Hang Out

Team Left you on CEEN: Emma, Nicholas, Celine, Elina

Idea

Our project, Hang Out, is an app that will allow people to find and create events near them. It incorporates the Ticketmaster Discovery API to help people find these events, and will also incorporate the Google Maps Directions API to allow other people to get to these events. This app will also keep track of the events that users go to, and help introduce users to users attending similar events so that they can attend and enjoy events together.

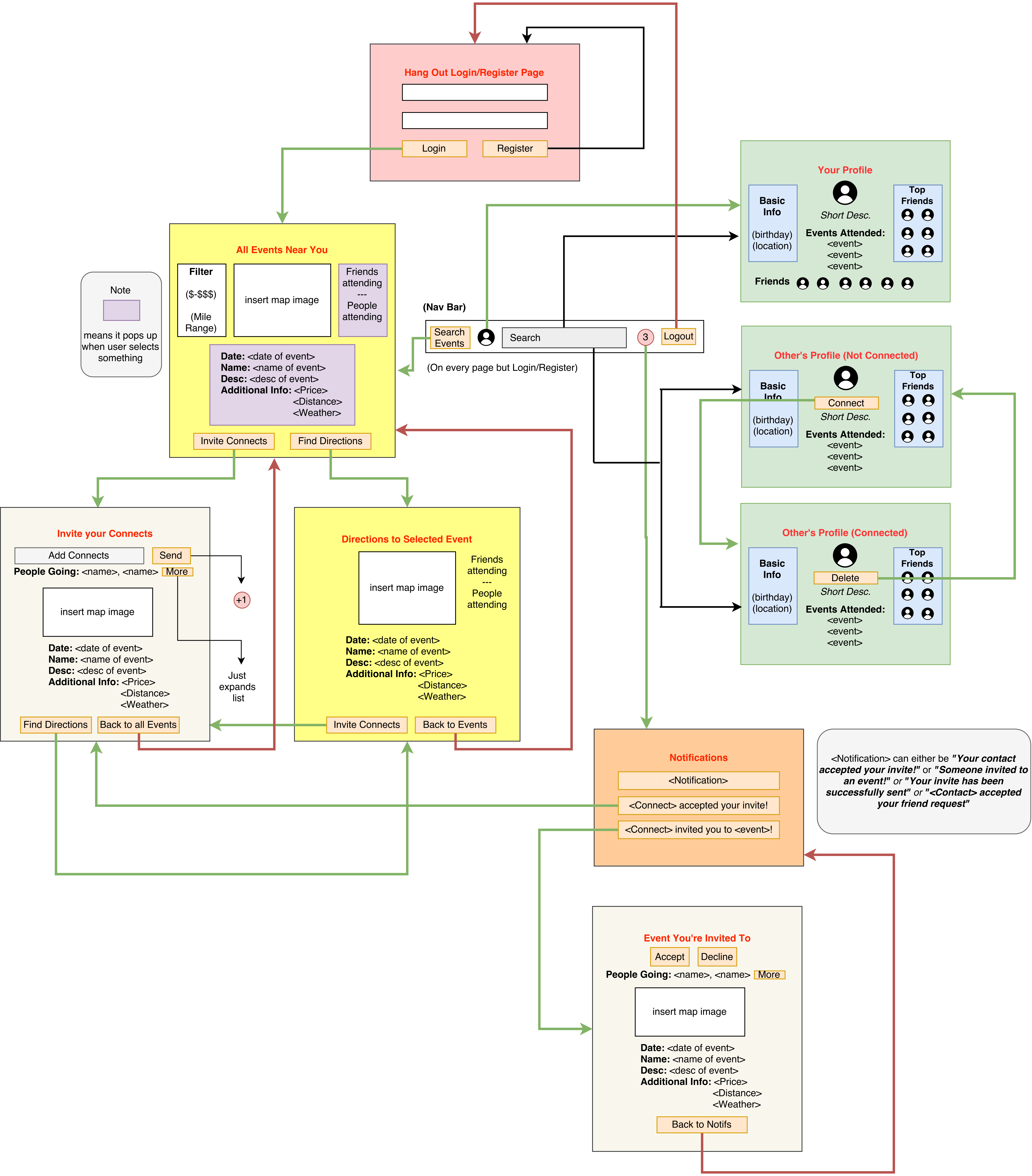
Details

- Create a master practical site where users can search for events hosted on TicketMaster within the New York City area, or create their own events that their friends can see. These events will show up on a Google Map with markers on the locations.
  - Events include fun, tech, food, etc; Basically, the event types TicketMaster offers alongside the types of events users are creating.
  - Users can:
    - Customize/filter what events they are interested in by category
    - Select the range (e.g 0-5 miles, 0-25 miles)
    - Select price range (e.g Free - $5)
    - Search for directions across several travel modes
    - Have user login, create profile, upload profile pictures
  - Having an account means being able to:
    - Record and display past events they've went to
    - See users who also attend similar events to befriend/connect with them and attend events together!

