

Software Requirements Specification

for

Expense Tracker

Version 1.0

Prepared by

Group Name: <*place your group name here*>

|  |  |  |
| --- | --- | --- |
| Chayse Jorgenson | 11678623 | [Chayse.jorgenson@wsu.edu](mailto:Chayse.jorgenson@wsu.edu) |
| Kevin Tong | <11686700> | [Kevin.tong@wsu.edu](mailto:Kevin.tong@wsu.edu) |
| Vitaliy Kudrik | 11603449 | [Vitaliy.kudrik@wsu.edu](mailto:Vitaliy.kudrik@wsu.edu) |

|  |  |
| --- | --- |
|  |  |
| Date: | 10/25/2019 |
|  |  |
|  |  |
|  |  |

Contents

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 1

1.6 References and Acknowledgments 2

2 Overall Description 3

2.1 Product Perspective 3

2.2 Product Functionality 3

2.3 Users and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3 Specific Requirements 5

3.1 External Interface Requirements 5

3.2 Functional Requirements 6

3.3 Behaviour Requirements 6

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 9

Appendix B - Group Log 10

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 |

# 

# *<In this template you will find text bounded by the “<>” symbols. This text appears in italics and is intended to provide explanations and guide you through the document. There are two types of comments in this document. The comments that are in black are intended specifically for the course. The comments that are in blue are more general and apply to any SRS. Please make sure to delete all of the comments before submitting the document**.>*

# Introduction

The Expense Tracker 1.0 is a web application to help users analyze their spending habits and set saving goals. The tracker will take user input and analyze that input to display their spending habits for them to reflect on.

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

## Document Purpose

This Document will describe in detail, the functional and nonfunctional requirements of the Expense Tracker 1.0. This product will track and analyze expense data as well as earnings data for the user. This Document will also serve as log to track changes to the project as it evolves.

This document will provide a concrete resource for the developers to look back on as they continue to work on the project and allow them to keep track of the requirements related conversations that occur throughout devlopment. It will also allow for the client and professor to evaluate how well the developers designed a product to match the specifications.

## Product Scope

The Expense Tracker 1.0 will take expense data from the user and use the data to provide the user with analytics. The data from user will provide the expense value, expense category, and date of expense. The program will give analytics such as weekly expenses, monthly expenses, yearly expenses, expenses based on category etc.

The program will also allow for the user to input their earnings and use that to show how much they are making or losing within specified time frames. This will allow for the user to better understand their financial situation and learn how to better reach their financial goals.

## Intended Audience and Document Overview

This document is intended for the client, the devlopers, the professor. The client will use this document to ensure that the developers are creating a product that fits their specification. The devlopers will use this document as a reference to ensure that they are creating a qualtiy product that follows the specifications. The Professor will use this document to analyze the students SRS writing abilities and to understand the scope of the project that they will be working on for the rest of the semester. Ideally this webpage’s audience will be the general public, so that they can use a simple program to let the see a very basic outline of the money they have spent on a certain time table and if they should fix how much they are spending to save more money.

## Definitions, Acronyms and Abbreviations

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.

TO DO: Please provide a list of all abbreviations and acronyms used in this document sorted in alphabetical order.>

## Document Conventions

<In general this document follows the IEEE formatting requirements. Use Arial font size 11, or 12 throughout the document for text. Use italics for comments. Document text should be single spaced and maintain the 1” margins found in this template. For Section and Subsection titles please follow the template.

TO DO: Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. Sometimes, it is useful to divide this section to several sections, e.g., Formatting Conventions, Naming Conventions, etc.>

## References and Acknowledgments

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document.

TO DO: Use the standard IEEE citation guide (attached) for this section.>

# Overall Description

## Product Perspective

 This product is a brand new webpage driven expense tracker. It has no previous components and is not a follow-on member of any other product family. This expense tracker will be compromised of a website that will allow the user to navigate all the features. These features will be coded using the Javascript language and the webpage will be setup using HTML. The webpage will feature a landing page for the user, so that they can traverse the different features that we will have on separate pages. The user will input income and expense information

on another webpage, and that webpage will also be connected to the

rest of the pages, allowing for easier access to the data they

wish to chart and or graph.

