Database

1. **Cache**:
   We plan to use Redis as the cache for the application. Each time a client enters a search query, it will first look at Redis store. If the query is found, it would fetch the response and return to the client. Otherwise, the get request would call Yelp and Google Custom Search APIs and then load the search result to a collection called 'search' in MongoDB as well as Redis.
2. **Data Schemas:**
   We used two schema in the application: UserSchema and TripSchema.

```
const user = new Schema({
  userId: { type: String, required: true }, //store ID
  encrypted: { type: String, required: true }, //encrypted pw
  trips: { type: Array, required: false }, });
```

As discussed, the user is storing their Id, password, and the trips taken.

```
const tripSchema = new Schema({
  userId: { type: String, required: true }, //store the user who created this
trip
  name: { type: String, required: true }, //store the name of the trip
  restaurants: { type: Array, required: false }, //store the restaurants
visited during this trip
  isComplete: { type: Boolean, required: false } //whether this trip is
current(this is optional)
});
```

The tripSchema aims to store user's experience on the website.

3. **Sequence for Cache:**