Requirements Document

Team PB-PI

February 3, 2018

Table 1: Team

Name	ID Number
Alissa Bellerose	27377320
Sabrina D'Mello	27739486
Melanie Damilig	
Tobi Decary-Larocque	27407645
Zain Farookhi	26390684
Giulia Gaudio	27191766
Jason Kalec	40009464
Damian Kazior	40016168
Johnny Mak	
Philip Michael	40004861
Ramez Nicolas Nahas	26718108
Steven Tucci	40006014
Shunyu Wang	40043915

1 System

1.1 Purpose

1.2 Context

1.3 Business Goals

2 Domain Concepts

Figure 1: Domain model

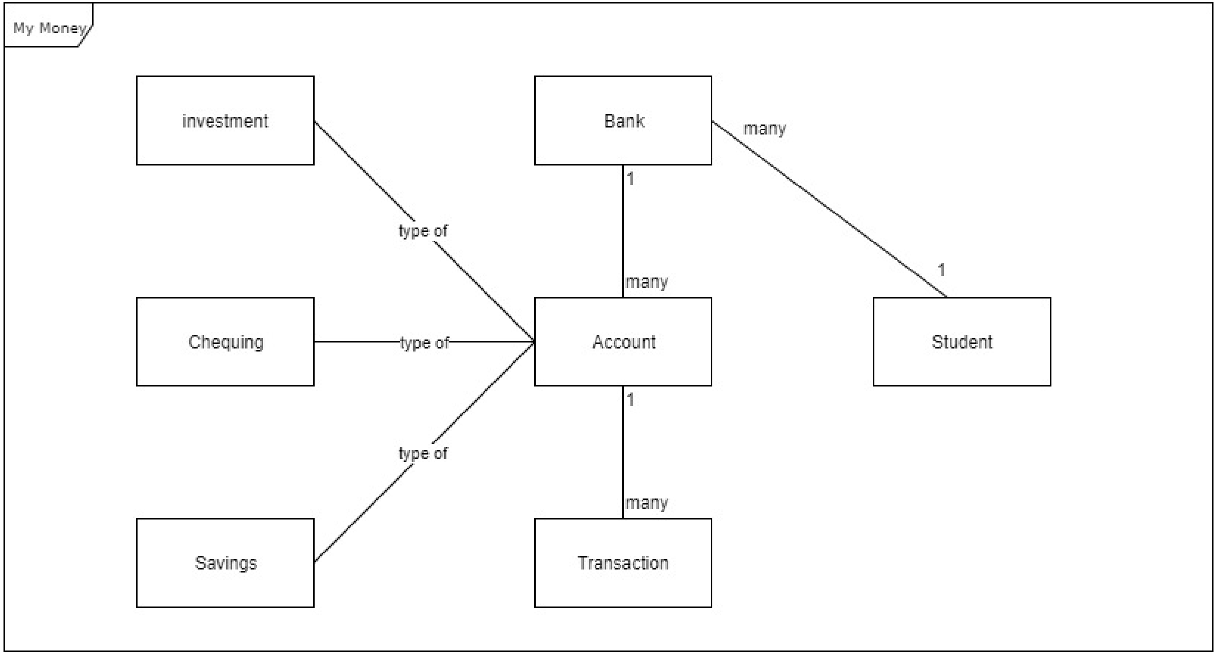

Table 2: Main domain concepts

Concept	Description
Student	Main user of the system. Student manually keeps the system up to date by entering every transaction performed during his day to day activities
Bank	Student's bank. When student has accounts accross different banks, his accounts stay organized within the application
Account	Account under given bank. Account can be of type chequing, savings or investment
Transaction	Ever activiny that sudent performs (spending or depositing money) is recorded as a transaction

3 System overview

myMoney application is a simple, personal founds management system. It tracks user's deposits and expenses across different banks and and account within these banks. System is maintained by the user and is not connected to any bank account.

