Reo K, Ely S, Dhiraj P, Ziyan L
Software Development - Period 6

API Project Design

Overview

- Our website is an itinerary planner that helps users create a basic schedule that consists of an event + a place to eat
- We will be using the first two API for core functionality, and the latter ones if time permits:
    - Eventbrite API (https://www.eventbrite.com/developer/v3/)
    - Google Places API (https://developers.google.com/places/)
    - Google Maps API (https://developers.google.com/maps/)
    - Open Weather Map API (https://openweathermap.org/api)

User Interface/Experience (UI/UX)

- An account is necessary to use the website!
- The login page is standalone, in that you can't login from the homepage
- On the homepage, users are given a form to narrow down the event search (current address, radius, budget, date) and a navbar overhead with buttons for viewing their plans, settings, and logging out
- When the form is submitted, the website loads a new page displaying a list of events (with relevant info: time, address, price, etc.) that meet the criteria provided in the previous form
- Once an event is selected, a new page is loaded, displaying a list of nearby places to eat, again determined based on the criteria provided in the initial form
- Once an eatery is selected, a new page showing the summary of the new schedule is shown (total cost estimate, time frame, etc.) and the user can return home using a button at the bottom + *an embedded google map displaying the whole trip
- The schedule is saved for later perusal via the 'My Plans' button in the home page

Project Components (Tentative)

login.html

- Allows user to login to an existing account or create a new one

home.html

- Loads a list of schedules the logged-in user has created in chronological order
- The page includes a navbar containing buttons: one for logging out and another for creating a new event

newEvent.html

- Displays an HTML form whose parameters are used to generate a list of events; submitting the form sends data to and loads listEvents.html based on the user's criteria

listEvents.html

- Generates a list of events using the Eventbrite API in conjunction with the information provided in the homepage's HTML form
- Each entry in the list contains relevant information pulled from Eventbrite (e.g. location, time)
- Selecting any event sends users to listFood.html

listFood.html

- Generates a list of eateries using the Google Places API in conjunction with the information provided in the homepage's HTML form
- Each entry in the list contains relevant information pulled from Google Places (e.g. cost)
- When picking eateries, assumes that *at least 2 hours* will be spent eating (to avoid the issue of arriving at an eatery just about to close)
- Selecting any eatery sends users to summary.html

summary.html

- Shows a summary of the schedule by combining information from the event and eatery previously selected
- *displays embedded google map with route from original address to event to eatery
- Saves schedule in his/her database table
- Contains a button that sends the user back to home.html

app.py

- The master Python file: calls upon the util Python files to carry out various functions of the site. Note both Ely and Reo will work on this

auth.py

- Takes care of account creation, and logging in/out
- When registering, validates that the username of the new account is unique by going through the *users* table in the database file

dispEvent.py/dispFood.py

- Retrieves data from the Eventbrite API and/or Google Places API (depending on the python file) based on the fields provided in the homepage's field
- Gives a list of events/eateries with information about each

dispSummary.py

- Displays a summary of the logged in user's most recent itinerary.

dispPlans.py

- Used in conjunction with myPlans.html to display all the schedules of a specific user
- Finds the table of the logged-in user within data.db and retrieves all of its contents for display

data.db

- Contains tables with information pertaining to registered accounts and individual account's schedules (see Database Schema below)

Database Schema

- We will have a single SQLite file that contains all the information necessary for the website to function properly
- Two types of tables: master account list, and individual user data
- The *users* table contains all usernames and corresponding (hashed) passwords
- Each user will receive their own table; the *<username>* table stores all the schedules he/she created
- Each record contains data for a single schedule: event date, event address, event time (start and end), event cost, eatery address, eatery cost

API Information

- Eventbrite API: up to 1000 free calls per hour
- Google Places API: up to 1000 free calls per day, each text search counts as 10 calls
- *Google Maps Embed API:
- *Google Maps Directions API:

Task Distribution

- Ely: Project Manager + Back End: dispEvent.py, dispFood.py
- Dhiraj: Front End: newEvent.html, eventList.html, foodList.html
- Ziyan: Front End: login.html, summary.html, home.html
- Reo: Back End: auth.py, dispSummary.py, dispPlans.py, database.db

Project Timeline

By Monday (12/5):

- Reo: Complete design document (written portion)
- Ziyan + Dhiraj: Add necessary maps/diagrams to document
- Ely: Review design document

By Tuesday (12/6):

- Ely:
- Dhiraj: newEvent.html - functional version
- Ziyan:
- Reo: auth.py

By Wednesday (12/7):

- Ely: dispEvent.py
- Dhiraj: listEvent.html - functional version
- Ziyan: login.html - functional version
- Reo: dispSummary.py

