

ABSTRACT 4

APPROACH 4

Objectives 5

User Views 5

Platform 5

Operating System: 5

Technologies Used: 5

Software Requirements: 5

SOFTWARE REQUIRMENT SPECIFICATION 6

Introduction 6

1.1 Purpose: 6

1.2 Scope : 6

2.1 Product Functions : 7

2.2 USE CASE MODEL : 8

2.3 ACTIVITY DIAGRAM : 9

DESIGN PHASE 10

Introduction 10

1.2 Overall System Design Objectives 10

1.3 Project Structure 10

USER INTERFACE 11

User Registration 11

FUTURE SCOPE OF THE PROJECT 15

[Online Banking]

[This is system is built to manage the details of customer accounts, transactions, balance, statement. It provides a simpler alternative to conventional banking system as the customer is not required to be at the banking to complete any transaction.]

[**F r a n c k D a v y T c h i e n k o u a**]

Wenzhou University

Online Banking System

Major: Computer Science & Artificial Intelligence

Name: Franck Davy Tchienkoua

Student Number: 19511160012

Due Date: 28/10/2021

ABSTRACT

This project “**Online Banking System**” provides us a simple interface for maintenance of a customer’s account at a bank. It can be used by bank institutions to provide a convenient channel for their customers to complete simple transactions.

Online Banking is one of the most important financial activities which will be carried out by any person who holds a bank account. There are various activities that can be carried out once you log in to your bank account. Once a user logs in he or she can check the bank balance, check bank account transaction history or account summary, transfer funds to another account. Whenever we deal with a banking system main concern should be the security related to banking transactions and account login activity.

APPROACH

Throughout the project the focus has been on presenting information in an easy and intelligible manner. The project could be of help for those who want to know about the fundamentals of online banking systems and want to develop software based on the same concept.

The proposed project uses a MVC (Model - View - Controller) approach to build the system. The use of MVC architecture is preferred for small to large scale projects. It effectively separate the GUI from the back-end and allow for easy maintenance of the project. With this approach we make sure that our code is as DRY (Don’t Repeat Yourself) as possible. Errors are less likely to occur and can be dealt with without affecting the entire project.

Objectives

1. Online opening of accounts
2. Online banking
3. Money Deposit
4. Money Withdrawal
5. Money Transfer
6. Transactions History
7. Balance Checking
8. Cash-In & Cash-Out

User Views

- ❖ Customers

Platform

Operating System:

- ❖ Mac-OS

Technologies Used:

- ❖ JavaFx
- ❖ FXML
- ❖ PostgreSQL
- ❖ CSS

Software Requirements:

- ❖ OpenJDK-16.0.2
- ❖ FXML-16 (Scene Builder)
- ❖ JavaFX-SDK-16
- ❖ PostgreSQL Server
- ❖ PostgreSQL JDBC 42.2.22

SOFTWARE REQUIREMENT SPECIFICATION

Introduction

1.1 Purpose:

The objective of the **Online Banking System** is to allow customers of any bank providing the system to be able to complete basic transactions that would normally require them to physically be at the bank. It'll also facilitate keeping all records of deposits, withdrawals and transfers and add extra features like profile update. In this manner, all transactions information can be available in a few seconds.

Overall, it'll make online banking system an easier job for the administrator and the customer of any bank. The system can be extended to include other more serious aspects of banking. Due to a change of project and time constraint, the system only priorities the basic features of an online banking system. We will later on work on the administrator aspect of the software and add more features for customers such as adding additional cards, changing profile picture, request for bank statement, etc.

The main purpose of this document is to illustrate the requirements of the project **Online Banking System** and is intended to help any organization to maintain and manage its customers' transactions.

1.2 Scope :

Without a **Online Banking System**, banks are limited to traditional banking methods (in-person), and with that comes some quirks.

The Online Banking system will store all the details of the customers including their personal information, accounts details, and transactions.

Register module: Register module will help in creating new accounts with the bank. This module will simplify the task of on paper registration. After a successful registration, the customer is given an account number and

customer id among other details to uniquely identify them in the system. They can also update basic information.

Login module: Login module will help in authentication of user accounts. Users who have valid login username and password can log into their respective accounts.

Account module: Account module provides the user with a glimpse of transactions they have made over time and history of what's been done in the system. They can quickly look check their balance and account number.

