Sprint Planning Document

Sprint 2

GetGuru

Team 12

Sharoon Srivastava

Ankush Jain

Pranav Punjabi

Vijay Srinivas

Murtuza Kainan

Sprint Objective

The objective of this Sprint document is to assign tasks that need to be carried out in the current Sprint. The tasks will be delegated equally among the teammates. A scrum master will also be assigned to facilitate meetings and coordinate with team members. In this sprint we'll implement the functionalities where a student can search for tutors based on ratings. Additionally, we'll allow the student to post ratings and reviews for any tutor.

Scrum master: Sharoon Srivastava

Meeting times: 3:00 PM, Tuesday & Thursday for 1 hour and 15 minutes & 5:30 PM, Monday for 30 minutes in Lawson B160.

Risks/Challenges: Whenever a student searches for a tutor, the result should take the least amount of time as possible.

Current sprint detail

The time required (in hours) to implement these functionalities is mentioned at the end of each of them.

Functional

As a student, I'd like to rate and review tutors

Acceptance criteria - A student should be able to rate any tutor and leave a review about the tutor's performance anonymously. User (student) should be able to rate anonymously.

Task 1 - Provide the requisite UI elements to the user to add reviews and ratings.

- Send the text review and the rating (5 point scale) to the backend to be stored in a database.
- Murtuza and Pranav
- 2 hours

Task 2 - Implement client side code to make request to the API.

- Network tasks will be execute each time a review is written and a rating is given. The information entered by the user will be extracted from the UI elements and be converted to the JSON format. This data will then be sent to the API for storage.
- Murtuza and Pranav
- 7 hours

As a tutor, I'd like to view my own ratings and reviews

Acceptance criteria - All tutors should be able to view their own ratings and reviews posted by any student. He should be able to view the ratings individually or together (average).

Task 1 - Allow user in tutor mode to view all ratings provided by each individual.

- Ratings provided by students for a particular tutor will be requested from the server and displayed in a separate scrollable tab fragment. Also, an average rating will be displayed in the same fragment. The tutor will not be able to edit the information appearing in this tab fragment.
- Murtuza and Pranav
- 7 hours

Task 2 - Allow easy navigation to reviews using tabs.

- The user (tutor) will be able to access the fragment from the default tabs view displayed when using the app as a tutor.
- Murtuza and Pranav
- 3 hours

As a student, I'd like to search for tutors with respect to subjects

Acceptance criteria - All students should be able to search for tutors based on subjects. There should be a drop down menu which lists all the available subjects and the student should also be able to search for specific subjects.

Task 1 - Create a UI for the user (student) in the search fragment of the TabHost.

- The UI elements of the search fragment will be created to allow users to filter tutors on the basis of subjects, ratings, and location.
- Murtuza and Pranav
- 2 hours

Task 2 - Send the user input to the server to allow it to look up in the database.

- A network task will be executed to request tutors to match the search criteria entered by a student.
- Murtuza and Pranav
- 3 hours

Task 3 - Receive data from server with the tutor info and display that in Android UI elements.

- The data will be received in the JSON format and will be parsed and then displayed in a listview so that it may be viewed by a student using the app.
- Murtuza and Pranav
- 4 hours

As a student, I'd like to shortlist tutors for future reference

Acceptance criteria - There should be a visible tab to access the shortlisted tutors. Additionally, there should be a button on each tutor profile which allows the student to easily shotlist him.

Task 1 - Add UI elements in the layout of the tutor profile.

- A tab for favorite/shortlisted tutors will be created in the default tab view for a student using the app. Creation of a layout for listing shortlisted/favorite tutors in the appropriate student tab fragment will also take place.
- Murtuza and Pranav
- 5 hours

Task 2 - Implement client side code to make requests to the API in order to store and retrieve favorite tutor information.

- The client will make requests to the API in order to fetch favorite/shortlist information as well as store it when a tutor has been shortlisted/favorited as a student. A network task will be required for each of these activities and the data will be sent and received in the JSON format.
- Murtuza and Pranav
- 10 hours

As a student, I'd like to view tutor profiles

Acceptance Criteria - When a student tries to access / view the tutor profile, a new page(?) with all the information, provided by the tutor (such as contact information, subjects, location etc) about the tutor should be visible.

Task 1 - Implement a layout and android activity for viewing tutor profile as a student.

