

TASWEER

A Photo-Sharing Social Media Platform

Stage 3 Report

Group 1

Abis Ali, Ammad Ahmed, Ayushi Amin,
Fahim Abdul Gani, Fayyaz Ameen,
Mayar Ahmed, Wesley Barreto

Contents

1. Introduction	3
2. Final Application Design and Implementation	7
3. Implementation Methodology	Error! Bookmark not defined.
4. Project Evaluation	68
5. Usability	100
6. Appendix	134

Introduction

Table of Content

1. Introduction	5
1.1 Overview	5
1.2 Purpose	6
1.3 Product Scope	6

1. Introduction

1.1 Overview

This document describes our progress on the completion stage of the application, specifying the system design in terms of the environments used for its development. It provides a high level overview of the application and technology used at the backend along with the database schema and the functionality of the system.

The reader is presented with the design choices of the User Interface by giving screenshots of the completed application with a brief explanation on every major screen. Changes in implementation methods used in the development of the project along with the duration of time taken to complete specific tasks is listed.

Further, the reader is presented with a detailed explanation of the various functional requirements achieved at this stage. A visual representation of one of the functions of our application is also presented using a sequence diagram.

The reader is then provided with the tasks completed and the functionality implemented, giving an explanation on how some of the system functions were tested for technical correctness. A user guide containing installation instructions for a user can be found along with instructions to download and run the app code for developers.

Future changes based on usability testing data gathered using a survey are also mentioned which could be implemented for newer versions of the application.

1.2 Purpose

The purpose of this report is to give a final detailed progression of the development of the multi-platform photo sharing application “Tasweer” by our company ZenTech. The report provides the reader with a reflection of the state of the application in its final stage, along with this, an overview of the requirements that have been achieved up to date is included as well.

High-level details about the various technologies, tools, languages etc. used while developing the application is provided.

This document is intended to be read by:

1. Rick Freedman, our client, and members that represent the client, to be able to examine the stage at which the application is on.
2. Dr. Talal Shaikh, the line manager of our team, who is to provide feedback for us.
3. The developers who are taking part in this application.

1.3 Product Scope

This project aims to develop a photo-sharing application that allows users to post pictures, like, comment, follow/unfollow other users etc. that is different from other photo-sharing apps present in the market. What makes this special from other apps is that this provides a gamified experience for other users, this app also does not allow users to post pictures that consist of any humans.

The targeted audience of this application are users who want to use a photo-sharing application with a gamified experience and who are 18+.

Final Application Design and Implementation

Table of Content

1. High-Level Overview of Technologies and Components	10
1.1 Google Cloud	10
1.2 Firebase	12
1.3 Realtime Database (Firebase)	12
1.4 Storage (Firebase)	14
1.5 Google Authentication (OAuth) (Firebase)	14
1.6 Google Analytics (Firebase)	15
1.7 Password Hashing (scrypt)	15
1.8 Machine Learning	16
1.8.1 Identifying Humans on images	16
1.8.2 Machine Learning Model	16
1.9 Firebase Firestore	17
1.10 Firestore Security Rules	18
1.11 Firebase Cloud Storage	18
1.12 Web Hosting	19
1.13 Google Maps API	19
1.14 Dependencies	20
2. Final System Design	22
2.1 Use Case Diagrams	22
2.2 Information Architecture Diagram	25
2.3 Sequence Diagram	27
2.4 Model View Controller	28
2.5 Database Design	29
3. Interface Design	30
3.1 Sign In	30
3.2 Home Screen	32
3.3 Upload Picture Screen	34
3.4 Map View	36
3.5 Profile Screen	38
3.6 Explore Screen	40
3.7 Communities Screen	41
3.8 Comments Section	43
3.9 Game Screen	44
3.10 Help Page	45
3.11 Photographer Profile	46

3.12 Gamer Profile	47
3.13 Awards	48
3.14 Landing Page	49
4. Implementation Methodology	49
Stage 1	51
Stage 2	52
Stage 3	53
5. Tests for Technical Correctness	54
5.1 Unit testing	54
5.2 End to End testing	54
5.3 Firebase Testing	55
5.4 Map view testing	56
5.5 Posting a Picture	57
5.6 Posting Comments	58
6. User Guide	59
6.1 Install the Application	59
6.2 Create an Account	59
6.3 Sign In	60
6.4 Follow User to Populate Home Page	61
6.5 Delete or Report a Post	63
6.6 User Chooses Picture to Upload	64
6.7 User Adds a Filter and Posts Picture	65
6.8 Access Your Own Profile	66
7. Installation Guide	67

1. High-Level Overview of Technologies and Components

1.1 Google Cloud

Cloud computing has had a tremendous influence on creativity and the digital ecosystem. It offers forward-looking companies the ability not only to increase flexibility, minimize costs and focus on key competencies, but also to completely reinvent how they function. For starters, by revamping internal workflows or consumer encounters as interactive interfaces that span from the data center to portable apps.

Google Cloud Platform (GCP), provided by Google, is a collection of cloud computing applications that operate on the same technology that Google uses for its own products. In addition to a collection of software solutions, it offers a variety of flexible cloud technologies, including computation, data storage, data mining and deep learning. Google Cloud Technology offers infrastructure as a business, a service platform, and applications as a service. (Google Cloud, 2021)

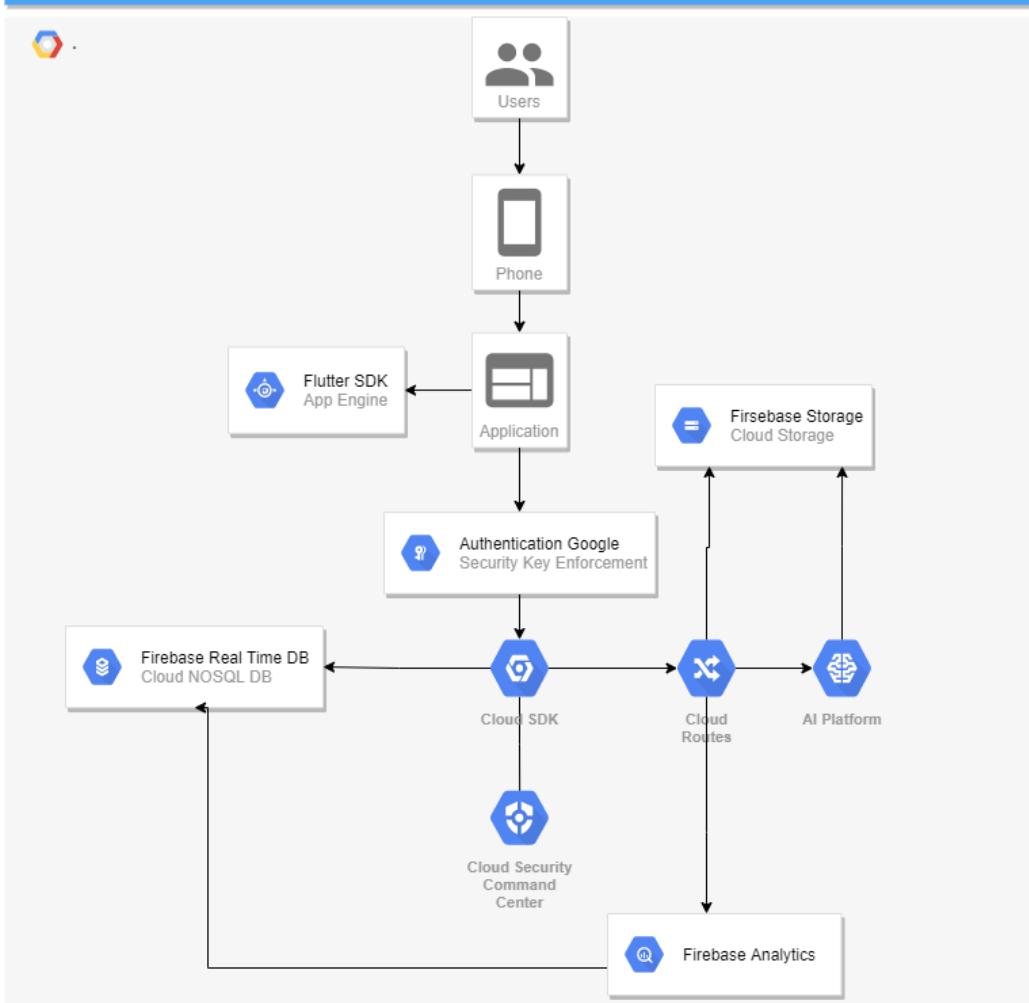


Figure 1: Package Diagram for Google Cloud Platform

1.2 Firebase

Firebase is a suite of tools for creating, developing, and building an app. This covers things such as analytics, authentication, databases, setup, file storage, push messaging, and the list goes on. The resources are being hosted in the cloud since a cloud platform is reliable, scalable and secure as well. Firebase's client SDKs connect directly with these backend systems without the need to set up any interfaces between your software and the infrastructure. Thus, unless you are using one of the Firebase database solutions, you normally write a database query function in your client software. (Edpresso, n.d.)

Few features of Firebase include:

1. Real Time database
2. Storage
3. Website hosting
4. Functions
5. Machine Learning
6. Analytics
7. Authentication
8. Crash and Debugging features

1.3 Realtime Database (Firebase)

The Realtime Database is a NoSQL database which, as such, has different configuration and features opposed to the relational database. The Realtime Database API is developed to give very fast-running processes. This helps to create a better real-time experience that will support millions of consumers without sacrificing responsiveness. Also, The Firebase Realtime Platform is a cloud-based database. Data is saved as JSON and linked to any linked device in real time. When designers create cross-platform applications for their iOS, Android and web apps, every one of their users share a single Realtime Database instance and instantly receive updates of the current data.

How NoSQL differs from MySQL:

1. NoSQL databases are non-relational.
2. NoSQL databases have dynamic schemas for unstructured data.
3. NoSQL databases are horizontally scalable.
4. NoSQL databases are document, key-value, graph or wide-column stores.
5. SQL databases are better for multi-row transactions whereas NoSQL ones are better for unstructured data like documents or JSON.

NoSQL databases have dynamic schemas for unstructured data, and data is stored in many ways. They can be column-oriented or as Key Value store. Which again gives us more benefits such as:

1. Creating documents without having to first define their structure.
2. The syntax is less strict.
3. Fields can be added on the go. (MongoDB, n.d.)

SQL databases are vertically scalable, this implies that they can expand the workload on a single server by increasing factors like CPU and RAM. While NoSQL databases are horizontally scalable. This ensures that you can manage additional traffic by embedding or connecting more servers to your NoSQL database. Lastly, NoSQL databases are the favored option for large data sets, which are constantly being changed.

Furthermore, The Realtime Database offers a versatile, expression-based rules language, dubbed Firebase Realtime Database Protection Rules which establishes how the data can be organized and what should be read from or written to the database. (Techotopia. 2017) (MongoDB, n.d.)

1.4 Storage (Firebase)

Firebase Cloud Storage is an efficient, quick, economical and Google-built object storage service. Cloud Storage Firebase SDKs apply Google encryption to your Firebase software file uploads and updates, independent of the performance of the network. Users can store images, audios, videos or other material that they create with our SDKs. Google Cloud Storage can be used on the server to access the same files.

Developers also use Cloud Storage Firebase SDKs to import and update files directly from clients. When the network connectivity is weak, the client will be able to retrieve the activity from where it left off, saving the users time and bandwidth. Cloud Storage stores their files in a Google Cloud Storage bucket, making them available from both Firebase and Google Cloud. This gives users versatility in uploading and downloading files from mobile clients through Firebase SDK. (Bohbot, 2021)

1.5 Google Authentication (OAuth) (Firebase)

Open authorization is a form of token-based identification that enables companies to exchange data through third-party platforms without revealing their usernames and passwords. Basically, OAuth is an intermediate who supplies third-party providers with a token that enables only unique account information to be exchanged. OAuth is an open-standard authentication protocol that defines how different applications and providers can securely allow authenticated access to their properties without necessarily exchanging the original, linked, single-login credential.

Firebase Authentication offers back-end infrastructure, such as simple SDKs, and fully prepared User interface libraries to authenticate users to your software. It facilitates authentication using passwords, phone numbers, common platforms such as Google and Twitter.

When a user signs in, we initially receive authentication codes from the user. These may be the e-mail address and password of the customer or the OAuth token of the Authentication Server. Then we transfer these details to the Firebase Authentication

SDK. Google services will then validate these credentials and return a reply to the client.

During an effective authentication, we may view the user's specific profile details and monitor the user's access to the data contained in other Firebase items. We may also use the given authentication token to validate the identity of users in your backend services.

Authenticated users can read and write data to the Firebase Realtime Database and the Cloud Storage by default. But we can monitor the access of these users by changing your Firebase Realtime Database. (Raible, 2017)

1.6 Google Analytics (Firebase)

The Data is collected from the app though the SDK, which is then accessible via a Firebase console in a dashboard. This dashboard offers comprehensive perspectives on the data such as briefing statistics like active users and demographic information to more detailed information like identification of the most frequently bought products. Google Analytics, a free predictive solution, is at the core of Firebase. Analytics incorporates all Firebase functionality and helps you to log unlimitedly on up to 500 different incidents that you can identify using the Firebase SDK. Analytics notifications help you understand the actions of your customers so that educated choices about software promotions and efficiency optimizations can be made. (Su, 2017)

1.7 Password Hashing (scrypt)

A hash function is a mathematical function, where you input any kind of digital data, such as strings, files, compressed folders and so on. The function then processes that data and outputs a hash value.

Scrypt is a delayed key derivation function that is a special kind of hash function intended to be very effective at generating hidden cryptographic keys. That is, the

intention of the Scrypt hash is to take any input data, and to construct a fingerprint of that data, but to do it very slowly. The typical use-case is to take a password and generate a private n-bit key that is much longer and more stable.

The algorithm includes the following parameters:

- Passphrase - The string of characters to be hashed.
- Salt - A string of characters that modifies the hash to protect against Rainbow table attacks.
- N - CPU/memory cost parameter. (Greenberg, 2016)

1.8 Machine Learning

1.8.1 Identifying Humans on images

One of the core functionalities of our apps is to ensure and protect the privacy of people whose images get uploaded on our social platform. A machine learning model being one of the best approaches that we could consider for this feature.

We decided to detect humans by using face recognition Since a machine learning model can perform real-time face detection. It is used for video games and various other applications.

1.8.2 Machine Learning Model

Some of the important libraries used are:

1. Tensorflow
2. OpenCV
3. Scikit-learn

1) Tensorflow: TensorFlow is an open source framework for machine learning that is end-to-end. It has a robust, scalable ecosystem of tools, libraries and community

resources that enables researchers to quickly create and deploy ML driven applications to push the state-of-the-art in ML and developers. We are using TF to recognize the faces on the images

2) OpenCV: OpenCV (Open Source Computer Vision Library) is a software library for open source computer vision and machine learning. It comprises a detailed collection of algorithms for computer vision and machine learning, both classic and state-of-the-art.

3) Scikit-learn: Scikit-learn (Sklearn) is Python's most useful and stable machine learning library. It offers a range of powerful machine learning and statistical modeling methods, including classification, regression, clustering and reduction of dimensionality via a Python consistency interface. This library is based on NumPy, SciPy and Matplotlib, and is largely written in Python.

Incorporating these Libraries from the Machine learning Kit of firebase we were able to use facial feature, tones, contour as a guidance for the model we built to recognize human faces and recognize the face

The Image selected by the user before getting posted parses through the machine learning model locally on the device, which a prime feature on the app as this allows the user to be offline and still que their image for getting posted

Using a very similar technology we are scoping on integrating it with our maps, by training our model to recognize famous landmarks and be able to geolocate it while storing the image on our database.

1.9 Firebase Firestore

In cloud firestore we have 4 different collections, namely chatroom, posts, reportedPosts and users. Within the users collection we have the UIDs of each user which is again stored in a hierarchy of collection, and within each of these said collections, it has multiple fields. This format of noSQL db allows easier scaling and is extremely flexible. It is also faster for development and deployment.

```
service cloud.firestore {
  match /databases/{database}/documents {
    match /<some_path>/ {
      allow read, write: if <some_condition>;
    }
  }
}
```

1.10 Firestore Security Rules

With Cloud Firestore Security Rules, we can focus on building a great user experience without having to manage infrastructure or write server-side authentication and authorization code. All Cloud Firestore Security Rules consist of match statements, which identify documents in our database, and allow expressions, which control access to those documents:

Every database request from a Cloud Firestore mobile/web client library is evaluated against our security rules before reading or writing any data. If the rules deny access to any of the specified document paths, the entire request fails.

1.11 Firebase Cloud Storage

Cloud Storage for Firebase is a powerful, simple, and cost-effective object storage service built for Google scale, hence we thought that would be a very viable resource. The Firebase SDKs for Cloud Storage add Google security to file uploads and downloads for your Firebase apps, regardless of network quality. We used the SDK's to store images and other user-generated content. On the server, we can use Google Cloud Storage API's to access the same files.

We used the Firebase SDKs for Cloud Storage to upload and download files directly from clients. If the network connection is poor, the client is able to retry the operation right where it left off, saving your users time and bandwidth.

1.12 Web Hosting

We decided to pivot from our initial plan of web hosting with firebase and decided to use Heroku and Netlify because Heroku is extremely user-friendly, and has a great community to help with issues and doubts. As Heroku is a container-based tool in the cloud, it has the support of many programming languages, a great variety of add-ons, and deployment provisioning are provided by the tool. This means we don't have to worry about the infrastructure details such as the OS versions or what libraries to install.

1.13 Google Maps API

The Google Maps API dependency is easily compatible with Flutter having a multitude of features. Updates from Google for the API are very regular and new features are added with each update easing the implementation of complex functionality in the project. Although a lot of documentation does not exist or is very hard to find for some of these features, we chose to use this API to follow the standard maintained by reputed organisations.

1.14 Dependencies

- `firebase_database`: Dependency used to implement chat functionality using the Realtime Database provided by Firebase.
- `google_sign_in`: Used to allow the user to sign in using their Google email and password (Google sign in API).
- `cloud_firestore`: Connecting and querying to and from the Firestore database is done using this dependency.
- `firebase_core`: Main dependency that allows the project to use Firebase to store all app data that is queried.
- `firebase_storage`: Dependency connects the app to the backend to store the images uploaded by the user directly into Firebase storage.
- `photofilters`: Generates a list of filters that can be added to a picture when a user is uploading one to be posted.
- `image_picker`: After the users' choice is selected, this dependency allows the user to upload an image from either their camera or directly from their phone's gallery.
- `image_cropper`: This dependency allows the user to crop, rotate and scale the image they choose to upload.
- `google_maps`: Google Maps API dependency used to get map functionality on different pages of the app using the uniquely defined API key for the project.
- `geolocator`: Getting the current or selected location of the user after permission to get location is granted by the user.

- `uuid`: This gives a post a unique id, making each one unique when storing it into the database.
- `clippy_flutter`: This Flutter package helps draw shapes onto the screen based on custom screen/widget coordinates. It is used in the project to create a triangular pointer for the marker on the map displaying an image at that location.
- `flutter_staggered_grid_view`: Grid builder used to generate grids of images.
- `provider`: Used to set up a Provider object that receives data to populate the info window of a map marker with its respective image.

2. Final System Design

2.1 Use Case Diagrams

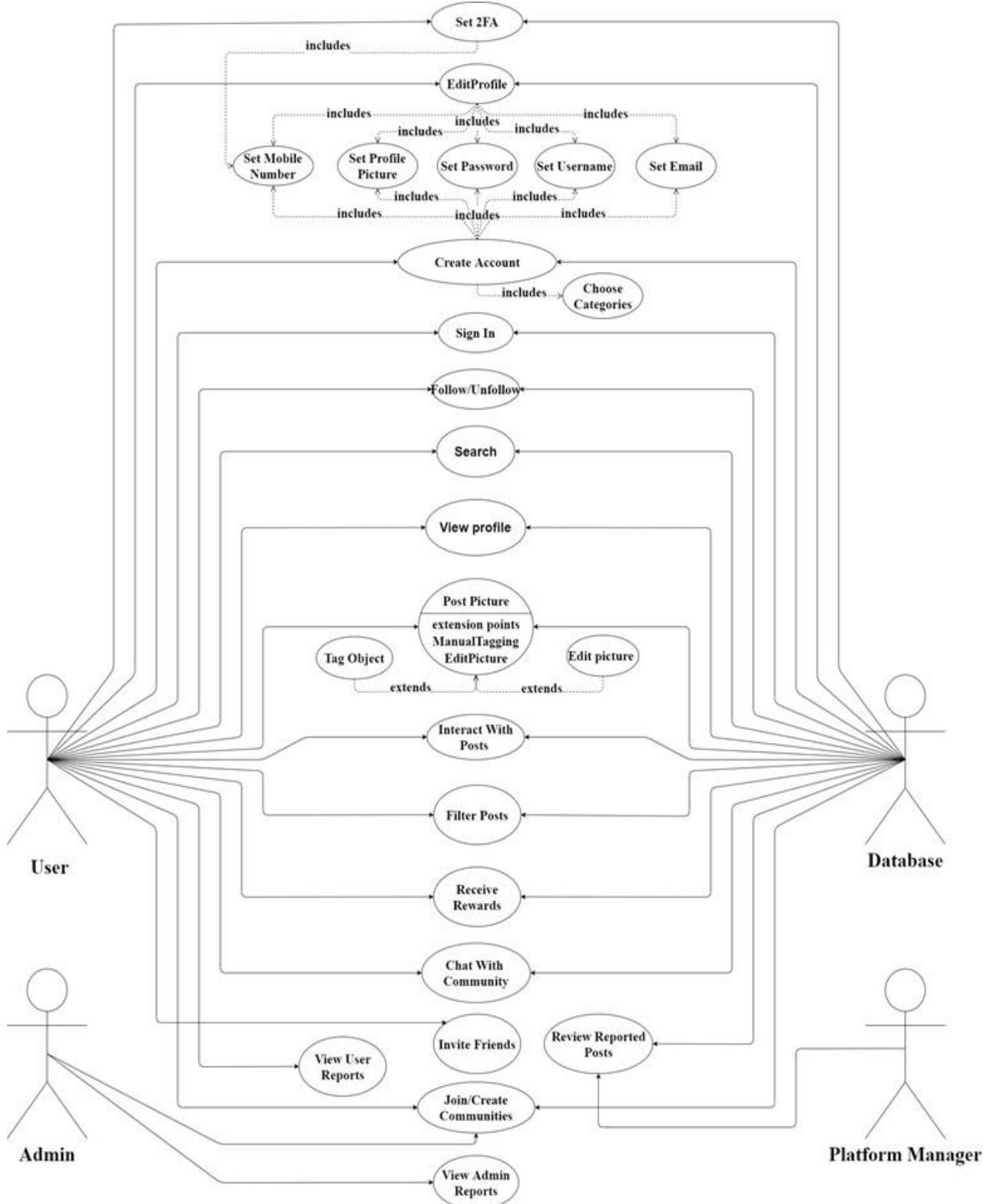


Figure 2: The Old Use Case Diagram Designed in Stage 1

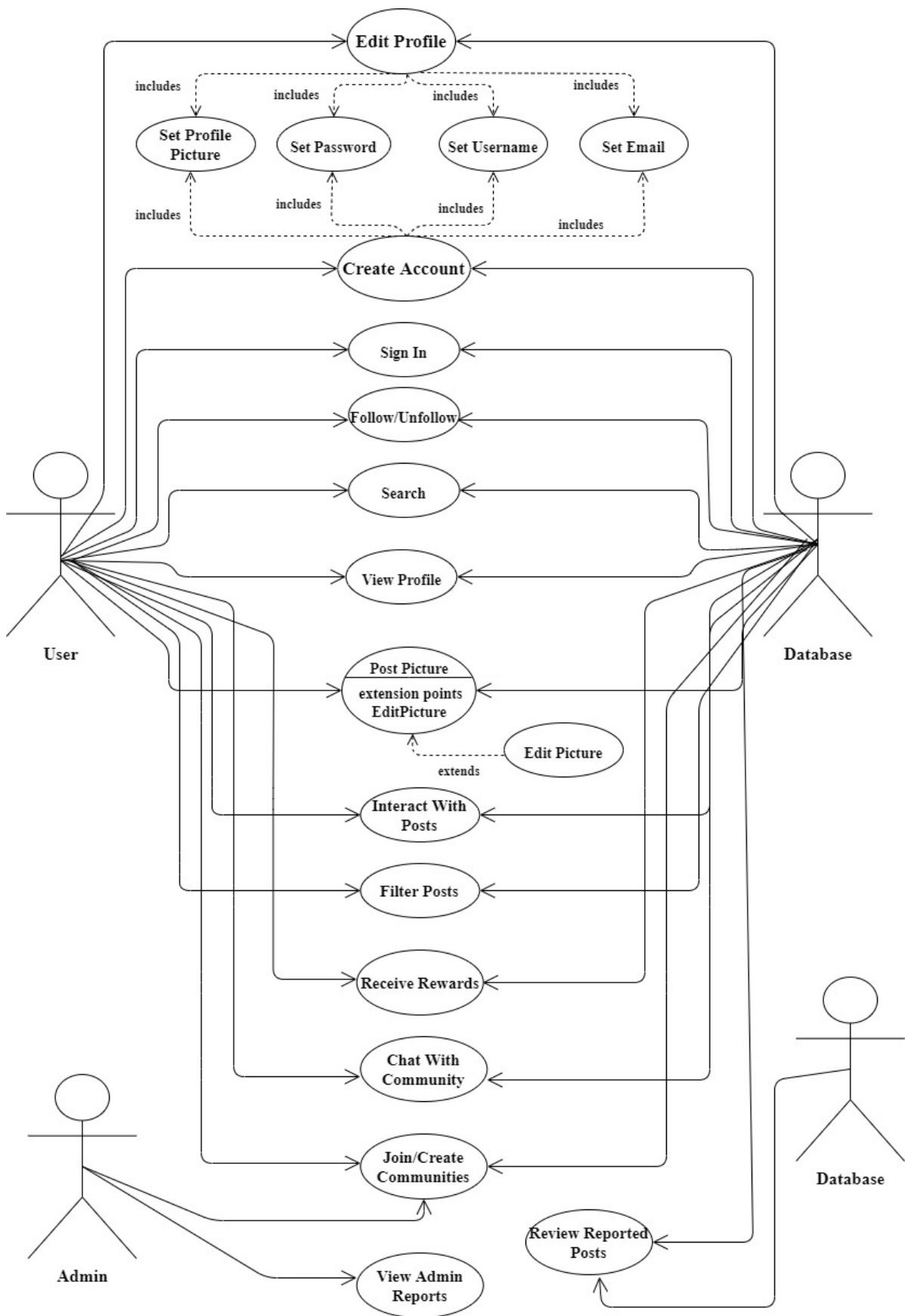


Figure 3: The Old Use Case Diagram Designed in Stage 3

In stage 1, a Use Case diagram has been created following the functional requirements of the system. Figure 2 represents that diagram. In stage 3, after having implemented the application and reflected upon the list of requirements that were created at the very first stage, we realise that some of those requirements were redundant or unnecessarily complex and could be replaced with something much simpler. Some were also not achieved due to time constraints. Figure 3 represents the Use Case Diagram of all the functionalities that have been implemented thus far.

