Bank Information Management System

User Requirements

A bank needs a system to support their important business activities and record relevant information. It should be a highly secure, internal bank system intended for the use of bank employees only, particularly customer assistants (CAs) and bank managers.

The bank's policy allows customer assistants to view the data about clients, other employees, bank branches, and the securities offered by the bank. Additionally, they should be able to update the clients' information and manipulate their accounts whenever it is necessary, like setting the limits, freezing and unfreezing an account, etc. It is important that they do not have the right to manipulate the clients' loans, but they can only view them.

One of the main tasks of a CA is to open new bank accounts for clients. The bank offers three types of accounts: checking, savings, and investment. When a customer assistant needs to open a new account, they choose the correct type and fill out the necessary information.

Bank managers should have control over the entire system. They should be able to perform all the functionalities of a customer assistant, and in addition to that, have the authority to manipulate the important information about the bank itself and adjust the data in accordance with the business decisions that they introduce. They should have control over the employees recorded in the system, change what securities the bank decides to offer to their customers, and modify the data about the bank's branches. They are also responsible for approving and rejecting clients' loan applications, so only they are allowed to record load information in the system.

The bank offers two types of loans: term and revolving. They should be properly stored with the information required for each, and the payments clients need to make for them should be calculated every month.

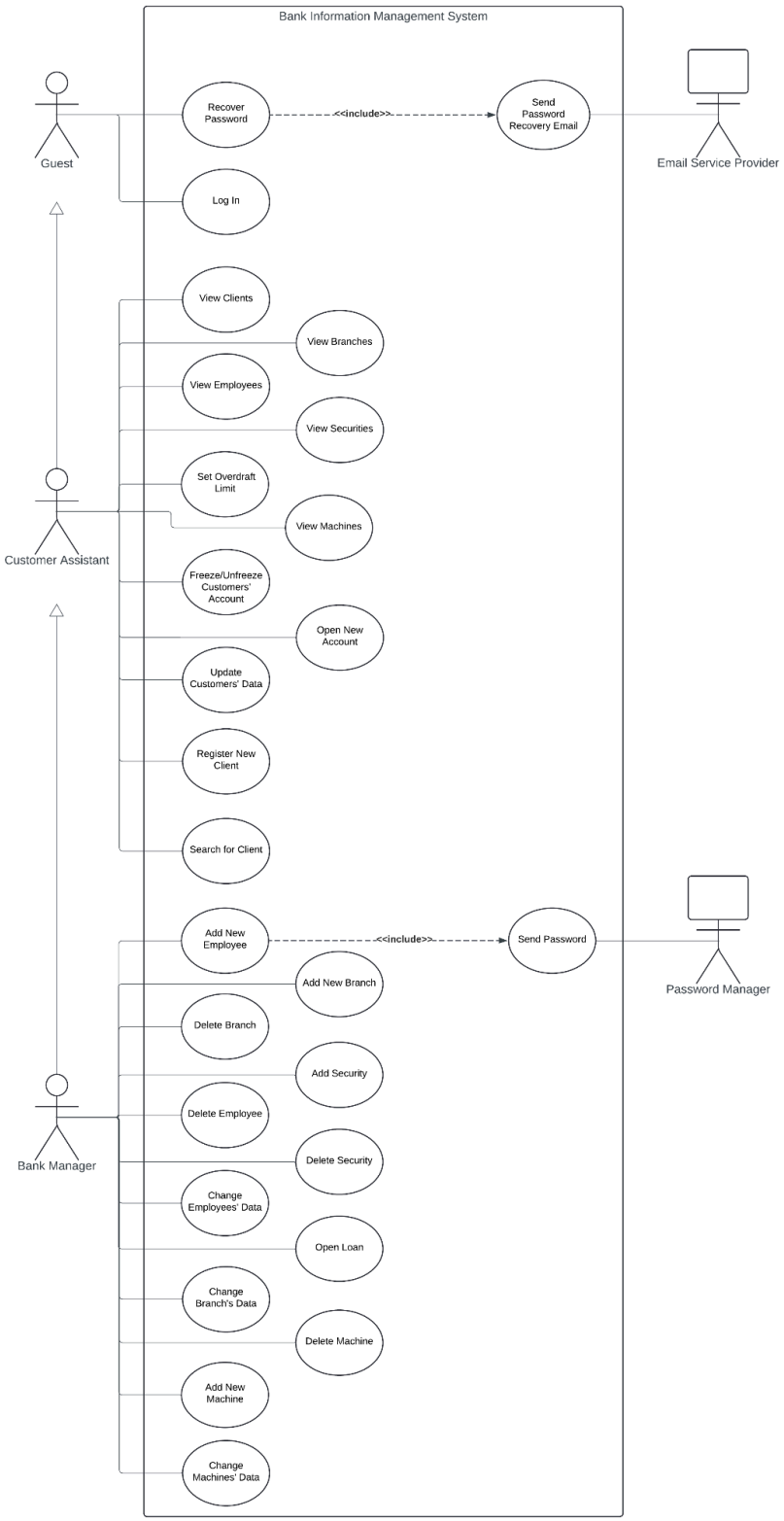
Bank managers should have full control over the employees' information and modify it as they wish. However, according to the bank's policy, the employees' salary cannot be increased or decreased by more than 50% of their current salary. Thus, it should be taken into account. Additionally, the bank is interested in knowing their interns and regular, full-time employees.

