

**Project Report**

**on**

**BANKING SYSTEM**

Submitted to

**AMBERSON HIGH SCHOOL**

in partial fulfilment of the requirements for the award of degree of

**HIGH SCHOOL**

**SUBMITTED BY SUPERVISED BY**

**COLLIN HUANG 000600609723 MS.NAMARTA VIJ**

**SANDY LYU 000600609687 TEACHER OF ICS3U**

**7100 BIRCHMOUNT ROAD, MARKHAM, ON L3R 4J2**

**CANADA**

**MARCH 2025**

***To whom it may concern***

This is to certify that the final project for ICS3U Grade 11 Computer Science, titled BANKING SYSTEM, has been successfully completed by Collin Wenqi Huang and Sandy Yongxin Lyu under the guidance and supervision of Ms. Namarta Vij.

As students of ICS3U, we have applied the programming concepts and problem-solving skills learned throughout the course to develop this project.It highlights our skills in coding programs based on the concepts learned in computer science.

This project is being submitted as part of the course requirements at Amberson High School, and it reflects our dedication, teamwork. We appreciate the support and guidance provided by our instructor throughout the learning process.

、

***Declaration***

I, Collin Huang, hereby declare that the final project for ICS3U Grade 11 Computer Science, titled BANKING SYSTEM, is my original work. This project has not been published or submitted elsewhere and was completed under the guidance of Ms. Namarta Vij as part of the course requirements. Any external literature, data, or resources used in this project have been properly cited and acknowledged in the reference section.

***March 1st, 2025***

*Collin Huang*

Student No.000600609723

***Declaration***

I, Sandy Lyu, hereby declare that the final project for ICS3U Grade 11 Computer Science, titled BANKING SYSTEM, my original work. This project has not been published or submitted elsewhere and was completed under the guidance of Ms. Namarta Vij as part of the course requirements. Any external literature, data, or resources used in this project have been properly cited and acknowledged in the reference section.

***March 1st, 2025***

*Sandy Lyu*

Student No.000600609687

***Table of Content***

1. -Introduction
2. -Project Objectives
3. -System Design and Implementation
4. -Features and Functionality
5. -Challenges and Solutions
6. -Conclusion
7. -HTML Implementation and Screenshots
8. -Team Contributions

***Introduction***

This project is a Banking System built using Java for the backend and HTML for the frontend. The system allows users to create accounts, check balances, deposit and withdraw money, and transfer funds. It also includes an admin panel where administrators can view all accounts, manage transactions, and block users if needed.

The backend is written in Java and follows basic object-oriented programming ideas, like using classes to organize data, keeping details private (encapsulation), and making the code reusable (inheritance). The system stores account and transaction data in ArrayLists, making it easy to manage multiple users and their activities. Each function, such as creating an account, logging in, making transactions, and checking balances, is handled in separate methods to keep the code organized.

The frontend is a simple HTML webpage that presents banking features in a clear and easy-to-read way. It includes:

A Home Page with a short introduction about the bank's services.

-An Account

Overview Table listing sample account details.

-A User Account Section explaining how users can manage their accounts.

-An Admin Section describing admin functions like viewing and blocking accounts.

The frontend is built with HTML, presenting users with a simple yet functional interface where they can access banking services. This project aims to simulate a real-world banking environment while applying programming concepts in a practical manner.

This report explains how the system works, how the code is structured, and how Java and HTML work together to create a simple banking application. The goal of this project is to practice Java programming concepts while also making a functional and easy-to-use banking system.

***Project Objectives***

The primary objectives of this project are:

-To apply Java programming concepts such as classes, objects, inheritance, and encapsulation.

-To implement basic banking operations like account management, transactions, and fund transfers.

-To create a modular system where each component handles specific functionalities (e.g., user management, transactions, admin functions).

-To develop an interactive and user-friendly web interface using HTML.

-To enhance problem-solving and debugging skills while handling user inputs and edge cases.

## 

## *System Design and Implementation*

The backend system is developed in Java, following OOP principles to structure the code efficiently. The core components include:

#### 1. Account Management

* Account class: Stores user account details (account ID, name, balance, type).
* Methods:
  + createAccount() – Creates a new user account.
  + viewBalance() – Displays the account balance.
  + deposit() – Adds money to the account.
  + withdraw() – Deducts money from the account.
  + transferFunds()
  + Transfers money between accounts.

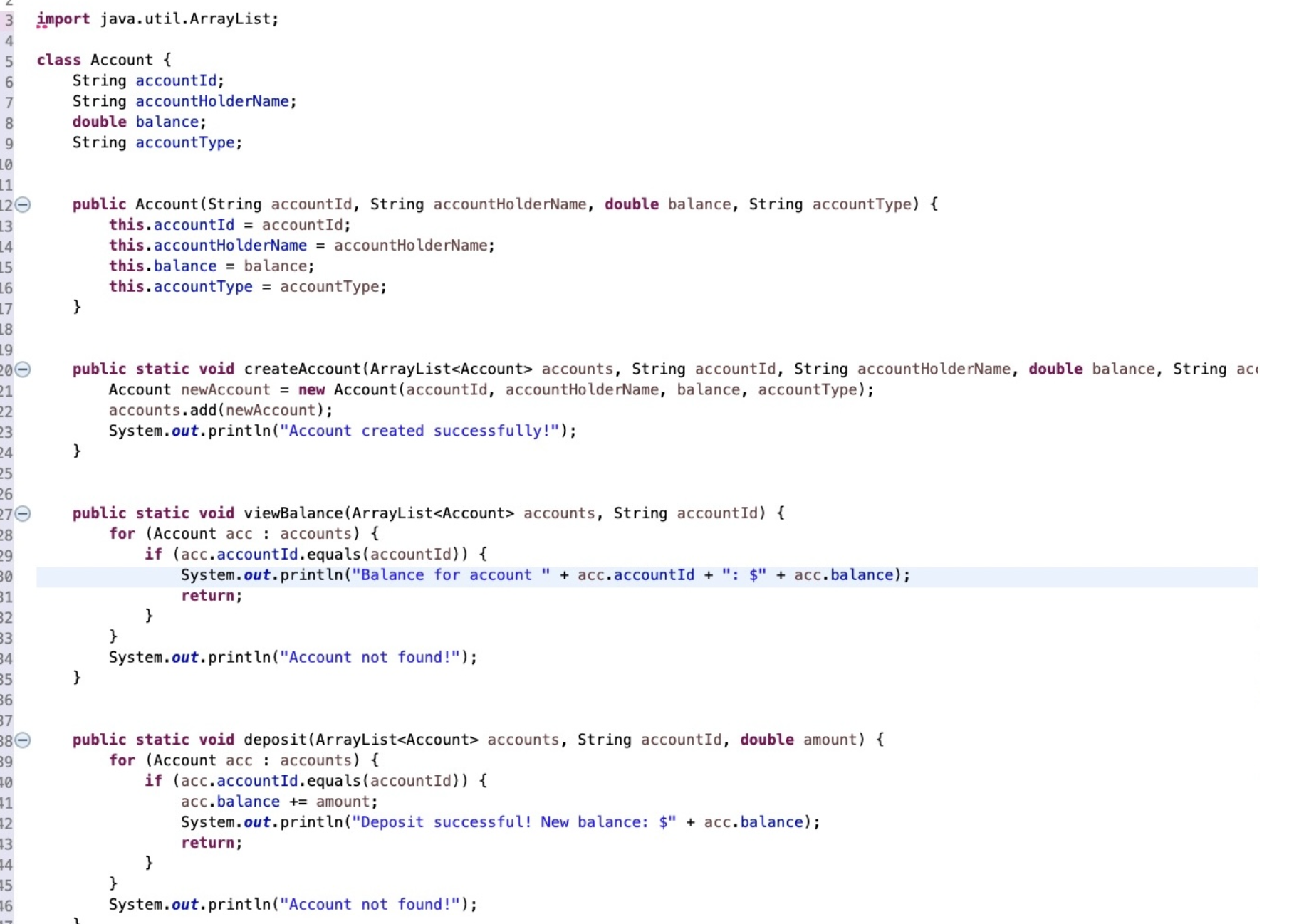
#### 

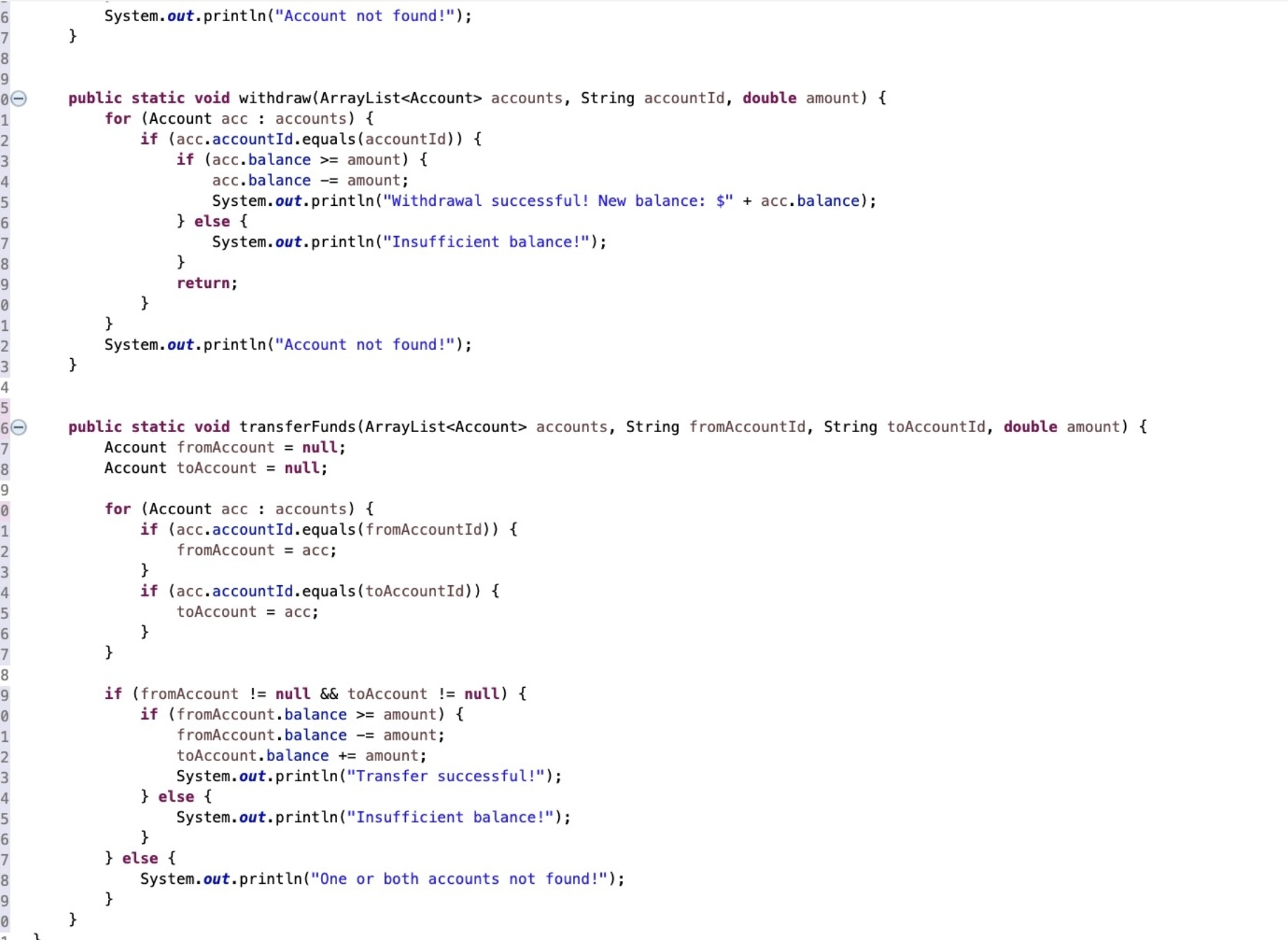
#### 

#### 

#### 

#### *Account Management JAVA ScreenShot：*





#### 

#### 

