Individual Report

Student Name: Ga Jun Young
Student ID: 16440714

Week 1 (11th September 2019) – Forming a Team

After learning that the module project required a team of four to five members, I decided to ask familiar friends to partake in the project as a team. The following are members of my team in first name alphabetical order:

- Jackie Ju
- Joiedel Agustin
- Kiowa Daly
- Rebecca Lobo

On forming the team, we agreed upon different objectives to achieve our goal. These primary objectives based on a small discussion includes:

1. To hold a weekly meeting every Wednesday as regularly as possible.
2. A team framework to help process the development.
3. To come up with mobile application concepts.

The purpose of the three objectives outlined above is to have a steady workflow between meetings, acknowledge the work that will be and has been completed, and to keep track of the potential end application.

Week 2 (18th September 2019) – Formulating Concepts

In week 2, we were introduced to the project concept, sustainability. In understanding the project concept, the goal of the team was to discuss potential applications that were in line with sustainability. Before diving into the whole discussion, we decided as a team on two things - the project framework, and the team name.

As a member of our team was a twitch streamer, we decided to name the team as Twitch.tv/Joiztik. A Facebook group chat was also formed by Rebecca to increase our team communications.

Following on, the framework of choice was Agile (Figure 1.1) as everyone on the team was familiar with an Agile development. Along with that we decided to use a Kanban board as our Agile tool. Kiowa and Rebecca were familiar with the online application, Trello, to implement the Kanban board, however Jackie and myself were more familiar with GitHub Project Board as a Kanban tool. I provided positive opinions on utilizing GitHub Project Board as the project was planned to be developed under the version control of GitHub. Acknowledging this fact, I believed it was more relevant to use GitHub Project Board as the two tools come hand in hand, as well as being able to post issues from the version control to the Kanban board. In the end, we agreed on setting our Kanban board onto GitHub Project Board, and our first cards (Figure 1.2) implemented, contained concept ideas from various members.

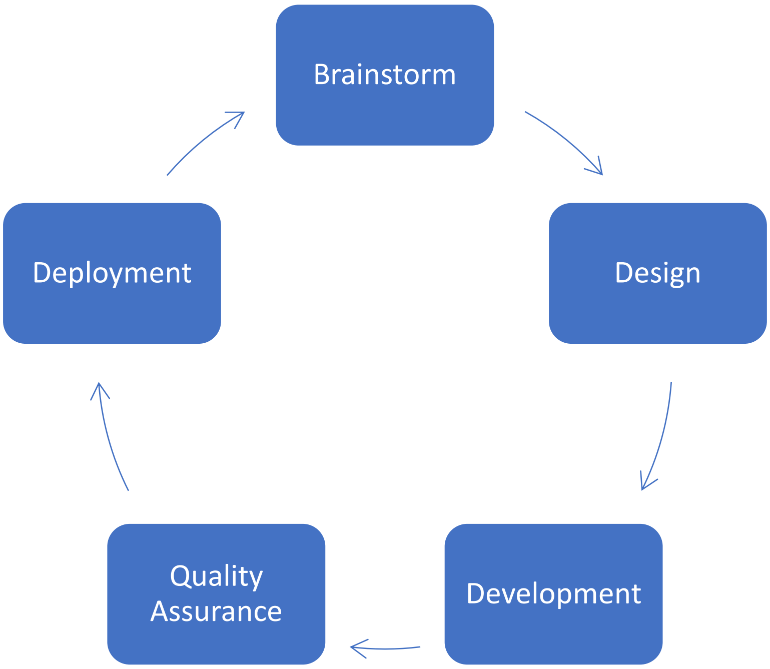
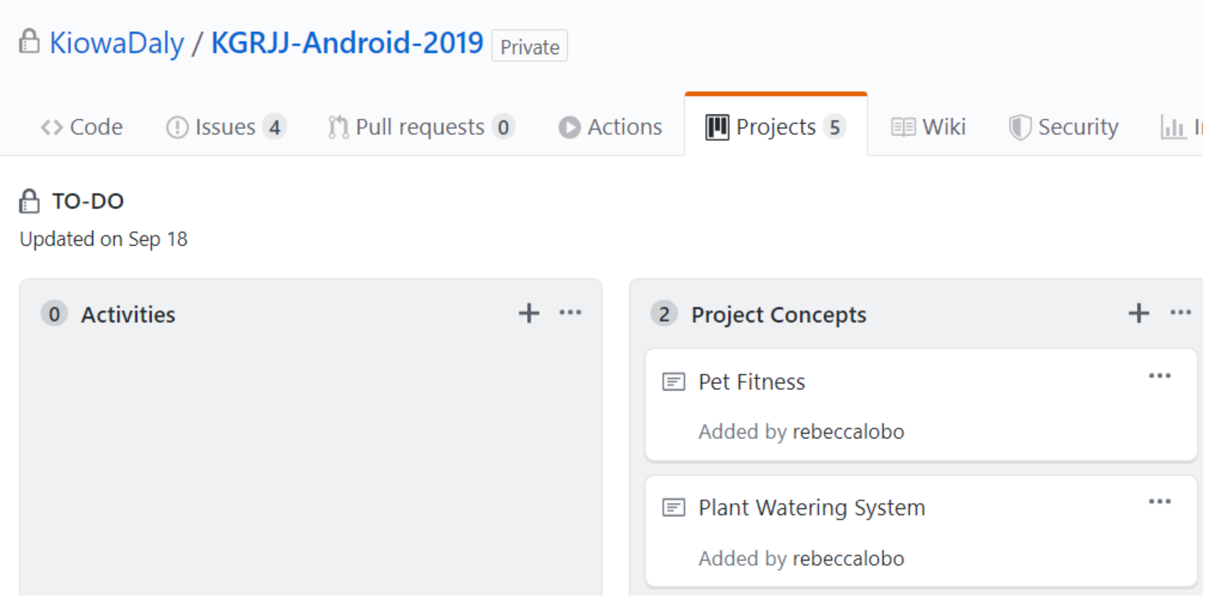


Figure 1.1 The Agile Model that our project management framework is based upon.



KiowaDaly / KGRJJ-Android-2019 Private

TO-DO

Updated on Sep 18

0 Activities

2 Project Concepts

Pet Fitness

Added by rebeccalobo

Plant Watering System

Added by rebeccalobo

Figure 1.2 First look into concept ideas formulated on GitHub Project Boards

We also decided to have Kiowa as the scrum master to hold our weekly sprints and to ensure everyone was on track with their task at hand.

After dealing with these components, we came back to formulating ideas for our mobile application. As the concepts were generated on the spot, the concepts provided were rather a simplistic side of sustainability. The mentioned application ideas included personal fitness application, pet fitness application and my personal idea was a plant well-being application. These formulated ideas were simply basic ideas which the underlying concept is to be expanded the following week.