Framework

We will be using the Bootstrap framework to beautify our site.

Site Map


Hang Out Login/Register Page

Login

Register

All Events Near You

Filter

($-$$$)

(Mile Range)

insert map image

Friends attending

People attending

Date: <date of event>
Name: <name of event>
Desc: <desc of event>
Additional Info: <Price>
<Distance>
<Weather>

Invite Connects

Find Directions

Note

means it pops up when user selects something

(Nav Bar)

Search Events

Search

3

Logout

(On every page but Login/Register)

Your Profile

Basic Info

(birthday)
(location)

Short Desc.

Events Attended:
<event>
<event>
<event>

Top Friends

Friends

Other's Profile (Not Connected)

Basic Info

(birthday)
(location)

Connect

Short Desc.

Events Attended:
<event>
<event>
<event>

Top Friends

Other's Profile (Connected)

Basic Info

(birthday)
(location)

Delete

Short Desc.

Events Attended:
<event>
<event>
<event>

Top Friends

Invite your Connects

Add Connects

Send

People Going: <name>, <name> More

+1

Just expands list

insert map image

Date: <date of event>
Name: <name of event>
Desc: <desc of event>
Additional Info: <Price>
<Distance>
<Weather>

Find Directions

Back to all Events

Directions to Selected Event

insert map image

Friends attending

People attending

Date: <date of event>
Name: <name of event>
Desc: <desc of event>
Additional Info: <Price>
<Distance>
<Weather>

Invite Connects

Back to Events

Notifications

<Notification>

<Connect> accepted your invite!

<Connect> invited you to <event>!

<Notification> can either be ***"Your contact accepted your invite!"*** or ***"Someone invited to an event!"*** *or* ***"Your invite has been successfully sent"*** *or* ***"<Contact> accepted your friend request"***

Event You're Invited To

Accept

Decline

People Going: <name>, <name> More

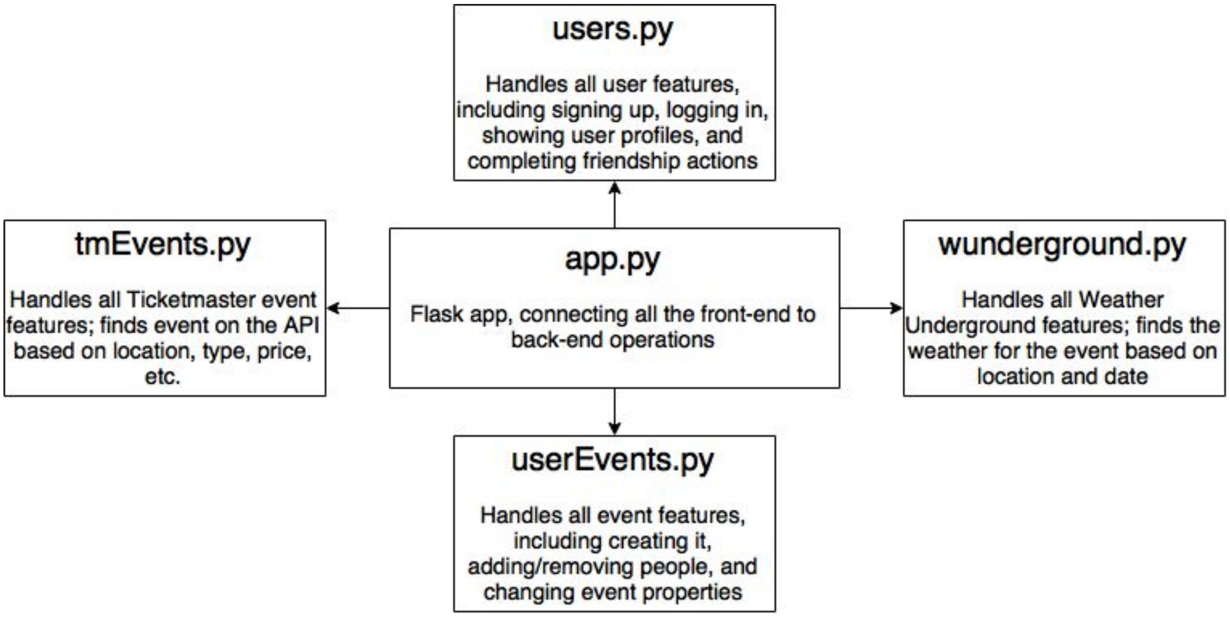
insert map image

Date: <date of event>
Name: <name of event>
Desc: <desc of event>
Additional Info: <Price>
<Distance>
<Weather>

