Jia-Han Chiam – jiahan@stanford.edu
Qianying Lin – qlin1@stanford.edu
Chuanqi Shen – shencq@stanford.edu
Kaidi Yan – kaidi@stanford.edu
GitHub Repo – https://github.com/sixkkdrx3/cs194.git

CS194 Project Proposal

Executive Summary

This proposal showcases our project, Maaap, a social media smartphone application for Android platform. We discuss our application's proposed user interface, and explain how it differs from existing competitors in the social networking space. We also delve into the technical details of our application's implementation, and discuss possible challenges and risks we might face while developing our application.

Description of the Project

We propose to make a social media smartphone application that allows users to post short, anonymous messages tagged with geographic location information. Our application can be described as a Yik Yak-style social media application with a map-based interface. We call our application Maaap, which is a combination of "Map" and the sound a sheep makes.

Our application's main interface will be a map filled with speech bubbles that we call "bleats". This will be somewhat similar to the interface of apps like AirBnb and Uber, except the icons will correspond to messages, instead of houses or cars.

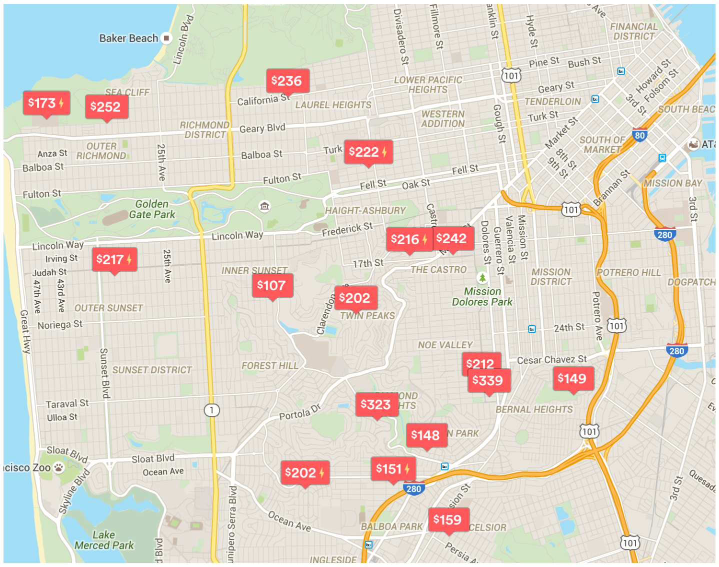


Figure 1. User interface of AirBnb

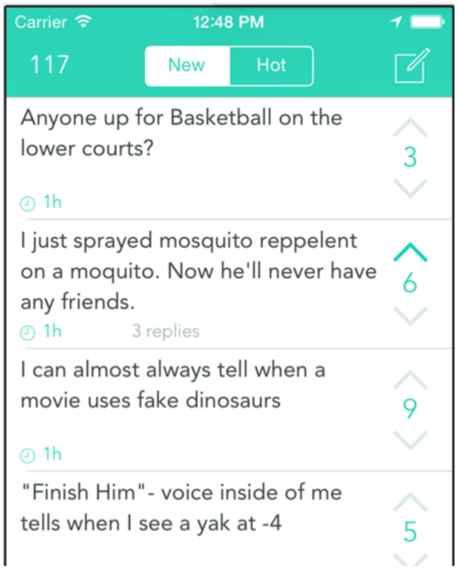
Each bleat, when tapped, will expand and display an anonymous message posted at that location, together with the time of posting. To scroll through the bleats, the user can utilize standard interface controls such as panning, zooming, dragging and pinching, to explore bleats at various locations. To provide the best experience, when zooming in or out, a specially chosen set of bleats, based on popularity and time of posting, will be shown prominently, while other bleats will accumulated into bleat clusters that can be expanded to view individual bleats. The most prominent bleats will be pre-expanded without explicitly being selected.

To post a message (i.e. when a user "bleats"), the user simply has to type a message and press the "Bleat!" button. A bleat will immediately appear on the map at the current geographic location of the user. Each bleat will expire after a set period of time. Users can also vote for bleats they find interesting, and bleats with more votes will appear more prominently on the map and last longer.

Need for the Product

In Maaap, bleats are accompanied by a specific piece of contextual information—the location. This geographical perspective allows users to easily narrow down to bleats in locations they are interested in. For example, a user can pan over and zoom into the Stanford Stadium to view the latest discussions about the ongoing game. By using the map as the main interface, the location of the user, instead of merely being an incidental piece of metadata, becomes a central property of a bleat. Tying each bleat to a physical location reifies the notion that there is a real person behind each post. Compared with a simple, abstract list of messages that most social applications provide, Maaap allows users to feel more viscerally connected to other posters.

Location can also carry semantically important information. For example, someone selling tickets to a show, or looking for basketball players as in the following Yik Yak screenshot, will not have to specify their locations because they will be automatically reflected in their post. Also, the bleat "Looking up at the supermoon tonight" becomes a lot more interesting if the reader knows that the post was posted from the middle of Lake Lagunita.



Anyone up for Basketball on the lower courts?

I just sprayed mosquito reppelent on a moquito. Now he'll never have any friends.

I can almost always tell when a movie uses fake dinosaurs

"Finish Him"- voice inside of me tells when I see a yak at -4

Figure 2. Sample screenshot of the Yik Yak user interface

Audience

We are not limiting ourselves to any specific group of people, because we believe that the appeal of our application is universal. Any message can be enriched by geographical information. Be it a birdwatcher exclaiming the sighting of a rare species near Bayshore, or a student proudly exclaiming the victory of the Stanford footballers at the Stanford stadium, providing the location often adds a compelling narrative for the underlying message.

