Introduction

This project will be developt by using the concepts of agile methodology. Since our team only consists of two members, we don't have a clear definition of who is the scrum master or product owner. We are doing the job of those concepts together. We are responsible for developing a mobile application for Android OS. We are working collaboratevly with the web part of the system, Team YeaH! who are responsible for the web application part of the system. The project will last for two sprints. From the first feed-back that we had from first part of the project documents. We've decided to carry some of the agile methodology concepts of our project to an online tool for agile development, pivotaltracker. The reason behind it is that in order to increase the dynamism and trackability of the product backlog and user stories. This document will include;

- A summary of the project.
- Our approach to the solution, from the aspect of a mobile application.
- The technology and the architecture that will be used.
- A domain model to have an overall idea over solution.
- The constraints of the application and will be mentioned how the application should behave.
- Also it will be explained that the meaning and limitations of abstract terms that will be mentioned.
- Definition of done.
- An overall modelling diagram which includes what is done in the first sprint.
- The review of first sprint.
- Planning part of second sprint.
- The review of second sprint.
- It will explain the minimal viable product and the potentially shippable increment of second sprint.
- All the user stories and backlog can be seen from here.

Project Summary

Hundreds of millions of digital photos are taken annually and most end up at some place in a hard drive, possibly never to be viewed. This project would use the geo-tagging and possibly other metadata of digital photos to produce an archive to enable the users to browse and find photos by a map-based navigation interface thanks to the geo-tagging information in the JPG headers. Other features the system would provide include adding new photos to the collection, removing existing ones from the collection, updating of photo information, along with some of the basic metadata type searching and functionality that conventional photo management tools support.

Overall solution from a mobile application

From the concept of being mobile, app part will provide taking photos from the mobile device and also it will handle geo tagging which makes the diffence compared to web application. The taken photos will be uploaded to system and will be included in an event which simply is a collection of photos. By providing that we will be ensure to keep users photos in an order by location. The application will do the all of the upload, download operations smoothly, without interrupting the user's work. The application will let user to navigate between events, and navigate between photos when an event opened from the map. Lastly, of course it will provide adding, deleting and updating of photos, events or information about them.

Technology and Architecture

- Mobile application will work on Android OS.
- Project will developt using Android Studio IDE.
- Google Map API to provide a dynamic well detailed map.
- Imgur API for downloading and uploading purposes.
- Also Imgur will provide storage for images.
- Junit and Android Instrument Tests will be used for testing.
- Pivotaltracker to handle agile concepts.
- LaTeX will be used for preparing documentations.
- Heroku will be the server for whole system.
- JSON format to provide some data flow between the website and app.
- PostgreSql for database.

