Sprint Planning Document

Sprint 1

GetGuru

Team 12
Sharoon Srivastava
Ankush Jain
Pranav Punjabi
Vijay Srinivas
Murtuza Kainan

Sprint Overview:

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.

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: Our framework of choice is Flask. Since none of the team members are familiar with the framework, learning how to use Flask will be a priority. Client server communication using HTTP requests with JSON also needs to be understood and implemented. Moreover, communicating with the database is new for all of us, therefore it might take up a significant amount of time to learn.

Current Sprint Detail:

<u>Functional</u>

As a user, I'd like to create an account

Task 1 - Create UI for the signup page, displaying all information the user needs to enter

- Design the elements and layouts to be spread upon the signup page. Create a form that allows user to add personal information.
- Pranav and Murtuza
- 10 hours

Task 2 - Implement an action when the user clicks the submit button, highlighting the incomplete sections, if any

- Establish page validation to check for completeness of mandatory fields in form. Respond to user with an error page urging to complete the form.
- Pranav and Murtuza
- 5 hours

Task 3 - Send request to server which contains the account information to store in the database

- Establish connectivity to the server using and send request using JSON format. Allow server to modify and update database with information from client end.
- Pranav and Murtuza
- 7 hours

As a user, I'd like to toggle between student and tutor views

Task 1 - Create profile pages for 'user as a student' and 'user as a tutor' for the android client

- Design the elements and layout for the profile pages and write the client side code for toggling between student and tutor profile pages.
- Pranav and Murtuza
- 10 hours

Task 2 - Allow user to add subjects of expertise in the tutor view

- Add UI elements such as the add button and a custom textview to display entered subjects.
- Pranav and Murtuza
- 3 hours

Non functional

As a developer I'd like to explore the functionalities within Android Studio

Task 1 - Learn layout schemas

- Learn how to implement various layouts in using the visual layout tool and XML. Layouts that are required for the app include but are not limited to tabbed layout, relative layout, and frame layout.
- Pranav and Murtuza
- 5 hours

Task 2 - Understand various tools needed to implement user interaction and navigation

- Learn about and implement UI events for different UI elements. Implement event listeners and reference UI element from the layout to event listeners for navigation and user interaction.
- Pranav and Murtuza
- 20 hours

As a developer, I'd like to set up a back-end server and respond to basic HTTP Requests.

Task 1 - Learn the syntax and programming rules for Python

- Download the latest version of python and run small codes to understand its working.
- Sharoon, Ankush and Vijay
- 5 hours

Task 2 - Download and install the Flask framework. Read and understand how it works

- Make sure that the Flask framework is setup properly on all our computers. Try implementing simple programs to understand how Flask works. An example would be returning a message on receiving a GET request from a client.
- Sharoon, Ankush and Vijay
- 5 hours

Task 3 - Implement a RESTful API using the Flask Framework in Python to interact with the client

- Implement classes in the RESTful API that would handle POST and GET requests from the client. Implement a class which would parse and return data in the JSON data format from the client.
- Sharoon, Ankush and Vijay
- 10 hours

Task 4 - Setup a simple HTTP server implemented using the RESTful API

- Setup an HTTP server on the localhost. Make sure that the server and client communicate efficiently and that the server is able to handle HTTP requests from the client.
- Sharoon, Ankush and Vijay
- 15 hours

As a developer, I'd like to set up a back-end server and respond to basic HTTP Requests

Task 1 - Understand how the back end server and MySQL communicate via the SQLAlchemy module

- Make sure that MySQL database is setup on all our computers. Try implementing simple programs to understand how SQLAlchemy works.
- Sharoon, Ankush and Vijay

- 10 hours

Task 2 - Implement the RESTful API using the Flask framework in Python to interact with the database
- Learn how the RESTful API interacts with the database. Implement the RESTful API to so that the server can communicate with the database and make HTTP requests.
- Sharoon, Ankush and Vijay
- 15 hours

Task 3 - Build tables in the MySQL database for different tasks and store/retrieve information from them
- Learn how to build tables in the database using a database management application like Sequel Pro and how to store/retrieve the required information from them. Then, implement database management classes in the back-end server which can make similar requests to the database.
- Sharoon, Ankush and Vijay
- 10 hours

Task 4 - Store information in MySQL database when there is a POST request from the client
- Add functionality in the database management classes in the back-end implementation to store information in the correct table whenever there is a POST request from the client.
- Sharoon, Ankush and Vijay
- 10 hours

Task 5 - Retrieve data from MySQL database when there is a GET request from the client

- Add functionality in the database management classes in the back-end implementation to retrieve data from the correct table in the database whenever there is a GET request from the client.
- Sharoon, Ankush and Vijay
- 10 hours

Total approximate time distribution :

Murtaza : 30 hours

Pranav : 30 hours

Sharoon : 30 hours

Ankush : 30 hours

Vijay : 30 hours

Total : 150 hours

Rest of the backlog :
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 tutor, I'd like to view my own ratings & reviews
4. As a student, I'd like to search for tutors based on location
5. As a student, I'd like to search for tutors based on ratings and reviews
6. As a student, I'd like to search for tutors based on subject
7. As a student, I'd like to shortlist tutors for future reference
8. As a student, I'd like to view tutor profiles
9. As a student, I'd like to instantly connect to the tutors through chat
10. As a student, I'd like to rate & review the tutors
11. As a student, I'd like to schedule a meeting with the tutor using an in-app scheduling assistant (if time allows)
12. As a user, I'd like to set my chat status (if time allows)
13. As a user, I'd like to view other's chat status' (if time allows)

Non-Functional:

14. As a user, I'd like to have fast response times
15. As a developer, I'd like my database to be secure by preventing SQL injections using string validations.
16. As a developer, I'd like a RESTful API built to handle requests from a variety of clients apart from the android app
17. As a developer, I'd like to validate all user accounts to prevent redundancy
18. As a developer, I'd like the amount of allowable downtime per month to be one hour
19. As a developer, I'd like to be able to test on a local development server easily.
20. As a developer, I'd like to have a development API server and a production API server
21. 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