Executive Summary Team 07 Project 2

Jintong Luo (Requirement), Xinyue Hu (Development), Wenlin Zhu (Validation)

Introduction

In this project, we are developing a software that can manage a banking system correctly and safely. The system would provide a functional interface of both APP and ATM for users, that means our users could access the service both online and with the ATM machines. Through the interface, users could get services like open & close accounts, deposit & withdraw cash and transfer efficiently.

Our design & basic functions

Functional requirements

Our target customer is those who prefer online payment, since we provide the APP service which could benefit transfer a lot. We would provide the services below,

- Open account
- Close account
- Deposit cash
- Withdraw cash
- Transfer
- Transactions record

If we complete the development of this phase, we might add more services like,

- Online payment
- Cooperate with other institution, e.g. pay for the insurance , tuition fee online via our APP

Nonfunctional requirements

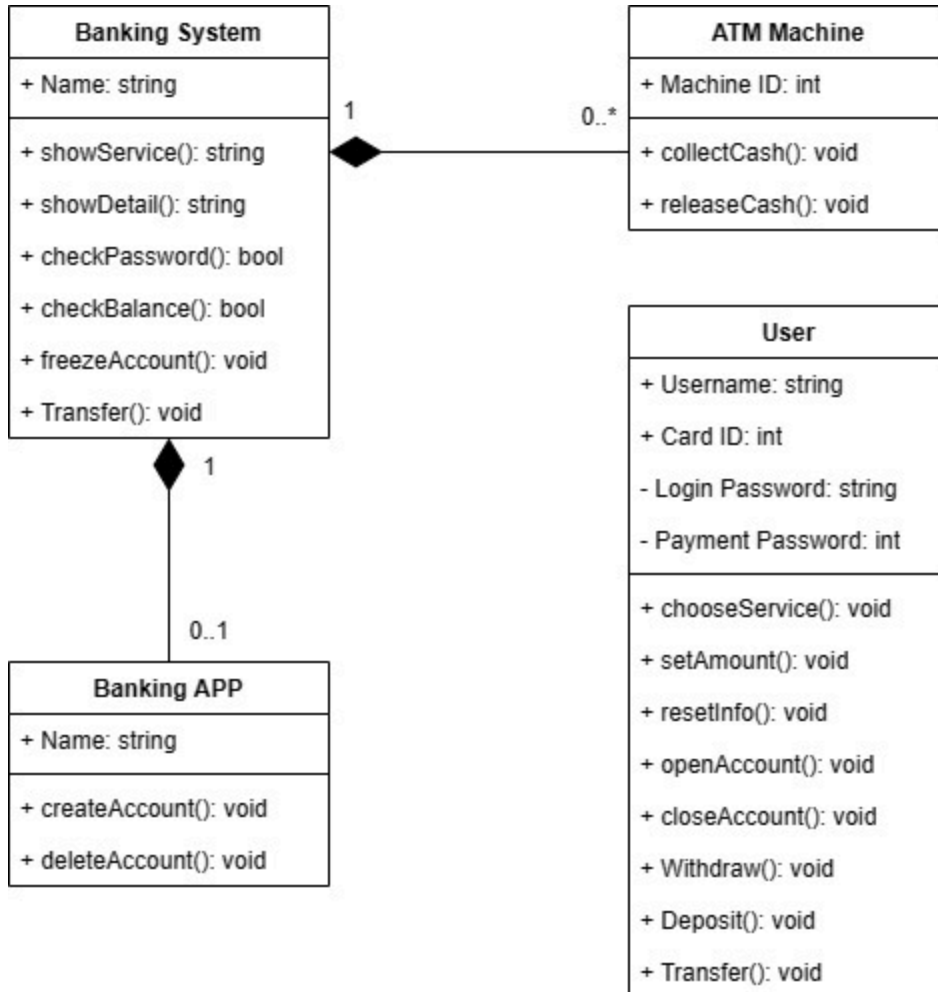As a banking system, our design should be both efficient and safe, that is

- The users could get clear instruction from the system UI.
- The system would protect the property of users by methods like checking passwords, asking twice before the users make his decision and saving the encoded passwords in the database.
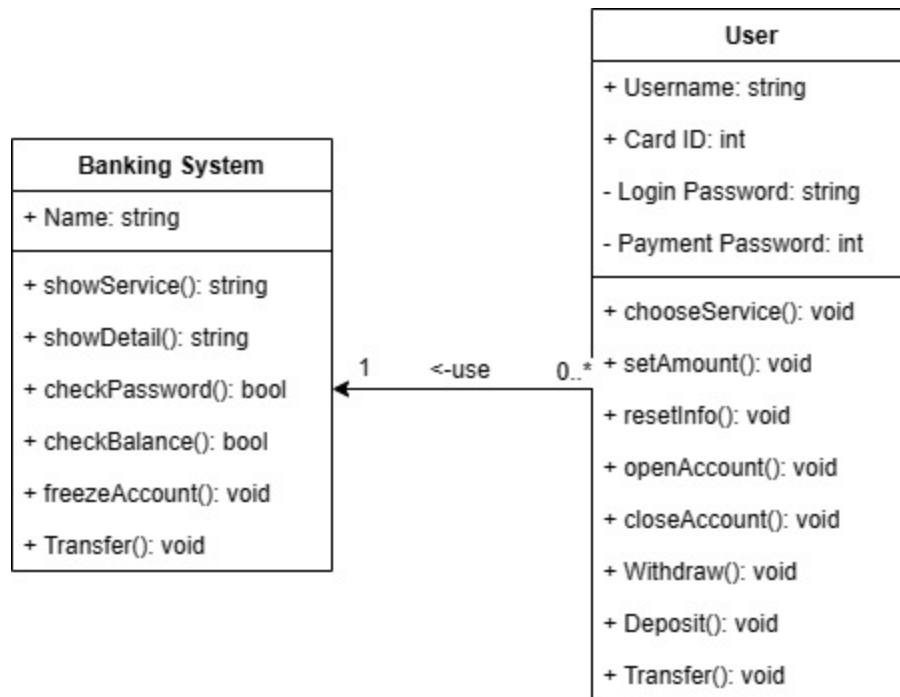
Our schedule

We have completed the requirement part, which you could find in our Gitlab repository, and we have been drawing the UML for development part then. After this consultation, we would start to develop our system via Python.