- An activity will be created to view tutor information when a student selects a tutor out of favorites or after searching. The student will be able to view tutor information without being able to edit it.
- Murtuza and Pranav
- 5 hours

Task 2 - Implement client side code to retrieve and display tutor information in the activity and its layout.

- The client will make requests to the API in order to fetch tutor information. A network task will be required for this activity and the data will be sent and received in the JSON format.
- Murtuza and Pranav
- 10 hours

Non functional

As a developer, I'd like to store the ratings and reviews of the tutors in the database

Task 1 - Store reviews as strings

- Add a column to the existing tutor table in the mysql database to store the reviews. Make sure that all the reviews are saved and separated in a way that it's easy to retrieve them from the table.
- Vijay
- 8 hours

Task 2 - Store ratings as strings

- Add a column to the existing tutor table in the mysql database to store the ratings.

  Make sure the ratings are saved and separated in way that it's easy to retrieve them from the table.
- Vijay
- 8 hours

Task 3 - Store average ratings

- Add a column to the existing tutor table to store the average rating for every tutor. Update the average everytime a new rating is added.
- Vijay
- 8 hours

As a developer, I'd like to store the location of each tutor

Task 1 - Store location of a tutor in the tutor table

- Add a column to the existing tutor table to store the location of every tutor. Add location of the tutor in this column as a string.
- Sharoon
- 5 hours

As a developer, I'd like to return the list of tutors matched by search criteria given by the student

Task 1 - Return a list of tutors based on subjects they teach

- Retrieve a list of the tutor ids from the subjects table according to the subject searched by the user. If the user searches for multiple subjects at a time, take the intersection of all the matching lists of tutor ids. Return the list of corresponding tutors from the tutor table.

- Ankush
- 10 hours

Task 2 - Return a list of tutors based on the specified ratings

- Retrieve the list of tutor ids from the Ratings table according to the ratings specified in the request and return the list of corresponding tutors from the tutor table.
- Ankush
- 10 hours

Task 3 - Return a list of tutors based on the specified locations

- Retrieve a list of the tutor ids from the location table according to the location searched by the user. If the user searches for multiple locations at a time, take a union of all the matching lists of tutor ids. Return the list of corresponding tutors from the tutor table.
- Sharoon
- 10 hours

Task 4 - Return a list of tutors based on a combination of search queries like ratings, location and subjects

- Retrieve the lists of tutor ids from the Ratings, Location and Subjects tables matching the request. Take the intersection of the lists and return the list of corresponding tutors from the tutor table.
- Sharoon
- 15 hours

As a developer, I'd like to return the tutor's profile when requested by the student

Task 1 - Return the requested tutor profile

- Whenever there's a request to view a tutor profile, the server should access the tutor table in the mysql database and return all the information available about the tutor corresponding to the id specified in the request, such as name, email, subjects, ratings and reviews etc
- Vijay

- 6 hours

As a developer, I'd like to store the list of shortlisted tutors for each student

Task 1 - Save the list of shortlisted tutors

- Create a new column in the student table in the mysql database which stores the ids of the tutors shortlisted by the student. The server should append this column with the new id of the tutor which is shortlisted by the student.
- Ankush
- 5 hours

Task 2 - Return the information of shortlisted tutors

- Whenever a student requests to view his shortlisted tutors, the server should access the mysql database and return the basic information about all the shortlisted tutors in json format.
- Ankush
- 5 hours
-

Total approximate time distribution :

Murtuza : 29 hours

Pranav : 29 hours

Sharoon : 30 hours

Ankush : 30 hours

Vijay : 30 hours

Total : 148 hours

Rest of the backlog :

Functional

1. As a user, I'd like to receive notifications when contacted via chat
2. As a user, I'd like to view my chat history
3. As a student, I'd like to instantly connect to the tutors through chat

4. As a student, I'd like to schedule a meeting with the tutor using an in-app scheduling assistant (if time allows)
5. As a user, I'd like to set my chat status (if time allows)
6. As a user, I'd like to view other's chat status' (if time allows)

Non Functional

7. As a user, I'd like to have fast response times
8. As a developer, I'd like my database to be secure by preventing SQL injections using string validations.
9. As a developer, I'd like a RESTful API built to handle requests from a variety of clients apart from the android app
10. As a developer, I'd like to validate all user accounts to prevent redundancy
11. As a developer, I'd like the amount of allowable downtime per month to be one hour
12. As a developer, I'd like to have a development API server and a production API server
13. As a developer, I'd like to be able to switch the server (development server or productions server) the android app is sending requests to