

Software Requirements Specification

for

Budget Forecast

Version <X.X>

Prepared by

Group Name: Team Budget

|  |  |  |
| --- | --- | --- |
| Julian Keller | <student #> | julian.keller@wsu.edu |
| Tyler Higgins | 11537332 | tyler.higgins@wsu.edu |
| <name> | <student #> | <e-mail> |
| <name> | <student #> | <e-mail> |
| <name> | <student #> | <e-mail> |

|  |  |
| --- | --- |
|  |  |
| Date: | October 7, 2018 |
|  |  |
|  |  |
|  |  |

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 | Julian Keller  Tyler Higgins | SRS document creation. | 10/07/18 |

# *TODO delete all comments before submission.*

# *<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

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

Budget Forecast is a simple web app designed to assist users in financial budgeting and planning for future expenses. This section will outline the basic purpose, scope, and intended audience of this project.

## Document Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.

TO DO: Write 1-2 paragraphs describing the purpose of this document as explained above.>

The purpose of this document is to give a detailed description of the requirements for the Budget Forecast Web app version 1.0. It will explain who the system is for, how it will work, the functionality of the system, the functional requirements and the non-functional requirements.

## Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals.

TO DO: 1-2 paragraphs describing the scope of the product. Make sure to describe the benefits associated with the product.>

The Budget Forecast system is a web app that will allow a user to quickly and easily budget their money as well as plan for future expenses. A user will be able to create various categories of expenses and keep track of how much money they have in each category. For example, a user could create a grocery category and allot $300.00 to it. They spend $50.00 at the grocery store and later subtract the $50 from the grocery category leaving them with $250.00 left for groceries. Additionally, a user could be saving for a new computer. They know ahead of time that the computer will cost $1500.00. The system will allow them to track how much they have saved so far, how much they still need to save, and estimate how long it might take them to reach their goal.

This is a beneficial system as it will allow users to make wise and informed financial decisions. A secondary benefit is that this system will be easy to use. It is a web app which will give a user easy access. The goal will be to keep the web app very user friendly and not over complicated.

## Intended Audience and Document Overview

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers (In your case it would probably be the “client” and the professor). Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

This document is intended for the client and the professor. The rest of this document contains document information, an overall system description, specific system requirements, and non-functional system requirements.

It is recommended that the client read Sections 2.1 – 2.3. These sections will explain the product perspective, the product functionality, and the characterstics of it’s users. Reading these sections will allow the client to have a good overall understanding of the product.

It is recommended that the professor read Section 2, 3, 4. Section 2 will give a solid description of what the system is, who it is for, and where it will operate. Section 3 will outline specific requirements of the system. Section 4 will outline the non-functional requirements of the system.

## 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.>

TODO add to this if you want when writing the second half.

* SRS – System Requirements Specification
* System and Product – Budget Forecast Website Application.
* User – a person who interacts with the Website Application.
* Web App – Website Applicatoin

## 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.>

This document follows the IEEE formatting requirements.

Formatting Convention Specifics

* Document uses Arial font with 1” margins.
* Sections use size 18 white font with a black background.
* Subsections use size 14 font.
* Text font size 11 with single spaced lines.

## 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.>

[1] IEEE Standards Association, “IEEE Citation Style Guide”

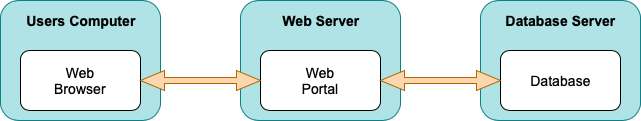
# Overall Description

## Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface. In this section it is crucial that you will be creative and provide as much information as possible.

TO DO: Provide at least one paragraph describing product perspective. Provide a general diagram that will illustrate how your product interacts with the environment and in what context it is being used, i.e., context diagram.>

This system is a self-contained web app. The web app will have a landing page which introduces the application to users. It will allow users to create an account and login to the website. Once logged in the user may create a budget and add values to it. A database will be used in order to store each users account and associated budget data. This will allow the user to access the data regardless of their location.



## Product Functionality

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, will be effective.

TO DO:

1. Provide a bulleted list of all the major functions of the system

2. **(Optional)** Provide a Data Flow Diagram of the system to show how these functions relate to each other. This is useful when there is a clear sequence for the functions being performed.>

* Create User Account
* Delete User Account
* Create Budget
* Delete Budget
* Create Individual Budget Categories
* Delete Individual Budget Categories
* Edit Budget Values
* Save Budget

## Users and Characteristics

<Identify the various users that you anticipate will use this product. Users may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience.

