:

1. Submit: Submit button will lead to transaction page if user enters correct account number and pin.

3.Transction Module:

On clicking the submit button of previous module user gets in the transaction module.



In transaction module user gets an option of checking their saving account balance, user also gets the option of Withdraw and Deposit from their Saving account balance.

Text Field in this module are:

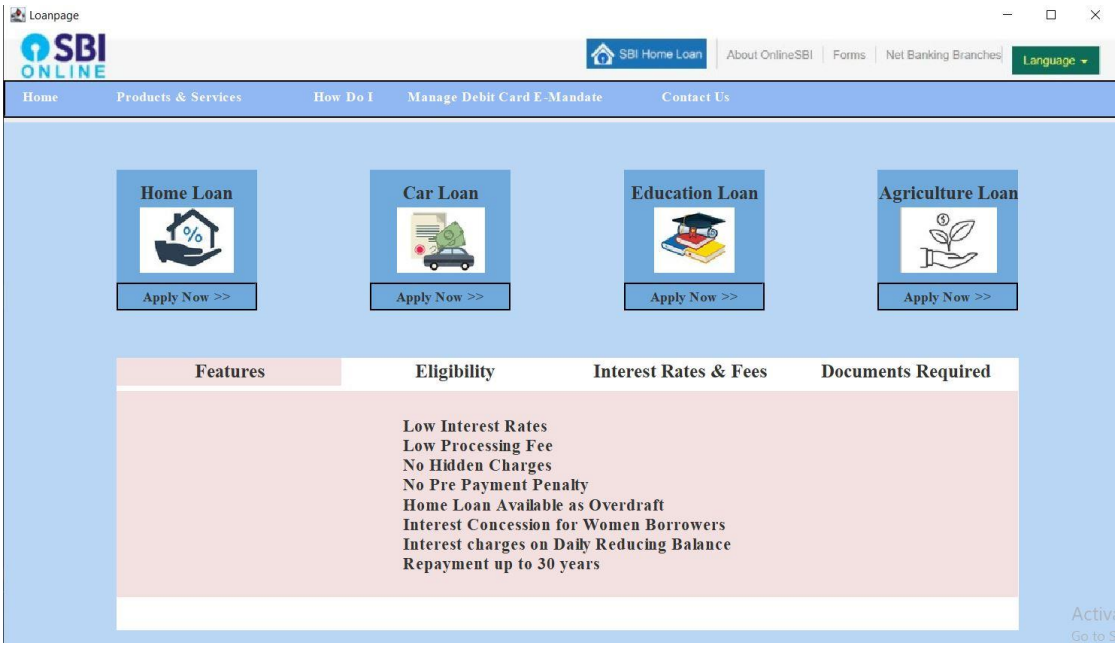
1. Account No: User have to enter account no. to look saving balance of your account.
2. Name: User need to enter their name to fetch correct information from database.
3. Enter Amount: User enters the amount of money to be withdrawn or deposited in this module.

Buttons in this module:

1. Withdraw: This button allows the user to withdraw the required amount from the bank and use it in other upi payment apps.
2. Deposit: This button allows the user to deposit the required amount in the bank.
3. Apply Loan: This button allow user to apply for a loan, it will redirect user to move into the Loan module.

4.Loan Module:

A loan module is a feature within an online banking system that allows customers to apply for and manage loans through the bank's website or mobile app.



This module gives various choices to the user regarding the type of loan he/she want to apply.

The icons present in this module are:

1. Home Loan:
2. Car Loan:
3. Education Loan:
4. Agriculture Loan:

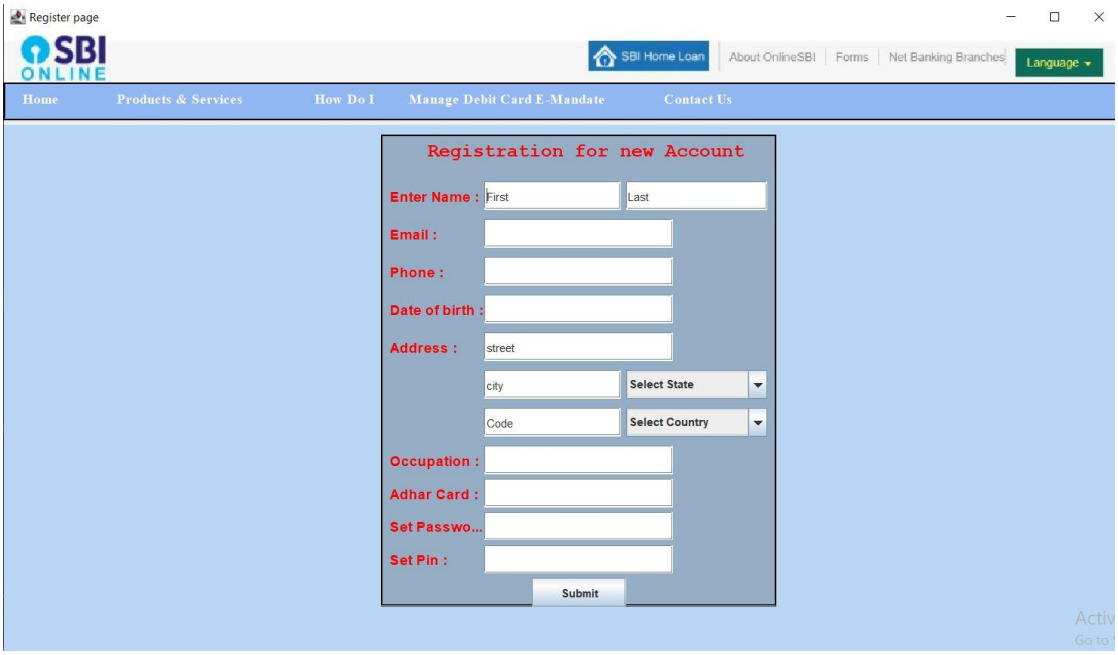
   Each icon has a apply button present in it, which will redirect the user to ApplyLoan – Module.

This page also shows Rules and Regulation for applying to the loan.

It has four buttons that is Features, Eligibility, Interest Rates & Fees and Documents Required. On clicking these buttons user gets detailed information about the Features, Eligibility, Interest Rates & Fees and Documents Required of the Loan.

5.Registeration - Module:

A registration module for an online banking system typically involves collecting information from the user and verifying their identity to create a new account. The module can include the following steps:



User Information: The user is prompted to provide their personal information such as full name, address, contact number, email address, and date of birth.

User Identification: The system may require the user to provide government-issued identification documents such as a passport or driver's license to verify their identity.

Account Information: The user is prompted to create a username and password for their account. They may also be required to answer security questions or set up a security token.

Confirmation: Once the user has submitted their information, the system can send a confirmation email or message to the registered email address or phone number to confirm their registration.

Account Activation: The user can activate their account by clicking on the confirmation link sent to their email or following the instructions in the confirmation message.

Additional Verification: The system may require additional verification for certain transactions such as adding a new payee or transferring a large amount of money.

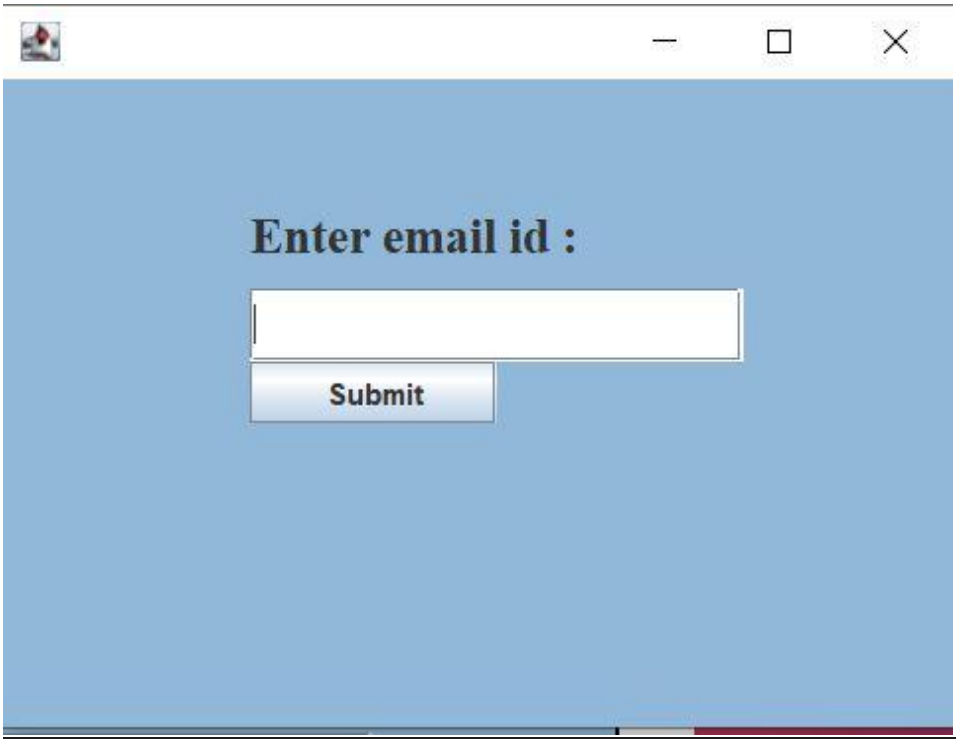
Overall, the registration module for an online banking system needs to ensure the security and privacy of user data while providing a smooth and efficient registration process.