#### 

#### 

#### 

#### 

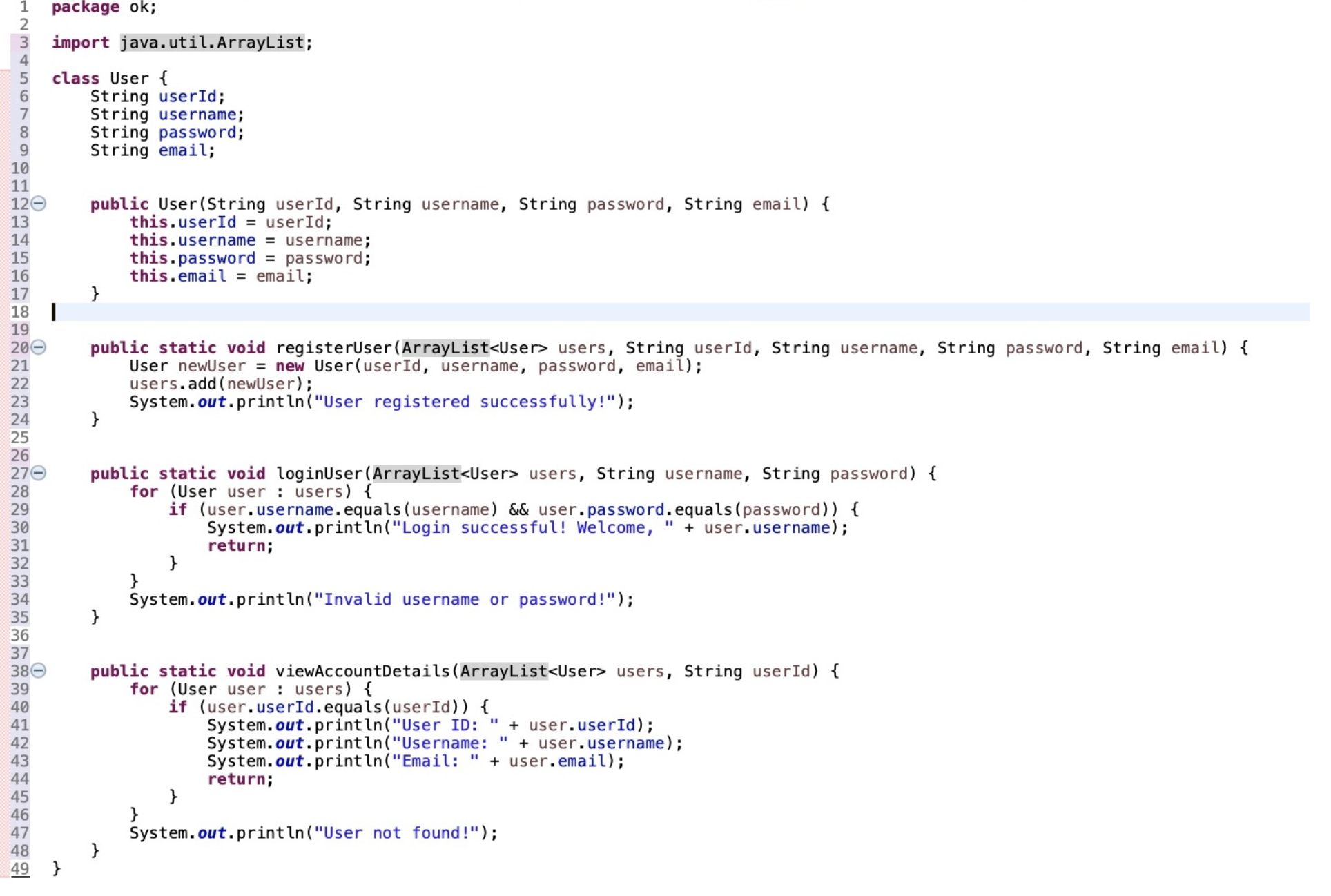
#### 

#### 

#### 

#### 2. User Management

* User class: Handles user details (user ID, username, password, email).
* Methods:
  + registerUser() – Registers a new user.
  + loginUser() – Authenticates users.
  + viewAccountDetails() – Displays user details.