Also, bank managers should control the ATMs they have at each branch. However, there are three different types: the machine that only allows clients to deposit money, another one which only allows withdrawals, and the third one which allows both.

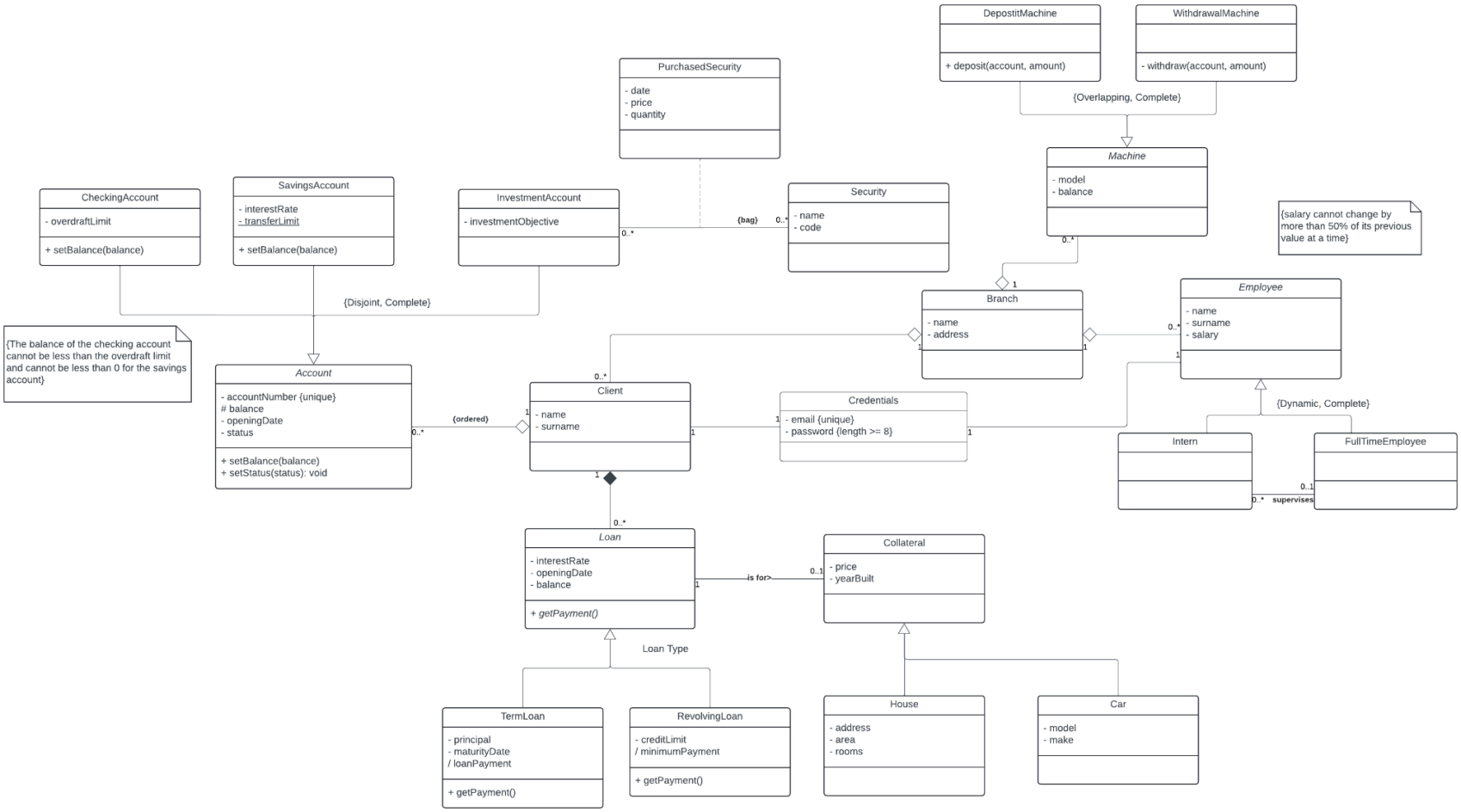
Lastly, the system should securely interact with two external systems which are crucial for the bank. First, there should be a way to safely generate and share new passwords for the

