Project

Course Code / Title: SWE 2111
Assessment: Project
Semester / Year: 2025
Division: 4691

STUDENT IDENTIFICATION:

Name: Abdulrahman Almuaythir	**ID: 445050211**
Name: Badr Al-Qahtani	**ID: 445050224**

Online banking System

1. Introduction

The Online Banking System is a web-based platform that facilitates customers in performing financial operations, including account balance inquiry, fund transfer, and transaction management in a secure and efficient manner.

It should be designed to provide users with convenience, reliability, and high data protection while simultaneously supporting bank staff with an effective tool to manage accounts and operations.

This will ensure full compliance with financial regulations, and it is intended to be easy maintenance and updates for the future to guarantee stability and performance.

1.1 Problem Statement

The core problem addressed by this project is the growing demand for secure, immediate, and comprehensive access to financial services that obviates the need for physical branch visits.

Traditional banking models impose significant friction, consuming customer time and increasing operational costs for the institution.

By developing a robust Online Banking System, the project seeks to resolve these inefficiencies, delivering 24/7 self-service capabilities while strictly adhering to high standards of security and transactional integrity."

1.2 Background Survey

Existing Solutions and Their Limitations

1.2.1 Traditional Banking (Manual/Legacy Systems): This fundamental approach requires in person visits or paper-based processes for nearly all key activities. While historically reliable, this method fundamentally lacks the speed, convenience, and 24/7 availability demanded by modern users, creating high operational friction for both the bank and the customer.

1.2.2 Competitor Online Banking Platforms: The direct competition consists of existing online and mobile services provided by major financial institutions.

These platforms typically cover core functionality such as User Login and Viewing Account Balances. However, many of these systems often prioritize rapid feature delivery over the deep integration of complexity, frequently resulting in potential gaps related to consistent security, regulatory compliance, and robust internal control.

1.2.3 The Differentiator: How Our Solution Stands Apart

Our proposed Online Banking System is specifically designed to address the weaknesses of existing solutions by focusing on integrating complex controls and oversight throughout the platform, rather than merely replicating basic features.

1.3 Proposed Approach

The proposed solution is the development of a secure, feature-rich Online Banking System built upon a structured, phase-based methodology.

System Goal:

To deliver all necessary banking functionality through a unified web interface, ensuring high security and data integrity.

Technical Solution Overview:

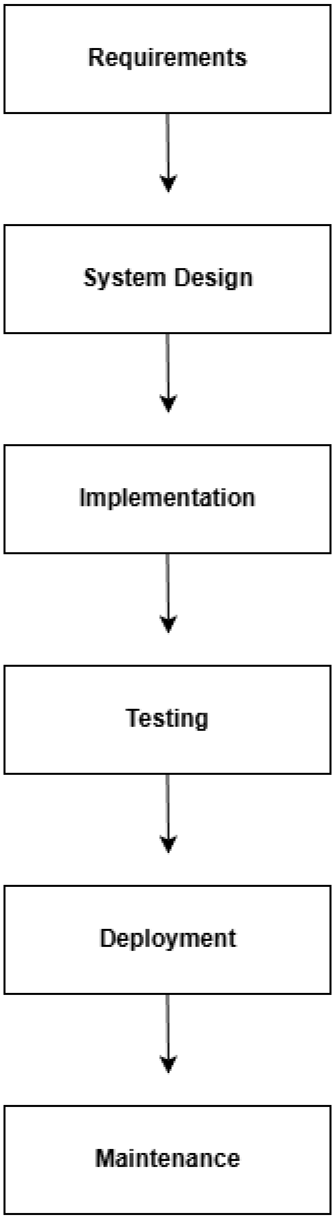
1. **System Structure:** The system will utilize a layered architecture: Presentation Layer (Frontend), Business Logic Layer (Backend), and Data Layer (Database/Storage). (This structure will be defined in your UML Class Diagram).

2. **Core Functionality:** Key transactions (UC4, UC2, UC13) will be implemented with integrated security checks, including funds verification.

3. **Security and Oversight:** Access requires validated credentials (UC1). High risk transactions are routed for approval (UC6), and all system actions are logged for compliance (UC9).

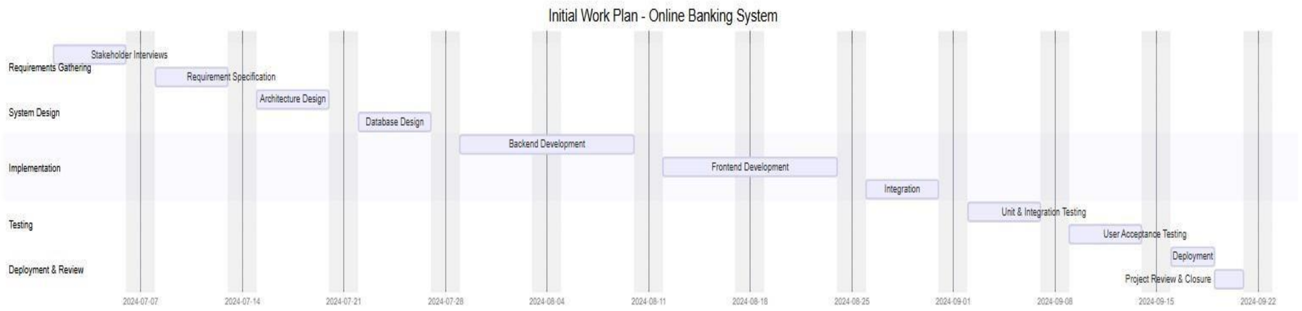
4. **Data Integrity:** Critical processes like Funds Transfer will utilize atomic transactions (R13) to guarantee ACID properties, ensuring that data is always consistent and reliable.

1.4 Software process model



This is the Waterfall system.

1.5 Initial Work Plan



2. Functional Requirements

2.1 User Authentication: The system should authenticate users through login and password.
2.2 Account Management: Users should be able to view their account balance, transaction history, and account details.
2.3 Funds Transfer: Users should be able to transfer funds between their own accounts or to third-party accounts.
2.4 Bill Payment: The system should allow users to pay bills (e.g., utilities, credit card payments).
2.5 Transaction Notifications: The system should send notifications for transactions (via email/SMS).
2.6 Security Features: The system should implement two-factor authentication for security during sensitive operations.
2.7 Transaction Search: Users should be able to search for past transactions based on different filters (date, amount, etc.).
2.8 Account Statement Generation: Users should be able to download or view monthly/yearly account statements in PDF format.

ID	Priority	Description
R1. User Authentication	High	The system should authenticate users through login and passwords.
R2. Account Management	Intermediate	Users should be able to view their account balance, transaction history, and account details.
R3. Funds Transfer	High	Users should be able to transfer funds between their own accounts or third-party accounts.
R4. Bill Payment	High	The system should allow users to pay bills (e.g., utilities, credit card payments).
R5. Transaction Notifications	Low	The system should send notifications for transactions (via email/SMS)
R6. Security Features	High	The system should implement two-factor authentication for security during sensitive operations.
R7. Transaction Search	Intermediate	Users should be able to search for past transactions based on different filters (date, amount, etc.).
R8. Account Statement Generation	Low	Users should be able to download or view monthly/yearly account statements in PDF format.

This table lists the key system requirements.