Deposits module: Deposits module helps with giving the user a way to deposit money into their account. It checks if the user has entered a valid amount. If they have, the transaction is successful and their balance is automatically updated along with how much they have deposited over time.

Withdrawals module: Withdrawals module helps with giving the user a way to withdraw money from their account. If the user has sufficient funds, the transaction is successful and their balance is automatically updated along with how much they have withdrawn over time.

Transfers module: Transfers module will the user transfer money from one account to another. It will check for account and amount validity.

Transactions module: Suppose there are hundreds of thousands of transactions and the user is looking for one in particular or transactions at a given time period. In a manual system this is a tedious task, but using this module we can easily search the transaction by specifying the transaction type, id, amount, date, etc in the search criteria.

Settings module: Settings module will help the customer change his username or password. Maybe they have been compromised or are not strong enough.

Logout module: Logout module will help the customer sign out of the system. The customer could be using the system on someone's laptop for example and will be bad if they didn't have a way to logout once they are done using the system.

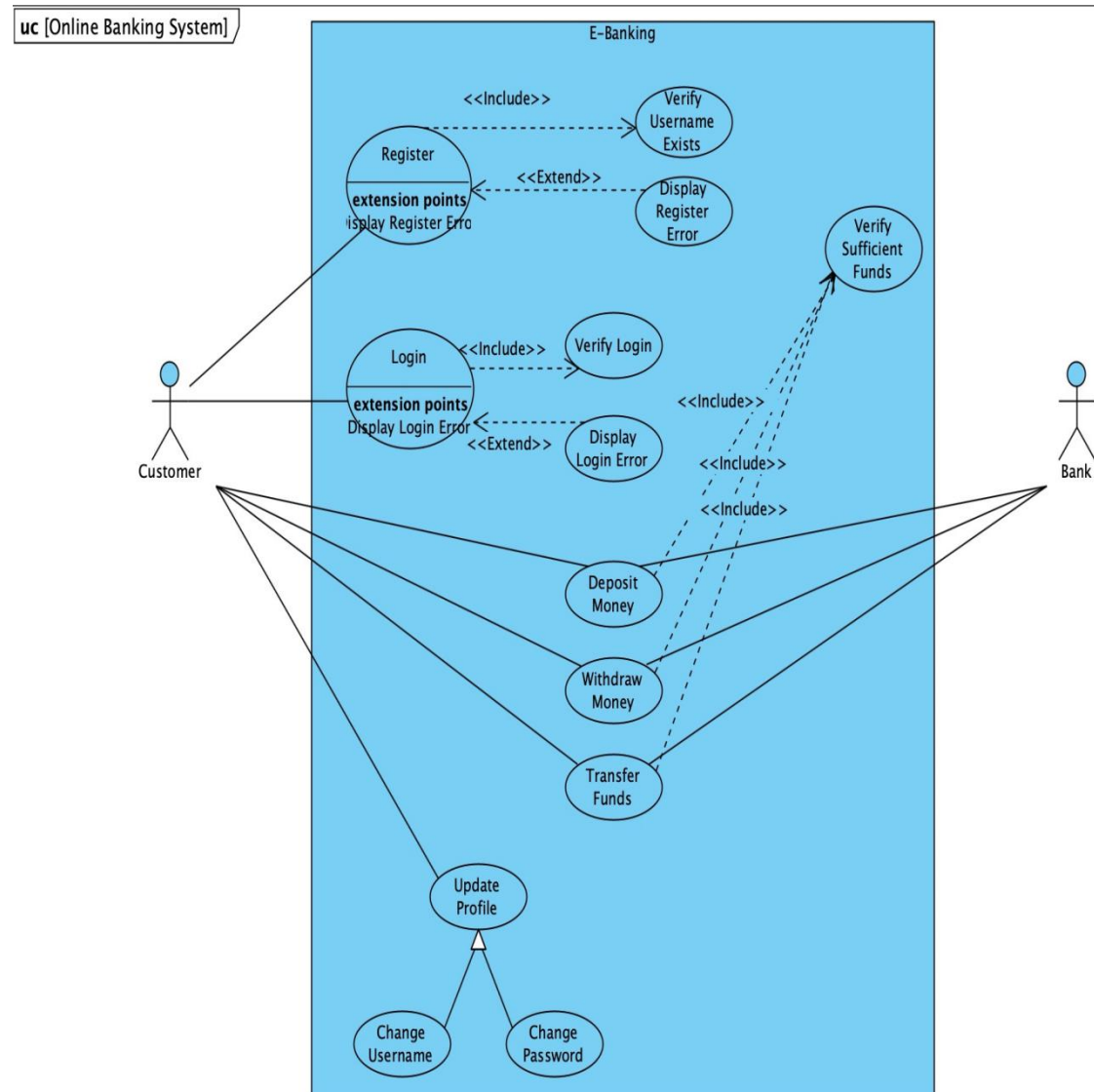
2.1 Product Functions :

There is only one user who will be using this product: A bank's customer.