## Product Functionality

* Landing page that lets them find the other features
* Income/Expense data reciever
* Income tracker that accepts both the income and expenses and sees if they’re making/losing money and other little things
* Expense tracking function which can show expenses for weekly/monthy/yearly and category specific expenses.

- The landing page allows the user to have ease of access to the other functions without having to traverse a bunch of webages to find exactly what they are looking for.

- The Income/Expense reciever is there so that the user can have a place to input their data, which is used later by the other functions of this product, so that we can provide proper data.

- The income tracker is a function that will add together the income and expenses of the user and will show them visually how their finances are looking in general.

- The expense tracking function will be more detailed and focus primarily on expense data. This function will have the ability to show expenses for multiple time frames and categorical expenses.

## Users and Characteristics

Our product is geared towards any person/entity that is looking to keep track of their finances and potentially use our software to make a change in that behavior. Our product is geared towards simplicity and ease of access, there are not any functions for accepting giant loads of information and different kinds of incomes, for example derivative options that some might be holding.

Our main user will be the average person who is not part of a huge corporation, that just needs to see where their finances are going wrong and where they can improve. This user has the priority when it comes to our functional design, so that they might be able to access every feature without any expertise on their part. There will be those who use our product for corporations and bigger entity’s, however they are not given priority when it comes to having functionality geared specifically for them, rather they must use our simple functions in their own creative ways to adapt for themselves.

## Operating Environment

Our product will be running on the computer that contains the files for the webpage. This means that it is portable and although it is not online, it is a website that is usable on the local machine. This means that our product will work in any environment that has access to a browser that access HTML and javascript, internet connection is not required. It also means that operating system will not be a conflict, it will work on Windows, Linux, Unix, etc. This product will basically be able to run as soon as someone clicks on the html file that we will provide.

## Design and Implementation Constraints

There are no hardware issues that will limit us from producing our product. Our product has no security considerations because it is a locally based product which only the computer user themselves will have access to. There are also no programming language requirements, because this is a webpage based tool, you can include other little scripts and functions from languages like Python or Java, as long as you know how to integrate them with HTML. The only issue would be a language barrier between us and the entire world. This is due to the fact that our product will be produced in English only, and we do not have any support to switch between multiple languages. The users will be responsible for maintaining and upgrading the software as they see fit, we only provide the base version.

## User Documentation

Our user manual would be rather simple because our product is self-explanitory. The user manual would show how to navigate the webpages. It would show the user how to input their income/expense data, and also how to remove income/expense data they no longer want. It will also show the user how to navigate the different charting/graphing tools that our product provides. All of these functions would be self explanitory due to their naming and general use conventions by the majority of people. A tutorial showing how each function works and how they fit together would also be approriate as seeing someone use the software is a lot easier then reading about how to use it.

## Assumptions and Dependencies

As of the creation of this first document, there are no assumed factors that will affect the requirements stated in the SRS. We have not planned to include any third-party or commercial components that will cause issues around the development of our webpage.

# Specific Requirements

## External Interface Requirements

### User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., Cancel) that will appear on every screen, error message display standards, and so on. Define the software components for which a user interface is needed.

TO DO: The least you can do for this section is to describe in words the different User Interfaces and the different screens that will be available to the user. Optional: You may also provide an initial Graphical User Interface design (does not have to be final).>

This webpage will contain user interfaces that will require the user to input information about their expenses for the webpage to store the data and calculate the information given to return an accurate estimate of their expenses and other data that could be provided to the user. These will be in the form of text boxes where the user will input their information. These text boxes will also include buttons such as “calculate” or “clear” which will be used to execute the webpage to calculate their expenses or clear their user input without the user needing to hover over the text box and deleting each character. Error messages will appear on the user interface if characters inputted by the user will conflict with the calculating process such as not having numerical values in the text box.

### Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware. You are not required to specify what protocols you will be using to communicate with the hardware, but it will be usually included in this part as well.