3. Non-Functional Requirements

3.1 Performance: The system should respond to user actions within 2 seconds for most operations.

3.2 Usability: The interface should be user-friendly and easy to navigate for both technical and non-technical users.

3.3 Reliability: The system should be highly available with minimal downtime.

3.4 Security: All sensitive data (e.g., passwords, transaction details) should be encrypted both at rest and in trans

ID	**Priority**	**Description**
R1. Performance	High	The system should respond to user actions within 2 seconds for most operations .
R2. Usability	High	The interface should be user-friendly and easy to navigate for both technical and non-technical users.
R3. Reliability	High	The system should be highly available with minimal downtime
R4. Security	High	All sensitive data (e.g., passwords, transaction details) should be encrypted both at rest and in transit

4. Use Case Modelling:

4.1 Log In (UC1)

Use Case Name: Log In

Actor: Customer, Bank Employee

Difficulty: Easy

Pre-condition: The actor must be registered in the system.

Post-condition: The actor is logged into the system.

Processes: The actor enters a username and password. The system verifies credentials. If valid, access is granted; otherwise, an error message is displayed.

4.2 View Balance (UC2)

Use Case Name: View Balance

Actor: Customer

Difficulty: Easy

Pre-condition: The customer must be logged in.

Post-condition: The account balance is shown.

Processes: The customer selects the "View Balance" option.
The system retrieves the current account balance. The balance is displayed on the screen.

4.3 Reset Password (UC3)

Use Case Name: Reset Password

Actor: Customer, IT Support

Difficulty: Intermediate

Pre-condition: The actor must provide account verification details.

Post-condition: The password is updated.

Processes: The actor requests a password reset. The system verifies identity. A new password is generated or set by the actor. A confirmation message is displayed.

4.4 Transfer Funds (UC4)

Use Case Name: Transfer Funds

Actor: Customer

Difficulty: Intermediate

Pre-condition: The customer must be logged in and have sufficient funds.

Post-condition: The amount is deducted from the sender and credited to the recipient.

Processes: The customer enters recipient details and amount. The system verifies account details and balance. If valid, the system processes the transfer. A confirmation message is displayed.

4.5 Manage Accounts (UC5)

Use Case Name: Manage Accounts

Actor: Bank Employee

Difficulty: Intermediate

Pre-condition: The actor must have administrative access.

Post-condition: The account details are updated.

Processes: The actor selects an account to manage. The system allows updates such as personal info or account status. Changes are saved.

4.6 Approve Transactions (UC6)

Use Case Name: Approve Transactions

Actor: Bank Employee, Regulatory Authority

Difficulty: Hard

Pre-condition: The transaction must be flagged for approval.

Post-condition: The transaction is either processed or denied.

Processes: The system notifies the actor about a pending transaction. The actor reviews transaction details. The actor approves or rejects the transaction. The system updates the transaction status.

4.7 Assist Customers (UC7)

Use Case Name: Assist Customers

Actor: Bank Employee

Difficulty: Intermediate

Pre-condition: The customer must have an issue or request.

Post-condition: The customer receives assistance.

Processes: The customer contacts the bank employee. The employee addresses the request or escalates the issue. The system logs the interaction.

4.8 Perform Testing (UC8)

Use Case Name: Perform Testing

Actor: Developer, QA Engineer

Difficulty: Hard

Pre-condition: A feature or system component must be ready for testing.

Post-condition: The features are tested, and issues are logged.

Processes: The developer runs unit tests. The QA engineer performs system-level tests. Any issues or bugs are reported for fixing.

4.9 Audit Compliance (UC9)

Use Case Name: Audit Compliance

Actor: Regulatory Authority

Difficulty: Hard

Pre-condition: The system must be operational.

Post-condition: The system remains legally compliant.

Processes: The system logs compliance-related activities. The regulatory authority reviews records. Necessary actions are taken if violations are found.

4.10 Monitor System (UC10)

Use Case Name: Monitor System

Actor: IT Support, Regulatory Authority

Difficulty: Hard

Pre-condition: The system must be running.

Post-condition: The system remains operational and compliant.

Processes: The system generates performance and security logs. IT supports reviews logs for technical issues. Regulatory authorities review logs for compliance violations.

4.11 Develop Features (UC11)

Use Case Name: Develop Features

Actor: Developer

Difficulty: Hard

Pre-condition: The system must be in a state where new features can be added.

Post-condition: The new feature is added to the system and available for use.

Processes: The project manager reviews the stakeholder requirements for new features. The developer designs and implements the new feature. The feature is integrated into the system and tested. The project manager approves the feature for release.

4.12 Fix Bugs (UC12)

Use Case Name: Fix Bugs

Actor: Developer

Difficulty: Hard

Pre-condition: The system must have bugs that need fixing (reported by users or discovered during testing).

Post-condition: The bugs are fixed, and the system is stable.

Processes: Bugs are reported by QA engineers or users. Developers analyze the issues and fix the bugs. The fixes are tested by QA engineers to ensure the issue is resolved. Once verified, the system is updated with the fixes.

4.13 Account Statement Generation (UC13) **Use Case Name:** Account Statement Generation

Actor: Customer

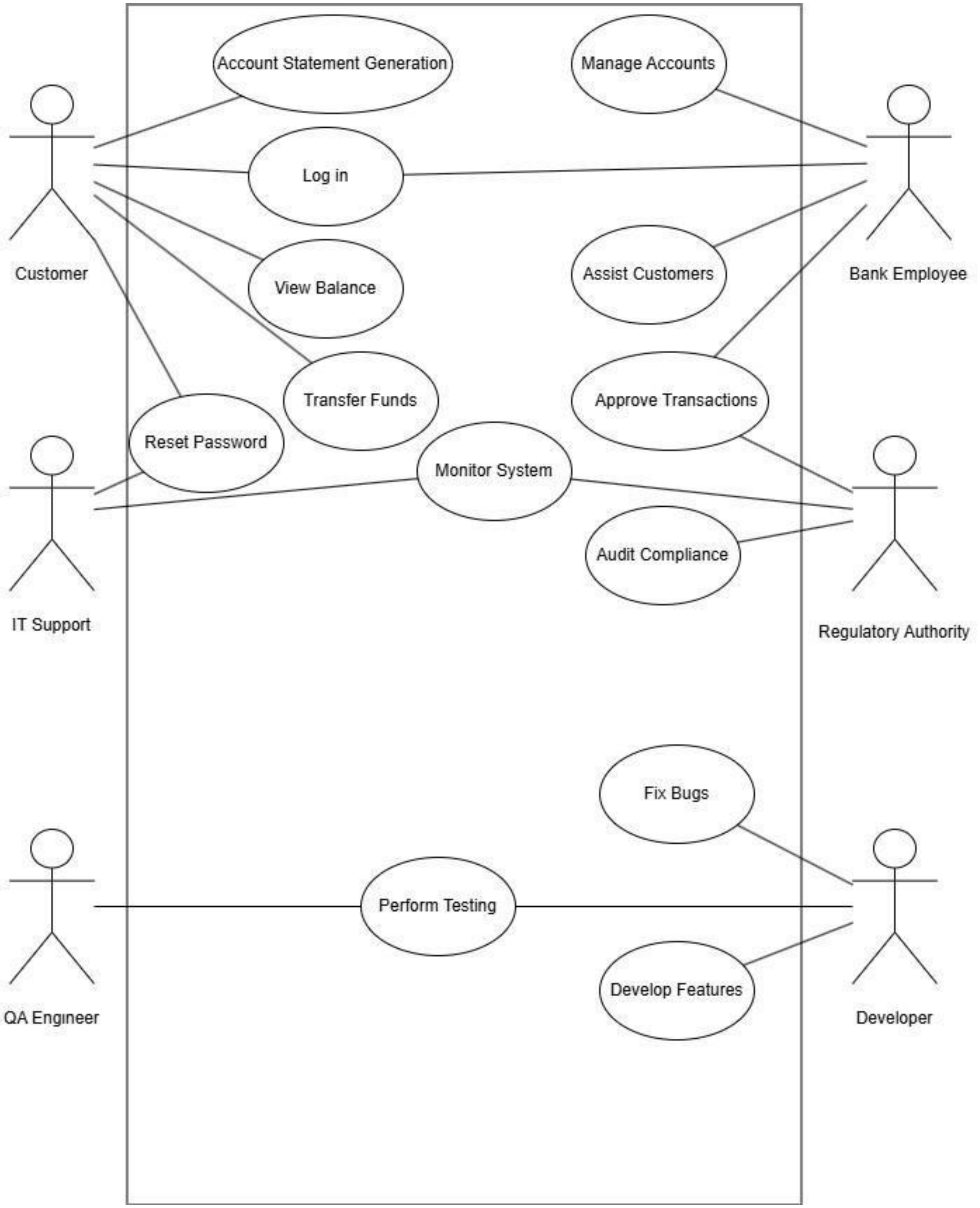
Difficulty: Intermediate

Pre-condition: The Customer must be successfully logged in to the system. The system must have a transaction history recorded for the requested period.

