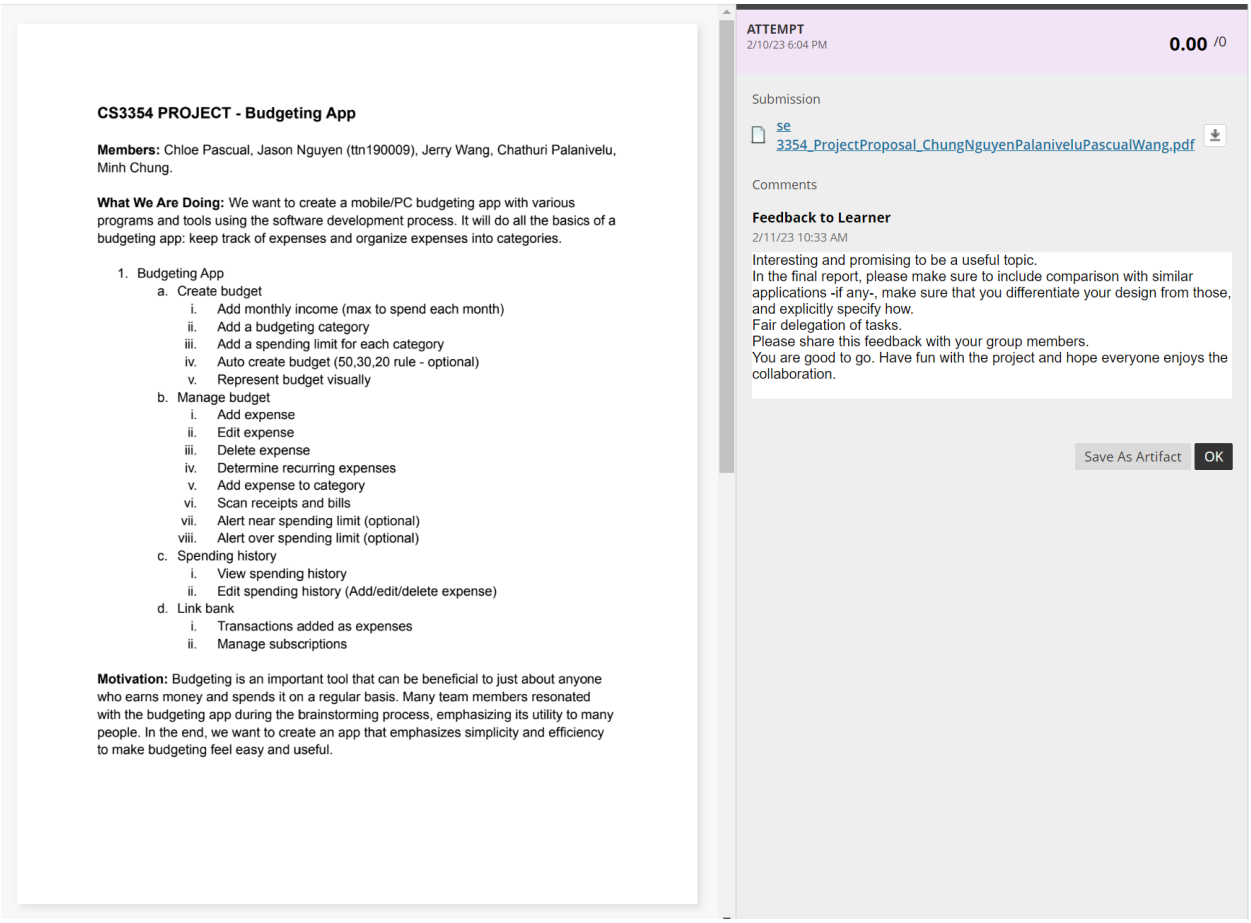
CS3354 Software Engineering
Final Project Deliverable 1

Budgeting App

Minh Chung, Jason Nguyen, Chathuri Palanivelu, Chloe Pascual, Jerry Wang

1. Final Project Draft Description with Feedback



CS3354 PROJECT - Budgeting App

Members: Chloe Pascual, Jason Nguyen (ttn190009), Jerry Wang, Chathuri Palanivelu, Minh Chung.

What We Are Doing: We want to create a mobile/PC budgeting app with various programs and tools using the software development process. It will do all the basics of a budgeting app: keep track of expenses and organize expenses into categories.

