Team 1: Sprint 1 Retrospective

Project Coordinator: Hasini Gunasinghe

Project Name: Frame

Craig Wilhite
Daniel Sokoler
Grant Jochums
Adam Worthington
Raaghavendar Karthikeyan

What Went Well?

In sprint 1 we created a functional user interface that is capable of taking in a string input, encoding it into a JSON packet, and sending it to the server. The server is receives the JSON packet and decodes it into the relevant information. That information is then stored in the SQL database. We are also able to send a request to the server to receive the top ten text entries and then display them in a pop up message.

Completed tasks:

- Built a class to encode and decode JSON as well as a document outlining the exact message formats expected.
- Created a simple MySQL database using Google's Cloud SQL storage. This database stores strings, the string's ID number, and the date and time that the string was added to the database so that the user can store and retrieve posts.
- Created a user interface that easily allows for the creation of a text post. The central media feed was created during this sprint. This is the main view that users will both and see and interact with within the application. Additionally, parts of the peek feed view were created.
- Created a RESTful interface for the client side to communicate with. This interface interacts with storage components offered by Google App Engine. From the engine we utilized the SQL storage and web hosting tools. The data transmission of the interface is handled through JSON packets.

What Didn't Go Well?

We were able to complete all of the user stories we set out to accomplish, however this sprint didn't go perfectly. Our git repository is messy and needs to be straightened out, when we receive the top ten posts from the server they only appear for about three seconds before disappearing. A second issue that we faced was the integration of the front end (the app on the phone) with the back end. Restful communication on Android devices was a little bit different than what we were expecting, and we did not realize that on Android devices network communication was not allowed to use the main thread.

How should you improve?

Our ability to realize what types of work take what amounts of time was our biggest issue this sprint. Taking more time midway through the sprint to examine what we've done so far and how far we have to go will benefit us greatly, as it would help us determine when we need to begin integration of our parts.

Solutions:

- Solving our messy repository will be simple. The fastest way to correct the repo would be to re-push all of the files in an organized manner so they overwrite the files currently there. Then we need to stick to the layout for the repo and only deviate from it if absolutely necessary. An alternative to the correction of the messy repository would be to utilize the tool Maven. We believe that a Maven project will be optimal due to the readability with respect to code and usability with respect to dependencies.
- The reason the top ten text posts only displayed for a few seconds because we never intend for our application to display a text post. Later on we want text submitted to be turned into an image and displayed accordingly. While it was not ideal to have the top ten posts disappear, it was really more of a way to ensure that we could successfully communicate with the backend. In future sprints we will be populating the display views with the posts recived from the server.
- The integration of the project as a whole will be less unwieldy in the future sprints due to the current integration being completed. When adding new key features to our project in sprint 2, we can simply integrate and test them right away now that we have a working project.