RewardBot Design Proposal

I pledge my honor that I have abided by the Stevens Honor Code

Jared Sirois	3/29/20
Ruthy Levi	3/29/20
Madeleine Daniell	3/29/20

Problem Statement:

Oftentimes in a crowdsource environment engagement between users can be difficult. There is a certain anonymity in a crowdsource environment, it's in the title, 'crowd', and many users do not engage with others, they simply gather whatever information they need and typically don't contribute. This type of environment without contributing users will not allow the domain to grow and reach new heights. Anybody that looks into the domain will see only a small group of users contributing while all other users simply "mooch" off of the others. Once this becomes a standard, this domain and environment will die in no time.

Bot Description:

The proposed bot to act as a solution with this problem is a RewardBot. The RewardBot is the catalyst that will encourage the users to become more active. RewardBot is simple in design, and thus relies heavily on interactions with users. This bot's main purpose is to be the user's tracker for all of their stats. It will track everything from contributions, account status, user-to-user interactions and multiple other status. Once certain thresholds are met, for example 25 contributions to a project, the bot will "reward" the user with some set amount of fake points and a badge. Even though the points and the badge are fake, the idea of being rewarded will give the user the desire to be rewarded again, because everyone loves being rewarded.

Once the user has that desire, it becomes easy to implement more and more rewards to get users even more involved. These badges and points could also be publicly displayed on the user's profile. You could specifically make rewards in areas that you want users to interact in more. And, obviously, actual rewards can be given to users after they gather a certain amount of badges and points. For example, after a user has gathered 100 badges they could be granted certain privileges other users wouldn't have; these users would be Trusted Users and could be granted certain permissions. A nice tagline

Use Cases:

User Case #1: Reward a user for their first contribution

1 Preconditions:

-User has made an account

-User has not made a contribution before

2 Main Flow:

User makes their first contribution to a project, and the progress is tracked [S1]. After the contribution the bot rewards the user with a badge and some points, which are displayed on the users profile [S2].

3 Subflows:

[S1] A user makes a contribution to a project and that is recorded on their profile, in both front end and back end.

[S2] After every update to a profile, the bot will check the profile to see if any progress matches the criteria for a reward. If it does, the reward is publicly displayed to the user and others.

4 Alternate Flows:

[E1] The user is not eligible for a reward

Use Case #2: Promote user to Trusted User

1 Preconditions:

-User is not already Trusted

2 Main Flow:

After the user has reached at least 100 badges and an amount of points TBD [S1] the bot elects the user to be promoted to a Trusted User [S2].

3 Subflows:

[S1] After every update to a profile, the bot will check the profile to see if any progress matches the criteria for a promotion. If it does, the bot elects that user to be a Trusted User.