1. Budgeting App
   a. Create budget
      i. Add monthly income (max to spend each month)
      ii. Add a budgeting category
      iii. Add a spending limit for each category
      iv. Auto create budget (50,30,20 rule - optional)
      v. Represent budget visually
   b. Manage budget
      i. Add expense
      ii. Edit expense
      iii. Delete expense
      iv. Determine recurring expenses
      v. Add expense to category
      vi. Scan receipts and bills
      vii. Alert near spending limit (optional)
      viii. Alert over spending limit (optional)
   c. Spending history
      i. View spending history
      ii. Edit spending history (Add/edit/delete expense)
   d. Link bank
      i. Transactions added as expenses
      ii. Manage subscriptions

Motivation: Budgeting is an important tool that can be beneficial to just about anyone who earns money and spends it on a regular basis. Many team members resonated with the budgeting app during the brainstorming process, emphasizing its utility to many people. In the end, we want to create an app that emphasizes simplicity and efficiency to make budgeting feel easy and useful.

ATTEMPT
2/10/23 6:04 PM
0.00 /0

Submission
se 3354_ProjectProposal_ChungNguyenPalaniveluPascualWang.pdf

Comments

Feedback to Learner
2/11/23 10:33 AM

Interesting and promising to be a useful topic.
In the final report, please make sure to include comparison with similar applications -if any-, make sure that you differentiate your design from those, and explicitly specify how.
Fair delegation of tasks.
Please share this feedback with your group members.
You are good to go. Have fun with the project and hope everyone enjoys the collaboration.

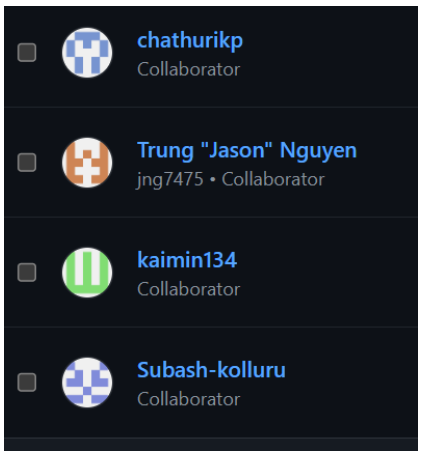
Save As Artifact OK

We will make sure to discuss similarities and differences of our app with other budgeting apps in the final report.

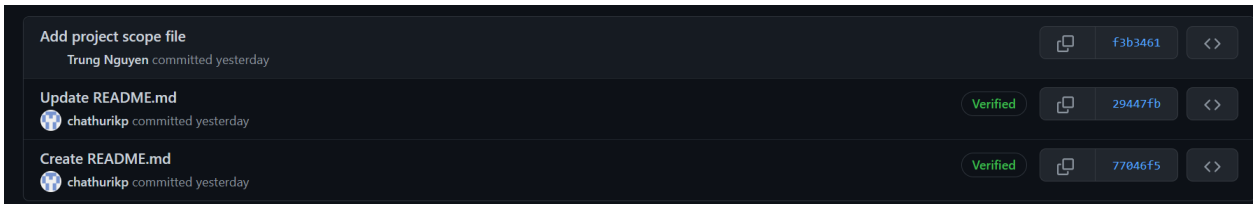
2. Github

Github Repository: https://github.com/ccmmp/3354-bugetingApp

Added Collaborators:



Commits to Repository:



3. Delegation of Tasks*

***For deliverable 1 each member worked on a section and tasks were split evenly**

- **a. Chloe**
  - i. Design ui/ux, branding, and frontend responsibilities.
  - ii. Backend help where applicable.
- **b. Jason**
  - i. Help with front-end development
  - ii. Researching and setting up most suitable database
  - iii. Implement APIs to connect app with the database
- **c. Chathuri**
  - i. Finding and implementing suitable APIs for scanning feature of the app
  - ii. Backend
- **d. Minh**
  - i. Finding how to protect privacy and security when linking bank accounts to the app.
  - ii. Frontend
- **e. Jerry**
  - i. Can help with front-end development
  - ii. Develop features for managing expenses, creating budgets, etc.

4. **Software Process Model**

We are using an incremental process which will divide the system's functionality into small increments that are delivered one after the other. Each further increment will expand on the previous ones until everything will be updated fully. This will allow us to see what does and doesn't work in the earlier increments, allowing us to fix them before going further. So, we can work in a stepwise manner and fix bugs and issues as they pop up. Since the incremental process will need a clear and complete definition of the whole system, it works for us since we know how our system will operate and its requirements.

