Team: JaJeJu

Roster: Jackson Zou -- Jeff Lin -- Jun Tao Lei

Project: Spendie

P2

Summary

The premise of this project is to design a site that tracks one's spendings over a period of time. The site will be able to display statistics for total transactions over a month such as spending breakdowns, total spendings, deviation from spending targets, etc. There will be tags that link to the different custom categories of transactions over a month.

A calendar will be included to display transactions over a month. The site will be able to keep track of loans or debt that may be viewable to other people or keep track of certain notes or todos. Users can also search for certain customized tags.

Front-end Framework

Bootstrap will be used as our front-end framework as it is the more familiar framework.

Components

- Add Transaction
    - Defaults to current date
    - Requires amount and transaction name
    - Also can store location and a note
- Spending Statistics
    - This shows the spending breakdown of all the transactions made in a calendar month based on the category of each transaction.
    - Statistics:
        - Average daily and monthly spending
        - Sum of amount spent in week and month
- Tags
    - Tags can be created and are used to categorize transactions
    - Allow the user to organize all transactions under one specific category or none
    - Searchable field
- Transaction

    - Each transaction is a collection of data including the transaction name, the transaction amount, the date, the location, and a note/comment
- Calendar
    - The calendar can track all transactions made in a calendar month
        - Each square on calendar displays the total amount spent in day
        - Hue of red represents spending amount relative to other days of the month (becomes more comprehensive over time as more data is collected)
        - Clicking on square opens page with all transaction details of the day
    - May be easier to build with external javascript framework
- Search
    - Users can search for a transaction based on title, date, note, or tag
- Todos
    - A list of todos (simple notepad)
- Debts/Loans
    - Requests
        - Give/get reminders for money to/from other users
        - Requires specific user_id to request
        - Will show notifications for any of these requests
        - Allows custom message