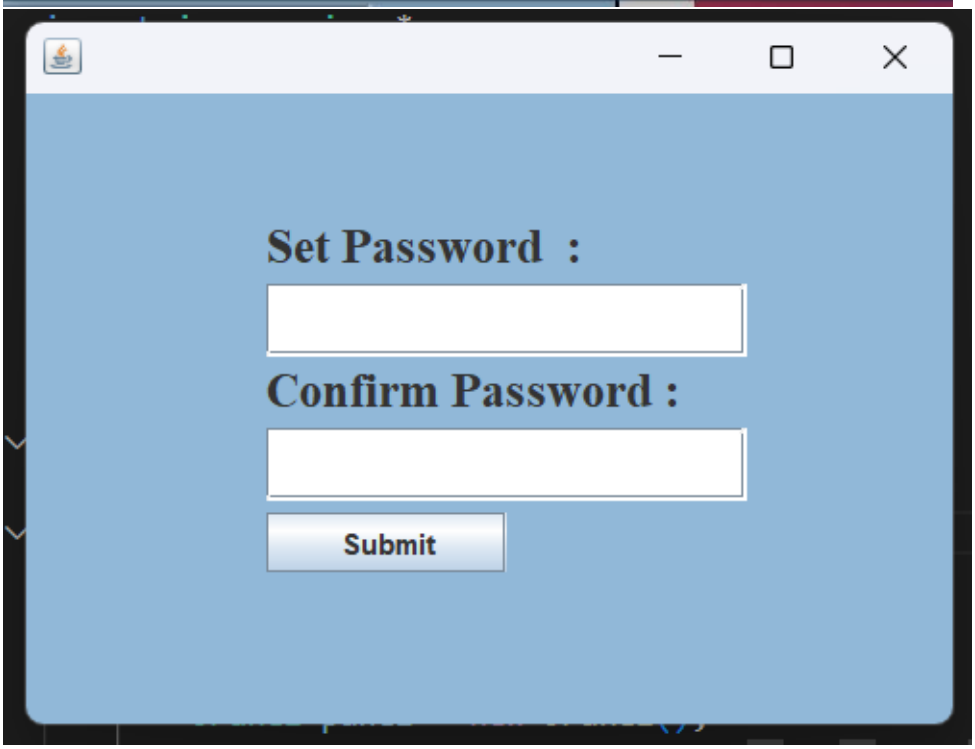
This module asks user to enter details like name, email, phone no, date of birth, address, occupation, aadhar card and also an option of Set password and set Pin for the new bank account.

Submit button in this page will record all the details of a new user to the database and create new account.

6.ChangePass – Module:

The change password module in an online banking system is a crucial component that allows users to update their login credentials for security purposes. It is important to ensure that this module is designed with user-friendly features, while also providing robust security measures to prevent unauthorized access to user accounts.





This module asks user to enter email id which is registered to the database of the bank, on clicking Submit button user moves to a new screen which asks user to set New password and rewrite the confirmed password.

Some key features of a change password module for an online banking system might include:

Accessibility: The module should be easily accessible from the user's dashboard or profile page, and should provide clear instructions on how to change the password.

Verification: To ensure the security of the user's account, the module should require users to provide verification, such as answering security questions or entering a verification code sent via email or SMS.