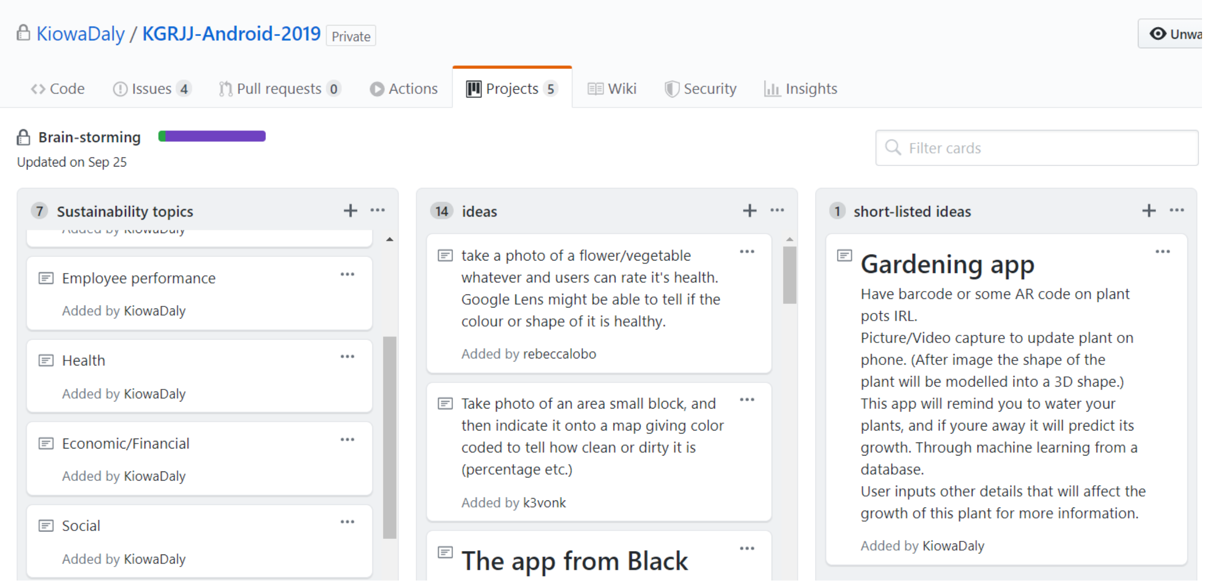
Week 3 (25th September 2019) – Brainstorm

For this week, the team met during the android tutorial slot. The team carried out "The 10 Plus 10 Method" following the guidelines of the lecture. For the two hours, we completed the various steps for this process in a trial sense. The steps included:

1. State your design challenge
2. Generate 10+ different design concepts that addresses that challenge
3. Reduce the number of design concepts OR repeat
4. Choose the most promising concept(s)
5. Produce 10 details / variations of that concept
6. Present your ideas to a group
7. As your ideas change, sketch them out

As the design challenge was based on sustainability, the team moved onto the second step which was to generate around 10 ideas. My personal ideas placed forward was as follows: gardening plant cycle application, fitness application using gyroscope to ensure users movement is correct and an environment application. Rebecca mentioned a crazy virus application idea that gains sustainability by infecting other phone users and spreading the virus.

After the 10 ideas generated, we short-listed the ideas. This included my personal gardening application idea shown on Figure 1.3. We decided to use the gardening application and come up with other design concepts with gardening (Step 3 and 4). I found an application called “iheurting” which had a similar gardening application aspect. Furthermore, I introduced some sensors that can be implemented for this application such as the thermometer sensor and the barometer sensor to check the pressure, humidity and temperature for the plant environment. Other concepts for this application was to create a 3D model of our flower into our application to mimic the growth life of the plant with suggested trimmings to keep the plant in a healthy state.



Brain-storming
Updated on Sep 25

7 Sustainability topics

Employee performance
Added by KiowaDaly

Health
Added by KiowaDaly

Economic/Financial
Added by KiowaDaly

Social
Added by KiowaDaly

14 ideas

take a photo of a flower/vegetable whatever and users can rate it's health. Google Lens might be able to tell if the colour or shape of it is healthy.
Added by rebeccalobo

Take photo of an area small block, and then indicate it onto a map giving color coded to tell how clean or dirty it is (percentage etc.)
Added by k3vonk

The app from Black

1 short-listed ideas

Gardening app
Have barcode or some AR code on plant pots IRL.
Picture/Video capture to update plant on phone. (After image the shape of the plant will be modelled into a 3D shape.)
This app will remind you to water your plants, and if youre away it will predict its growth. Through machine learning from a database.
User inputs other details that will affect the growth of this plant for more information.
Added by KiowaDaly

Figure 1.3. Shows GitHub Kanban board of the short-listed applications

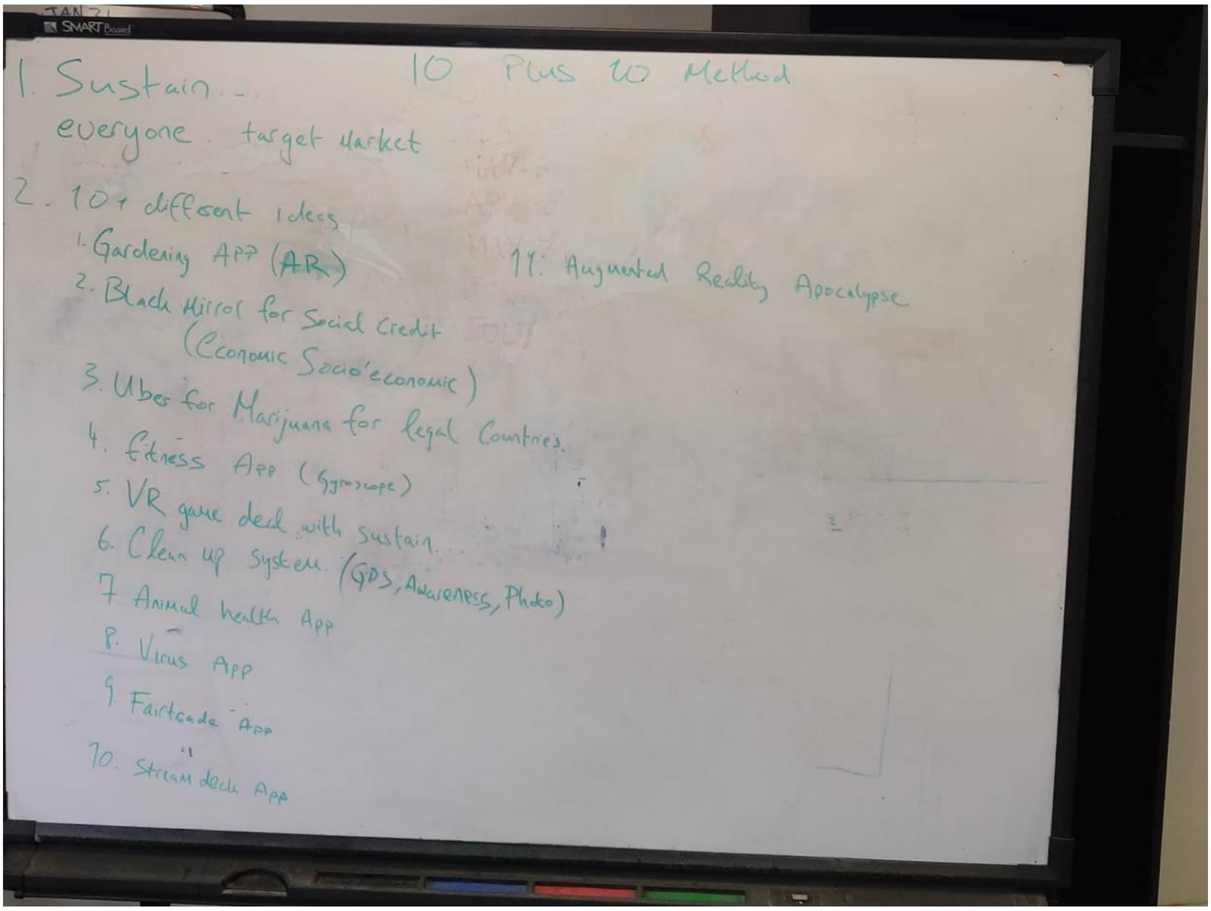
As the 2-hour tutorial ended before we could present the ideas to other groups, we decided to skip step 6 and focus on step 7 at home. In doing so, I searched up the term sustainability.

This reverted my thoughts to a more environmental thinking and thus came about an environmental application that displays the "cleanliness" of different areas. I held this idea in mind before suggesting it to the group the next week.

Week 4 (2nd October 2019) – White Boarding (10 Plus 10 Method)

On week 4, we decided to restart "The 10 plus 10 Method" process. This contained an additional methodology where we complete "The 10 plus 10 Method" through white boarding.

I took it upon myself to write the various steps onto the board while providing a talk on how the different concepts might be a solution to our project. The first step was to ensure that everyone knew that the group project was about sustainability. Next, I listed all the design ideas that were produced on Week 3 with some additional concepts created by my fellow team members on the day.