TO DO: Please provide a short description of the different hardware interfaces. If you will be using some special libraries to communicate with your software mention them here. In case you have more than one hardware interface divide this section into subsections.>

### Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems (Windows? Linux? Etc…), tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.

TO DO: The previous part illustrates some of the information you would usually include in this part of the SRS document. To make things simpler, you are only required to describe the specific interface with the operating system.>

This webpage should be useable on all operating systems. This webpage will be originally worked on a windows operating system. The webpage will deal with its own internal database once the users have given their input on information for the expenses to be calculated. The input from the user will be data based on their daily,weekly,or monthly spending and the data being shown to the user from the webpage will reflect the users data input to give them estimates of their expenses and possible solutions or recommendations on saving their income.

### Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

TO DO: Do not go into too much detail, but provide 1-2 paragraphs were you will outline the major communication standards. For example, if you decide to use encryption there is no need to specify the exact encryption standards, but rather, specify the fact that the data will be encrypted and name what standards you consider using. >

This webpage will not have any requirements concerning e-mails, so that users may receive the information they are looking for instantly without having to dig through their messages. This webpage should be working on all web browsers, however the recommended web browser will be Google Chrome. This webpage will not require any electronic forms or network server communications. That data given from the user will be completely removed from the data system once the user has given their data input and the webpage has given them the information the webpage provides.

## Functional Requirements

*< Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. This section is the direct continuation of section 2.2 where you have specified the general functional requirements. Here, you should list in detail the different product functions with specific explanations regarding every function.*

*TO DO: Break the functional requirements to several functional areas and divide this section into subsections accordingly. Provide a detailed list of all product operations related to these functional areas.*

LANDING PAGE

The landing page will contain information on what the webpage will be about and how to navigate through the webpage to reach the different applications that is available in the webpage. The landing page will contain links to certain parts of the webpage that allows the user to interact with.

INCOME/EXPENSE DATA RECIEVER

The data reciever will be where the user will input the data they want calculated and the webpage will store that data and then convert the data given to produce expected values for the user to see.

INCOME TRACKER

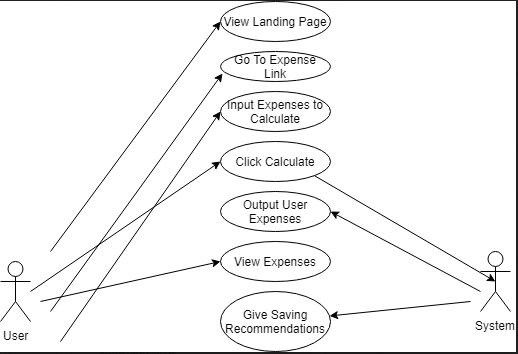
The income tracker will take the data inputted by the user and then use the algorithim to produce values of the users net gain or loss per time table. This will help users to figure out if they should cut back on spending or that they are spending enough to be saving their income.

EXPENSE TRACKING FUNCTION

Once the user has given their data input into the tracking function, the function should gather their data and produce values for each category of specific expenses ranging from necessary expences such as groceries or bills, or luxury expenses such as dining out or going to the theatres. These expenses will be totaled for weekly/monthly/yearly so that the user can see the scope of their expenses and be able to plan out how they should be spending their income more wisely or recommendations to save more.

## Behaviour Requirements

### Use Case View



# Other Non-functional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.

TODO: Provide relevant performance requirements based on the information you collected from the client. For example you can say “1. Any transaction will not take more than 10 seconds, etc…>

## Safety and Security Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied. Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements.

TODO:

* Provide relevant safety requirements based on your interview with the client or, on your expectation for the product.
* Describe briefly what level of security is expected from this product by your client and provide a bulleted (or numbered) list of the major security requirements.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.

TODO: Use subsections (e.g., 4.3.1 Reliability, 4.3.2 Portability, etc…) provide requirements related to the different software quality attributes. Base the information you include in these subsections on the material you have learned in the class. Make sure, that you do not just write “This software shall be maintainable…” Indicate how you plan to achieve it, etc.>

# Other Requirements

<This section is **Optional.** Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A – Data Dictionary

*<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>*

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 the Teaching Assistant to determine the effort put forth to produce this document>