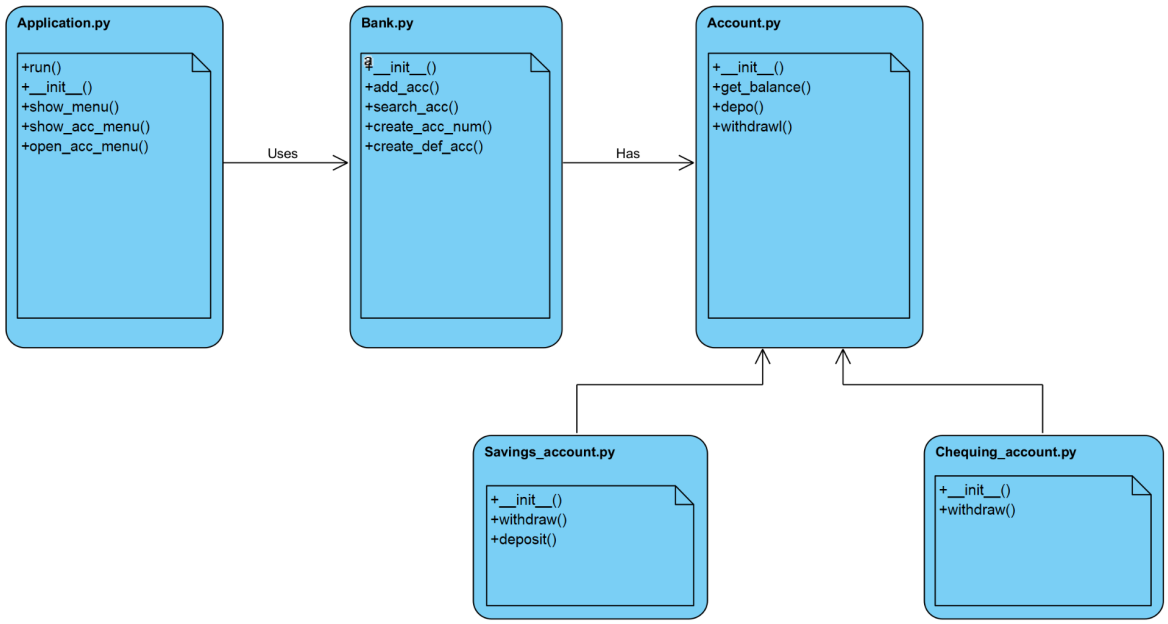
Class Relationships and Separation of Concerns
Programming Assignment 3

Marcel Wrzecionek

2023-11-18



For this assignment, the task was to create a simulation of a bank account system. In this assignment 5 different files have been created: application.py, account.py, bank.py, savings_account.py as well as chequing_account.py. Each of these files had its own unique task which then would be imported within each other to simulate the banking system.

Application.py: This file had been created as the main file to run the simulation. It has multiple functions within itself that all work to prompt text statements, account menus, main menus and more.

Account.py: This file had been created under the Account class to create the accounts themselves, simulate the balance checks, deposits and withdrawals of the money added.

Bank.py: This file had been made under the Bank class to simulate the creation of accounts at the bank, searching for the account number when logging in and creating account numbers/default accounts.

Savings_account.py: This file was created to act as the bank customers savings account. In this file the functions under the class SavingsAcc() each had their own deposit and withdrawal values. It had been created to keep the savings accounts balance separate from the chequing accounts balance as the two accounts must be separated.

Chequing_account.py: This file, similar to the savings account file, was created to act as the bank customers chequing account. Its main goal is to separate the balances and values between chequing and saving.

*These files are imported in one another to allow them to work together and prompt all import text statements and values when needed.

When creating this code, I had come across a few minor issues such as typos, or syntax errors. The idea of what I wanted the code to be like was less of an issue as I had worked in the banking field in the past so it was still fairly fresh to me how it worked (more or less). To overcome these issues, I had to debug constantly and run the code when I thought I had found the issue. I kept doing this until the code ran the way I wanted it to. In the future I would like to make the code more detailed and realistic, for example the bank number generation in my code would be 812781729 when in reality it would be a bit more complex. Another thing I would like to do is make the interface more user friendly. Rather than have it look like basic python code, I would like to try and implement some kind of design to make it more intriguing.