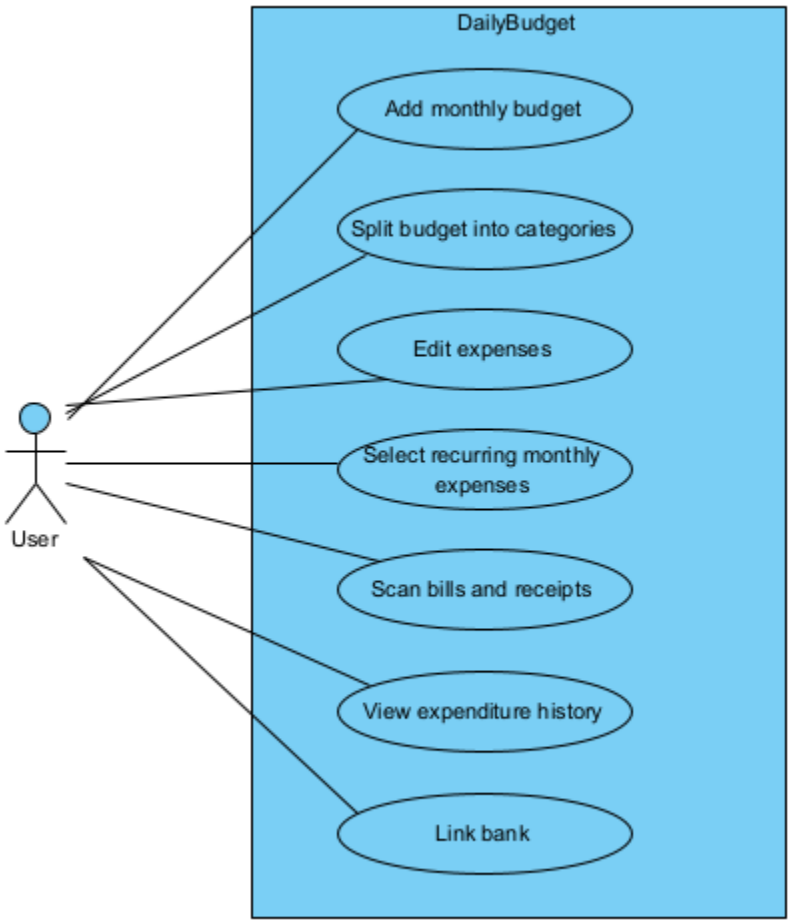
5. **Software Requirements**
    a. **Functional**

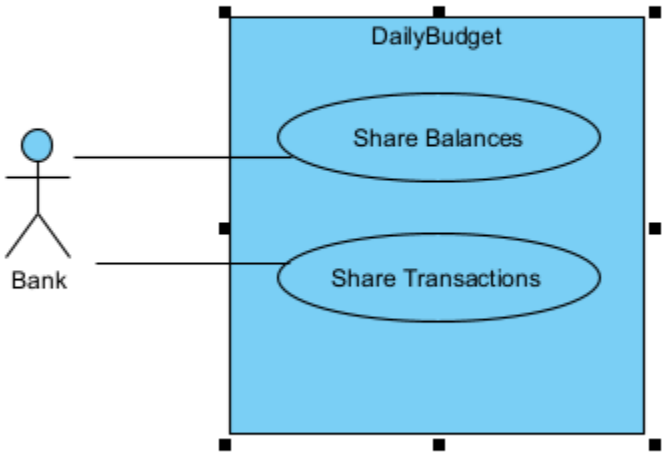
- User registration: The app should allow users to create an account and sign in securely.
- Expense tracking: The app should allow users to track their expenses by category (e.g. groceries, rent, entertainment).
- Budget planning: The app should allow users to set monthly or weekly budgets for each category.
- Notifications: The app should send notifications to users when they approach or exceed their budget limits.
- Goal setting: The app should allow users to set financial goals (e.g. saving for a vacation, paying off debt) and track progress towards these goals.
- Reporting: The app should generate reports and visualizations of spending habits and trends.

