Wager

Group Members

- Michael Swart (mswart@princeton.edu)
- Richard Bush (richard.bush@princeton.edu)
- William Chance (wchance@princeton.edu)
- Tyler Kaye (tkaye@princeton.edu)

Overview

Wager will be an iOS application that provides a social media channel for betting. Briefly, the app will allow a user to post a bet that they would like to make and then anyone that they are connected with on Facebook can see the bet and choose to accept it. There will be a feed for users that shows all posed bets and another feed that shows the progress of made bets, such as the acceptance of the bet and the result of the bet.

Requirements and Target Audience

The target audience is anyone that is willing to make a bet. We expect most of our users to be college aged. Many of the users will be sports fans, but we also would like to make the app usable for mundane bets, such as the over/under on the average midterm score for a class.

There are two main reasons why we believe that this app will be useful for our target audience. The first is that, right now, it can be difficult to find someone who is willing to take the opposite side of a bet. The number of people that I can ask to take the bet are limited to a small social circle of people that I believe are interested in the bet and that I directly ask to take the bet. By allowing any of my Facebook friends who have the app to see a posted bet, the number of people that can see the bet increases greatly. Someone who I would not have believed to be interested in the bet can now accept the bet. Second, it will provide a history of bets, which could be an interesting way to learn about friends regardless of whether a user is interested in betting. For example, if I see my friend posting bets on the New York Jets in some cases and posting bets for the New York Giants in other bets, then I will know that they are not a diehard fan of either team. If I see that many of my friends have been betting on hockey games, I would know that I am not alone in my interest for the sport.

Since sports betting is illegal, no user will be forced to pay up; instead, the losing party will have the option of paying or not. However, since the nature of the app is social, there will be a user rating for each user that shows what percentage of bets this person has paid. A person with a low rating is less likely to be bet with in the future, so it is in every user's best interest to settle bets. The bets will be with Facebook friends, so the real life connection makes it more likely for the loser to be honest and pay up because of the social ramifications of not doing so.

Though there are some apps that provide the some of this functionality, there is no application that has a clean, usable UI and compelling UX. We hope to make the UX both easy and entertaining by providing a feed of bets both in progress and completed, connectivity to Facebook and Venmo, Vegas odds for sports betting from scraped from the web, and individual

user profiles compete with a player rating and past bets. One application that already exists that has similar functionality is called “bait.”
Venmo allows for peer-to-peer payments that have a message associated, so it is possible to use this directly for betting. However, Wager intend to play the role of match maker as well as history. On venmo, the opponent in a bet must already be decided, while here we are setting up opponents. It is a more specific channel in that it is strictly for bets and so payments like “movie ticket” and “gas money” will not be a part of Wager.

Functionality

Looking at the bet feed:
A User opens the app and is brought to the homepage. This homepage consists of 2 lists, sorted by time. The first list contains all posed bets that have not yet been accepted. Each bet has the same information: who posted the bet, the time at which it was posted, the amount of money that the bet is for, and a short message about the context of the bet (i.e. “The Warriors will beat the Cavaliers in 5 games or less”). The second list contains all accepted bets and contains the same information as the first one (bet poster, context, time, amount of money) plus now the person who accepted the bet, a response message is posted below the original poster’s bet context (i.e. “The Warriors will win, but it will take them more than 5 games”), and the status of the bet (“completed”, “in progress”) are listed.

Each bet in the list can be liked or commented on. A like button will be clickable and a list of people who liked the bet will be shown. Bets can be clicked on to show all comments on the bet (i.e. “I agree with Tyler that the Warriors will win. Bad choice of bet Will!”), sorted by the time at which the comments were posted.

User Profile
Whenever a name is listed, it will be clickable. When clicked, the button will lead to a user profile page. Users will also be searchable by Name or username through a search box in the homepage. The top will have general information like the users name (“Michael Swart”), username (“@mike”), and player rating (“5/6 paid bets”), and a description (“I will take any bet against the Warriors”). Below this header, there will be 3 lists, organized the same as the list of bets on the home page: Active bets, past bets, and posted bets. A list of liked bets could be easily added as well.

Searching for a user
On the top of the homepage, there will be search bar that will allow users to search for users by name or username. For example, I could search for the user listed above by searching for “Michael Swart” or “@mike”. Searching “Michael” will return a list of all friends named Michael. If time permits, functionality to search for specific bets whose message contains part of the search will be added. For example, a user could search for a bet made on the Warriors by searching “warriors”.

There will be no autocorrect feature, so terms must match exactly.

Placing a bet
There will be a button on the homepage that allows a user to post a bet. They will be brought to a page that has the inputs like “Bet message” and “amount” and allows the user to provide a

time for which the bet is active (if the Princeton basketball game starts at 7:30, then the bet should not be allowed to be accepted afterwards).