10 Plus 10 Method

1. Sustain..
everyone. target Market

2. 10+ different ideas
1. Gardening APP (AR)
2. Black Mirror for Social Credit
(Economic Socio'economic)
3. Uber for Marijuana for legal Countries.
4. Fitness App (Gyroscope)
5. VR game deal with Sustain...
6. Clean up System. (GPS, Awareness, Photo)
7. Animal health App
8. Virus App
9. Fairtrade App
10. Stream deck App
11. Augmented Reality Apocalypse

Figure 1.4 displays "The 10 plus 10 Method" steps 1 and 2.

As part of step 2, I introduced the "Clean Up System" application which involved promoting pollution awareness using image analysis. I described the different tools we could use to accomplish this goal such as the hardware GPS to track user's location and the user's camera to take a photo. The photo would then obtain the geo-location of where the user took the photo. These photos would then be placed on the map indicating the "cleanliness" of a specific area on the map.

This idea proved to be more sustainable in helping to promote environment awareness, as users would not enjoy seeing their hometown being labelled as “dirty” and would affect many people in the process. Whereas, the gardening application had a focus on users who wanted to plant and seek a fruitful growth of their plant. We believed that creating a 3D model of the plant would be a hard task to accomplish. Therefore, we scrapped the promising concept from Week 3 and promoted the "clean up system" as our main promising application.

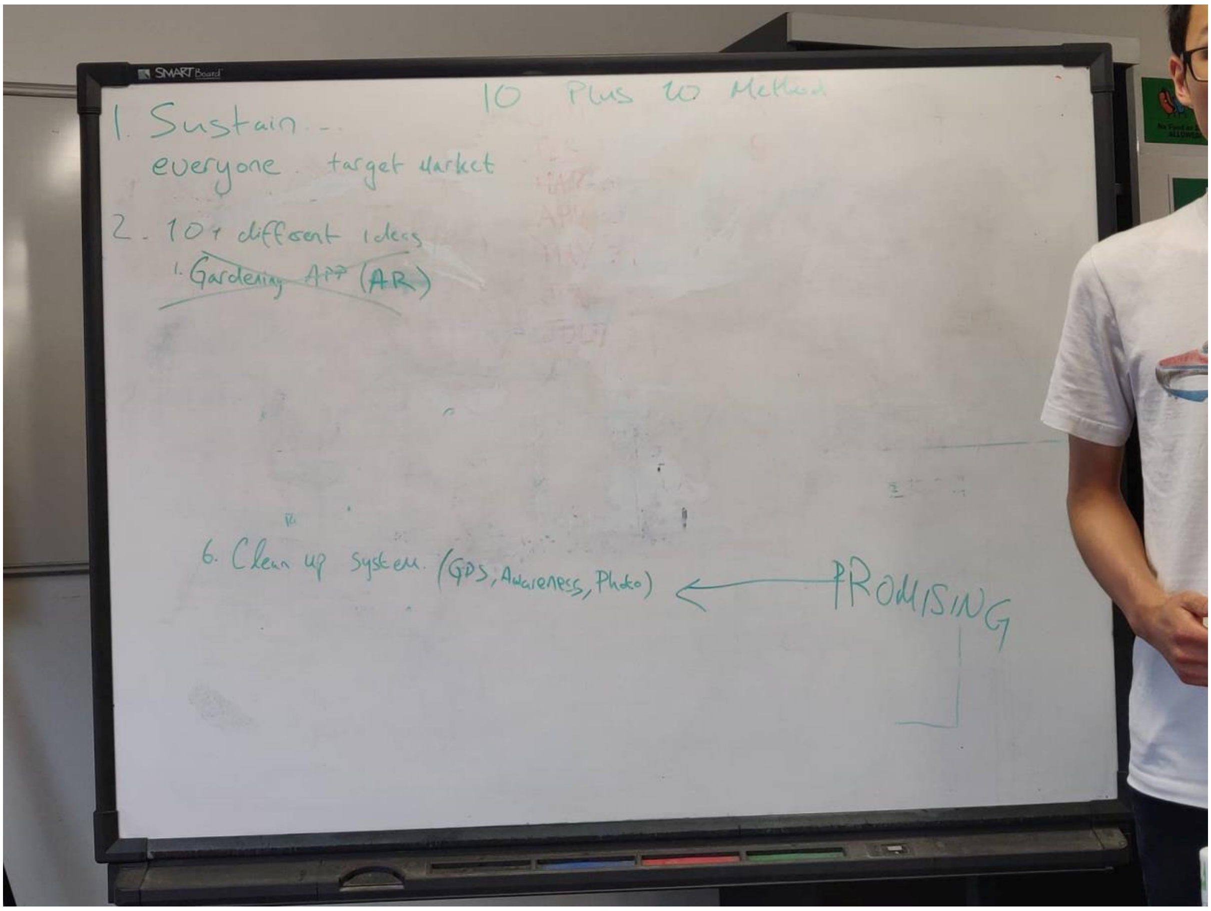


Figure 1.5 – Choosing a promising concept for “The 10 plus 10 Method”

The team agreed on the solution, so we began creating design features and variations of the project as shown on Figure 1.6. One of the main aspects of this application was to create a login system to store user images. As Kiowa had been building a login system since Week 2, we decided to give the task over to Kiowa. We decided on two other features on the day, such as the Map functionality, and a wheel toolbar menu. I suggested the different components to a Map feature. One such idea was a color-coded environment where green represented a clean environment, and red represented a polluted environment. Along with that, images should be displayed onto the Map as an image set, on a marker.

One other suggestion included the wheel toolbar. The toolbar would include buttons for other activities within the application, which wasn't discussed at this meeting. The meeting ended with a half-finished method development plan, as we did not have time to continue. Therefore, the next meeting involved a continuation of this methodology.