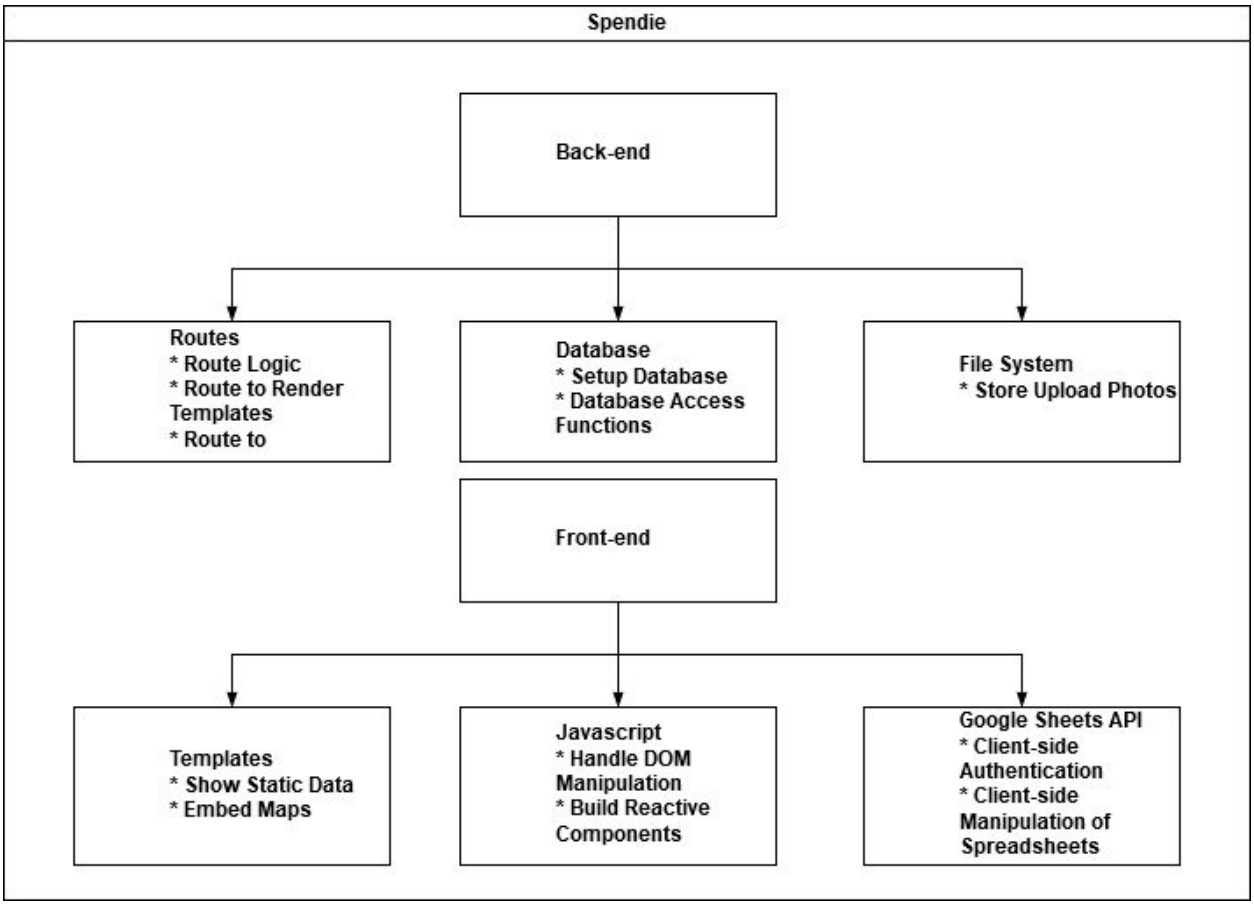
API Integration

- Integration with the Google Sheets API and the Google Maps Embed API
    - The Google Sheets API allows the user to export his/her data to his/her Google account.
        - Requires Google's external javascript file to authenticate, create, and manipulate spreadsheets client-side.
            - https://apis.google.com/js/api.js
    - The Google Maps Embed API allows the user to view the location where the user made a transaction.