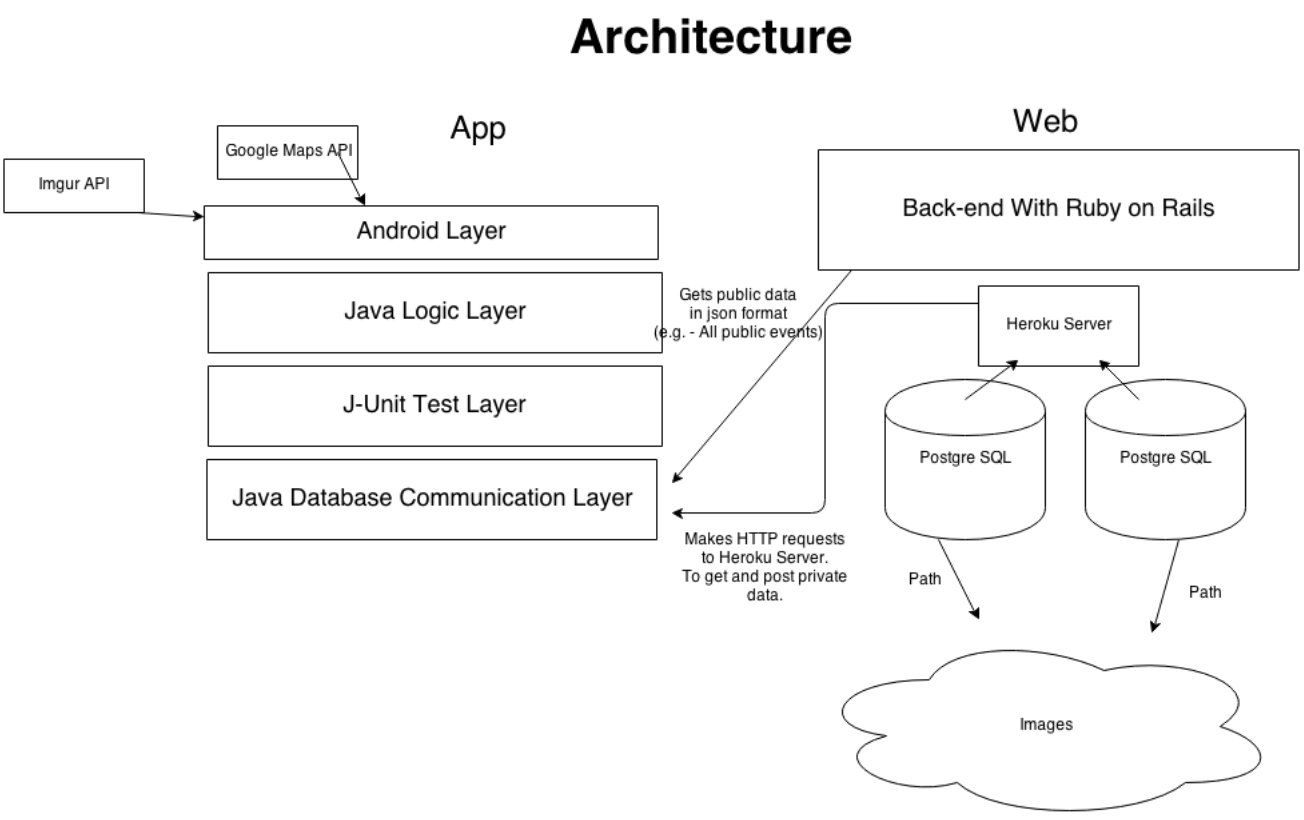

Figure 1: Architecture

Domain Model

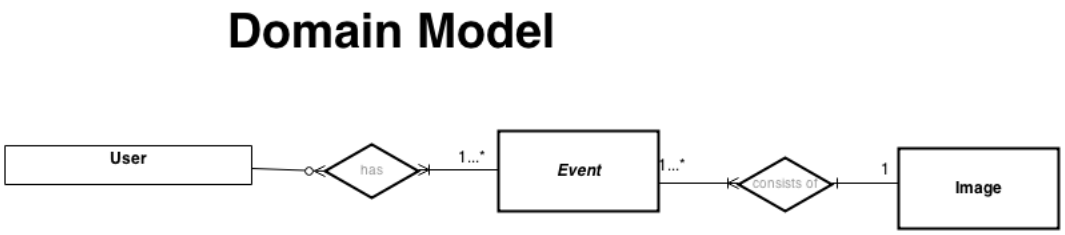

Figure 2: Domain Model

Constraints

Constraints for development

The most effective constraint of this project in the development part is that it is a classroom setup. Thus, the project almost has no budget and also a very short time of development phase. Building such a system in reality would require much more budget and time to have a fully-working system. The application requires constant fixes, restorations to be able to work smooth, without crashes.

Constraint for end-users

- Photo Map Navigation system's app part is only built for Android OS.
- The system can only work with JPEG files.
- The files that will be uploaded should be location tagged, either with the photos taken with Photo Map Navigation mobile application or some other application that can location tag JPEG files. Otherwise system will ignore files that don't have location information.
- Tagged location information should be in degrees-minutes-seconds format.
- However, they will be represented in a map by latitude and longtitude values.
- Photos will be uploaded to the Imgur anonymously. If the URL of this file is shared, it can be seen by anybody, eventhough it is a private file.
- A **public event** means that files and informations in it can be seen by anybody, eventhough it is an unregistered user. It is conributable if permission is given.
- A **restricted event** means that files and informations in it can be only seen by registered users who have given permission by the creator of event. It is contributable if permission is given.
- A **private event** means that files and informations in it can be only seen by the creator of the event. It is not contributable by any other users.
- **The creator of the event** is determined by who initializes, takes the first photo of event.
- An unregistered can't take photos, one can only see public events.
- The mobile application is memory dependent. Working it on a low memory device or with highly loaded memory can crash it easily.

