NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

School of Computer Science and Engineering
College of Engineering

Geographically-aware mobile application

Managing events and interest group communications

Student: Lee Chong Yan Supervisor: Mr Oh Hong Lye

BACKGROUND

Crowdsourcing has become increasingly prevalent as people source for more information and services from one and another over social media platform.

The current platforms like Reddit and social media platforms, like Twitter have provided a vibrant avenue for people to share information. However, for time-sensitive information, the current pull forum-based approach may not work well as questioners may not feel the real-time response to their questions. For geographical context-specific question, the information shared may be irrelevant if the topic is too geographically specific and is thus only wise to be targeted at audience in close vicinity.

OBJECTIVE

The objective is to develop a cross platform mobile application where users can setup channels to disseminate information on events or any special interest topics. The geographical and time aware nature of the application allows users to access and subscribe to more relevant information based on where and when they are.

For other non-geographic specific channels, a recommendation engine would be implemented to push the relevant channels to the users.

SYSTEM DESIGN

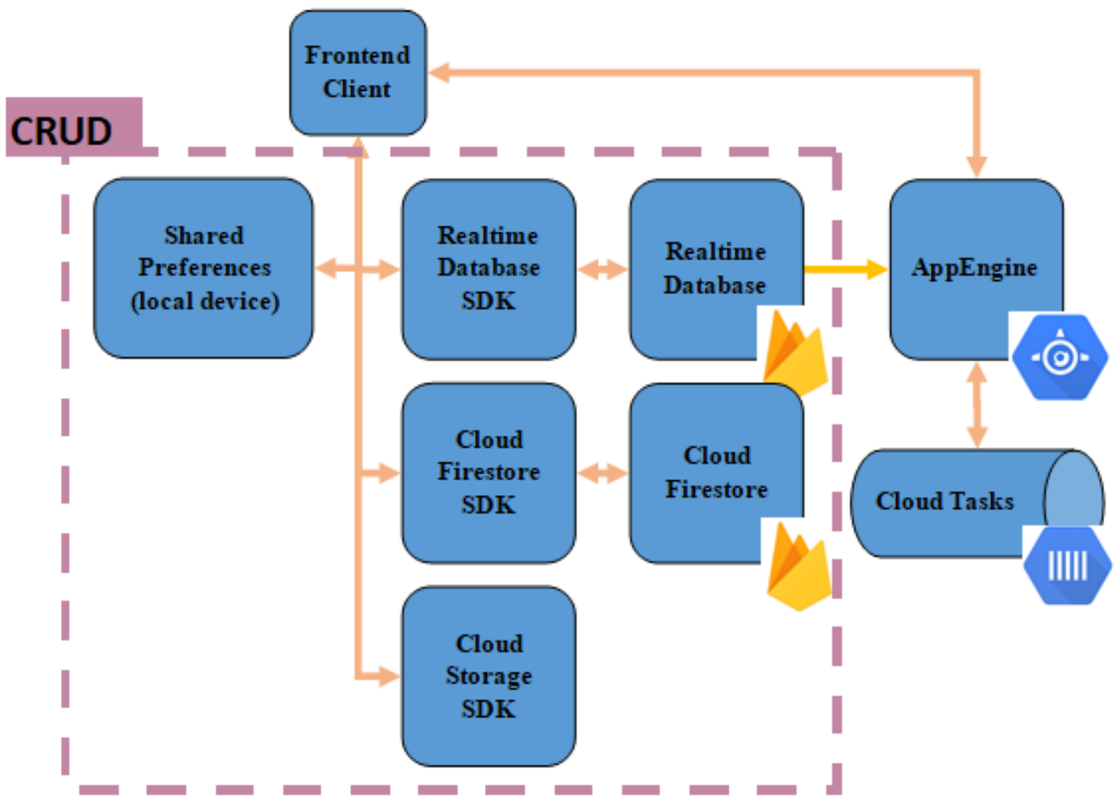

Figure 1: Backend Architecture of Application

For backend, we use microservice architecture with the Flask API on Google AppEngine so that new services can be added easily. Events will be pushed to the Cloud Tasks to be performed by the services asynchronously, which can be scaled independently. Database operations will be done with Google SDK to reduce cost.

For frontend, we use Flutter which is cross-native due to its performance. State persistence will be done both locally and remotely depending on the security requirements.