2.2 Information Architecture Diagram

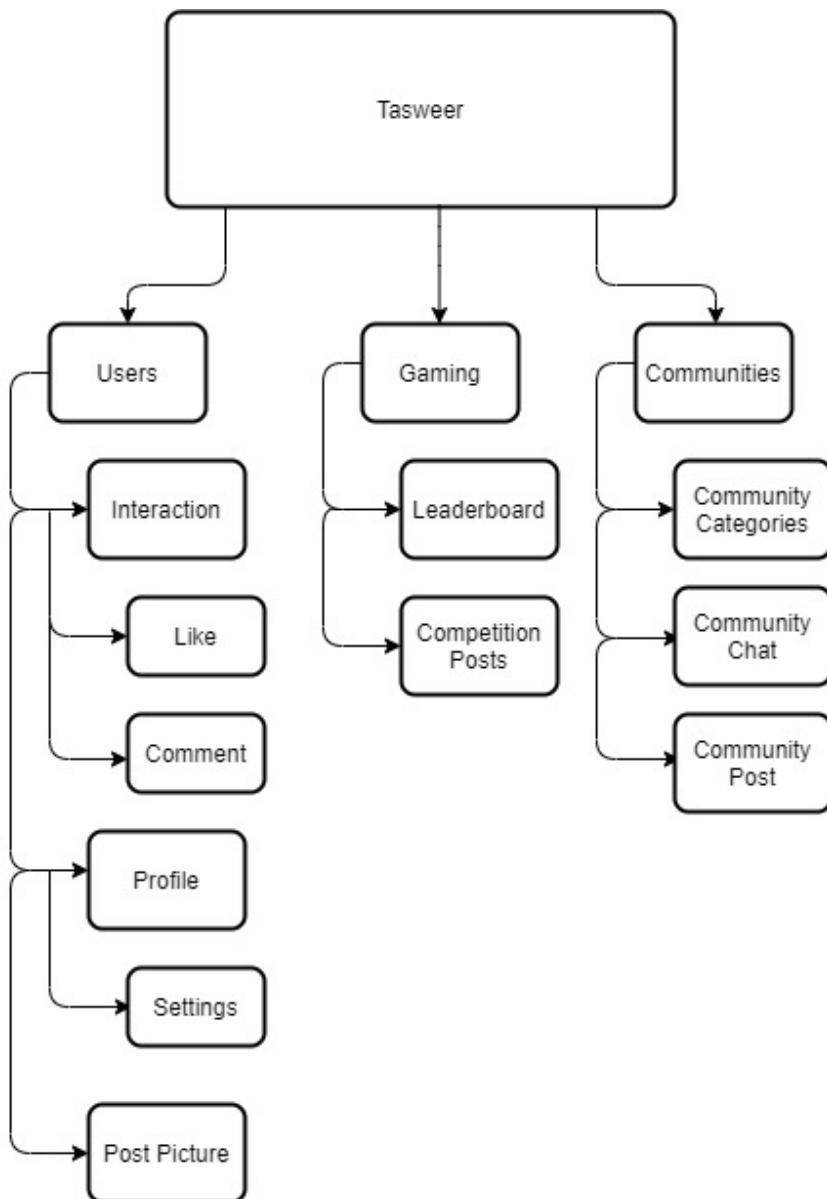


Figure 4: Information Architecture Diagram.

The above diagram shows the hierarchy of the application. The application has been divided into 3 main parts – Users, Gaming, and Communities.

The Users block is subdivided into Interaction, Profile and Post Picture.

Interaction is subdivided into like and comment as the users will be able to interact with the posts on the system by liking and commenting.

Profile is split into settings, where the users will be able to change their username, password etc.

The Gaming block is subdivided into Leaderboard, Competition Posts.

The competition posts will be displayed randomly for different users on the game screen.

The leaderboard will display the various users' ranks in the competition. Each participant will be ranked based on the number of likes that they receive on their post. The rating of the pictures will be done on the game page which is independent of the home screen.

The Communities block is split into Community Categories, Community Chat and Community Post.

The community categories include 9 broad categories under which communities based on that category can be created and joined. Users can search for different communities across all categories as well.

The community chat screen is present within each community. Users who are a part of the community can chat with other members here.

The community post screen displays the various posts which have been posted by the different members of the community. The posts will be displayed by showing the most recent post at the top.

2.3 Sequence Diagram

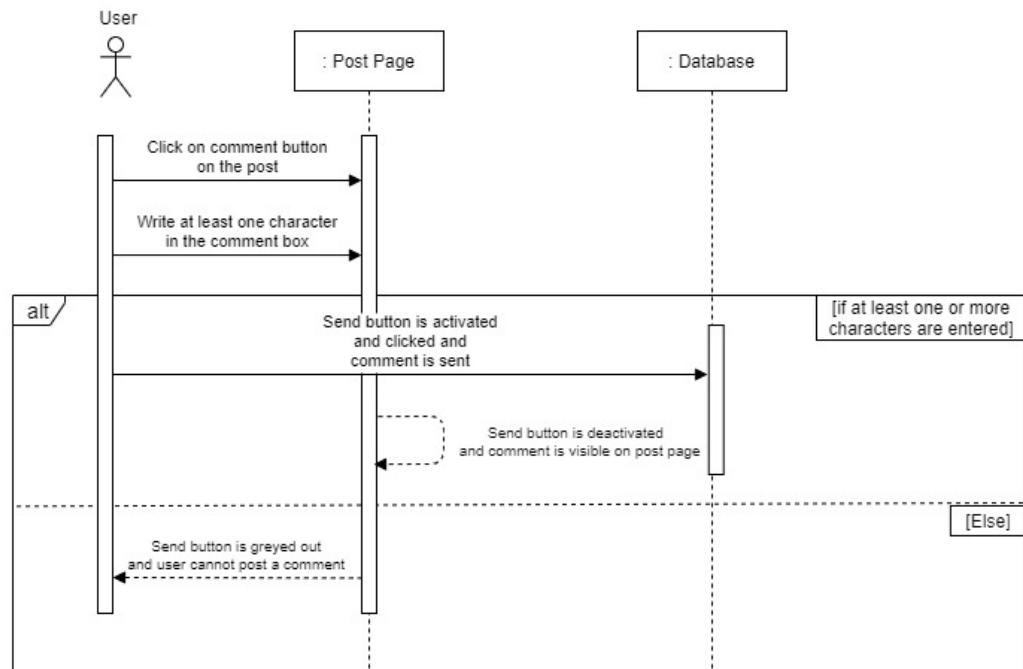


Figure 5: The Sequence Diagram for Posting a Comment Within the System

If the user decides to comment on a post, they can do so by clicking on the comment button which will direct them to the post page.

In order for the send button on the comment box to be activated, the user is required to enter at least one character in the comment box. Once the user has entered their comment, the send button gets activated and the user can click on the button to post their comment.

On clicking, the comment is sent to the database and the send button is deactivated. The comment will be visible on the page and the user can comment again if they wish to do so.

In case at least one character is not entered, the send button is greyed out and the user will not be able to post a comment. This is done in order to prevent empty comments from being posted onto the screen.

2.4 Model View Controller

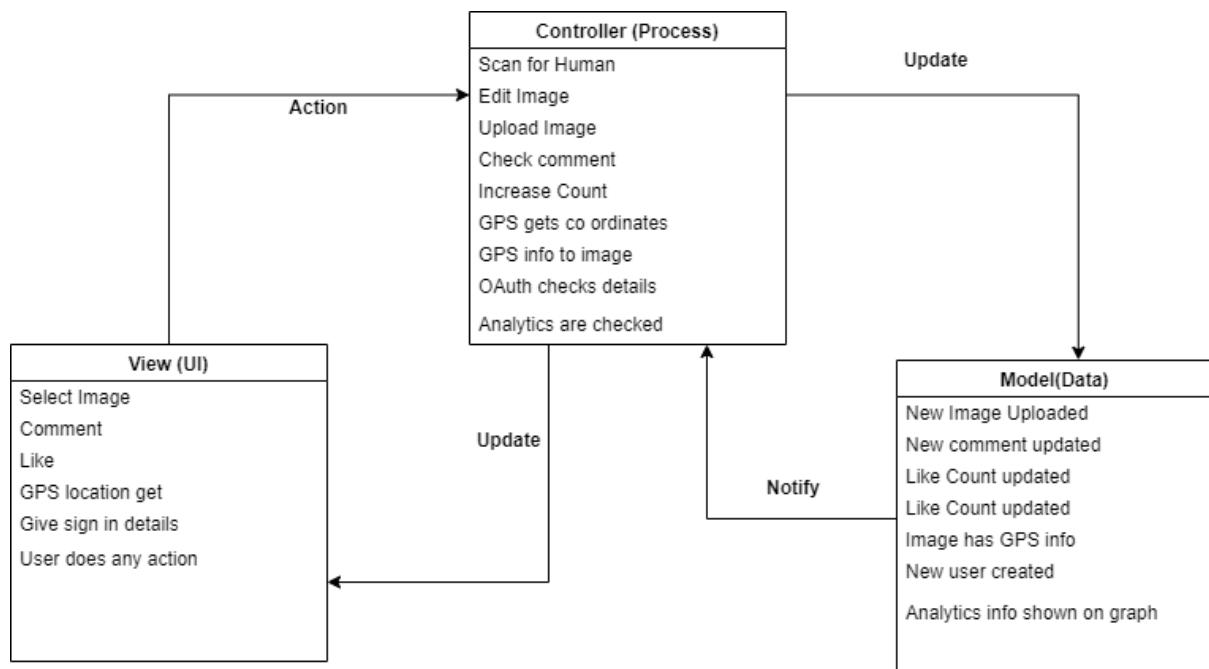


Figure 5: Model View Controller Diagram

2.5 Database Design

Table	
PK	Users
	User_Id
	Username
	User_Level
	User_email
	User_exp
	User_location
	User_password
	User_DOB
	User_mobile
	User_profilepicture
	User_categories
	User_gender
	User_follows
	User_followers
	User_accounttype
	User_joineddate
	User_ip_address
	Communities
	Is_admin
	Interested_categories

Table	
PK	Post_uploaded
	User_Id
	Post_Id
	Picture_link
	Post_Tags
	User_exp
	Location_Of_Tags_On_Picture
	Post_caption
	Post_Likes
	Post_Liked_By
	Post_Comments
	Post_Points
	Post_Multiplier
	Post_Category
	Post_Community
	Post_share_link
	Picture_Metadata
	Post_Is_reported

Table	
PK	Profile
	User_Id
	Username
	User_Name
	User_Bio
	User_level
	User_Xp
	User_followers
	Post_Id
	Profile_link

Table	
PK	Metadata
	Picture_link
	Picture_size
	Picture_Bit_Depth
	Picture_Compression
	Picture_Colour_Type
	Picture_Interlace
	Device_Type

Table	
PK	Communities
	Community_Id
	Num_user
	post_ID

Table	
PK	Weekly Challenges
	Challenge_id
	Challenge_date
	Award_id

Table	
PK	Leaderboard
	User_Id
	User_rank
	User_points
	User_Name
	User_profilepicture

Table	
PK	Awards
	Award_Id
	Award_xp
	Award_category
	Date_unlocked
	Game_id
	User_id
	User_awards
	Award_pic

Table	
PK	Admins
	User_id
	community_name

Table	
PK	competiton
	Competition_id
	Competition_theme
	Users_taking_part

Figure 6: Data schema for the system

3. Interface Design

3.1 Sign In

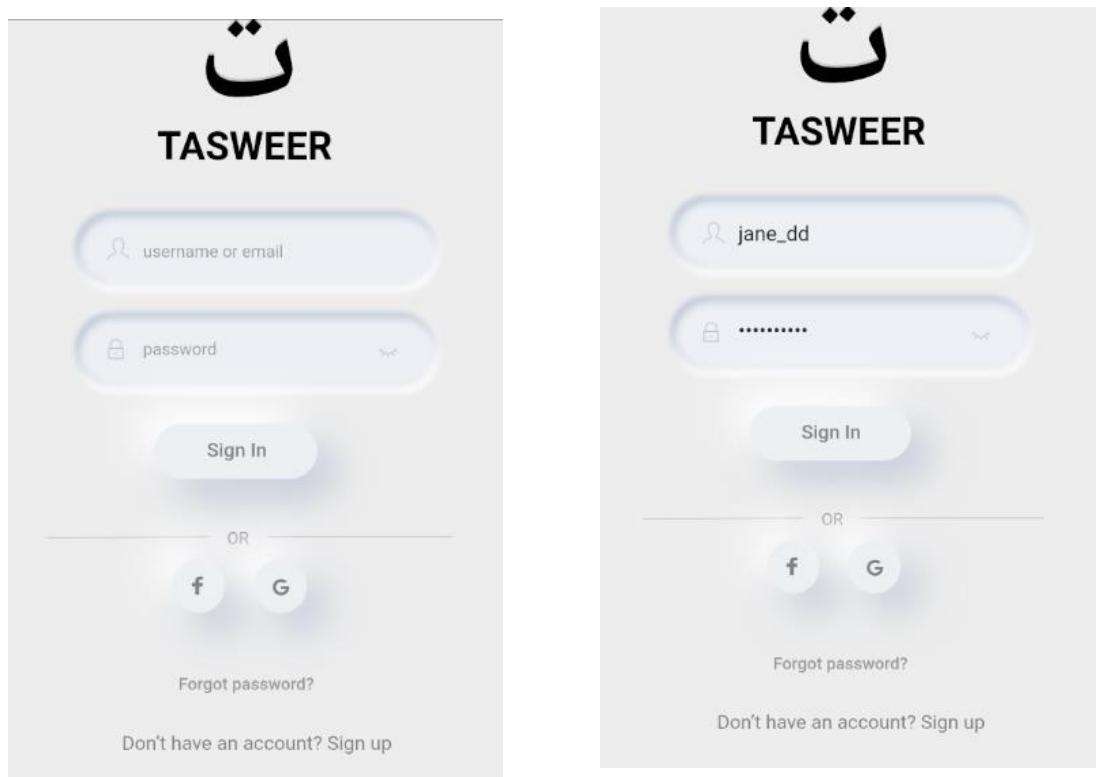


Figure 7: Sign In pages before and after the details have been filled.

Create an Account

User Name *

Required Field.

Full Name *

Required Field.

Password *

Invalid password,
make sure it is at least 6 characters.

Email *

Required Field.

Date of Birth *

Afghanistan

I agree to [Terms & Conditions](#)

Continue

Already have an account? [Sign in](#)

Create an Account

User Name *

jane_dd

Full Name *

Jane Eyre

Password *

Email *

jane_dd@gmail.com

Date of Birth *

02/06/2001

United States

I agree to [Terms & Conditions](#)

Continue

Already have an account? [Sign in](#)

Figure 8: Sign Up pages before and after the details have been filled.

In the sign up page, users are required not to leave any of the text fields empty otherwise they would not be able to continue the registration process, as seen in Figure 8. In order to fill in their date of birth correctly, a dependency has been used to allow for a calendar to pop-out and for the users to navigate around it to select their birth year, birth month and date. Another dependency has been used as well, to allow users to input their nationality. Using the dependency was a better option as it has a search feature that allows users to search through a large list of countries to select theirs.

3.2 Home Screen

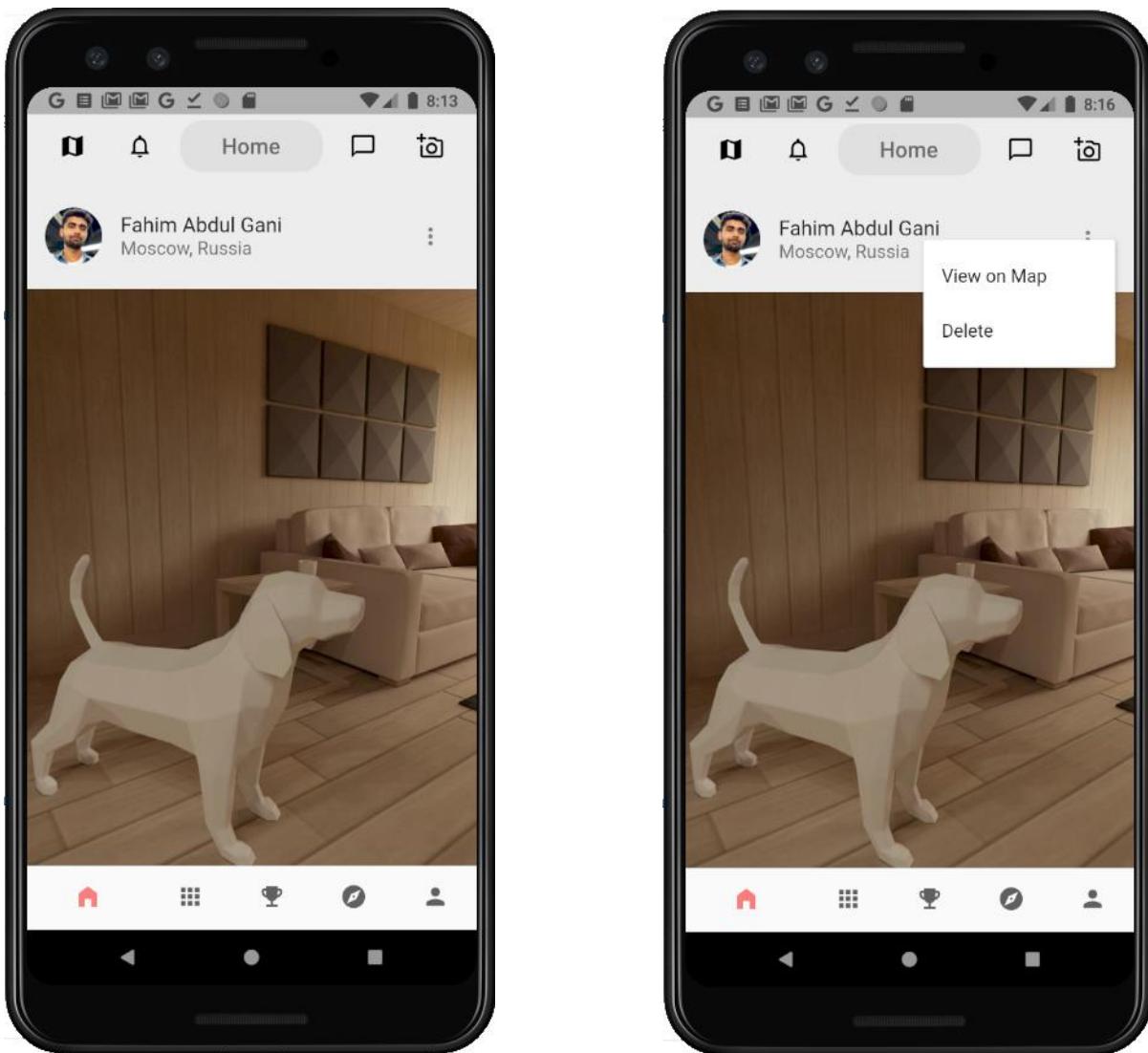


Figure 9: Home screen page where users' posts are easily seen.

The home screen contains a map icon and a camera icon at the top of the page. The user can press the camera icon to go to the image upload page as shown above and the map button to go to the map view.

Posts uploaded by the user from the image upload screen, will appear on the home screen. When a user follows someone their posts will appear on their homepage.

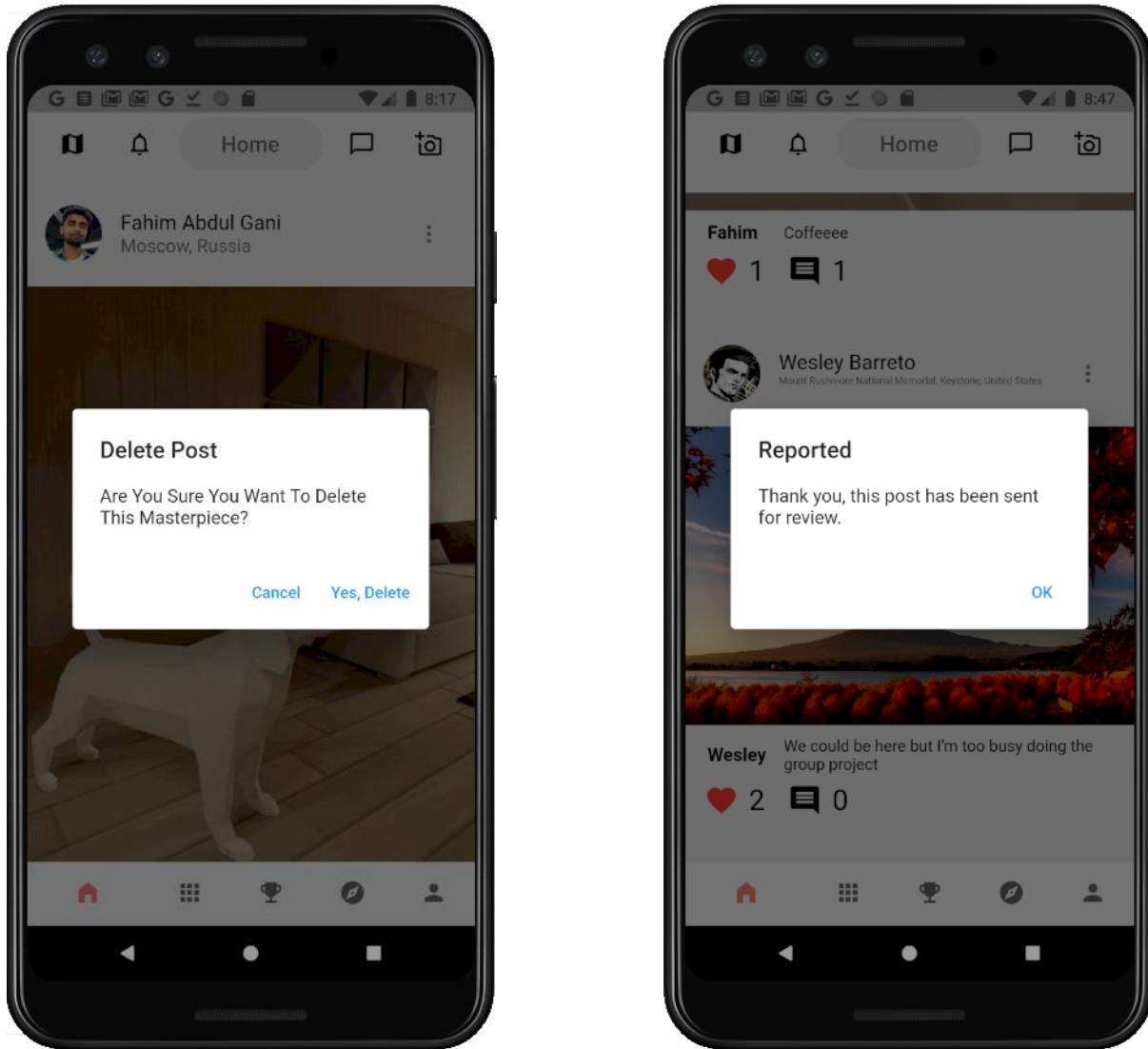


Figure 10: A post is deleted on the left and a post is reported on the right.

If a user posts a picture and they are not content with it for any reason, they can delete the post. A pop-up will appear to confirm if they want to delete the picture and if they agree, then the post disappears from the home page. It will also be removed from their profile page once they visit. Any information on the removed post in the database will be removed.

Users also have the ability to report posts. A reported post will be sent to the admins and the admins will review if the post should be taken down or not.

3.3 Upload Picture Screen

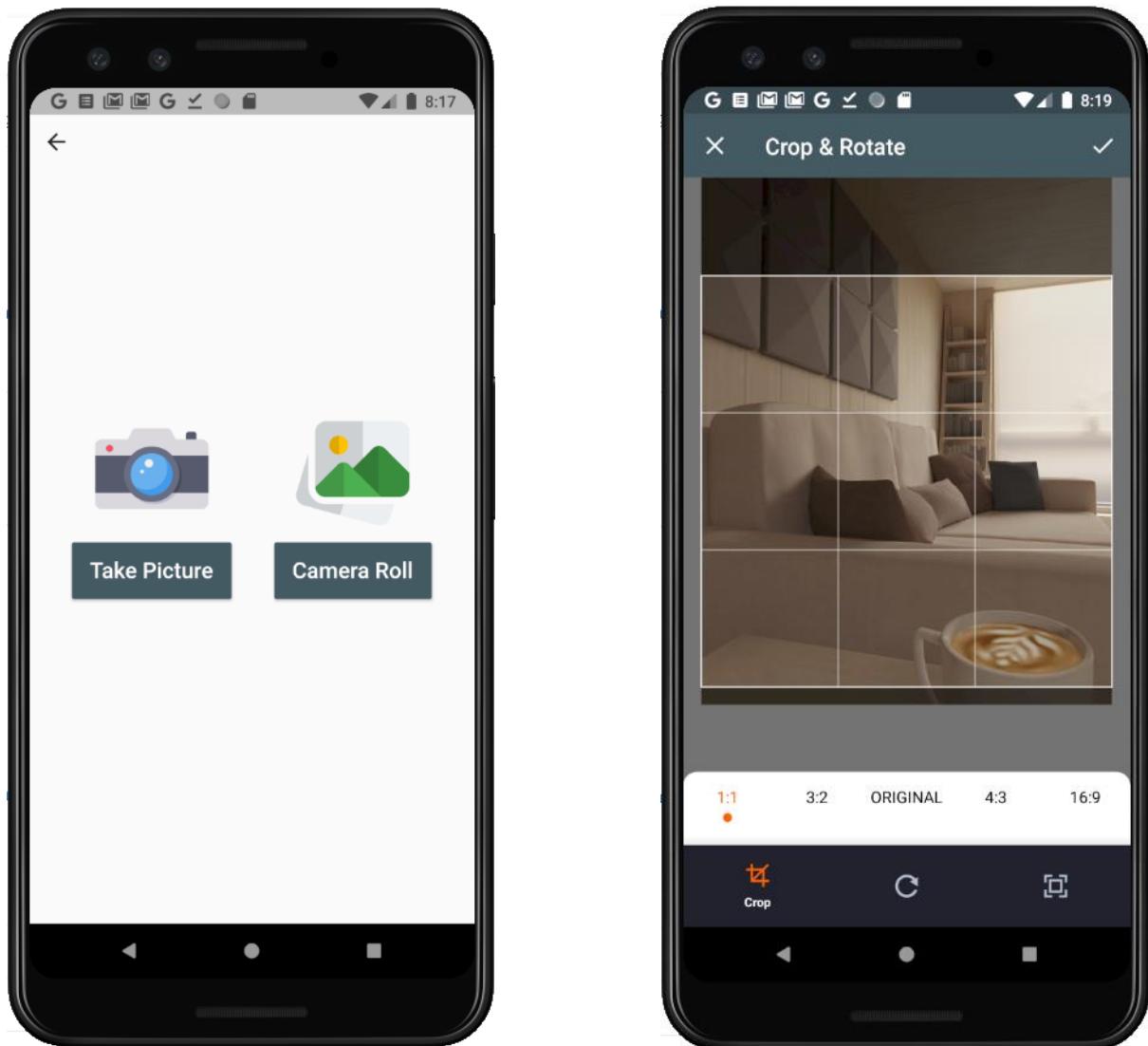


Figure 11: The process of uploading a picture.