TO DO:

1. Describe the pertinent characteristics of each user. Certain requirements may pertain only to certain users.

3. Distinguish the most important users for this product from those who are less important to satisfy.>

There are two types of users that will use the system. Users who are experienced budget users and those who are inexperienced budget users.

Users with no budget experience will need to understand the fundamentals of how to use a budget prior to being able to fully use the system. They will also need to learn the basic functionality of the system.

Users with budget experience will not need to learn the fundamentals. Instead they will just need to learn the basic functionalities of the system. These are our most important users as they will most likely be the ones making greater use of the system.

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist. In this part, make sure to include a simple diagram that shows the major components of the overall system, subsystem interconnections, and external interface

TO DO: As stated above, in at least one paragraph, describe the environment your system will have to operate in. Make sure to include the minimum platform requirements for your system. >

The system will be able to operate on web browsers available on Linux, MacOS, Windows, Android, and iOS operating systems. These web browsers include Google Chrome, Firefox, and Safari.

Minimum Browser Requirements:

* Google Chrome version 69.0.3497
* Firefox version 62.0.3
* Safari version 11.1.2

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).

TO DO: In this section you need to consider all of the information you gathered so far, analyze it and correctly identify relevant constraints.>

Microsoft Edge and Internet Explorer 11 are not officially supported, although the user may be able to run this site on either web browser, it is not recommended since it will compromise the usability and reliability of the web app. Language requirments include html-5, CSS, and Javascript with EMCAScript 6. The software team will be providing the appropriate matainence and security updates to the site. For security purposes, the development team must implement a type of secure encryption while transferring data to the web server, database server, and to the user. All web data must use the HTTPS protocol.

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.

TO DO: You will not actually develop any user-manuals, but you need to describe what kind of manuals and what kind of help is needed for the software you will be developing. One paragraph should be sufficient for this section.>

There will be an online help page. The online help page will feature a Frequently Asked Questions (FAQ) section when the User first loads the help page. There will also be categories that the user can click on to get detailed help on navigating/using the budgeting app. Another thing that will be implemented is help and tips on budgeting. Finally, there will be a ‘contact support’ section in case the help section is unable to help with their needs, where the user will fill out their question or issue, and have it resolved by a support agent.

## 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. >

One assumption is that since the app is cross-platform (Meaning it works on both mobile and desktop devices), There will be the same functionality across all platforms, however the UI might be different due to mobile devices display constraints. We assume the user will be using Google Chrome version 69.0.3497, Firefox version 62.0.3, Safari version 11.1.2, this software will most likely not render properly in Microsoft Edge and Internet Explorer since it will not officialy be supported.

# 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).>

There will be two different types of user interfaces:

1. For desktop users – We will have a GUI with a navigation bar on top, as well as a login button. If the user chooses to login, they will be prompted with a username and password and then taken to their budgeting “home” page, which will display their balance in each budgeting category they created, recent transactions, and a button to add more money to their budget.

* If the user does not login, they will have an option to Sign up, beside the login button. The home page will display information about our budgeting app as well as a couple budgeting tips & tricks.

1. For mobile users – the UI will be simplified with a login button, sign up button, and content about our budgeting app.

* If user clicks login, will be prompted with username/password, there will then be displayed their balance in their most recently created budget. There will be a side button on the side to display a menu of options. Each option will take you to a different section.

### 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.>

We will communicating with an outside server with an online database to send/retrieve user information.

### 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.>

Operating environments include the latest version of Google Chrome, Safari, and Mozilla Firefox. In mobile devices, we will communicate with the Android and iOS platform when prompting for imput to request the keypad.

### 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. >

Major communication interfaces include HTTPS, E-mail (for support services and sign up), and web browser. Sensitive data such as email, addresses, names, etc, will be encrypted.

## 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.*

**Create User Account**

* Using a sign-up button, a user can create their account. Requirements to complete an account are a users valid e-mail address, physical address, password (with confirmation) and annual income. Additional but optional information would be phone number.

**Delete User Account**

* After a user signs into their account, they can click on the “Settings” link to go to their settings, one of the options will to delete their account, if the user presses that button, they will be prompted with their password then their account will be permenately deleted.

