WASHINGTON STATE UNIVERSITY
VANCOUVER

Software Requirements Specification

for

Budget Forecast

Version <X.X>

Prepared by

Group Name: Team Budget

Julian Keller	**11535284**	**julian.keller@wsu.edu**
Tyler Higgins	**11537332**	**tyler.higgins@wsu.edu**

Date: October 7, 2018

Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Julian Keller Tyler Higgins	SRS document creation.	10/07/18

1 Introduction

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.

1.1 Document Purpose

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.

1.2 Product Scope

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.

1.3 Intended Audience and Document Overview

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.

1.4 Definitions, Acronyms and Abbreviations

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

1.5 Document Conventions

This document follows the IEEE formatting requirements.

Formatting Convention Specifics

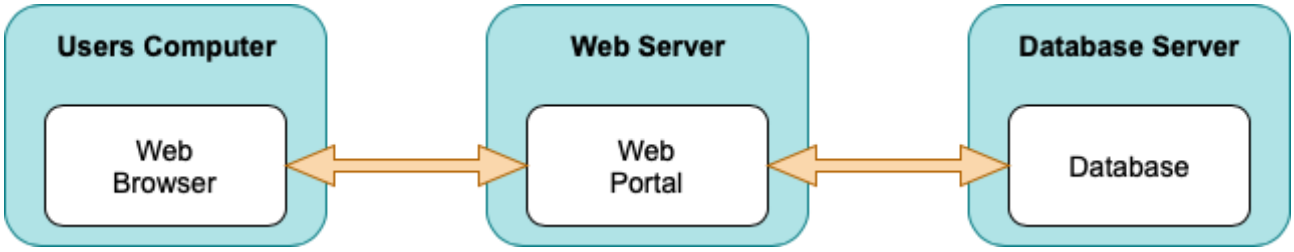
- Document uses Arial font with 1” margins.
- Sections use size 18 white font with a black background.
- Subsections use size 14 font.
- Text font size 11 with single spaced lines.

1.6 References and Acknowledgments

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

2 Overall Description

2.1 Product Perspective



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.

2.2 Product Functionality

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

2.3 Users and Characteristics

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.

2.4 Operating Environment

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

2.5 Design and Implementation 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.

2.6 User Documentation

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.

2.7 Assumptions and Dependencies

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.

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

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.
2) 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.

3.1.2 Hardware Interfaces

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

3.1.3 Software Interfaces

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 input to request the keypad.

3.1.4 Communications Interfaces

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.

3.2 Functional Requirements

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.

Log into User Account
- Using two text boxes and a sign-in button a user can enter the username and password for their existing account to gain access to their budget and information.

Log out of User Account
- Using a logout button, the user can choose to log out of their account.

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.

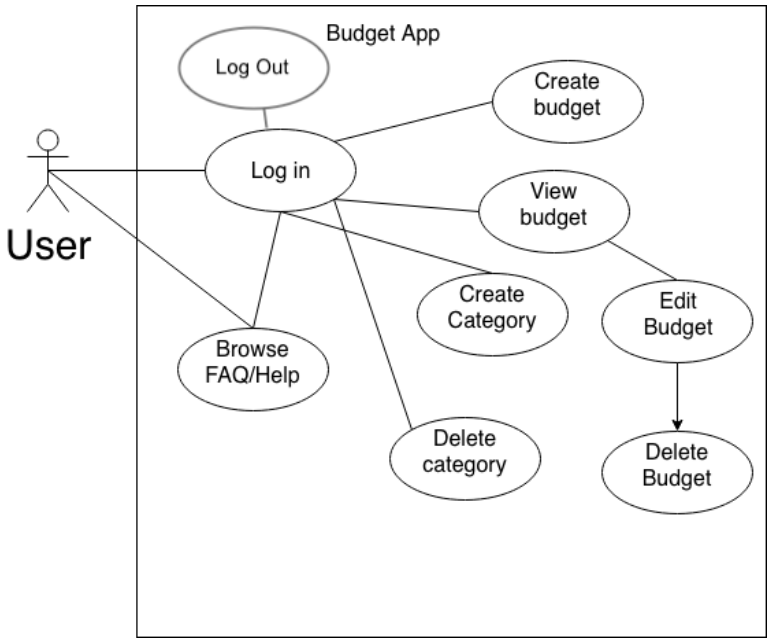
Create Individual Budget Categories
- A user from their dashboard can easily create categories for their budgets by clicking on the "+" button, it will let them name the category, and drag the budgets of their choosing into that category. Placing budgets in a category adds their goals to create a total 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. Any balance budgeted in the deleted category will become available for the user to allocate to a budget category of their choosing.

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.

3.3 Behaviour Requirements

3.3.1 Use Case View



4 Other Non-functional Requirements

4.1 Performance Requirements

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).

4.2 Safety and 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.

4.3 Software Quality Attributes

This section outlines the system's maintainability, reliability, and usability.

4.3.1 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.

4.3.2 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.

4.3.3 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.

5 Other Requirements

There are no additional requirements at this time.

Appendix A – Data Dictionary

- Logged In – The user must be logged in to do these actions.
  - View Budget – The user can view the data in their budget.
  - Edit Budget – The user can view and edit the data in their budget.
  - Create Budget – The user can name, create, and add data to their budget.
  - Delete Budget – The user can view and delete their budget.
  - Create Category – The user can name and create a category within their budget.
  - Delete Category – The user can view and delete a category with their budget.
  - Log Out – The user can log out of their account

- Logged Out – The user must be logged out to do these actions.
  - Log In – The user can enter their username and password to login to their site.
  - Create Account – The user can create an account by providing an account username, password, email address, and income.

- Loggin In or Logged Out – The user can be logged in or out to view these pages.
  - Home Page – The user can view the home page of the site.
  - FAQ/ Help Page – The user can view and read the frequently asked questions and help page.

Appendix B - Group Log

- 10/01/18
  - Finalized Project Idea
  - 45 minutes

- 10/05/18
  - Group communication via Telegram Messenger about project details.

- 10/07/18
  - Group communication via Telegram Messenger to review and finalize SRS documentation.