

Software Requirements Specification

for

Expense Tracker

Version 1.0

Prepared by

Group Name: KVC Inc.

|  |  |  |
| --- | --- | --- |
| 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 |
| --- | --- | --- | --- |
| 1.0 | Chayse Jorgenson  Vitaliy Kudrik  Kevin Tong | Creation of SRS. | 10/25/2019 |

# 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.

## Document Purpose

This Document will describe in detail, the functional and nonfunctional requirements of the Expense Tracker 1.0. This product will track and anaylze 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 conversatiosn 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 the user will provide will be 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.

This document will go over the functional and nonfunction requirements of the the Expense Tracker 1.0 as well as work a as log to keep track of changes to said requirements and the project grows and evolves.

## Definitions, Acronyms and Abbreviations

There are currently no acronyms or abbreviations being used within this document. As the project evolves, accronyms and abbreviations incorporated will appear here.

## Document Conventions

This document follows IEEE formatting requirements. It uses Arial font throughout the document. Section headers use Bold size 18 font, Sub-section headers use Bold size 14 font, text uses size 11 font and comments are denoted by italics. This document maintains 1” margins is single spaced.

## References and Acknowledgments

In its current state, this document does not contain any other documents to reference

# 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/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 his finances are going wrong and where they can improve. This user has the priority when it comes to our functional design, so that he might be able to access every feature without any expertise on his 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

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.

TO DO: Provide a short list of some major assumptions that might significantly affect your design. For example, you can assume that your client will have 1, 2 or at most 50 Automated Banking Machines. Every number has a significant effect on the design of your system. >

# Specific Requirements

## External Interface Requirements

### User Interfaces

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

The website will interact with files that it stores directly on the local machine. This will allow us to avoid security issues as well as allow the user to operate the app without access to the internet. Aside from this file system there will be no other hardware interfaces with the software.

### Software Interfaces

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

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

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

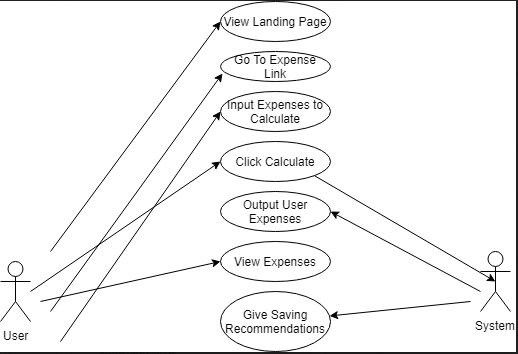
The income tracker will take the data inputted by the user and then use the algorithim to produce values of the users net gain or loss per time table. This will help users to figure out if they should cut back on spending or that they are spending enough to be saving their income.

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

Our performance requirements are limited to the speed of the users personal computer. This is a lightweight website application that does not require much speed or memory in order to run. There is also no storage in any databases either, rather we just create little files in the repository where the main webpage file is held. There is no connection to a server or the internet, thus there is no lag to be worried about. The only requirements again are hardware requirements on the user’s side, and any PC that can run a browser will be able to run our functions. The only thing the user would need to account for is the memory required to hold a few graphs/charts that are created by our tool.

## Safety and Security Requirements

This product has no safety or security requirements from the creators. This is because our product is locally run and not connected to any databases. Our product also does not run on the internet, even though it is a webpage application. As such there is no risk of leaked income information as a result of using of our software. However, the only risk that must be considered is the fact that our files might be found and used by a hacker that might’ve compromised the users PC with means outside of our control. The simple way to avoid this issue is to make sure that the user understands that our tool is fully in their control and as such, all of the information they create using it will be their responsibility to protect and maintain.

## Software Quality Attributes

Sections:

### 4.3.1 – Adaptabiltiy/Flexibility: Our product is a combination of HTML and javascript code. This means that the addition of more html webpages and more javascript code is always possible. Our code will be easily readable and well commented so that if someone were to change any of our functions, they could do so with ease.

### 4.3.2 – Availability/Portability: Our product is locally run and not connected to the internet or any databases. As such anyone with access to a browser with the ability to run HTML/Javascript code would have access to our product. This must always be the case, because we are geared towards the average person and not towards big corporations.

### 4.3.3 – Correctness/Reliability: Our product is based on using correct algorithms to provide our users with charts/graphs that show them how they are spending their money. This means that our algorithms have to be fool-proof and work within reasonable Big-O time complexity.

### 4.3.4 – Maintainability/Robustness: Our product is very simple and as such there is no need to have it maintained as the laws of addition and subtraction are not going to change. The only maintainace needed would come as a result of the user having made their own changes that might need to be revised.

### 4.3.5 – Usability: Usability is very important to our product. It is a product geared towards the average person and as such each function must come with a reasonable name and a clear description. If a function is unavoidably complex, then there must be a tutorial or some kind of informative tool that will allow the user to understand how to use the function without understanding the intricities of the function.

### 4.3.6 – Testabiltiy: Our code will be tested using minimalistic use cases as well as excessively large cases to make sure that our product can withstand anything our user will throw at it. The code itself is also in a separate javascript file, and as such, our users can go in and test each individual algorithm personally as they see fit. Thus our testability is very diverse and doesn’t allow many bugs to come through.

# Other Requirements

As of right now there are no other requirements. As the project evoloves we will add those requirements here.

Appendix A – Data Dictionary

*As of right now there is nothing that requires definitions, as the project develops we will add those here.*

Appendix B - Group Log

10/14

Met in person brainstormed ideas discussed basic requirements for the Expense Tracker

(60 mins)

10/21

Met in person to flesh out more details on requirements

(90 mins)