

Prj_4_FamilyBudget Requirements Specification

Version 1.0

01 March, 2017

Developed by:

Pop Ioana Andreea

936, 2017

Version History

Version	Description of Change	Author	Date
V_01	Modification of the Introduction section	Pop Ioana Andreea	27 February, 2017
V_02	Modification of the entire document	Pop Ioana Andreea	19 March, 2017

Contents

Prj_4_FamilyBudget Requirements Specification	1
Version 1.0	1
01 March, 2017	1
1. Introduction.....	3
1.1. Purpose.....	3
1.2. Scope	3
1.3. Definitions, Acronyms, and Abbreviations.....	3
1.4. Document Overview.....	3
2. Product/Service Description.....	3
2.1. Product Context.....	3
2.2. User Characteristics.....	3
3. Requirements	4
3.1. Functional Requirements	4
3.2. User Interface Requirements	4
3.3. Usability	4
3.4. Data Management.....	5
4. User Scenarios/Use Cases	5

1. Introduction

The system is designed to help a community of members with the management and planning of the community's budget.

1.1.Purpose

The system's overall purpose is to keep track of all the members in the community and all their contributions to the community's budget.

1.2. Scope

The document is an useful outline of the requirements asked by the client.

1.3. Definitions, Acronyms, and Abbreviations

The document contains the following terms:

- **community:** collective of members that manages its budget
- **member:** a person that contributes monthly to the community's budget
- **entry:** information stored about incomes or costs added to the budget

1.4. Document Overview

The document is short and briefly presents the main requirements of the system. Although not all requirements are complete and unambiguous, the document gives a good overview of the requirements.

2. Product/Service Description

This product is designed to ease the process of budget management and planning for a community of people. It provides 3 basic features such as adding a new member to the community along with information regarding the income he/she will contribute monthly to the community's budget, adding a new income or cost at the budget management and viewing the evolution of incomes and costs for a member in the community so that it is easier to keep track of the budget changes. All of these actions can only be performed by a member of the community, this securing the data stored about the budget.

2.1.Product Context

This is a standalone product that is not dependent on any other existing products.

2.2.User Characteristics

For now the product is designed for the **general user** with the following profile:

- Family member
- Average computing skills
- Requires quick access to information
- Little to no experience with budget management

3. Requirements

3.1. Functional Requirements

List the functional requirements (FR) of the system.

Section/ Requirement ID	Requirement Definition
FR_1.0.	The system shall allow the user to add a new member to the community.
FR_1.1	The system shall request information about the new member from to user, specifically the name.
FR_1.1.1	The system shall validate the input data provided by the user: the name should be a string of alphabetic characters.
FR_1.2	The system shall inform the user if the member was added successfully or not.
FR_2.0	The system shall allow the user to add a new entry at the budget management.
FR_2.1	The system shall request information about the new budget entry from the user, specifically the type of the entry, the value of the entry and the member id.
FR_2.1.1	The system shall validate the input data provided by the user: the type of the entry should be income/cost, the value of the entry should be a positive number and the member id should be an integer.
FR_2.2	The system shall inform the user if the entry was added successfully or not.
FR_3.0	The system shall allow the user to view the evolution of incomes/costs for an existing member.
FR_3.1	The system shall request information from to user about the member whose incomes and costs should be listed, specifically the member id.
FR_3.1.1	The system shall validate the input data provided by the user: the member id should be an integer.
FR_3.2	The system shall inform the user if there are no entries to be shown.

3.2. User Interface Requirements

The system should provide a friendly menu of options the user can choose from, clear written requests when user input data is required and it should inform the user about the error that have been encountered while processing their request.

3.3. Usability

Learnability:

- The features' names should be suggestive
- The error messages should be clear and unambiguous
- The system should be easy to learn

Memorability:

- The user interface should be very simple and straightforward

3.4.Data Management

The data persistence is done by saving all members and budget entries to their corresponding files.

The members are saved in the “membersF.txt” file as follows:

- each line represents a member with its data
- all member data is separated by a “;”
- each different member is separated by a new line
- the data on a line is written as: name of the member ; id of the member

The budget entries are saved in the “budgetF.txt” file as follows:

- each line represents an entry with its data
- all entry data is separated by a “;”
- each different entry is separated by a new line
- the data on a line is written as: type of entry ; value of entry ; id of the member that added the entry

4. User Scenarios/Use Cases

The 3 major Use Scenarios are:

- User adds a new member
 - Enters the add a new member option
 - Inputs data regarding the new member (name and id)
 - Receives an error message if the input data is invalid
 - Receives an error if the id already exists
 - Otherwise the user is notified user that the action was completed successfully
- User adds a new entry
 - Enters the add a new entry option
 - Inputs the data regarding the new entry (type, value and member id)
 - Receives an error message if the input data is invalid
 - Receives an error if the given type does not exist
 - Receives and error message if the member id does not exist
 - Otherwise the user is notified user that the action was completed successfully
- User visualizes the evolution of incomes and costs for a member
 - Enters the view all entries for a member option
 - Inputs the member id
 - Receives an error message if the input data is invalid
 - Receives and error message if the member id does not exist

- Otherwise the program will print to the user all budget entries for the given member