    b. **Non-Functional**

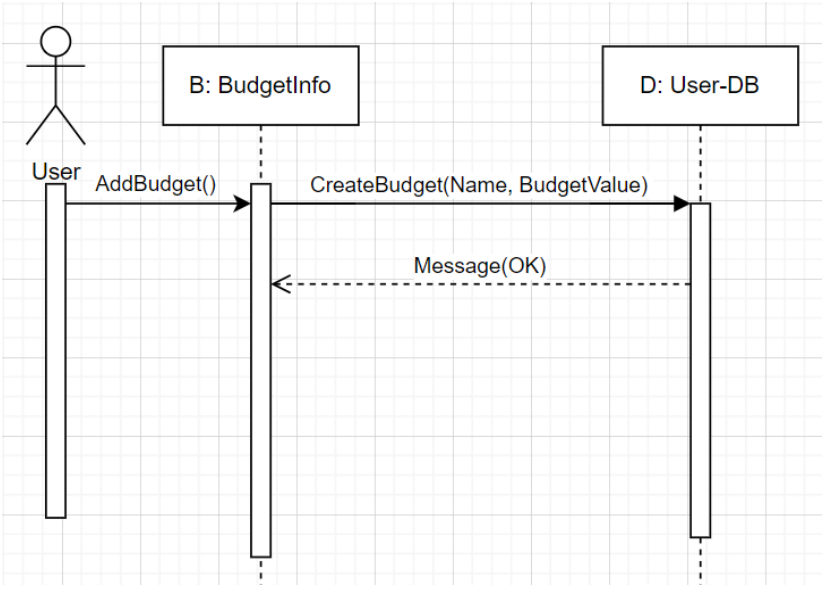
- Security: The app should be designed with strong security features to protect user data.
- Usability: The app should be user-friendly and easy to navigate, with a clear interface and intuitive controls.
- Performance: The app should be fast and responsive, with minimal lag or delay.
- Availability: The app should be available for use on multiple devices and platforms (e.g. iOS, Android, web).
- Reliability: The app should be reliable and stable, with minimal crashes or bugs.
- Scalability: The app should be designed to handle a large number of users and a growing database of financial data.