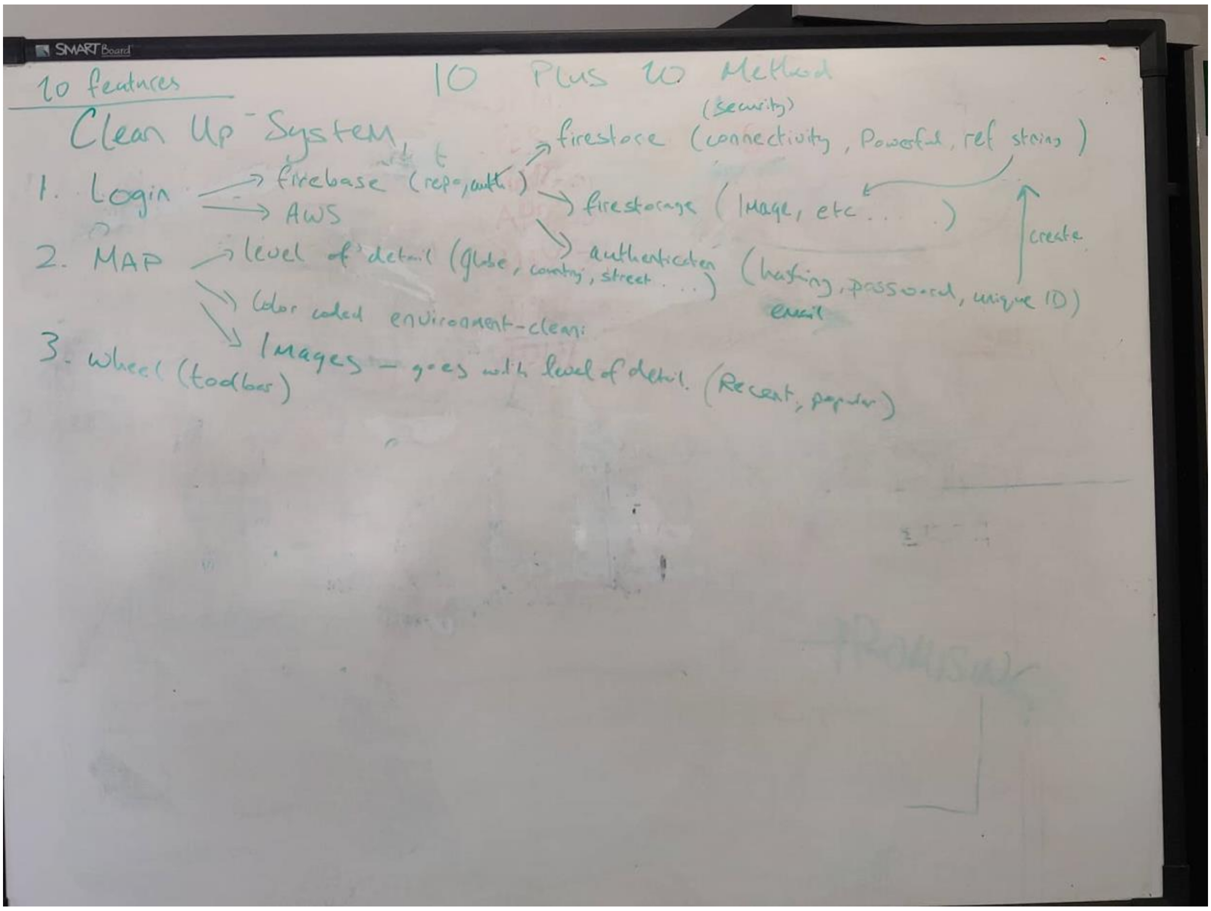

SMART Board
10 features
10 Plus 10 Method
Clean Up System
1. Login
firebase (repo, auth)
AWS
firestore (connectivity, Powerful, ref strings)
(security)
firestorage (Image, etc ...)
authentication (hashing, password, unique ID)
email
create
2. MAP
level of detail (globe, country, street ...)
Color coded environment-clean
Images - goes with level of detail (Recent, popular)
3. wheel (toolbar)

Figure 1.6 displays features of the “Clean Up System” application

Week 5 (7th & 9th of Oct 2019) – 10 Plus 10 Method, Task Management & Prototyping

This week we concluded “The 10 plus 10 Method” by outlining finishing features that will be included in the final product. Additional features such as a unique avatar to represent user location, a set of wheel buttons (home, profile, logout) to replicate activities that may be included into our final product (Figure 1.7). Again, I wrote on the board for this task.

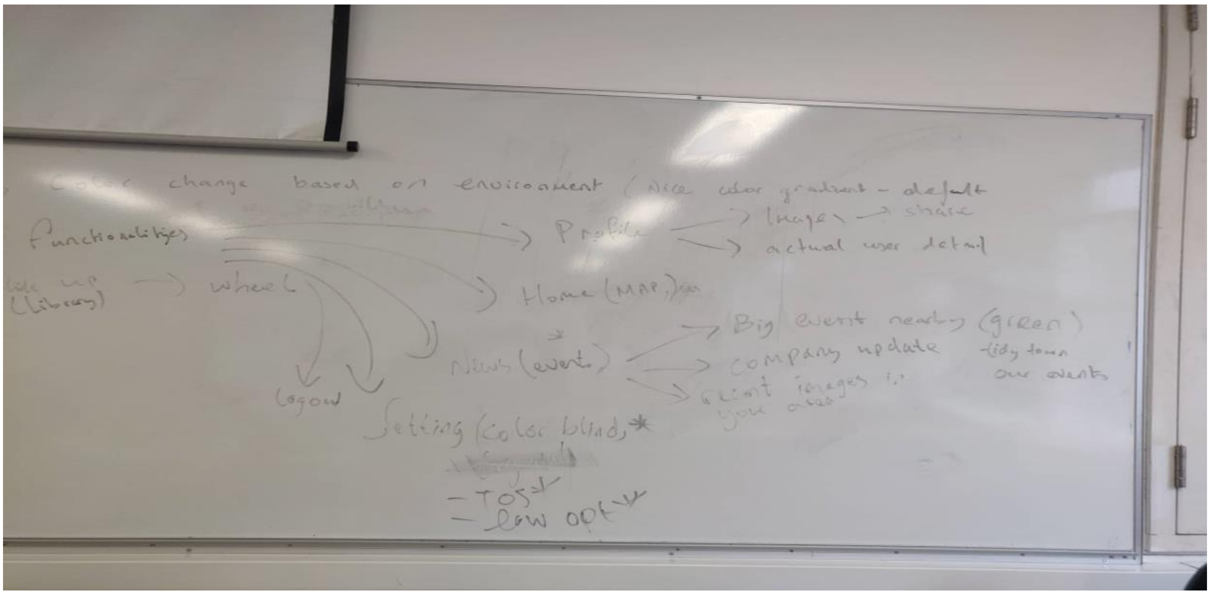
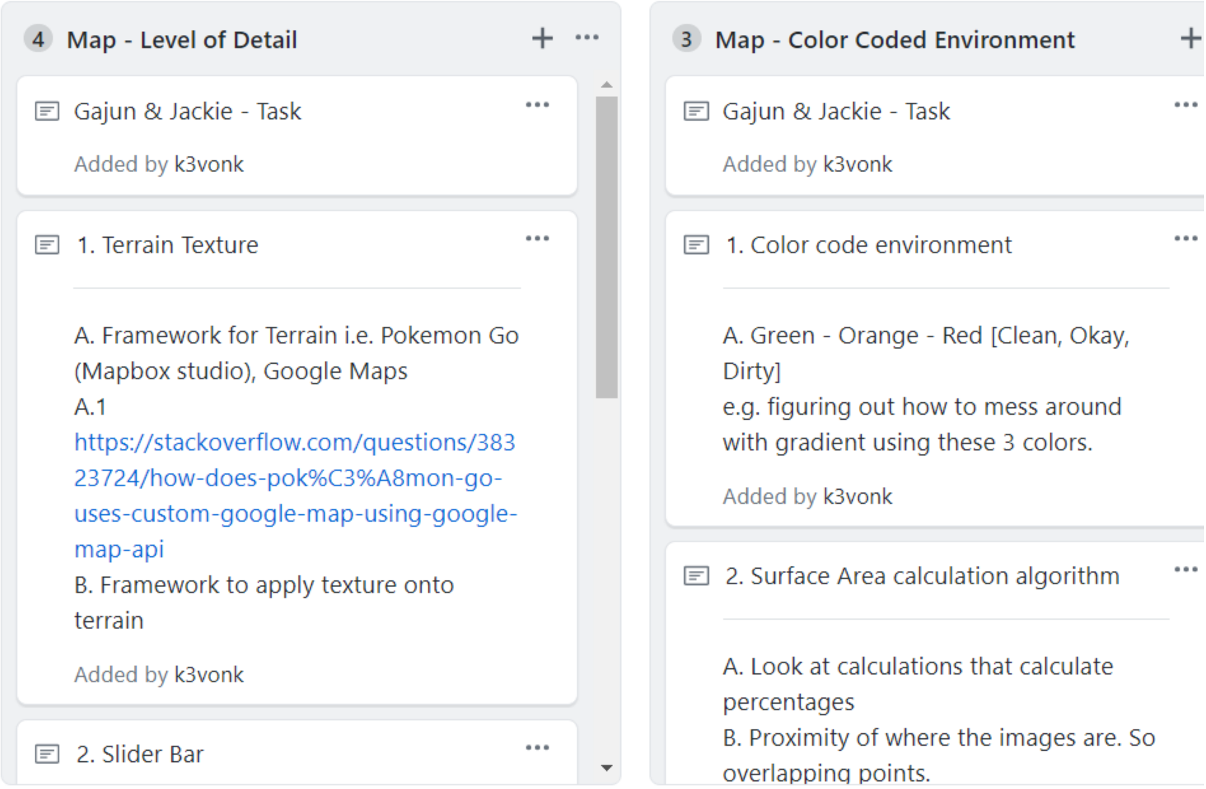


Figure 1.7 shows our finishing functionalities for the wheel menu.

Since the features of our project were now defined, it was appropriate to delegate tasks to each member of the team. The following tasks were delegated based on member knowledge of a task. This was subjected to change as schedule of members collided with productivity of the application.

1. Kiowa and Rebecca choose to complete the following:
   a. Image analysis
   b. Registration/Login
   c. User Experience and Interface
2. Jackie and I choose to complete the following (Figure 1.8):
   a. Google Maps API implementation
   b. Terrain texture
   c. Map colour-code mapping
3. Joiedel choose to complete
   a. News activity of wheel menu.



4 Map - Level of Detail

Gajun & Jackie - Task

Added by k3vonk

1. Terrain Texture

A. Framework for Terrain i.e. Pokemon Go (Mapbox studio), Google Maps
A.1
https://stackoverflow.com/questions/38323724/how-does-pok%C3%A8mon-go-uses-custom-google-map-using-google-map-api
B. Framework to apply texture onto terrain

Added by k3vonk

2. Slider Bar

3 Map - Color Coded Environment

Gajun & Jackie - Task

Added by k3vonk

1. Color code environment

A. Green - Orange - Red [Clean, Okay, Dirty]
e.g. figuring out how to mess around with gradient using these 3 colors.

Added by k3vonk

2. Surface Area calculation algorithm

A. Look at calculations that calculate percentages
B. Proximity of where the images are. So overlapping points.

Figure 1.8 Kanban cards created by myself (k3vonk GitHub username)

As the tasks were split up between sub-groups, we further broke down the components as shown in Figure 1.8.

Finally, we started working on the paper prototype for the project because the task was due on Week 6. The breakdown of tasks allowed our team to draw our prototypes. I took this opportunity to draw out the different map views along with their color-coded "cleanliness" classification. Figure 1.9 displays a map view on different levels with their colour-coded classification on the left, whereas the prototype on the right depicts an algorithm to achieve the colour simulation of a dirty environment bubble.

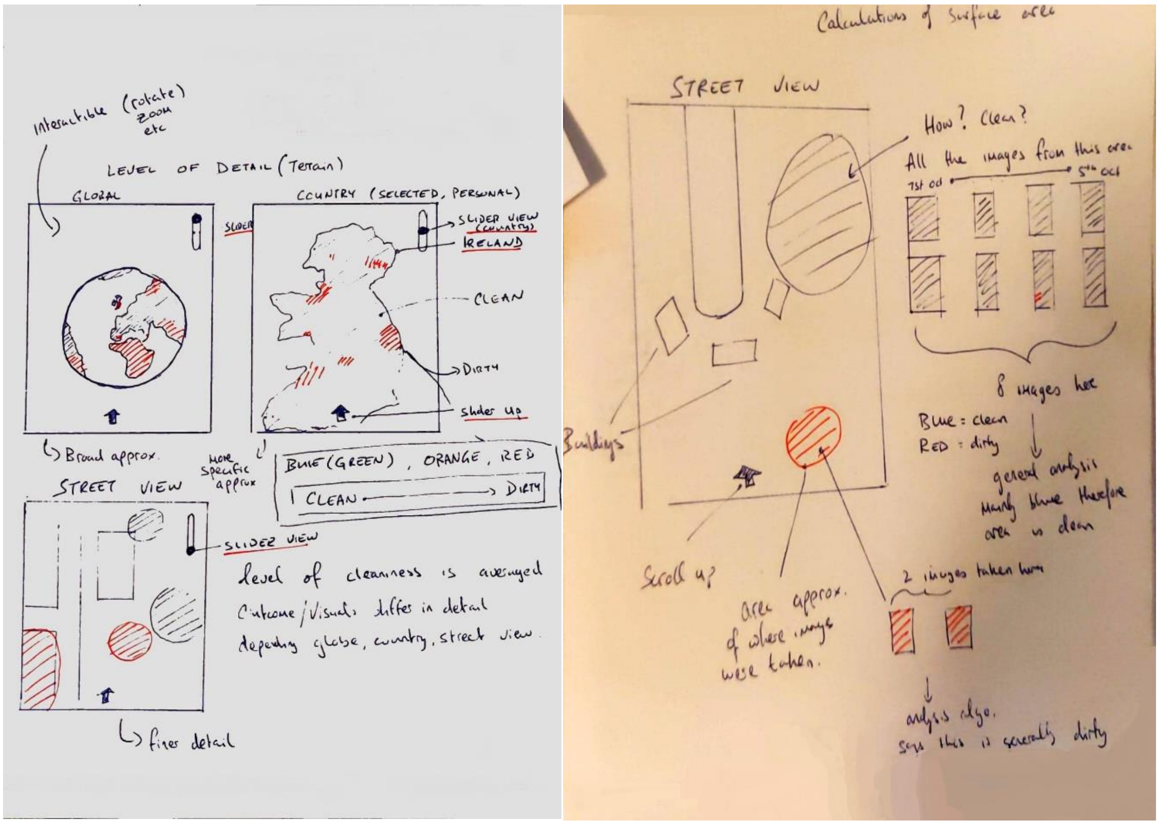

Interactible (rotate) zoom etc
LEVEL OF DETAIL (Terrain)
GLOBAL
COUNTRY (SELECTED, PERSONAL)
SLIDER
SLIDER VIEW (COUNTRY)
IRELAND
CLEAN
DIRTY
Slider up
Broad approx.
More specific approx
BLUE (GREEN), ORANGE, RED
CLEAN DIRTY
STREET VIEW
SLIDER VIEW
level of cleanness is averaged
Outcome / Visuals differ in detail
depending globe, country, street view.
finer detail
Calculations of surface area
STREET VIEW
How? Clean?
All the images from this area
1st oct
5th oct
8 images here
Blue = clean
RED = dirty
general analysis
Mainly blue therefore
area is clean
Buildings
Scroll up
area approx. of where images were taken.
2 images taken here
analysis algo.
says this is generally dirty

Figure 1.9 Paper prototype of color-coded maps based on the environment's cleanliness.

Week 6 (13th & 16th of October 2019) – Into the Application

To begin the week, the team agreed to have Rebecca to submit the paper prototype and that the app name to be EnviRevive which was a combination of words like environment and revive. This name was brought up by Kiowa, Rebecca and I, where environment is the premise of the application and we want to build awareness of pollution to the world such that it brings a revival to our environment.

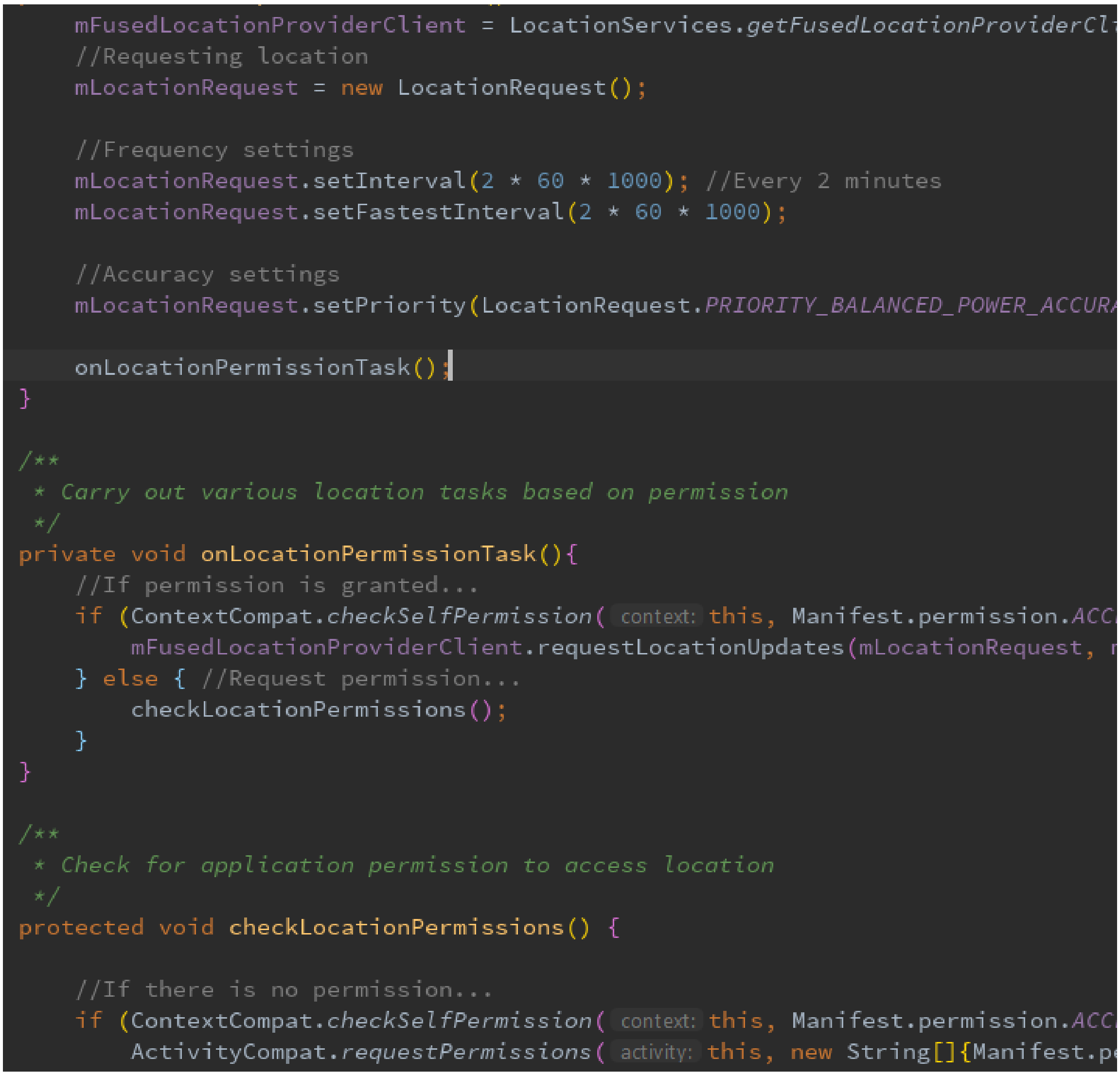
I began my task by researching about map APIs for the android system. A solution to this topic was to utilize Google Maps API. I decided to follow through with this API as it meant that I had to register for a Google Cloud Platform service (PaaS), this enabled potential Google APIs that could be implemented in the future. Along with that, as a first-time user GCP provides $300 worth of free credit to use their service, that meant that if the API would receive high traffic usage then it would consume the free credit first. Another note is that I was able to create an account for GCP without linking to a billing address.

I also figured out that Android Studio provided a default GoogleMaps Activity template, which I utilized to bring about the default features of GoogleMaps. As well as that, I used a numerous set of tutorials from all sort of sources, ranging from CodeLabs, StackOverflow, and GoogleMaps API documentation to build a GoogleMaps Activity using GoogleAPIClient.

To access GoogleMaps API, I had to create an API key from Google Cloud Platform. I followed the simple instructions provided by Google API documentations which included, enabling the GoogleMaps API, acquiring the package name of our project, and the SHA-1 certificate fingerprint of each of our member's Android Studio Software. The instructions permitted development with GoogleMaps API for each of our team members only. APK version of the map activity was installed onto our phone to check if Google Maps ran on the phone without SHA-1 certificate fingerprint, this was a success but the current solution only located users to a hard-coded location.

Week 7 (23rd of October 2019) – Continuation of Google Maps Activity

Week 7 began with further research onto the topic of Google Maps Activity. I realized that GoogleAPIClient was a deprecated class which was no longer used. A new class called FusedLocationProvider was required to track user's location, this meant that the whole Google Maps Activity had to be overhauled. Location permissions was required for the manifest, location call back functions had to be created and an update timer was in place to track user movements every two minutes.



```
    mFusedLocationProviderClient = LocationServices.getFusedLocationProviderCl
    //Requesting location
    mLocationRequest = new LocationRequest();

    //Frequency settings
    mLocationRequest.setInterval(2 * 60 * 1000); //Every 2 minutes
    mLocationRequest.setFastestInterval(2 * 60 * 1000);

    //Accuracy settings
    mLocationRequest.setPriority(LocationRequest.PRIORITY_BALANCED_POWER_ACCURA

    onLocationPermissionTask();
}

/**
 * Carry out various location tasks based on permission
 */
private void onLocationPermissionTask(){
    //If permission is granted...
    if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.ACC
        mFusedLocationProviderClient.requestLocationUpdates(mLocationRequest, r
    } else { //Request permission...
        checkLocationPermissions();
    }
}

/**
 * Check for application permission to access location
 */
protected void checkLocationPermissions() {

    //If there is no permission...
    if (ContextCompat.checkSelfPermission( context: this, Manifest.permission.ACC
        ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.pe
```

Figure 1.10 A Google Maps Activity code to demonstrate the various functionalities involved.

I ensured that proper commenting and indentation was used so my team members can understand my code (Figure 1.10). At the same time, I created a side project that mimicked the Map activity of EnviRevive, where I experimented with various functionalities of the Map such as camera animations which were planned for later implementation of the code.

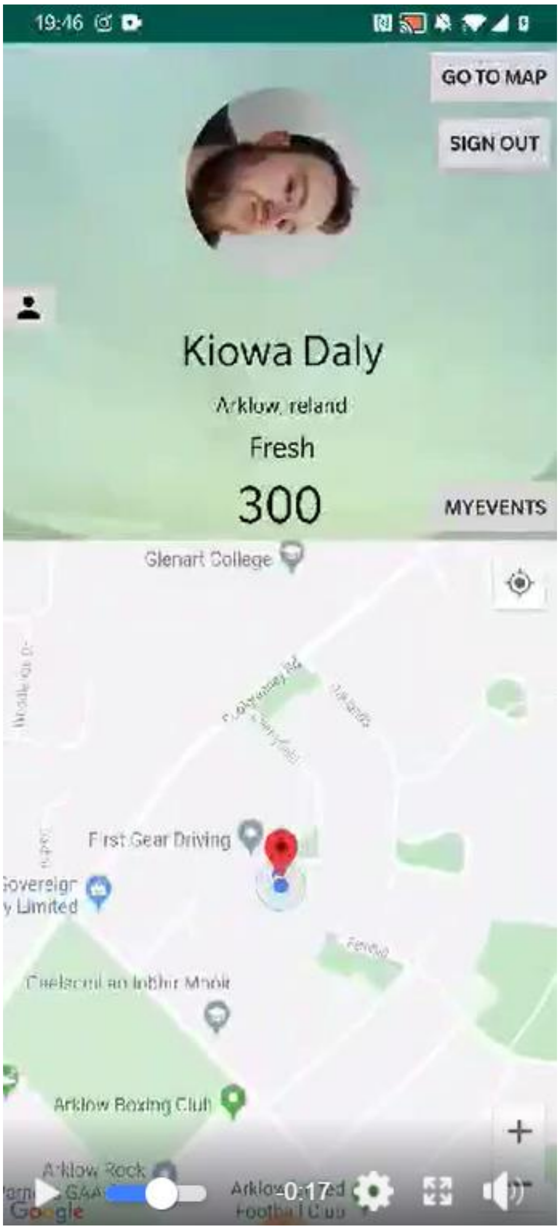
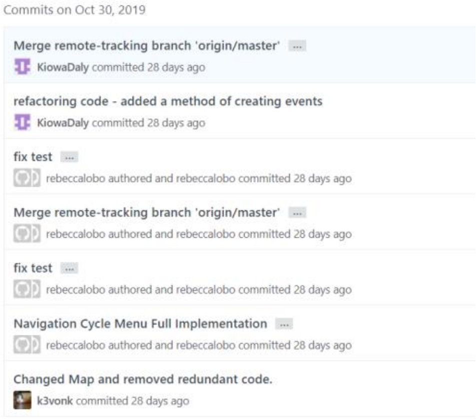


Figure 1.11 An image of the finalized Map activity used alongside Kiowa's Profile activity.

Week 8 (28th of October – 2nd November) – Beginning of Heatmap

To begin this week, I refactored the code to my Map activity and removed redundant code such as unused variables and experimented functions that were not useful anymore like some save state functions (Figure 1.12).



Commits on Oct 30, 2019

Merge remote-tracking branch 'origin/master' ...
KiowaDaly committed 28 days ago

refactoring code - added a method of creating events
KiowaDaly committed 28 days ago

fix test ...
rebeccalobo authored and rebeccalobo committed 28 days ago

Merge remote-tracking branch 'origin/master' ...
rebeccalobo authored and rebeccalobo committed 28 days ago

fix test ...
rebeccalobo authored and rebeccalobo committed 28 days ago

Navigation Cycle Menu Full Implementation ...
rebeccalobo authored and rebeccalobo committed 28 days ago

Changed Map and removed redundant code.
k3vonk committed 28 days ago

Figure 1.12 GitHub commits on Oct 30th.

Along with that, I discovered a good methodology to cluster images together that had similar classification while producing colour points on the map. This solution was based on SnapMap and Google's heatmap. SnapMap was a feature of Snapchat that displays the activities of "Live Stories" on an area, where red represented high activity and green represented low amount of activities. SnapMap also provided a feature where a user can touch the map to retrieve a recent "Live Story".

hern Territory
Queensland
Australia
South Australia
New South Wales
Sydney
Victoria
Melbourne
Tasmania
+
−

Google's heatmap provided a solution to represent features as a cluster. SnapMap uses the heatmap algorithm to calculate its activity density for an area. As a result, I decided to use Google's heatmap to calculate environment cleanliness which helps display the environment's tidiness through colours. I followed their documentation procedure on a test application. It required a JSON file of LatLng values which were the longitude and latitude of where an entity was located on the map. Using sample Google data of police station locations, I passed the list to the HeatmapTileProvider to generate a heatmap on my activity.

Figure 1.13 Heatmap of police station locations in Australia on Google Maps Activity.

Week 9 & 10 (4th – 17th of November) – Pause in Progress

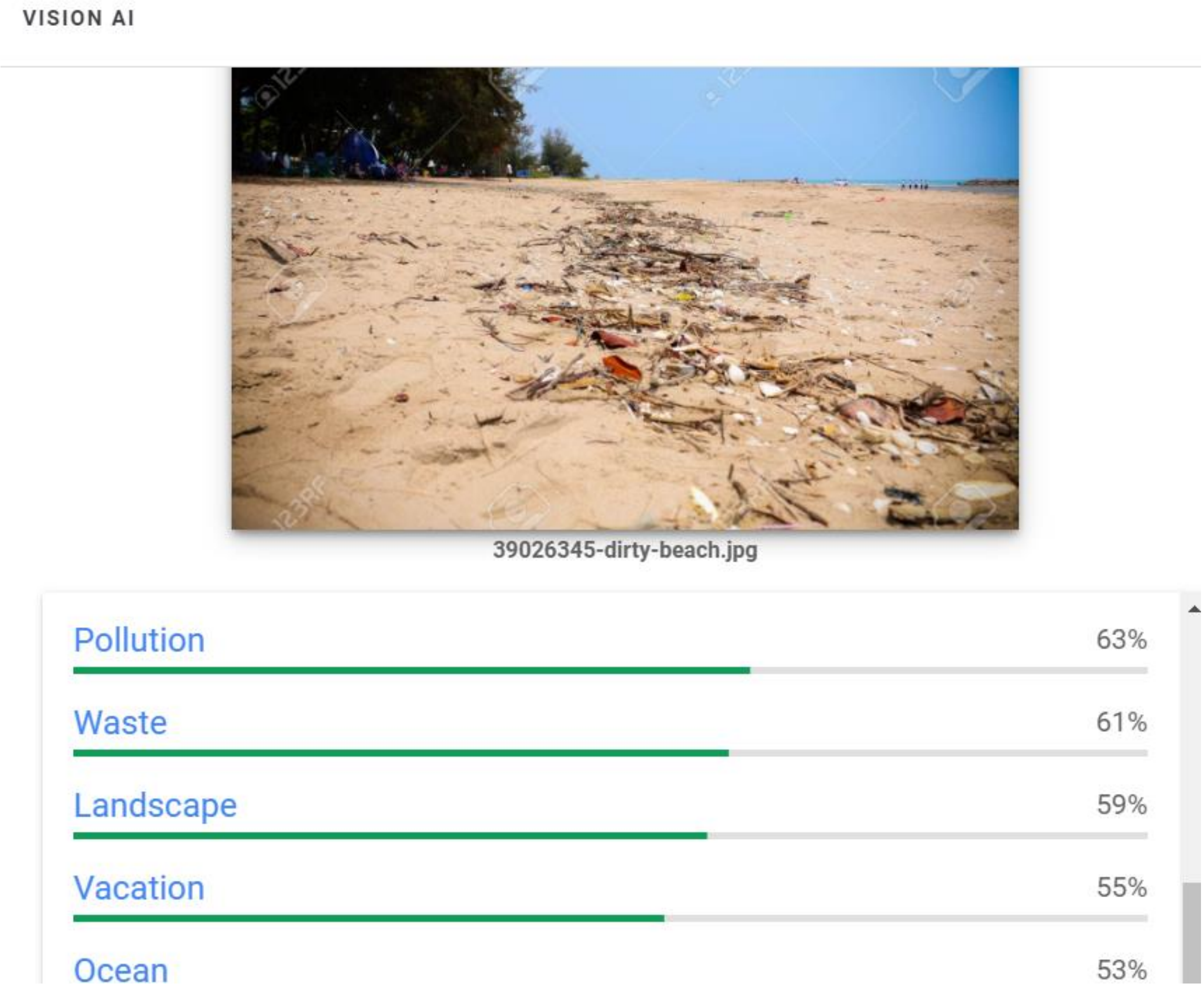
I had a pause on development beginning on week 9 that ended on week 10. The halt involved focusing on writing the Final Year Project – Interim Report for the end of week 10.

Week 11 (18th – 23rd of November) – Starting Image Analysis

In week 11, the remaining tasks revolved around the heatmap and the image analysis. As Kiowa and Rebecca were busy around their functionalities of the application and our other teammates required time for exam preparation. I was given the task of image analysis. Kiowa had prepared some background research into this topic and suggested using Vision API. Vision API is another set of tools owned by Google. I found out that Vision API provided two types of image analysis, AutoML Vision and Vision API. AutoML Vision provided a set of tools to train and customize learning models whereas Vision API provided a pre-built trained learning model. On Google's Vision API page, I was able to test a sample image of a beach covered with discarded rubbish. The outcome provided confidence percentages with all sorts of classification labels. I noticed in Figure 1.14 specific labels such as "Pollution", "Waste", and "Litter". These labels became our primary objective to search out when dealing with image analysis. That is the heatmap colour is based on the confidence values of pollution, waste, and litter. An absence of these labels indicated a clean environment for the geo-location of that image.

To understand how Vision API worked, I was required to look at Google's sample Vision API project as there wasn't clear documentation on implementing this API. I found out that I had to enable the Vision API on Google Cloud Platform to test their sample project, along with that I had to register a billing address to use this specific API.

I tried several iterations of integrating Vision API into the project, one solution involved implementing it on BaseActivity (The hub of where all our activities begin). After giving this solution a try, I ended up having an image pop-up onto the MapActivity screen.



Pollution	63%
Waste	61%
Landscape	59%
Vacation	55%
Ocean	53%

Figure 1.14 Vision API analysis of a dirt beach image.

However, this introduced a code smell to our BaseActivity identified as a God class that carried too much functions. As a result, I developed a new activity called ImageAnalysisScreen activity that mimiced how the sample project worked with a change to its output to producing a HashMap. In doing so, I made a new class alongside to create a GridView for the bottom half of the screen to try and replicate how Vision API's web page displayed their labels. This involved setting up a grid view adapter that took in the hashmap and parsed the labels from the hashmap to each cell on the grid view. Each grid view had a progress bar, a label, and a confidence percentage. However, there was no ordering to the hashmap data so it didn't present as well.

As a result, I took the opportunity to order the hashmap iteration based on their confidence values. This outputted a grid by grid of labels represented by confidence values in descending order. Next, I wanted to display the negative labels that would indicate a dirty environment. I extracted the labels "Pollution", "Waste", and "Litter" and colored their progress as red to have a negative connotation and placed them on the top of the grid view list. Figure 1.15 displays a sample result of my ImageAnalysisScreen, however for testing purpose I extracted a different label to test the red color. In Figure 1.15, I noticed that the image was rather blurry and had a discussion with Kiowa on fixing this problem. The solution was to take the picture from external storage, instead of directly from the camera as android produces a lower quality bitmap image.

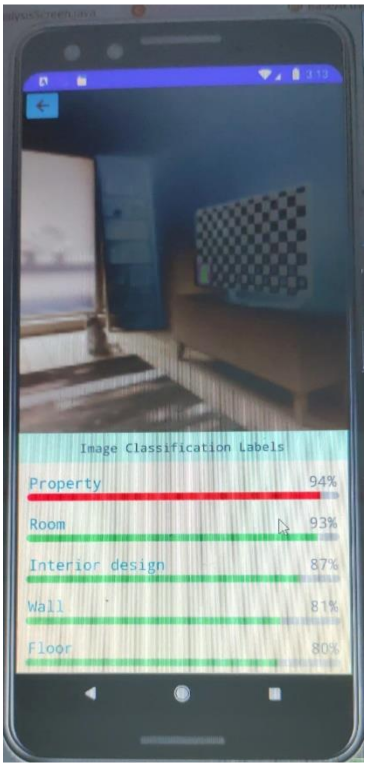


Image Classification Labels

Property 94%

Room 93%

Interior design 87%

Wall 81%

Floor 80%

Figure 1.15 A phone emulator displaying the ImageAnalysisScreen activity.

During this process, I shifted my heatmap task to Jackie. I gave him a brief run down on how the code functions. I suggested improvements such as using weighted heatmaps where our weights were based on the confidence values. It was also important to change the gradient colour scheme for non-dirty environment photos such that the interpolation of colours is confined to green. This source of knowledge included links to StackOverflow and Google's heatmap documentation.

Week 12 (25th – 29th of November) – Group Report (Final Entry)

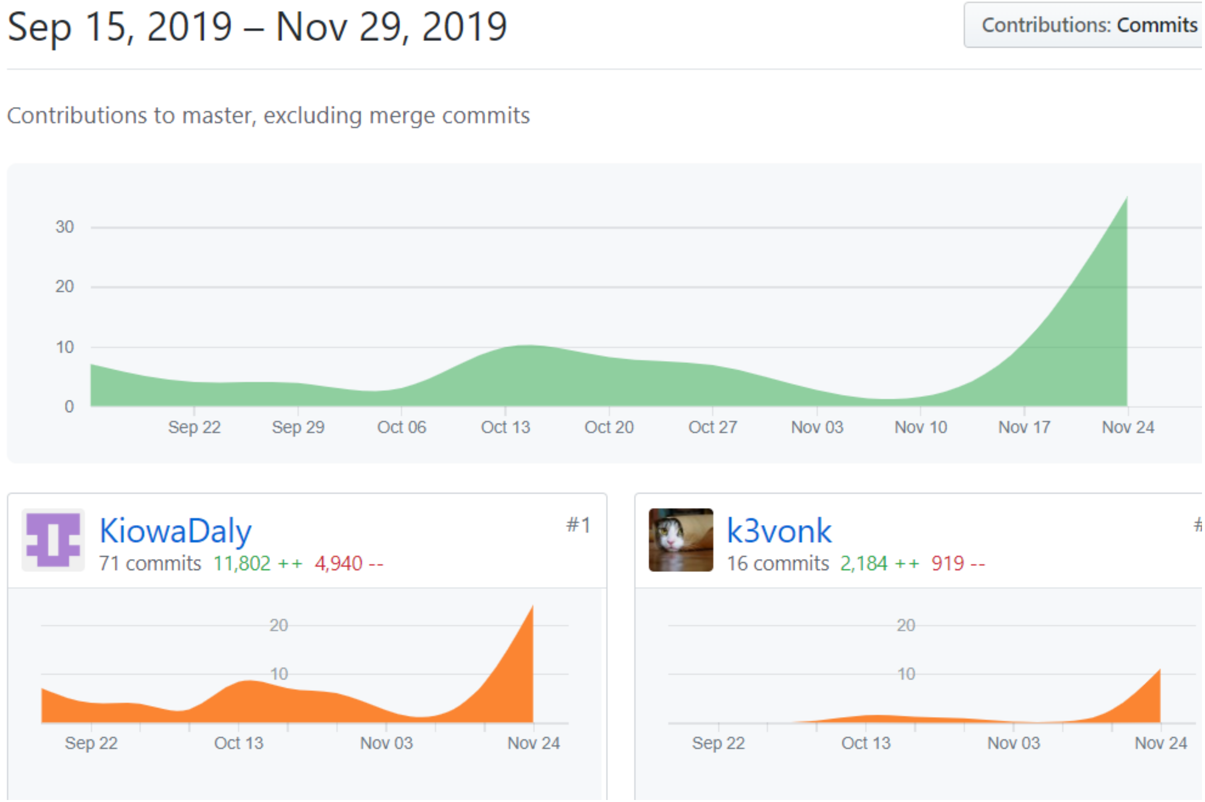
Week 12 began with a variety of testing ensuring that the image analysis tool worked and at an appropriate rate. Kiowa, Jackie and I came across a bug, where a fresh install of the application would reject the user from opening the camera. This was solved by finding out where methods were being called at the wrong time and thus, passing in null references to different objects.

I also had a conversation with Jackie to understand the situation of the HeatMap implementation. Jackie described the process as a difficult implementation and therefore, we looked at other solutions. A concluding solution to the heatmap was to have two heatmap providers, where a button would switch between the two heatmaps. This would allow users to check out the clean areas on the map and the dirty areas on the map.

After my personal testing phase, I began writing up the group report. I was in charge of writing the report in latex as I had experience in using latex from a previous project. As a result, I ended up writing the abstract, introduction, and a piece on Cloud Vision AI for the literature review. I also compiled everyone else's work onto the report, ensuring everything was referenced appropriately. Afterwards, I distributed the pdf version of the report to everyone to submit later at their own convenience.

The group also ensured that contribution is discussed amongst the team and therefore, contribution would not be uneven between the different members. That is we recognized each members work and fairly distribute the percentage amongst each other.

Contribution



Sep 15, 2019 – Nov 29, 2019

Contributions: Commits

Contributions to master, excluding merge commits

KiowaDaly #1
71 commits 11,802 ++ 4,940 --

k3vonk #
16 commits 2,184 ++ 919 --

Figure 1.16 displays the total amount of commits I added to EnviRevive on GitHub.

Contribution (Total for the whole team 100%)

As we had 5 members on the team, a maximum amount of contribution would result to 20% per member.

	Contribution
Project development (Code)	20% [Google Map, Cloud Vision AI, Heatmap]
Group report	20% [Abstract, Introduction, Lit. review, similar applications]

Table 1. A contribution table representing personal achievements in the project.