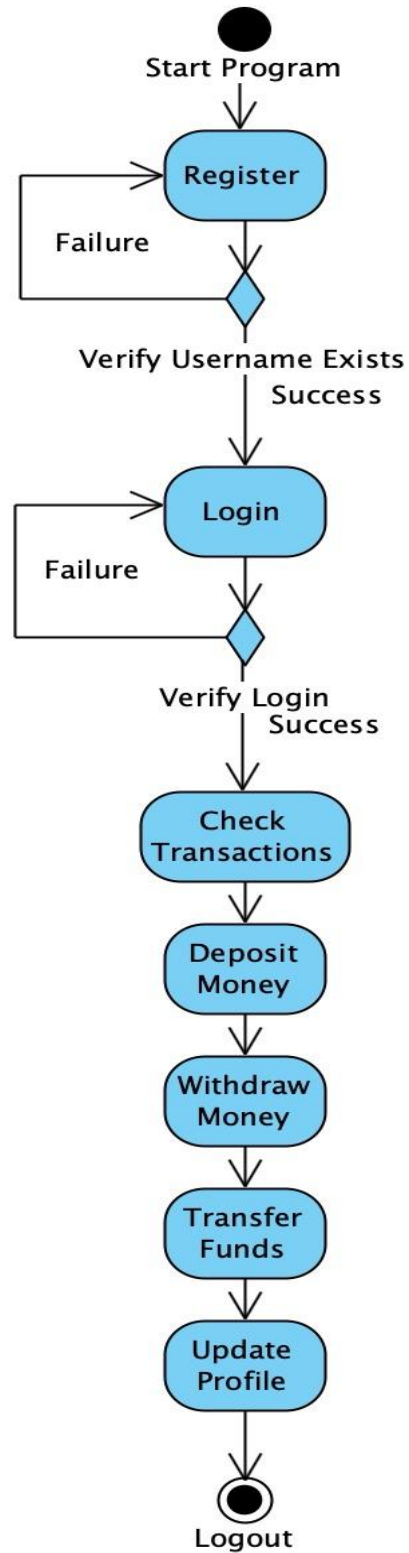
The features that are available to the Administrator are:

- Checking Balance
- Update username or password
- Deposit money
- Withdraw money
- Transfer money
- Checking personal information
- Checking Transactions and Account History

2.2 USE CASE MODEL :



2.3 ACTIVITY DIAGRAM :



DESIGN PHASE

Introduction

1.1 Scope and purpose

A purpose statement affects the design process by explaining what the developer wants the project to do, rather than describing the project itself.

The Design Document will verify that the current design meets all of the explicit requirements contained in the system model as well as the implicit requirements desired by the customer.

1.2 Overall System Design Objectives

The overall system design objective is to provide an efficient, modular design that will reduce the system's complexity, facilitate change and result in an easy implementation. This will be accomplished by designing strongly cohesion system with minimal coupling. In addition, this document will provide interface design models that are consistent, user friendly and will provide straight forward transition through the various system functions.

1.3 Project Structure

Database Package: this package contains database configuration classes for the system.

Views Package: this package contains everything related to the graphical aspect of the system. It is responsible for rendering what the user sees when using the software.