6. Use Case Diagrams

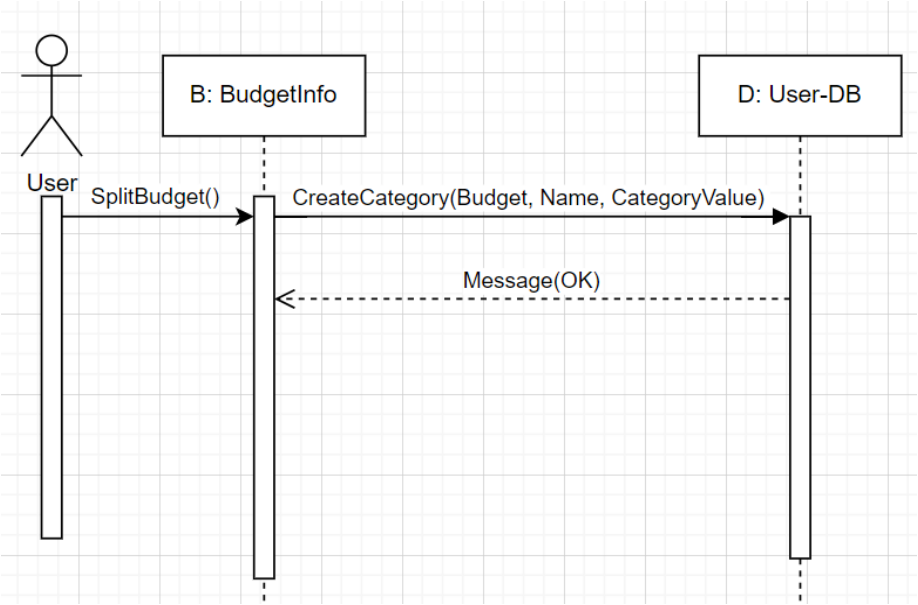




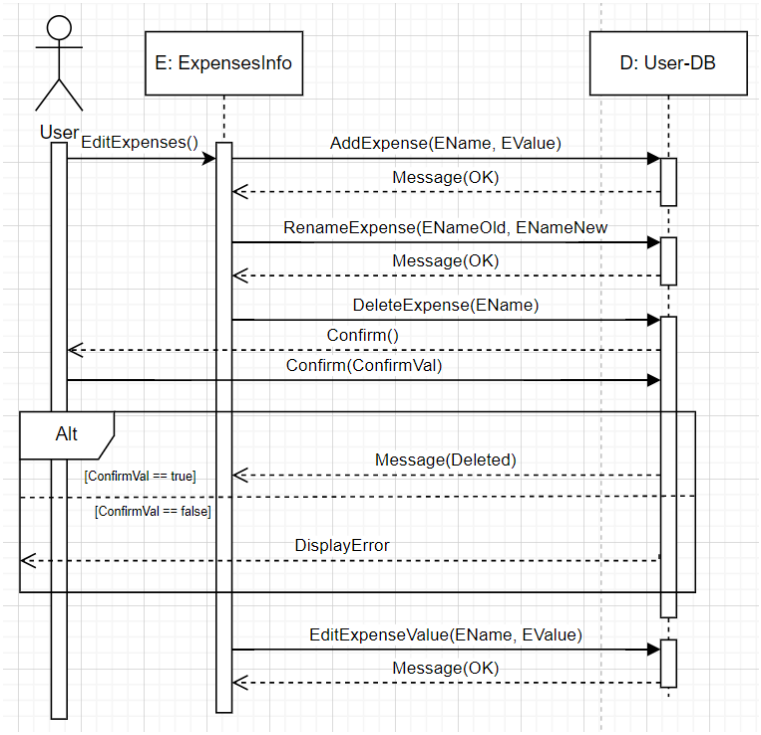
7. Sequence Diagrams



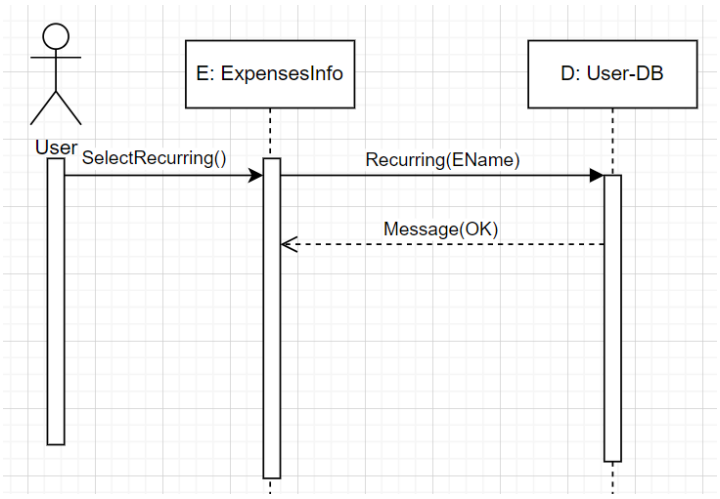
User: Add Monthly Budget



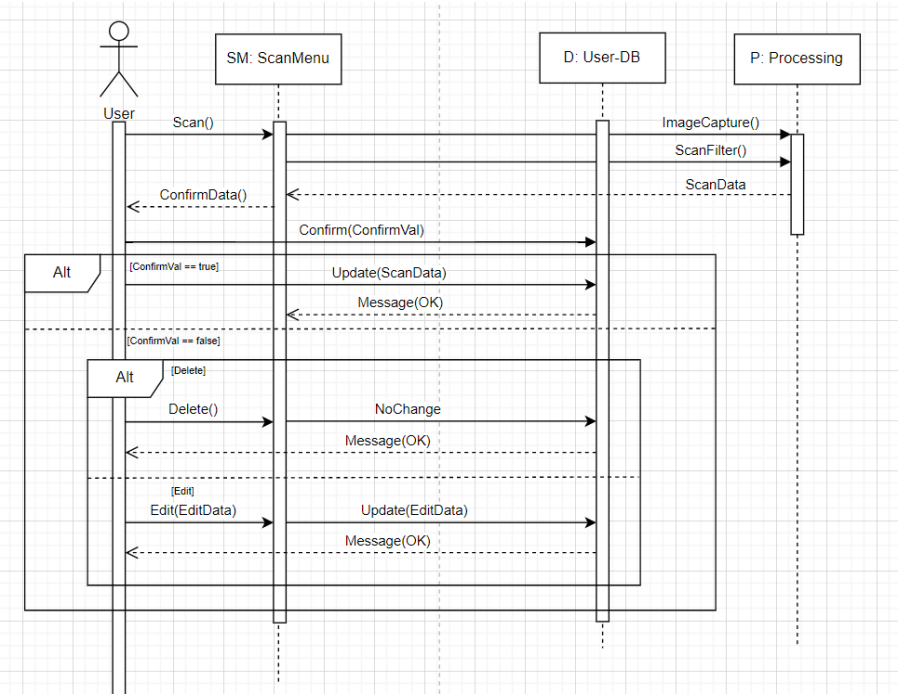
Uder: Split Budget Into Categories