**Create Budget**

* After a user creates an account and signs in, they will have an option to create a budget. This can include anything from retirement, to a down payment for a car, or anything the user wishes to budget for. The user will be prompted for a goal for this budget, a name, and and initial balance for the budget. (can range from $0 to the goal amount). When finished, their new budget will be displayed on their dashboard.

**Delete Budget**

* A user can decide to delete a budget if needed. To do this they click on the budget they want to delete and then click the “delete budget” button on the bottom of the screen. Once the user presses that button, they confirm with their password, and the budget will be removed from their dashboard.

**Create Individual Budget Categories**

* A user from their dashboard can easily create categories for their budgets by clicking on the “+” button, it will let them name the category, and drag the budgets of their choosing into that category. Placing budgets in a category adds their goals to create a total goal amount.

**Delete Individual Budget Categories**

* If the user decides they no longer want their budget categorized, then they can delete it by selecting the category and pressing the “Delete” button. When they press “Delete” the category will be deleted, however the budgets in the former category will still be available.

**Edit Budget Values**

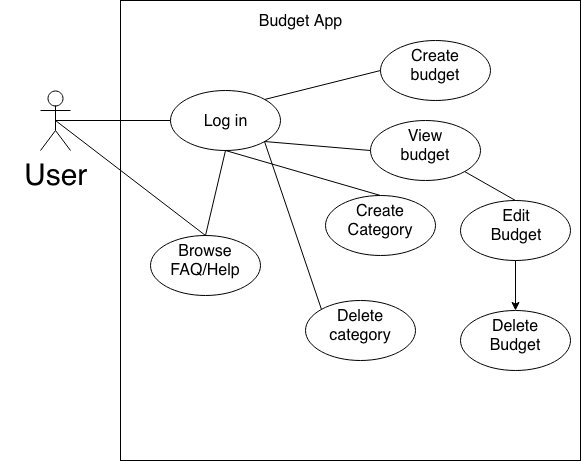
* If a user needs to adjust their budget, such as your goal on a particular budget, they can click on the budget you wish to edit, then click the “Edit Values” button, they then can change the value to the appropriate goal. To save the changes they click “OK” or to discard changes, they can press “Cancel”, which they will be prompted with a confirmation dialog. For changing the balance toward their end goal, they simply click the “Add money to budget” button and it will add the amount the user specifies to the current balance and save the new value.

## Behaviour Requirements

### Use Case View

<A use case defines a goal-oriented set of interactions between external actors and the system under consideration.

TO DO: Provide a use case diagram which shows the entire system and all possible actors. Do not include detailed use case descriptions (these will be needed when you will be working on the Test Plan), but make sure to include a short description of what every use-case is, who are the actors in your diagram.>



# 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…>

The app should take no longer than 5 seconds to update the server’s database after a user has made any changes to their budget, categories, or profile. Each page should take no longer than 1 second to load. The app should be able to be optimized for the device the user prefers, and all load/update times should be around the same time across all platforms (Desktop, mobile devices, phones, 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.
* Since this app will use some of the user’s personal/private information, security is a top priority. It is required that the web app is to be updated with the latest security features available, as well as the servers holding the users data also be updated with the latest security features. If there is discovered any leak of user information, such as usernames and passwords, we will require the user to prompt for another password.
* 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.>
* The highest security is a must for this application, since it is a web app. Here are the list of security requirments:

1. Using the https protocol
2. Using the highest level of encryption software while data is being transferred from the server to the user, or vice versa.
3. Requiring the user to change their password every 90 days.
4. Requiring the user for a security question and answer.
5. Using advanced Captcha to keep bots from creating accounts and compromising the system.

## 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.>

### Maintainability

It will be required to use EMCAScript6, and use ESLint error checking in the code. We will also enforce a cs320 style guide as our standard for readability of the code.

### Reliability

We will enforce a strict reliabilty policy. We will schedule routine matainence at least twice a month in order to maintain the sites reliability as well as update security features. There will be no longer than a 24 hour downtime for routine matainence. If there are a lot of users reporting an issue, our team will fix the issue promptly with as little downtime as possible.

### Usability

There will be provided a lot of tools in order to make our system as usable as possible. Our software will be simple to navigate, with simple menus and minimizing submenus as much as possible. Our main page will be streamlined, meaning no loading or refreshing a webpage as a user updates their budgets. If the user is visioned-impaired, the user can click on the zoom button on their web browser, we will ensure the size of the font will be bigger without compromising the layout of the webpage.

# 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>

* 10/01/18
  + Finalized Project Idea
  + 45 minutes