A ML algorithm is then used to detect the presence of a person not allowing the user to post it if a person is visible. If the image is approved by the ML algorithm, it is sent to the database for storage and is retrieved to build the timeline of posted images that appear in the home screen. An issue we faced was that the algorithm does not detect people in an image when rotated by more than 90 degrees. This issue can be fixed by training the model with more images that are rotated. [See Section 1.8 Machine Learning]

The user can choose to upload an image from either the camera roll or using the in-app camera. On selecting an image, the user is taken to a screen to crop, rotate and scale the image which has been implemented using a dependency called “image_cropper”.

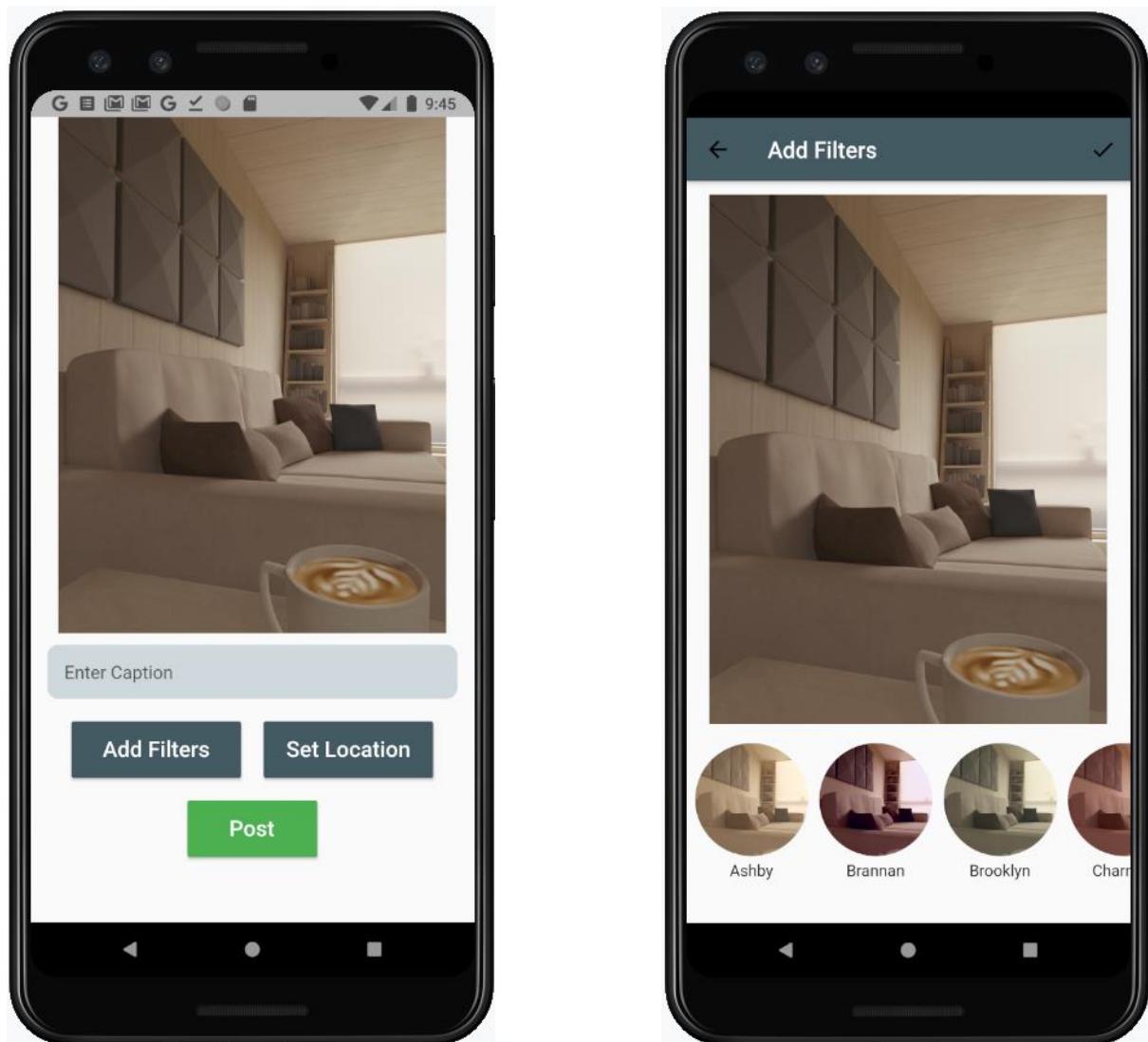


Figure 12: Users can add a location, filter and a caption to their post.

The user can also add filters to the selected image, choosing from a large list of filters given by the “photofilters” dependency.

3.4 Map View

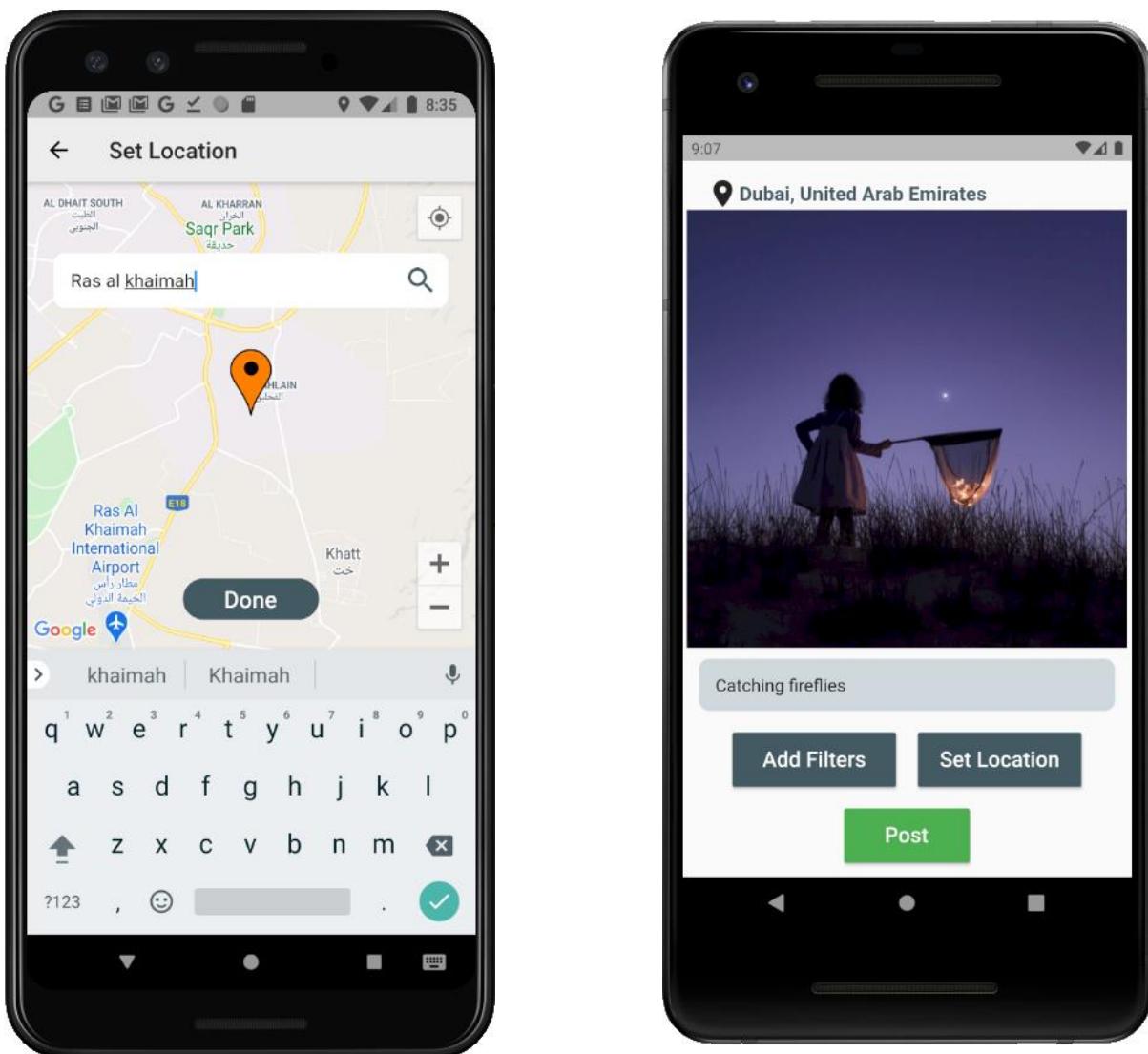


Figure 13: Users can set location for their posts.

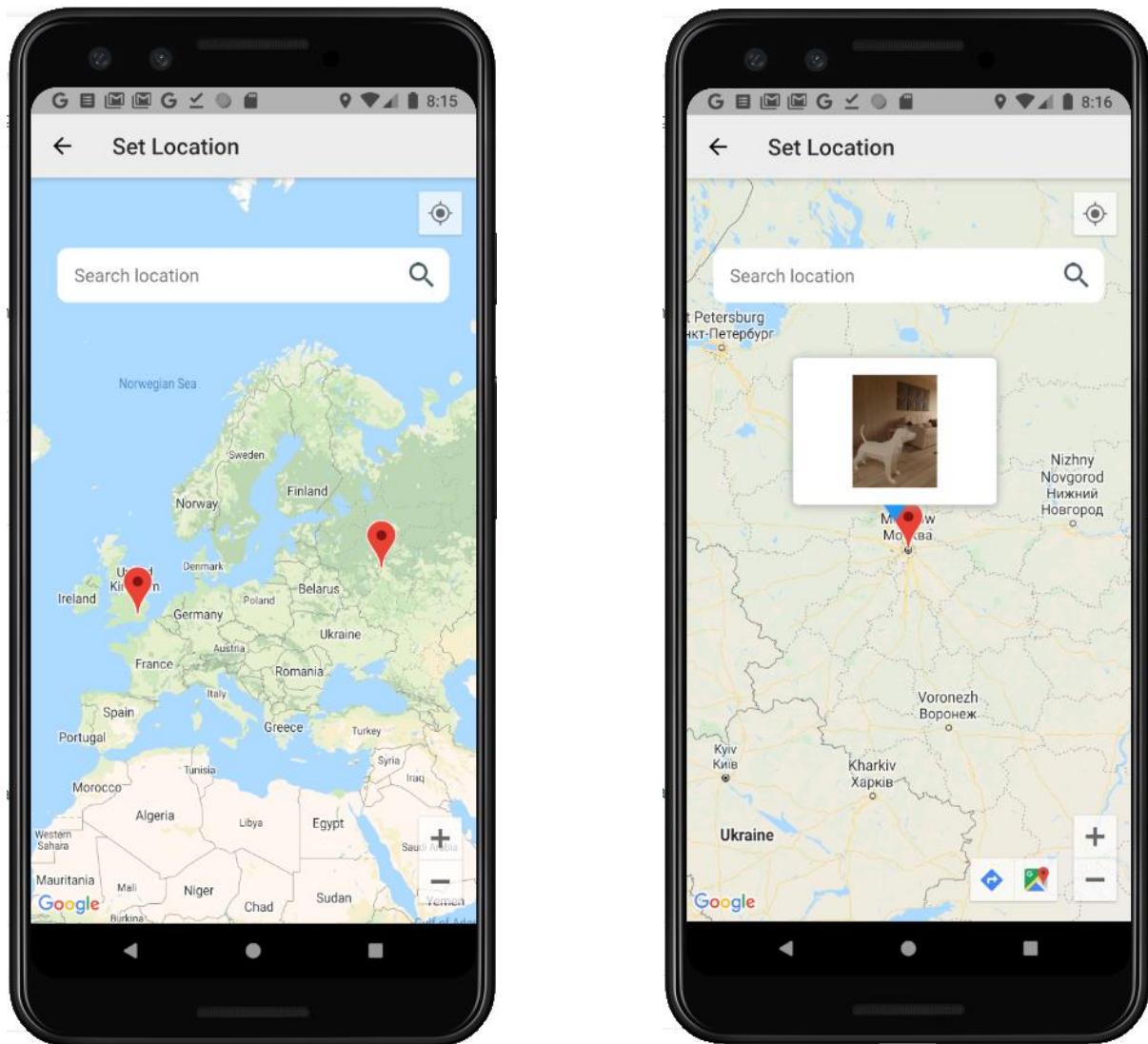


Figure 14: Posts tagged with location will show on the map.

Users are allowed to attach their locations to their posts (Figure 13) and once they do that they can view the post on the location they set from the map view. This allows users to browse the map and see what other users have captured for each location.

3.5 Profile Screen

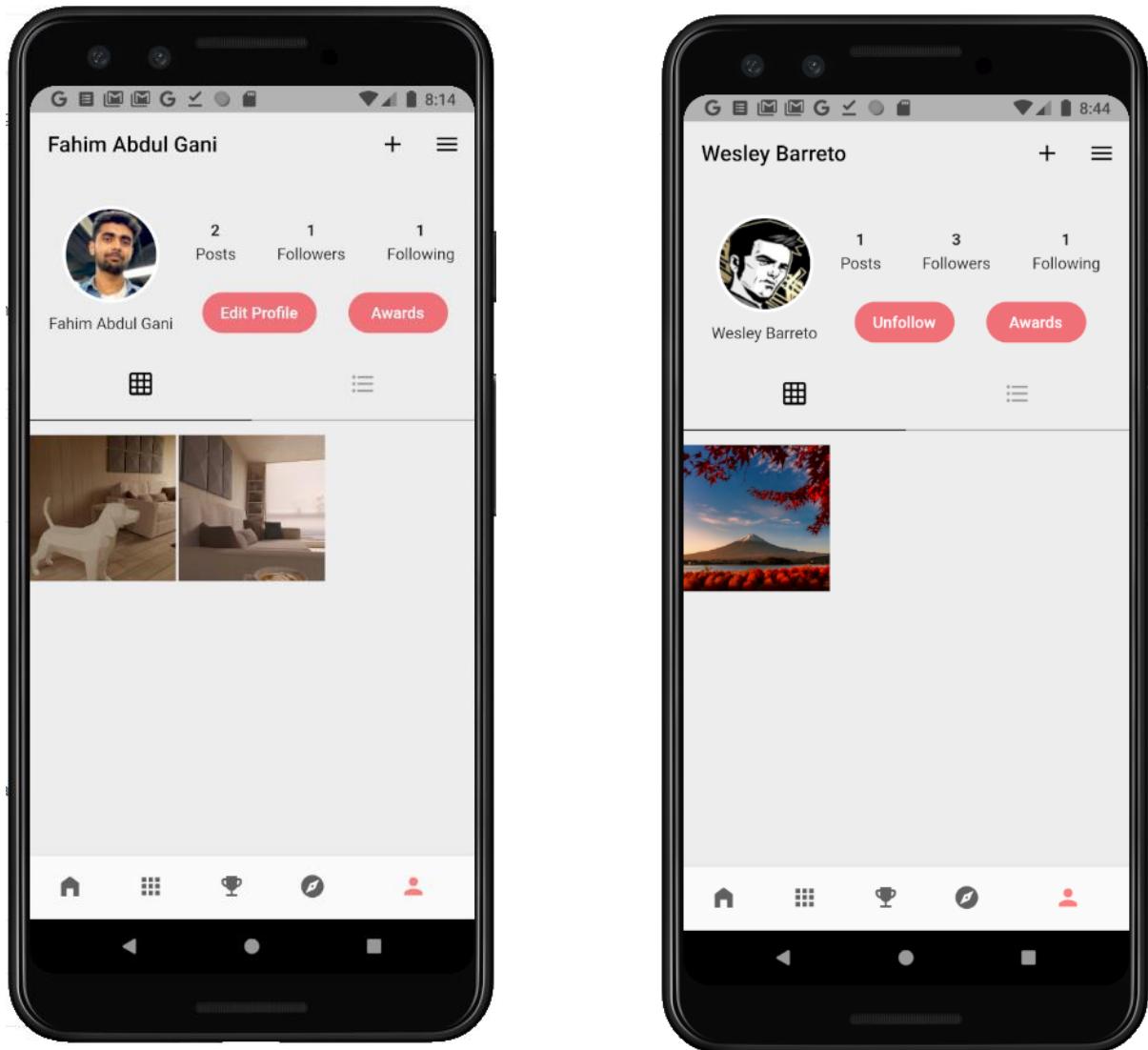


Figure 15: Examples of the profiles of users on the system.

In the profile screen, the posts of the user are displayed in a grid. The user should be able to access the settings panel by clicking on the 3 bars present on the top right corner of the screen.

On clicking the icon, opens a list of settings. Through the settings area, the users can choose to log out, access the help page, etc. as seen on Figure 16.

Users can upload pictures via the profile screen too, by clicking on the plus icon (next to the settings icon). They can upload using their camera or through their gallery. On posting, the picture gets added to the grid of posts on the screen.

For this stage, the posts are statically placed and arranged.

The ‘Edit Profile’ and the ‘Awards’ buttons, the user can edit their username, profile picture, etc. and they can view the list of awards they received in the competitions that they were a part of. The posts will also be arranged dynamically as the user posts or deletes a picture. The user will also be able to enlarge their posts, view likes and comments by just clicking on the picture.

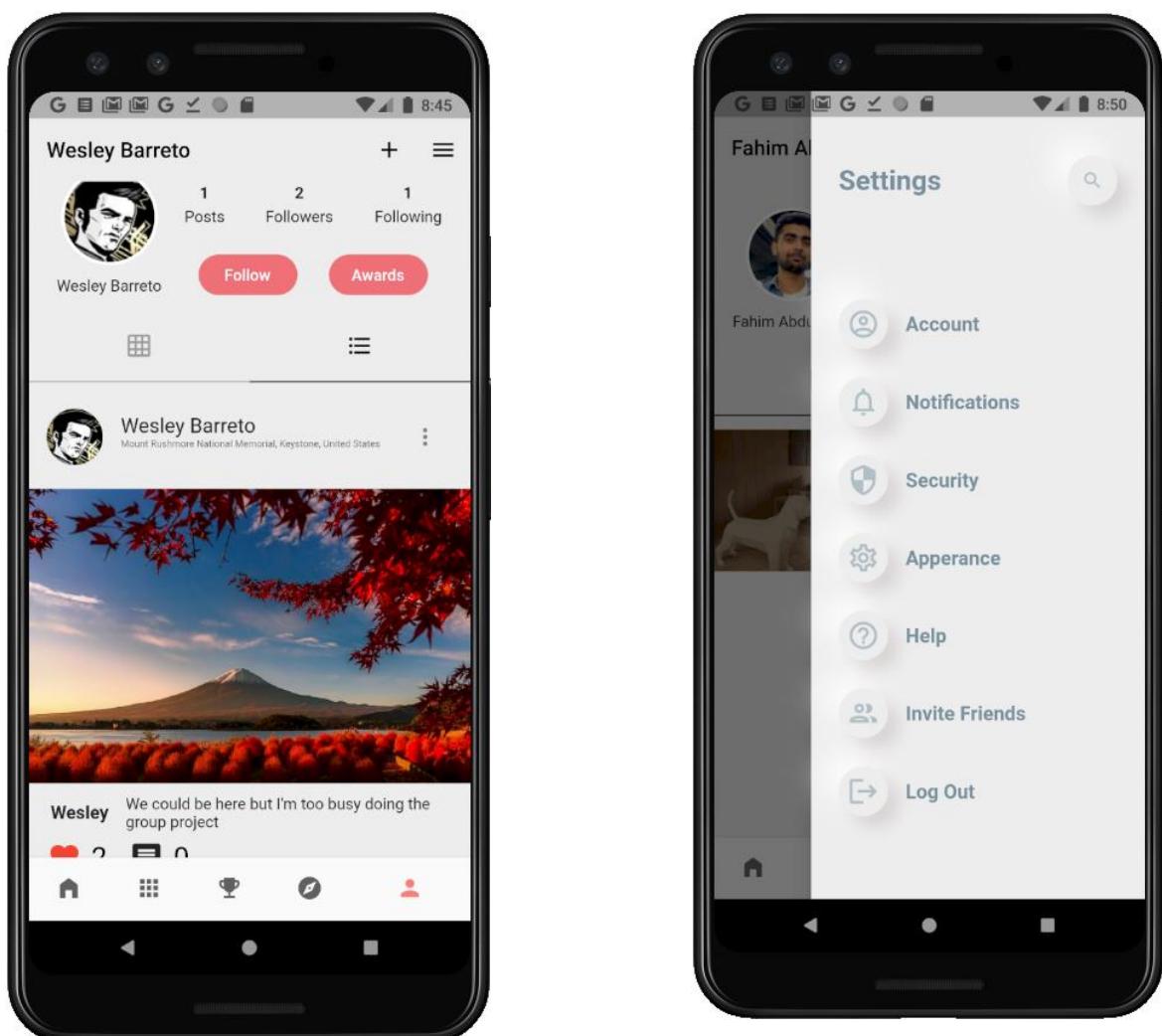


Figure 16: User posts and user settings.

3.6 Explore Screen

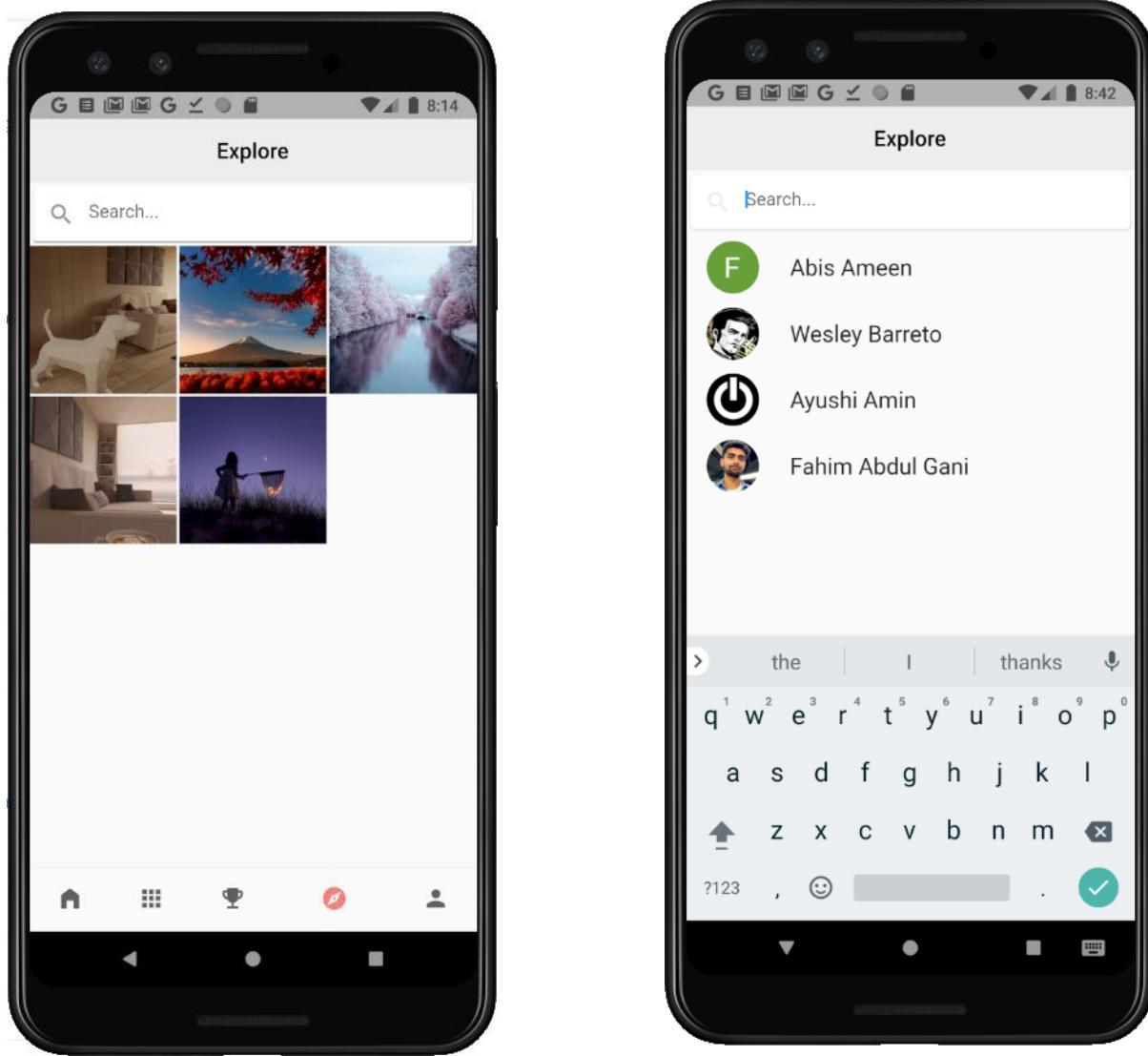


Figure 17: The explore page shows the most popular posts and the option to search for users.

The explore screen will show the most popular posts on the application. This allows the users to see what is receiving the most attention on the application. The users can also search for other users from the explore screen using the search feature. Once they search for users, they can view their profile and follow them if they want.

3.7 Communities Screen

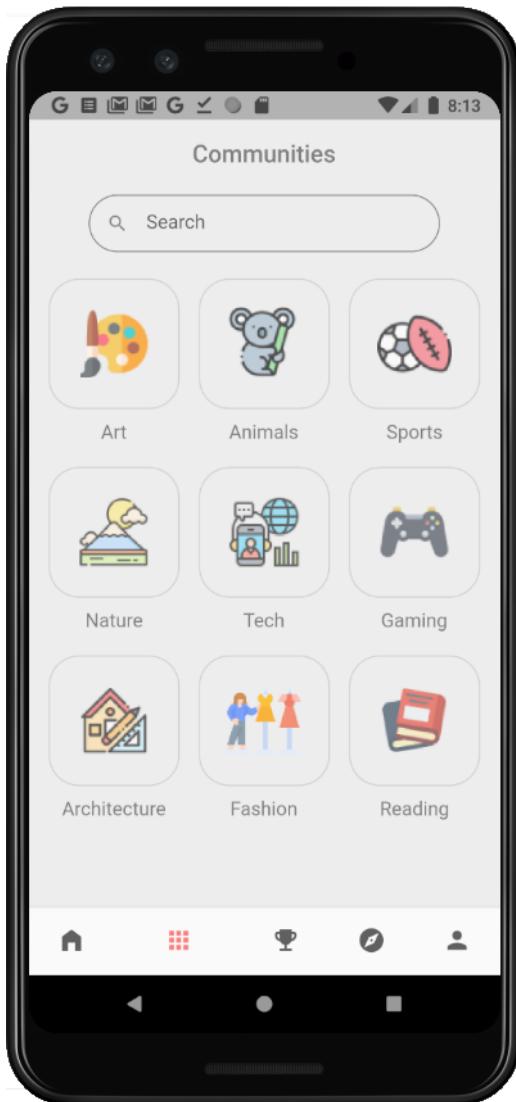


Figure 18: Examples of the communities feature within the app.