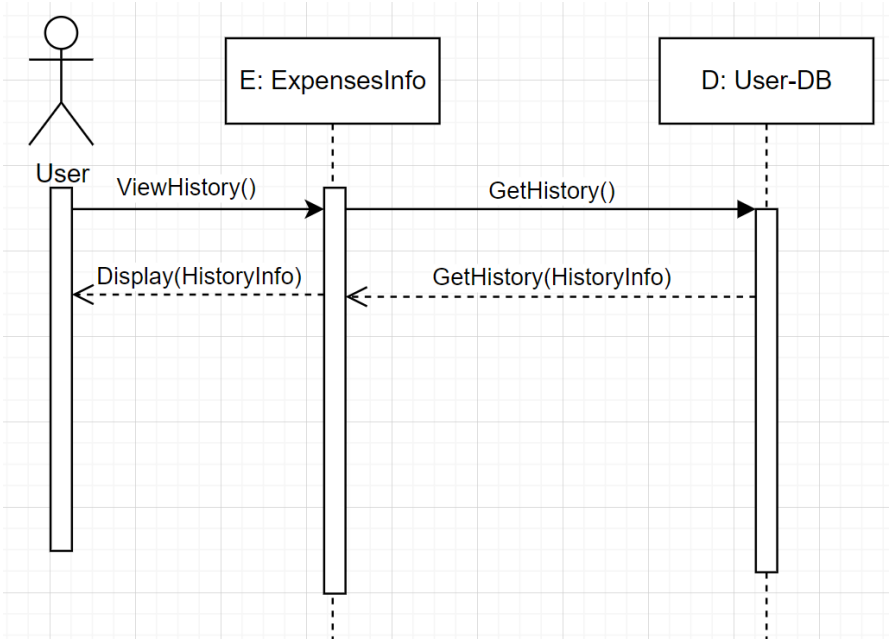
User: Edit Expenses



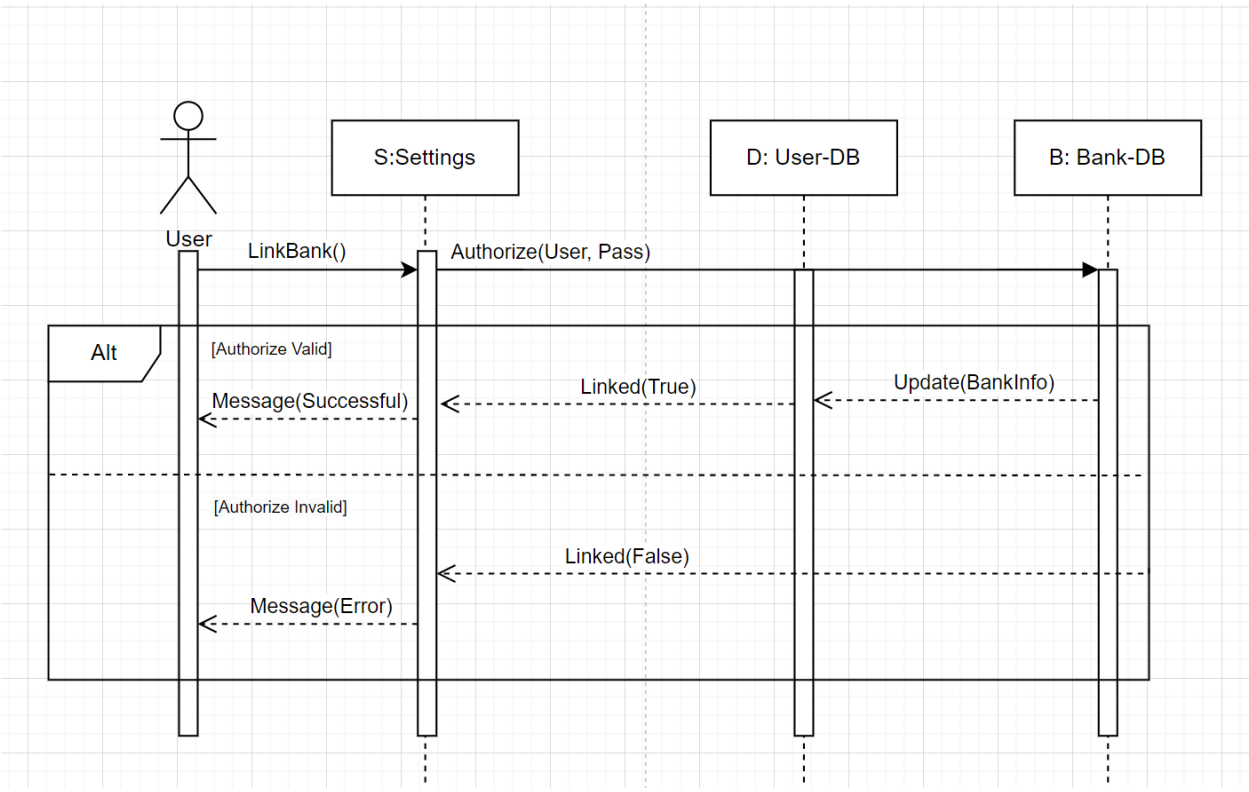
User: Select Recurring Expenses



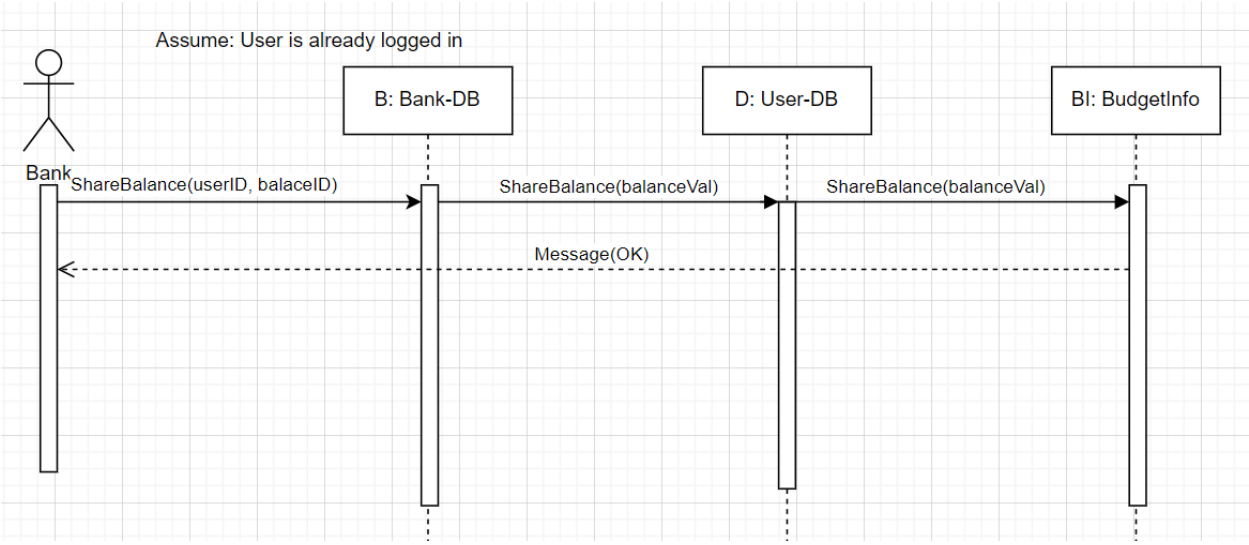
User: Scan Bills and Receipts



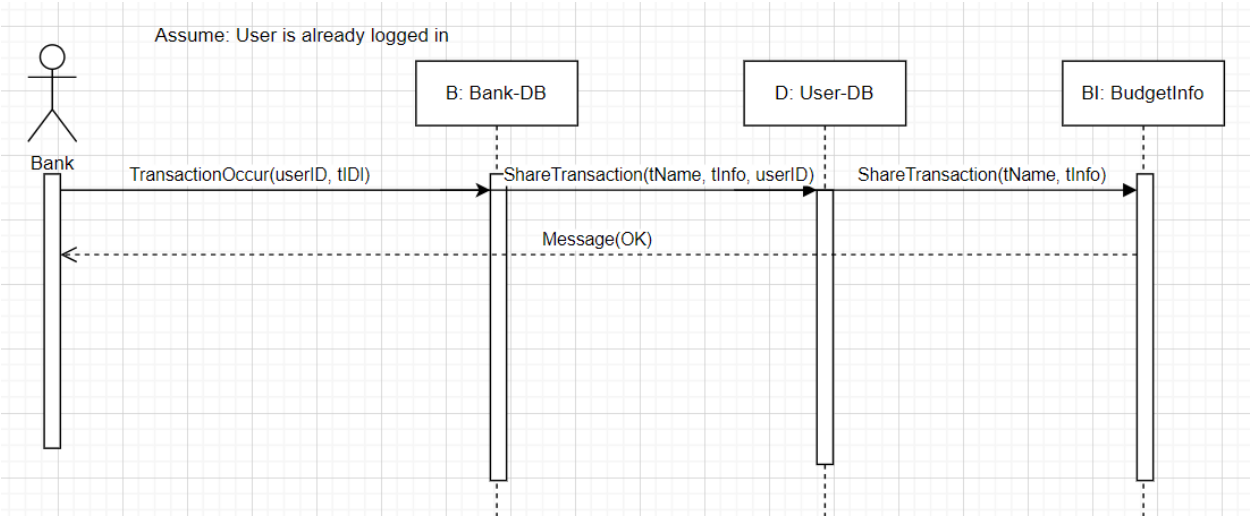