Definition of Done

- Working collaboratevly with web team.
- Map is loading efficiently.
- All code is working.
- All unit tests passed.
- Database interaction is working.
- Whole project pushed into GitHub.

Overall Class Diagram

In UML format;

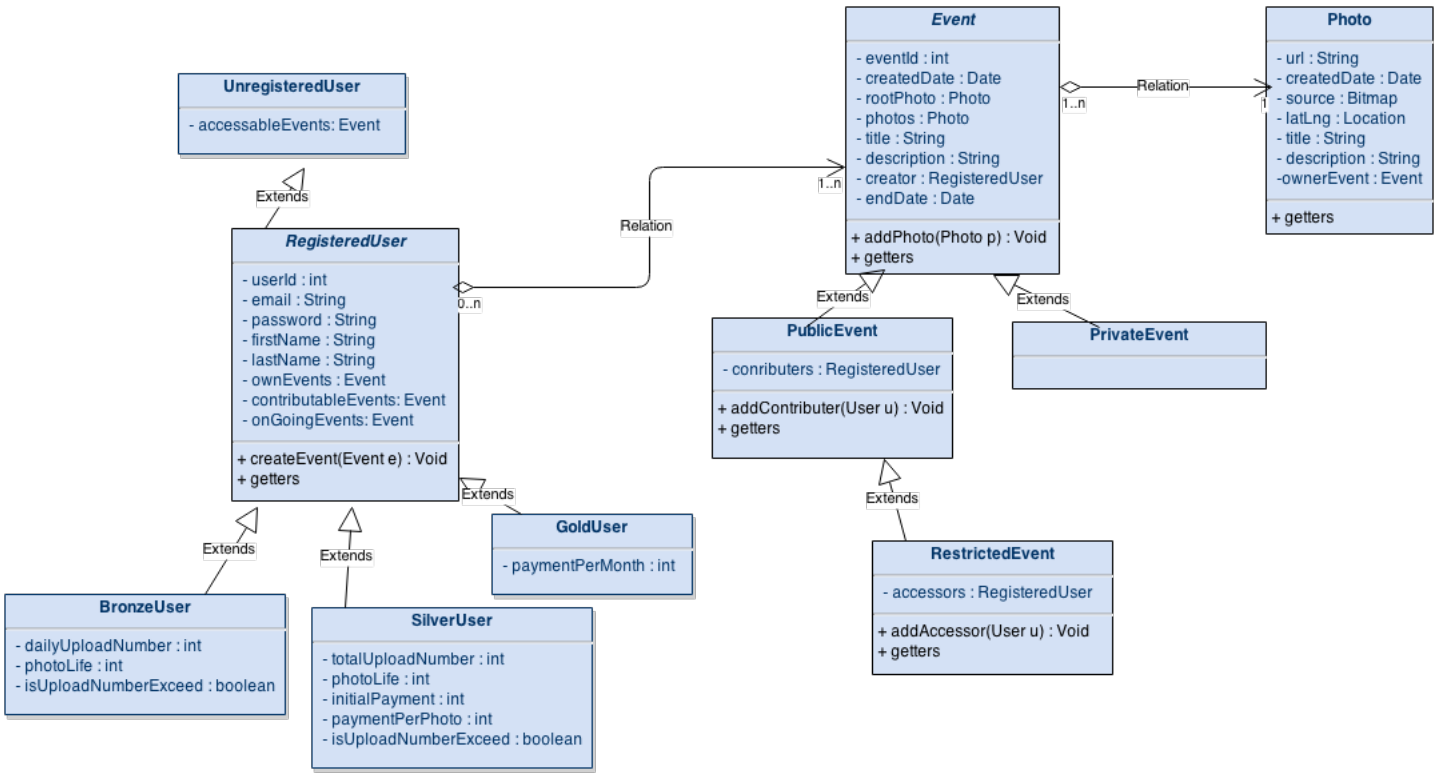

Figure 3: Class Diagram

First Sprint Review

BurnDown Chart

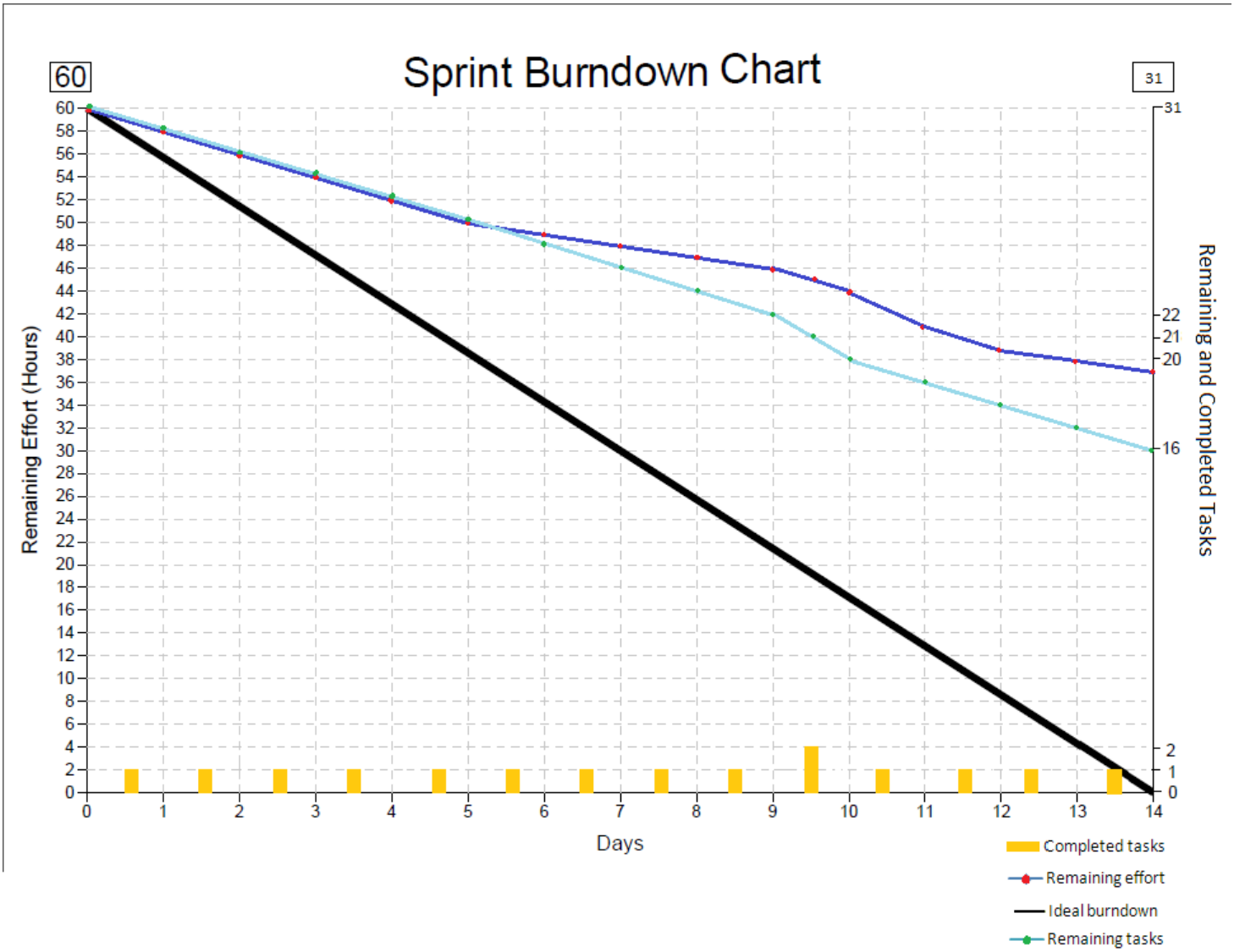

Figure 4: Burndown Chart

Simply first sprint is failed, under the definitions of *there are still exists a lot of tasks that are undone*. However, when we considered the experience of the team and constraints of the project that are mentioned above, the progress is admirable. Also, another problem that we've faced is about GitHub repositories. Since the web team uses the framework Ruby On Rails which uses an ORM tool to deploy it in directly from GitHub into Heroku server, the projects being in the same repository interrupted that vital task. Thus, we ended up having no server to communicate no data to flow into our mobile application. However, currently we have an application that can take pictures, geo taggin, uploading to imgur, and seeing them on a map by location without the concept of an event. When the repository problem fixed which we will seperate the repositories soon, the communication between server and also

web team will increase rapidly. Thus, we are hopeful that we will have a much better velocity in second sprint and we will cover the main concepts of the project.