The communities page contains 9 broad categories which are: Art, Animals, Tech, Sports, Nature, Architecture, Gaming, Reading and Fashion. It also has a search bar where users can search for communities that are present in all these categories.

On clicking a category, the user is redirected to a page containing the list of all the communities they are a part of as well as all the other communities that are present within the category.

On clicking a community in that list, the user is redirected to that communities page which contains a chat area where all the members of the community can chat with each other and a posts area where the all the community posts will be displayed. The chat area uses the “intl” to format the time when the user sends a message. The time is displayed along with the chat message in the chat area.

3.8 Comments Section

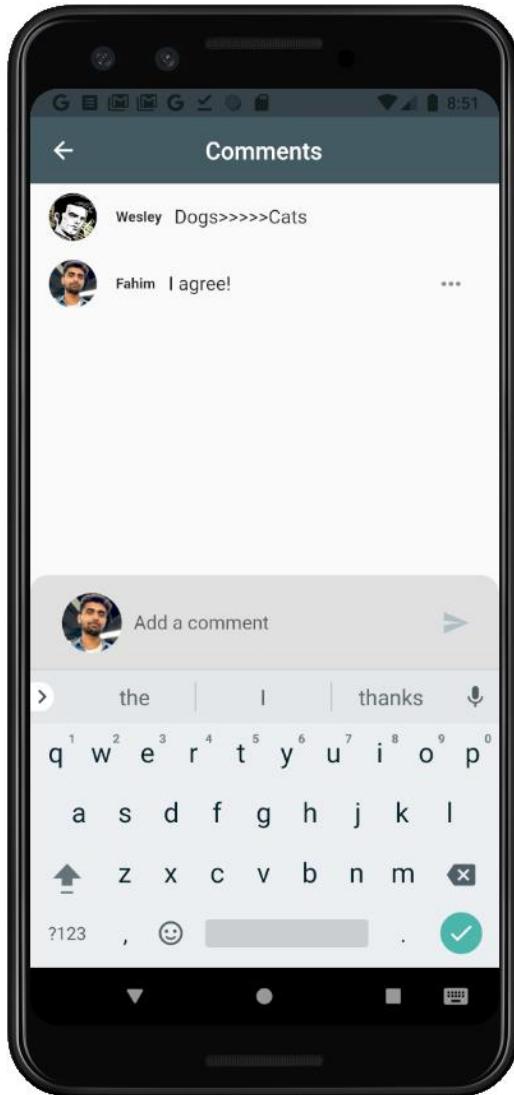


Figure 19: Examples of how comments on a post look.

Users can leave comments on other users' posts. Leaving a comment will provide users with points and hence this would promote interaction within the system.

3.9 Game Screen

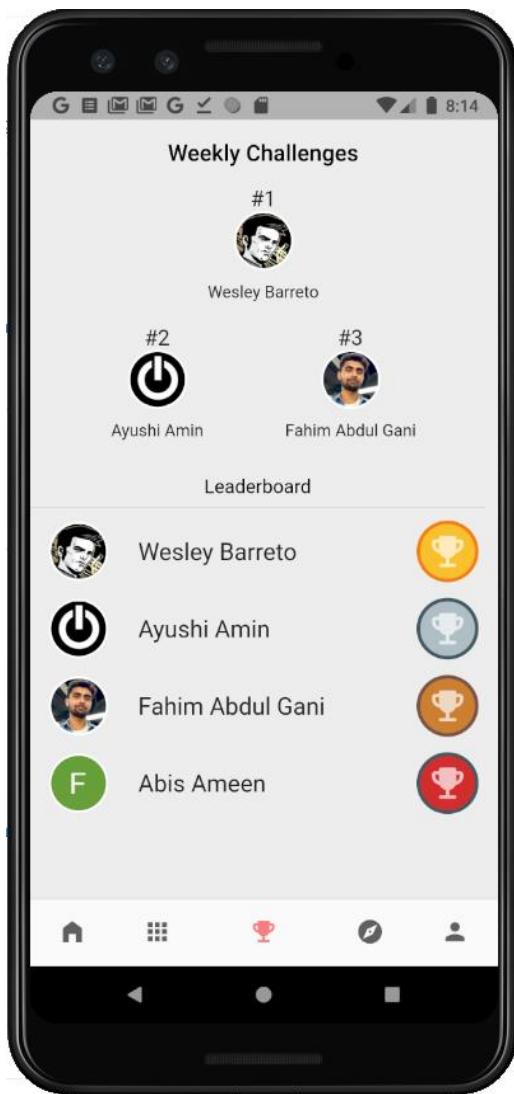


Figure 20: The leaderboard feature of the feature.

Users would collect points through interacting with other users on the application such as liking a post or commenting on it. The more points they collect the higher they go on the leaderboard. Users can go on the leaderboard to see who the top users are. This is a part of the gamification of the application.

3.10 Help Page

The screenshot shows the 'HELP CENTRE' section of the app. It includes a 'Welcome to the Help Centre!' message, a 'Frequently Asked Questions (FAQs)' section with links to video tutorials for 'How to use the application', 'Sign Up for an Account', 'Login Troubleshooting', and 'Account Settings'. It also features a 'Report a Problem' section with links for 'Spam or Abuse' and 'Something Isn't Working', both leading to report forms. A 'Feedback' section contains links for 'Provide General Feedback' and 'Provide Feedback', both also leading to form pages.

HELP CENTRE

Welcome to the Help Centre!

Frequently Asked Questions (FAQs)

[How to use the application?](#)
Please refer to the following link for a [video tutorial](#).

[Sign Up for an Account](#)
Please refer to the following link for a [video tutorial](#).

[Login Troubleshooting](#)
Please refer to the following link for a [video tutorial](#).

[Account Settings](#)
Please refer to the following link for a [video tutorial](#).

Report a Problem

[Spam or Abuse](#)
Please refer to the following link and [report a problem](#).

[Something Isn't Working](#)
Please refer to the following link and [report a problem](#).

Feedback

[Provide General Feedback](#)
Feel free to provide any feedback you have towards the app. [Fill a feedback form](#).

[Provide Feedback](#)
Feel free to provide any feedback you have towards the app. [Fill a feedback form](#).

Figure 21: The help page of the app.

This help section is accessible from the profile page of any user (Figure 15: Profile Screen), where on the right side there is a button that once gets clicked will reveal a sidebar that contains user settings. Users can access the help page to check the frequently asked questions, where there is a video tutorial to guide them through using the app and common issues they may face. They also can report a problem by clicking on a link that will direct them to a form where they can fill in the information regarding their issue and a staff member would get back to them within 48 hours.

Feedback is always welcomed as users may spot bugs, or have recommendations they may want to see being implemented in the application. Such ideas can be reached out to us via the feedback form.

3.11 Photographer Profile

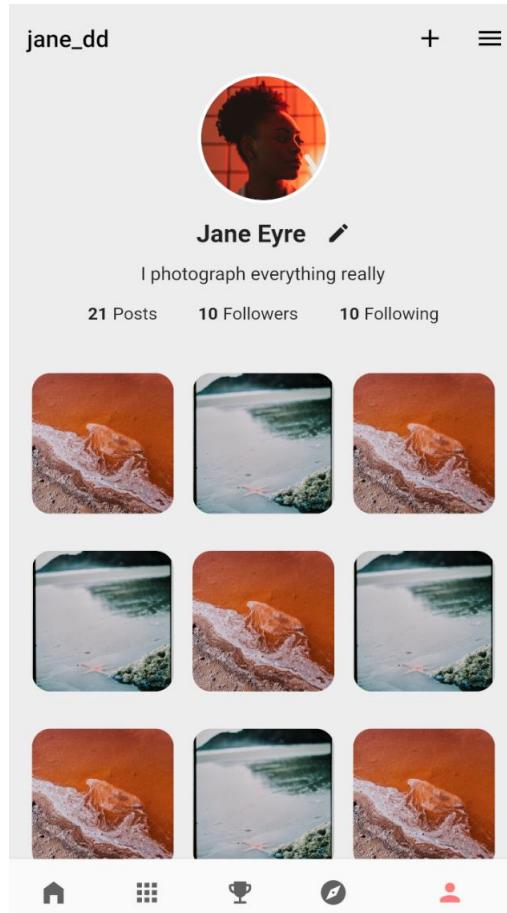


Figure 22: Photographer profile.

To allow a wide range of users to customise their experience on our application, users will have the option to select the type of account they prefer to have during the registration process. They also have the chance to change their choice of account even if they signed with a general account. A photographer's profile is meant to showcase the work they have captured, so it is a much simpler interface to draw away from any distractions that users might find on the screen. The different interface also helps to differentiate between the general account so when users choose to change accounts.

3.12 Gamer Profile

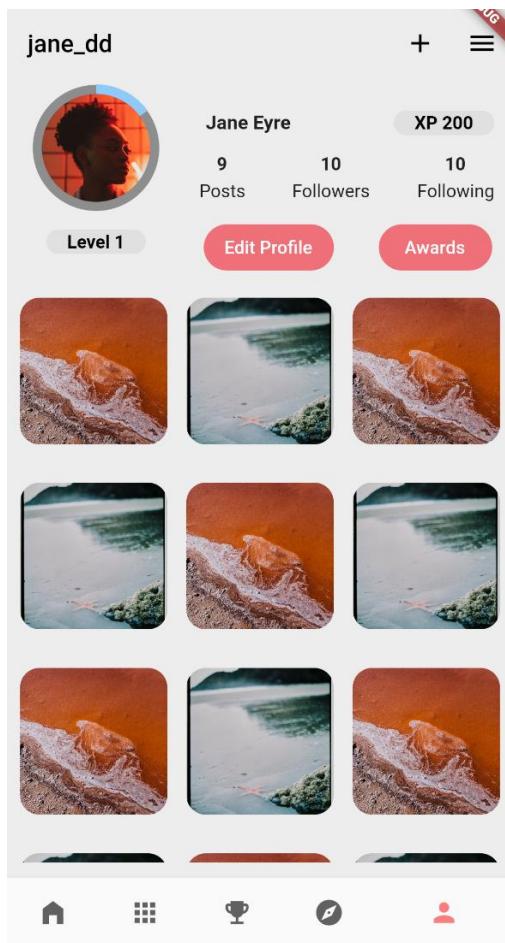


Figure 23: Gamer's profile.

Users who want a gamified experience while using the application can choose the gamer profile. This allows them opportunities of gaining points when they like posts or leave comments. Their score gets accumulated and they are able to unlock levels and awards. They also get points when they post pictures and can unlock category-specific awards. In a gamers profile, other users will be able to see the level they are at, the blue progress bar that visually showcases their progress until they unlock the next level. Other users can also view the awards they gained.

3.13 Awards



Figure 24: Awards a user unlocked.

The awards the system gives out vary. Users can post pictures and they can be awarded depending on which category does the picture fall under. Users can also gain awards for completing simple tasks such as liking a post or posting a comment. The awards section comes with gamers profiles.

3.14 Landing Page

Company landing page link: <https://zentech2.herokuapp.com/>

The company landing page was created using basic HTML, CSS and JavaScript code containing the company's description, list of recent projects, current and past clients, the team members and other relevant information. The app logo for Tasweer within the "Our Projects" section of the website links to the app landing page that showcases some of the features of the app along with links to a helpful video demonstrating the prototype and an online version of the prototype itself.

Both these landing pages dynamically respond to size changes which enable them to be viewed on different screen sizes. (See Section: Appendices for screenshots of the company website)

4. Implementation Methodology

This section discusses the implementation methodology used during the project and how each stage ma

In the beginning, the team had a discussion and believed the best method to use in the project was Scrum. Instead of using the traditional method, that would mean that a leadership would exist in the form of a Scrum Master, we found that the team's dynamic and synergy worked better as a self-organising team. All the members made the choices of how best to accomplish their work after discussions held during stage 1 instead of being directed by a Scrum Master hence benefiting from collective wisdom. (Schwaber and Sutherland, 2020)

However, our team found further division made communication clearer and allowed for more efficiency, so that different teams would be working on the same tasks simultaneously. For the sprints, the team would work towards completing their tasks before the stages' deadlines as well as completing their tasks before additional deadlines set within each sub-team. For example, the sub-teams would aim to

complete their tasks before their next group meeting. That would usually be held every 5 to 7 days to ensure transparency in work and accountability.

In stage 1, everyone in the team came up with various user requirements (functional and non-functional) i.e. the product backlog to be implemented in our system and drafted a project plan (Gantt Chart), which was to be executed from stage 2 onwards.

During stage 2, the team broke down into two sub-teams: front-end (Ayushi, Mayar, Wesley and Fahim) and back-end (Abis, Ammad and Fayyaz). These sub-teams were formed based on what task each member chose and in each of these sub-groups, there was no leader. In these sub-groups, each member would choose which task they would prefer working on and deadlines would be set on a weekly basis. The members would then meet up and review each other's work and then each of them would choose the next task that they preferred working on.

However, before the stage 2 deadline, Wesley (from front-end) and Abis (from back-end) paired up to integrate the front and back-end together. Whereas, the other members of the main group completed the remaining tasks, for instance, Ayushi and Mayar worked on the stage 2 report and were later joined by Wesley and Fahim into completing the report (the remaining members, filled in their area of expertise).

During stage 3, the teams were split even more. Out of a team of 4 (front-end), some handled the front-end part of the system and the stage 3 report, while other members shifted from front end to join the back-end team. Here also, members would choose whichever task they preferred working and there was no leader who assigned tasks to them. They would then again meet up on a weekly basis and they would then choose which task they would prefer working on.

There were cases of pair programming as well. For example, Ayushi and Mayar coded and debugged some aspects of the front-end, whereas Wesley and Fahim coded together. At times, Ayushi, Mayar, Wesley and Fahim would conduct coding sessions together to complete some of the functionalities of the system.

Refer to table below for a visual representation of how the tasks were broken down for each stage in the project and the duration each task took.

Moreover, Section: 2.1 Success of Implementation Methodology goes into how discussions whether this method was successful and refers to examples throughout the entire project.

The tables specified below identifies the various tasks completed in each stage and the duration taken to complete them.

As work was done simultaneously, the deadlines were flexible as tasks shuffled around across members and were paused momentarily in favour of working on other tasks of more importance. Hence, there are no accurate start and end dates and all that is available is an approximation of the duration that each task took.

Note: Duration is the number of days/weeks spent on the task mentioned in the below table.

Stage 1

Task	Duration
Requirements	1 week + 2 weeks updating
UML Diagrams	2 weeks
Use Case Specification	1 week
Project Costing	1 week (including changes)
Project Plan	1 week
Risk Analysis	1 week
UI Design	2 weeks
Usability Study	3 days

Report (writing, editing, compiling)	2 days
--------------------------------------	--------

Stage 2

Task	Duration
Website	2 weeks
Database initialisation, design and set up	2 weeks
User Signup and Login implementation and testing	1 week
Home Screen implementation and testing	1 week
GPS	2 weeks
Machine Learning Model	3 weeks
Report (writing, editing, compiling)	1 weeks
Communities Screen (front-end) implementation and testing	2 days
Explore Screen implementation and testing	3-4 days
Frontend and Backend Integration	4-5 days
Profile Screen implementation and testing	4-5 days
Uploading functionality and testing	1 week
App Testing (after compilation)	2 days

Stage 3

Task	Duration
Post caption	1 hour
Frontend and Backend Integration	4 weeks
GPS remodelling and testing	3 weeks
Usability Study	2-3 days
Report (writing, editing, compiling)	1 week
Admin Analytics	1 week
Communities Screen, implementation (front-end and backend) testing	3 weeks
App Testing (after compilation)	2 weeks

5. Tests for Technical Correctness

5.1 Unit testing

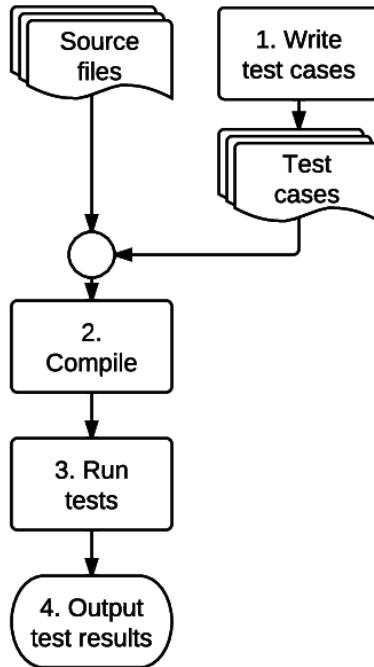


Figure 26: Process of unit testing.

Unit tests helped us to verify the behavior of a single function, method, or class. The `test` package in flutter provides the core framework for writing unit tests, and the `flutter_test` package provides additional utilities for testing widgets. Code that is tested by checks is more stable than code that is not. If a potential update breaks something in the code, our developers will be able to quickly locate the source of the problem rather than having to sift through a large codebase.

5.2 End to End testing

End-to-end testing is a form of software testing that involves testing an application's workflow from start to finish. This approach aims to simulate real-world usage situations in order to assess the system's communication and data confidentiality.

The steps below are required for our testing:

- Analyze requirements
- Set up a test environment
- Analyze software and hardware requirements
- List down how every system needs to respond
- Design test cases
- Run tests, study and save results

5.3 Firebase Testing

Test matrix	Test type
Matrix #646730	Instrumentation
Matrix #546149	Instrumentation
Matrix #663810	Instrumentation
Matrix #886799	Instrumentation
Matrix #586883	Instrumentation
Matrix #488151	Robo
Matrix #593381	Instrumentation
Matrix #873836	Instrumentation

Figure 27: Tests failing and passing.

Firebase Test Lab is a cloud-based software development infrastructure that allows us to test our app on a variety of devices and settings and see how it behaves in real-world situations. Test Lab uses real, production devices running in a Google data center to test your app. The devices are flashed with updated APIs and have

customizable locale settings, allowing you to road-test your app on the hardware and configurations it'll encounter in real-world use. When we run a test or a set of test cases against devices and configurations. Test Lab runs the test against the app, then displays the results as a **test matrix**.

Devices × Test Executions = Test Matrix

5.4 Map view testing



Figure 28: Visual representation of geolocation feature.

Geolocation is the identification or estimation of the real-world geographic location of a mobile phone or computer. Because we are dealing with devices and networks, it is important to be aware of the issues we may encounter that are uptime, connection speed and accuracy. Hence, we conducted geolocation testing. The technique of identifying the geographical location of a device via the internet. In other words, this kind of testing includes everything related to the position of an object in space, actions based on the object's position regarding other objects, and the effect on different processes and areas. We tested the point and the path. A Path is made up of a series of Points. To make a Path, you must join different or more Points. As a result, this feature will view how a system traveled during the day or construct a route to the destination node. We also tested the point at which the main nodes connect. Point parameters are latitude and longitude, since elevation coordinates

are used more rarely and only in specific cases, like atmospheric pressure calculation. (Google Developers, 2021)

5.5 Posting a Picture

We tested this by sending pictures taken from the in-app camera function of the app. Since our system does not allow for pictures that contain humans to be posted, we ran our face detection function [described above in section 5.9] on the picture that was to be posted.

The following are known issues with our current face detection function:

1. The function sometimes does not detect the presence of human faces in pictures. This was observed to be the case when the size of the face was small in the image (shots that are not close-up pictures).
2. The function mistakenly identified an animal for a human in one of the pictures.
3. The AI model identified human faces in one of the images we used for testing, but when the same image was rotated by 180 degrees, it failed to recognise any faces and allowed posting of that image.

We will train the model with more test images and images that are rotated, in order to resolve these issues. These should be resolved by stage 3 of the project.

5.6 Posting Comments

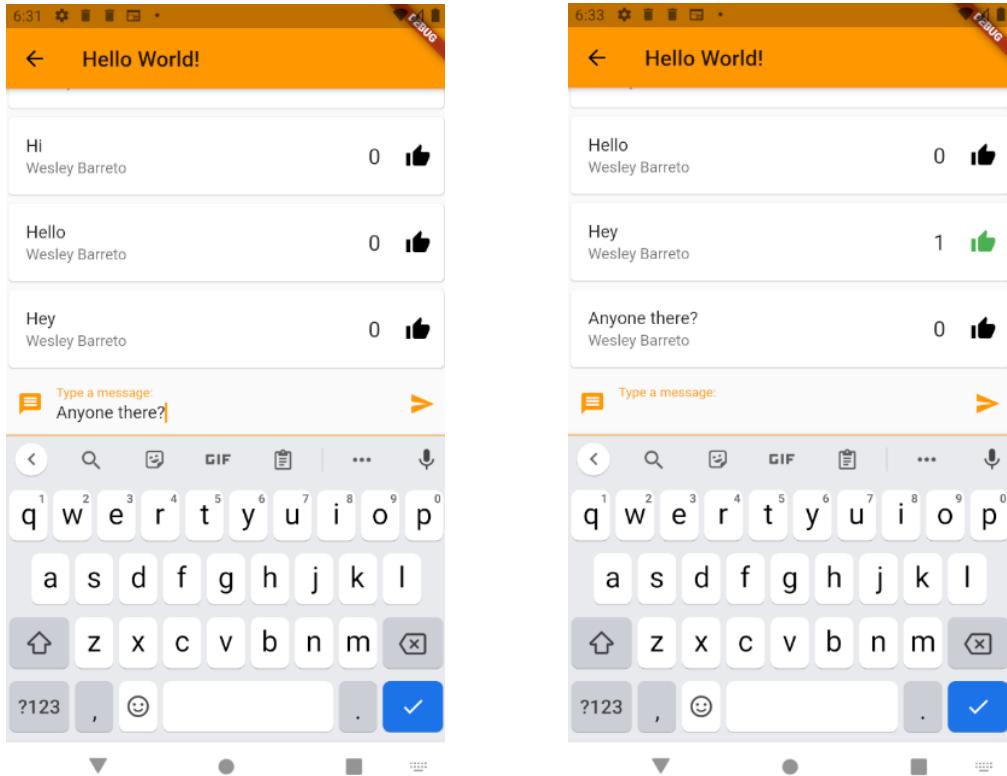


Figure 29: Posting comment functionality.

We tested posting comments to the database by creating a sample UI which contained a text box and list of cards of comments that were posted to display the results. On posting a comment from the text box, it would get sent to the database and be displayed on the comments card along with the name of the user. The like button on a comment card was also functional and was registered in the database.

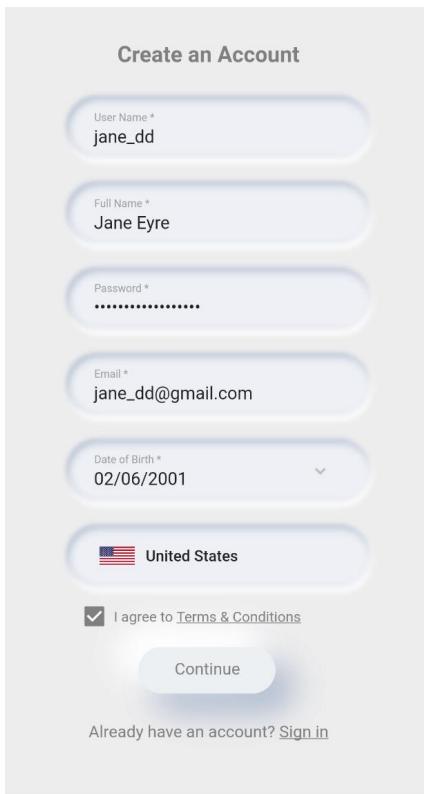
6. User Guide

This section will guide potential users of the system on how to get started with the application and navigate their way through to use its features.

6.1 Install the Application

Users can refer to the following link to install the application on their mobile devices.
https://drive.google.com/file/d/14KLx8PNy3vA466_Ouj5mcPnSZanMbKHN/view?usp=sharing

6.2 Create an Account

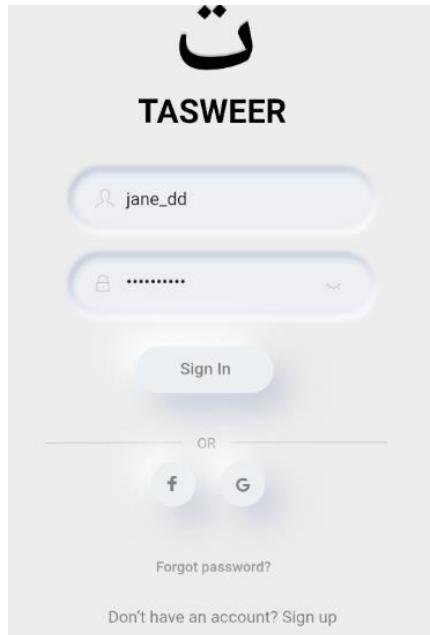


The screenshot shows a mobile-style 'Create an Account' form with rounded corners and light gray background. It consists of several input fields and a checkbox area.

- User Name ***: jane_dd
- Full Name ***: Jane Eyre
- Password ***: (redacted)
- Email ***: jane_dd@gmail.com
- Date of Birth ***: 02/06/2001
- Nationality**: United States (with a small USA flag icon)
- Agreement**: A checked checkbox next to the text "I agree to [Terms & Conditions](#)".
- Continue**: A blue button at the bottom.
- Sign in**: Text at the bottom right linking to existing account sign-in.