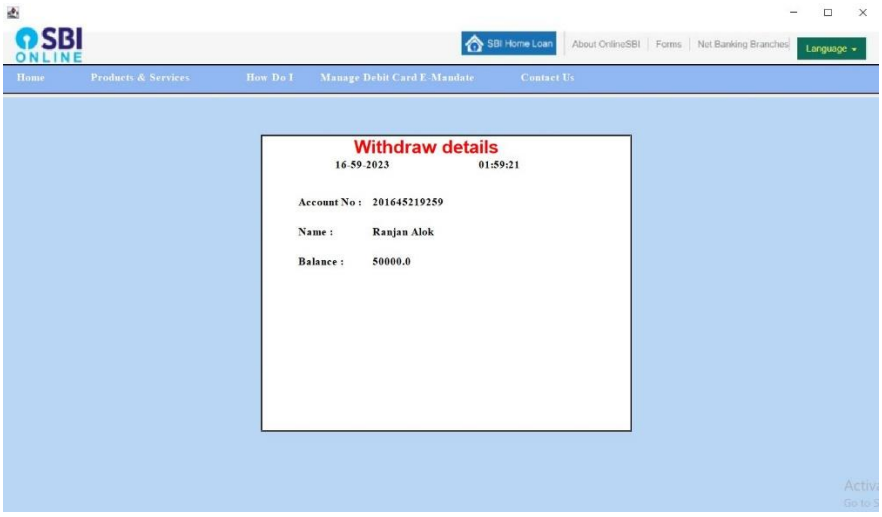
Password Requirements: The module should enforce strong password requirements to ensure that the user's new password is secure. This might include requiring a minimum length, a combination of letters, numbers, and special characters, and preventing the use of commonly-used passwords.

Confirmation: Once the user has successfully changed their password, the module should provide confirmation of the change, either via a message on the website or an email confirmation.

Overall, the change password module in an online banking system should be designed to provide a balance between ease of use and robust security measures. By ensuring that users can easily change their passwords while also enforcing strong security measures, the module can help to protect user accounts from unauthorized access and data breaches.

7.Receipt Module:

The receipt module in an online-banking system is a feature that allows users to view and download receipts for their transactions. When a user completes a transaction, such as transferring funds to another account, paying a bill, or depositing money, the system generates a receipt that confirms the details of the transaction.



The receipt typically contains information such as the date and time of the transaction, the amount of money transferred or deposited, the account numbers involved, and any fees or charges associated with the transaction. It may also include a reference number or transaction ID for future reference.

In an online-banking system, users can access their receipts through their account dashboard or transaction history. The system may also send a copy of the receipt to the user's email address or mobile phone for their records.

The receipt module provides users with an easy way to keep track of their financial transactions and maintain a record of their account activity. It also helps to ensure accuracy and transparency in the banking process, which can help to build trust and confidence among users.

Design:

Designing an online banking system involves creating a user-friendly interface for customers to access their accounts and manage their finances. The following are some important considerations when designing an online banking system:

Security: Security is a top priority when it comes to online banking. The system should have strong authentication mechanisms to ensure that only authorized users can access their accounts. This can include two-factor authentication, biometric authentication, and other security measures.

User Experience: The online banking system should be intuitive and easy to use. The user interface should be clean, simple, and easy to navigate. The user experience should be consistent across all devices and platforms.

Accessibility: The online banking system should be accessible to all users, including those with disabilities. This means providing alternative text for images, keyboard navigation, and other accessibility features.

Account Management: Users should be able to view their account balances, transactions, and statements. They should also be able to transfer funds, pay bills, and manage their accounts.

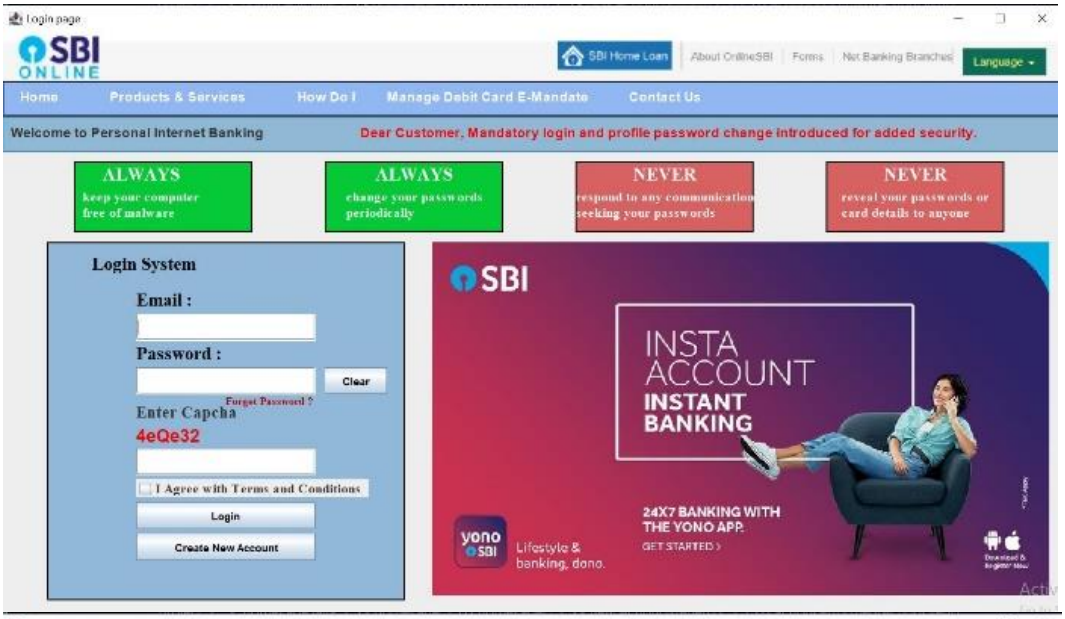
Mobile-Friendly: With more and more people accessing the internet through their mobile devices, it is important that the online banking system is mobile-friendly. This means designing a responsive interface that adjusts to different screen sizes and is optimized for touchscreens.