Controllers Package: this package as the name implies controls the behavior of the system. More precisely, it is composed of several controller

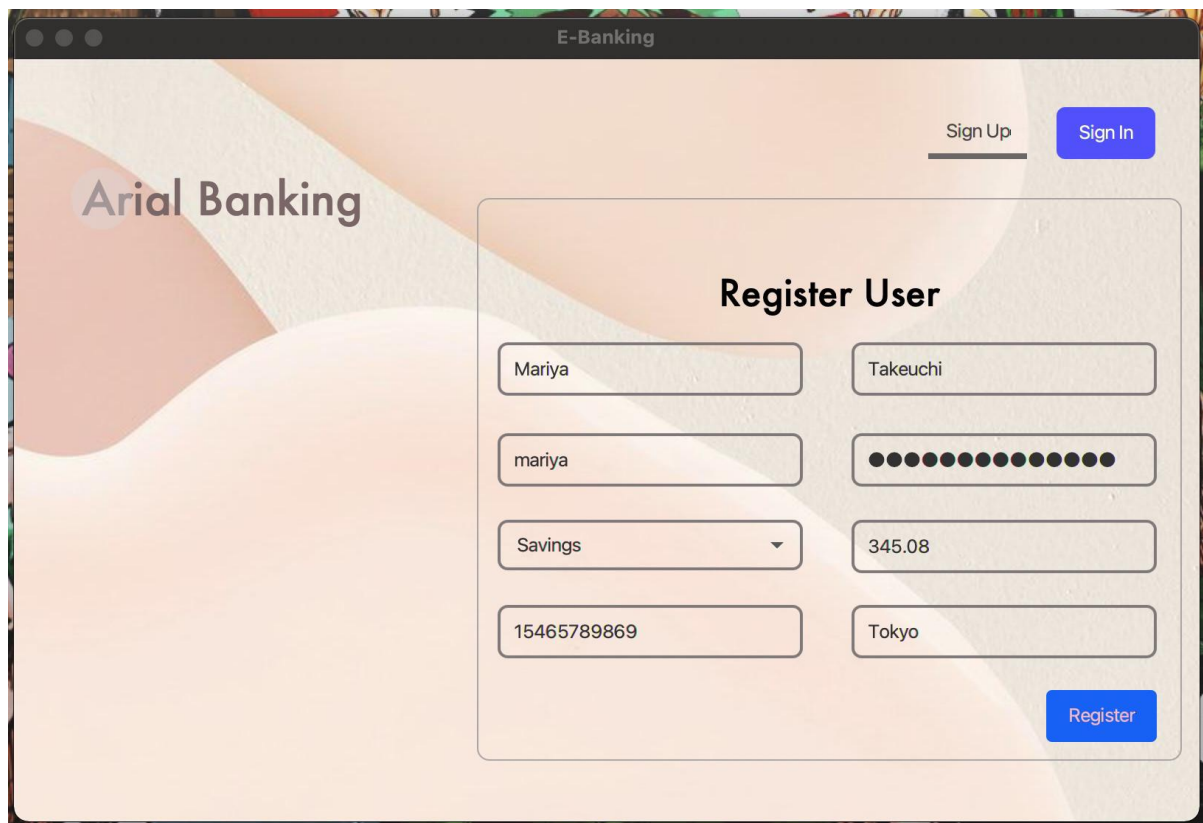
classes for each view template in the system. It makes sure that every action taken by the user is answered with the correct information.

Models Package: this package is more abstract compared to others in the system. It contains sets of rules that the Controllers package need in order to function. It designed to eliminate data redundancy in the project and allow for better security. It is easily manageable and scalable to meet the programmer's needs.

Public Package: this package is responsible for running the program and switching between pages. It is the main part of the system.

