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

The following is the list of major tasks that need to be tackled with according timeframes. The completed tasks are crossed out and the members responsible for tasks are mentioned in parenthesis. Currently, we collectively discussed and planned our design and from our general consensus and depending on each member’s interest, we divided our work such that Dana/Saadiyah will work on making bets and its corresponding features, Zulsar will work on the users and authentication system and Jonathan will work on the “friending”, “invites” and automated emailing features. Each member would also be likely to be working on the client side code and the user interface of the features. Nevertheless, there also might be slight modifications to this allocation later.

From Nov 4 -> Nov 11: (Design + Teamwork Plan)

- ~~Research on external libraries (All team members together)~~
- ~~Implement authentication (passport.js - user signup/login) (Zulsar)~~
- ~~Make bets feature (specify content, times, frequency, milestones and monitors, etc) (Saadiyah/Dana/Jonathan)~~
- ~~Add test suites as the we work (All members add tests to each feature they’re working on)~~

- ~~Decide on API to be implemented (All members together)~~
- ~~Document design decisions (All members together)~~
- ~~Start working on MVP: coding, coding (All members together)~~

From Nov 11 -> Nov 18: (MVP implementation)

- Make “friending” and “invites” feature (Jonathan)
- Handle reminders of the pending bets (Saadiyah/Dana)
- Implement automate emailing/updating (using cron job) (Jonathan)
- Design client side code (using Angular.js) (All members together)
- Design basic User Interface for MVP (All members together)

From Nov 18 -> Nov 25: (Revised Design)

- Revise our design and update design documents according to the mentor reviews given for earlier design and changes resulted from MVP implementation stage (All members together)
- Add data design and code design challenges to our design (All members together)
- Start planning for the programming stage changes to be made to MVP as a result of design revision and other challenges that we might encounter (All members together)

From Nov 25 -> Dec 2: (Coding)

- Use Venmo API *(after MVP stage)* and start interacting the app with Venmo (Zulsar)
- Design User Interface *(after MVP stage)* (All members together)
- Implement the changes that will result from earlier stages and mentor reviews (All members together)
- Polish all features and make them fully functional (Bets -> Dana/Saadiyah, Emailing/Reminders -> Jonathan, Authentication/Venmo -> Zulsar)

Risks

1. The pace of the team meetings is slowed down by the distractions and unrelated conversations. Such risk can result in a longer and less-productive team meetings.
   Mitigation: We set time limits to complete a task, thus if we are getting distracted, somebody mentions the deadline and the project is back on track. *(This has been tried and tested during our meetings and has worked well so far).* Also, we disallow the usage of Facebook and other potential sources of distractions as much as possible.

2. A teammate has a harsh week with exams/deadlines for other classes and cannot participate in team meetings for a couple days.
   Mitigation: The busy teammate should give heads up to the rest of the team, keep track of major code/documentation changes and do some work after he/she is done with the hell week.

3. A team member incorporates a package/ functionality that badly breaks a feature very close to a deadline. For example: A teammate produced a bug that halts the progress of the rest of the team and cannot detect it. This may lead to the violation of time constraints imposed for each part.
   Mitigation: The other team members will collectively help the member to find the bug while some members still continue working on the other features. Since we work together at the same place, the one who coed up any part being affected can help in the debugging.

4. Team members who are working on the same feature have different opinions during the implementation. Some members might have new nice ideas and want to incorporate them and change the old design structure.
   Mitigation: When the conflict arises, the members should discuss it with other team members as soon as possible and all members will collectively discuss pros and cons of each choices and will choose by the majority of opinions. Also, the teamwork contains a tentative task distribution, and all the digressions should be announced and discussed either via email/during the meeting.

5. Some teammates feel that they might not be able to finish their tasks by a given deadline:
   Mitigation: Other teammates will jump in and help finish that task because it is more important to get that done and move on. If the task is very complicated and need a lot of time and other members have also many more things to do, members will collectively discuss the design over again as soon as possible and might decide to choose a different implementation.

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.