Background Information

Domain:

Lifestyle - Personal financing application

Topics:

Personal Finance - Budgeting, tracking expenses

Analytics - Viewing expense statistics over a period of chance

Theme - Modern

Problem:

Everyday consumers are becoming more vulnerable to going over budget. Due to this, families that operate within the lower to middle class are continuously having difficulties tracking their day-to-day expenses and their budgets for certain periods of time.

Sub-Problems:

- It is difficult to remember one's day-to-day expenses. This, in turn, provides an extra challenge when considering the overview of the user's monthly finances.
- Continuously failing to maintain the monthly budget for necessities can be extremely demotivating for the user. The effectiveness of personal finance can be heavily influenced by one's ability to stay motivated when providing a stable budget.

Purpose and Audience

The purpose of XPENSE is to provide an outlet for the layman to locally track, manage, and view the user's expenses and how they are managed. This application is suited for those who need assistance in budgeting their expenses, such as those who have recently gaining a steady income, and families having trouble with their net expenses.

Storyline

This application should provide tools for tracking, viewing and managing one’s expenses. They will be able to create ‘wallets’ where they place their budget (weekly, monthly, or one-time) and subtract funds from that wallet by noting the user’s expenses. These expenses will be viewable based on the date entered, the name, the value, and the expense type. In addition to the standard view, the user will able to track average spending on various expense types in the month, to evaluate their spending habits.

Required Functionality: Functional Requirements

Login System

- Upon initialization, the system shall display multiple forms querying for a user’s login, and password.
  - Additionally, the user shall have the ability to select “Login” to verify one’s user information, or “Register” to be able to register for an account.

Registration System

- The system shall display multiple forms, including: “username”, “password”, “first name”.
- The system shall display buttons to register onto the system, and to return to the login system.
  - The system requires require the following information for a valid registration:
  - Username: 3-16 characters that are alphanumeric, or include an underscore
  - Password: 8-24 characters, with at least 1 uppercase letter, 1 lowercase letter and 1 number.
  - Name: 2-24 characters, that belong to the following characters: A-Z,space “ “, and dashes “-”
  - The user has not previously registered in the system.

- When the user enters a valid set of inputs, the system shall notify the user through a message that they have successfully been registered into the application.
    - The system shall successfully register the user onto the database.
    - The system shall return the user to the login screen.
- When the user enters an invalid set of inputs, the system shall notify the user that the system has not registered the user.
    - The system shall return the user to the registration screen.
- When the user enters a username and a name that is previously associated with another user, they will be prompted with a question, asking if they would like to change their password.
    - If the user clicks yes, the system shall query the user with the old password, twice, and their new desired password twice to verify the user, and update their password.

Menu System

- Upon correctly logging in, the system shall bring the user onto the main menu of the application.
- The system shall also greet the user with a message.
- The system shall provide the user with buttons to bring the user towards the 'Wallets' system, the 'Expenses' system, and the 'Analytics' system.

Wallet System

- Upon entering the wallet system, the system shall provide an overview of all of the user's wallets.
    - These overviews include the name, amount and wallet type of each wallet.
    - The wallets will have a maximum $1,000,000 fund and a maximum.
- The system shall provide a button to the user to add a wallet.
    - Upon selecting the "add a wallet" option, the system shall query the user for a valid wallet name, and a wallet type. (one-time, weekly, monthly)

    - The wallet name is considered valid if it is between 3-16 characters, belonging in the following set of characters: A-Z, a-z , 0-9 and space.
    - If the wallet type is weekly or monthly, the system shall query the user for an additional periodic fund.
- Upon clicking on a wallet, the system shall display a more specific view of the wallet, providing expenses and funding of that wallet (up to the last 5 activities).
    - The user shall have the ability to add expenses and add funds to these wallets.

Expense System

- Upon entering the wallet system, the system shall display a history of expenses, previously entered by the user.
    - The system shall provide tools to sort the history by expense type, and by account.
- The system shall provide a button that shall query the user for the necessary information to add an expense to their wallet.
    - This includes: Name, expense type, amount, wallet name.
- If the enter correctly enters the information, the system shall provide a message to notify the user that they have entered the information successfully.
- If the user has incorrectly entered information, the system shall provide a message to notify the user that they have incorrectly entered information.
- After the notification, the system shall clear all fields to allow the user to re-enter another expense.

Analytics System

- Upon entering the analytics system, the system shall provide the user with the total amount spent on expenses for the month.
- The system shall also display the average monthly spending amount for the last full month
- The system shall also display the net differential for the amount spent along with the average.

  - E.g. if the user has an average spending of $1500.00 and they have currently spent $800.00, the differential shall display $1500.00 - $800.00 = $700.00.

Required Functionality: Non-Functional Requirements

Performance Requirements

- The system shall provide a < 0.5 second response time for all users, given an internet connection within close proximity to the server. (< 500km)
- Under heavy load, the system shall provide a bound of at most 2 second response time for all users, given an internet connection within close proximity to the server. (< 500km)

Usability Requirements

- The system shall be usable by new users, and have a general understanding of the basic functionality within < 15 minutes of use.
- The system shall be usable by desktop users on a mobile device and must have a general understanding of the interface within < 5 minutes of use.
- The system shall be usable by mobile users on a personal computer and must have a general understanding of the interface within < 5 minutes of use.

Portability Requirements

- The system could be used on mobile platforms, through the use of an integrated web browser, such as (1) Safari, (2) Google Chrome, and (3) Mozilla Firefox.

Reliability Requirements

- The system should be able to withstand at most 1000 users (as of 01-2018).
- The system shall withstand all users with a system uptime of 98%

Security Requirements

- The system shall deny all users that provide the wrong login or registration information.

- The system shall store all login and password data in an encrypted format.

Desired Functionality

Bank Synchronization

Context

The purpose of bank synchronization is to automate the funding and consider bank accounts as wallets. The advantage of implementing a synchronization with bank accounts are a more efficient workflow, which improves the usability of the application for users.

Complications

Most banks hold a high standard as to which applications are synchronized with bank accounts. As XPENSE grows as a financing application, there will be further improvements in security standards, usability and scalability. These advancements will show banks a future opportunity for allowing XPENSE to integrate to their respective platforms as a major player in the personal financing scope.

Advanced/Visual Statistics

Context

Providing advanced statistics and graphical interpretations of data will allow users to meticulously view their expenses. This allows for further usability for experience users and new users alike.

Complications

With advanced concepts in statistics correctness is a large complication for XPENSE. It is difficult to manage a large sample of data and design complex concepts for Software Developers, as it is outside of the field of study. To overcome this issue, the XPENSE development team would require someone with a business or statistics expertise to provide and verify advanced concepts.

Examples

- Variance: What is the variance between the average spending for N months, where N is the total time (in months) since registration.
- Long-term average graphs: Provide graphical interpretation of trends towards spending habits of users, towards a longer period of time

Technical Specifications

Operating System: Windows OS, Linux, Mac OS, Android OS, iOS

Physical Platforms: Personal Computers, Android Devices, Apple Devices

Web Platforms: Google Chrome (PC/Android) , Safari, Other web applications a future implementation

Languages: HTML, CSS, Javascript

Frameworks: Node.js, Express, Angular.js

Application Dependencies: Atom (Text-Editor), MongoDB (Database Application), Google Chrome (Testing/Implementation)