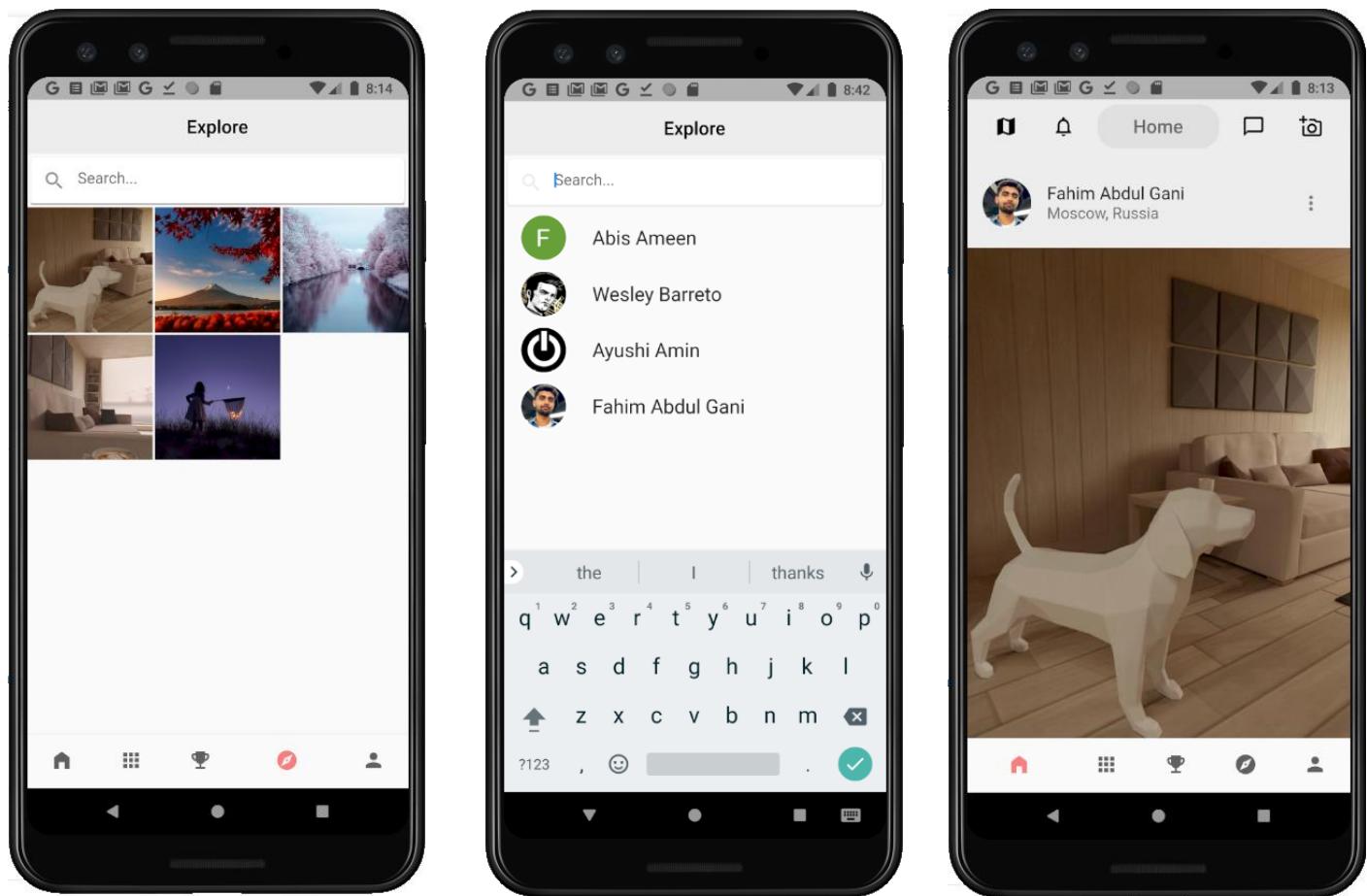
Users are to key in the information the application requires such as their choice of username, their full name, a password for at least 8 characters, date of birth and nationality of the user. They also have to agree to the terms and conditions only to be allowed to press continue. Otherwise, the continue button would not allow the user to proceed.

6.3 Sign In



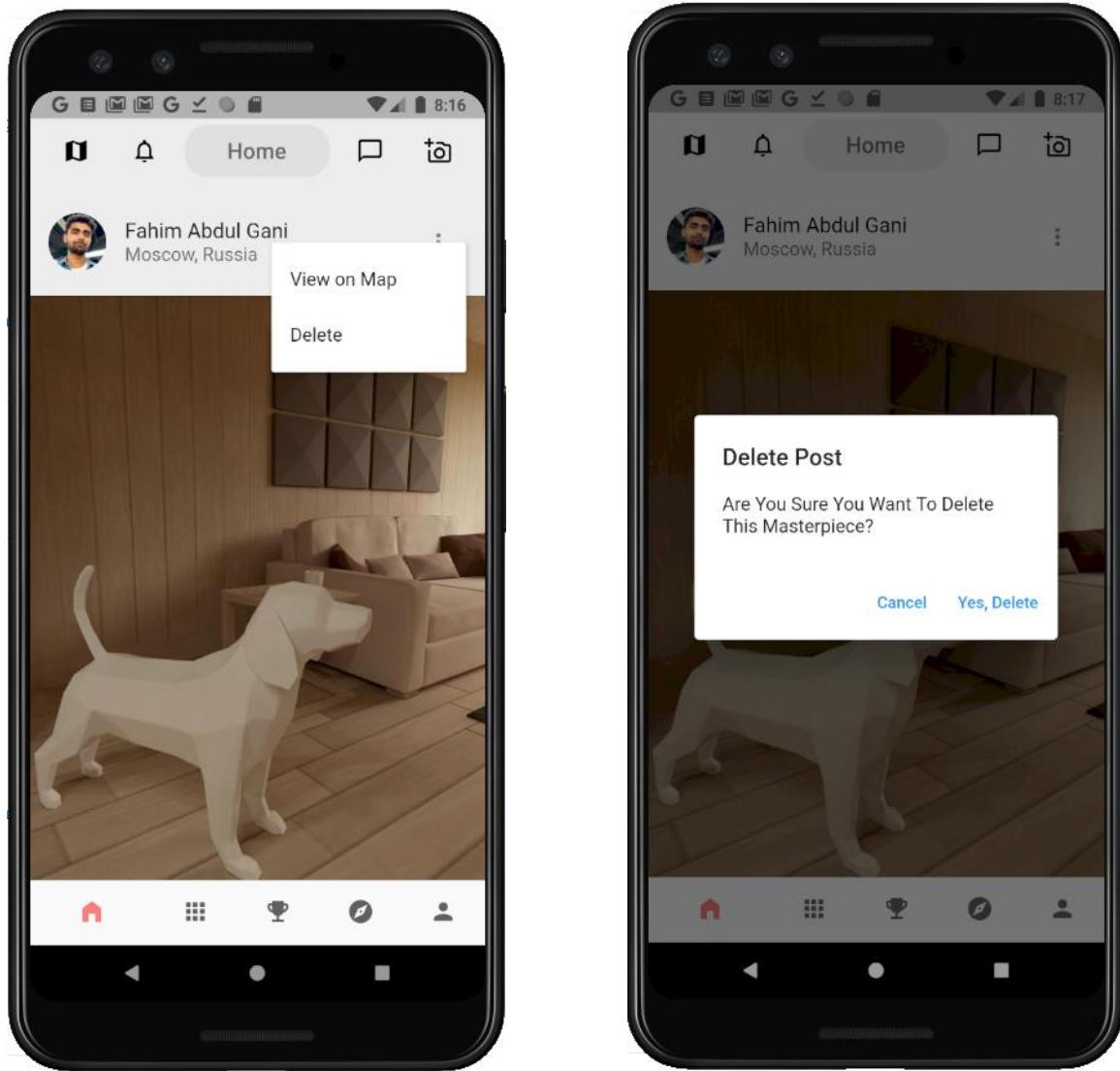
The user will now sign in with the credentials the system accepted. They can also sign in using Google authentication.

6.4 Follow User to Populate Home Page



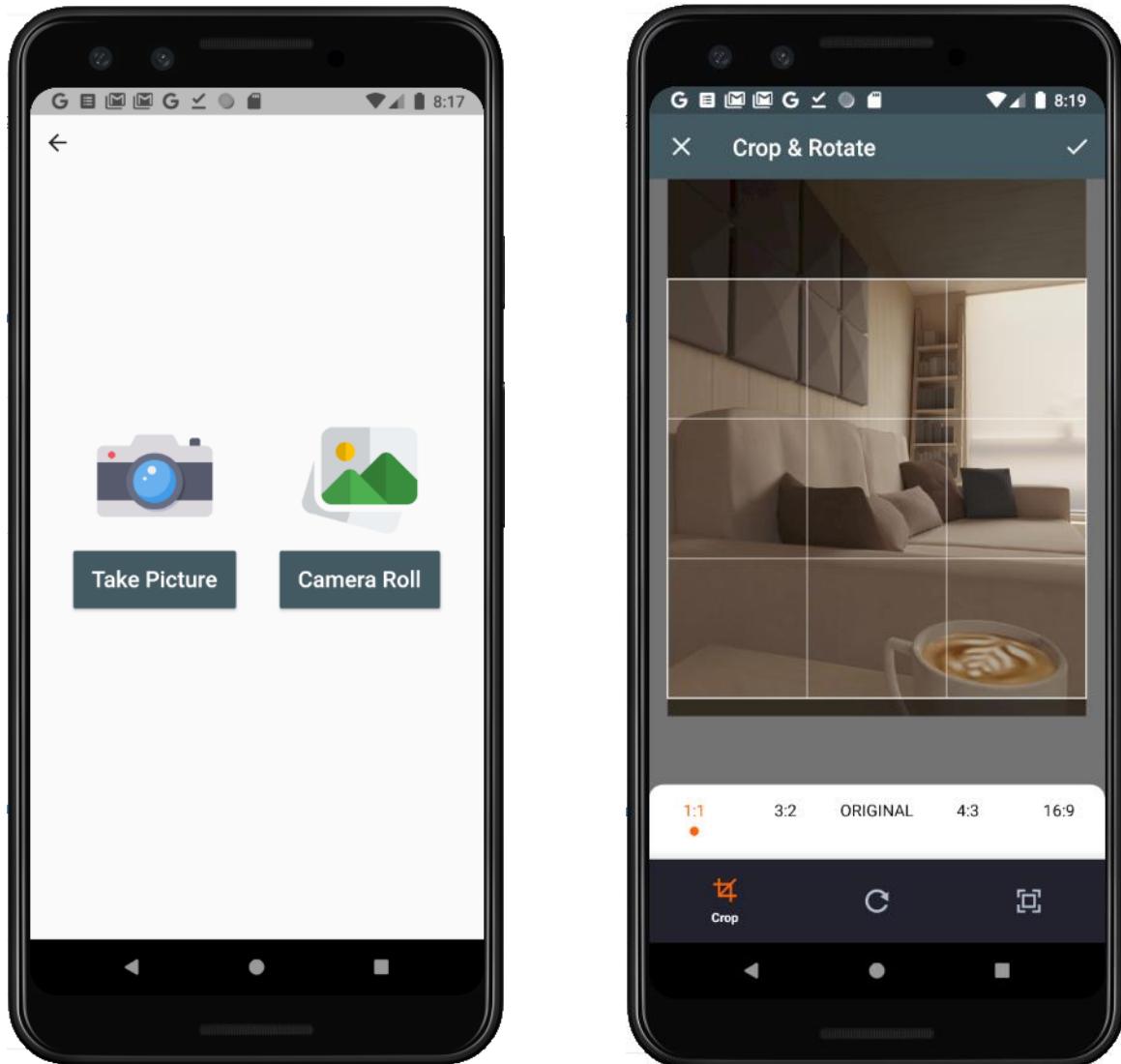
Once users sign in, the home page will direct them to follow users in order to populate their home page. Hence, the users can go to the explore screen and see the popular posts there and choose to anyone whose content they like. Once they do that, their home page will look like the following. The screen below showcases that process. From the left to the right.

6.5 Delete or Report a Post



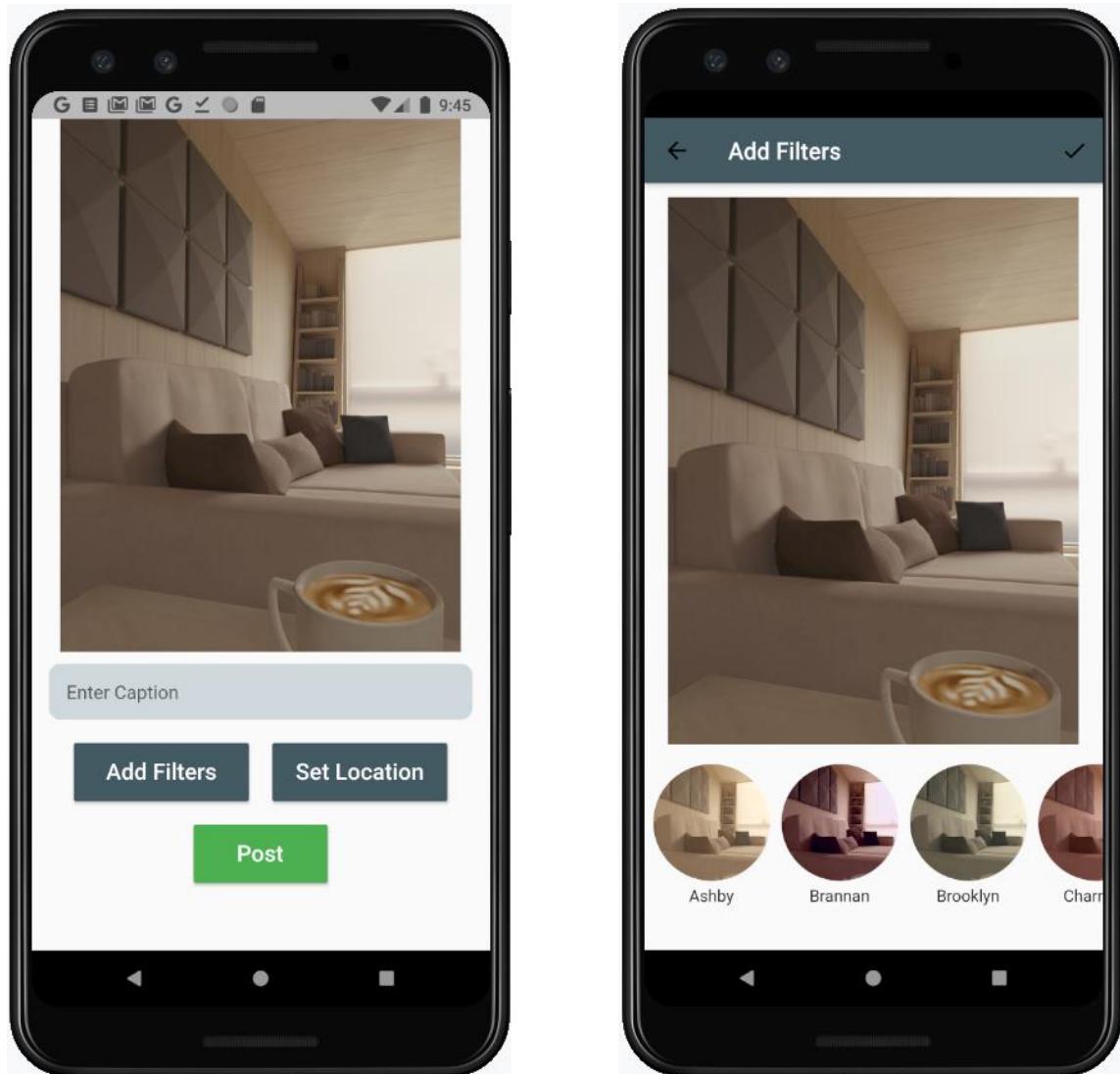
If a user wants to delete a post of theirs, they can click on the 3 dots to access the delete option. The post will only get deleted once the user confirms the options to delete. This feature is created so if users accidentally click on delete they can revert back.

6.6 User Chooses Picture to Upload



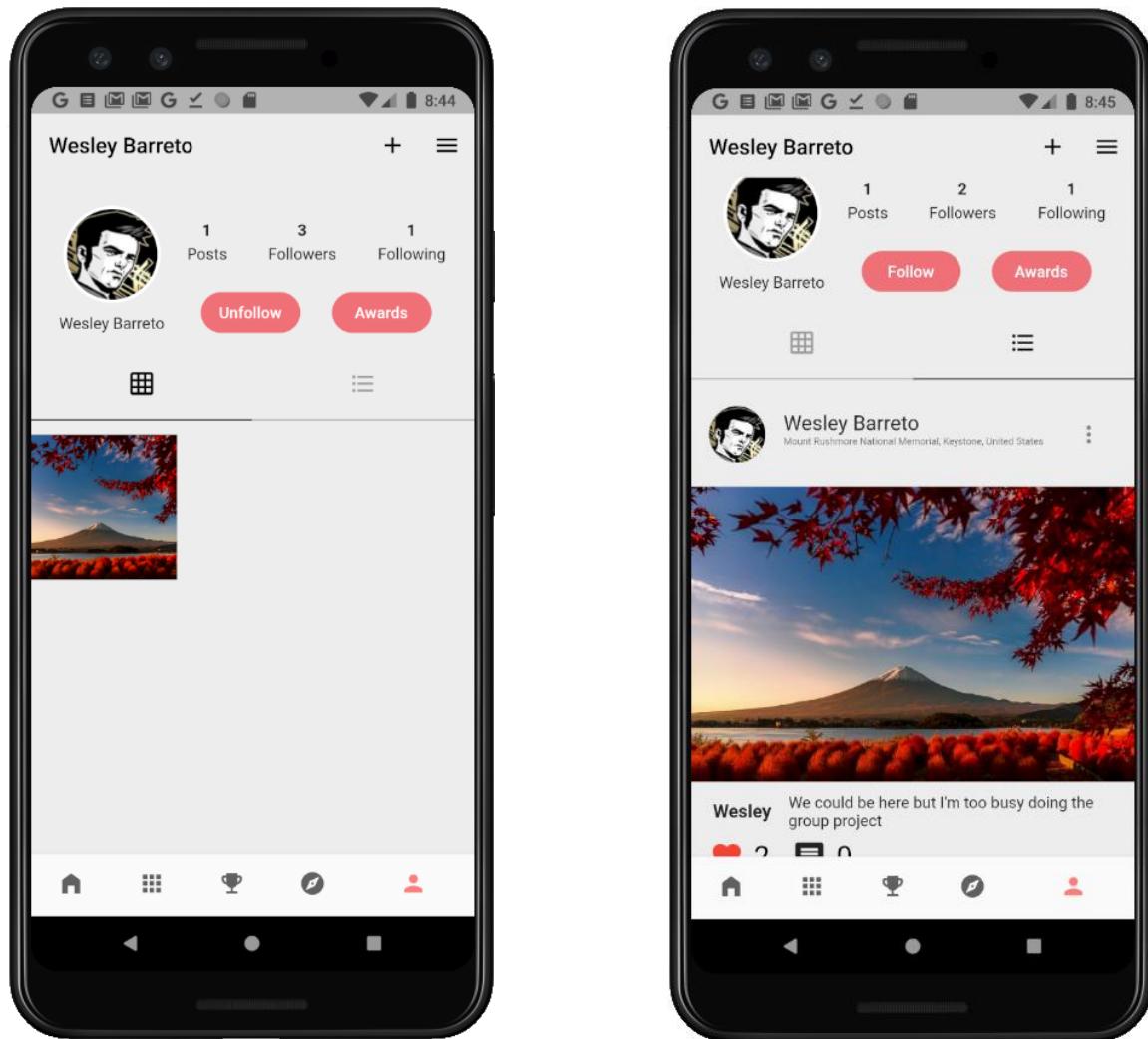
Users can choose to take a picture from within the application or upload a picture from their storage. Once they do they will be directed to the step on represented in the picture on the right where they have the option to crop or rotate. The ML Kit implemented within the application will search the picture for any humans as the system does not allow uploading human pictures. Take notice if the picture contains a human, the user will not be able to upload.

6.7 User Adds a Filter and Posts Picture



The user can add a filter to their photo of choice, add a caption and the location of the post if they want. Once the user customises their post, they can post the picture and it will be visible in their profile and to whoever follows them.

6.8 Access Your Own Profile



The user can access their profile by clicking on the most right icon in the navigation bar. Once they do that, they can see their posts, followers, following and have the option to view their posts in a grid of 3 x 3 or in list view which is the picture on the right that will show one post at a time.

7. Installation Guide

1. Install Flutter: <https://www.flutter.dev/docs/get-started/install>
2. Install Android Studio: <https://www.developer.android.com/studio/install>
3. Download and extract the Tasweer project zip file.
4. Get the SHA fingerprint keys by following these instructions below -
<https://developers.google.com/android/guides/client-auth>
5. Contact the group admin(s) and send the SHA key to get added to Firebase.
6. Follow the instructions below to get the latest dependencies -
<https://dart.dev/tools/pub/cmd/pub-get>
7. Run the main.dart file.

Project Evaluation

Table of Content

1. Organisation	70
1.1 Group Organization	70
1.2 Group Collaboration	71
1.3 Problems Faced in Stage 1	72
1.4 Problems Faced in Stage 2	72
1.5 Problems Faced in Stage 3	73
1.6 Original Plan VS Updated Plan	73
2. Implementation	74
2.1 Success of the Implementation Approach	Error! Bookmark not defined.
3. Language, Tools and Techniques Used	78
3.1 Tools Used	78
3.2 Basic DevOps, Version control and CI/CD	81
3.3 Requirements	Error! Bookmark not defined.
Usability	100
1. Introduction	102
1.1 Aim and Objectives	102
1.2 Scope	103
2. Overview	103
3. Test Plan	104
3.1 Objectives	Error! Bookmark not defined.
3.2 Participants	Error! Bookmark not defined.
3.3 Task Scenarios	105
3.4 Metrics	107
4. Testing Protocol	107

1. Organisation

1.1 Group Organization

Originally in stage 1, there were 8 members in our group and as per the project plan designed for stage 1, the following roles were assigned to each member as follows:

1. Abis – Cloud Engineer, Security Architect, Database Admin
2. Ammad – Back-end and Front-end Developer, Web Developer
3. Ayushi – Back-end and Front-end Developer, Web Developer
4. Fahim – Back-end and Front-end Developer, UI/UX Designer
5. Fayyaz – Cloud Engineer, Marketing Manager, Database Admin
6. Mayar – Back-end and Web Developer, UI/UX Designer
7. Wesley – Back-end Developer, Database Admin
8. Zainab – Front-end Developer, Database Admin

However, in stage 2 due to the exit of a member, the roles of each member were updated and changed to the following as shown below:

1. Abis – Back-end Developer, Database Admin
2. Ammad – Back-end Developer
3. Ayushi – Front-end Developer
4. Fahim – Front-end Developer, Web Developer
5. Fayyaz – Machine Learning Engineer
6. Mayar – Front-end Developer
7. Wesley – Front-end and Back-end Developer

The above decided roles turned out to be successful as each role was chosen based on personal skills, member's preference, and their experience in that area. The roles were also chosen to ensure maximum efficiency.

1.2 Group Collaboration

In the duration of the project, we collaborate with each team member through our meetings. In the meetings that were held every 5 days, we would discuss various aspects of the project such as how to proceed further, how to tackle problems or bugs within the application. We, also, would showcase our progress in every meeting, and each member would choose a task they want to work on. Members would provide their opinions and feedback after reviewing what each member did.

Mayar, Wesley and Abis alternated between taking the minutes of the meeting and uploaded it to GitHub, so that others can refer to any important points or cross check the tasks that they must work on.

We initially used Asana as a productivity tool, but due to multiple issues with registrations and integrating the workflow within the tool, we created “Trackers” on WhatsApp, where each member would write the tasks they are tackling and update their tracker once they have completed their task of choice. Since this tracker system is implemented within WhatsApp, a tool we use for our regular communication, it ensured that everyone would have access to this tool and would be able to use it easily.

For each stage, the team would be divided into sub-teams to tackle specific parts of the system. In stage 1, there was the Use Case Diagram team, consisting of Ayushi, Mayar and Zainab tackled creating the Use Case diagrams that represent the system. The UI/UX team, consisting of Abis, Fahim and Mayar, collaborated to create the interface for the system. The UML Diagram, Fahim and Wesley, who created sequence diagrams and data flow diagrams. Ayushi, Wesley and Zainab, collaborated and explained each aspect of the use case diagram by providing the use case specification. Abis worked on various risks that we might come across throughout the course of the project. Ammad broke down the number of hours each member would spend on the task that they are implementing and accumulated the cost spent on the project. Fayyaz drafted and prepared the project plan (gantt chart) and the various project decisions like the type of language we would use while developing the application, the team roles, etc.

While developing the application during stage 2, Ayushi, Mayar and Wesley had regular meetings to discuss which functionality each one would work on for the front-end part of the application and would also review each other's implementation of the functionality that they worked on previously. Fahim would be working on developing the company and app landing page, then joined the front-end team to complete more functionality of the system. Fayyaz trained the AI/ML model into recognizing human faces and preventing them from being uploaded onto the system. Abis and Ammad would also meet up on several occasions in order to discuss what tasks to implement in the back-end.

The division of the team into sub-teams to concentrate on specific tasks and work parallel to each other in order to complete as much as possible.

For stage 3, Ayushi and Mayar continued working on the remaining parts from the front-end aspect as well as tackled the report (and usability). Fahim and Wesley also worked on the front-end, where Wesley worked on the back-end alongside Abis.

1.3 Problems Faced in Stage 1

The team did not face any critical issues during stage 1.

1.4 Problems Faced in Stage 2

All the problems that rose up were solved by the collaboration of the team members. For example, during stage 1, the decision to use Flutter as the developing environment was due to the availability of a plugin that converts the Adobe XD prototype into functional code in Flutter. However, in stage 2, we have realised that the code that the plug-in generated was buggy, did not produce the output we wanted and was not efficient. This realisation came upon implementing multiple screens for the system, hence we had to re-do everything due to the level of bugs caused by the generated code.

Another problem that was faced towards the end of stage 2, was an issue of integrating the front-end components to the back-end. This was happening during the final week before the deadline, hence the team was broken down to ensure there were enough members working on the integration part as well as enough members working on the report, so that those two tasks were happening simultaneously without any issues. The majority of the problems were solved with the team collaboration and members using their expertise to solve any technical problems.

1.5 Problems Faced in Stage 3

There were countless minor bugs in the flutter code that caused the team to rebuild already implemented functionalities again. In addition, a Flutter update caused the dependencies used within the application to upgrade which lead to incompatibility with existing code.

1.6 Original Plan VS Updated Plan

We did not manage to stick to the original plan (refer to Appendix) designed in stage 1 due to the following reasons:

- The exit of a team member played an important role in refactoring our plan. Their tasks had to be split amongst other team members, which caused us to update the plan accordingly.
- The change in our technology also played a major role when we updated the Gantt Chart. The team decided to use Firebase instead of AWS for the back-end. Firebase uses NoSQL whereas AWS offers SQL databases. NoSQL databases provide a better real-time experience due to it being horizontally scalable which means that additional traffic can be managed through the connection to more servers to the database instead of increasing the power on CPU or RAM to match the traffic. In addition, the syntax used is less strict and database fields can be added on the go. Moreover, due to the nature of the data the system collects such as images posted by users, the various

metrics such as number of likes received on a post, etc. Those unstructured data are not predefined and are flexible hence they do not work well with SQL as it relies on a strict organization of a model of data (AltexSoft, 2020).

- Thus, most of the tasks which were related to AWS had to also be changed accordingly to match Firebase's requirements.

2. Implementation

As mentioned previously, the Gantt Chart had to undergo changes due to the exit of a team member and change in technology to be used for the system.

