

Team 1: Sprint 2 Retrospective

Project Coordinator: Hasini Gunasinghe

Project Name: Frame

Craig Wilhite

Daniel Sokoler

Grant Jochums

Adam Worthington

Raaghavendar Karthikeyan

What Went Well?

In sprint 2 we fully implemented picture as well as integer based functionalities such as rating and flagging. The phone encodes a picture in base 64 which then is sent to the server via a JSON packet ultimately being stored in a mysql database as a text datatype. The communication portion of our app seems on point and relatively in its final state for the presentation.

Completed tasks:

- Built a class to encode and decode JSON as well as a document outlining the exact message formats expected.
- Created front-end elements to take picture, review the picture before posting, and send the picture to the server.
- Created UI elements to allow for focusing on a media content item in the media feed. This allowed for the focused media content item to assume all screen real-estate and also show future comments associated with the media content.
- Updated the MySQL database to store images and their various attributes. This includes the picture's ID, it's vote number, it's number of flags, it's gps location, and it's tags.
- The database is also set up to hold comments and their attributes. This includes the comment itself, the ID of the user who posted the comment, and the ID of the post that the comment refers to.
- A database manager was created to separate the storing and retrieving of media from the Restlet communications. It generates MySQL queries to get and store posts, as well as update them (upvote, downvote, flag, etc).
- 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 see and interact with within the application. Additionally, parts of the peek feed view were created.
- Extended the implementation of the RESTful interface to deal with pictures, voting, and flagging in both the GET and POST methods. The interface now offloads the sql workload to the new sql manager class.
- The JSON api and standard are up to date for all required communication packets including pictures, video, text to image, voting, flags, search tags, comments, and locations.(note that video is still being worked on)
- Created the front end for taking videos through the inbuilt phone camera, but the video shot is not being sent.
- Text to image conversion is ready to be integrated into the application and then tested.

What Didn't Go Well?

We were unable to reach all of our goals for this sprint including video and text to image as the primary examples. We ran into issues like the file size being too large and the information not being transmitted correctly when we went to actually send the files to and from the device and the server. So we made the decision to prioritize everyone's focus on getting pictures functional first and then continuing onto video and text to image. This took longer than expected. Our text to image has code written, but was never implemented or properly tested due to time restraints. For video we only need to find a way to stream the file back and forth reliably because it is too large to be sent all at once.

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 to begin the integration of our separate parts earlier. This would enable us the ability to catch bugs and problems sooner.

Solutions:

- The reason that the application could only display one image at a time was due to memory issues with the android device that we are using to test. Our image took up almost the entirety of the phone's memory, limiting the number of pictures to one. We fully plan on finding a solution to this during sprint three, as our app is based entirely on displaying many pictures in an organized fashion. We managed to complete picture communication from the server to the device, so focusing during Sprint 3 on refining how we store and process our images will help with this problem.
- The integration of the parts of the project dealing directly with picture processing will be less unwieldy in the future sprints due to us having a functional (albeit unwieldy) method of dealing with pictures. In sprint 3, when we want to try out a different method, we can use the better parts of that method, and integrate those with our already working implementation to speed up our integration and testing.