Requirements Document

Team PA-PK

17 January 2012

Table 1: Team

Name	ID Number
Anne-Laure Ehresmann	27858906

Contents

List of Figures

List of Tables

1 System

1.1 Purpose

The purpose of this document is to define requirements for the desktop application myMoney. There exists a plethora of software for money management, each greatly varying in design due to the complex and multifarious clientele. This document may thus be to orient the development of the application. It may be used by:

Table 2: Document Users

Users and customers	To give feedback about the requirements.
System developers	To understand what functions and properties the system must contain.
Testers	To test the system against the requirements.
Writers of user manuals	To get material for user manuals.
Project team	To follow-up the status of the project against the requirements.

1.2 Business Goals

2 Domain Concepts

3 Actors

4 Use Cases

4.1 Overview

Figure 1: Use Case Diagram

Table 3: Use Case 1 - Create user account

Action	**Create user account**
Case ID	02
Summary	User gives information about a new user account, system validates it and creates the account.
Scope	money and budget management application
Level	user-goal
Actors	**User**
Stakeholders and Interests	1. User: Wants fast and easy account creation, clear and comprehensible display, proof of successful account creation. 2. Company: Wants user interests to be fulfilled, wants to prevent erroneous input, wants fast communication with the local accountdatabase as well as fault tolerance in case of database conflicts, issues with editing authorisation, or other possible database problems.
Pre-Conditions	User has opened the application and is in the startup menu.
Success Guarantee	Account successfully saved in local account database, with name and password as specified by the user.
Main Success Flow	1. User enters a username and password. 2. System validates username and password (format, whether username is already used, etc). 3. System creates new account. 4. System notifies the user of the successful account creation, then returns to home menu.
Exceptions	
PostConditions	
Priority	
Traces to Test Cases	

Use Case 2

5 Non-Functional Constraints

6 Data Dictionary

7 References

A Description of File Format: Tasks

Describe input file format.

Table 4: Use Case 2 - Delete user account

Action	**Delete user account**
Case ID	02
Summary	User deletes a user account from the local accounts database, removing all bank accounts information associated with that account as well.
Scope	money and budget management application
Level	user-goal
Actors	**User**
Stakeholders and Interests	1. User: Wants easy navigation and secure account deletion, no risk of accidental deletion, proof of successful account deletion. 2. Company: Wants user interests to be fulfilled, wants to ensure clean deletion from the database.
Pre-Conditions	User has logged in the user account he wants to delete.
Success Guarantee	user account is successfully deleted, all associated bank information is deleted, and user is returned to the home menu.
Main Success Flow	1. User enters the account settings, then selects 'delete account'. 2. System brings up a confirmation menu to ensure that this selection was no accidentally entered. 3. User affirms his choice. 4. System successfully deletes all relevant entries in the local database, then notifies the user of this successful deletion. 5. User confirms having read this notification. 6. System brings the user back to the home menu.
Exceptions	
PostConditions	
Priority	
Traces to Test Cases	

B Description of File Format: Persons

Describe output file format.

Table 5: Use Case 3 - Add bank account to a user account

Action	**Add bank account to a user account**
Case ID	03
Summary	User gives information about a new bank account, system sends it to the bank for verification then creates necessary entries in the local database once the bank approves the information.
Scope	money and budget management application
Level	user-goal
Actors	**User**, Bank
Stakeholders and Interests	1. User: Wants fast and easy account creation, clear and comprehensible display, proof of successful account creation. 2. Company: Wants user interests to be fulfilled, wants to prevent erroneous input, wants fast communication with the local account database as well as the bank, wants fault tolerance in case of database conflicts, issues with the bank, or other possible local database problems. 3. Bank: Wants to satisfy its customer base, wants correctly formatted account information given to its API, inexpensive and non-redundant communication of bank account data to third-party applications.
Pre-Conditions	User has logged in a user account and is in the user home menu.
Success Guarantee	Bank account successfully saved in local account database, with information corresponding the data validated by thebank.
Main Success Flow	1. User enters his bank name, bank account number, card expiry date, and card security code. 2. System validates input, and sends it to the bank to be verified and connected. 3. Bank validates bank account details. 4. System records the valid bank account details securely in its database, and notifies the user of the successful addition of the bank account in the database.
Exceptions	
PostConditions	
Priority	
Traces to Test Cases	