The Gantt Chart represents what was implemented during stage 2 and 3. The task owner in the Gantt Chart does not represent who tackled the task in the actual

During phase 1, which began in stage 2, the tasks that were completed were the majority of the “App Development” section of the Gantt Chart as seen in

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUCK DATE	DURATION	DEPENDENCIES
0.1	App Development					
0.1.1	Splash / Sign up / Login	Fahim	1/11/21	1/24/21	10	-
0.1.2	Home Screen	Ammad	1/11/21	1/24/21	10	-
0.1.3	Profile / Notification panel	Mayar	1/11/21	1/24/21	10	-
0.1.4	Game feed	Fahim	1/24/21	2/4/21	9	-
0.1.5	Communities template	Ammad	1/24/21	2/4/21	9	-
0.1.6	Post template	Mayar	1/24/21	2/4/21	9	-
0.1.7	Leaderboard and Progress template	Ayushi	1/11/21	1/24/21	10	-
0.1.8	Documentation	Ayushi	1/24/21	2/4/21	9	-
0.2	Real Time Database					
0.2.1	Prepare the datapoints	Wesley	1/11/21	1/15/21	5	-
0.2.2	Define JSON structure	Ammad	1/16/21	1/23/21	5	0.2.1
0.2.3	Define DB security rules	Abis	1/16/21	1/23/21	5	0.2.1
0.2.4	Documentation	Ayushi	1/24/21	2/1/21	6	0.2.2
0.3	Firebase (GCP)					
0.3.1	Set up real time DB	Abis	1/11/21	1/15/21	5	-
0.3.2	Set up authentication	Abis	1/15/21	1/18/21	2	-
0.3.3	Set up Google Storage (firebase)	Fayyaz	1/18/21	1/22/21	5	0.3.2
0.3.4	Set up webhosting (firebase)	Abis	1/24/21	1/28/21	4	0.3.3
0.3.5	Documentation	Mayar	2/1/21	2/3/21	3	-

Diagram 1: Phase 1 Gantt Chart

Phase 2 took place during stage 2 where one of the tasks completed included the redesigning the entire application. (see section: improvements to design) This change was brought up from the usability study conducted in stage 1, where some of the comments referenced that the user buttons were not clear and the user interface lacked colour. Thus in stage 2, the redesigning of the UI was made to ensure that the buttons are clearly defined and visible and there is contrast of colour between all the elements in every page of the interface.

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUUE DATE	DURATION	DEPENDENCIES
1.1	Machine Learning					
1.1.1	Test and scope custom models	Fayyaz	2/7/21	2/13/21	5	-
1.1.2	Firebase scoping of ML services	Fayyaz	2/7/21	2/13/21	5	-
1.1.3	Integeeration of Facial Recognition	Fayyaz	2/14/21	2/20/21	5	1.1.1
1.1.4	Integeeration of Landmark detection using ML	Ammad	2/20/21	2/27/21	5	1.1.3
1.1.5	Deploy and test the integerated models	Abis	2/27/21	3/6/21	5	1.1.4
1.1.6	Documentation	Wesley	2/27/21	3/6/21	5	1.1.4
1.2	App Development					
1.2.1	Feedback and Review	Fahim	2/7/21	2/13/21	5	-
1.2.2	Improvements and redisgns	Ammad	2/14/21	2/20/21	5	1.2.1
1.2.3	Integrations with RDS and testing	Mayar	2/20/21	2/27/21	5	1.2.2
1.2.4	Documentation	Ayushi	2/27/21	3/4/21	4	1.2.3
1.3	GPS and Geolocation					
1.3.1	Integrate API requests for Maps	Ammad	2/7/21	2/13/21	5	-
1.3.2	Integrate machine learning model for landmark detection	Fayyaz	2/14/21	2/17/21	3	1.3.1
1.3.3	Access Testing and Documentation	Ammad	2/17/21	2/19/21	3	1.3.2
1.4	API					
1.4.1	Scope and structure API calls	Fahim	2/14/21	2/17/21	3	-
1.4.2	Management and integeration	Wesley	2/21/21	2/23/21	2	1.4.2
1.4.3	Testing and optimization	Wesley	2/24/21	2/27/21	3	1.4.2
1.4.4	Documentation	Fahim	2/28/21	3/2/21	2	1.4.3

Diagram 2: Phase 2 Gantt Chart

During Stage 3, the remainder pages from the front-end were completed. The feedback was collected in the form of usability. Presentation and Marketing was unnecessary to complete during this stage.

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUUE DATE	DURATION	DEPENDENCIES
2.1	Security					
2.1.1	Configuring hashing functions	Abis	3/7/21	3/10/21	3	-
2.1.2	Setting up debugging keys	Abis	3/11/21	3/13/21	2	2.1.1
2.1.3	O-Auth configuration (tokens)	Fayyaz	3/14/21	3/20/21	5	2.1.2
2.1.4	DDOS protection and downtime monitor	Abis	3/7/21	3/20/21	10	-
2.1.5	Android pro-guard implementation	Abis	3/21/21	3/24/21	3	2.1.4
2.1.6	Documentation	Wesley	3/25/21	3/26/21	2	2.1.5
2.2	Feedback and Review					
2.2.1	Create the Beta app on Staging	Fahim	3/11/21	3/15/21	3	2.1.1
2.2.2	Share the app wiht user base	Ammad	3/16/21	3/17/21	2	2.2.1
2.2.3	Collect Feedback and Improvements	Mayar	3/18/21	3/20/21	2	2.2.2
2.2.4	Implementation of improvements	Ayushi	3/20/21	4/6/21	12	2.2.3
2.3	Presentation and Marketing					
2.3.1	Create a to market strategy	Fayyaz	3/7/21	3/10/21	3	-
23.2	Prepare a demo deck	Fayyaz	3/11/21	3/13/21	2	-
2.3.2	Create posters and Advertisments	Ayushi	3/14/21	3/20/21	5	-
2.3.2	Launch the App	Fahim	4/7/21	4/8/21	2	2.2.4

Diagram 3: Phase 3 Gantt Chart

2.1 Success of the Implementation Approach

Tasks could be shuffled easily, teams could be easily dissolved and new teams are created to tackle specific tasks.

To get a quicker release of a usable product to users and to increase quality while lowering costs we decided to work with the scrum framework. Scrum helped the development team to work concurrently instead of sequentially. Our programmers worked in a perpetual cycle, they did not wait for all information to be provided and everything to be perfectly clear before beginning to code. Throughout the project's lifetime, it is adaptable and changeable. The same is true of code checking, which is performed on a regular basis rather than at the end of the coding process. The same can be said for checking code, which we did on a regular basis rather than at the end when all coding is done.

We found that using this progressive strategy was very effective because we were able to fulfill all of our requirements on time, despite the fact that some tasks had to be moved around during tasks to allow members of the group to focus on other curriculum responsibilities. As a result, any issues we did find were addressed in the following sessions.

We deemed it was necessary to have the project management process going on within our team. Once done correctly, it improves the efficiency of any aspect of the company. It frees up the team to concentrate on the work that counts, rather than being distracted by assignments which go behind schedule or plans that lead to disaster. It equips them to generate results that have a direct effect on the company's bottom line.

Our project management was divided into the following stages:

1. Planning
2. Initiation
3. Execution
4. Monitoring
5. Closing

3. Language, Tools and Techniques Used

3.1 Tools Used

Programming Languages:

1. **Dart (Flutter)** – The choice of SDK used to develop the app was Flutter. Flutter was used as it uses Dart programming language which offers various libraries and frameworks that can help us build a multi-platform application with minimal development time and using just one source code.
2. **HTML** - We used HTML to build our website. Since it is the most easy to use, high browser support and SEO friendly. A major advantage of HTML is that it is free. We do not need any software for HTML, no plug-ins are needed for it.
3. **CSS** - We used CSS for describing the presentation of Web pages, including colors, layout, and fonts. CSS is independent of HTML and can be used with any XML-based markup language
4. **JavaScript** - HTML and CSS are languages that give structure and style to our web pages, JavaScript gives web pages interactive elements that engage a user. Also it was the best option since JavaScript is the only programming language native to the web browser.
5. **NoSQL** - We decided to go with NoSQL rather than MYSQL because NoSQL databases allow us to store data "as it is." Key value stores allow us to store simple data structures, while document NoSQL databases allow you to store a variety of flat or nested structures. Because of the schema agnostic nature of NoSQL databases, they are capable of managing any kind of change. In our case, NoSQL databases are often better suited to storing and modeling structured, semi-structured, and unstructured data in one database.

6. **JSON** - In relational database models, tables are used to store data in forms of rows and columns. Whereas in a non-relationship database model, there are no records or tables needed, the data is stored in the database in native types that correspond to the available JSON types. This is useful in our application as JSON uses objects (label-value pairs) and arrays (list of values), hence to refer to posts the data would consist of a list of values like: link to an image, the user who posted the picture, the likes on a post, the comments on a post. Firebase makes use of JSON (Snowflake, 2021).

IDEs Used:

1. **Visual Studio Code** - This was a natural choice as all the members have had experience working with VS Code. Visual Studio Code was also used as most of the programming languages mentioned above are supported by VS Code.
2. **Android Studio** - This IDE provides various emulators in different sizes that act as a virtual phone. These emulators were used for testing out the various functionalities of the application while implementing them. Due to the ease of integrating emulators in Android Studio, many members were using it as their primary IDE. It was also much easier to load and fire off the system from Android Studio rather than VS Code.

Version Control:

1. **GitHub** - Our group used GitHub as our version control environment. It was our main code base. Version control is a technique for keeping track of changes to documents and data so that we still know which version is the most recent. It also allows us to keep track of previous versions in case you need to see what has changed or restore a previous version. Dependency concerns have come to the forefront as a factor that can greatly affect a project. Whenever a new version of a file is added, the assigned reviewers and approvers will automatically see the latest version when they review it. There is no need for them to search in another database or scroll through the whole version history to find and select the newest file.

Other Tools:

- 1. Adobe XD** - Adobe XD was the application of choice to design the user interface. It was chosen mainly for the plugin "XD to Flutter" that allows the transformation of the prototype into functional code. The team agreed on this choice as it would save time to use a code that the plugin generates based on our prototype instead of writing it from scratch. Our entire system was designed in a single XD document as Adobe XD allows for multiple artboards to be created in a single file. Adobe XD produces files that are lightweight, so it was easy to exchange the interface between team members who would want to add more tweaks to the design. It also supports creating interactive prototypes where linking the pages and animating buttons or transitions was done with ease and it allowed us to easily visualize our ideas and how users can interact with our system. Sharing the prototype was easy, as Adobe XD allows for a link of the prototype to be generated and the prototype could be viewed on any web browser without any installations required. This was very helpful when conducting the usability study in stage 1 as participants had to interact with the prototype to fill out the questionnaires. There are many resources available online hence researching to create the interface was straightforward.

- 2. Google Docs and Sheets** - The team used Google Docs and Sheets as collaboration could be done easily on it. It allows multiple people to edit or write information simultaneously. There is an automatic save feature which guarantees that the files will never be subjected to loss, which is helpful in case a member's laptop crashes. Google docs allowed us to keep tracks of the various changes implemented on the document. In case someone deletes an important part of the document, it can be easily retrieved from the history. Members would share the document with each other, and this was helpful while drafting the stage report.

3. Zoom and MS Teams - Due to the ongoing pandemic, the team used Microsoft Teams and Zoom to hold meetings with each other or the team as a whole. This allowed us to have weekly meetings and sometimes hold meetings during the weekends. The members would share their screen to show what they have implemented, help others fix any bug issues or discuss any ideas which could help implement the app. Drafts between members were exchanged on Teams through the group channel. They were organized by categories to allow the member to keep track of the files that did not need to belong on GitHub.

3.2 Basic DevOps, Version control and CI/CD

Continuous integration (CI) is a software procedure in which code is committed to a shared repository on a regular basis. When a developer commits code more often, it discovers errors faster and decreases the amount of code he or she must debug when identifying the cause of an error. It is much easier to merge improvements from various members of a software development team when code is updated frequently.

GitHub runs the CI tests and includes the findings in the pull request, allowing us to see if the change in our branch causes an error. If all of the CI checks in a workflow pass, the improvements we pushed are ready for a team member to review or merge. If a test fails, it is likely that one of our improvements is to blame.

For us, Continuous Delivery is the next logical step after Continuous Integration. Our application is not only designed and checked each time a code update is pushed to the development environment, but it is also deployed on a regular basis. Also, Continuous Delivery tests the code automatically, but manually triggering the deployment requires human input.

4. Product

4.1 Requirements

The below table identifies the status of the functional and non-functional requirements that were designed for our system in Stage 1. They are color coded based on their completion status.

Status Legend –

Status	Color
Fully Implemented	Green
Partially Implemented	Yellow
For future development	Orange
Cancelled due to various reasons	Red

4.1.1 Functional Requirements –

F-UR 1	Registration	
Description: The system must allow the user to create a new account or log in to the system.		
F-UR 1-1	The user must be able to create a new account by providing: <ol style="list-style-type: none">1. Full name2. Email3. Date of Birth4. Username5. Password	Green
F-UR 1-1.1	The system must check for uniqueness of usernames.	Green

F-UR 1-2	The user must have an option to provide their phone number.	
F-UR 1-3	The user must have an option to set their profile picture.	
F-SR 1-4	The system must send an email to the registered email for verification.	
F-UR 1-5	An existing user must be able to sign into the platform by providing their email/username and password.	
F-UR 1-6	An existing user must be able to sign out of the platform.	
F-SR 1-7	The system should have a two-factor authentication process.	
F-SR 1-7.1	The user should be provided with a 4-digit code on the registered mobile number when logging into a new device if two-factor authentication is enabled on the user's account.	
F-SR 1-7.2	The user should be given a new code after 5 failed attempts.	
F-SR 1-7.3	The system must be able to resend a new code to the user on request.	

F-UR 1

The Google Authenticator dependency was used along with email sign in for stage 3. We did not add 2FA due to time constraints.

F-UR 2		Preferences
Description:		
	User must be able to choose and update their preferences	
F-UR 2-1	The user must be able to edit: a) Their username b) E-mail c) Profile picture d) Password.	Green
F-UR 2-2	The user must be able to choose from a list of themes and categories they are interested in.	Orange
F-UR 2-3	The user must be able to change their choice of categories and themes at any time.	Orange
F-SR 2-4	Users must be shown recommended pictures based on popular pictures and profiles.	Green
F-UR 2-5	The user must be able to customize their profile page	Orange
F-UR 2-5.1	The system must allow the users to select the type of account they want to create.	Green
F-SR 2-5.2	The system should have different default themes and layouts for the user's personal page for different types of accounts.	Green

F-UR 2-5.3	Users must be able to pin pictures in the highlight section of their profile.	
------------	--	--

F-UR 2

The user has the ability to change basic information about themselves and choose the type of account that they want to create. We decided to push the other sub-requirements for future versions as there is no scope for them in the current state of the application.

F-UR-3		Social Interaction
Description:		
	The system must allow the user to interact with other users and their posts.	
F-UR-3-1	Users must be able to follow other users.	
F-UR-3-1.11	If user A follows user B, then user A must be able to view: a) Pictures posted by user B b) The number of followers that user B has	
F-UR-3-2	Users must be able to unfollow other users.	
F-UR 3-3	Users must be able to view posts by other users.	
F-UR 3-3.1	The system should allow users to zoom in on posts.	
F-UR 3-3.2	Users must be able to like and comment on pictures of other users.	

F-UR 3-3.3	The system should not allow users to write disrespectful comments on posts.	
F-UR 3-3.4	Users must be able to report comments and posts to the platform manager for review.	
F-UR 3-4	Users could invite their friends from other social media platforms (Facebook, Google, Instagram etc.)	
F-UR 3-5	Users must be able to view notifications for recent activity.	

F-UR 3

Users can currently view posts uploaded by both themselves and other users in the timeline on the home screen. They can also comment on pictures, like the comments of other users and report them. In F-UR-3-1.2, users account types have been dissolved as a feature. It was viewed as an additional feature that is not crucial to the core of the application. (Refer to section: Interface Design section for a look for the user account types feature).

The system will not automatically check for disrespectful language in comments. This is because during implementation, it was not possible to find a suitable dataset containing a fully inclusive list of offensive language that the system could detect. Hence, a feature allowing users to report comments was created. Since language is all about nuances, having the users report comments that they feel are offensive has proved to generate much better results as well as allow for cultivation of an environment where users are looking out for each other by reporting such posts.

Push notifications are available from Firebase to notify users about any update in data that relates to them.

The ability to invite friends from other platforms such as Facebook, Google, etc. was not implemented.

F-UR-4		Posts
Description:		
FUR-4-1	The user must be able to post pictures on the platform	
FUR-4-2	The user must be able to give a caption to the picture.	
FUR-4-3	The system must be able to detect the presence of people in pictures and not allow them to be uploaded.	
FUR-4-3	The user must be able to delete a picture posted by them.	

FUR-4-4	In a community, users must be able to post pictures relevant to the community's themes/topics.	
F-UR 4-5	Users must be able to tag objects (animals, buildings, etc.) in their pictures.	
F-UR 4-5.1	The system should include an automated tagging feature to place a picture in a general broad category.	
F-UR 4-6	The system must be able to include GPS information about the picture posted by the user.	
F-UR 4-6.1	The user must be able to choose if they want to include GPS data in their picture.	
F-UR 4-6.2	The user must be able to view their posted pictures on a map, based on the location it was photographed.	
F-UR 4-7	The system must perform a reverse search on a picture to be posted to check for duplication.	
F-UR 4-8	Users must be able to add filters and edit photos within the app before posting them.	
F-UR 4-9	The system could include photo-manipulation services such as stamp overlays and GIF generation.	
F-UR 4-10	The mobile app must have support to take pictures within the app using the in-built camera feature.	

F-UR 4

Implementing an automated tagging feature would require a ML model which would take time. Reverse search, stamp overlays and GIF generation could be implemented in the future versions of the app.

F-UR 5		The Gaming Element
Description:		The user must be able to take part in weekly competitions generated by the system.
F-UR 5-1	The system must generate a topic relevant to that time or any other random theme/topic that the photography competition will be based on.	Orange
F-UR 5-2	Users must be able to post a maximum of 2 pictures a day in the competition section.	Green
F-UR 5-3	The user must be awarded XP for posted pictures.	Green
F-UR 5-4	Users must have an account level that is used to group them for competitions. Account level increases with XP.	Orange
F-UR 5-5	Users must get points for liking game posts to ensure that they like other users' posts.	Green
F-UR 5-5.1	Game posts must appear randomly in a user's home page/game feed.	Red
F-UR 5-5.2	Users must only be able to get points from a post if that post appeared randomly to another user on the home feed.	Red

F-UR 5-5.3	<p>Users must only get and receive points from game posts based on the following criteria:</p> <ul style="list-style-type: none"> a) Maximum number of game posts that can be liked is 50 posts per day. b) User is awarded with 0.1 points for liking a game post. c) 1 like = 1 point for recipient. <p>1 post = 5 points.</p>	
F-UR 5-6	<p>The user must be awarded trophies/badges for unlocked achievements.</p>	

F-UR 5

We decided that for this stage the game would be a timed competition and users will be placed on a leaderboard based on their cumulative score of points across the pictures that they post within the duration of the competition.

Users would get awards (badges) for finishing in the top 3 of a competition as well as for tasks such as liking another user's post, gaining 100 followers, etc.

F-UR 6	Weekly Reports
Description:	The system must generate and send weekly summary reports to users/admins.
F-UR 6-1	<p>A weekly report must contain the following:</p> <ul style="list-style-type: none"> a) Number of pictures posted.

F-UR 6-1.1	A user report must contain the following: a) Number of followers gained. b) Number of followers lost. c) Awards earned d) XP gained e) List of communities the user is a part of f) Total number of likes g) Hours spent on the app.	
F-UR 6-1.2	An admin report must contain the following: a) Number of new pictures uploaded b) Number of new members c) Number of users that left the community d) Number of reported posts	

F-UR 6

Weekly reports are generated from Google Firebase Analytics which can be viewed by anyone with admin access to the database.

F-UR 7 Communities and Search		
Description:		
a)	Users must be able to create and join communities.	
b)	Users must be able to search the system	
F-UR 7-1	The user creating a community must be given admin privileges to moderate its content.	
F-UR 7-1.1	Admins must be able to view the weekly admin reports generated by the system.	
F-UR 7-2	Users must be able to join communities of their choice.	

F-UR 7-3	The system must allow users to chat with other members in the communities they are a part of.	
F-UR 7-4	<p>Users must be able to search the system for:</p> <ul style="list-style-type: none"> a) Other users based on username or real name (if provided) b) Pictures based on location (if provided) 	

F-UR 7

Currently, the application only allows users to create and join communities and share posts in them. An admin user for communities will be implemented in a later release. The system does have an admin user that can see reported posts and take necessary action on them.

F-UR 8 Featured and Filtered Posts		
<p>Description:</p> <p>The system must have a featured section</p>		
F-UR 8-1	<p>The featured section must contain:</p> <ul style="list-style-type: none"> a) Pictures with most likes. b) Pictures from popular communities/users. 	
F-UR 8-1.1	<p>The featured section should be updated regularly.</p>	

F-UR 8-2	<p>The system must provide the user the ability to filter posts:</p> <ul style="list-style-type: none"> a) Home Page: <ul style="list-style-type: none"> • Posts from people you follow • Posts from the communities that the user is a part of • Hottest Posts • Recent Posts • Random Posts (This filter will account for user points and is enabled by default) b) Explore Page: <ul style="list-style-type: none"> • Featured Posts (Default) • Recent Posts (Hot posts posted in the past 24 hrs). 	
----------	---	--

F-UR 8

The Explore page currently contains posts from all users in the system ordered in descending order by their number of likes. Filtering posts can be done in a future release.

F-UR 9 The Map View	
	<p>Description:</p> <p>The system should have a map view of all the photos taken by the user.</p>
F-UR 9-1	<p>Users must be able to zoom and pan across the map to view pictures they have taken around the world.</p>
F-UR 9-2	<p>Users must be able to view all pictures taken in a specific location by clicking on the grouped pictures in the map view.</p>

F-UR 9

The system uses the Google Maps API that allows users to set a location of their choice while posting a picture and enables them to view images that have been posted on a map.

4.1.2 Non-Functional Requirements –

Security	Description	
NF-SR 1-1	The system must store user login info securely.	
NF-SR 1-2	The system must store all user data securely.	
NF-SR 1-3	The system must encrypt all HTTPS traffic and API calls with SSL/TLS.	
NF-SR 1-4	The system must auto-update all existing software used to build the system.	
NF-SR 1-5	The system should route traffic in accordance with AWS security protocols.	
NF-SR 1-6	The system must require the user to give a strong password (A-Z, a-z, special characters and numbers).	
NF-SR 1-7	The system must provide the user with a CAPTCHA while signing up.	
NF-SR 1-8	The system must use the Android Application Sandbox, which isolates user's app data and code execution from other apps.	

Hardware	Description

NF-SR 2	The system must be hosted on the AWS cloud with resources such as EC2 instance, S3 bucket, RDS instance and security groups.	
---------	--	--

The technology has staged from stage 1 to stage 2 and hence this functional requirement no longer applies to our system.

Usability	Description	
NF-SR 3-1	The system must be easy to use.	
NF-SR 3-2	The system's UI must be responsive and fast.	
NF-SR 3-3	Important UI elements must be clearly visible and easy to understand.	
NF-SR 3-4	The system UI must use familiar terms and styles potentially known to the user.	
NF-SR 3-5	The system could have support for dark-mode.	
NF-SR 3-6	The user must be able to easily navigate through different sections of the app.	

Performance	Description	
NF-SR 4-1	The system must have a quick search system.	
NF-SR 4-2	The system must be able to perform fluidly in times of high user traffic.	

Legislation	Description	
NF-SR 5	The system must follow GDPR guidelines.	

Software	Description	
NF-SR 6-1	The system should have support for desktop, Android, Apple and Windows devices.	
NF-SR 6-2	The website must be able to run across all widely supported web browsers.	
NF-SR 6-3	The system could include functionality to improve accessibility for disabled users (colorblind support, support for screen readers).	

Scalability	Description	
NF-SR 7	The system must initially support 100 users and should be scalable.	

4.2 What is Special About Our Product

1. Face Detection

With ML Kit's face detection API, we can detect faces in an image, identify key facial features, and get the contours of detected faces.

With face detection, we can get the information we need to perform tasks like embellishing selfies and portraits, or generating avatars from a user's photo.

Because ML Kit can perform face detection in real time, we can use it in applications like video chat or games that respond to the player's expressions.

When we use the model , it detects the face through a couple of methods. The first one being facial landmarks with each part of the face including eye, mouth, etc and various other feature probabilities like smiling and open eyes.

The second method uses face contour detection, it gets a list of points for each facial feature that was detected. These points represent the shape of the feature. The following image illustrates how these points map to a face.

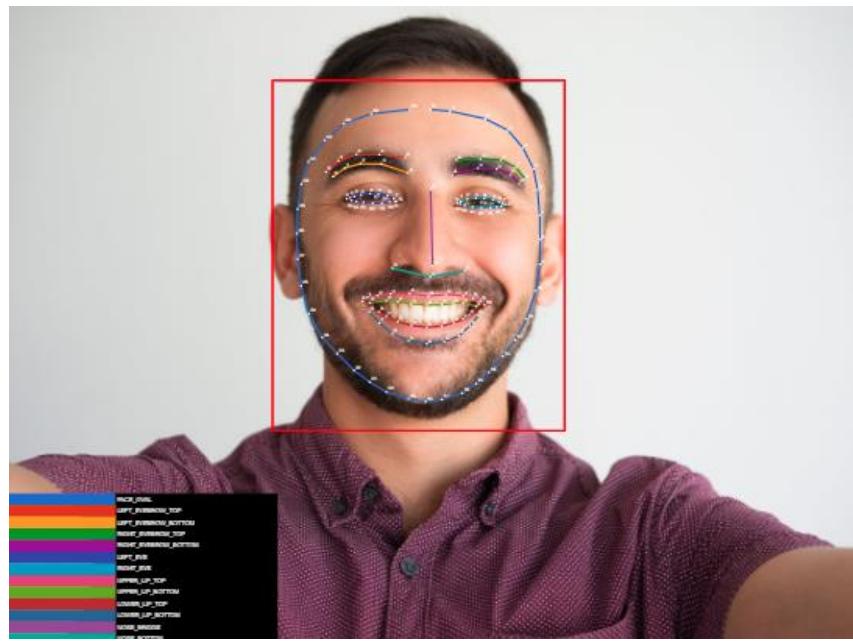


Figure 30: The ML Kit's Face in practice: scanning human face.

Facial feature contours	
Nose bridge	(505.149811, 221.201797), (506.987122, 313.285919)
Left eye	(404.642029, 232.854431), (408.527283, 231.366623), (413.565796, 229.427856), (421.378296, 226.967682), (432.598755, 225.434143), (442.953064, 226.089508), (453.899811, 228.594818), (461.516418, 232.650467), (465.069580, 235.600845), (462.170410, 236.316147), (456.233643, 236.891602), (446.363922, 237.966888), (435.698914, 238.149323), (424.320740, 237.235168), (416.037720, 236.012115), (409.983459, 234.870300)
Top of upper lip	(421.662048, 354.520813), (428.103882, 349.694061), (440.847595, 348.048737), (456.549988, 346.295532), (480.526489, 346.089294), (503.375702, 349.470459), (525.624634, 347.352783), (547.371155, 349.091980), (560.082031, 351.693268), (570.226685, 354.210175), (575.305420, 359.257751)
(etc.)	