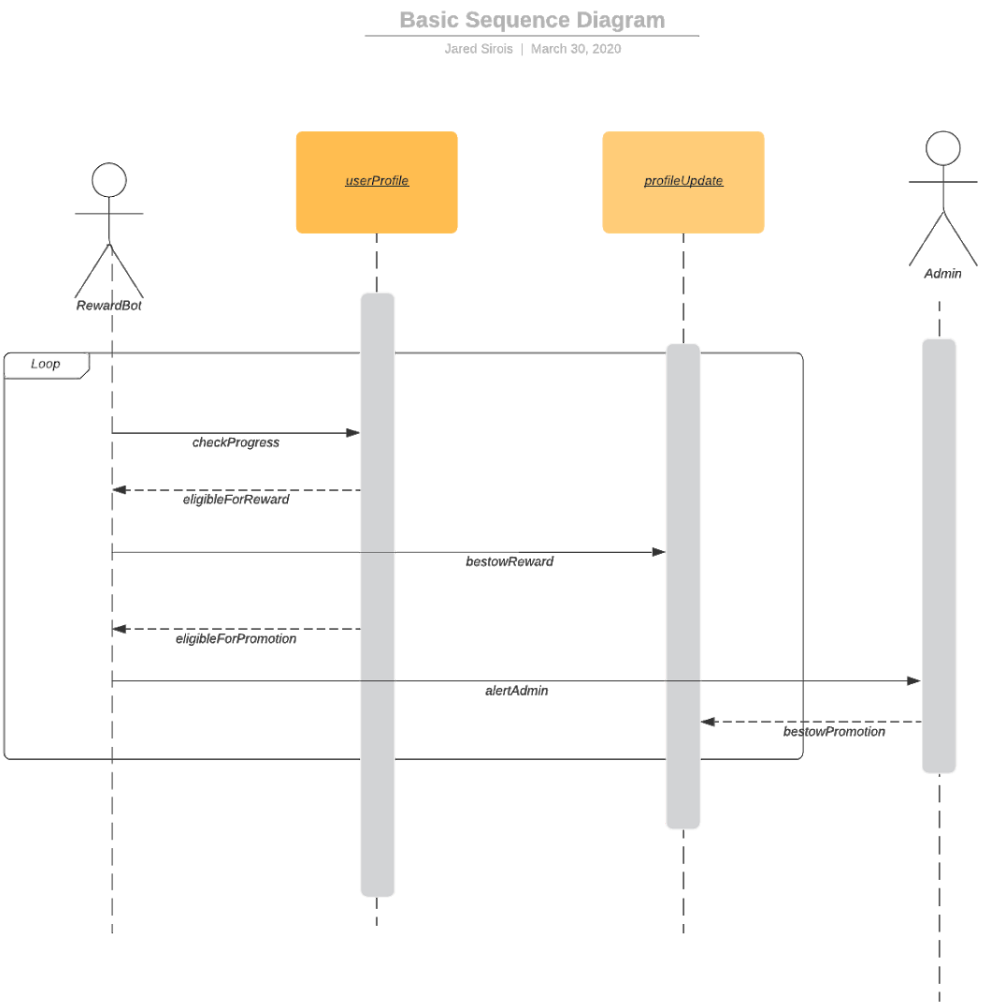
[S2] The bot doesn't automatically promote a user to Trusted User, it actually sends a notification to an admin letting them know the user is available, and the admin must promote that user.

4 Alternate Flows:

[E1] User is not eligible to be promoted

[E2] Admin doesn't promote the user to Trusted

Design Sketches:



Architecture Design:

To put it simply, there are really three main functions that make up the bot. The first function is the one that reads through all the progress on a user's profile and sees if they deserve a badge of promotion, the second function is the one that rewards a badge, and the last function is the one that alerts an admin the user is able to become a Trusted User. The first function is fairly easy to input, as all profile progress and data would be stored in the back end, and it would be easy to make the bot check the profiles each time it is updated with new info. The second function seems simple at first but might prove more difficult. This function is basically a conditional; if this criteria is

met then award the badge. However, the more rewards that are implemented it is likely this function will get more complicated as it tracks more and more conditionals. The third function will likely just be a smaller scale version of the second function; it will also basically see a conditional to see if the user is able to be promoted. This function will be smaller because not only will it not have as much criteria to monitor but it also doesn't make the user Trusted, it lets the admin know so that the admin could make the final decision.

The best language to program this bot in is probably Python; the bot will probably work best when used in communication platforms such as Slack or Discord.

Below is a diagram of a more in depth look at how the bot will function. This model gives a closer look into what the code might look like.

Ideally this diagram embodies the full code of the bot and this diagram loops every time a user's profile is updated. When referring to the “user profile being updated” this really just refers to anytime the user does anything in the application. This is because this bot will only be used in larger apps, like Slack and Discord, so in reality the bot isn’t something that can go and change things like a secure user profile for this large app, so instead the bot is really just a script that runs constantly. The decision box that is labeled “profileCriteria=badgeCriteria?” represents the bot checking if the user profile has any stats that match the criteria for a reward.