User Documentation User Story, UCSC Tutor App, 11/30/15

Overview:

UCSC Tutor is an Android app that allows UCSC students with ucsc.edu emails to find and message tutors for classes they would like help in. In the app, students are able to interact with tutors through an in-app messaging feature. Messaging allows students and tutors to schedule a time to meet up. Students are also able to search for tutors depending on their availability, ratings and reviews, and classes they are willing to tutor for. Both users (tutor and student) have a profile in which they are able to edit their bio, profile picture, and receive/give reviews. Featured in the app is a timer that calculates how much money the student owes the tutor and the payment is carried out through Venmo.

User End Usage:

If you are a student signing up for UCSC Tutor, the first step is to click “Sign-up as student” on the login page. This will allow for a UCSC student to create a user name and provide some basic information and optionally upload a profile picture. After this happens, an e-mail verification message is sent to the specified (UCSC) e-mailing address.

The same process as above is required for tutors, except for clicking “Sign-up as tutor” instead. The tutor must also add qualifying classes to which he/she can tutor as well as creating a bio on prompt.

Both users are redirected to a home page on login/sign-up completion. Students and tutors both have editable profiles, reached by an edit button. These profiles consist of a profile picture, a bio (description), credentials (for tutors), courses a user wants to be tutored in (students), and a schedule of availability. There is also a logout button for ease of use in the upper right corner of the view.

When viewing another user’s profile, you have the ability to message, view the schedule of, or start a session for tutoring with the opposing user. You can also see average ratings for the user based on three relevant categories.

Clicking on message will open up a messaging activity in which you can directly chat in real time with another user. After sending a message to a tutor (as a student) the respective users are added to each other’s contacts(reachable by a swiping left on the homepage). Viewing their schedule allows you to see what time frames you have that are compatible. Lastly, starting a session takes you to a timer which will calculate amount owed in venmo at the end of the session.

Students also have the ability to search for tutors with a search engine, reached by swiping right on the homepage. The search allows students to filter tutors by different fields such as class, ratings, and schedule.

How a developer could build/extend our code:

There are various ways a developer could make further progress with our code. First of all, the search functionality is slightly lacking in the sense that it does not currently allow students to search based off of ratings. This could definitely be done by analyzing how we're storing reviews as meta-data in Parse for each user, and then adding additional constraints to the queries being performed. One could also hard-code in the entire list of classes available at UCSC. At the moment, they are simply being stored in Parse based off of whatever strings the user enters when prompted to enter their classes. This is all we had time to implement, but without a time constraint, this is easily doable.

The overall look of the UI could be improved, mainly in the sense that we did not utilize material designs available to us. However, we did not focus on the pleasantness of the UI as much as we focused on functional aspects, so this is just a matter of rearranging and re-coloring elements that are already present in our app.

As far as payment goes, we believe that having sessions stored in Parse such that when a student and tutor start a tutoring session, their user IDs would be connected in a Session object. This would allow the developer to display a history of student/tutor interactions, along with how much was paid, etc. This would greatly decrease the liability we would currently have if we were to release this app to the public.

Messaging could also have some improvements. Specifically, the SDK we used (Sinch) seems not to always be the most reliable. It will occasionally take a long time to send or receive a message, and other times, the service completely stops working for a period of time. We would strongly encourage any potential future developer to look into a messaging service such as Google Cloud Messaging (GCM). This would allow for push notifications to be enabled (something that was seemingly impossible to achieve with Sinch), which would, of course, increase the usability of our app. That being said, push notifications in general would be a huge addition to our app. We simply ran out of time, so we were unable to successfully implement them.