Figure 31: The ML Kit's Face in practice: information gathered from the scan.

The top capabilities of this kit is:

- Recognize and locate facial features
- Get the contours of facial features
- Recognize facial expressions
- Track faces across video frames
- Process video frames in real time

2. Geo-Tagging

We are using Cloud based map customization and management which allows users to location tag their posts. We use Google Maps API to render a slate of the map and provide customization on top of it that allows users to upload their locations of choice.

4.3 Robustness of Final System and Bugs/Constraints

Our system uses Google's Cloud Firestore database at the backend and Google Cloud Storage for storing the images posted by our users. Cloud firestore has a free plan for using their services which includes:

- 1 GiB of Stored Data
- 10GiB/month of Network Egress
- 50K/day document reads
- 20K/day document writes
- 20K/day document deletes

Cloud Storage has the following services included in it's free plan:

- 5GB of storage capacity
- 1GB/day downloaded data
- 20K/day upload operations
- 50K/day download operations

Usability

Table of Content

1. Introduction	102
1.1 Aim and Objectives	102
1.2 Scope	103
2. Overview	103
2.1 Test Plan	103
2.2 Test Protocol	104
2.3 Mockup Results	104
2.4 Conclusion	104
2.5 Appendices	104
3. Test Plan	104
3.1 Objectives	107
3.2 Participants	107
3.3 Task Scenarios	105
3.4 Metrics	107
4. Testing Protocol	107
4.3 Questionnaires	109
4.3.1 Pre-Questionnaire Results	120
4.3.2 Task Scenarios Results	122
4.3.3 Post Questionnaire Results	129
5. Future Changes to the System	131

1. Introduction

1.1 Aim and Objectives

This document describes the final usability study conducted on our application “Tasweer”. It provides a detailed overview of the testing plan and the testing protocol implemented in this study.

This system can be used by people interested in using photo-sharing applications. Additionally, one of Tasweer’s main features is that it allows users to receive a gamified experience by performing simple tasks such as posting pictures, liking and commenting on them, etc. This expands the scope of the application to a much wider audience, who are interested in playing games or anyone who is looking to showcase their work to other users.

The aims of conducting this usability study is to see how easily the participants can interact with our application, how intuitive the design is and to see if they enjoy their experience while using the application.

The intended readers of this document are:

1. Rick Freedman, our client, and members that represent the client, to be able to examine the stage at which the application is on.
2. Dr. Talal Shaikh, the line manager of our team, who is to provide feedback for us.
3. Dr. Abrar Ullah, who will tackle the assessment of the document.
4. The developers who are taking part in this application.

1.2 Scope

The main purpose of the system is providing a user-friendly platform for people to showcase their work in the form of pictures, receive likes and feedback in the form of comments on it. The system also allows users to receive a gamified experience.

The user gains points by interacting with posts by liking and commenting on other posts and by receiving likes and comments on their posts well. They will be able to track the metrics on their posts, view the various posts present on the explore screen.

In addition, the system does not allow for human photos to be uploaded and databases that will be used will contain personal information such as email address, details of the photograph, etc. of each user who are currently registered onto the application.

2. Overview

2.1 Test Plan

This section gives a brief overview of how and on who the final application will be tested on. It also gives a detailed explanation of the various kinds of data to be collected from the study. The section also provides the reader with a set of aims that our team wishes to achieve from this test.

2.2 Test Protocol

This section provides a layout of how the test is to be performed on the users taking part in the study. From this study, we will receive qualitative and quantitative data which will help us analyze the data and draw conclusions.

2.3 Mockup Results

Based on the quantitative and qualitative data received, we can summarize the suggestions and make the recommended changes to our application.

2.4 Conclusion

This section provides the reader with a conclusion drawn from the questionnaires, feedbacks, and the suggested changes to be made in our application.

2.5 Appendices

This section consists of the consent form that is to be filled by the participant before taking part in the study. The pre and post questionnaires are included in this section as well.

3. Test Plan

3.1 Objectives

The objective of this test is to understand if the participants find it easy to navigate through the app and interact with it. The questionnaires are designed in a way to collect feedback to confirm if the participants can interact with the application's core application. With the feedback that we gain, we aim to recognize where we can improve the application and view any possible changes that could be implemented in the future.

Aim:

- To collect quantitative and qualitative data from the questionnaires that the users fill out.

- To recognize any issue with our app design and usability of the interface.
- To use the collected feedback and make improvements to the app in the future.

3.2 Participants

- The participants of this test will be students/staff of Heriot Watt University who are aged 18 and above.
- A group of 6-8 participants would be an ideal number for the usability study.
- Ideally, the audience would be participants who have a background with photo sharing applications. Additionally, if the participants have knowledge of games that would be valuable.

3.3 Task Scenarios

Questions	Related Requirements
1. Sign in screen	F-UR 1-5 An existing user must be able to sign into the platform by providing their email/username and password.
2. Explore screen	F-UR 3-1 Users must be able to follow other users. F-UR 7-4 Users must be able to search the system for other users based on username or real name (if provided). F-SR 2-4 Users must be shown recommended pictures based on popular pictures and profiles.
3. Home screen	F-UR 8-2 The system must provide the user with posts from people you follow and hottest posts. F-UR 3-3.2 Users must be able to like and comment on pictures of other users. F-UR 3-3.4 Users must be able to report comments and posts to the platform manager for review. F-UR 4-6 The system must be able to include GPS information about the picture posted by the user. F-UR 4-6.1 The user must be able to choose if they want to include GPS data in their picture.

	<p>F-UR 4-6.2 The user must be able to view their posted pictures on a map, based on the location it was photographed.</p> <p>F-UR 4-8 Users must be able to add filters and edit photos within the app before posting them.</p> <p>F-UR 4-10 The mobile app must have support to take pictures within the app using the in-built camera feature.</p>
4. Leaderboard screen	<p>F-UR 5 For the tasks mentioned under this screen, F-UR 5 was tested out in general. No specific functional requirement was tested out.</p>
5. Profile screen	<p>F-UR 3-2 Users must be able to unfollow other users.</p> <p>F-UR 4-3 The user must be able to delete a picture posted by them.</p> <p>F-UR 1-6 An existing user must be able to sign out of the platform.</p> <p>F-UR 3 F-UR 3 is tested out for the task “View posts in single post mode instead of grid view.”. In this task, the user interacts with the single/grid view feature mentioned in the page.</p>

The test scenarios will be measured on a likert scale that ranges from 1 to 5, with 1 being Very Hard and 5 being Very Easy.

- 1 - **Very Hard**: The participant was unable to complete the task due to great difficulty faced when interacting with the system.
- 2 - **Hard**: The participant completed the task albeit with hardship.
- 3 - **Neutral**: The participant faced some issues but successfully completed the task.
- 4 - **Easy**: The participant completed the task but stumbled upon minor problems.
- 5 - **Very Easy**: The participant completed the task with no problem.

Given the use of a likert scale, the participants will have to pinpoint how they felt their interaction with the system had gone. If they choose 5 for a given scenario, which is “Very Easy”, then that means that they did not face any issues while using the application, found it very usable and has no requests to change anything within the application. Hence, said scenario is user-friendly. Whereas, if they choose 1 for a

given scenario, which is “Very Hard”, then that means there are major issues with the design that lowers the usability.

Qualitative data is collected as well in the questionnaire. This comes in the form of feedback and questions about the user preferences such as which section of the application was their favourite. From that data we can draw conclusions on what was the user’s favourite and for what reasons to be able to understand how we can improve the system.

3.4 Metrics

1. Participants task ratings (Likert Scale)
2. Participants interest in the app (Qualitative Feedback)
3. User Experience Feedback (Likert Scale + Qualitative Feedback)

4. Testing Protocol

4.1 Aim

The aim of this test is to have you interact with Tasweer, our application, that takes a different approach to photo sharing by including a gaming experience. As Tasweer caters to individuals of many interests, we have many features available for that purpose. The gamification part, for example, is meant to cater to people who want a more exciting experience from sharing their photos by taking part in weekly competitions. The interface you are about to see is the mockup of our app and we intend to improve it from the feedback we will receive today.

4.2 Introduction

Before interacting with the prototype, I will ask you to kindly fill out a pre-questionnaire to grasp your level of familiarity with photo sharing applications. After that, I will ask you to interact with the prototype of our application, Tasweer, given certain scenarios. You will then be prompted with questions based on the given scenarios, answering on a scale of 1 to 5, and providing suggestions for the pages you have interacted with if you wish to do so. All the details entered will be stored, protected and anonymized and will not affect you in any way.

Finally, a post-questionnaire will be provided for you to complete. The purpose is to understand your impression of the application and the experience you have received interacting with it in the duration of filling out the questionnaires. All questions asked in the duration of this test will not be traced back to you, and whatever way you have chosen to answer them, will be respected as they are your own interpretation of the application and your own personal experience. We hope you will be as honest as possible as all feedback will be considered equally for the betterment of the design and the user experience. The test can be stopped at any time.

Link to Mock-Up Video from Stage 1: <https://youtu.be/bJf325ROaWk>

4.3 Questionnaires

4/2/2021

TASWEER USABILITY

TASWEER USABILITY

Welcome to the Tasweer application Usability Questionnaire!

To take part in the experiment you can begin by filling out the consent form which is on the next page.

* Required

Tasweer
Heriot-Watt University
Consent to Act in an Experimental Study as a Subject

Principal Investigator:

- 1) Abis Ali (aa337@hw.ac.uk)
- 2) Ammad Ahmed (aa349@hw.ac.uk)
- 3) Ayushi Amin (aa497@hw.ac.uk)
- 4) Fahim Abdul Gani (fa97@hw.ac.uk)
- 5) Fayyaz Ameen Mohammed (fa83@hw.ac.uk)
- 6) Mayar Abdalla Mohamed Abdellatif Ahmed (ma470@hw.ac.uk)
- 7) Wesley Barreto (wgb1@hw.ac.uk)

Line Manager: Mr. Talal Shaikh (T.A.G.Shaikh@hw.ac.uk)

Description: The purpose of this study is to understand how easy our system is to users and collect data from the questionnaire to gather an understanding of how users interacted with our application and their general opinion on it.

CONSENT
FORM FOR
USABILITY
EXPERIMENTS:

All personal information obtained is safely stored in a secure database and will be kept confidential. Little to no risks are involved to participate in this study. Participating will not effect the results of the course you are pursuing (if you are a student of Heriot-Watt University) or impact the relationship you have with the university in any way. You are free to refuse to take part in this usability study and can withdraw from this study at any time, should you choose to participate. Such a decision taken by you would in no way negatively effect your standing with the university.

Voluntary consent: I certify that I have read and understood the above contents. I understand that the team is open to any questions relevant to this usability study and is ready to answer them. By providing my signature below, I confirm that I have volunteered to participate in this study and have agreed to the publication of my responses and feedback for scientific purposes, as long as my identity is not disclosed.

Certification by the investigator: I certify that I have clarified the essence of the study as well as its motivations, potential benefits and possible risks in the case of participating in this usability study to the above person. I have also answered any questions that have been asked and proceeded with this study only after having received the signature, giving consent for participation.

1. Signature (Name/Initials): *

2. Today's Date *

Example: January 7, 2019

3. Occupation *

Mark only one oval.

Student

Staff

Pre-Questionnaire

Hi! Kindly fill the pre-questionnaire for the Tasweer application

4. Age: *

Mark only one oval.

18-20

21-25

26-33

34-41

42-49

50+

5. Gender: *

Mark only one oval.

- Male
- Female
- Prefer not to say

6. How often do you use social media? *

Mark only one oval.

- Never
- Rarely
- Sometimes
- Often
- Always

7. Which social media platform do you use? *

Check all that apply.

- Instagram
- Facebook
- Snapchat
- Twitter
- LinkedIn
- Reddit
- Tumblr
- I don't use social media

Task

Scenarios

Please follow one of the two options to complete the survey:

1. Please click on the following link to download the app's APK (please allow "Installation From Unknown Sources" in settings before installing the APK). Since this is a debug apk, it works only on android emulators at the moment.

To use the APK on an android emulator, download android studio, download and set up an android virtual device from the AVD manager. Once the emulator is running, drag and drop the APK from your file explorer into the emulator. You should be able to find the APK in the app drawer (app section of your emulator).

Links: https://drive.google.com/file/d/14KLx8PNy3vA466_Ouj5mcPnSzAnMbKHN/view?usp=sharing

2. Or contact the person who sent you the link to demo the app using Microsoft Teams. Thanks!

The following sections will include questions where you had tasks to complete on each screen in the application. Fill your answers appropriately.

Kindly perform the following tasks that are available on the sign in screen:

Task 1 – Sign In

- a) Please sign in using your Google account.

8. Were you able to sign in? *

Mark only one oval.

- Yes
 No

9. How easy was it for you? *

Mark only one oval.

1 2 3 4 5

Very Hard Very Easy

10. Any suggestions for the Sign in page? (Optional)

Task 2 – Explore Screen

Kindly perform the following tasks that are available on the explore screen:

- a) Search for "Fahim" and follow him.

11. Were you able to find "Fahim" and follow him? *

Mark only one oval.

- Yes
 No

12. How easy was it for you to find and follow "Fahim"? *

Mark only one oval.

1 2 3 4 5

Very Hard Very Easy

13. Were you able to find popular posts from people you don't follow? *

Mark only one oval.

- Yes
 No

14. Any suggestions for the Explore page? (Optional)

Task 3 – Home
Screen

Kindly perform the following tasks that are available on the home screen:

- a) Like a post.
- b) Comment on a post.
- c) Upload a picture and customize it e.g. add a location to it, select a filter of your choice, etc.
- d) Check map view and see if the photo you uploaded is on the map.
- e) Upload a picture with a human in it.

15. Please describe what changes happened to your home screen after following "Fahim". *

16. How easy was it for you to like and comment on a post? *

Mark only one oval.

1 2 3 4 5

Very Hard Very Easy

17. How easy was it for you to report a post? *

Mark only one oval.

1 2 3 4 5

Very Hard Very Easy

18. What additional option(s) did you add to your uploaded post? *

Check all that apply.

- Location
- Filter
- Crop to aspect ratio
- Rotate
- Nothing
- I was not able to upload a picture

19. Were you able to view your post on the home page? *

Mark only one oval.

- Yes
- No

20. How easy was it to search for your post on the map? *

Mark only one oval.

1 2 3 4 5

Very Hard Very Easy

21. Any suggestions for the Home page? (Optional)

Task 4 –
Leaderboard

Kindly perform the following tasks that are available on the leaderboard screen:

- a) View the leaderboard.
- b) Find your name on the leaderboard.

22. How easy was the leaderboard to interpret? *

Mark only one oval.

1 2 3 4 5

Very Hard Very Easy

23. Were you able to locate your name on your leaderboard?

Mark only one oval.

Yes

No

Task 5 – Profile

Kindly perform the following tasks that are available on the profile screen:

- a) Unfollow "Fahim".
- b) View posts in single post mode instead of grid view.
- c) Delete your post.
- d) Sign out from your account.

24. Were you able to unfollow "Fahim"? *

Mark only one oval.

Yes

No

25. How easy was it to view posts in single post (list view) mode? *

Mark only one oval.

1 2 3 4 5

Very Hard Very Easy

26. How easy did you find it to delete a post? *

Mark only one oval.

1 2 3 4 5

Very Hard Very Easy

27. How easy was it for you to sign out from your account? *

Mark only one oval.

1 2 3 4 5

Very Hard Very Easy

28. Any suggestions for the Profile page? (Optional)

Post-Questionnaire

Thank you for taking part in the experiment. Kindly fill in the post- questionnaire.

29. I found the application unnecessarily complex. *

Mark only one oval.

1 2 3 4 5

Strongly Disagree Strongly Agree

30. I was easily able to navigate my way through the app. *

Mark only one oval.

1 2 3 4 5

Strongly Disagree Strongly Agree

31. The app's design is intuitive and straightforward. *

Mark only one oval.

1 2 3 4 5

Strongly Disagree Strongly Agree

32. I would use this app because of the service it provides. *

Mark only one oval.

1 2 3 4 5

Strongly Disagree Strongly Agree

33. Would you recommend this application? *

Mark only one oval.

1 2 3 4 5

Highly Not Recommend Highly Recommend

34. Overall, any suggestions you would like to give?

35. In the application, which section did you like the best? *

36. In the application, which section did you like the least?

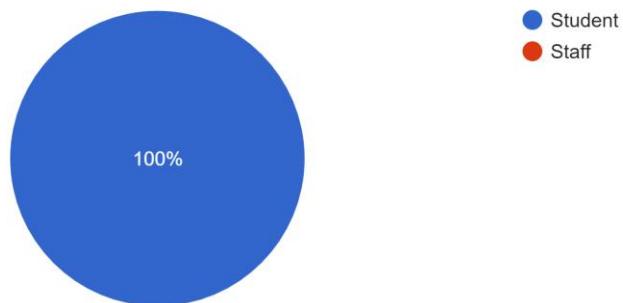
This content is neither created nor endorsed by Google.

Google Forms

4.3.1 Pre-Questionnaire Results

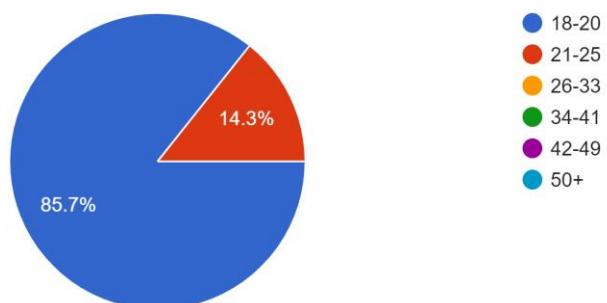
Occupation

7 responses



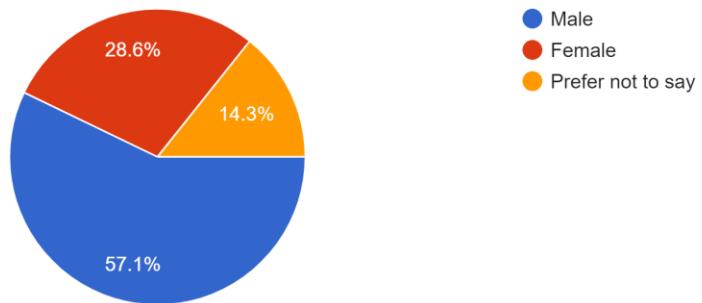
Age:

7 responses



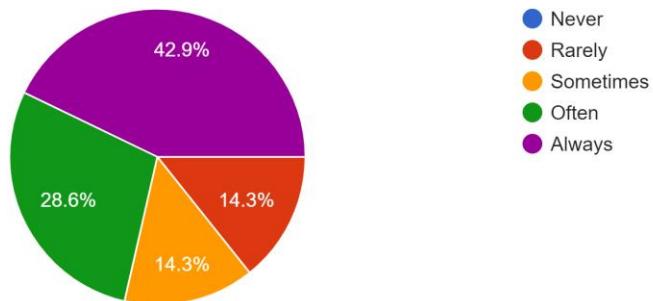
Gender:

7 responses



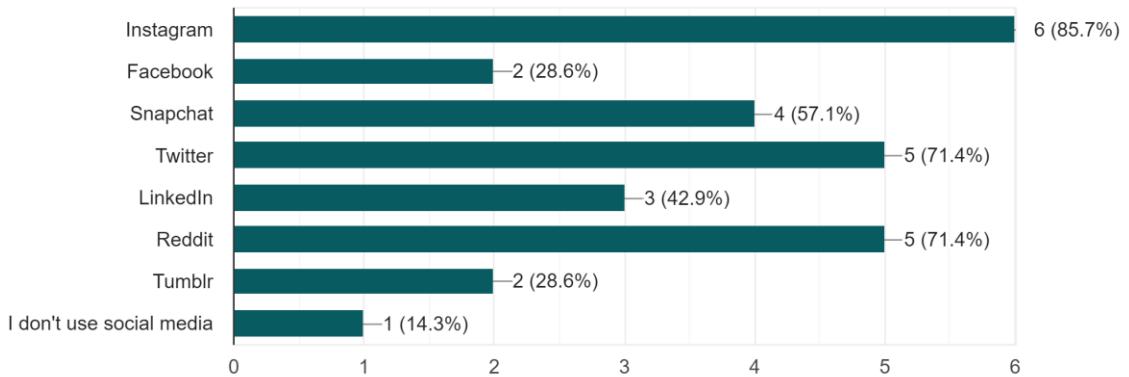
How often do you use social media?

7 responses



Which social media platform do you use?

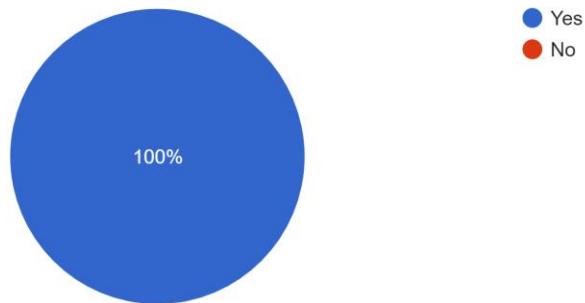
7 responses



4.3.2 Task Scenarios Results

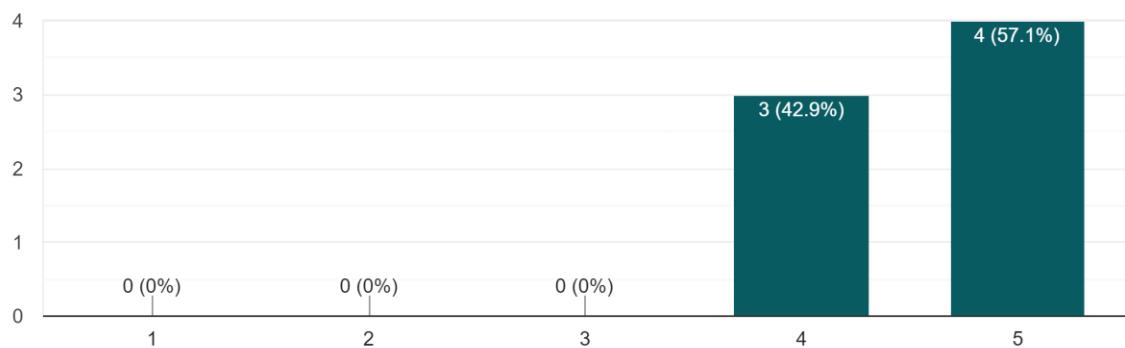
Were you able to sign in?

7 responses



How easy was it for you?

7 responses

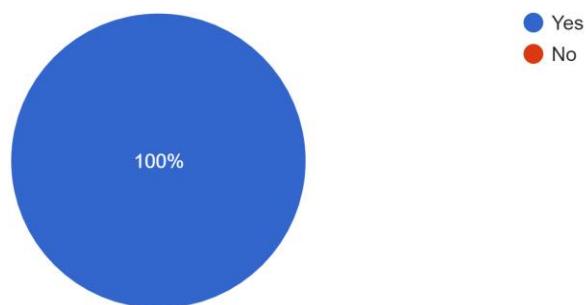


Any suggestions for the Sign in page? (Optional)

You can allow sign in with email.

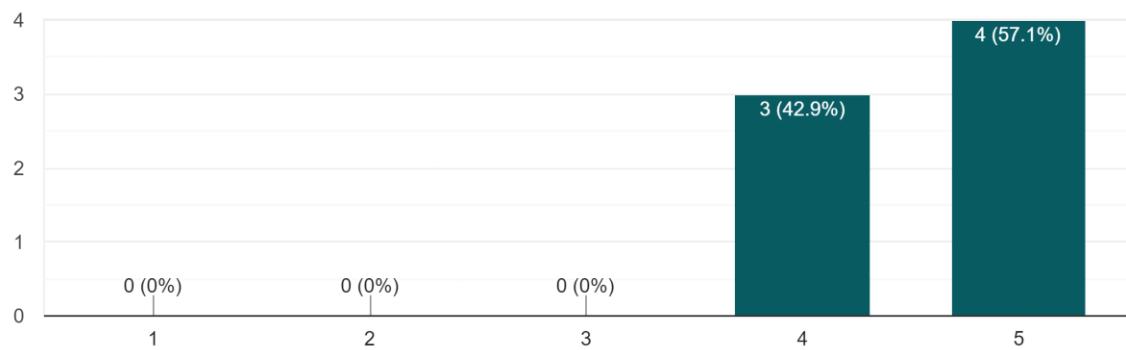
Were you able to find "Fahim" and follow him?

7 responses



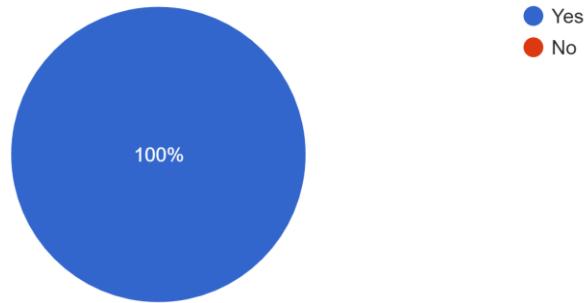
How easy was it for you to find and follow "Fahim"?

7 responses



Were you able to find popular posts from people you don't follow?

7 responses



Any suggestions for the Explore page?(Optional)

nothing

Posts can be in the middle and scrollable.

Please describe what changes happened to your home screen after following "Fahim".