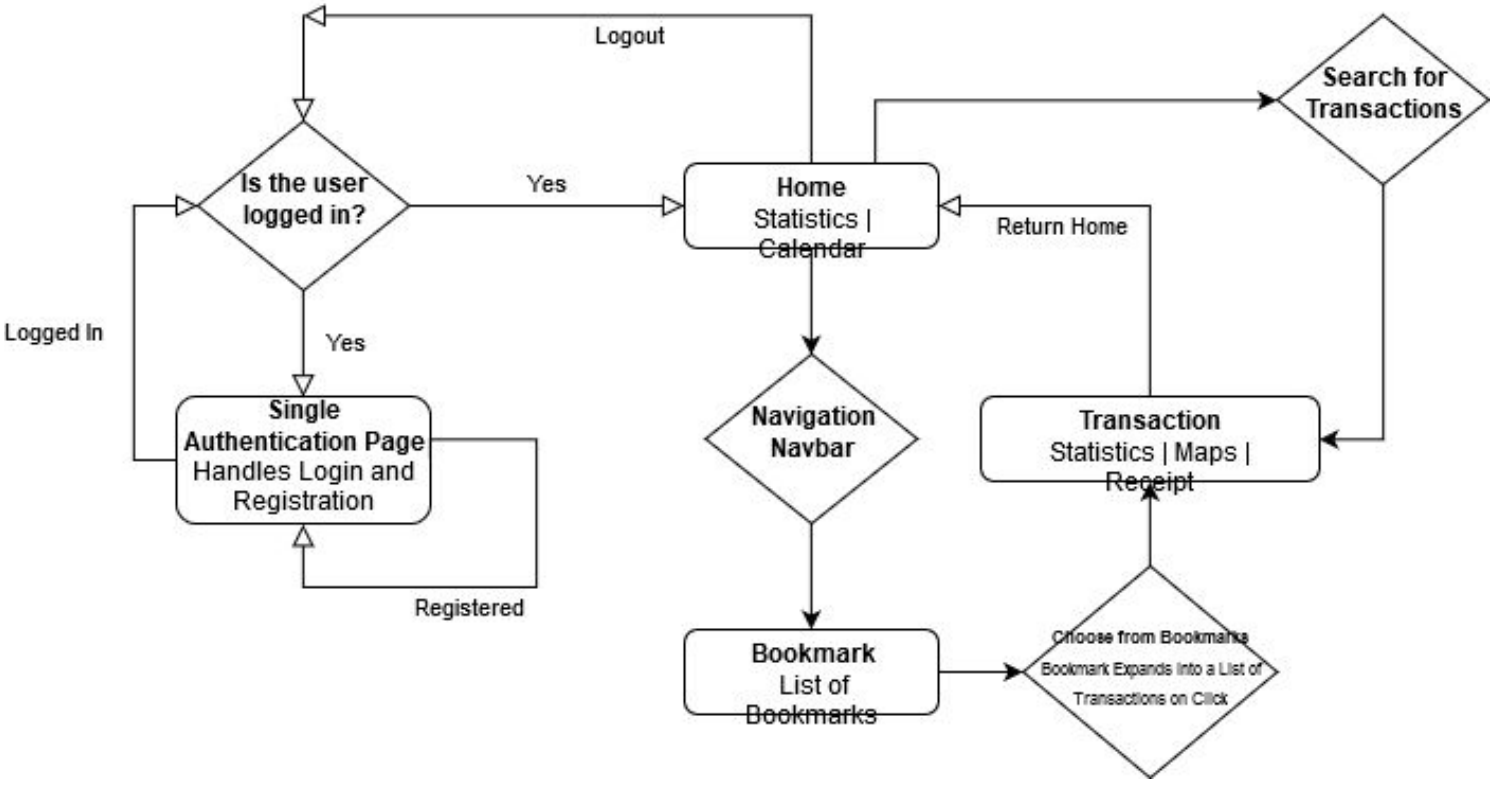
Component Map

- Frontend:
    - Javascript will allow for user interaction with the site

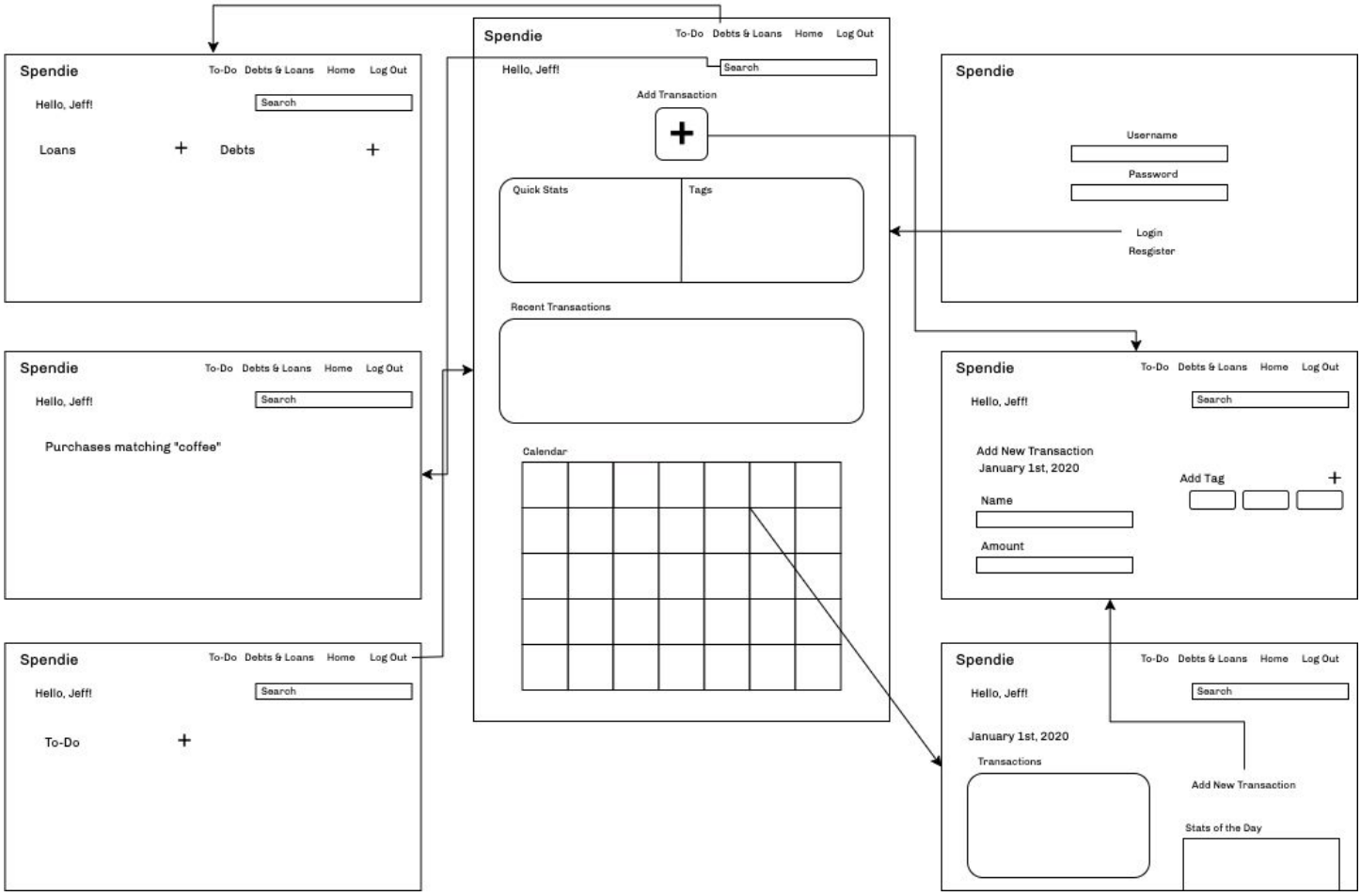
    - Bootstrap will be used to format site and give aesthetics
- Backend:
    - Flask will serve as an intermediary between front end-user interaction and database
    - SQLite databases will store user inputs and return information as necessary