Competitive Products

There are a number of competitive products that serve similar functionalities as Maaap does. One of them is Yik Yak, a social media smartphone application that allows people to start and read discussions within small geographic pockets (e.g. campus zones). Yik Yak is intended for sharing primarily with those in proximity to the user, potentially making it more intimate and relevant for people reading the posts.

The major advantage Maaap has over Yik Yak is the primary interface through which the messages are displayed to the user. Instead of simply showing a list of messages, our application display a map which users can scroll and zoom to read nearby messages. This additional feature makes Maaap more convenient for many use cases. For instance, if someone wants to hold a group study session at Green Library, they can easily bleat, and this bleat will appear on other users' screens as a speech bubble over Green Library.

There are other popular social networks, such as Facebook, which also has status updates and geotagging functionality. However, Facebook is not anonymous and has a much broader mission (allowing users to share photos and articles, for example), meaning that it occupies a different niche from our product. Facebook users primarily use the website to connect with people they know in real life, while Maaap provides a anonymous platform where people can interact with strangers, offering an opportunity for people to make new friends and increase bonding among a community.

Technologies Needed

The frontend of our product will be a mobile application. We choose to develop on Android as the Android platform has the largest global platform share and is open-source, allowing developers more extensive access to the operating system internals.

Our mobile application requires extensive use of a map interface. We will use Google Maps, as it has extensive API that allow external users to easily use Google Maps features.

To know where bleats are posted, we need to know the geographic location of the bleat. All modern smartphones are equipped with a GPS system, and the standard Android APIs allow us to easily acquire the phone's location coordinates. To ensure security, all communication channels between the mobile phone and our backend will use SSL for encryption.

We will also require a database to store all the messages and metadata. Our database setup does not need to be excessively complicated, and a key-value store will serve us adequately. We will host our backend on Amazon EC2, and will use DynamoDB, a highly scalable and highly available key-value store, for our database. To ensure security and anonymity of content, we will also encrypt information on our backend with SHA-2.

Resource Requirements

We will mainly develop our application using freely available and open-source resources. Therefore we do not require access to special development tools. For example, the Android Development Kit we will develop on is freely available. We may however, require some financial assistance, as we will use several Amazon AWS features heavily. Although we do not need to pay for EC2 instances if we create free-tier instances, we require DynamoDB for our database records.

Potential Approaches

As the main purpose of using Maaap is to allow users to find out where interesting things are happening, one might come up with other approaches that can achieve similar goals. One potential alternative we have considered is to build an application that is similar to SUPost - users can post new threads about activities they are doing, and describe their locations in the post. Users can also filter the activities based on different types, just as SUPost does for different types of products users sell.

While this idea sounds easy to implement, there are two major disadvantages for this alternative - first, it is less illustrative to indicate locations using words as compared with marking the location on a map. Secondly, it is harder to customize the region which the user is interested in if we choose the alternative - the best we can do is to give options such as “within 1 mile”, “within 5 mile” and so on. For Maaap, on the other hand, we allow users to zoom in or zoom out the map in an arbitrary manner, thus giving more flexibility in their region of interest.

Risk Assessment

This project can be challenging in the following aspects - first of all, we want to limit the amount of inappropriate content shown on Maaap. One major criticism of Yik Yak is that people often abuse Yik Yak by sending inappropriate messages. We intend to use our voting system to reduce the likelihood of such events happening on Maaap. Users can up-vote bleats they like and they can also down-vote bleats they deem inappropriate. Bleats with significant number of downvotes will be automatically quarantined. In this way, we allow the users to cultivate the kind of environment they are comfortable with.

In addition, we want to ensure reliability at all times. Our application is heavy on images and animations, and thus computationally expensive. As thousands of bleats can occur at a given place, ensuring a smooth and responsive user interface must be a priority. We will refine our system to intelligently choose a subset of bleats to prominently feature on the application, and hide the other bleats. This will greatly reduce the computational cost of rendering the interface and reduce visual clutter. As we plan on achieving high availability, we also plan on conducting heavy stress tests before releasing the product.

Privacy is also one of our major concerns. Our application requires access to sensitive information (specifically, geographical location), which may turn off potential users. For example, Yik Yak faced backlash when it was revealed that they were collecting and storing the geographical information of users, even when users opted to turn the feature off. We feel that being frank and explicit about our application’s usage of geolocation, and its central importance to the application’s functionality, will reassure users of our benign intentions. For example, Uber’s use of the GPS feature is rarely called into question, as users acknowledge that it is fundamental to the operation of the app.

Next Steps

Since we would like to have a map as our main interface, we need to explore various ways of displaying maps and rendering important visual information on top of them. To accomplish this, we will need to research the relevant APIs. More importantly, we will need to design, prototype and evaluate various candidate user interfaces for our application, to ensure it does not become too visually cluttered, as this would intimidate potential users.

We also need to design a scalable backend that will allow us to store and retrieve user data.

We plan on using Android Studio as our main development platform. Below is a step-by-step list detailing the steps needed for the creation of the application.

1. Pop-up dialog to show the instructions
   a. The pop-up dialog is scrollable. Instructions are split across different pages.
   b. There is a ‘cross’ on top right corner of the dialog. Upon tapping on the ‘cross’, the dialog disappears.
   c. Tapping on anywhere outside the dialog makes the dialog disappear as well.
2. Map main page

. Explore options. The current option is to import Google Map.

a. When there is a news feed by certain person in a certain location, there will be a speech bubble.

b. Clicking on the flag brings the user to a new page with details of that news feed.

c. The user can zoom in and out on the map as well as scroll.

3. Input news page

a. There is a tab on the application where the user can input their own news.

b. After clicking on the tab, the user can enter a simple message detailing the event.

4. Design a database

a. Store the userID, geographical location and the message.