For your convenience, we will provide some useful UML then.

The participants of activities relating to banking system can be categorized into User, APP and ATM. The relationship of them are as follows:

Banking System

+ Name: string

+ showService(): string
+ showDetail(): string
+ checkPassword(): bool
+ checkBalance(): bool
+ freezeAccount(): void
+ Transfer(): void

ATM Machine

+ Machine ID: int

+ collectCash(): void
+ releaseCash(): void

1   0..*

1

0..1

Banking APP

+ Name: string

+ createAccount(): void
+ deleteAccount(): void

User

+ Username: string
+ Card ID: int
- Login Password: string
- Payment Password: int

+ chooseService(): void
+ setAmount(): void
+ resetInfo(): void
+ openAccount(): void
+ closeAccount(): void
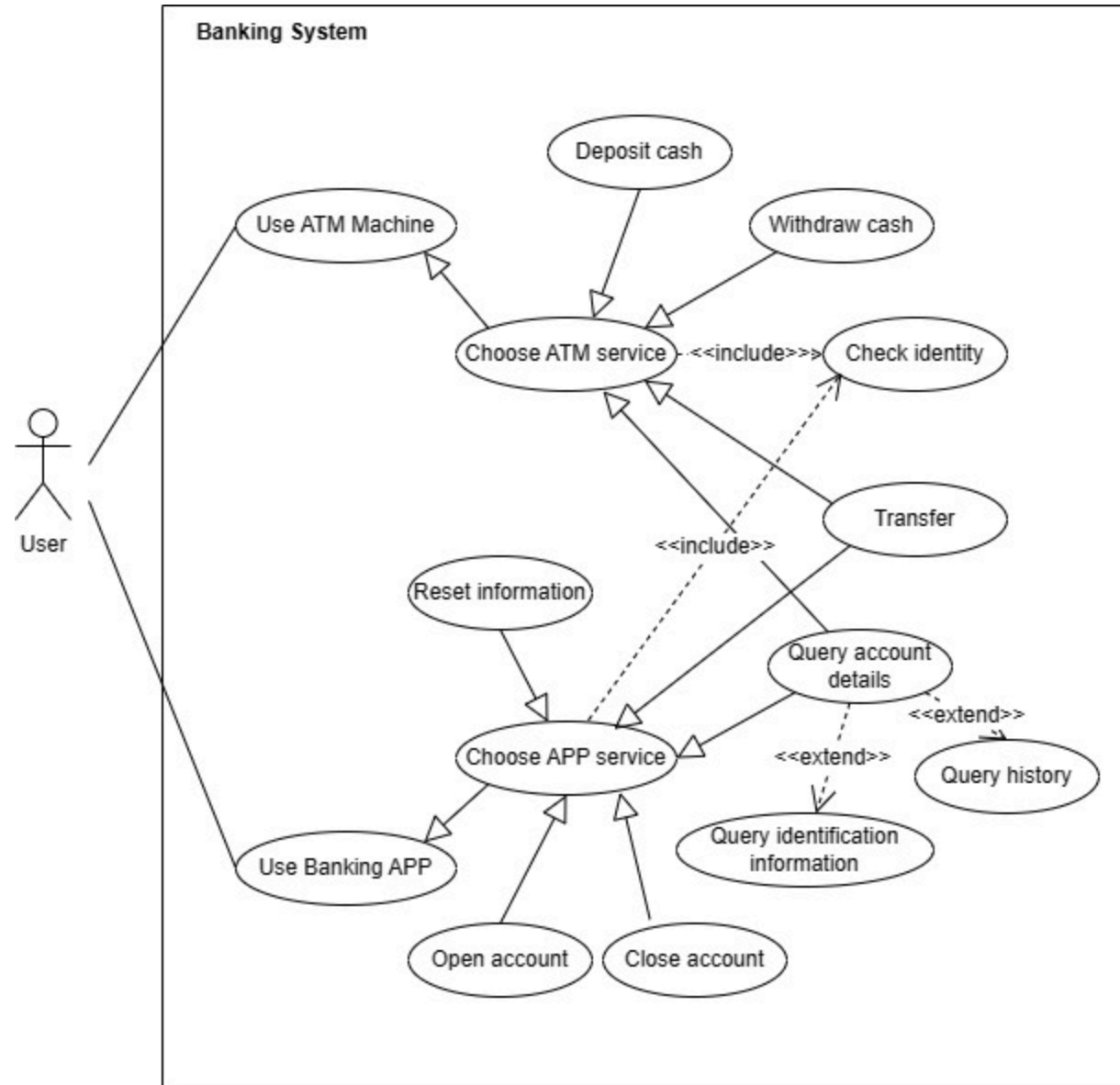+ Withdraw(): void
+ Deposit(): void
+ Transfer(): void

The relationships between the user and the banking system are shown as follows:

In addition, the system can achieve the following use cases from the user's and the bank's perspectives:

Our questions

- What is the formula , i.e. input & output, for our development?

- What should be included in the system architecture? Are they releted to our development?

- If we consider the hackers who might attack our database for information, e.g. passwords, what should be the intensity of our encryption algorithm?

Thank you!