By Thursday (12/8):

- Ely: dispFood.py
- Dhiraj: listFood.html - functional version
- Ziyan: summary.html - functional version
- Reo: database.db, dispPlans.py

By Friday (12/9):

- Ely: Double check all functionality/bug fixes
- Dhiraj: Make website uniform/pretty.
- Ziyan: home.html - and work with Dhiraj to make website pretty
- Reo:

Over the weekend (12/11):

- Ely: Add additional features/APIs

- Dhiraj: Incorporate additional features/make website super cool
- Ziyan: Incorporate additional features/make website super cool
- Reo: Add additional features/APIs

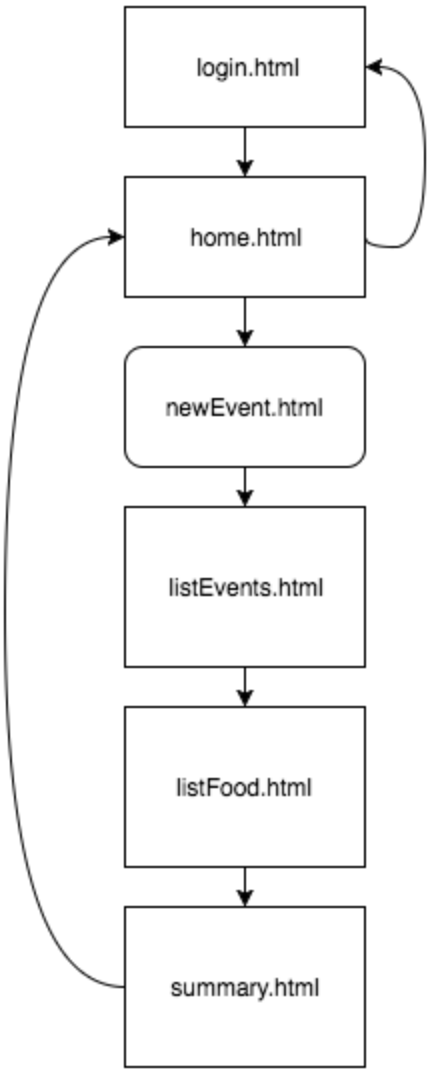
By Monday (12/12):

- Everyone: Finish the project and get a decent amount of sleep

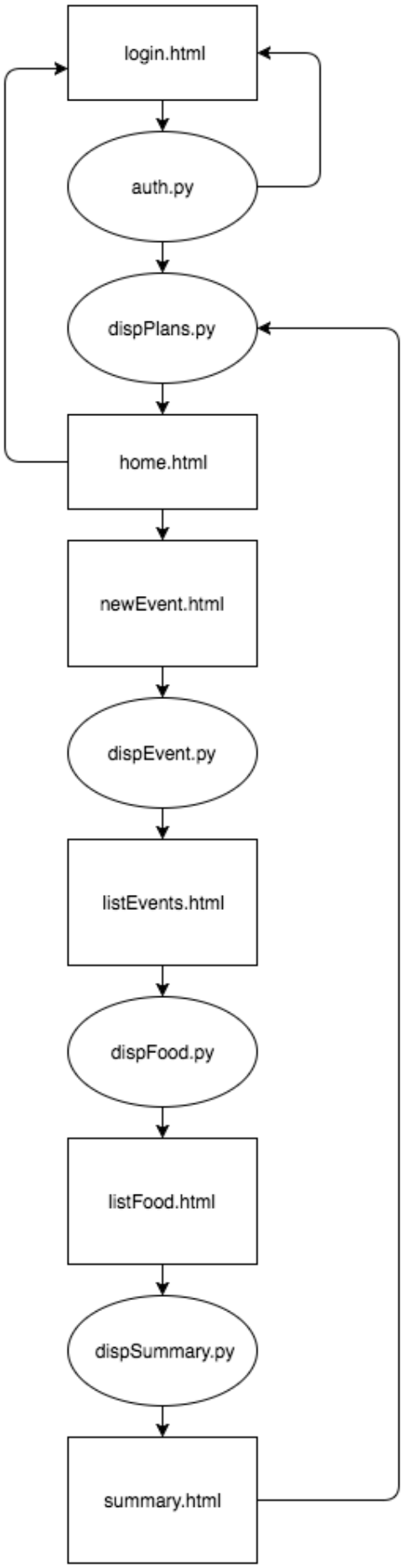
**extra feature that will be implemented if time permits*

<u>Maps</u>

Site Map:



Components Map:


login.html
auth.py
dispPlans.py
home.html
newEvent.html
dispEvent.py
listEvents.html
dispFood.py
listFood.html
dispSummary.py
summary.html