



By Warren Spencer

Table of Contents

Table of Contents	<u>ii</u>
Revision History	<u>ii</u>
1. Introduction	<u>1</u>
1.1 Purpose	<u>1</u>
2. Overall Description	<u>2</u>
2.1 Product Perspective	<u>2</u>
2.2 Product Functions	<u>2</u>
2.3 Operating Environment	<u>2</u>
3. External Interface Requirements	<u>3</u>
3.1 User Interfaces	<u>3</u>
3.2 Hardware Interfaces	<u>3</u>
3.3 Software Interfaces	<u>3</u>
4. System Features	<u>4</u>
4.1 System Feature 1	<u>4</u>
4.2 System Feature 2	
4.3 System Feature 3	
5. Other Nonfunctional Requirements	<u>5</u>
5.1 Performance Requirements	

Documentation for CentDash

1. Introduction

1.1 Purpose

Managing personal finances can be quite overwhelming. Everyday individuals are faced with financial decisions such as spending, tracking expenses, budgeting, and the overall task of the management of their finances. With the cost of living increasing daily, financial decision making can be challenging to many people as there are risks to overspending and making bad decisions with their income. 'CentDash' emerges as a guide, providing a platform that allows its users to keep track of their bills and expenses, monitor their spending, set a budgeting plan, and engaging them with financial reporting in order to make better decisions financially. With intuitive functionality along with its clean interface, 'CentDash' aims to enrich its users by calculating and analyzing their financial data in order to help them make better decisions and guide them to achieving financial growth.

2. Overall Description

2.1 Product Perspective

'CentDash' is a web-based software application that aims to help individuals with the overall management of their personal finance by providing a data-integration platform that gives its users the power to foster financial growth and stability with features that gives them the tools in order to make better decisions on spending, saving, and budgeting.

2.2 Product Functions

At its core, 'CentDash' offers the following major functionalities:

- **Income and Expense Tracking:** Allows users to track all streams of income and record and categorize their personal expenses and transactions.
- **Calculation and Analysis:** Calculates and analyzes all financial activities in order to provide insight into making better financial decisions.
- **Financial Dashboard:** A clean and user-friendly graphical dashboard that allows users to visualize their financial data and information.
- **Financial Statements and Reporting:** Provides users with financial statements in order to assess their financial position.
- **Resource Hub:** Generates articles, videos, and podcasts related to personal finance in order to increase financial literacy and awareness.

2.3 Operating Environment

The application is developed exclusively to be accessed on web-based browsers such as Microsoft Edge, Google Chrome, and Mozilla Firefox. While many personal finance applications are based on mobile devices, 'CentDash' will only be available as a web-based application.

2.4 Design and Implementation Constraints

The application's tech stack and technical architecture is to utilize HTML5, CSS, JavaScript, jQuery, AJAX, and Bootstrap for the frontend and Python Flask for the backend with SQLite for handling the database of the application.

3. External Interface Requirements

3.1 User Interfaces

My software application utilizes HTML, CSS, and JavaScript along with Bootstrap libraries. Leveraging these frontend languages will offer users a clean and rich UI that is easy to use and navigate.

Sample screens will include:

- A login screen that gives the users the option to sign in or sign up for the application.
- A sidebar with internal links to webpages of the application.
- An interactive dashboard that allows users to visualize their financial data.
- A budgets, net worth, and statements page that allows users to record and view financial activity.

Consistent elements across screens:

- Buttons that allow users to add input fields and calculate financial data.
- Interactive inner text that changes accordingly based on user input of financial data.
- Interactive dashboard widget cards that update based on user's input on Budgets and Net Worth webpages.

3.2 Hardware Interfaces

This software application is primarily for web browsers (Microsoft Edge, Google Chrome, and Mozilla Firefox).

3.3 Software Interfaces

The application interfaces with Python Flask and SQLite for backend services including authentication, and database.

- Incoming Data: User authentication details, user input for feature functionalities.
- Outgoing Data: Results from queries, and data to be displayed.

Both incoming and outgoing data are also integrated with jQuery to receive and record financial data by the user.

4. System Features

4.1 System Feature 1

User Authentication:

4.1.1 Description and Priority:

Feature: Secure User Authentication and Seamless Onboarding

Priority: *High*. Ensuring data privacy is imperative for this software application.

4.1.2 Stimulus/Response Sequences:

- **User Action:** Launches the application for the first time.
 - **System Response:** Presents the user with a login screen with the option of signing up.
- **User Action:** Clicks on the sign-up link.
 - **System Response:** Redirects user to the sign-up page.
- **User Action:** Inputs sign-up data such as email and password.
 - **System Response:** Validates the user's sign-up information for the application and redirects to the login screen.
- **User Action:** Inputs login credentials on subsequent visits.
 - **System Response:** Signs user into the application.

4.1.3 Functional Requirements:

REQ-1: Integrate SQLite to support email and password-based authentication.

REQ-2: Ensure password criteria (minimum length, alphanumeric, special characters).

REQ-3: Provide feedback for invalid credentials or errors during registration/login.

REQ-4: Allows users to delete their account and information.

4.2 System Feature 2

Interactive Integration of Financial Data:

4.2.1 Description and Priority:

Feature: Interactive Integration of Financial Data inputted by the user with jQuery

Priority: *High*. Interactive elements are the foundation of the application, ensuring that financial data can be visualized and analyzed by the user.

4.2.2 Stimulus/Response Sequences:

- **User Action:** Enters financial data on Budgets and Net Worth webpages.
 - **System Response:** jQuery records and displays financial information for users on Budgets and Net Worth webpage.
- **User Action:** Records financial information on Budgets and Net Worth webpages.
 - **System Response:** Stores financial information and updates Dashboard and Statements webpages with it accordingly.
- **User Action:** Deletes financial information on Budgets and Net Worth webpages.
 - **System Response:** Deletes financial information and updates UI accordingly.

4.2.3 Functional Requirements:

REQ-1: Create an interactive and responsive Dashboard and also a comprehensive Statements page.

REQ-2: Integrate JavaScript's jQuery library to incorporate interactive elements for UI.

REQ-3: Update Dashboard and Statement's interface accordingly based on user's input of financial data.

4.3 System Feature 3

Web Scraping for Personal Finance Articles:

4.3.1 Description and Priority:

Feature: Web Scraping for Personal Finance Articles for 'CentDash Hub' with Python Flask's BeautifulSoup library.

Priority: *Medium*. Engaging users with personal finance articles will help the user make informed decisions and also increase financial literacy and awareness.

4.2.2 Stimulus/Response Sequences:

- **User Action:** Clicks on 'CentDash Hub' link on sidebar.
 - **System Response:** Python Flask's BeautifulSoup scrapes the web for links related to personal finance articles.

4.2.3 Functional Requirements:

REQ-1: Utilize web scraping for CentDash's 'CentDash Hub' feature.

REQ-2: Integrate Python Flask's BeautifulSoup library to generate personal finance articles from the web.

REQ-3: Update with new articles each day for the user.

4.3 Performance Requirements

Performance Requirements:

1. **Interactive Functionality:** The Dashboard and Statements features should record financial data in real time allowing users to visualize and analyze their financial information.
2. **Data Handling:** 'CentDash' should be capable of handling loads of financial information inputted by the user. This scalability ensures the best quality of input of financial information.
3. **Security Performance:** Users personal information should be safe and protected, ensuring the best quality of security of the application.