Software Requirements Specification Template

CptS 322—Software Engineering

The following annotated template shall be used to complete the Software Requirements Specification (SRS) assignment of WSU-TC CptS 322. The instructor must approve any modifications to the overall structure of this document.

Template Usage:
Text contained within angle brackets ('<', '>') shall be replaced by your project-specific information and/or details. For example, <Project Name> will be replaced with either 'Smart Home' or 'Sensor Network'.

Italicized text is included to briefly annotate the purpose of each section within this template. This text should not appear in the final version of your submitted SRS.

This cover page is not a part of the final template and should be removed before your SRS is submitted.

Acknowledgements:
Sections of this document are based upon the IEEE Guide to Software Requirements Specification (ANSI/IEEE Std. 830-1984). The SRS templates of Dr. Orest Pilskalns (WSU, Vancover) and Jack Hagemeister (WSU, Pullman) have also be used as guides in developing this template for the WSU-TC Spring 2005 CptS 322 course.

Application for Managing Family Budget

Software Requirements Specification

1.0

10.9.2015

Joonatan Heiskanen
Lead Software Engineer

Prepared for
WSU-TC CptS 322—Software Engineering Principles I
Instructor: Saleh Hadi, Ph.D.
2015

Revision History

Date	Description	Author	Comments
22.09.2015	Version 1.0	Joonatan Heiskanen	First revision of the document

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Joonatan Heiskanen	Lead Software Eng.	
	Ph.D. Hadi Saleh	Instructor, CptS 322	

Table of Contents

1. Introduction

1.1 Purpose

This document is written to the developers of the application, to help them in developing this product. Some parts can also be useful to the users, but they are not the target audience.

1.2 Scope

This software will be a bookkeeping system for anybody who needs help with balancing their budget. The system will be designed to provide easy-to-access information on the budget of the current user, and in this way help with planning purchases and spending behavior.

1.3 Definitions, Acronyms, and Abbreviations

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.

1.4 References

This subsection should:
(1) Provide a complete list of all documents referenced elsewhere in the SRS, or in a separate, specified document.
(2) Identify each document by title, report number - if applicable - date, and publishing organization.
(3) Specify the sources from which the references can be obtained.
This information may be provided by reference to an appendix or to another document.

1.5 Overview

The remainder of this SRS document contains the description of the application and the functions it will be able to perform. In chapter 2 the general information about the application is given, such as a general description and some basic functionalities. In chapter 3 the specific requirements for the application are given.

2. General Description

The application will have two kinds of users: master users and normal users. Different user types will be able to perform different tasks in the program. The product will allow the users to add and subtract their remaining money from their total budget, as well as see statistics of their spending behavior in the past. Also the master user can create a list of allowed items, that only these items can be purchased.

2.1 Product Perspective

There are currently no other products or projects that would affect the production of this application.

2.2 Product Functions

The application will allow the master user to provide a starting sum, the starting budget.
The application will allow a normal user to input a purchase event into the system. Purchase event will contain name of person who spent money, what the money was spent on, how much was spent, and date of purchase.
The application will allow for the master user to create statistics from past purchase events
The application will allow the master user to create and delete user profiles.

2.3 User Characteristics

The users of this application will be very varied in knowledge and background. The application will be implemented in a way that any user familiar with using a graphical user interface will be able to use this application.

2.4 General Constraints

The application will be usable by a combination of mouse and keyboard. Other input methods will not be supported in development.

2.5 Assumptions and Dependencies

The software is being developed for current hardware with currently available tools, so no assumptions or dependencies are made.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface will be a graphical interface, navigated by a combination of mouse and keyboard. Main navigation will be done with the mouse, and writing inputs will be done with the keyboard.

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.2 Functional Requirements

3.2.1 Adding and deleting users

3.2.1.1 Introduction

The software has two levels of users, and it is possible to create new users and delete old ones

3.2.1.2 Inputs

Name, level. No inputs in deleting

3.2.1.3 Processing

The input data will be saved for future use. In deleting a user, the user is chosen from a list, and a button is pressed. After confirmation, the selected user is deleted.

