Profile

Coded mostly by Alex Redden

Not Yet Implemented/Issues:

-The search feature in the friends panel.

-When I started coding this I did not have a good understanding of react or javascript and so I made a lot of redundant components and functions, so even though it is supposedly functional I will eventually remove/alter a lot of these components.

Currently implemented react components:

Profile:

displays empty profile page with image, user name, phone number and email of user.

feeds invited, attended, not attended previous parties (sorted beforehand by attended, .invited, not attended) to ProfilePreviousAtt, ProfilePreviousInv and ProfilePreviousNat.

feeds current userid to ProfileInvitedPartiesIntermediate**.**

feeds hosted party ids to ProfileHostedParties.

feeds all friend ids to ProfileFriends.

ProfilePreviousAtt: displays parties previously attended in the previous parties table.

ProfilePreviousInv: displays parties previously invited to in the previous parties table.

ProfilePreviousNat: displays parties invited to but not attended in the previous parties table.

ProfileInvitedPartiesIntermediate: sorts parties by whether the user has been invited to, going to, or declined the invite and feeds to ProfileInvitedParties.

ProfileInvitedParties: feeds party information to DeclinedRow, InvitedRow, and GoingRow, each of which display party information in 'invited to' table.

DeclinedRow: Displays information about parties invited to but declined.

InvitedRow: Displays information about parties invited to and not responded.

GoingRow: Displays information about parties invited to and accepted.

ProfileHostedParties: Displays information about future parties that the user is hosting in the hosting table.

ProfileFriends: Displays one user friend as a table cell which contains a name and profile image in the friends table.

Currently implemented server functions:

getParty: returns all info about a party

getUserName: returns the first and last name of a user

getHostedParties: returns party ids of parties hosted by user

getInviteInfo: returns party objects that user has been invited to

getDeclinedInfo: returns party objects that user has declined attending

getGoingInfo: returns party objects that user is attending

getPrevParties: returns party objects that user has either attended, not attended or been invited to and ignored in an array called prevParties.

Host:

Coded by Mira and Cheyan

Functionality of Host:

- The Host page is where users can register their party which will get stored in the data base. The properties of a party include the *content* which include name of the party, where it is hosted, zip code etc. The *supplies* and invited *guests* list which are just empty arrays that the user keeps adding onto depending on what supplies and which guests they want to attend the party. Also, I have a supply and user value that keeps track of the elements that will get added or deleted onto the the supplies and guests array.

Not Yet Implemented/Issues:

- When the user issues a party, they are asked to input in the people they want to invite and the supplies that they need. Both the invited guests and the supplies are properties in the party object and are initially empty arrays in which the user will dynamically add elements onto the array . Ideally, I would like the user to click enter and for the invited guests and supplies to be added onto the array and to display onto the appropriate tables. However, I was not able to implement this at this time, and just have a button in which the user has to manually press in order for the invited guests and supplies to be added to the empty arrays and updates on to display on the tables. I plan on fixing my issues by reading more JavaScript documentation.

Implemented React Components:

- Host: This holds the majority of the functionality which includes posting a party, and adding the guests and supplies for that party
- Host-Item: Responsible for deleting guests and supplies for a party
- Page: Responsible for connecting posted parties onto the data base

Implemented Server Functions:

- postParty: iterates through the inputs and calls the creates new party to make the party
- addInvitee: adds the value inputted for guests onto the array
- addSupply: adds the value inputted for supply onto the array
- handleChange: update value in response to user input

[No functionalities were cut out, and the mock objects are the same ones submitted with the ER Diagrams]

Party-Info

Coded by Ed Fennessey

React Components:

- PartyInfo
    - Displays the main party information (address, host, description) from mock database
    - Displays the PartyInfoInvited React Components
    - Contains toggle buttons and utilizes server methods for changing the open/private status of the party in the mock database and dynamically updates the displayed information accordingly
- PartyInfoInvited
    - Shows users who have been invited to the party in a cell
    - Renders proper status of user invite (going, pending, declined)

Server Functions:

- getPartyData
- getInvitedData
- setPartyPrivate
- setPartyOpen

Not Yet Implemented:

- The tabs for “Supplies” and “Complaints” do not yet function. This will probably require separate React Components to be made for each of the three tabs to be contained within the PartyInfo component. The information displayed on this tabs will not be difficult to implement once these components are restructured.
- Badge on the “Complaint” tab is hardcoded in and does not reflect the actual number of complaints a party has received
- Search bar to find friends and invite them to the party is not implemented

Complaint:

Coded by Vlady

Functionality of Complaint:

- This page is used for when a user wants to complain about a loud party near by.

Implemented React Components:

- Complaint: consists of a map image that depicts the locations of the parties nearby and several Address-Entry objects depending on how many parties are nearby
- Address-Entry: imports a function from the server that allows it to pass down an address and return information about parties that are nearby. This information is then passed down to the address entry component, which displays the proper addresses for all of the nearby parties

Functionality of React Components:

1. This information from the complaint page is passed down to the address entry component, which displays the proper addresses for all of the nearby parties
2. The address entry object retain all of the information about a certain party and upon a click from the user displays a Modal where the user can file the complaint
3. Once entered the complaint is written onto the database to the correct party.
4. This is achieved by importing a function from the server that takes in a complaint and a party id and writes to the database.
5. Furthermore, the Modal has two buttons, a ‘submit’ and a ‘close’ one
6. Once the user clicks on the submit button the complaint is entered in the database and a “complaint submitted” message appears in the textbox.
7. Once the user closes the modal he/she return to the complaint page.

Not Yet Implemented:

- Separate modal for confirming complaint sent not yet implemented

Admin & HomePage & Page

Coded by Cheyan Setayesh

React Components:

- AdminPage
    - Display administration information which is essentially all data in the database.
    - Components are displayed using react-griddle which is a lightweight table library
    - Contains option to swap between user and party information
    - On a row being clicked the appropriate row information is displayed in a modal
    - Rows are background colors are adjusted based on number of complaints (per user object) or time until party (red - it has passed, yellow - 24 hours until party, white - 24+ hours until passed)
    - Settings options allows to hide or show columns
    - Rows are filterable based on any column data
- AdministratorRow
    - Allows for the administration option to be edited
    - Renders proper status of user invite (going, pending, declined)
- AvatarRow
    - Custom griddle row for displaying the user avatar
- HomePage
    - Simple html index page with links to all appropriate pages
- Page
    - Navbar with links to appropriate pages

Server Functions:

- getAuthorData
- getAdminInformation

Database Functions

- readCollection

Not Yet Implemented:

- Administration option not editable this is due to the griddle column meta data not being dynamically updated, this is a bug in the library.
- CSS of settings and show parties is inconsistent

Updated ER Diagram: