CS 180 Split Team Design Document

Overview

Split household bills with roommates, to figure out costs for a group vacation, or just to remember when a friend spots you for lunch, view your balances, track spending trends, set up email reminders for bills, and much more!

Design

Database

Using Firebase, we will have our data organized as in the provided ER diagram.

Currently everything is stored in the requests collection in firebase. A request consists of:
User
Recipient
Amount Owed
Comment
Closed Status
Timestamp

User

Our users are created and stored using Firebase authentication

Request

The request class has a unique request ID number, amount requested, tags pertaining to the request, a creator and recipient. Users will be able to send requests, send reminders, as well as close/complete requests when they are finished.

Bare Minimum Request:

> Recipient:
> Amount:
> Comment: (optional)

The open and closed requests are displayed separately for the user. The user has the capability to close a request from the UI.

UI

The UI is built using bootstrap. Currently we have the home page for when no user is signed in. From here a user can sign up or log in. Each of these has their own page as well.

Then we have the main dashboard page which displays all of the users open and closed requests as well as the total amount owed in the open requests.

We will finally create a user page to display the extra details for a user account that we add later.

Alternatives Considered

We originally used no UI framework but we realized it was not scalable as we needed to add more functionality.

Testing

Unit Testing will be performed using a JavaScript testing tool. (probably one of this list:) https://www.pixelcrayons.com/blog/web/top-5-javascript-testing-tools-for-your-web-developer/

*(Testing should be completed before branching to github)

Roadblocks

Our biggest roadblock has been figuring out how to use asynchronous nature of firebase, but once we figured that out it has been much smoother sailing.