**Client meeting report: RealQA**

Stephanie Zheng, Lydia Chen, Ethan Cohen, Danny Segura

**User/Functional Requirements:**

* User must be able to create an account and log in with username and password
  + User accounts and passwords must be secure
  + If user fails to log in, the user will be prompted to try again, username or password may be incorrect
* Users can post questions, view questions, and answer them
  + Questions are sorted by relevance, freshness, answer count, location, and popularity.
    - Users may sort the questions according to the criteria listed above
  + Questions and answers are usually short, as users currently input through a mobile app
    - Implementation of web interface may allow further research to determine how platform (web, mobile) affects question/answer length
* Users can (up/down)vote on questions and the answers to questions, which can be ranked based on the votes
* Question recommendation system:
  + Depending on past actions (questions asked, answered), users are recommended questions to answer
  + User locale is also used when recommending questions/answers
  + Questions/Answers are tagged to further help the recommendation/relevancy algorithm
* A weekly lottery/reward system is in place to certain users who meet a set of requirements. The users are entered in a drawing for a prize
  + When a user is selected to receive a prize, they will be notified on the webapp and instructions on how to receive the prize will be available to them
* Web interface’s UI is intuitive, similar to app version’s interface
  + Simplistic to encourage posting
  + It will be menu driven, and will provide buttons, drop down menus, and pages for the user to interact with.
* Users can view their profile, answered questions, recommended questions, inbox, and lottery results through a menu
  + Users can change information and settings in their profile, such as their tags
* If users need help, there is a help menu with a tutorial, about, bug reports/feedback submissions, and an agreement document
* Utilize plugins such as Facebook API so users can query/answer through more oft-used platforms.
  + Chrome/Firefox desktop notifications
* Users may post messages to other users (chat functionality)

**Non-functional Requirements:**

* Time constraint of 1 semester
* Schedules of individual team members, clients
* When user accesses any screen in the webapp, it should load in 5 or less seconds on average internet connection speed
* Learning time for a new user should not be greater than 1 hour