

User Requirements Document

Iteration 1

Team PA-PI-a

11 February 2018

Table 1: Team

Name	ID Number
Melanie Taing	40009850
Laurie Gagnon	22943433
Wayne Yiel Leung	26586988
Jordan Rutty	27300107
Alice Barkhouse	27486782
Michael Foo	40000225
Pierre-Andre Leger	40004010
Colin Greczkowski	40001600

Contents

1	System	3
1.1	Purpose	3
1.2	Context	3
1.3	Business Goals	3
2	Domain Concepts	4
2.1	Overview	4
3	Actors	4
4	Use Cases	5
4.1	Overview	5
4.2	Use Case 1	6
4.3	Use Case 2	7
4.4	Use Case 3	8
4.5	Use Case 4	9
4.6	Use Case 5	10
4.7	Use Case 6	11
5	Non-Functional Constraints	12
6	Glossary	12
7	References	13

1 System

1.1 Purpose

The purpose of this document is to define requirements and introduce the myMoney System application, as part of the partial requirements for COMP 354. This document will describe the myMoney System domain concepts, actors, use cases and constraints.

1.2 Context

The myMoney System is a desktop application developed to help a person manage their finances. It allows individuals to supervise and organize their money.

1.3 Business Goals

The myMoney System is meant to facilitate the coordination all of a user's financial transactions into one simple application. It is intended for students and early career professionals who manage and oversee all aspects of their budgets.

Clients will be able to log and monitor their daily transactions. They will be able to create different types of accounts, add recent transactions and generate monthly reports from past transactions. Additionally, users will be able to import spreadsheets from their banks, and add and categorize budgets.

2 Domain Concepts

2.1 Overview

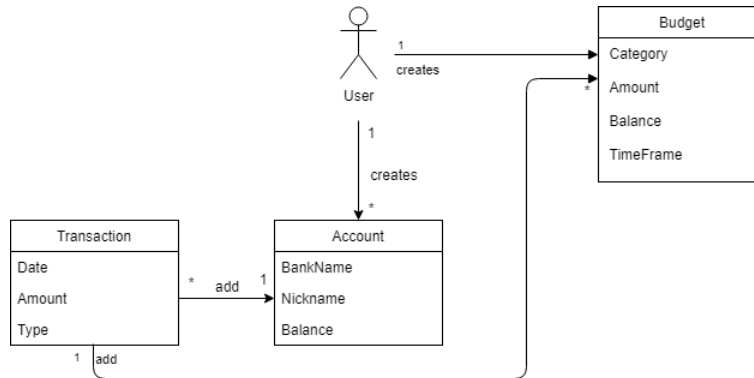


Figure 1: Domain concepts

A user creates a new account on the application, modelling a personal account linked to a financial institution. The user can add transactions to the account. The user creates a budget, selecting from a list of predefined categories.

3 Actors

The individual user is the sole actor interacting with the myMoney application. It is meant to be used only by the user.

4 Use Cases

4.1 Overview

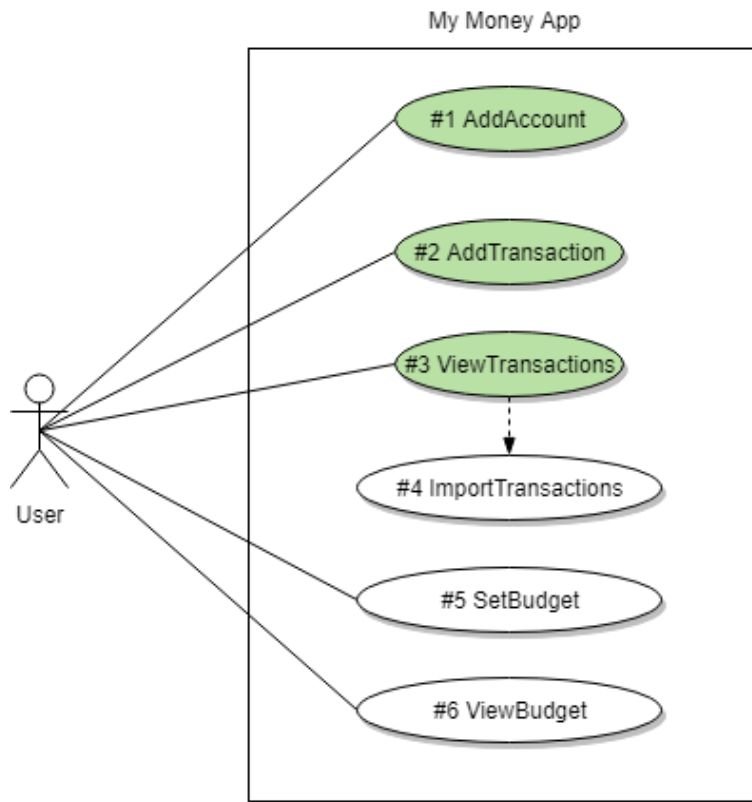


Figure 2: Use Case Diagram

4.2 Use Case 1

Table 2: Use Case 1

Use Case ID:	1
Use Case Name:	AddAccount
Priority:	High
Goal/Purpose:	Add a bank account to the application.
Actors:	User
Pre-conditions:	App must be running.
Main Flow:	<ol style="list-style-type: none">1. The user selects the option to add a new account.2. The user selects an account type.3. The user enters account details.4. The user confirms the addition of a new account.5. The system adds a new account associated with the user in the application.
Exception Flow(s):	The required fields have not been entered or the details entered are invalid.
Post-Conditions:	There is a newly added bank account associated with the user with the specified corresponding details.
Tests/Traces:	