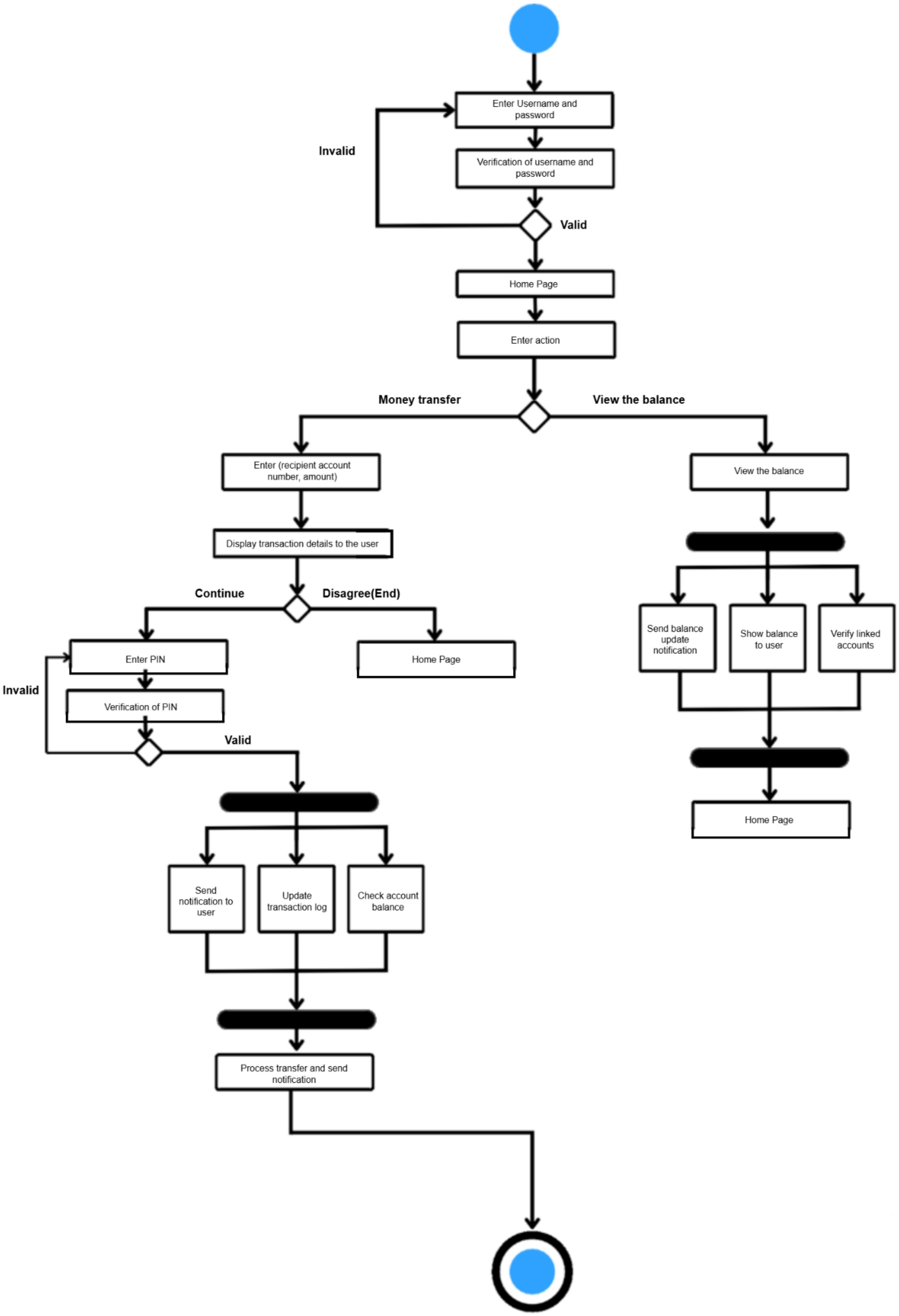
Post-condition: A structured account statement is generated and presented to the Customer (as a viewable or downloadable PDF).

Processes: The Customer selects the "Generate Statement" option. The Customer selects the start and end dates. The system retrieves all transactions for the period. The system formats the data into the mandatory PDF template. The system delivers the PDF statement to the Customer.