Spendie
Back-end
Routes
* Route Logic
* Route to Render Templates
* Route to
Database
* Setup Database
* Database Access Functions
File System
* Store Upload Photos
Front-end
Templates
* Show Static Data
* Embed Maps
Javascript
* Handle DOM Manipulation
* Build Reactive Components
Google Sheets API
* Client-side Authentication
* Client-side Manipulation of Spreadsheets

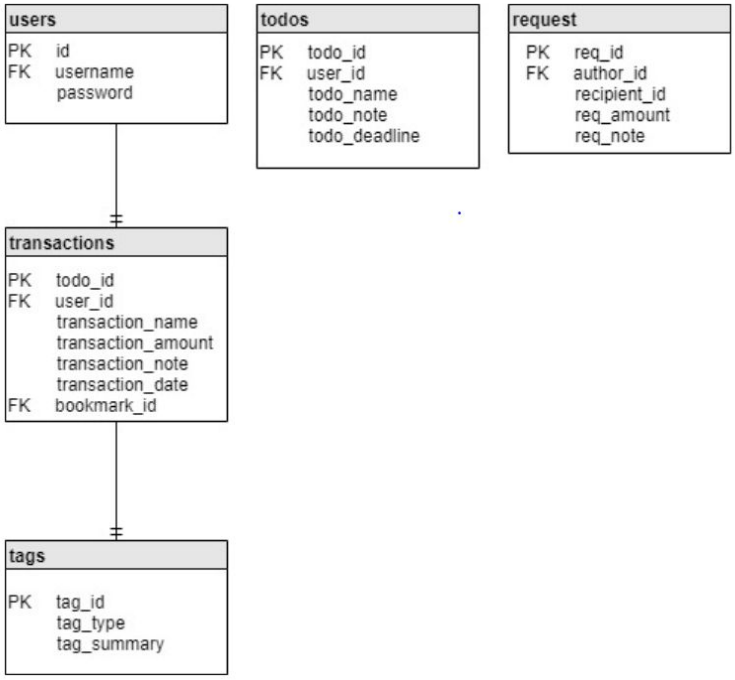
Site Map


Logout
Search for Transactions
Is the user logged in?
Yes
Home
Statistics | Calendar
Return Home
Logged In
Yes
Single Authentication Page
Handles Login and Registration
Navigation Navbar
Transaction
Statistics | Maps | Receipt
Registered
Bookmark
List of Bookmarks
Choose from Bookmarks
Bookmark Expands into a List of Transactions on Click

Design



Database Diagram



Design

This application will have some distinction between the client-side and server-side functions. The application will serve server-side rendered templates and static javascript files. Besides serving some static javascript files, the application will be monolithic.

For the most part, the application front-end will serve server-side rendered templates for trivial operations like viewing transaction statistics. However, static javascript files will be served for interactive components like the calendar. Although a template would probably still be used for such components, the template HTML will link to the javascript files.

The application back-end will handle database access operations, file system storage operations, and authentication.

While the Google Maps Embed API can be used by embedding the template with an iframe, the Google Sheets API would probably be implemented client-side. This is because it would make more sense to have Google authentication be performed client-side to avoid building Google authentication into the user flow. Therefore, it would be likely that the client would fetch data from the application back-end using Javascript, and populating the Google spreadsheet with that data.

Roles

- Jackson
    - Back-end
        - Flask Routing Logic
            - Check if a user is logged in, etc.
        - Flask Routes for Transactions and Tags
            - Query Database
- Jeff
    - Front-end
        - Templates and Styles
            - Bootstrap and CSS
- Jun Tao
    - Back-end
        - Setup Database
        - Setup Authentication
    - Front-end

- Javascript
- Client-side Google Sheets API