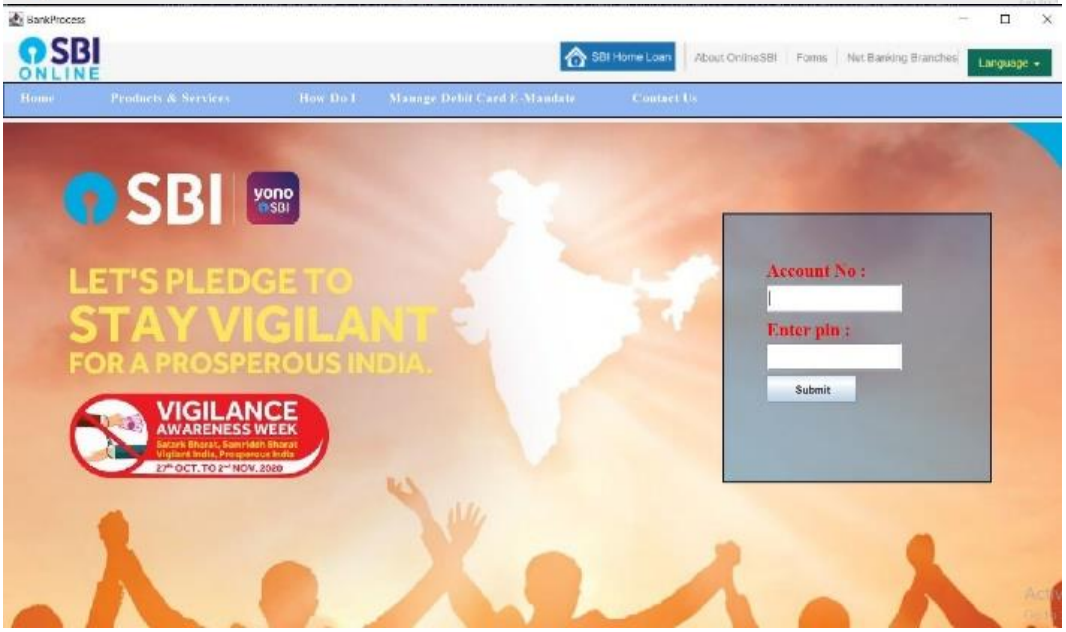
Integration: The online banking system should be integrated with other banking services, such as investment management and loan services. This allows customers to manage all their financial needs in one place.

Customer Support: The online banking system should have robust customer support mechanisms, including chatbots, phone support, and email support. This ensures that customers can get the help they need when they need it.

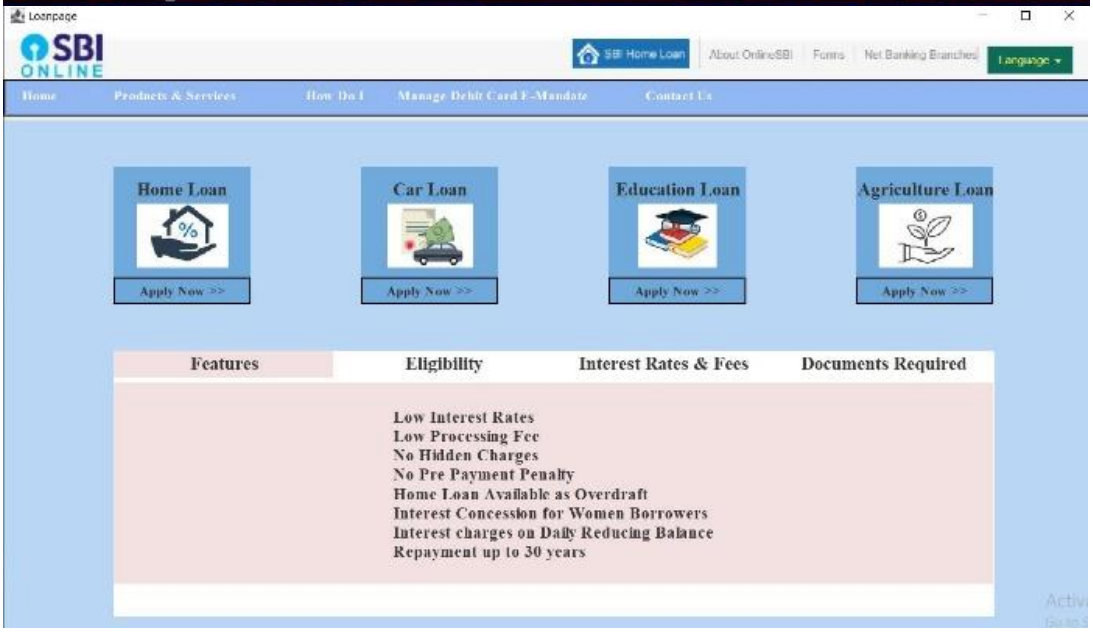
In summary, designing an online banking system requires careful consideration of security, user experience, accessibility, account management, mobile-friendliness, integration, and customer support. By keeping these factors in mind, designers can create a system that meets the needs of customers while providing a seamless and secure banking experience.