USER INTERFACE

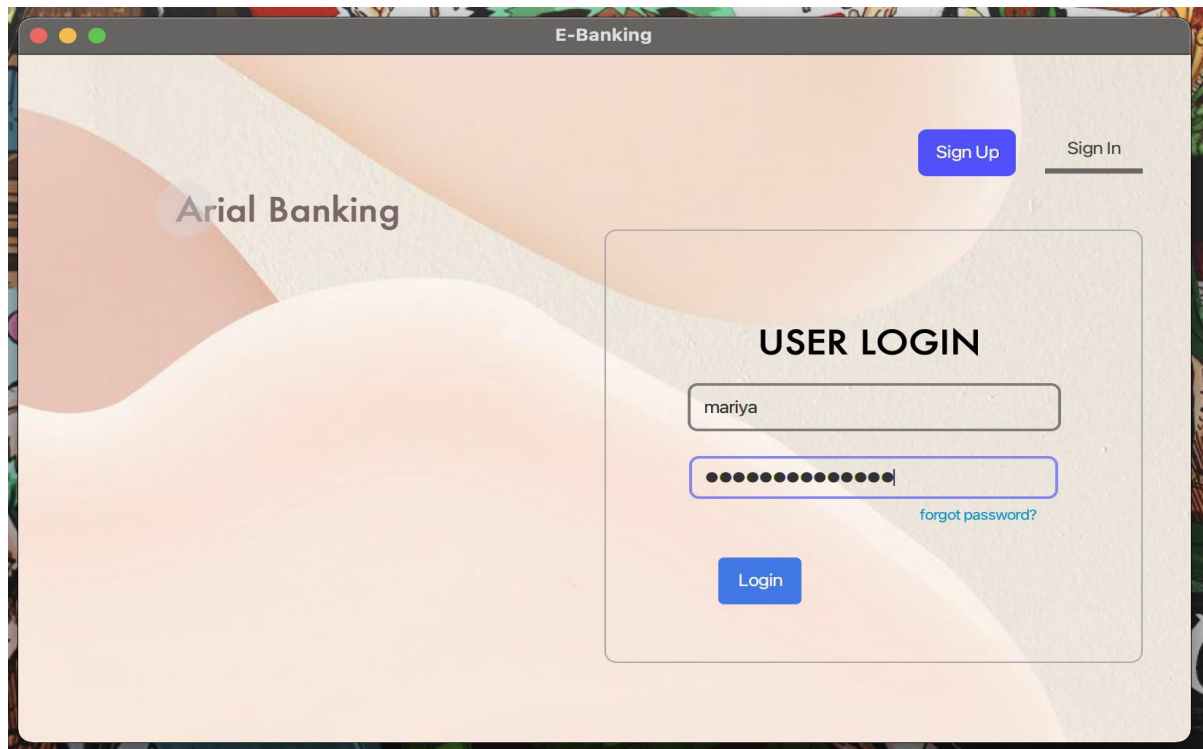
User Registration



The screenshot displays the 'E-Banking' application window. On the left, the 'Arial Banking' logo is visible. In the top right corner, there are links for 'Sign Up' and 'Sign In'. The central focus is the 'Register User' form, which contains the following fields and controls:

- First Name: Text input field containing 'Mariya'.
- Last Name: Text input field containing 'Takeuchi'.
- Username: Text input field containing 'mariya'.
- Password: Password input field represented by 12 dots.
- Account Type: A dropdown menu currently showing 'Savings'.
- Balance: Text input field containing '345.08'.
- Phone Number: Text input field containing '15465789869'.
- Location: Text input field containing 'Tokyo'.
- Register Button: A blue button labeled 'Register' located at the bottom right of the form.

User Login



The image shows a web browser window titled "E-Banking". The background is a light beige color with abstract orange and white shapes. In the top right corner, there are two buttons: "Sign Up" (blue) and "Sign In" (underlined). The "Arial Banking" logo is on the left. The main content area is titled "USER LOGIN". It contains a text input field with the username "mariya", a password input field with masked characters "••••••••••••••••", and a "Login" button (blue). A link "forgot password?" is located below the password field.