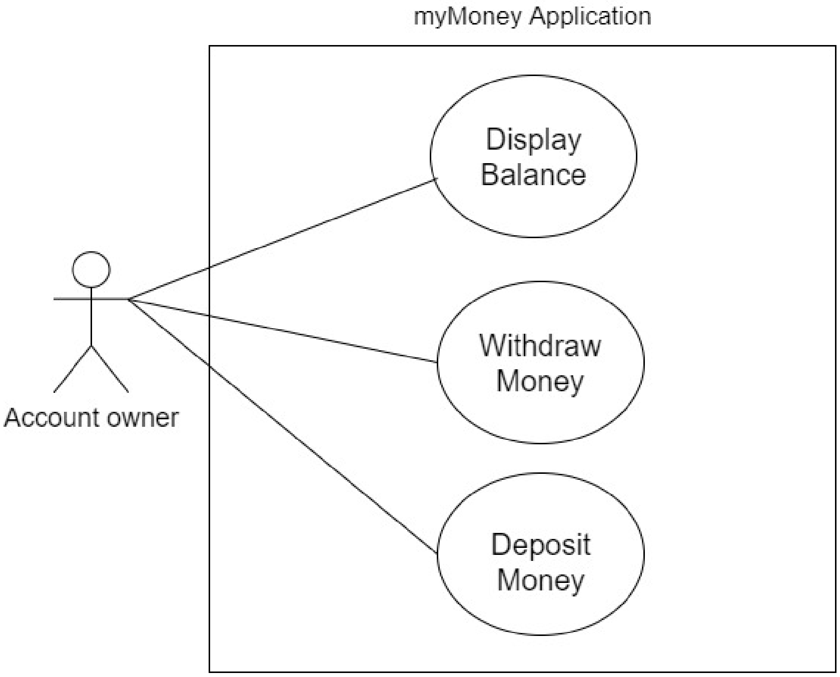

Figure 2: Use Case Diagram

4 Actors

TODO

5 Use Cases

TODO

5.1 Overview

Use Case 1

ID
UC-MWA-001

Name
Money Withdrawal.

Goal
The user withdraws an amount of money from the selected account.

Actors
Primary Actor - owner of the account.

Precondition
User is the owner of the account.

Main Scenario

1. Primary actor indicates to withdraw an amount from a selected account.
2. System verifies the account exists.
3. System prompts primary actor for the amount to withdraw.
4. User enters amount to withdraw.
5. System verifies that the account contains sufficient funds.
6. System prompts user for confirmation.
7. User confirms.
8. System subtracts amount of money requested from selected account, confirms withdrawal.

9. Use case ends succesfully.

Exceptions

1. Account does not exist.
2. Account has an empty of negative balance.
3. Account contains insufficient funds.

Postcondition
The amount is subtracted from the selected account.

Priority
Critical.

Traces to Test Cases
Add when test cases done.

Use Case 2

ID
UC - MDA - 002

Name
Deposit Money.

Goal
User succesfully deposits an amount of money the selected account.

Actors
Primary Actor - owner of the account.

Precondition
User is owner of the account.

Main Scenario

1. Primary actor indicates to deposit an amount to a selected account.
2. System verifies that the account exists.
3. System prompts user to enter amount to deposit.
4. System adds amount of money deposited to selected account and confirms.
5. Use case ends succesfully.

Exceptions

1. Account does not exist.

2. Amount is zero or negative.

Postcondition
The amount is added to the selected account.

Priority
Critical.

Traces to Test Cases
Add when test cases done.

Use Case 3

ID
UC - DBA - 003

Name
Display Balance.

Goal
Display balance of chosen account to user.

Actors
Primary Actor - Owner of the account

Precondition
Account exists.

Main Scenario

1. Primary Actor selects account to be displayed.
2. System displays balance.

Exceptions

1. Account does not exist.
2. System cannot retrieve data for chosen account.

Postcondition
Account balance is displayed for user.

Priority
Critical.

Traces to Test Cases
Add when test cases done.

6 Non-Functional Constraints

7 Data Dictionary

8 References

A Description of File Format: Tasks

Describe input file format.

B Description of File Format: Persons

Describe output file format.