Software Requirements Specification

for

Personal Budget Manager Application

Version 0.1

Prepared by

|  |  |  |
| --- | --- | --- |
| Name | Student ID | e-mail |
| Danny Shash |  | danny.shash@gmail.com |
| Ganji Amarendher | 25764246 | a\_ganji@encs.concordia.ca |
| Siming Huang |  | hsim47@163.com |
| Tony Lac |  | tony.lac@polymtl.ca |
| Xinjie Zeng |  | xinjiezeng@gmail.com |

|  |  |
| --- | --- |
| Instructor: | Dr. Nora Houari |
| Course: | COMP 5541 [2184 DD](https://moodle.concordia.ca/moodle/course/view.php?id=112064) |
|  |  |
|  |  |
|  |  |

Contents

Contents ii

Revisions iii

1 Introduction 1

1.1 Document Purpose 1

1.2 Product Scope 1

1.3 Intended Audience and Document Overview 1

1.4 Definitions, Acronyms and Abbreviations 1

1.5 Document Conventions 2

1.6 References and Acknowledgments 2

2 Overall Description 3

2.1 Product Overview 3

~~2.2~~ Product Functionality 3

2.3 Design and Implementation Constraints 3

2.4 Assumptions and Dependencies 3

3 Specific Requirements 4

3.1 External Interface Requirements 4

3.2 Functional Requirements 4

3.3 Use Case Model 4

3.4 Class Diagrams 7

4 Other Non-functional Requirements 7

4.1 Performance Requirements 7

4.2 Safety and Security Requirements 7

4.3 Software Quality Attributes 7

5 Other Requirements 8

Appendix A – Data Dictionary 8

Appendix B - Group Log 9

Revisions

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| Draft Type and Number | Full Name | Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded. | 00/00/00 |

# 

# Introduction

*<TO DO: Please provide a brief introduction to your project and a brief overview of what the reader will find in this section.>*

The purpose of this document is to collect, analyze, and define the high-level needs and features of Personal Budget Manager Application (PBM), an application to help individuals control their expenses.

The details of how PBM fulfils these needs are detailed in the use case and more

details will be available in the upcoming design phase.

## Document Purpose

This Software Requirements Document (SRD) describes the specification of the PBM, which is in partial fulfillment of the requirements of COMP 5541. It will define the high-level requirements of the PBM system, it’s product functions, user cases, analysis model, assumptions and dependencies. The analysis model will include use case diagrams, class diagrams and sequence diagrams.

## Product Scope

This document addresses the high-level requirements of PBM that will be used as a reference for the implementation of the system. The use case diagrams explain the interaction of the user, any individual and the result from it. The goal of the system is to provide an overview of the expenses, that are day-to-day or recurring and manage them in an efficient manner.

The development plan is divided into 3 iterations and the document will be updated accordingly to address the content in that phase.

## Intended Audience and Document Overview

## Definitions, Acronyms and Abbreviations

### Acronyms

PBM Personal Budget Manager Application

SRS Software Requirement Specification

### Definitions

Purchase A type of day-to-day expense

Bill A type of recurring expense

## Document Conventions

## References and Acknowledgments

# Overall Description

## Product Overview

PBM is to help individuals control their expenses.

## Product Functionality

PBM will have the following functionality.

* View all expenses of the spending list (in arbitrary order).
* Add a new expense.
* Update an existing expense (Mark an expense as paid or not paid).
* Remove an expense.
* The expense table should refresh its display automatically when the expenses in the expense list change.

## Design and Implementation Constraints

PBM is a desktop application implemented using Java Swing as user interface.

## Assumptions and Dependencies

# Specific Requirements

## External Interface Requirements

### User Interfaces

### Hardware Interfaces

### Software Interfaces

## Functional Requirements

## Use Case Model

The following diagram shows the main use case of PBM application.