E-Banking

Sign Up Sign In

Arial Banking

USER LOGIN

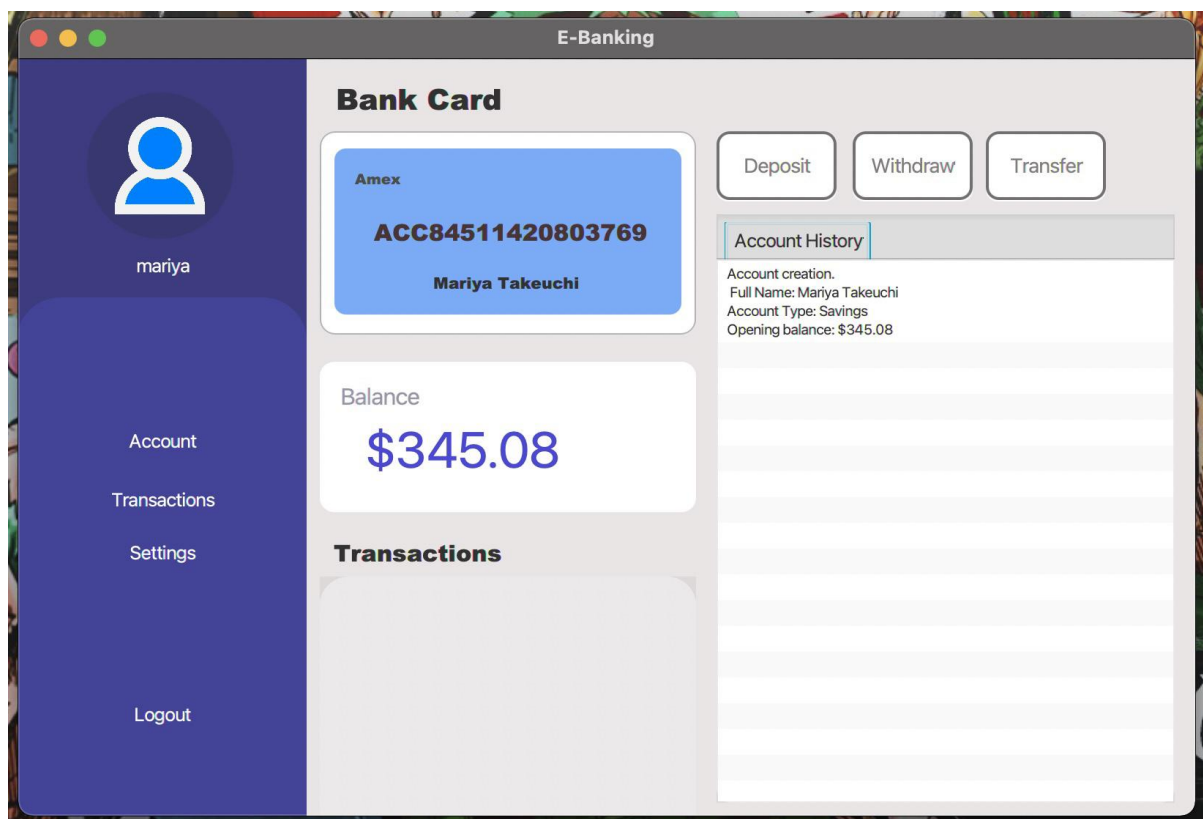
mariya

••••••••••••••••

[forgot password?](#)

Login

User Account



The image shows a web browser window titled "E-Banking". The interface is divided into a dark blue sidebar on the left and a main content area on the right. The sidebar contains a user profile icon, the name "mariya", and a list of menu items: "Account", "Transactions", "Settings", and "Logout". The main content area is titled "Bank Card" and features a blue card with the Amex logo, the card number "ACC84511420803769", and the name "Mariya Takeuchi". Below the card is a "Balance" section showing "\$345.08". To the right of the card are three buttons: "Deposit", "Withdraw", and "Transfer". Below these buttons is an "Account History" section with a list of transactions. The first transaction is "Account creation." with details: "Full Name: Mariya Takeuchi", "Account Type: Savings", and "Opening balance: \$345.08".

