

Milestone 2: Project Outline

Alicia Teo, Christina Chen, Fred Chang, Lucy Kuo

1. Motivation for the idea/description of the problem the application solves

Today, if one were to plan a trip, including what to see, where to stay, and what to eat, it involves cross referencing multiple sources like Google Maps, hotel websites, Expedia, Yelp, Foursquare, etc. Our webapp, called **EatStayLove**, integrates everything by becoming the **one-stop-shop** for helping you find all the best hotels and restaurants based on the sights you wish to see.

The user chooses a city and landmarks to visit, and the app will provide hotel and dining suggestions that are in the area. The app will then summarize the user's itinerary.

2. List of features you will definitely implement in the application

- Landmark selector (user can add to a list the places they wish to visit)
- Hotel recommendation system (users will be shown hotel recommendations)
- Restaurant recommendation system (users will be shown nearby restaurant recommendations with curated restaurant reviews)
- Trip summary (Users will see the landmarks they have chosen, the hotel, and restaurants they have chosen)

3. List of features you might implement in the application, given enough time

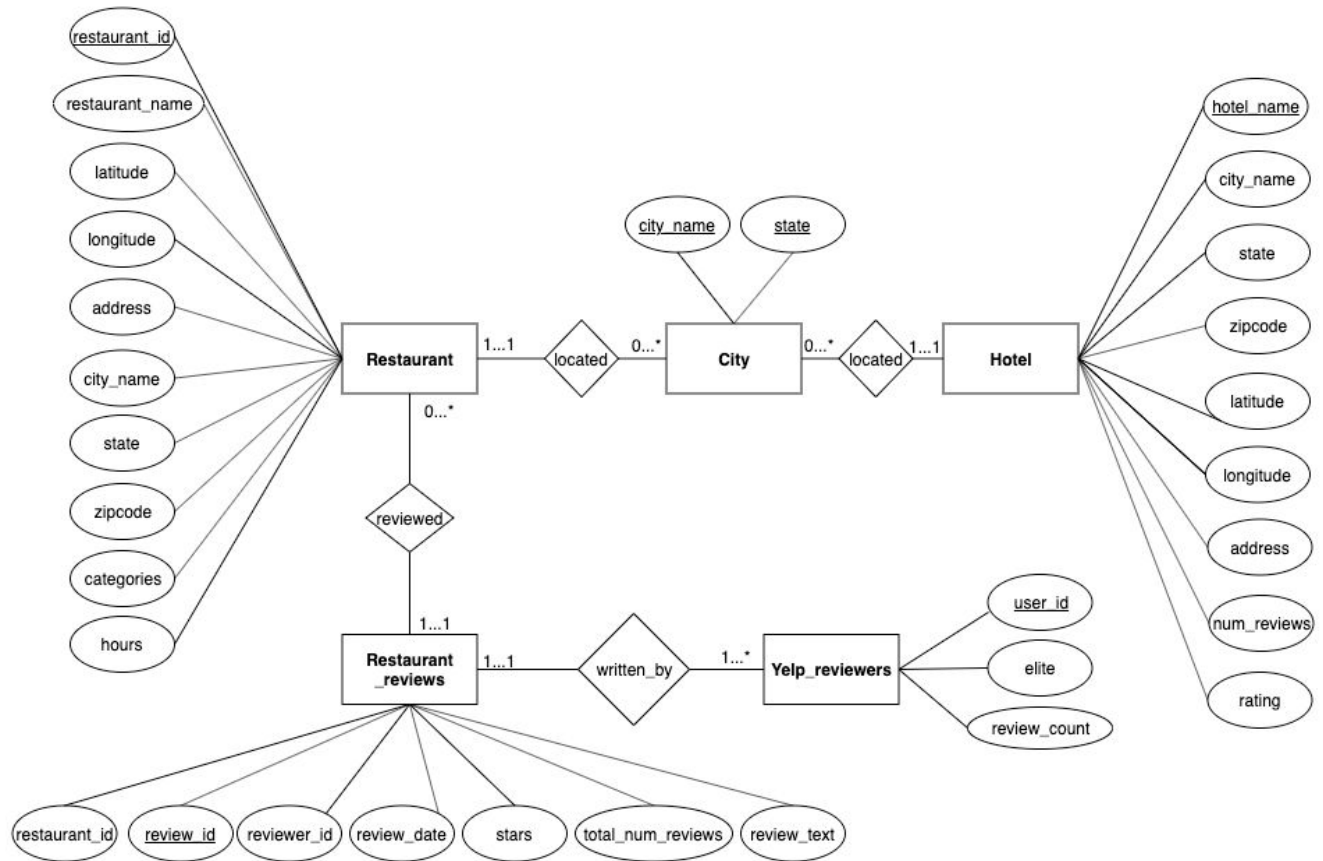
- Cloud hosting on AWS
- Real Time-Streaming Data: We will try to display the real-time tweets including the hashtags of the attractions chosen (e.g. #empirestatebuilding) so that the user is up to date with the hype of that attraction.
- Google Maps integration: We will try to display hotels, attractions, and restaurants on a Google Map.