4.3 Use Case 2

Table 3: Use Case 2

Use Case ID:	2
Use Case Name:	AddTransaction
Priority:	High
Goal/Purpose:	Add a transaction to a bank account.
Actors:	User
Pre-conditions:	There must be at least one bank account which allows transactions associated with the user registered on the application.
Main Flow:	<ol style="list-style-type: none">1. The user selects the desired bank account.2. The user selects the option to add a transaction.3. The user enters the type of transaction, the date and the amount in the provided form.4. The user selects if they want the transaction to be included in a budget.5. The user confirms the transaction.6. The system adds the new transaction to an existing account and/or an existing budget.
Exception Flow(s):	The required fields have not been entered or the details are invalid.
Post-Conditions:	There is newly added transaction on the application associated with the account and corresponding details the user has entered.
Tests/Traces:	

4.4 Use Case 3

Table 4: Use Case 3

Use Case ID:	3
Name:	ViewTransaction
Priority	High
Goal/Purpose:	The user requests a report of their monthly transactions of their accounts.
Actors:	Any user
Pre-conditions:	The user must be logged into their account.
Main Flow:	<ol style="list-style-type: none">1. The user requests the report.2. The system displays the report, which includes account, bank name and amount information for each transaction during the month.
Exception Flow(s):	-
Post-Conditions:	The user has access to their monthly report.
Tests/Traces	

4.5 Use Case 4

Table 5: Use Case 4

Use Case ID:	4
Name:	ImportTransaction
Priority	Medium
Goal/Purpose:	The user imports a spreadsheet of transactions into an account.
Actors:	Any user
Pre-conditions:	The user must be logged into their account and have an account to import transactions into.
Main Flow:	<ol style="list-style-type: none">1. The user selects the account they wish to import transactions into.2. The user selects the option to import transactions.3. The user selects the file they wish to import.4. The system adds the imported transactions to the selected account.
Exception Flow(s):	The file the user is trying to import into the account is missing, corrupted or contains invalid information.
Post-Conditions:	The users account is updated with the information imported from the file.
Tests/Traces	

4.6 Use Case 5

Table 6: Use Case 5

Use Case ID:	5
Name:	SetBudget
Priority	Medium
Goal/Purpose:	The user sets a budget for a category selected from a pre-defined list
Actors:	Any user
Pre-conditions:	The user must be logged into their account.
Main Flow:	<ol style="list-style-type: none">1. The user selects the option to set a new budget.2. The user selects a Budget Category from pre-defined list.3. The user enters desired amount for selected budget category.4. The user enters duration of selected budget.5. The user selects the option to confirm.6. The system adds the newly created budget to the application.
Exception Flow(s):	The information entered by the user is invalid.
Post-Conditions:	The users account is updated with the newly created budget for a specified category.
Tests/Traces	

4.7 Use Case 6

Table 7: Use Case 6

Use Case ID:	6
Name:	ViewBudget
Priority	Medium
Goal/Purpose:	View current budget transactions affecting the budget for a certain category.
Actors:	Any user
Pre-conditions:	The user must be logged into their account and have at least one budget created.
Main Flow:	<ol style="list-style-type: none">1. The user selects the budget category they wish to view2. The user selects the option to confirm.3. The system displays the current budget breakdown, which includes transactions and remaining balance for the category.
Exception Flow(s):	No transactions have occurred.
Post-Conditions:	The transactions and the remaining balance to spend of a chosen category are displayed.
Tests/Traces	

5 Non-Functional Constraints

- Accuracy
 - All digits and decimal points must be accurate.
 - All mathematical operations must be accurate.
- Operating constraints
 - Must run on all operating systems.
- Usability
 - The interface must be easy to use and understand.
 - Only one user at a time may operate the application.

6 Glossary

Table 8: Glossary

Term	Definition
Account	The users profile that other sub-categories are associated with.
Amount	A sum of money, in dollars.
Balance	Remainder of money left to spent on a planned budget amount.
Bank account	The in-app representation of a users bank account.
Budget	The preferred amount the user can spend during a specified time frame.
Budget Category	The type of budget, ie groceries or transportation.
Chequing	The in-app representation of a users Chequing Account.
Credit	The in-app representation of a users Credit.
Invalid	The input is not accepted by the app.
Money	A general term for currency in different denominations.
Payment	A debit to a specified account.
Savings	The in-app representation of a users Savings Account.
Time Frame	An amount of time in days.
Transaction	The various functions, such as payment or deposit, applied to an account.
Transaction Type	The type of transaction, ie cash or credit card.

7 References

Argawal, S. "Domain Modelling" (Current February 5, 2018)

Argawal, S. "Use Cases" (Current February 5, 2018)

Dalbey, J. "Non-function Requirements" (Current February 5, 2018)

Sinning, D. "COMP 354: Introduction to Software Engineering - Introduction to UML" (Current February 5, 2018)

Larman, C. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd edition, Prentice-Hall, 2005.