newly-hired employees so that they gain access to do their jobs. Second, there should be a convenient, fast, and, most importantly, secure way of recovering a password by an employee if they forget it.

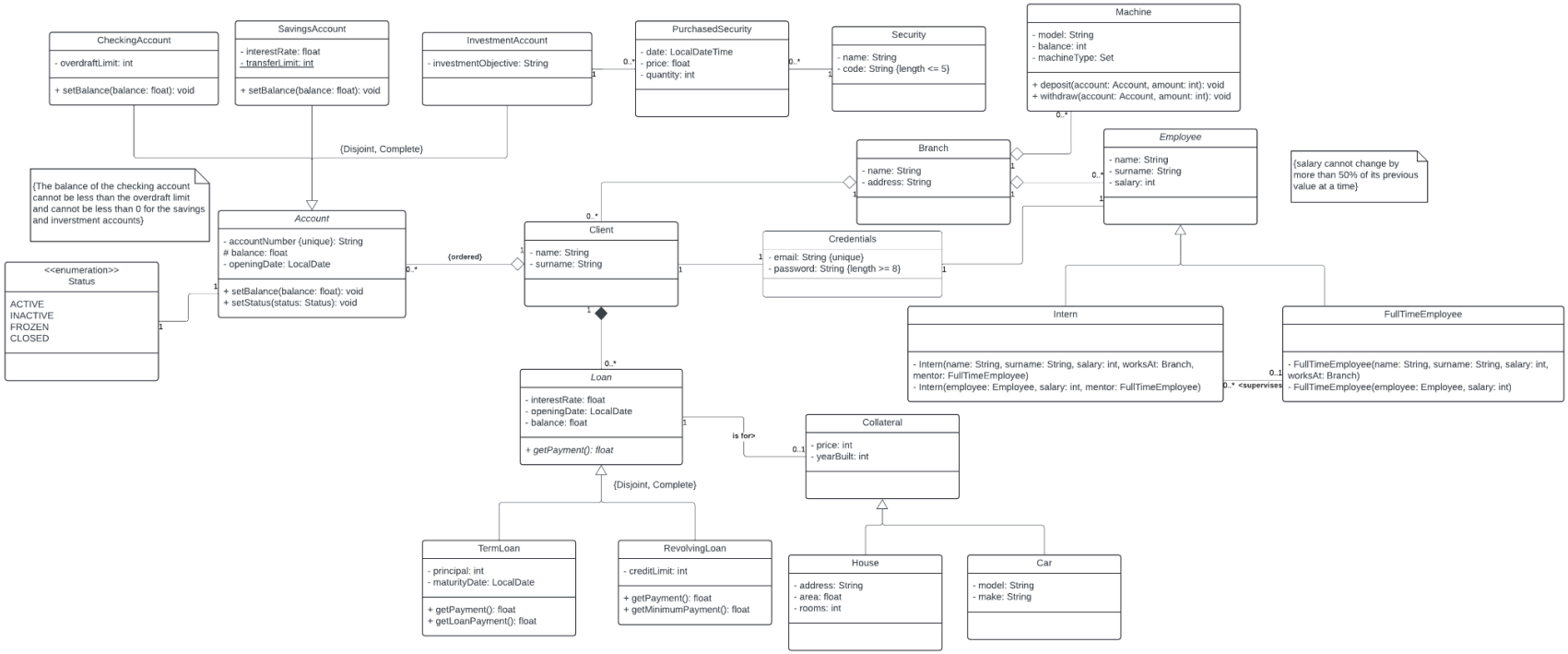
Use Case Diagram



Analytical Class Diagram



Design Class Diagram



Open New Account Use Case

Scenario

1.1 Actors

No.	Use-case step description
1.	User

1.2 Purpose and context

The user wants to open an account for the bank’s client.

1.3 Dependencies

1.3.1 Included use-cases

None

1.3.2 Extended use-cases

None

1.4 Assumptions and pre-conditions

No.	Use-case step description
1.	The user is logged in.

1.5 Initiating business events

The user entered the home page.

1.6 Basic flow of events

No.	Use-case step description
1.	The system displays all clients.
2.	The user selects a client.
3.	The system displays client information and accounts.
4.	The user clicks “Add account”.
5.	The system displays different account types.
6.	The user selects the “checking” type.
7.	The system displays a field for the overdraft limit.
8.	The user fills out and submits the form.
9.	The system validates the form.
10.	The system opens the account.

1.7 Alternative flows of events

1.7.1 Investment Account Type

Trigger: During step 5 of the Basic Flow, the user selects the “investment” type.

No.	Use-case step description
1.	The system displays a field for the investment objective.
2.	Return to step 7 of the Basic Flow.

1.7.2 Savings Account Type

Trigger: During step 5 of the Basic Flow, the user selects the “savings” type.

No.	Use-case step description
1.	The system displays a field for the interest rate.
2.	Return to step 7 of the Basic Flow.

1.7.3 Improperly Filled Values

Trigger: During step 8 of the Basic Flow, form validation fails.

No.	Use-case step description
1.	The system displays the errors to the user.
2.	Return to step 7 of the Basic Flow.