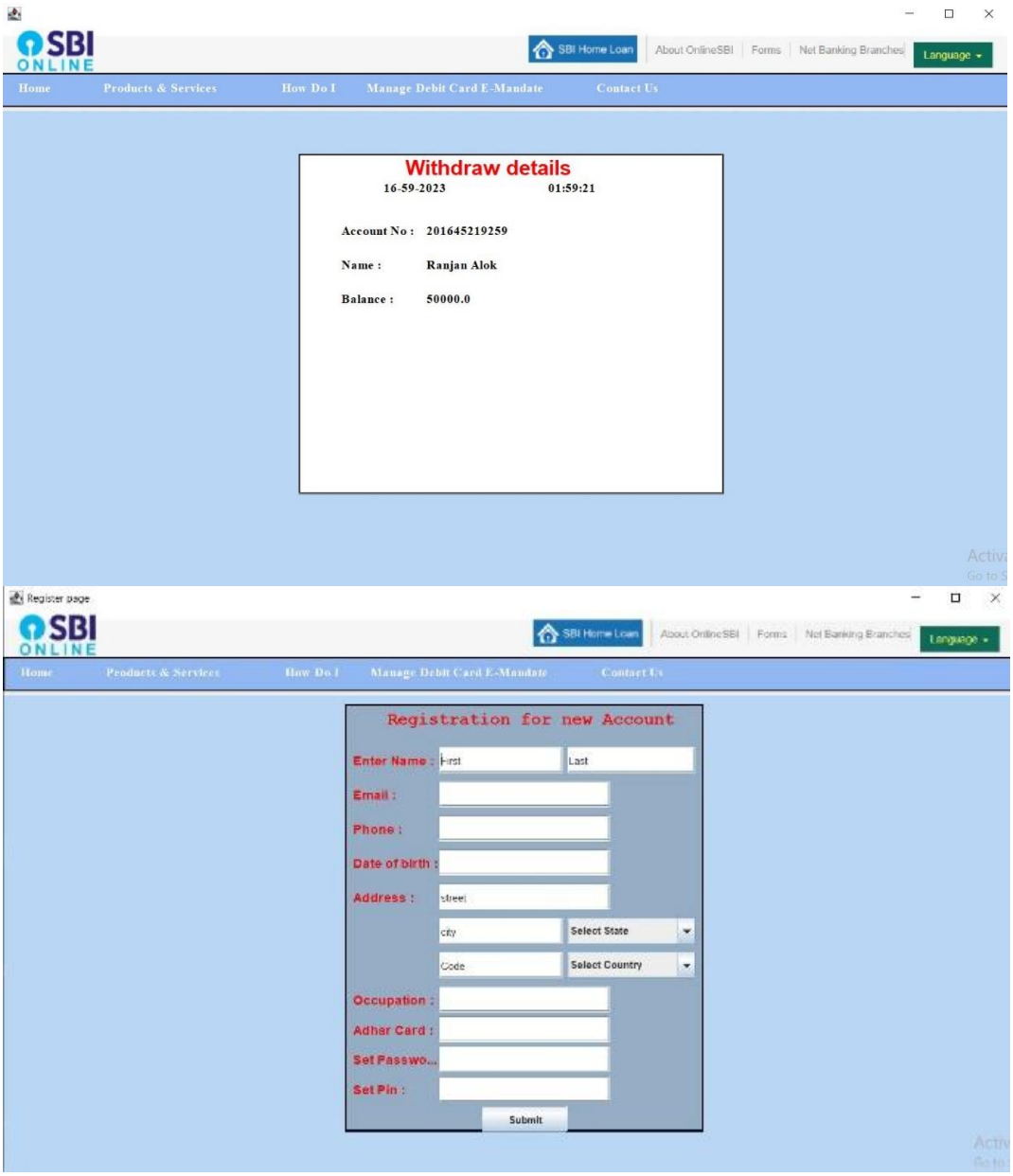
Screenshots:






BankProcess
SBI ONLINE
SBI Home Loan
About OnlineSBI
Forms
Net Banking Branches
Language
Home
Products & Services
How Do I
Manage Debit Card E-Mandate
Contact Us
Welcome to Personal Internet Banking
Apply Loan
SBI
yono SBI
LET'S PLEDGE TO
STAY VIGILANT
FOR A PROSPEROUS INDIA.
VIGILANCE AWARENESS WEEK
Account No :
Name :
Ranjan Alok
Balance :
Enter Amount :
Withdraw
Deposit


Loanpage
SBI ONLINE
SBI Home Loan
About OnlineSBI
Forms
Net Banking Branches
Language
Home
Products & Services
How Do I
Manage Debit Card E-Mandate
Contact Us
Home Loan
Apply Now >>
Car Loan
Apply Now >>
Education Loan
Apply Now >>
Agriculture Loan
Apply Now >>
Features
Eligibility
Interest Rates & Fees
Documents Required
Low Interest Rates
Low Processing Fee
No Hidden Charges
No Pre Payment Penalty
Home Loan Available as Overdraft
Interest Concession for Women Borrowers
Interest charges on Daily Reducing Balance
Repayment up to 30 years



Explaination:

The specific steps involved in an online banking system may vary depending on the particular bank and its software, but generally, the following steps are involved:

Registration: Customers must first register for online banking by providing their personal and account information to the bank. This information may include their name, address, phone number, email address, and account number.

Login: After registering, customers can log in to their online banking account using a username and password. Some banks may also require additional security measures, such as two-factor authentication or biometric authentication.

Account overview: Once logged in, customers can view their account balances, transaction history, and other account information. They may also be able to access additional services, such as bill payment, money transfer, and account management.

Transactions: Customers can initiate transactions, such as transferring money between accounts, paying bills, or depositing checks using their mobile device or computer.

Security: Online banking systems are designed to be secure, and banks take several measures to protect their customers' information and transactions. These measures may include encryption, firewalls, fraud detection, and identity verification.

Support: If customers have questions or issues with their online banking account, they can typically contact customer support via phone, email, or chat.

Overall, an online banking system provides customers with convenient access to their account information and enables them to perform a range of transactions from the comfort of their own home or office.

Sample Code:

→To create a frame

```
homePage frame = new homePage();

        frame.setTitle("Login page");
        frame.setBackground(Color.CYAN);
        frame.setVisible(true);
        frame.setLocation(100, 100);
        frame.setSize(1200, 700);
        frame.setLayout(null);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setResizable(true);
        frame.setBackground(Color.LIGHT_GRAY);
```

→**AWT container:-**is a component that contain other AWT components.

```
Container container = getContentPane();
        container.setLayout(null);
```

method : -

getContentPane() :- retrieves the content pane layer so other component can added .

setLayout():- set the layout for container

→ **Label** :-A display area for short string or an image or both.

Example

```
JLabel header = new JLabel(i);
        header.setBounds(0, 0, 1200, 55);
        add(header);
```

→**ImageIcon** :-An implementation of the icon interface that paints from images .

Example:

```
ImageIcon i = new ImageIcon(getClass().getResource("menu.jpg"));
```

→**JPanel** :- is a generic lightweight container .

```
JPanel panel1 = new JPanel();
        panel1.setBackground(new Color(0, 100, 250, 100)); //method used
to set background for panel
        panel1.setLocation(0, 55); // method to set location
        panel1.setSize(1200, 40); //setsize of panel
        panel1.setLayout(null);     //setlayout
        panel1.setBorder(BorderFactory.createEtchedBorder(2,
Color.BLACK, Color.BLACK));  //setBorder
        add(panel1);
```

→**JTextField** : -to take user input

```
JTextField capchafield = new JTextField();
        capchafield.setBounds(100, 230, 200, 30);
        panel.add(capchafield);
```

