Bring Back the Yak

Section 0 – Identification

Project Leader: Sarah Zhou

Group Members:
Angela Mao (am23@princeton.edu)
Jose Rodriguez (jmrq@princeton.edu)
Andre Xiong (axiong@princeton.edu)
Sarah Zhou (syzhou@princeton.edu)
Victor Zhou (vzhou@princeton.edu)

Section 1 – Overview

For our project, we are developing a location-based iOS app – to be named – that allows people to post funny and relatable posts, ask questions, share pictures, etc. with the people around them through an anonymous platform. Each post begins will a baseline radius, so only people within that geographical radius will be able to see, vote, and comment on that post. As more people interact with it through up-votes and comments, the radius will grow, allowing an increasing number of people in the area to be able to see the post.

The motivation behind this unique type of scalability is the idea that generally people from the same area find the same jokes funny or the same posts relatable. We want to push these popular posts as far as they remain relatable and funny and allow our users to share these thoughts without the pressure of their name attached to the post.

This app is inspired by the existing app, Yik Yak, and our team has analyzed their successes, but also their pitfalls. Yik Yak was also a location-based social app, where each post had a strict limit of a 1.5-mile radius. In early 2016, Yik Yak introduced “handles,” so users needed to register a username, eliminating the pure anonymity aspect. Since then, Yik Yak has further exacerbated this issue, and its user base has greatly suffered. Our app hopes to fill the niche that Yik Yak has left by keeping it simple and bringing it back to what made it so popular in the first place – anonymously sharing relatable posts, jokes, and pictures.

Section 2 – Requirements and Target Audiences

The problem that we are trying to solve is the current lack of a social, but anonymous, platform. In general, people are less likely to post when they know their name will be attached to it – if the joke isn’t that funny or the post doesn’t receive many likes, then the person may regret sharing in the first place. Our app hopes to mitigate this issue, encouraging people to share their thoughts without any social pressure.

Our intended users are college students. We hope to provide a space for students to simply share their thoughts. As seen from previous evidence, it is more likely that witty punchlines will be posted about something on campus, but our app does not exclude posts that aim to discuss more serious topics anonymously. College students in the same age groups generally share similar experiences, and knowing that other students are going through the same thing as well is comforting.

There are currently two other competitors in this market: Yik Yak and a Facebook group per campus dedicated to memes – for example, Princeton's is "Princeton Memes for Preppy AF Teens."

1. As briefly described in Section 1, Yik Yak has introduced so many new features – profile pictures, biographies, links to other social media accounts, direct message – that undermine the anonymity factor. Our app, on the other hand, will remain simple, clean, and minimal. In addition, each post on Yik Yak had a limited 1.5-mile radius. You could view other locations' posts by using their "Explore" feature, but those posts often weren't as funny as those of your current location simply because you couldn't relate to them. Our app allows users to focus on their current location, and due to our variable radius based on popularity, posts that are overwhelmingly popular will naturally find their way to them.

2. Memes pages have sprouted up on many college campuses as a reaction to the decline of Yik Yak, but because it's through Facebook, there is no way to post anonymously. Knowing that every post is linked to your profile, people hesitate to share their thoughts out of fear of not receiving likes or of people outwardly deeming the meme not funny enough. Our app allows people to post without this hesitation.

Section 3 – Functionality

User – Browsing feed

User can refresh feed by pulling down on the table view. Each post consists of the text and/or picture, the net number of votes, up-vote / down-vote buttons, and when it was posted. User can up-vote or down-vote individual posts. Posts will disappear once it reaches –x votes. User can click on individual posts to view a detail view (see second use case). User can switch between sorting feed by "hot" or "recent."

User – Clicking on an individual post from the feed

Post displays at top of screen. User can scroll down to see comments sorted by time of comment. Each comment displays: the text, net number of votes, up-vote / down-vote buttons, when it was posted, user's icon (randomized every time). User can comment. User can up-vote or down-vote other users' comments.

User – Making a post (text, photo, or both)

User has a limited number of characters for the text part of the post. User can add photo from camera roll or take a photo from their camera. User can post when either text or picture is added – user cannot post blank posts.

User – Viewing "profile"

User can view a feed of all previously posted posts in chronological order. Posts retain all comments, votes, etc., but user cannot comment or vote on any of these posts. This feed is not visible to any other users.

User – Receiving phone notifications

User can turn on / off phone notifications. User will receive a notification each time that their post was replied to.

Section 4 – Design

- iOS
    - Native iOS app written in Swift
    - Built using Xcode
    - Communicates with server solely via the server's exposed API
    - Tracks user location using built-in iOS libraries

- Server
    - Node.js server (written in Javascript), hosted (free tier) on Heroku
        - Handles requests
    - MongoDB database, hosted (free tier) on mLab
        - Stores all app data, including posts, comments, user info
    - Exposes an HTTP API for clients

- Tools
    - Github

Section 5 – Timeline

Week of 3/26

- Repository setup
- Server, Database, and Hosting setup
- API for "registering" new user account done
- API for getting Hot feed done
- API for getting New feed done
- API for creating new posts done

- Design mock-ups for first round of functionality (see Project Prototype)
- Determine general design and theme of the app (colors, graphics)

Week of 4/2

- Begin first round of functionality

Week of 4/9

Friday, 04/14 - Project Prototype

- Barebones of app
- Basic functionality (one feed sorted by "recent", refreshing feed, write post, up-vote, down-vote, splash screen)

- Design mock-ups for second round of functionality (see Alpha Test)

Week of 4/16

- Learn HTML / CSS
- Begin second round of functionality

Week of 4/23

Friday, 04/28 - Alpha Test
- Additional functionality (comments and comment votes, account page, notifications for posts, add feed sorted by “hot”, switch between feeds)
- Remove posts at -x downvotes

- Barebones of webpage

Week of 4/30

Friday, 05/05 - Beta Test
- App and webpage polished, done

- Submit request to publish to the App Store as soon as possible

Week of 5/7

Monday, 05/08 - First demo day
Tuesday, 05/10 - Second demo day
Sunday, 05/14 - Project DUE (by 11:59 pm)

Section 6 – Risks and Outcomes

We have two primary risks:

1. *iOS Development* – Swift 3 is vastly different from Swift 2, and it is not backwards compatible. Our team will have to learn Swift 3, and those who have not worked with Xcode before will have to learn how to maneuver that as well. In addition, we hope to include an interactive portion in our presentation, which will require people in the class to install the app from the App Store before the week of presentations. Thus, we will need to acquire an app developer’s license to publish to the App Store and get the app approved before we present. The approval process takes a variable amount of time, so we are aiming to submit a request as soon as possible the week before presentations.

2. *Webpage Development* – Most of our team is not familiar with HTML / CSS, so we will have to work more diligently near the end of the project to learn the language and produce a webpage on par with the clean graphics we hope to have in our iOS app itself.