#### 

#### 

#### 

#### 

#### 

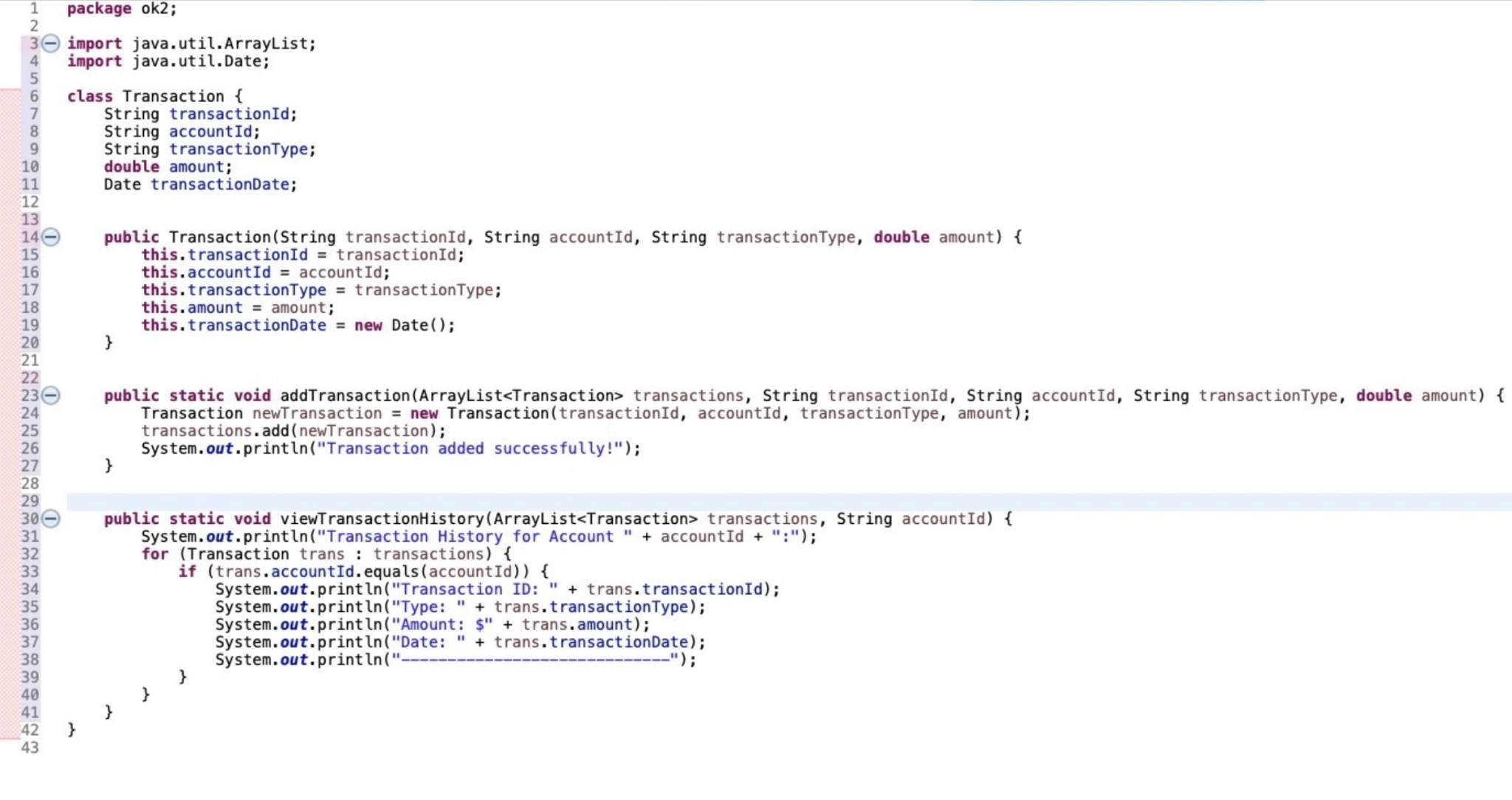
#### 

#### 3. Transaction Management

* Transaction class: Stores transaction details (transaction ID, type, amount, date).
* Methods:

-addTransaction() – Records transactions for user activity tracking.

-viewTransactionHistory() – Displays a user’s past transactions.



#### 

#### 

#### 

#### 

#### 

#### 

#### 4. Admin Panel

* Admin class (inherits from User class) with extra privileges.
* Methods:
  + viewAllAccounts() – Displays all accounts.
  + blockUser() – Blocks a user from accessing the system.
  + manageTransactions() – Allows the admin to review and monitor transactions.



### 

### 

### 

### 

### 

### 

### 

### 

### *HTML Frontend*

The web interface provides a user-friendly experience for interacting with the banking system. The HTML pages include:

* Home Page – Displays an introduction to the banking system and its services.
* Account Overview Table – Lists sample accounts with columns for ID, name, balance, and status.
* User Account Section – Explains account features and includes a link to a signup page.
* Admin Section – Describes admin privileges, such as user management.

The frontend is kept simple with essential HTML elements like headings, tables, paragraphs, and links. A background color and formatting enhance readability.

***Features and Functionality***

The Banking System includes the following core functionalities:

### 

### User Features:

* Create an account with a unique Account ID.
* Deposit and withdraw funds with proper balance updates.
* View balance and transaction history.
* Transfer money between accounts securely.

### 

### Admin Features:

* View all user accounts and balances.
* Block users from the system if needed.
* Manage transactions and review activity logs.