1.8 Extension points

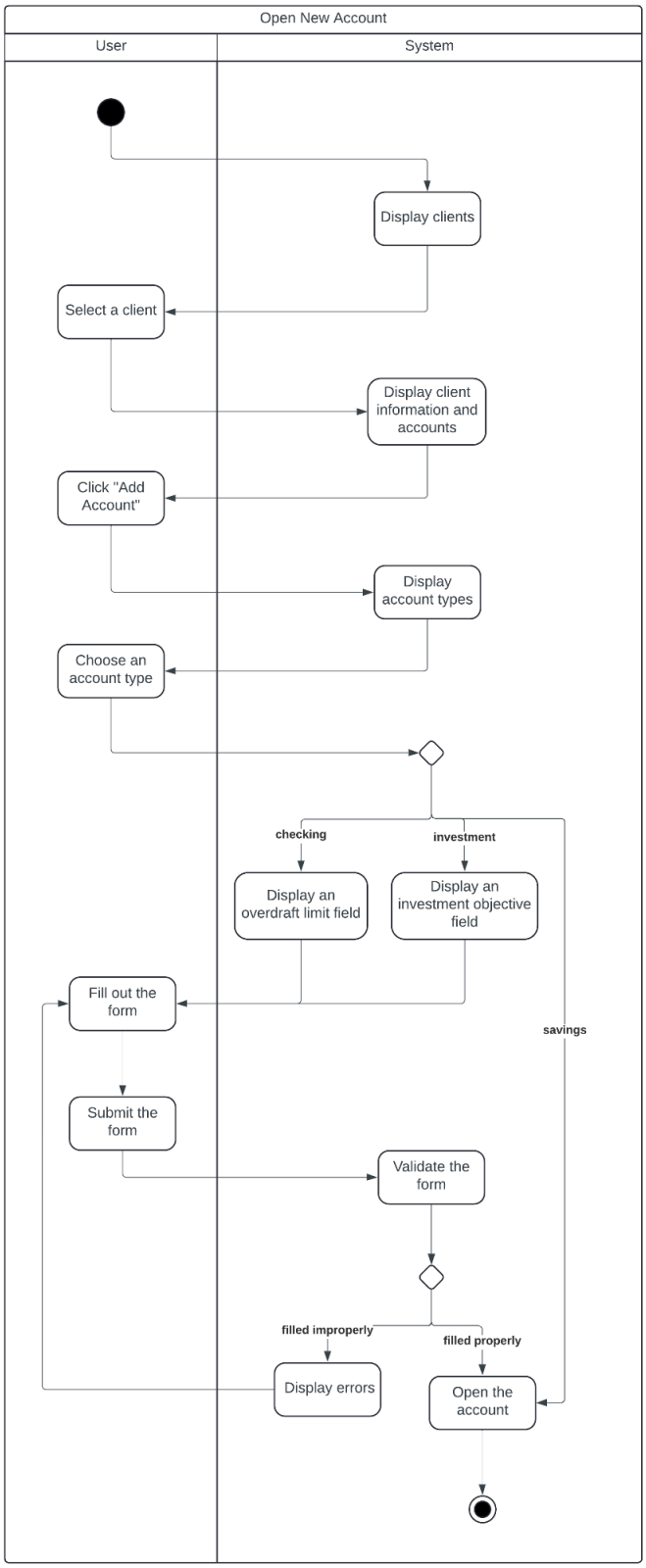
Extension point condition

None

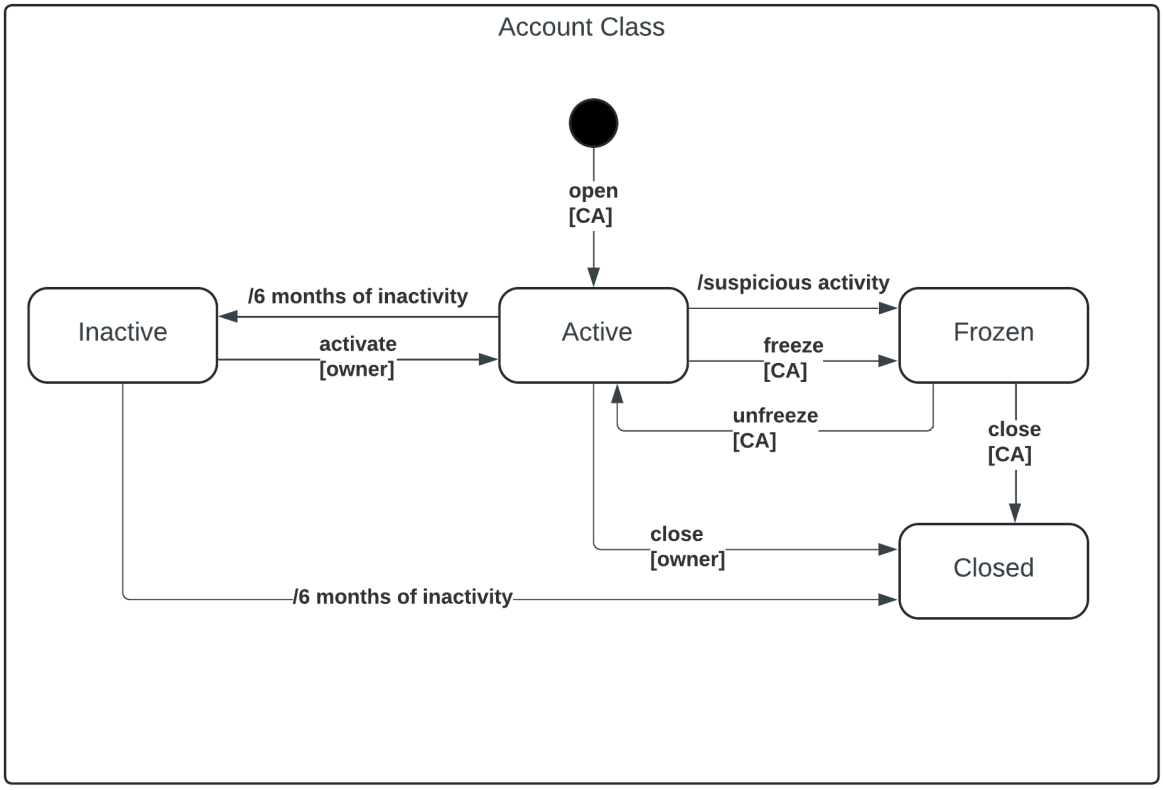
1.9 Post-conditions

No.	Use-case step description
1.	A new account is opened for the selected user.

