GreenHaven Database Design

Database Choice: Firebase Firestore

Firebase Firestore is the ideal choice for the GreenHaven app. Its real-time data synchronization is crucial for the GreenHaven app where users will be able to see live updates to reviews and ratings without needing to refresh the app, providing a seamless user experience. Eventually GreenHaven will have to scale, and Firestore handles this automatically, making it an excellent choice for when more users start using the app. Some of the more key features of Firestore, when it comes to the GreenHaven app, are the security and authentication as well as the ability to integrate with Google services easily. GreenHaven will use Firebase Authentication which integrates seamlessly with Firestore, ensuring that user data is protected, and access is controlled. GreenHaven will be using the Google Places API which, like other Google services, integrates well with Firestore. This will simplify the process of fetching and displaying up to date information about the green spaces.

Collections and JSON Structures

1. **<u>Users Collection</u>:**
    a. Purpose: Store user information for authentication, personalization, and data that is user specific such as favorites and reviews.
    b. Document Structure:

```
{
   "userId": "string", // Auto-generated ID by Firestore "email": "string",
   "username": "string",
   "createdAt": "timestamp",
   "reviews": ["string"], // Array of review IDs
   "favorites": ["string"], // Array of place IDs (from Google Places API)
   "lastPlacesLookedAt": ["string"] // Array of place IDs (recently searched)
}
```

c. Explanation: This collection stores essential user information, including unique user IDs for authentication and user management. The reviews field is an array of review IDs that the user has written, the favorites field is an array of place IDs that the user has favorited, and the lastPlacesLookedAt field stores the place IDs of the user's recent searches.

2. **Reviews Collection**:
    a. Purpose: Store user generated reviews for green spaces
    b. Document Structure:

```
{
    "reviewId": "string", // Auto-generated ID by Firestore
    "placeId": "string", //Referenced from Google Places API
    "userId": "string", // Foreign key referencing Users collection
    "rating": "number",
    "comment": "string",
    "timestamp": "timestamp"
}
```

    c. Explanation: This collection stores reviews, ratings, and comments provided by users for various green spaces. The placeId references the green space ID from the Google Places API, and the userId references the Users collection, establishing relationships between these collections.

Purpose, Implementation, and Interactions

1. **Users Collection**:
    a. *Purpose*: This collection is crucial for user authentication, personalization, and storing user specific data like ‘favorites’ and ‘reviews’
    b. *Implementation:* When a user registers and is logged in, their information is stored in this collection. Each user document includes their unique ID, email, username, an array of review IDs, an array of favorite place IDs, and an array of recently searched place IDs.

c. *Interaction:* Users register or log in to the app, which validates their credentials against the Users collection. Once authenticated, users can write reviews, favorite places, and perform searches. The reviews, favorites, and lastPlacesLookedAt fields are updated accordingly.

2. **Reviews Collection**:
    a. *Purpose:* This collection holds user-generated reviews for green spaces, essential for providing feedback and ratings.
    b. *Implementation:* When a user submits a review, a document is created in the Reviews collection with the review details, including references to the user and the green space.
    c. *Interaction:* Users write reviews for green spaces, which are stored in this collection. These reviews are linked to both the user who wrote them and the green space they pertain to. The app displays these reviews on the green space detail pages.

Example Data and Usage Scenarios

1. **Nature Enthusiasts**:
    a. *Scenario*: Emily, a nature enthusiast, logs into the GreenHaven app to explore new green spaces for hiking.
    b. *Adding a new user:*
        i. Example Users collection document:

```
{
  "userId": "emily123",
  "email": "emily@mail.com",
  "username": "emily123",
  "createdAt": "2024-07-01T12:00:00Z",
  "reviews": [],
  "favorites": [],
  "lastPlacesLookedAt": []
}
```

    c. *Submitting a review:*

i. Example Reviews collection document:


```
{
   "reviewId": "review789",
   "placeId": "place456",
    "userId": "emily123",
   "rating": 5,
   "comment": "Great hiking trails with beautiful views!",
   "timestamp": "2024-07-02T08:00:00Z"
}
```


ii. Example Users collection document after submitting review:


```
{
     "userId": "emily123",
      "email": "emily@mail.com”,
      "username": "emily123",
      "createdAt": "2024-07-01T12:00:00Z",
      "reviews": ["review789"],
      "favorites": [],
      "lastPlacesLookedAt": []
 }
```


2. **<u>Families</u>**:
    a. *Scenario:* The Johnson family wants to find a safe and enjoyable park for some weekend fun.
    b. *Favoriting a green space:*
        i. Example Users collection document after favoriting a green space:


```
{
    "userId": "johnsonFamily",
    "email": "family@example.com",
    "username": "johnsonFamily",
    "createdAt": "2024-07-01T12:00:00Z",
    "reviews": [],
    "favorites": ["place789"],
     "lastPlacesLookedAt": []
}
```

3. **<u>Tourists</u>**:
    a. *Scenario:* Terry, a tourist, searches for popular green spaces in a new city.
    b. Recently searched places being stored:
        i. Example Users collection document after searching for green spaces:


```
{
    "userId": "terryTourist",
    "email": "terry@mail.com",
    "username": "terryTourist",
    "createdAt": "2024-07-01T12:00:00Z",
    "reviews": [],
    "favorites": [],
    "lastPlacesLookedAt": ["place123", "place456"]
}
```


4. **<u>Working Professionals</u>:**
    a. *Scenario:* Winston, a working professional, wants to find a nearby park to relax during his lunch break.
    b. *Using recent searches and favorites:*
        i. Example Users collection document after searching and favoriting a green space:


```
{
    "userId": "winstonPro",
    "email": "winston@mail.com",
    "username": "winstonPro",
    "createdAt": "2024-07-01T12:00:00Z",
    "reviews": [],
    "favorites": ["place789"],
    "lastPlacesLookedAt": ["place123", "place789"]
}
```


The GreenHaven app will use Firebase Firestore to effectively handle user data, reviews, and personalized features like favorites and recent searches. By organizing the data into clear collections and taking advantage of Firestore's real-time and offline capabilities, the

app will deliver a strong and responsive user experience. This setup ensures that each type of user, from nature enthusiasts to tourists, enjoys a tailored and engaging experience.