



Facebook + YikYak
=
Facemask



Vision

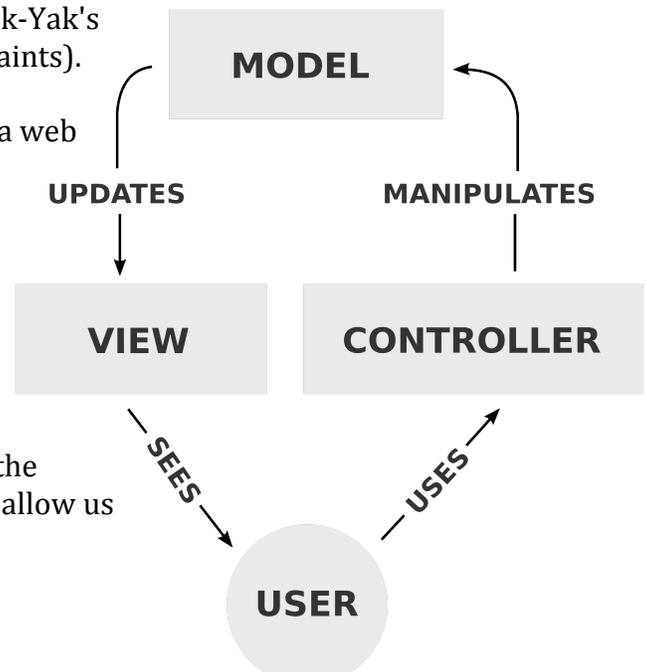
Have you ever caught yourself seeing a hilarious video but being too embarrassed to post it to social media? Maybe you have strong opinions about gun laws but fear your friends roasting you over your opinion. Or maybe you're having a blast watching "Dora the Explorer" but again, can't tell anyone.

Facemask is an anonymous Facebook timeline. More specifically, it is an application that will connect with your Facebook account and allow you to make posts that can be seen in the timelines of all your Facebook friends. Similarly, your timeline will be composed entirely of posts made by yourself and your Facebook friends. Thus, Facemask is very similar to Yik-Yak, except the timeline gets generated by posts made by Facebook friends rather than posts made by people who are geographically near. This project has multiple compelling use cases. It could be used as an anonymous confession page, except all the confessions that you read will only be read by your friends, who you assumingly care about more than the set of all people who are geographically near you. When reading/responding to confessions and other sensitive information, perhaps people will be more compassionate knowing that it is a friend of theirs who is struggling. Facemask would be a place where you could post anything that you would want your friends to see, without necessarily having it tied to your identity. You could post the crazy thing you saw on the bus today. You could post poetry. You could post problems that you're having in your marriage. Facemask could facilitate discussion with people you are likely to care about, while severing the source of discussion from your identity.

Software Architecture

In terms of the user interface, this app can certainly be built. It will be nearly identical to Yik-Yak's design (except much simpler due to time constraints). The timeline just needs to contain vertically sequential blobs of text. To start we may create a web application, and then move to iOS if we have time, but this would depend on what the team decides if funded. We will use the model-view-controller architecture to organize our application.

The key component used to make Facemask possible is the Facebook API. It will only be used to access the friends of the user. A hypothetical technology we could use to create the Facemask backend is Apache Thrift. This would allow us



to define services in any supported language (e.g. Java, C++, etc.), and act as an interface to be accessed by the front end. This would allow us to isolate our backend so that, if we wanted to make a web service as well as an iOS app, the difference would only be the UI code.

One interesting aspect of this project, from a technical point of view, is how to minimize the ways that users can find out the identity of the individual who made a post. For example, if posts were associated with each user, and a user's timeline was generated by iterating through their Facebook friends and loading posts made by each friend, then there would be an easy way to validate which user made a post. That is, if you saw a post, and you thought that the post was created by friend A, then you could unfriend friend A on Facebook, refresh your Facemask timeline, and then if the post was no longer in your timeline (i.e. it was not fetched since you are no longer friends), then it would validate that the post was made by that friend. A solution for this would be to instead have a timeline associated with each user (e.g. a list of post IDs). Then, when a user makes a post, it appends a post ID to the Facemask timeline's of each of their friends. If a user deleted a friend on Facebook, their Facemask timeline would remain unchanged.

Challenges and risks:

The biggest challenge that I see in developing the product on schedule is learning the technologies we will use to implement it. I have pretty limited front-end web development experience, and no mobile development experience. Ideally we would be making an iOS app. Thus, it's hard to project how long it will take to implement these components.