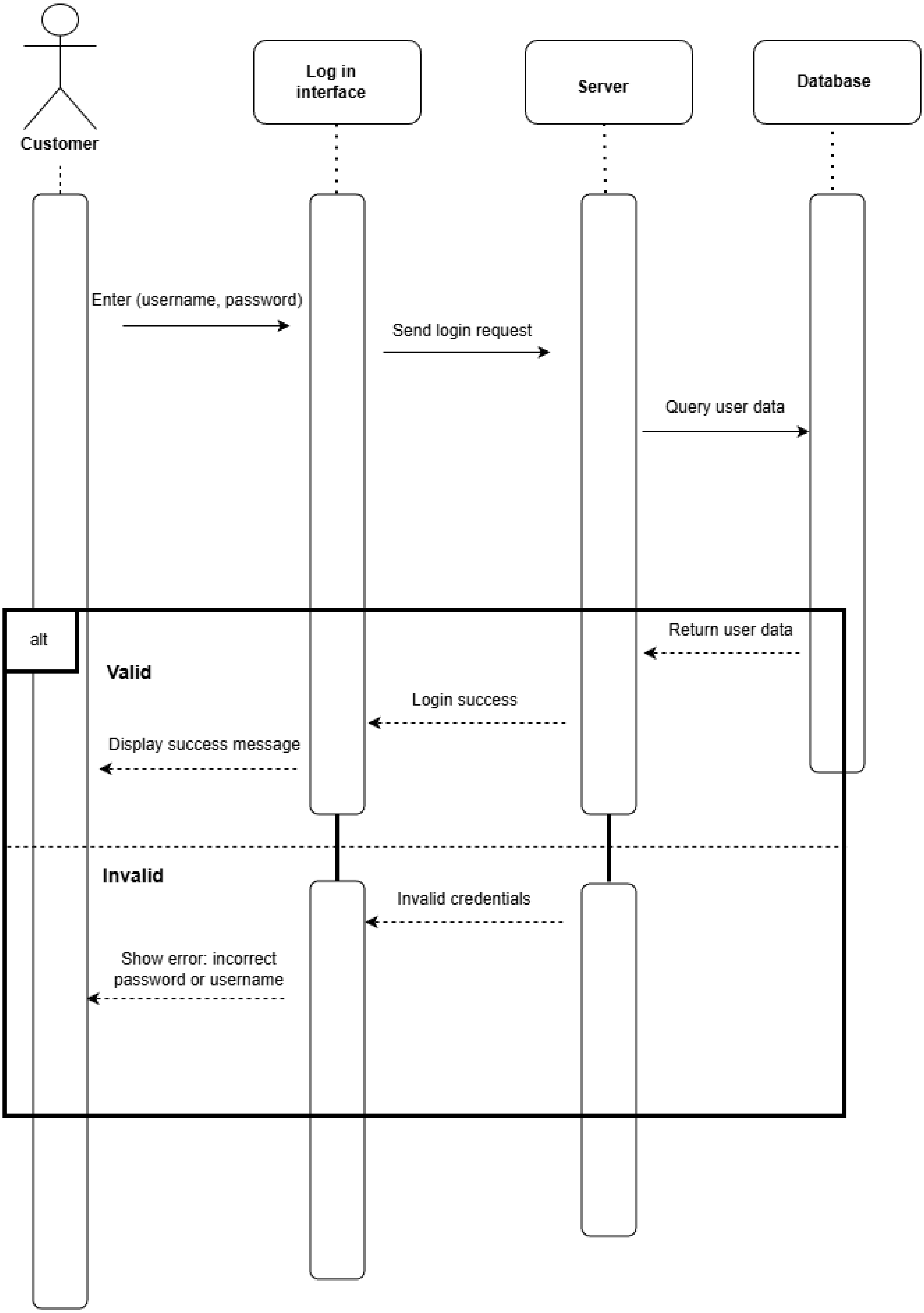
5. Use case diagram



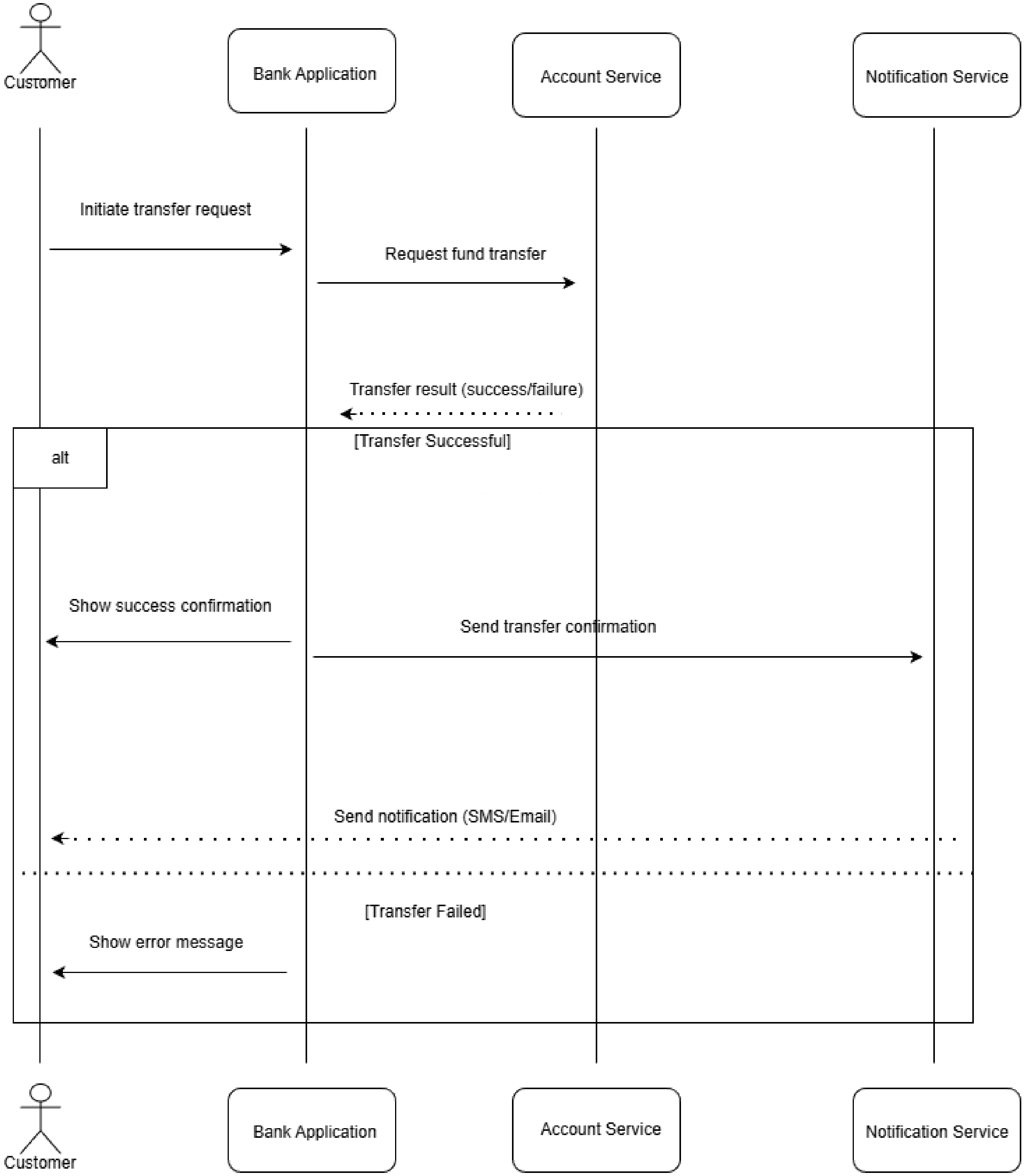
6. Process Model (Activity Diagram):



7.Sequence Diagram for (Log in)



7.1Sequence Diagram for (Transfer Funds)



8. Creating a Class Diagram:

1. **User:** Represents the customer in the banking system.
   Attributes: username, password, email, accountNumber
   Methods: login(), logout(), changePassword()

2. **Account:** Represents the user's account.
   Attributes: accountNumber, balance, accountType
   Methods: viewBalance(), deposit(), withdraw(), transfer()

