6.170 - Project ibetcha
Team: Zulsar Batmunkh, Saadiyah Husnoo,
Dong Hyug Lim, Dana Mukusheva

Team Work Plan

Stakeholders

Our app is geared towards users of any age or group. All that is needed is a Venmo account to process transactions.

Our app has a potential to become an additional service of Venmo, since it will attract more customers to use their system in the everyday life. Thus, Venmo Developers division is a potential stakeholder. Also, the app can be redesigned to be a Facebook application and work within Facebook, which will ease the “friending”, “reminding” and “inviting” features.

The admins of the website are the developers of the app, and a general user does not have an access to make changes in any data except for the specified cases (checking off).

Resources

Our app is designed to interact with Venmo. However, since Venmo handles money transaction, the app needs to be approved by the Venmo Developers division in order to access and use their API.

There might also be legal implications because of the nature of our app, which involves betting (sports betting is illegal).

Security enforcement is also a major aspect, given that our app deals with monetary transactions.

Tasks

- ~~Research on external libraries~~
- ~~Implement authentication (passport.js - user signup/login)~~
- ~~Make bets feature (specify content, times, frequency, and monitors, etc)~~
- ~~Add test suites as the we work~~
- ~~Decide on API to be implemented~~
- ~~Document design decisions~~
- Use Venmo API *(after MVP stage)*
- Make “friending” and “invites” feature
- Handle reminders of the pending bets
- Implement automate emailing/updating (using cron job)
- Design client side code (using Angular.js)
- Design User Interface *(after MVP stage)*

Security Concerns

- Since money is involved with ibetcha, there’s a lot of risk involved regarding user protection. Extensive measures should be taken to protect users from unauthorized charges, credit card frauds or theft of personal information or other crimes.

- Not having robust security system, thereby exposing our website to cross-site scripting attacks. As a result, attackers could inject scripts into our app and users could be redirected to web content controlled by the attacker under the guise of our app.
- Not correctly implementing authentication and session management might allow outside attackers to compromise users' passwords or personal information and assume users' identities.
- Hackers might get into our system disguised as authenticated users and might disrupt the app, interacting with other users and falsely checkoffing them to hoard their money.
- Malicious users might try to drown our server with a vast number of requests or actions that will degrade our app performance (could create multiple number of bets with maximum frequencies, and creating many pending checkoffs, that would require a lot of memory repeatedly and sending repeated emails to users).
- When users invite other users via email, we need to make sure that no other information is exposed, and the url included in the email is not used for purposes other than signup or invitation
- We need to make sure that a user cannot spam people via our email functionality.

Mitigations:

- To secure users personal information further, we use Passport.js to authenticate the users and hash passwords while stored in the database. Moreover, we'll minimize the sensitive data stored in ibetcha by using Venmo to secure money transfers. Therefore, all transactions will be processed through Venmo and users' credit card and other personal information would not be stored in ibetcha.
- We ensure that monitors can be chosen only from the friends list; thus no strangers can modify user's profile and his/her ongoing bets. We also encourage interaction between users as an informal identification to identify outside attackers as early as possible. Moreover, we could also introduce extra security questions whose answers are only known to the trusted parties to secure user identities.
- Furthermore, most of the objects created in the back-end are immutable and safely stored in the MongoDB. Only recognized and authorized users can access and mutate fields. We will also sanitize our inputs.
- For all API requests, we can check for the login information and return false if not logged in.
- In all emails, we do not expose any personal information. In addition, we can check if a user has an email before sending an email, or we can make the email a required field for signup process.
- To prevent spams, we can limit emailing non-users to just invites. The rest of the emails, we check if the recipient of the emails are the members of ibetcha. Also, the user does not get to write the content of the emails.

Minimum Viable Product

Our plan for MVP stage of the development is to finish the main functionality of the app, including user signup and authentication with passport.js, creating bets, inviting friends via email, remind the monitors about pending checkoffs and notifying involved users about the outcome of the bet. Users will be able to view profiles of other users to see the statistics of the bet history.

On this stage we postpone interactions with Venmo API because we need to get approval to use their API. Therefore, the actual money transfers and adding Venmo account are not functional. Instead, we will implement a system to track the amounts owed as well as to whom the amount is owed(payee). The payee can then “clear dept” to mark that transaction as completed. In this manner, we can simulate the money transfer mechanism without involving actual money.

Also, until we resolve all the issues with APIs, we do not proceed with the design of User Interface.