<insert MainUC>

### *Use Case 1: Launch the PBM application*

**Description:** User open the PBM application.

**Actor(s):** User

**Goal:** User able to launch PBM application by clicking an icon on the desktop.

**Preconditions:** PBM is installed on the system. Users allowed to access the application exists in the system.

**Main scenario:**

1. user enters the name and password

**Alternative scenarios:**

1. User can't remember the password.

1.1. User recovers the password.

1.2. User enters the name and password.

2. User doesn't exist

2.1 User registers to the system

2.2 User enters the name and password

**Post-conditions:** PBM application is open. The list of all the purchase expenses displayed on the main window for the current month.

### Use Case 2: set the type of currency

**Description:** User sets the type of currency

**Actor(s):** User

**Goal:** User able to set the type of currency

**Preconditions:** User already logged in to the system

**Main scenario:**

1. After logging in to the system, there is a page to select currency

2. User selects the type of currency

3. User clicks the "save" button

4. Information is sent and stored in database

### Use Case 3: Add a new expense

**Description:** User adds a new expense

**Actor(s):** User

**Goal:** User able to add 2 types of expenses

**Preconditions:** User already logged in to the system

**Main scenario:**

1. User chooses to add a new exprense

2. User prompted to choose the expense types

3. User enters the details for the selected expense types

4. System validates the data

5. System add the expense into the list

6. System displays the main window with the list of existing expenses for the type just entered

**Alternative scenarios:**

**Post-conditions:** The list of all the expenses displayed on the main window for the selected time period.

### Use Case 4: Modify an existing expense

**Description:** User can modify an existing expense

**Actor(s):** User

**Goal:** user is able to modify an expense type belogs to a certain period

**Preconditions:** User already logged in to the system.

**Main scenario:**

1. User selects a row from the list and modifies allowed details, like paid, amount, any additional details, etc...

2. System modifies the expense entry

3. System refreshes the main window with existing expense list

**Alternative scenarios:**

1. User changes the expense type for the displayed period

1.1 System fetched the expense list for the period chosen by the user

1.2 System displays the expense list in the main window

Same steps as in the main flow.

2. User changes the time period for the displayed expense type

2.1 System fetched the expense list for the period chosen by the user

2.2 System displays the expense list int he main window

Same steps as in the main flow.

3. User changes the expense type and the time period

3.1 System fetched the expense list for the period chosen by the user

3.2 System displays the expense list int he main window

Same steps as in the main flow.

**Post-conditions:** The list of all the expenses displayed on the main window for the selected time period.

### Use Case 5: Remove an expense

**Description**: User chooses to remove an expense

**Actor**(s): User

**Goal**: User able to remove an entry in the selected expense list

**Preconditions:** User already logged in to the system.

**Main scenario:**

1. User selects a row in the list to be removed

2. system removes the expense entry

3. System refreshes the main window with existing expense list

**Alternative scenarios:**

**Post-conditions:**

### Use Case 6: User able to change the display

**Description:** User able to change the display of expense report

**Actor(s):** User

**Goal:** User able to see the list of expenses for the type selected for that period.

**Preconditions:** User logged in to the system

**Main scenario:**

1. User selects the expense type and the time period

2. System fetched the relevant data

3. System displas the data

**Alternative scenarios:**

**Post-conditions:**

### Use Case 7: Add income

**Description:** User add income

**Actor(s):** User

**Goal:** User able to add income

**Preconditions:** User already logged in to the system

**Main scenario:**

1. User chooses to add income

2. User enters the details of account, category and amount

3. User clicks the “save” button

4. Information is sent and stored in database

**Alternative scenarios:**

**Post-conditions:**

## Class Diagrams

### Design Pattern

PBM is implemented using MVC architecture.

<insert PBM\_MVC>

### Interaction diagrams

The main classes and their relations are defined as below.

<insert MainCD>

# Other Non-functional Requirements

## Performance Requirements

## Safety and Security Requirements

## Software Quality Attributes

# Other Requirements

Appendix A – Data Dictionary

Appendix B - Group Log

<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>