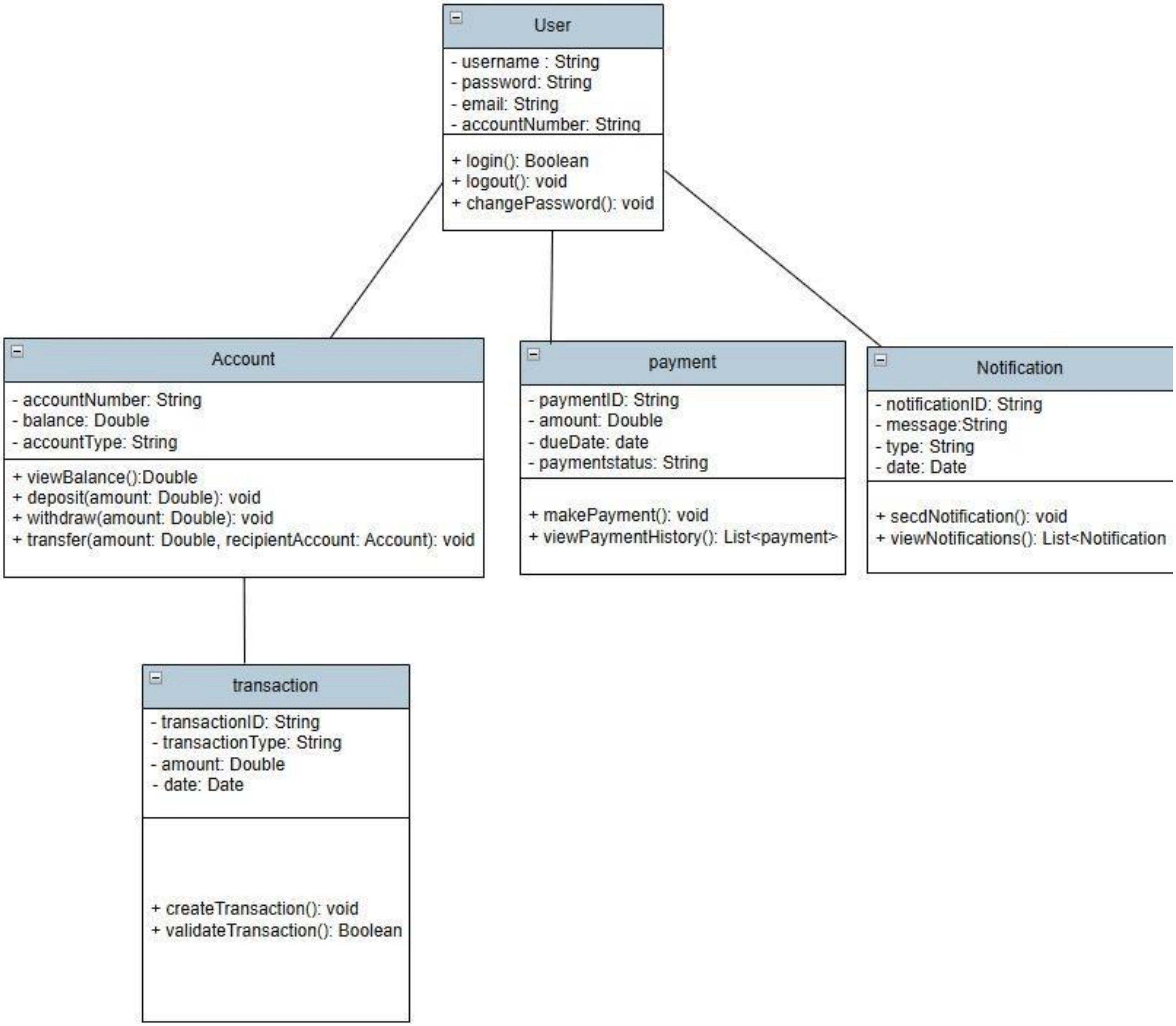
3. **Transaction:** Represents a transaction that occurs in the system. **Attributes:** transactionID, transactionType, amount, date
   Methods: createTransaction(), validateTransaction()

4. **Payment:** Represents a bill payment or a similar financial operation.
   Attributes: paymentID, amount, dueDate, paymentStatus
   Methods: makePayment(), viewPaymentHistory()

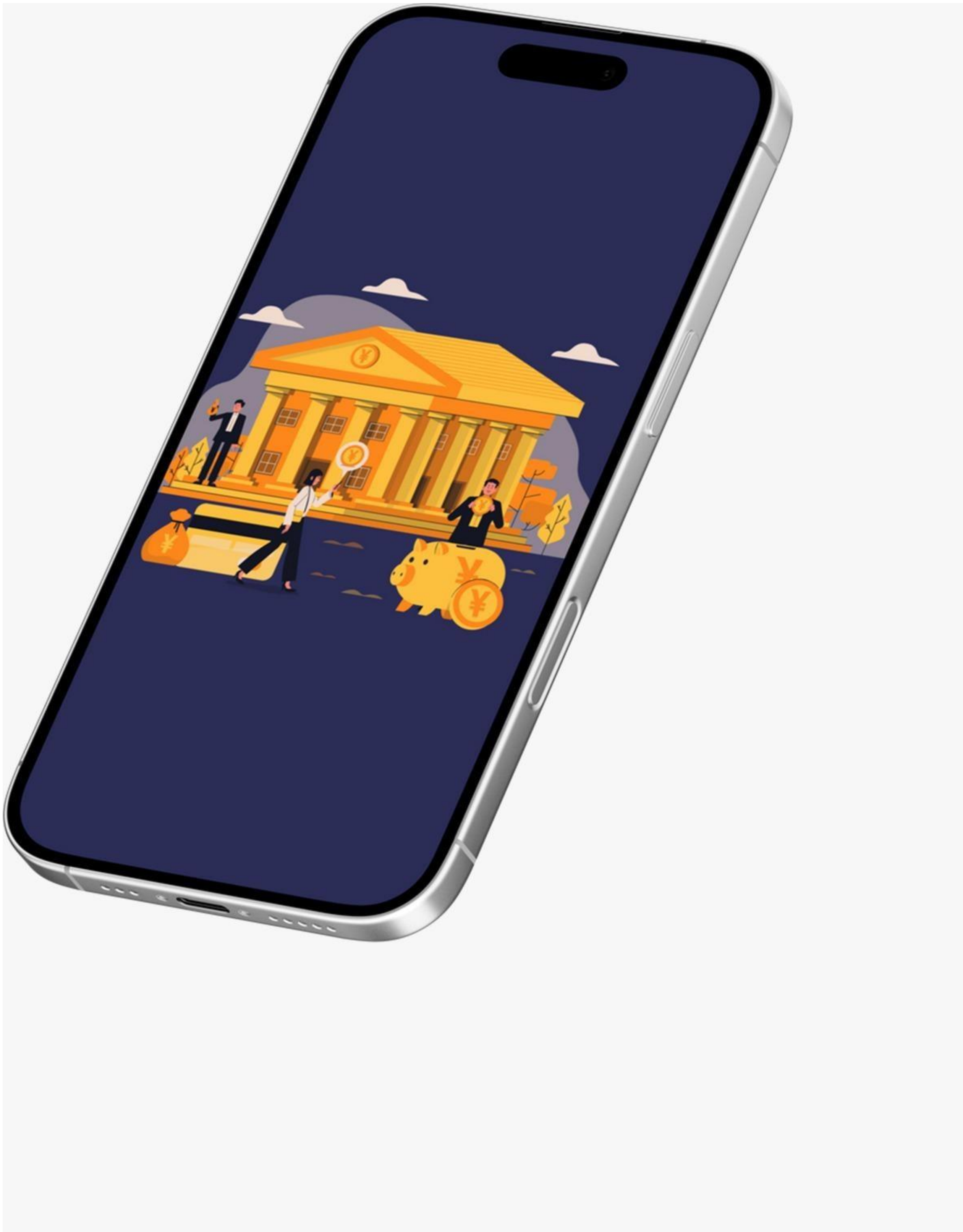
5. **Notification:** Represents the notifications the system sends to users.
   Attributes: notificationID, message, type, date

   Methods: sendNotification(), viewNotifications()This diagram would show the relationships between the classes, such as:
   User has one or more Accounts.
   Account can have many Transactions.
   User can have multiple Notifications.