User: View Expenditure History



User: Link Bank

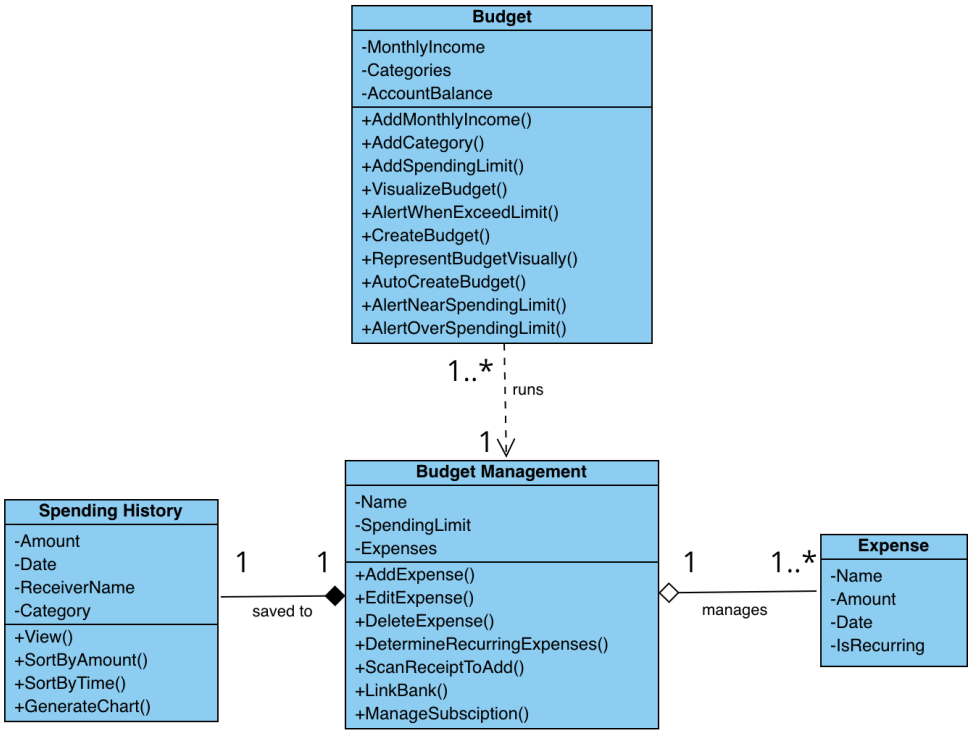


Bank: Share Balances



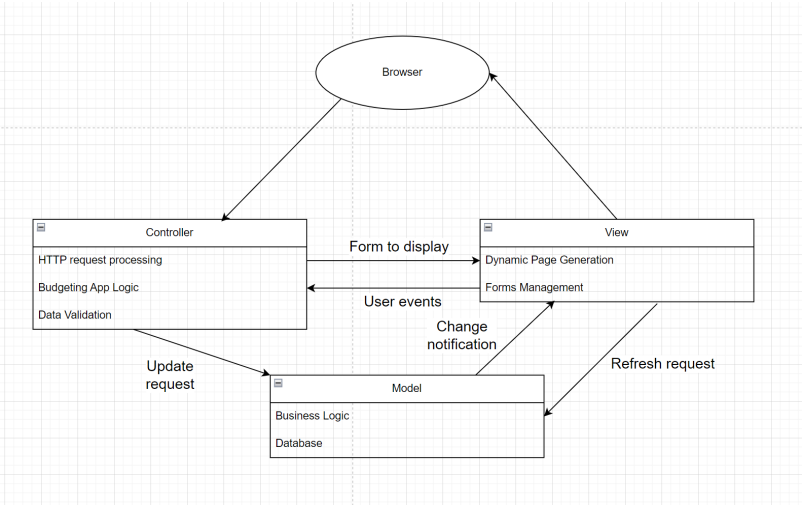
Bank: Share Transaction

8. Class Diagram



9. Architectural Design

For the architectural design of this project we decided to go with Model-View-Controller (MVC). This budgeting web-based application handles a good amount of data for each user, displays that data, and that data can be interacted with in a number of ways, because of that MVC is a nice fit.



Architecture for web-based budget app