**Flash party app**

*Mike 10/20/17*

***Backend:***

**Basic Idea =** Hangout Matching + Automatic Hangout Suggestion:

* Hangout sessions can be **proposed** by individual users with specifications mentioned below;

*(When 2 or more Hangout sessions with similar activities/interests, time and location exist in our database, the server attempts to combine the Hangouts. Larger group is created only if there’s mutual agreement among the Hangout initial proposers.)*

* OR alternatively, our system automatically matches people and make smart Hangout recommendations:

*(user can update their interests (topics, activities, etc.) at any time, while the app automatically sends device location to Flash Party Inc. server with user id. The server then* ***constantly matches people for Hangout sessions****. The Flash Party hangout suggestions are dynamically displayed on the app feed.)*

Data structure to upload:

* **User ID**: only a number ID for database verification. This ID can be indexed and searched in the User Data collection at anytime. This sounds like cross-table in relational database, but I do prefer keeping the data organized and easily sorted, searched.
* **Timestamp :** this is encoded automatically by MongoDB, but we could also make this one of the data points to be submitted.
* **Geolocation :** this is the most important piece of information, since our app’s main selling point of location based Hangout recommendations. The GPS coordinates can be pulled from the device location API and should require the user to allow privacy permissions before doing so.

Location Updates:

* Idea 1: GPS location is only sent when the user INITIATES a Flash Party or is actively looking for a Flash Party nearby that suits his/her interest.
* Idea 2: Continuously send GPS coordinates data in 5-minute interval, and the app notifies the user when a potential hangout match is found. (Consider location services while app is running in the background, or disable it to conserve power usage).

Important Information about a Hangout Session:

* Duration of hangout;
* Location of Meetup;
* Topic of interest/Proposed Activities;
* Basic profiles of Participants/Interested Flashers;
* Rules: No smoking / Bring Pálinka, etc. 🡪 how to ensure hangout safety? Identity verification?

***Database:***

In the database, we’re going to correspond each user ID with a user object, with necessary information. This could include name, age, gender, nationality, interests etc. Make sure usernames are distinct and assign user IDs for each user so that it’s easier to index and search in the database.

There're three categories of data we'd like to store. So we will create 3 separate Collections in MongoDB (maybe other NoSQL databases, but I have implemented MongoDB before, so I’ll stick to it at the moment.), with some constraints on the schema for convenient in data processing:

1. *Flash User Account Data*

Write to this collection whenever a new app user is creating an account (check duplicate username first). Collect relevant information from registration forms/retrieved from Facebook, Twitter APIs. Format it in appropriate JSON object and store in the document.

A crude example of the JSON string:

{ \_id: ObjectID(5969ca6ce77620267a1247f2),

user\_id: 130423,

name: “MikeFlasher”,

gender: “male”,

age: 21,

interests: [“music”, “politics”, “desserts”, “skydiving”, “travel”]

}

1. *Real-time location data:*

The app collects this data in real-time/set intervals, or if the user prefers, only collects locational data when the user wants to. This data is stored in the Location data collection, where the sequence is sorted based on timestamp. Further sorting is possible on similar locations, activities, interests, etc.

For the Hangout recommendations, the back-end app sorts and grabs location data in real time and make clusters of user JSON objects. It then cross compares the topics of interests, activities, etc. to match the objects. The app then look at the user\_id of the matched location senders, and send them notifications regarding the Hangout recommendation.

{ \_id: ObjectID(5469ca2ce726402737a8127g2) 🡪Includes timestamp information, can be extracted later on.

user\_id: 130423,

activities: [“drinking”, “arcade”, “study Hungarian”, “Lángost eszunk”],

location: [34.2345, 120.3412]

}

1. *Hangout session Collection:*

This collection collects all the proposed and suggested Hangout sessions created. We can perform searches based on many different parameters such as the topic, activities, location, time, duration or even just the name of the event.

It’s also easy to delete a Hangout session if the event has been proposed for over a specified amount of time, or the proposer decides to cancel the event.

{ \_id: ObjectID(5469ca2ce726402737a8127g2) 🡪Includes timestamp information, can be extracted later on.

Hangout\_name: “Tech Talk Rocks”,

proposer\_id: 130423,

activities: [“talk about tech news”, “Lángost eszunk”, “a filmet nézunk”],

location: [34.2345, 120.3412]

}

***Frontend:***

Front Page (Scroll View): The most important frontend element of this app is the Hangout Feed, where the user can scroll through the proposed hangouts within specified distance and meeting time/duration (e.g. , within 2 hours, for around 1 hour). Each tableViewCell in the feed would display the **name of the Hangout** (named by the initial proposer)**,** **time/duration, location, proposed activities/topic of conversation** and **number of active participants.**

User Account Creation:

* Some basic user profile/info can be requested in forms. Or alternatively, we can ask for permission to retrieve relevant information via REST API calls from social media sites such as Facebook, Twitter, LinkedIn, etc.;
* With Social media integration, users can follow their friends/invite friends onto the Flash Party platform. They can view their friends’ activities or invite friends to a particular Hangout session;
* Maybe some customization features: user can update personal interests, what they would like do or to talk about/what kind of people they want to hang out with at the moment;
* Button for **Suggest topics** 🡪 Create a Flash Party initiative.