Second Sprint Planning

Since the first sprint failed in a sense, our first job is to catch up fastly. The first aim is to provide data flow between app and server so that we can implement concepts of event, registration and logging in. To have a better understanding, the backlog and user stories are online.

PSI of second sprint

The concepts of events which are collection of photos will be provided by the definition under public, restricted or private. Events will be represented on a map by one photo which is called root and the whole photos on an event can be seen by clicking the root photo on the map. The second sprint will also cover up registration and logging in. Also users will be allowed to delete, update the photos and informations about them. The registration will have three types, bronze, silver and gold which will have the restrictions or the features that are provided to us in the definition of project document. User will be able to navigate between events' root photos on the map.

Second Sprint Review

BurnDown Chart

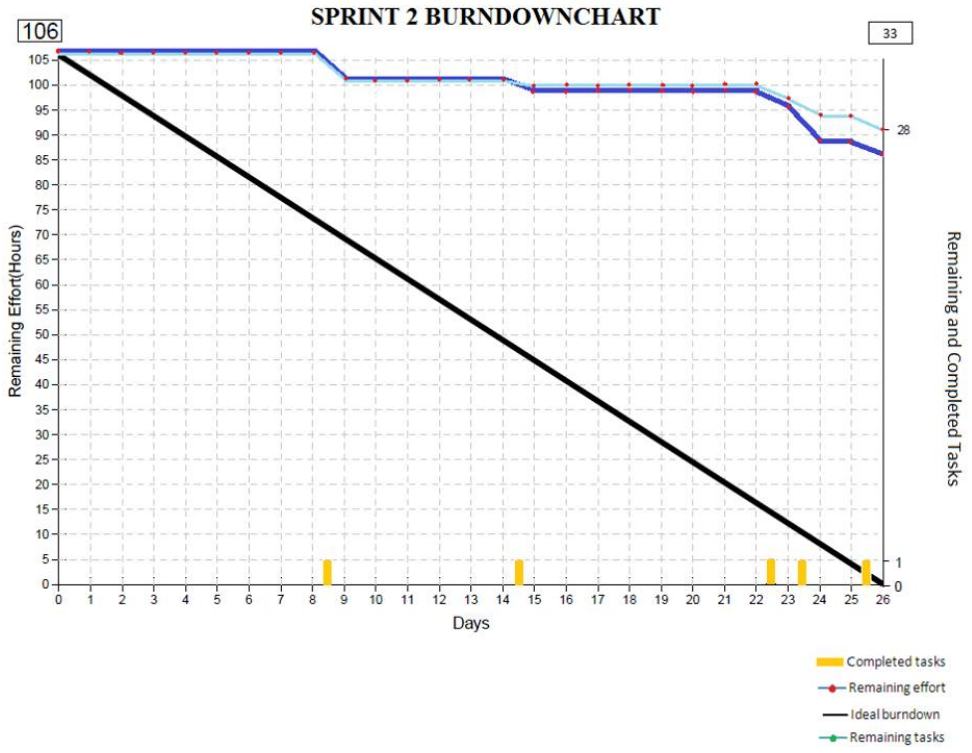

Figure 5: Burndown Chart 2

The reason that a lot of work is not done is because of we still cannot communicate with web part as also mentioned in the first sprint's review. However, we have started over our project to implement model-view-presenter design pattern, which suited our project very well. We didn't try to refactor the code that we wrote in first sprint to implement the model-view-presenter because refactoring would take much more time than implementing a new one. We have accomplished to write some of the test cases of our application which are vital and prior to system. Most of the time we've waited for web team to implement the API that we will use by doing HTTP Requests to provide data flow. All in all, we have a hope that we can succesfully finish our aim for the MVP, to reach the MVP, all we need is the some initial connection with the RESTful API that web team will accomplish. If not so, maybe we can try to write a simple server for our own, whereas there is a chance that we may not finish it on time, with the additional work of implementing the server and creating the database.

MVP

The MVP that we desired to have is that allow users to take photos, uploads them to the server and can see them on a map by location.