INDY-12 College Student Budget App (Frugal U)

SOFTWARE REQUIREMENTS SPECIFICATION (SRS)

Course: CS 4850-01 Senior Project

Semester: Spring 2024

Professor Perry

Date: 2/5/2024

Team INDY-12

		
Joshua Schoen	Logan Getzfred	Chase Mo

Table of Contents

1.0 Introduction

1.1 Overview

Frugal U is a college student budget app for young adults. It's an app that won't replace a bank account but plainly an app where users can create an easy budget for the main college expenses.

It will help students budget and learn where their money is going; it will have budget limits created by students and a Pie Chart breaking down each expense as a percentage of their income.

Its UI is going to be so easy to understand that even your grandma could use it, and a database will hold all the unique information the student inputs.

The signup system will use a student's college email address, with full authentication to avoid hackers. However, if someone were to crack into a student's account, they wouldn't be able to access anything incriminating as no financial information like credit or debit card information would be present there. This app is going to be a logging system for expenses, and a bank account will not be connected.

Each expense is going to be broken down into specific college and young adult-specific categories. Some expenses, such as housing and food, is going to be generic, but this is more aimed at college students with categories such as tuition and class-specific expenses like books and Pearson fees.

1.2 Project Goals

The goal of this application is that each student

- Is going to be able to know exactly where their money is going in one easy-to-use app.
- Able to learn and understand budgeting and what makes a great budget.
- Will see a visual representation of their expenses as a percentage of their budget via a pie chart.
- Is going to be able to access financial tips they weren't privy to without spending time googling.
- Relieve stress and wind down by understanding how their money is spent.

1.3 Definitions and Acronyms

UI: Also called User Interface, is the perspective of the user of an application, how the layout is presented, and how the buttons are organized.

API: Application programming interface: these APIs are responsible for storing and retrieving information that makes the app possible. APIs are going to be used to design limits, add expenses, and store general information unrelated to finance.

1.4 Assumptions

Students should have a baseline understanding of how to use mobile apps; however, it is not required as there are tutorials on how exactly to use the app.

The application requires internet access to store and retrieve information from the database.

2.0 Design Constraints

2.1 Environment

- The app is designed for iOS and Android mobile platforms.
- Requires internet connectivity for syncing data and retrieving updates.
- App is going to be made in Expo.dev

2.2 User Characteristics

- Users are going to be college students who are young adults aged 18-28. They can have varying degrees of financial literacy, they should at least understand what a budget is and what an expense is.
- Users need not necessarily have experience creating a budget, as the app will have built-in tutorials on how the process works.

- Users should have at least one source of income, either an allowance or some kind of job that pays a solid wage. However, even minimum wage jobs are going to be compatible with this app.

2.3 System

- The system should be able to support access to a large user base.
- Scalable database that can hold hundreds if not thousands of specific user information such as usernames, passwords, and individual budget profiles.
- The System should be able to handle any Android and IOS version, such as Android 10,11, etc.

3.0 Functional Requirements

- Login and Password Authentication: Secure login process with password recovery options. Users will sign in with their email and a unique password.
- Sign up: Users can enter their college email, first name, and password with college email confirmation.
- Profile Button: This will display unique user information such as first and last name, email address, and the ability to reset passwords or change colleges.
- Display Home Page: Overview of the user's expenses as a percentage of their income stream.
- Expense Tracking: Users can input and categorize expenses.
- Budget Setting: Users can set budget goals for different categories based on their income stream.
- Financial Tips and Education: Access to articles and tips on managing finances tailored to college and financial aid specific to that college.
- Alerts and notifications-to-unseizable notifications for budget limits and financial tips to navigate income easily from their paycheck via income stream. (Weekly, bi-weekly, semi-monthly, monthly).
- Having a button to view custom-set budget limits for user-specific categories.

4.0 Non-Functional Requirements

4.1 Security

- Authentication and Authorization: Use secure authentication mechanisms (e.g., OAuth 2.0) to manage user access and ensure users have appropriate permissions for different levels of data access.
- Regular Security Audits: Conduct regular security assessments and updates to protect against vulnerabilities, including SQL injection attacks, which are pertinent to MySQL databases.

- Data Privacy Compliance: Ensure compliance with relevant data protection regulations (e.g., GDPR, CCPA) to protect user privacy and data.

4.2 Capacity

- The app should be able to handle a user base of hundreds or even thousands of active users.
- The app's database should handle hundreds or even thousands of daily queries for active users.
- The app should plan for adequate server resources (CPU, memory, storage) to handle the expected load, with monitoring in place to scale or adjust as needed.

4.3 Usability

- The app should have an easy-to-use UI interface that a college student should be able to easily navigate.
- The app should have a feedback system such that students can leave feedback for the improvement of the UI
- The app should have accessibility systems in place like text-to-speech, microphone settings, and scalable font sizes for all different types of eye-sights.

4.4 Scalability

- The app should be able to scale up in the database as numbers climb above the tens of thousands.
- The app should be able to optimize the database such that cloud services could be used for an excess of information.

5.0 External Interface Requirements

5.1 User Interface Requirements

- The application will have a clean, student-friendly interface with easy navigation.

5.2 Hardware Interface Requirements

- Compatible with smartphones running iOS 11 or above and Android 8 or above.

5.3 Software Interface Requirements

- The application will need to be able to connect and communicate (Query information) from a database.
- The application will need to get specific information from websites (information from university websites or other general financial websites).

5.4 Communication Interface Requirements

- Utilizes HTTPS for secure communication with the server.

6.0 Glossary

- UI (User Interface): The means by which the user and a computer system interact, specifically the use of input devices and software.

- UX (User Experience): The overall experience of a person using a product such as a website or a computer application, especially in terms of how easy or pleasing it is to use.

- API (Application Programming Interface): A set of protocols, routines, and tools for building software applications, specifying how software components should interact.

- GDPR (General Data Protection Regulation): A regulation in EU law on data protection and privacy in the European Union and the European Economic Area.

- Database: A private entity that is connected to the main application is used for storing information.