SP-26 PURPLE BUDGET APP

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Course: Senior Project Section 01 CS4850

Semester: Spring 2024

Professor Perry

Date: 2/8/2024

Team:

Cameron Lowry

Ann Nguyen

Alexandra Clark

ONLY USE PARTS OF THIS DOCUMENT THAT MAKE SENSE FOR YOUR PROJECT ! ! ! !

Table of Contents

1.0 Introduction

1.1 Overview

The budget app is a financial management tool designed to help users track, manage, and optimize their finances. With a user-friendly interface and robust features, the budget app is ideal for everyone who needs to achieve their financial goals and make informed decisions about their money.

1.2 Project Goals

Expenses tracking: Develop a platform allowing users to record and categorize their monthly income and spendings effortlessly.

User-friendly interface: Design an intuitive and appealing user-interface to enhance the overall user experience, making financial management accessible to diverse people.

Enhancing financial awareness: by providing a visualizing incomes and expenses through line chart and pie chart, the app enhances the user's awareness of their financial habit and encourages smarter spending and saving decisions.

Privacy and Security: Prioritize the implementation of security to ensure the confidentiality and integrity of user's financial data.

1.4 Assumptions

- Users will have basic knowledge of smartphones and how to navigate mobile apps.

- The app will mainly target people between 18-35 who are interested in improving their personal finance management.

- Users will have access to stable internet connections to use the app.

2.0 Design Constraints

Time Constraints: The project must be completed within the semester.

Budget Limitations: The project must be completed with free or near free resources.

2.1 Environment

Hardware Environment: The app will run on both iOS and Android devices.

Software Environment: The app will be compatible with the latest version of IOS and Android.

Network Environment: The app will require an internet connection for certain features.

2.2 User Characteristics

User Characteristics: Age Range: 18-35

Tech Proficiency: Moderate

Varied financial Literacy: users may have varying levels of financial literacy and familiarity with budgeting concepts. The app should provide clear explanations, tooltips, and educational resources to support users with various levels of financial knowledge.

2.3 System

Architecture: The budget app is designed for iOS and Android devices

Database: User data, profiles, transaction history, and budget information, is stored locally on the user's device.

Third-Party Integrations: The app integrates with PLAID to access user's bank information.

3.0 Functional Requirements

- Login and Password Authentication:
  - Users should be able to create an account with our app. Each user should have a unique username and password.
  - The app must verify credentials provided during login to ensure security.
  - Error handling and logging: the app should provide informative error messages to users in case of authentication failures (incorrect password, account locked, etc.)

- Display Homepage:
  - Upon successful authentication, the app should display the homepage/ dashboard providing an overview of the user’s financial status.
- Show upcoming expenses on a balance sheet.
  - A calculation must show the end balance at the end of a week and monthly periods.
- Allow the user to set a financial goal.

- Allow user to add upcoming expenses to the balance sheet.
    - Users should be able to add upcoming expenses through a click of a button.
    - The App should be able to edit any expense currently on the balance sheet.
- Display Insight Page:
    - A line graph showing the relationship between income and expenses over a period.
    - Expenses categories breakdown: a pie chart showing the distribution of expenses across different categories (groceries, transportation, entertainment, etc.)
- User Account Management Page:
    - Users should have the ability to access and manage their account settings, including profile information, security setting, and linked bank account inside it.
    - The app also should allow users to update their account information such as email address and address.
    - Users also should be able to change the passwords and set the notification.
- Navigate to Linking bank account:
    - Users should be able to link their bank accounts securely to the app.
    - The app will use a plaid website to link bank account securely.
    - The process of linking bank accounts should be friendly, with clear instruction and guide the user through necessary steps.

4.0 Non-Functional Requirements

4.1 Security

- Failed login must be tracked and restricted after a certain number of unsuccessful tries for security process.
- All sensitive user data, including passwords, financial transactions, and personal information should be encrypted both in transit and at rest to prevent unauthorized access.
- Implement strong authentication to ensure that only authorized users can access the app.
- Keep a detailed record of everything users do in the app, then if something goes wrong, it can undo the action and find out what happened and who did it.

4.2 Capacity

- Performance: The app should be capable of handling large volumes of simultaneous users and transactions without experiencing slowdowns or downtime.
- Scalability: make sure the app's design can handle more users and data as it grows without slowing or breaking.

- Load testing: test the app regularly to find any problems that could slow it down when lots of people are using it and fix them so the app can stay fast and reliable even during busy times.

4.3 Usability

- Consistency: The User interface should be consistent with Itself. This includes front sizes for titles, headers, and body texts. The app also should have a consistent color palette across all pages.
- Error Handling: Notify the user what went wrong if the app crashes or an error is found. Create a bug reporting section to allow users to notify the developers if there is a bug.

4.4 Other

- The app must be confidential, integrated, and availability.
- Design the app to be robust and resilient, capable of recovering from failure and error without any data loss or service disruption.
- Structure the app codebase in a modular and well-organized manner, with a clear separation of components, easy to maintain and enhance in the future.
- Ensure the app can integrate and communicate with other systems, services, or platforms.

APPENDICES