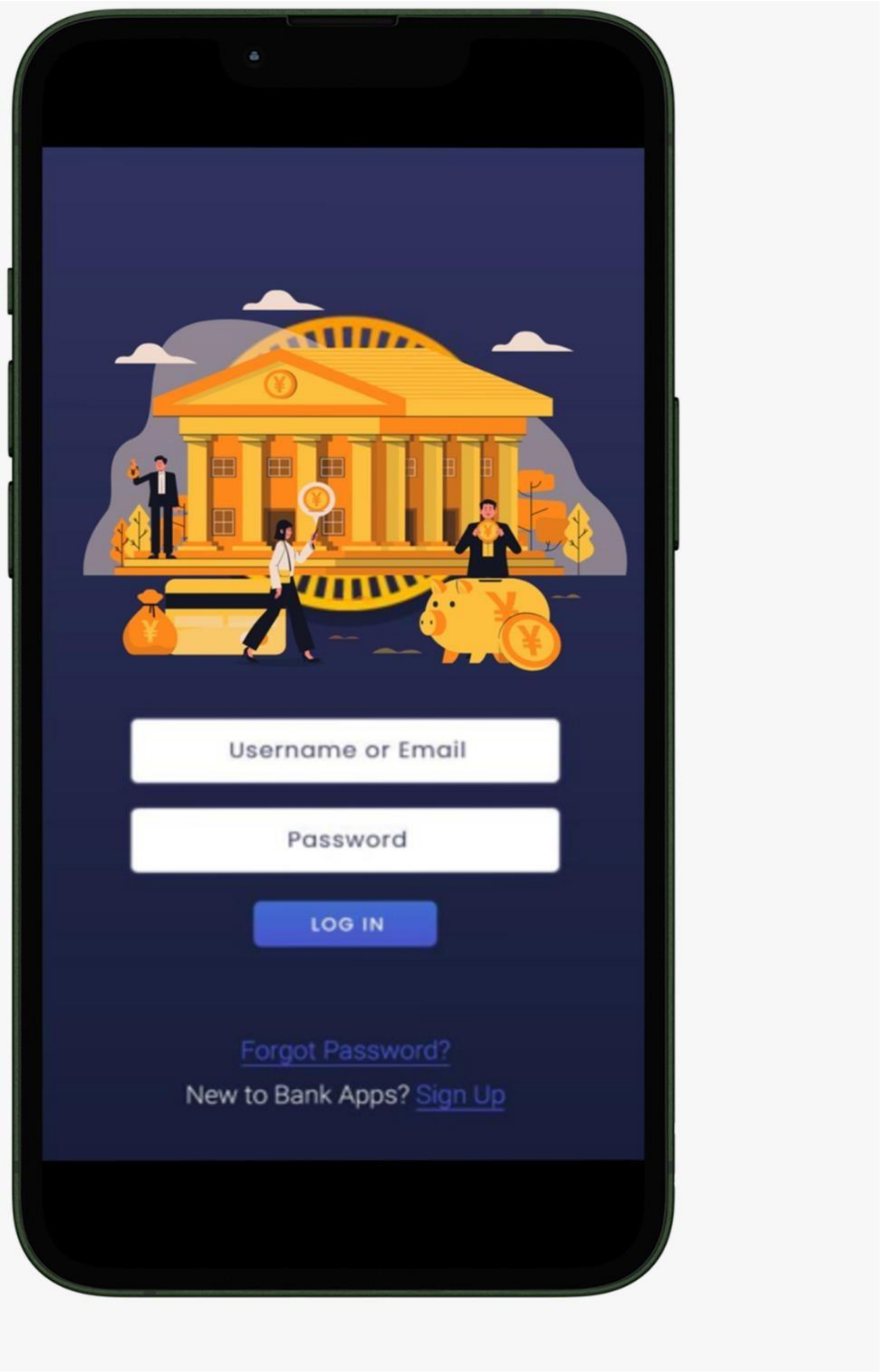
8.1 Class Diagram



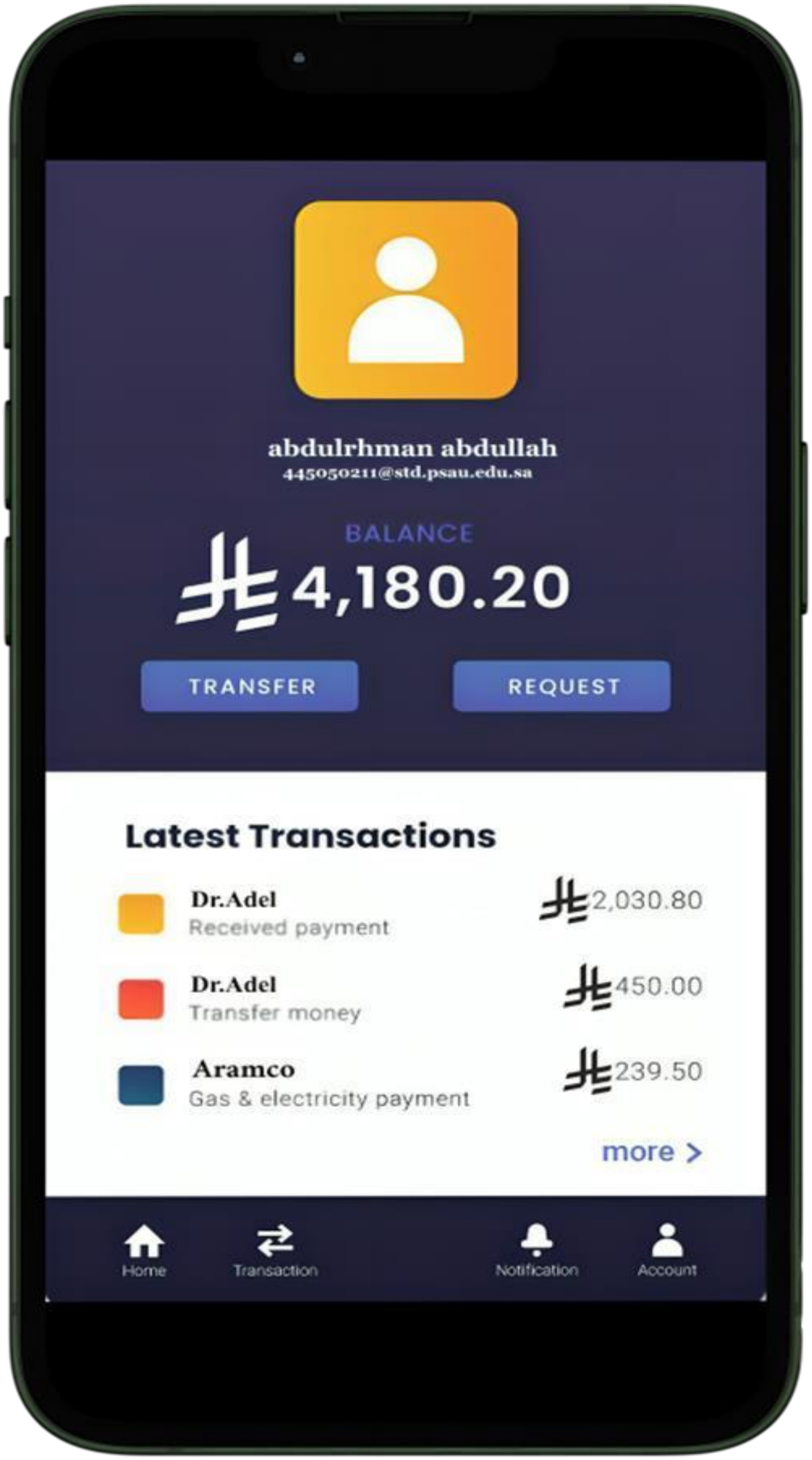
Introduction to the App:



Log in Screen:



View balance Screen:



abdulrhman abdullah

445050211@std.psau.edu.sa

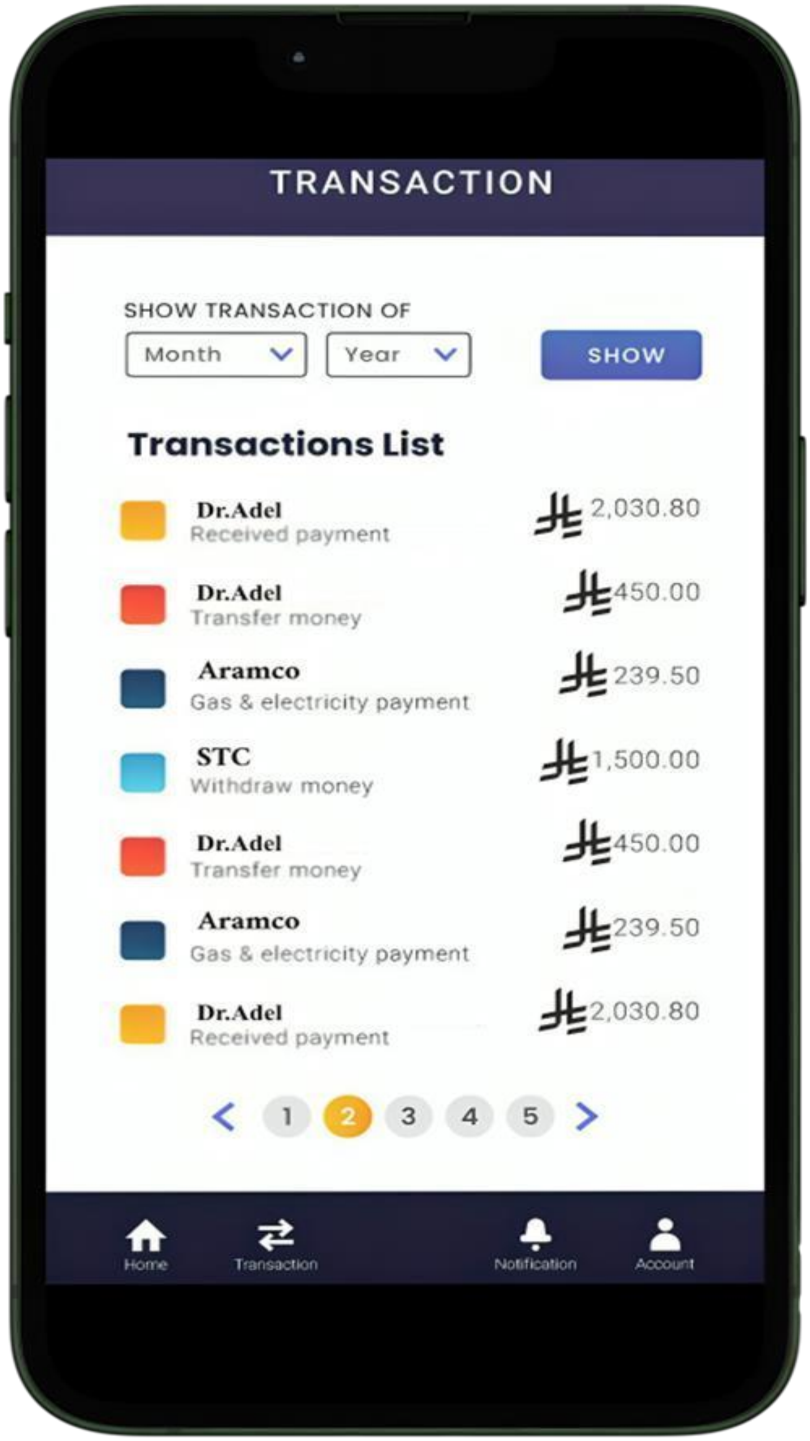
BALANCE

⃁4,180.20

Latest Transactions

🟧	Dr.Adel Received payment	⃁2,030.80
🟥	Dr.Adel Transfer money	⃁450.00
🟦	Aramco Gas & electricity payment	⃁239.50

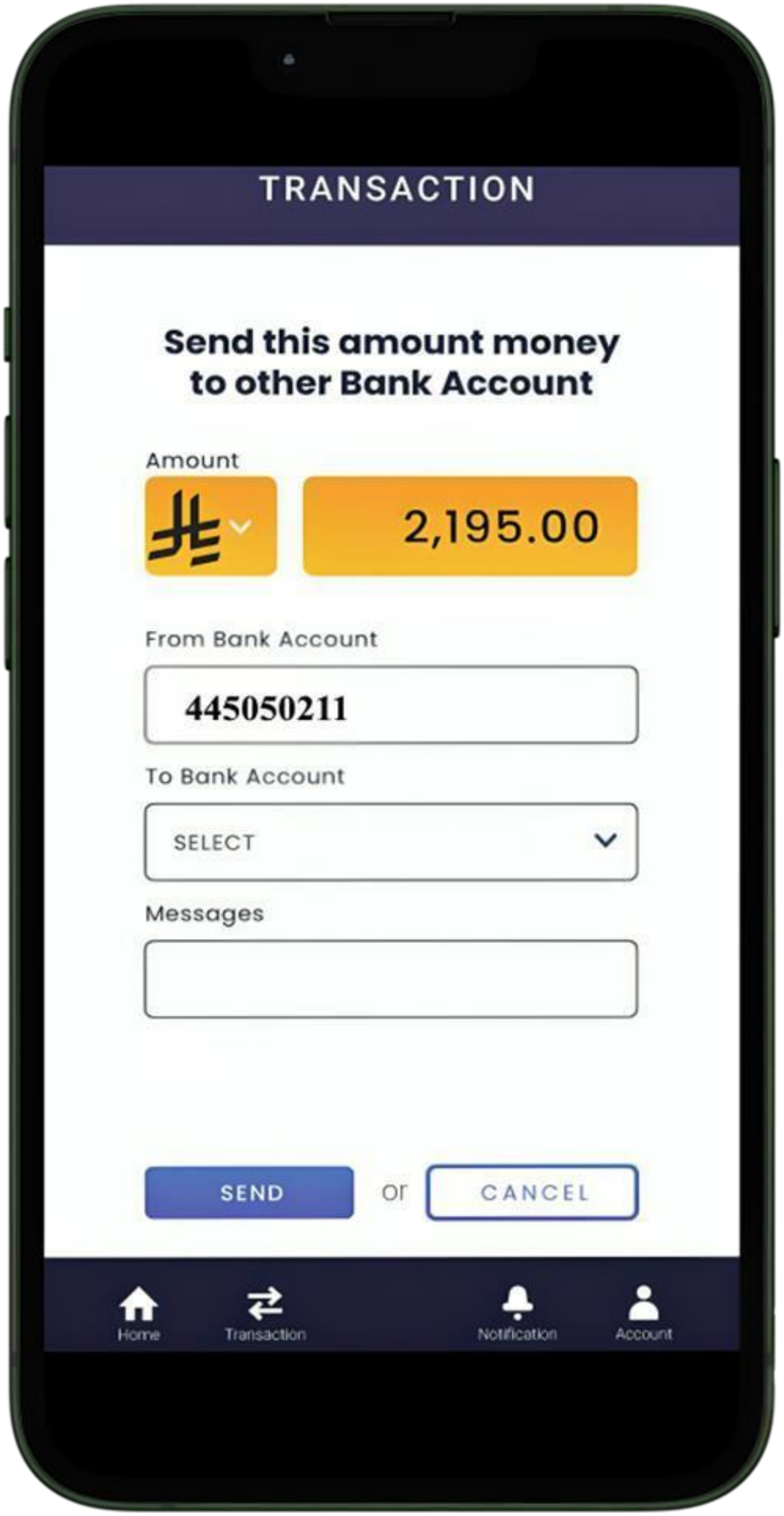
Transaction Screen:



Transactions List

Dr.Adel Received payment	⃁ 2,030.80
Dr.Adel Transfer money	⃁450.00
Aramco Gas & electricity payment	⃁239.50
STC Withdraw money	⃁1,500.00
Dr.Adel Transfer money	⃁450.00
Aramco Gas & electricity payment	⃁239.50
Dr.Adel Received payment	⃁2,030.80