Table 6: Use Case 4 - Remove bank account from a user account

Action	**Remove bank account from a user account**
Case ID	04
Summary	User deletes a bank account from the local database.
Scope	money and budget management application
Level	user-goal
Actors	**User**
Stakeholders and Interests	1. User: Wants easy navigation and secure account deletion, no risk of accidental deletion, proof of successful account deletion. 2. Company: Wants user interests to be fulfilled, wants to ensure clean deletion from the database.
Pre-Conditions	User has logged in the user account whose active association with a bank account is the one the user wants to delete.
Success Guarantee	bank account is successfully removed from that user account, all associated bank information is deleted from the local database, and user is returned to the account home menu.
Main Success Flow	1. User selects the account he wants to remove, then selects 'remove account'. 2. System brings up a confirmation menu to ensure that this selection was no accidentally entered. 3. User affirms his choice. 4. System successfully deletes all relevant entries in the local database, then notifies the user of this successful deletion. 5. User confirms having read this notification. 6. System brings the user back to the home account menu.
Exceptions	
PostConditions	
Priority	
Traces to Test Cases	

Table 7: Use Case 5 - View Transaction for Specific Bank Account

Action	**View Transaction for Specific Bank Account**
Case ID	05
Summary	User selects specific bank account and views transactions associated with with selected bank account
Scope	money and budget management application
Level	user-goal
Actors	**User**
Stakeholders and Interests	1. User: Wants quick and convenient viewing of previous transactions from one specific bank account 2. Company: Wants to give user ability to micromanage every aspect of the application down to each bank account and transaction.
Pre-Conditions	User has added one or more bank accounts to his myMoney account.
Success Guarantee	User can view transaction by bank account.
Main Success Flow	1. User selects specific bank account from list of all accounts. 2. System displays all previous transactions under under specified bank account. 3. User selects desired transaction. 4. System shows all information about desired transaction, such as date and amount withdrawn, deposited, or transferred.
Exceptions	
PostConditions	
Priority	
Traces to Test Cases	

Table 8: Use Case 6 - View All Transactions

Action	**View All Transactions**
Case ID	06
Summary	User can view all transactions that have been made from all bank accounts
Scope	money and budget management application
Level	user-goal
Actors	**User**
Stakeholders and Interests	1. User: Wants easy and convenient viewing of all transactions among all bank accounts. 2. Company: Wants user interests to be fulfilled.
Pre-Conditions	Account has been created and one or more bank accounts have been added to the system.
Success Guarantee	User is able to conveniently view all transactions from all institutions in one display.
Main Success Flow	1. User selects View All Transactions option. 2. System shows all transactions across all accounts on one display. 3. User can select desired transaction in the list. 4. System shows all information regarding selected transaction such as date and amount.
Exceptions	
PostConditions	
Priority	
Traces to Test Cases	

Table 9: Use Case 7 - View Transactions by Category

Action	**View Transactions by Category**
Case ID	08
Summary	User can view transaction according to categories of spending.
Scope	money and budget management application
Level	user-goal
Actors	**User**
Stakeholders and Interests	1. User: Wants to view transactions in different categories of spending such as rent, bills, leisure, etc., to view and monitor spending habits and allocate budget to each category of spending. 2. Company: Wants to optimize the ease in which a user can allocate his/her budget, and track spending habits.
Pre-Conditions	User has made transactions and labeled the transaction according to which category it belongs.
Success Guarantee	User can easily categorize and view transactions.
Main Success Flow	1. User selects View All Transactions . 2. System shows all transactions across all accounts on one display. 3. User selects to view transaction by categories. 4. System groups transactions by similar category. 5. System displays transactions grouped by category.
Exceptions	
PostConditions	
Priority	
Traces to Test Cases	