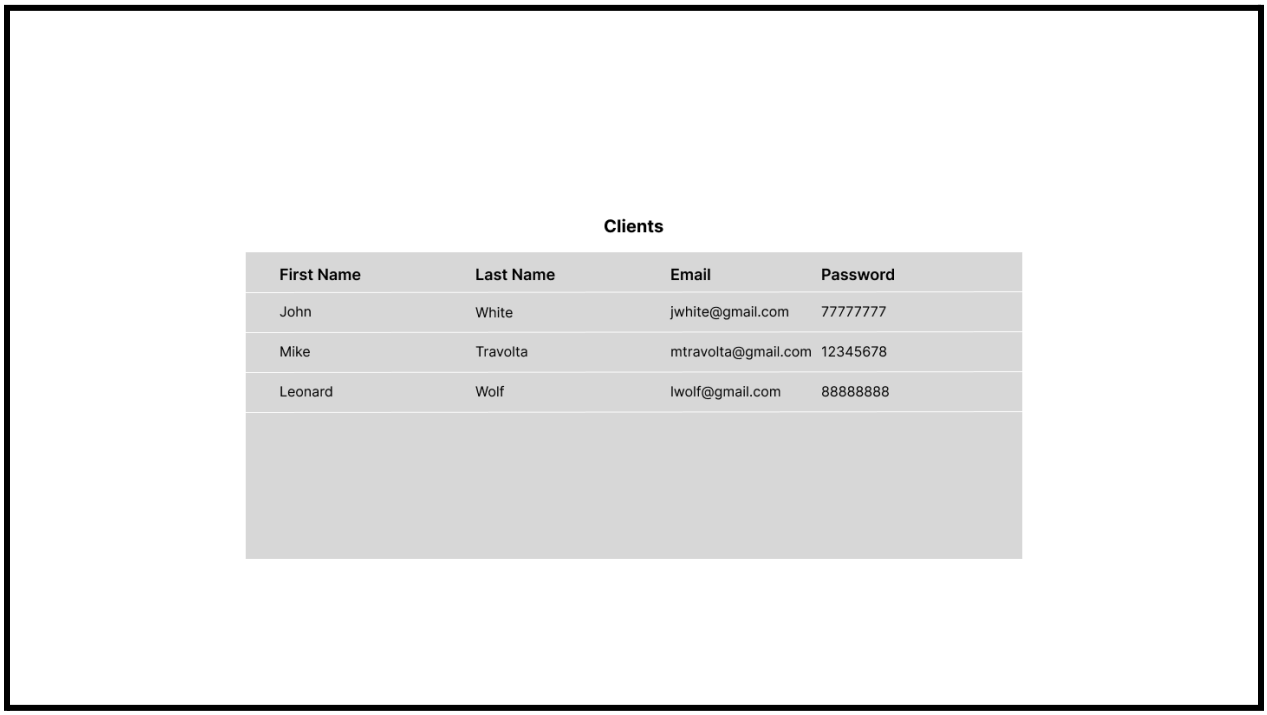
Activity Diagram

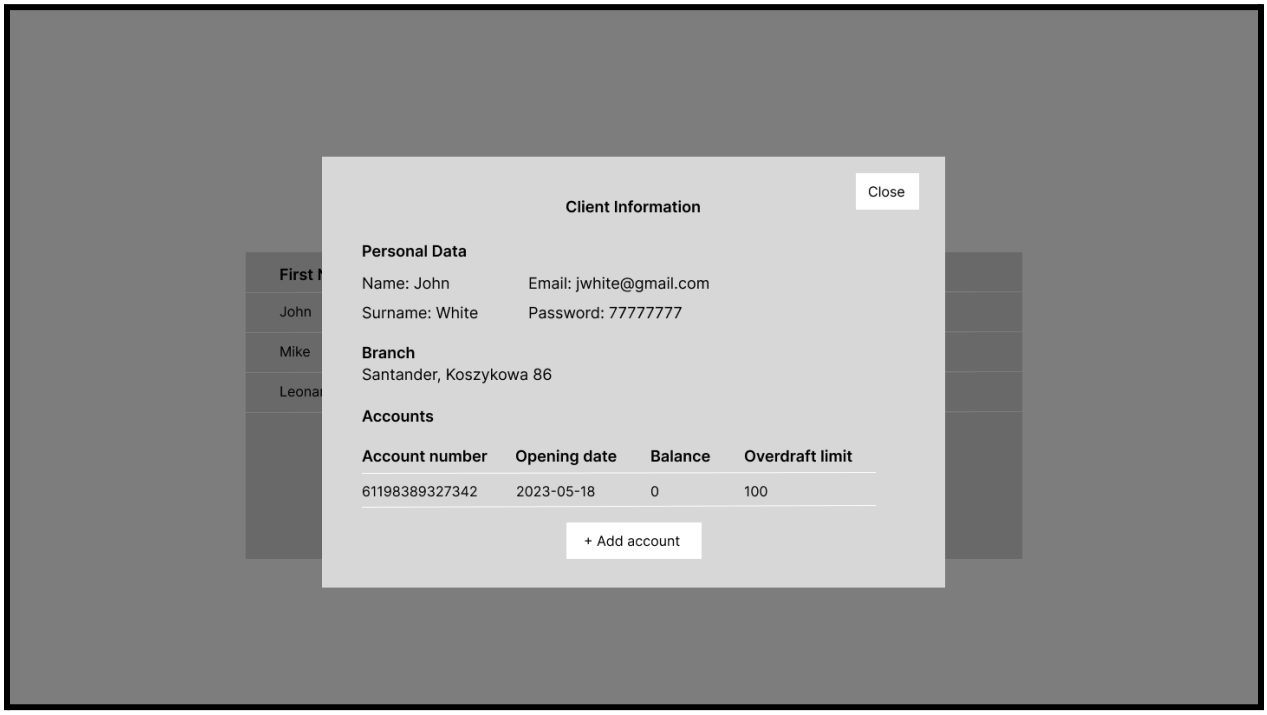


State Diagram



GUI Design


Clients
First Name
Last Name
Email
Password
John
White
jwhite@gmail.com
77777777
Mike
Travolta
mtravolta@gmail.com
12345678
Leonard
Wolf
lwolf@gmail.com
88888888


Close
Client Information
Personal Data
Name: John
Email: jwhite@gmail.com
Surname: White
Password: 77777777
Branch
Santander, Koszykowa 86
Accounts
Account number
Opening date
Balance
Overdraft limit
61198389327342
2023-05-18
0
100
+ Add account

Choose account type

Checking account

Savings account

Investment account

What should be the overdraft limit?

Overdraft limit

Back | Open →

Implementation Discussion

The system is implemented in Java and not all the concepts presented in the analytical class diagram are supported with out-of-the-box solutions.

Design Class Diagram

The design class diagram introduced the following changes:

- The hierarchy for the overlapping inheritance of the machine class was flattened, leaving only the machine class. It introduced an additional machineType attribute in the class to differentiate between different variations.
- To implement dynamic inheritance for the Employee subclasses, the copy constructor principle is utilized in each subclass.
- The bag association is implemented with the help of an intermediate class to represent the securities the client purchased.

Dynamic Analysis

As a result of the dynamic analysis, the following changes were introduced:

- The status of the account class is represented with an enum attribute which can be changed using the setStatus(status) method.