E-Banking

Bank Card

Amex

ACC84511420803769

Mariya Takeuchi

Balance

\$345.08

Transactions

Account History

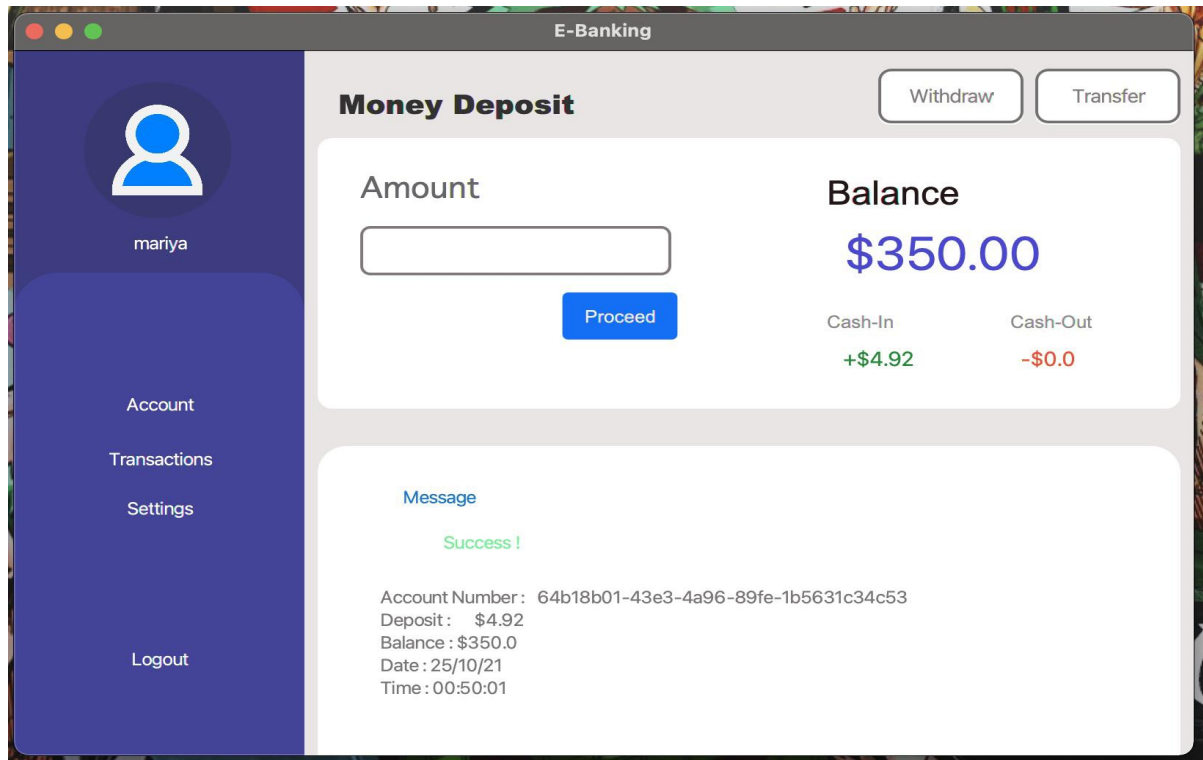
Account creation.
Full Name: Mariya Takeuchi
Account Type: Savings
Opening balance: \$345.08

Deposit Withdraw Transfer

mariya

Account
Transactions
Settings
Logout

Make Deposit



The screenshot shows the 'Money Deposit' page of an E-Banking application. On the left is a dark blue sidebar with a user profile icon and the name 'mariya'. Below the profile are links for 'Account', 'Transactions', 'Settings', and 'Logout'. The main content area has a light gray header with the title 'Money Deposit' and two buttons: 'Withdraw' and 'Transfer'. The 'Deposit' button is active. Below the header, there is a form with an 'Amount' input field and a 'Proceed' button. To the right of the form, the 'Balance' is displayed as '\$350.00' in large blue text. Below the balance, there are two columns: 'Cash-In' showing '+\$4.92' in green and 'Cash-Out' showing '-\$0.0' in red. At the bottom, a 'Message' section indicates 'Success!' in green. The message details are: Account Number: 64b18b01-43e3-4a96-89fe-1b5631c34c53, Deposit: \$4.92, Balance: \$350.0, Date: 25/10/21, Time: 00:50:01.

E-Banking

Money Deposit Withdraw Transfer

Amount

Balance
\$350.00

Cash-In: +\$4.92
Cash-Out: -\$0.0

[Proceed](#)