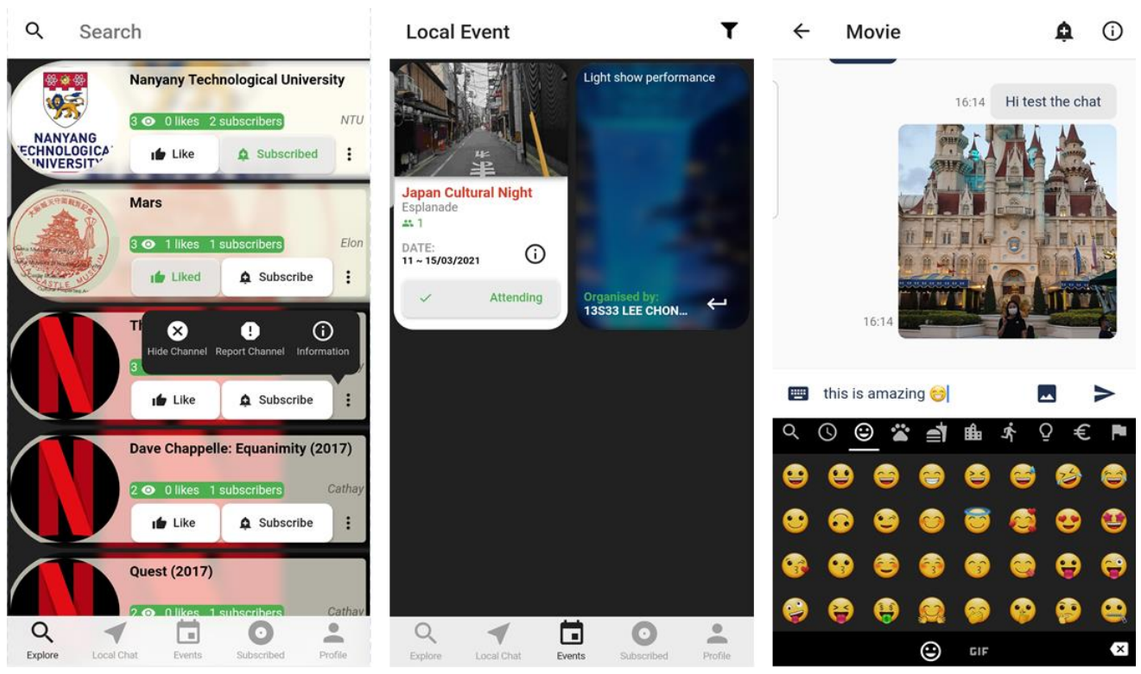

Figure 2: Frontend of Application

GEOGRAPHICAL QUERY

We use Geohash, combining longitude and latitude to query for nearby information based on user's current location.

ANONYMOUS CHAT HANDLE

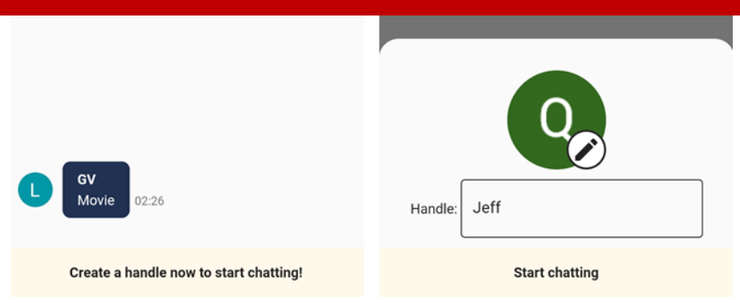

Figure 3: Handle Creation

To ensure that users stay safe over the Internet and to prevent abuse of identity, users can create their own handles before chatting in any channel, or they can stick with their default username.

RECOMMENDATION SYSTEM

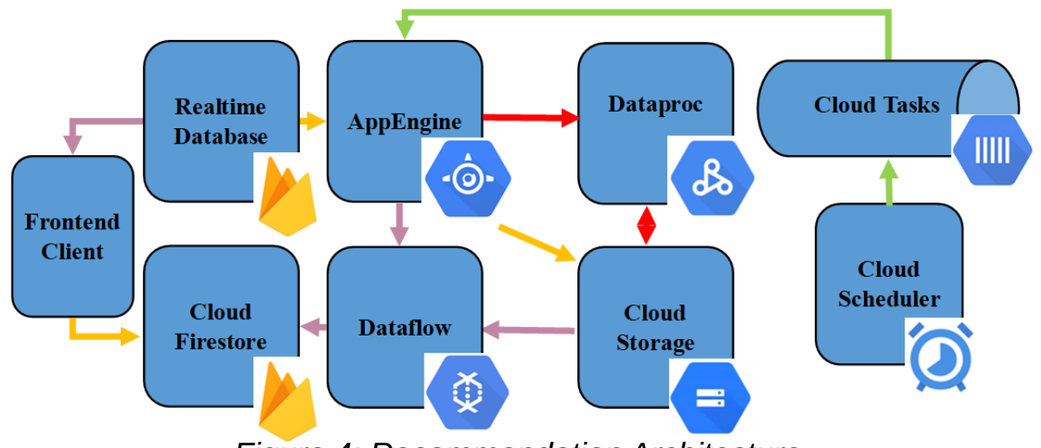

Figure 4: Recommendation Architecture

Offline architecture is used, where new recommendations will be created at the end of everyday. User activity is recorded based on their hide, view, like and subscription of increasing rating score.

1. Cron sends requests to Cloud Tasks, triggering service
2. AppEngine exports user log to Cloud Storage
3. Dataproc cluster will be set up and ALS (Alternating Least Square) algorithm will be run, outputting the results back to storage
4. Dataflow will be used to parallelize the transfer of results from storage to Firestore for fast client retrieval