3.2.1.4 Outputs

Log event, message for successful or unsuccesful add/delete

3.2.1.5 Error Handling

If the master user tries to delete the last master user, an error message will be given and the user won't be deleted.

3.2.2 Setting a budget

3.2.2.1 Introduction

The master user can set the starting budget, or modify the amount at any point. These modifications are recorded into a log.

3.2.2.2 Inputs

Amount to which the budget will be set

3.2.2.3 Processing

User inputs the new amount, presses a button, and the new amount will be displayed as the new budget to all users.

3.2.2.4 Outputs

Log event, message for successful modification of the total sum.

3.2.3 Purchase events

3.2.3.1 Introduction

Purchase event is when a user buys an item, and inputs it into the system

3.2.3.2 Inputs

User who made purchase, name of purchased item, price of purchased item, date of purchase

3.2.3.3 Processing
A log event is created from the inputs. Balance is deducted according to the price of purchased item

3.2.3.4 Outputs
Log event

3.2.5 Log events and statistics

3.2.5.1 Introduction
All users can access the spending statistics. Here they can see on a graph how they have been spending money on a given interval

3.2.5.2 Inputs
Time interval to be inspected

3.2.5.3 Processing
The application will retrieve the amount of money left on each day in the given interval, and plot the values on a graph. This graph will be displayed to the user.

3.2.5.4 Outputs
Graph of the spending behavior of all users in the given time interval

3.3 Use Cases

3.3.1 User creates a profile in the application

a) User opens the program.
b) User clicks “Master user options” button in the main menu.
c) Application prompts the user to input his login information, meaning name and password. If there are no created profiles, no login prompt is shown.
d) User clicks “Create new profile” button in the options.
e) Application prompts the user to input the name of the new profile and the level of the user (master or normal).
f) User inputs the required information and clicks the “Create profile” button
g) Profile is created and saved

3.3.2 Normal user inputs a purchase event

a) User opens the application.
b) User clicks the “Input purchase” button, and an input window opens.
c) User inputs the name of the purchase, price, and who made the purchase into the appropriate input boxes
d) Message is displayed with info whether operation was successful or unsuccessful, and the main menu is displayed.

3.3.1 User creates a budget

a) User opens the application
b) User clicks the “Master user options” button
c) The application prompts for login name and password
d) After correct login information is given, user clicks the “Add budget” button
e) The application prompts the user for name of budget and the initial sum.
f) After clicking “Add budget” button the application saves the new budget

3.4 Classes / Objects

3.4.1 Budget

3.4.1.1 Functions
addBudget
removeBudget
changeAmount
addProduct
addUsertoBudget
printBudget
createLog

3.4.1.2 Attributes
amount
startDate
productList

3.4.2 User

3.4.2.1 Functions
addUser
removeUser

3.4.2.2 Attributes
name
level

3.4.3 Purchase event

3.4.3.1 Functions
addPurchase
removePurchase

3.4.3.2 Attributes
productName
productPrice
date
purchaser

3.5 Non-Functional Requirements

3.5.1 Performance

No user action should take more than one second to complete

3.5.2 Reliability

The program should not crash with any given input

3.5.3 Availability

3.5.4 Security

The master user functions will be protected by a password chosen by the user

3.5.5 Maintainability

3.5.6 Portability

The program will run on any machine running the appropriate version of Java virtual machine.

3.6 Inverse Requirements

No inverse requirements

3.7 Design Constraints

No design constraints at the current moment

3.8 Logical Database Requirements

The program will store data locally, into a data structure implemented into the program. No external databases are needed.

3.9 Other Requirements

None so far

4. Analysis Models

4.1 Sequence Diagrams

4.3 Data Flow Diagrams (DFD)

4.2 State-Transition Diagrams (STD)

5. Change Management Process

The lead designer is the only person who can submit changes to the document. Changes are added into the document, and a marking is made that this is now a new version. If necessary, all changes will be logged into a separate text file.

A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS's overall set of requirements.

Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.

A.1 Appendix 1

A.2 Appendix 2