Transaction Screen:



TRANSACTION

Send this amount money to other Bank Account

Amount

2,195.00

From Bank Account

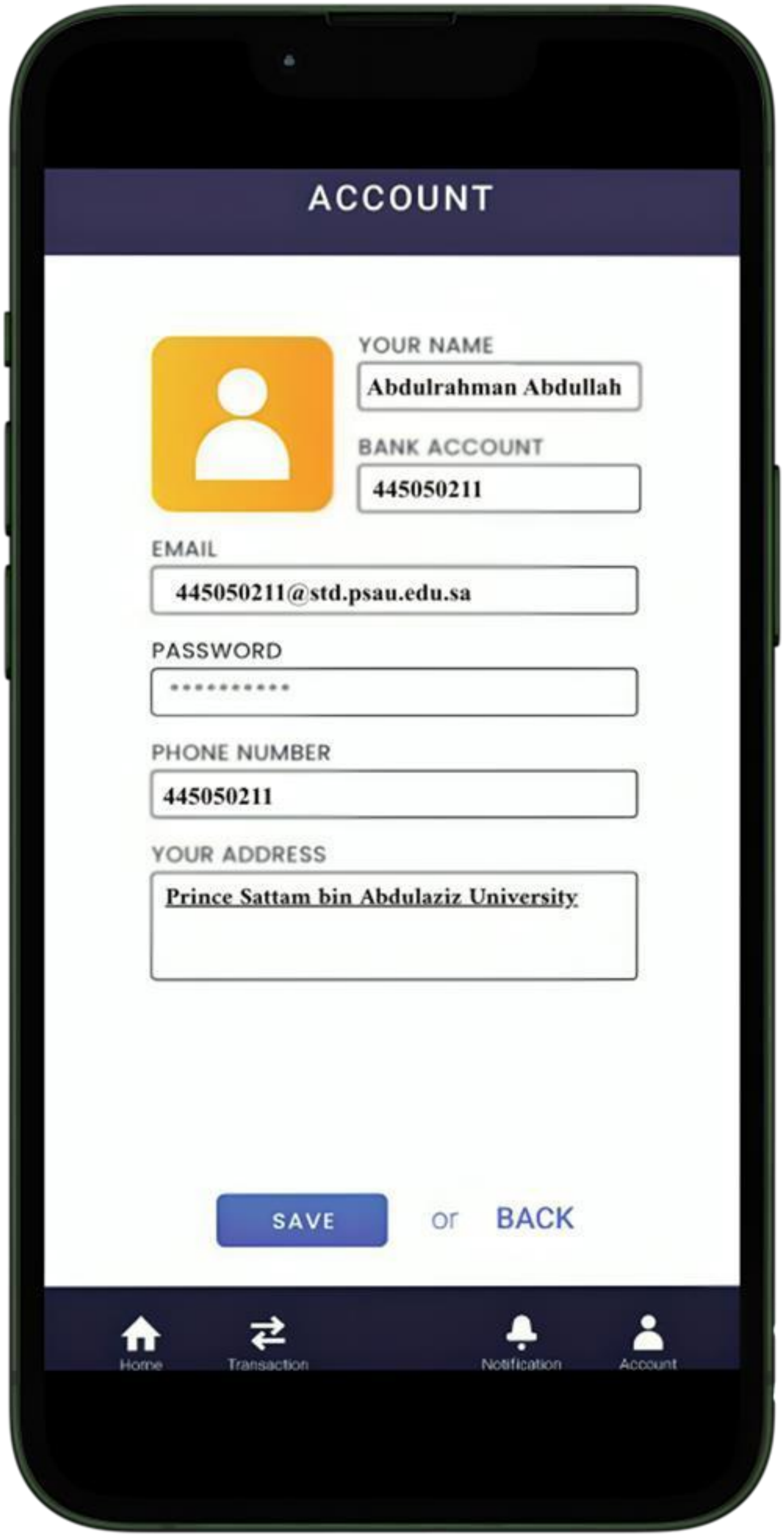
445050211

To Bank Account

SELECT

Messages

Account Screen :



ACCOUNT

YOUR NAME

Abdulrahman Abdullah

BANK ACCOUNT

445050211

EMAIL

445050211@std.psau.edu.sa

PASSWORD

****

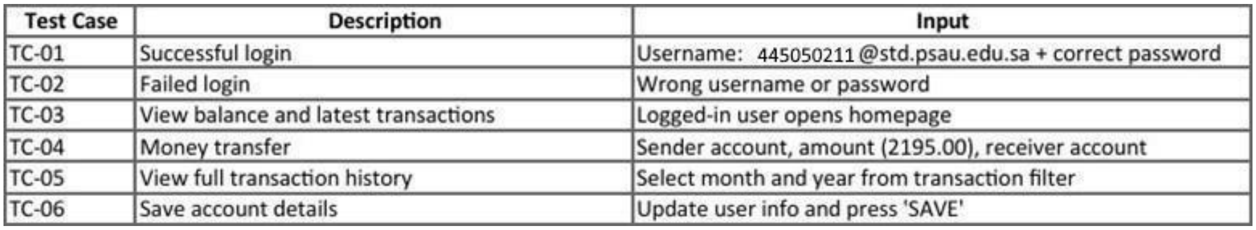
PHONE NUMBER

445050211

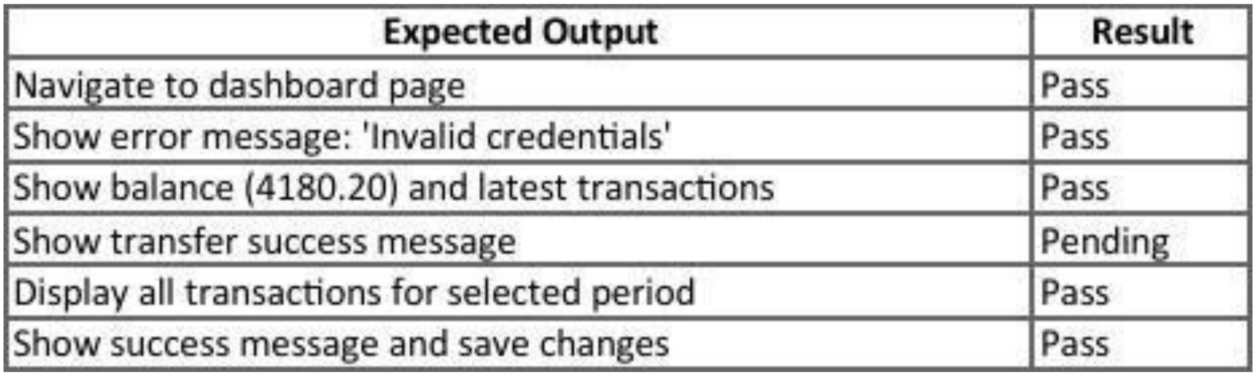
YOUR ADDRESS

Prince Sattam bin Abdulaziz University

Test Cases:



Test Case	Description	Input
TC-01	Successful login	Username: 445050211 @std.psau.edu.sa + correct password
TC-02	Failed login	Wrong username or password
TC-03	View balance and latest transactions	Logged-in user opens homepage
TC-04	Money transfer	Sender account, amount (2195.00), receiver account
TC-05	View full transaction history	Select month and year from transaction filter
TC-06	Save account details	Update user info and press 'SAVE'



Expected Output	Result
Navigate to dashboard page	Pass
Show error message: 'Invalid credentials'	Pass
Show balance (4180.20) and latest transactions	Pass
Show transfer success message	Pending
Display all transactions for selected period	Pass
Show success message and save changes	Pass