Back to Notifs

- **Login.html** - Where a user can create an account or login to their account
- **Main.html** - Displays a map with events around you, friends, and event info.
- **Directions.html** - Loads the event info and accesses the APIs for directions and weather
- **Create.html** - Allows you to create an event and invite friends
- **Notifications.html** - Notifies user of new information regarding friends or events
- **Your profile/Others' profile (connected)** - Shows user info such as events attended
- **Other's profile (unconnected)** - Shows limited user info

Components


users.py

Handles all user features, including signing up, logging in, showing user profiles, and completing friendship actions

tmEvents.py

Handles all Ticketmaster event features; finds event on the API based on location, type, price, etc.

app.py

Flask app, connecting all the front-end to back-end operations

wunderground.py

Handles all Weather Underground features; finds the weather for the event based on location and date

userEvents.py

Handles all event features, including creating it, adding/removing people, and changing event properties

- **TicketMaster Discovery API:** Can provide events by keyword, time, location, etc. which we will use to get larger events that may be of interest to users and their friends (as opposed to user-created events, which will only have as guests the people the user-creator has invited)
- **Google Maps:** Will allow us to embed a map into the homepage, with markers showing the location of recommended events
- **Google Maps Directions API:** Can provide directions from starting point to final destination by several travel modes (walking, driving, bicycling, transit) with the expected duration of travel. We will use this to help users know how to get to and from events they might attend, helping to plan their trip.

- **Weather Underground API:** Can provide information to users regarding weather, which will allow them to decide whether or not they should hold/go to the event on that day.
- **app.py:** Bridge between the front- and back-end, which will take information given through the front-end, process it along the proper route/proper utils function, and bring this back to the front-end side, back to the user.
- **users.py:** will be used to handle action related to user accounts: from login/register authentication, to generating the content of each user's profile, to executing friendship actions.
- **TABLE users:** will store the information users.py relies on
- **userEvents.py:** will be used to handle actions related to user-generated events: creation, adding/removing attendees, changing properties.
- **TABLE userEvents:** will store the information userEvents.py relies on
- **tmEvents.py:** will be used to handle related to ticketMaster events and the Ticketmaster Discovery API, mainly getting events from API based on user-specified parameters (location, type, price, etc.).
- **wunderground.py:** will be used to handle the weather actions and the Weather Underground API, mainly finding the weather for the day of the event based on location and date.
- **TABLE friendships:** will store information about friendships for use by any/all pages

Database Schema

TABLE users

TEXT username	**TEXT email**	**TEXT salt**	**TEXT pass**	**TEXT imgLink**	**TEXT eventIdList**	**TEXT types**
Stores usernames	Stores corresponding email addresses	Randomly generated string added to pass pre-hash	Hashed pass	Link to profile image of user	List of ID's of events (TicketM ID) attended by user, each ID separated by a '-'	List of types of events user enjoys separated by a '-'

TABLE friendships/connects

TEXT initiator	TEXT friend	TEXT status	INT friendaversary
Username of user who requested friendship with ‘friend’	Username of user who ‘initiator’ requested to be friends with	Status of friendship: pending (requested, not yet accepted), declined (requested, rejected), or accepted (requested, accepted)	The date on which the friendship was accepted. Stored as mmddyyyy

TABLE userEvents

TEXT name	TEXT creator	TEXT desc	INT date	INT time	INT ID	TEXT attending
Name of event	Username of event creator	Description of event (by creator)	Date of event as mmddyyyy	Time of event as hhmm	Event ID	String of usernames of attendees separated by ‘-’

Division of Labor

Emma - Project Manager, Back End
Elina - Back End
Celine - Front End
Nick - Front End w Back End

Timeline

12/6 - Finish Login System and Database System
12/7 - Finish create and view events templates with Ticketmaster API
12/8 - Integrate Google Maps API and Weather API
12/9 - Create Profile display templates
12/10 - Add additional features
12/11 - Finish Project