The system ensures error handling by preventing invalid inputs, checking for insufficient balances, and handling authentication failures

## *Challenges and Solutions*

During development, we faced some challenges were encountered, and solutions were implemented:

***Challenges:***

-Managed user authentication securely

-Handling invalid transactions (e.g., overdraft withdrawals)

-Ensuring correct fund transfers

-Improving code organization

***Solutions:***

-Used Java String comparison to match usernames and passwords

-Implemented balance checks before processing withdrawals

-Verified account existence and sufficient balance before allowing transfers

-Used modular methods for each feature, reducing redundancy

These solutions ensure that the system runs smoothly, efficiently, and securely.

|  |
| --- |

## 

## *Conclusion*

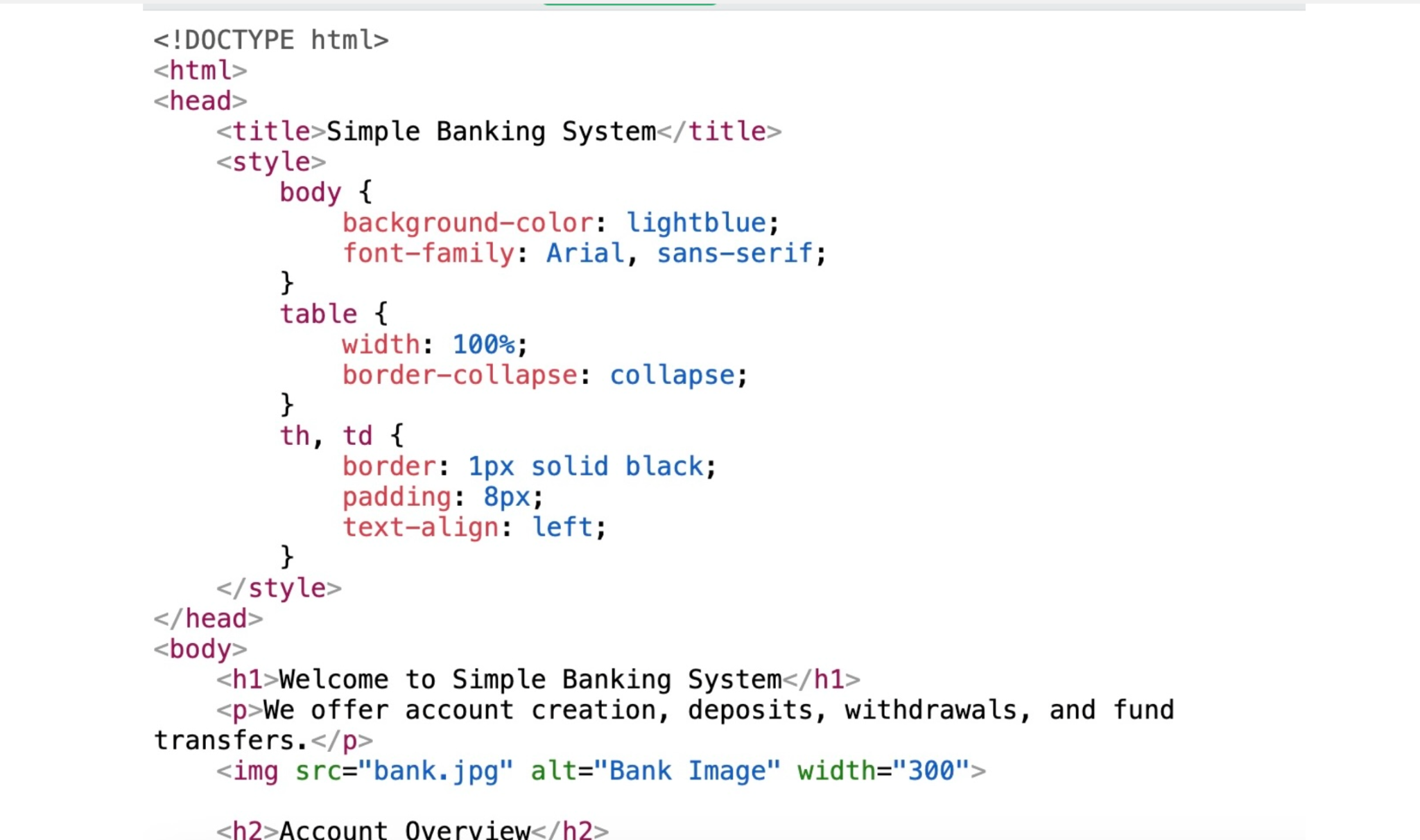
This Banking System Project successfully applies Java programming concepts while simulating real-world banking operations. By implementing features such as account creation, deposits, withdrawals, and fund transfers, the project provides a practical learning experience in software development.

The project also demonstrates the importance of structured programming, as Java’s OOP principles helped organize the code into reusable components. The HTML frontend provides an accessible way for users to interact with the system, making it user-friendly and easy to navigate.

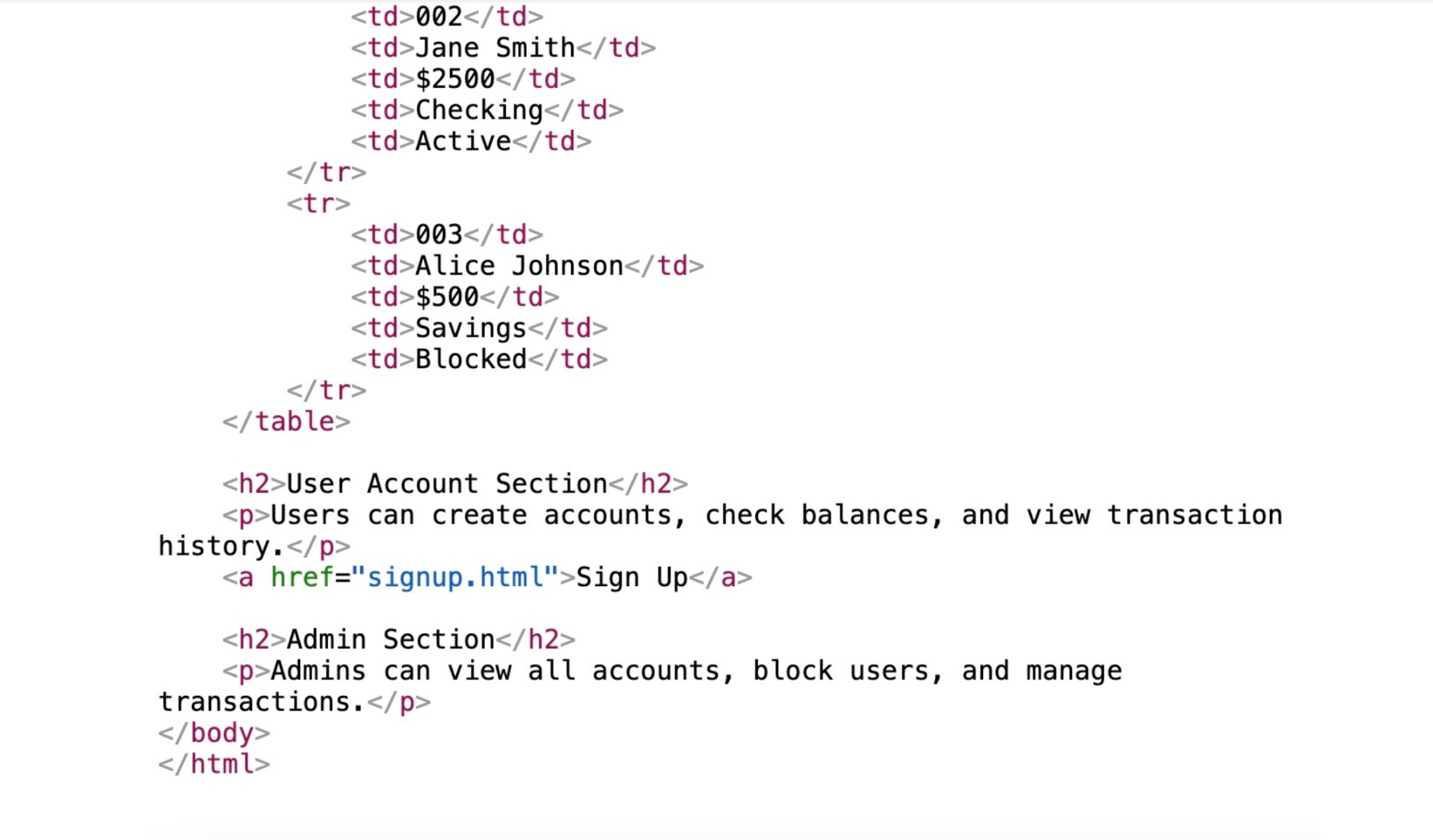
In conclusion, this project strengthens coding skills, enhances problem-solving abilities, and lays a foundation for future projects involving web-based financial applications. With further improvements, additional features such as database integration and a graphical user interface could be added to make the system even more powerful and interactive.