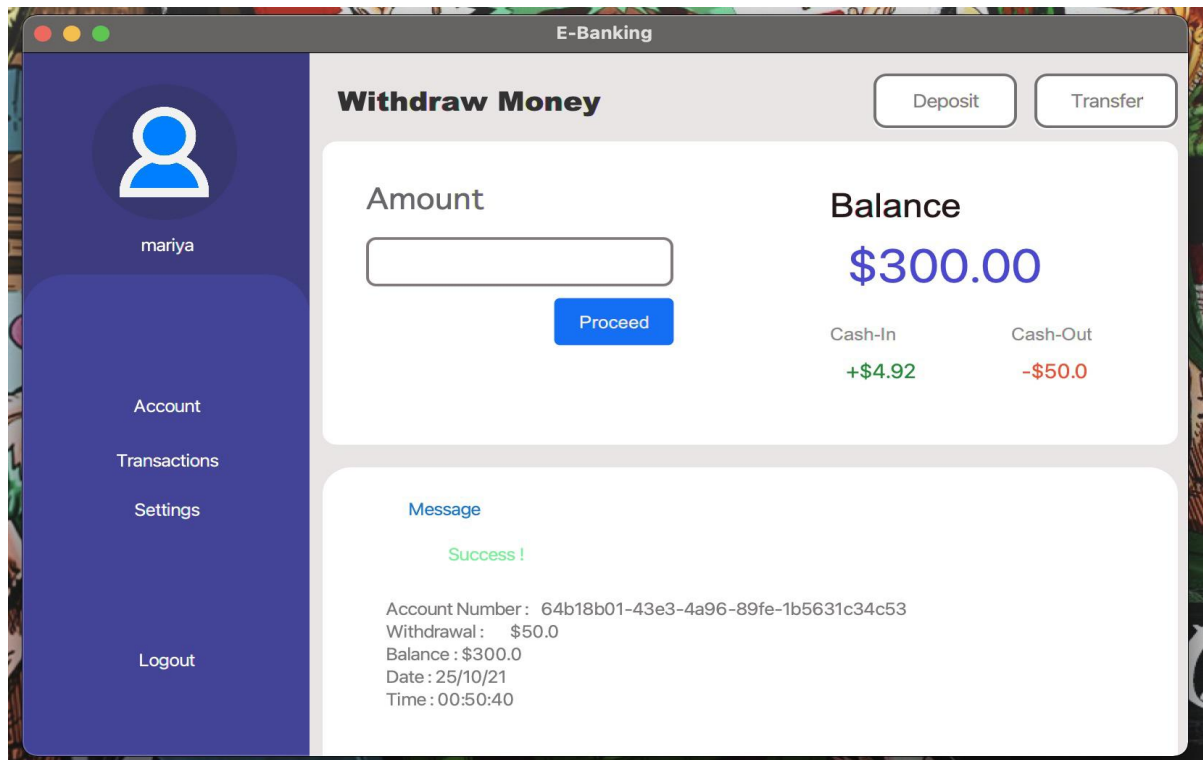
[Message](#)

Success !

Account Number: 64b18b01-43e3-4a96-89fe-1b5631c34c53
Deposit: \$4.92
Balance: \$350.0
Date: 25/10/21
Time: 00:50:01

[Account](#)
[Transactions](#)
[Settings](#)
[Logout](#)

Withdraw Money



The screenshot shows the 'Withdraw Money' page of an E-Banking application. The layout is similar to the deposit page, but the title is 'Withdraw Money' and the 'Withdraw' button is active. The 'Amount' input field is empty. The 'Balance' is displayed as '\$300.00' in large blue text. Below the balance, 'Cash-In' shows '+\$4.92' in green and 'Cash-Out' shows '-\$50.0' in red. The 'Message' section indicates 'Success!' in green. The message details are: Account Number: 64b18b01-43e3-4a96-89fe-1b5631c34c53, Withdrawal: \$50.0, Balance: \$300.0, Date: 25/10/21, Time: 00:50:40.

E-Banking

Withdraw Money Deposit Transfer

Amount

Balance
\$300.00

Cash-In: +\$4.92
Cash-Out: -\$50.0

[Proceed](#)


[Message](#)

Success !

Account Number: 64b18b01-43e3-4a96-89fe-1b5631c34c53
Withdrawal: \$50.0
Balance: \$300.0
Date: 25/10/21
Time: 00:50:40

[Account](#)
[Transactions](#)
[Settings](#)
[Logout](#)

Transfer Funds



mariya

Account

Transactions

Settings

Logout

E-Banking

Money Transfer

DepositWithdraw

Transfer To

Account Number

Amount

Proceed

Balance

\$174.51

Cash-In

Cash-Out

+\$4.92

-\$50.0

Message

Success !

Account Number : ACC84511420803769
Transfer to : ACC95952307442633
Deposit : \$125.49
Balance : \$174.51
Date : 25/10/21
Time : 00:51:53

Transactions Table

[illegible]

Account Settings

E-Banking

Settings

Full Name : Mariya Takeuchi

Account ID : 64b18b01-43e3-4a96-89fe-1b5631c34c53

Card Number : ACC84511420803769

Username : mariya

Password : ●●●●●●●●●●●●●●●●

Balance : \$345.08

Creation date : 2021-10-25 00:45:17.775445

Save changes

FUTURE SCOPE OF THE PROJECT

- ❖ The Online Banking System can be enhanced to include some other functionality like bank statement.
- ❖ An administrator view can be added to manage the system.
- ❖ The system can be expanded to integrate different branches of the bank.
- ❖ Integrate other aspects of banking into the system. Some more abstract essential for a fully function-able bank.
- ❖ Support for other banks API can be added.