4. List of pages the application will have and a 1-2 sentence description of each page

- **Page 1: City + Attractions**
User selects a city, the app will retrieve a list of attractions for that city using foursquare API. The user can then select the attractions he/she is interested in.
- **Page 2: Hotels**
User selects the distance from attractions, and price range of hotels. Based on the attractions selected, the app will show a list of recommended hotels.
- **Page 3: Restaurants**
User will see Yelp photos from the restaurants that are near (0.25 mi from) the hotels selected along with their titles and addresses. Users can filter by cuisine, rating, and price range.

User will also see recent, high quality reviews about the restaurant (3 most recent reviews from yelp elites)
- **Page 4: Trip Summary**
Based on the city selected, user will get a full summary of the attractions, hotels, and restaurants that are closely related.

5. Relational schema as an ER diagram



City(city_name, state)

Hotel(hotel_name, city_name, state, zipcode, latitude, longitude, address, num_reviews, rating)
city_name, state foreign key linking to city(city_name, state)

Restaurant(restaurant_id, restaurant_name, latitude, longitude, address, city_name, state, zipcode, categories, hours)
city_name, state foreign key linking to city(city_name, state)

Restaurant_reviews(restaurant_id, review_id, reviewer_id, stars, review_text, review_date, total_num_reviews)
restaurant_id foreign key linking to restaurant(restaurant_id)
reviewer_id foreign key linking to yelp_reviewers(user_id)

Yelp_reviewers(user_id, elite, review_count)

6. SQL DDL for creating the database

```

CREATE TABLE city(
    city_name VARCHAR,
    state VARCHAR,
    PRIMARY KEY(city_name, state)
  )
  
```

)

```
CREATE TABLE hotel(  
    hotel_name VARCHAR,  
    city_name VARCHAR,  
    state VARCHAR,  
    zipcode CHAR(5),  
    latitude DECIMAL(8,6),  
    longitude DECIMAL(8,6),  
    address VARCHAR,  
    num_reviews int,  
    rating FLOAT,  
    PRIMARY KEY (hotel_name),  
    FOREIGN KEY (city_name, state) REFERENCES city(city_name, state)  
)
```

```
CREATE TABLE restaurant(  
    restaurant_id VARCHAR,  
    restaurant_name VARCHAR,  
    latitude DECIMAL(8,6),  
    longitude DECIMAL(8,6),  
    address VARCHAR,  
    city_name VARCHAR,  
    state VARCHAR,  
    zipcode CHAR(5),  
    categories VARCHAR,  
    hours VARCHAR,  
    PRIMARY KEY (restaurant_id),  
    FOREIGN KEY (city_name, state) REFERENCES city(city_name, state)  
)
```

```
CREATE TABLE restaurant_reviews (  
    restaurant_id VARCHAR,  
    review_id VARCHAR PRIMARY KEY,  
    reviewer_id VARCHAR,  
    stars int,  
    review_text VARCHAR(255),  
    review_date VARCHAR(255),  
    total_num_reviews int,  
    FOREIGN KEY (restaurant_id) REFERENCES restaurant(restaurant_id),  
    FOREIGN KEY (reviewer_id) REFERENCES yelp_reviewers(user_id)  
);
```

```
CREATE TABLE yelp_reviewers(  
    user_id VARCHAR PRIMARY KEY,  
    elite VARCHAR(255),  
    review_count int  
);
```

7. Explanation of how you will clean, pre-process, and ingest the data into the database

- **Foursquare Landmarks:** API call, no need to clean
- **Hotels:** Using Data Miner Chrome plug-in to scrape data. Since output is already in CSV format, minimal cleaning is required to filter out bad data (e.g. hotels that are in the list but not yet opened). Will be saved as a SQL database table.
- **Yelp Restaurants & Reviews:** Minimal cleaning should be required since Yelp data is already in desired format. We will filter out unnecessary columns to reduce size of table. Will be saved as SQL database tables.

8. List of technologies you will use

- ReactJS or AngularJS, HTML, CSS for front end
- NodeJS middleware
- SQL database

9. Description of what each group member will be responsible for

Alicia: Data cleaning, FourSquare API, queries for front-end

Christina: Populate Database with Yelp Information, AWS Setup, queries for front-end

Fred: Front-end, Data Scraping - Hotels, Google Maps API

Lucy: Front-end, AWS Setup