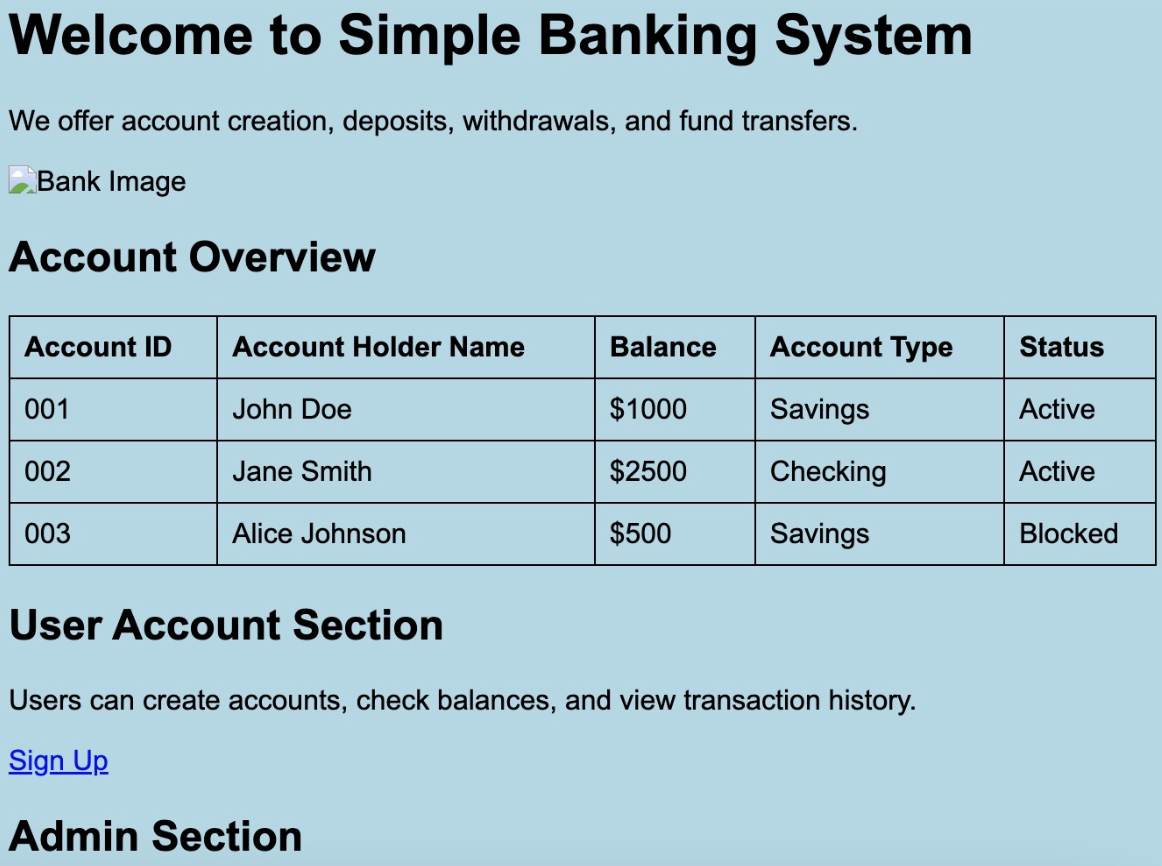
## *HTML Implementation and Screenshots*

Below are the screenshots of the HTML implementation showcasing the frontend design:









***Team Contributions***

* **Java Development(Sandy):**

-Developed the backend system for account management, transactions, and admin controls.

-Applied OOP principles such as inheritance, encapsulation, and modular programming.

-Implemented error handling and security checks for transactions.

* **HTML Frontend(Sandy):**

-Designed and structured the web interface for users and admins.

-Created account tables, forms, and navigation elements.

-Applied basic styling and formatting to enhance readability.

* **Final Report(Collin):**

-Documented project objectives, system design, and features.

-Explained the backend and frontend implementation in detail.

-Included screenshots and diagrams for better visualization.

* **PowerPoint Presentation(Collin & Sandy):**

-Summarized key points for a clear and concise presentation.

-Structured slides to explain system workflow and implementation.

-Ensured easy-to-understand content for the audience.