→**addMouseListner**:-is a class that gets notified when there is a change in the mouse state. Changes of the mouse can be pressing clicking and releasing it.

```
f11.addMouseListener(new MouseAdapter() {
            @Override
            public void mouseClicked(MouseEvent e) {
                f11.setForeground(Color.RED);
            }
        });
```

→**addActionListner:-**is a class that gets notified when button is clicked.

```
clear.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                userfield.setText("");
                passfield.setText("");
            }
        });
```

→**JOptionPane**:- is a class library thatmakes it easy to pop up a simple dialog box that either provides as information message or asks for simple input from the user.

showMessageDialog –

```
JOptionPane.showMessageDialog(login, "Invalid Email Id ! ");
```

showConfirmDialog -

```
int ch = JOptionPane.showConfirmDialog(null, "Are you sure ?");
```

→**JButton** – Swing component to create button

```
JButton submit = new JButton("Submit");
        submit.setBounds(50, 180, 100, 30);
        panel.add(submit);
```

→**JCheckBox** – to create cheakbox

```
JCheckBox term = new JCheckBox("I Agree with Terms and Conditions");
        term.setBounds(100, 265, 260, 20);
        term.setFont(new Font("Times New Roman", Font.BOLD, 15));
        panel.add(term);
```

→**Mysql** use

```
try {
                    Connection con = (Connection) DriverManager
                            .getConnection("jdbc:mysql://localhost:3306/
login", "root", "ranjan5044");
                    String query = "Select pass from bank where
email=?";
                    PreparedStatement x = (PreparedStatement)
con.prepareStatement(query);
                    x.setString(1, email);
                    ResultSet rs = x.executeQuery();
                    while (rs.next()) {
                        String name = rs.getString("pass");
                        if (pass.compareTo(name) == 0) {
                            String stt = capchafield.getText();
                            if (stt.compareTo(st.toString()) == 0) {
                                if (term.isSelected()) {
                                    new Bankprocess();
                                } else {
                                    JOptionPane.showMessageDialog(term,
"Accept Terma and Condition");
```

```
                                }
                            } else {
                                JOptionPane.showMessageDialog(captcha,
"Captcha Not matching");
                            }
                        } else {
                            JOptionPane.showMessageDialog(panel,
"Username and password do not matched !");
                        }
                    }
                    con.close();
                } catch (Exception exception) {
                    System.out.println(exception);
                }
```

Outcome of the Project:

Here are some possible outcomes of an Online Banking System project using Java Gui:

Successful implementation of the project: If the development team follows best practices and implements the project according to the requirements, the outcome could be a fully functional online banking system that meets the client's expectations.

Security vulnerabilities: An online banking system holds sensitive financial information of users, making it a prime target for cyber attacks. If the development team doesn't implement appropriate security measures, the outcome could be security vulnerabilities that expose users' financial data.

Performance issues: An online banking system needs to handle a large number of concurrent users and transactions. If the system's performance isn't optimized, it could lead to slow response times, timeouts, or system crashes. User adoption: Even if the online banking system is technically sound, if the user interface is not user-friendly, users may not adopt it. The outcome in this case could be low adoption rates or dissatisfaction among users.

Maintenance and support: Once the online banking system is deployed, it needs to be maintained and supported. If the development team doesn't provide adequate maintenance and support, the outcome could be system downtime, bugs, and other issues that impact user experience.

Overall, the outcome of an Online Banking System project using Java depends on various factors. Proper planning, execution, and maintenance are crucial to ensure a successful project outcome.

Conclusion:

Online banking systems provide a convenient way for customers to access their bank accounts and perform transactions from anywhere and at any time. The use of Java programming language offers several advantages, such as platform independence, security, and scalability.

The project involved designing and implementing various modules, such as user authentication, account management, transaction processing, and reporting. In the authentication module, different methods like username and password, biometric authentication, and one-time passwords could be implemented to enhance the system's security. The account management module allows customers to view their account balances, transaction history, and manage their personal information like contact details, address, and other related details.

The transaction processing module allows customers to perform different types of transactions such as funds transfer, bill payments, and other banking transactions. The reporting module generates various reports such as transaction history, account statements, and other financial reports that are useful for both customers and bank

administrators. To ensure the system's security, various security measures such as data encryption, firewalls, and intrusion detection systems could be implemented. Testing and quality assurance are essential parts of software development, and various types of testing like unit testing, integration testing, and system testing should be performed to ensure the system's quality.

Overall, the Online Banking System project using Java provides a useful solution for banks and their customers to streamline banking operations and improve customer service.