There will be an option of setting up an opponent from the outset, which means that the bet will not be posted as open and instead will already have a listed opponent.

Since this process is very important and a little confusing, here is a specific example. A Princeton student thinks that the midterm in COS 340 that they just took was very difficult and that the average score on the test will be below a 75%. They are confident enough that they are correct that they open up Wager and post a bet with the description “The average on the COS 340 midterm will be lower than 75%.” They put the amount at $10. The professor said that exams will be returned after Spring break, so the student sets the end time of the bet to 9:00pm on the Sunday of break. They post this bet and wait for one of their classmates or friends to accept it.

Featured Bets
There will be a group of featured bets that users can click on easily and choose their side. These bets are ones that we expect many users to bet on as the context is popular. Though some may not be, many of these featured bets will be sports bets that have the Vegas odds already worked into a message, so the user must only choose a side and amount. We will scrape the web for these odds, so many of these featured bets will be automated. If a user chooses a side for one of these bets that their friend has already bet against, then a prompt to accept this existent bet will come up before they create a new one.

If one of these bets is not sports related (i.e. “Leonardo Decaprio will win another Oscar this year”), it will be manually entered.

Accepting a bet
A user sees a posted bet and can click on it to accept it. They are prompted to enter a response message, which will allow them to explicitly state their side. This bet is then no longer part of the posed bets and moves to the active bets category.

For example, if a user saw the COS 340 midterm bet and expected the average score to be an 85%, then they could accept the bet and say explicitly “I bet that the average will be over 75%” or they could say something like “no chance”.

Completing a bet
One of the two people involved in the bet can end the active bet once the outcome is determined by pressing a “end bet” button. This brings up a screen that allows the user to choose a winner (“me” or “him/her”) and write a message. The other user is then prompted to accept the victory or loss. If accepted, then the losing person can link their venmo account and can pay through that or can choose to pay elsewhere. Once the user has paid, they must log that they paid and the winning party must accept the payment. If they do not pay, their player rating drops.

For example, the COS 340 midterm statistics are released and the average is a 71%. User A then goes in and ends the bet giving the reason “The average was a 71%!.” User B now must accept the loss and they are prompted to pay the wager amount. Once User B pays, User A then says that they did actually pay up and then the bet is now part of the archived/completed bet list.

For featured bets, the bets might auto complete depending on if we have time to implement this.

Design

We will create an iOS app using Swift and Xcode.
The database will by MySQL
The backend will by in python.
We will be using a github repository.

There will the following tables in the database

(a) bets will contain the following headers: bet_id, challenger_uid, accepter_uid, accepted_datetime, wager_amount, subject, accepter_response
(b) users will contain the following headers: uid, full_name, username, gender, venmo_id, facebook_id, profile_picture (large binary object), user_since (date), user_rating, description
(c) bet_status will contain the following headers: bet_id, is_opponent_chosen (boolean), is_complete (boolean), complete_message, loser_uid, winner_uid, is_paid (boolean), featured_id (-1 by default if it was not featured), paid_time
(d) bet_likers will contain the following headers: bet_id, liker_uid
(e) bet_commenters will contain the following headers: bet_id, commenter_uid, message, message_datetime
(f) featured_bets will contain: featured_id, message, is_sport (boolean), team_A, team_B, plus_minus

Depending on how we choose to have users sign up, a username could be unique and therefore we could get rid of uids all together, instead indexing on the unique username.

Timeline

Website created for the project
database set up and web scraper set up
Facebook integration (friend lists, profile pic potentially, etc)
Bet creation in app (link database/python handler with the actual swift implementation)
Likes and comments
Bet flow completion

Risks and Outcomes

The web scraper could provide difficulty because a source is unclear at this point. ESPN's API is private and other sources are not as accessible. The good news is that the featured bets are not a part of the core loop of the app, so if it does not work we can omit that part and the app will still provide the intended service.

Bet acceptance and payment acceptance could provide a problem. In an ideal world, everyone would be honest, but we know that this is not true. If someone ends a bet saying that they won and they actually lost or if someone lost and then does not accept the defeat, we are unsure of how to handle this problem eloquently at this point. Similarly, if a user paid but then the winner says that they did not pay, then there is also a problem. Fixing these could be an issue that we encounter. Hopefully, the social media aspect will help us and we will be able to use the larger population of users it to our advantage (i.e. my friends can down vote a false claim for winning).

Facebook integration is not necessary as well. We could just make users make an account with us and then add friends by username (as is true with snapchat). If the facebook integration is too difficult, then we could do it this way and it would be much simpler. It would, however, be less ideal in finding friends.