I was able to see the mans posts

His posts appeared on the home page

Fahims posts show on the home screen after following Fahim

His posts showed up there

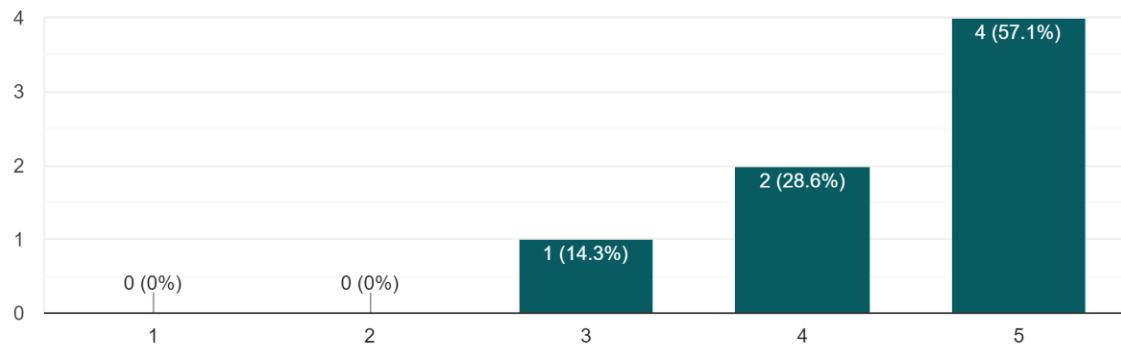
Follower number

Refreshes with her posts

I saw his posts

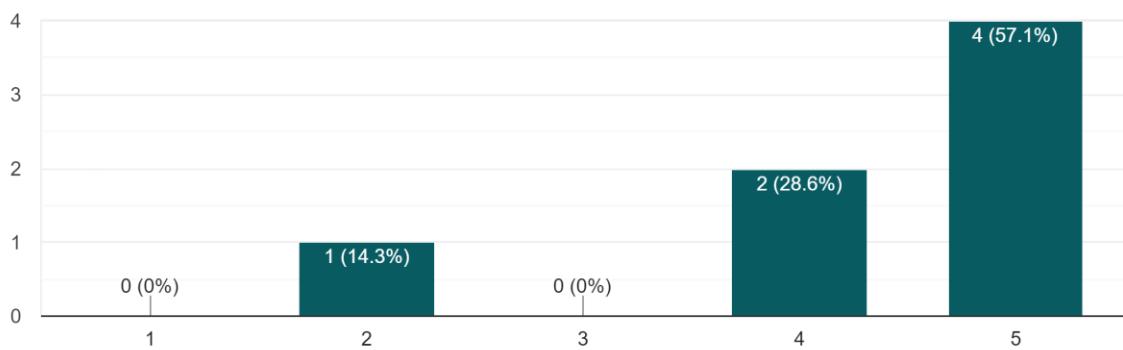
How easy was it for you to like and comment on a post?

7 responses



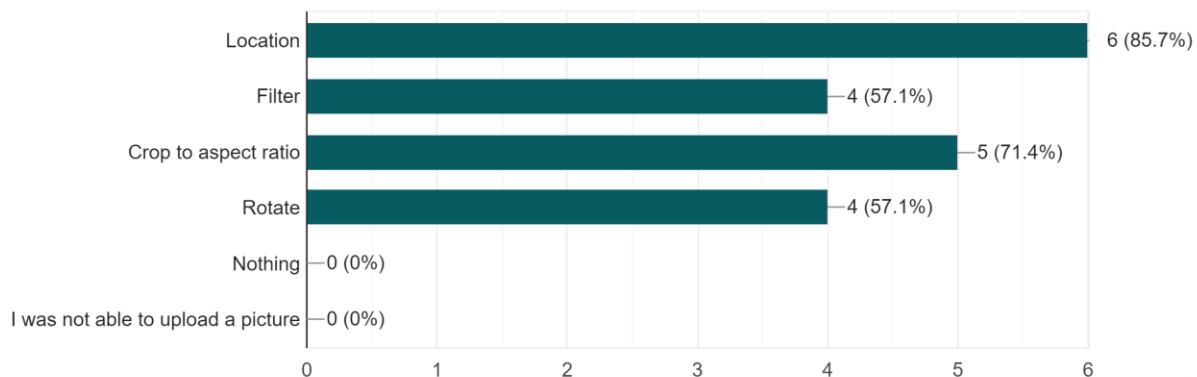
How easy was it for you to report a post?

7 responses



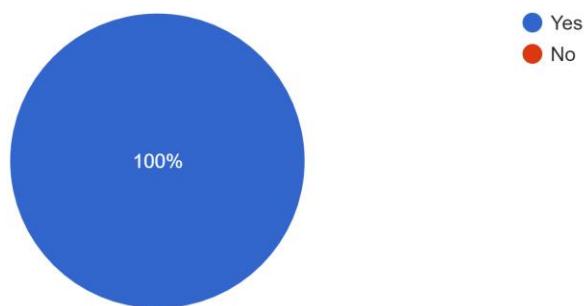
What additional option(s) did you add to your uploaded post?

7 responses



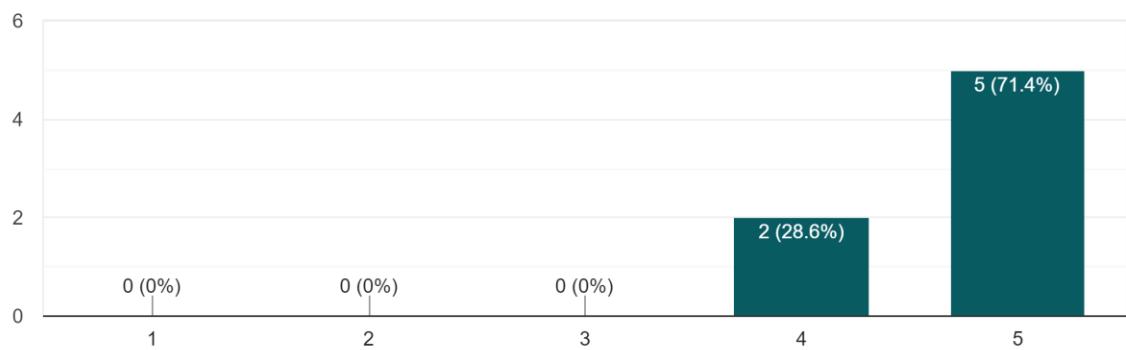
Were you able to view your post on the home page?

7 responses



How easy was it to search for your post on the map?

7 responses

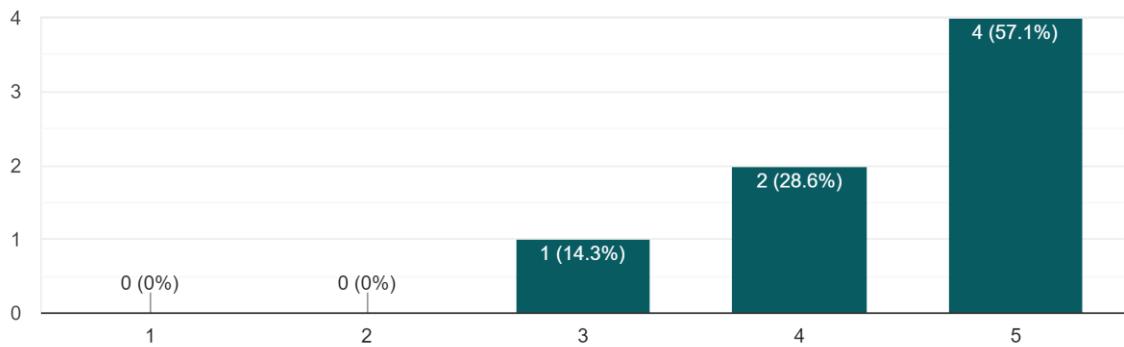


Any suggestions for the Home page? (Optional)

Home Page was good.

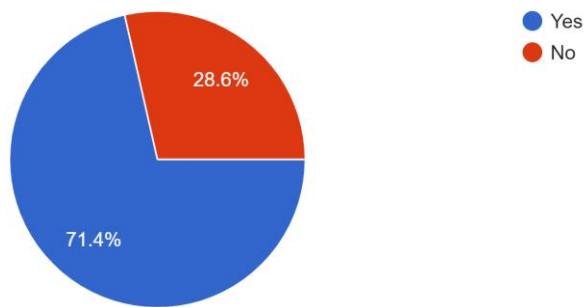
How easy was the leaderboard to interpret?

7 responses



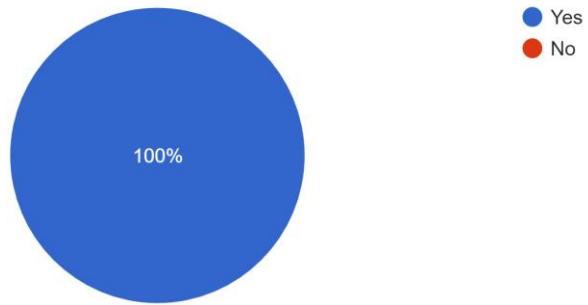
Were you able to locate your name on your leaderboard?

7 responses



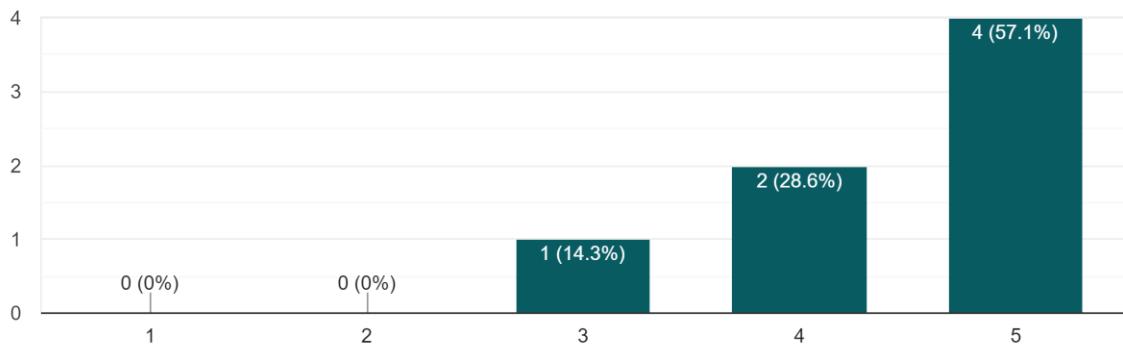
Were you able to unfollow "Fahim"?

7 responses



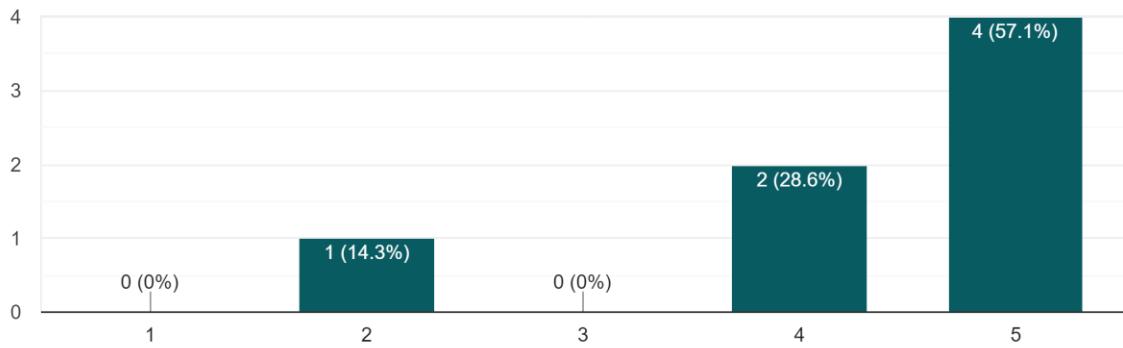
How easy was it to view posts in single post (list view) mode?

7 responses



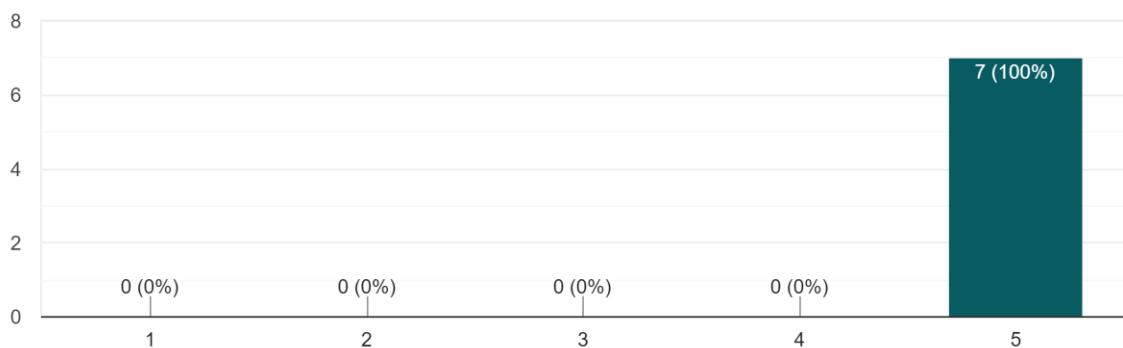
How easy did you find it to delete a post?

7 responses



How easy was it for you to sign out from your account?

7 responses



Any suggestions for the Profile page? (Optional)

liked the grid and list view feature

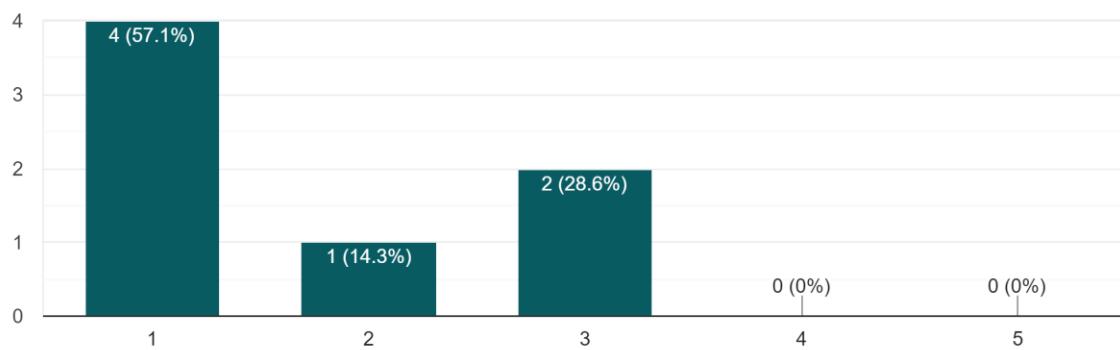
(Bug) I could see the add post button and settings drawer icon on someone else's profile

Entering a caption felt mandatory.

4.3.3 Post Questionnaire Results

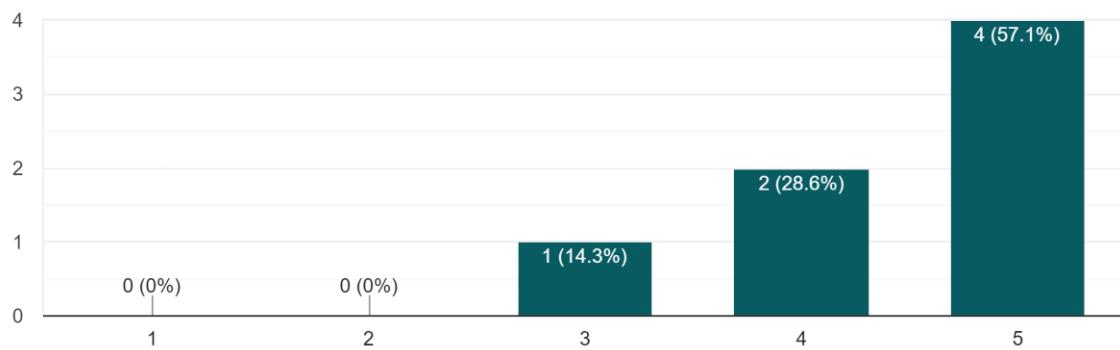
I found the application unnecessarily complex.

7 responses



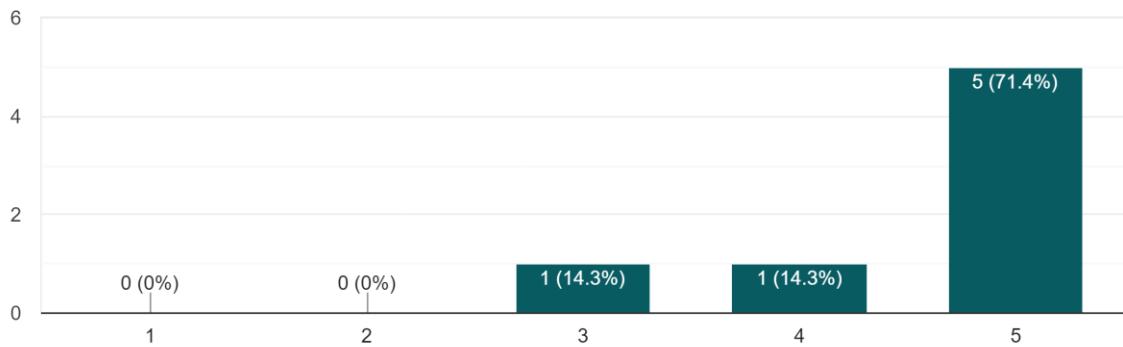
I was easily able to navigate my way through the app.

7 responses



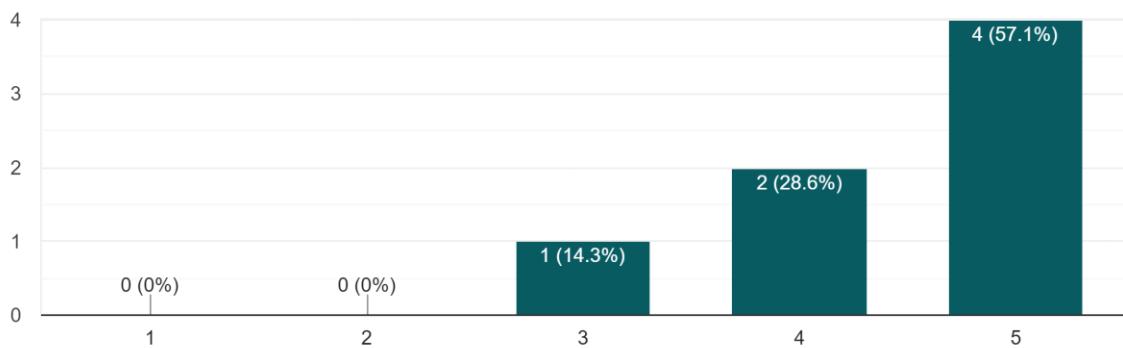
The app's design is intuitive and straightforward.

7 responses



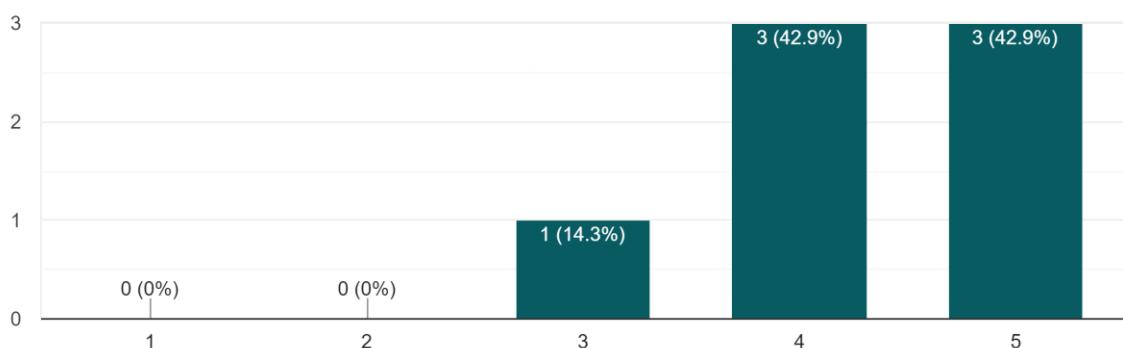
I would use this app because of the service it provides.

7 responses



Would you recommend this application?

7 responses



Overall, any suggestions you would like to give?

The ui was aesthetic and pleasing to the eye.

Put map on explore page.

Love it

In the application, which section did you like the best? 7 responses

Post

posts on map

The map view was probably the coolest

Home Page

The searching for posts on map

Upload

Profile

In the application, which section did you like the least? 3 responses

Comments

The Settings pannel because was expecting to see more working parts

None

5. Future Changes to the System

Based on the usability data and responses from test users, most of the recommendations were received based on the UI and location of buttons that were responsible to show the applications main functions. Suggestions like the source of the Map View button's placement in the Explore page rather than in the Home Screen, reducing the number of buttons visible on a certain page etc. can be taken into consideration when delivering a polished version of the completed application.

References

- AltexSoft. (2020). *Structured vs Unstructured Data: Compared and Explained*. [Online] Available at: <https://www.altexsoft.com/blog/structured-unstructured-data/> [Accessed 28 Mar. 2021].
- Bohbot, K., 2021. *Firebase Storage: What It Is and How It Works | Back4App Blog*. [online] Back4App. Available at: <https://blog.back4app.com/firebase-storage/> [Accessed 4 February 2021].
- Edpresso. n.d. *What is Firebase?*. [Online] Available at: <https://www.educative.io/edpresso/what-is-firebase> [Accessed 4 February 2021].
- Greenberg, A., 2016. *Hacker Lexicon: What Is Password Hashing?*. [Online] Wired. Available at: <https://www.wired.com/2016/06/hacker-lexicon-password-hashing/> [Accessed 4 February 2021].

Google Cloud. 2021. *Google Cloud Overview*. [Online] Available at: <https://cloud.google.com/docs/overview> [Accessed 4 February 2021].

Google Developers. (2021). Google Maps Platform | Google Developers. [online] Available at: <https://developers.google.com/maps/documentation> [Accessed 1 Apr. 2021].

Google Firebase (2021). Firebase Pricing Plans | Available at: <https://firebase.google.com/pricing?authuser=0> [Accessed 1 Apr. 2021]

MongoDB. n.d. *What is NoSQL? NoSQL Databases Explained*. [Online] Available at: [https://www.mongodb.com/nosql-explained#:~:text=NoSQL%20databases%20\(aka%20%22not%20only,based%20on%20their%20data%20model.&text=They%20provide%20flexible%20schemas%20and,data%20and%20high%20user%20loads.](https://www.mongodb.com/nosql-explained#:~:text=NoSQL%20databases%20(aka%20%22not%20only,based%20on%20their%20data%20model.&text=They%20provide%20flexible%20schemas%20and,data%20and%20high%20user%20loads.) [Accessed 4 February 2021].

Raible, M., 2017. *What the Heck is OAuth?*. [Online] Okta Developer. Available at: <https://developer.okta.com/blog/2017/06/21/what-the-heck-is-oauth> [Accessed 4 February 2021].

Schwaber, K. and Sutherland, J. (2020). *The Scrum Guide The Definitive Guide to Scrum: The Rules of the Game*. [online] . Available at: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100>. [Accessed 30 Mar. 2021].

Snowflake. (2021). *JSON Examples*. [online] Available at: <https://www.snowflake.com/trending/json-examples> [Accessed 29 Mar. 2021].

Su, B., 2017. *What is Google Analytics, and why is it important to my business?*. [Online] Medium. Available at: <https://medium.com/analytics-for-humans/what-is-google-analytics-and-why-is-it-important-to-my-business-8c083a9f81be> [Accessed 4 February 2021].

Techotopia. 2017. *Firebase Realtime Database*. [Online] Available at: https://www.techotopia.com/index.php/Firebase_Realtime_Database [Accessed 4 February 2021].

All additional references used for the project are listed either in-line in code files or in the References.txt file found in the project zip.

Appendix

Table of Content

1. Original Plan	136
1.1 Original Plan – Phase 1	136
1.2 Original Plan – Phase 2	137
1.3 Original Plan – Phase 3	138
1.4 Updated Plan - Phase 1	139
1.5 Updated Plan – Phase 2	140
1.6 Updated Plan – Phase 3	Error! Bookmark not defined.
2. Landing Page	143
2.1 Company landing page (PC/iPad View)	143
2.2 Company landing page (Mobile View)	145
3. Meeting Minutes	149

1. Original Plan

1.1 Original Plan – Phase 1

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUUE DATE	DURATION	DEPENDENCIES
0.1	App Development					
0.1.1	Splash / Sign up / Login	Fahim	1/11/21	1/24/21	10	-
0.1.2	Home Screen	Ammad	1/11/21	1/24/21	10	-
0.1.3	Profile / Notification panel	Mayar	1/11/21	1/24/21	10	-
0.1.4	Game feed	Fahim	1/24/21	2/4/21	9	-
0.1.5	Communities template	Ammad	1/24/21	2/4/21	9	-
0.1.6	Post template	Mayar	1/24/21	2/4/21	9	-
0.1.7	Leaderboard and Progress template	Ayushi	1/11/21	1/24/21	10	-
0.1.8	Documentation	Ayushi	1/24/21	2/4/21	9	-
0.2	RDS					
0.2.1	Prepare the datapoints	Wesley	1/11/21	1/15/21	5	-
0.2.2	Structure the primary tables	Fayyaz	1/16/21	1/23/21	5	0.2.1
0.2.3	Structure the secondary Tables	Zainab	1/16/21	1/23/21	5	0.2.1
0.2.4	Create relations and Finalize Schema	Fayyaz	1/24/21	2/1/21	6	0.2.2
0.2.5	Documentation	Zainab	2/1/21	2/3/21	3	-
0.3	Cloud					
0.3.1	Creating IAM / AWs Organizations	Zainab	1/11/21	1/15/21	5	-
0.3.2	Basic Architecture	Abis	1/15/21	1/18/21	2	-
0.3.3	Create EC2, RDS, Bucket	Abis	1/18/21	1/22/21	5	0.3.2
0.3.4	Initialize Backend services	Zainab	1/24/21	1/28/21	4	0.3.3
0.3.5	Documentation	Fayyaz	2/1/21	2/3/21	3	-

1.2 Original Plan – Phase 2

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUUE DATE	DURATION	DEPENDENCIES
1.1	RDS operations					
1.1.1	Create Primary and Secondary tables	Fayyaz	2/7/21	2/13/21	5	-
1.1.2	Create predefined queries	Wesley	2/7/21	2/13/21	5	-
1.1.3	Test functionality and accessibility	Fayyaz	2/14/21	2/20/21	5	1.1.1
1.1.4	Integration with access points	Abis	2/20/21	2/27/21	5	1.1.3
1.1.5	Deployment on AWS	Abis	2/27/21	3/6/21	5	1.1.4
1.1.6	Documentation	Wesley	2/27/21	3/6/21	5	1.1.4
1.2	App Development					
1.2.1	Feedback and Review	Fahim	2/7/21	2/13/21	5	-
1.2.2	Improvements and redesigns	Ammad	2/14/21	2/20/21	5	1.2.1
1.2.3	Integrations with RDS and testing	Mayar	2/20/21	2/27/21	5	1.2.2
1.2.4	Documentation	Ayushi	2/27/21	3/4/21	4	1.2.3
1.3	Cloud					
1.3.1	Connect backend services	Zainab	2/7/21	2/13/21	5	-
1.3.2	Setup API Gateway and Route 53	Abis	2/14/21	2/17/21	3	1.3.1
1.3.3	Access Testing and Documentation	Zainab	2/17/21	2/19/21	3	1.3.2
1.4	API					
1.4.1	Structure the APIs	Fahim	2/14/21	2/17/21	3	-
1.4.2	Build the API	Fayyaz	2/21/21	2/23/21	2	1.4.2
1.4.3	Integrate and test API requests	Fayyaz	2/24/21	2/27/21	3	1.4.2
1.4.4	Documentation	Fayyaz	2/28/21	3/2/21	2	1.4.3

1.3 Original Plan – Phase 3

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUUE DATE	DURATION	DEPENDENCIES
2.1	CI / CD					
2.1.1	Create a staging environment	Abis	3/7/21	3/10/21	3	-
2.1.2	Create a production environment	Abis	3/11/21	3/13/21	2	2.1.1
2.1.3	Integration between pro and stage	Zainab	3/14/21	3/20/21	5	2.1.2
2.1.4	Create Unit tests	Wesley	3/7/21	3/20/21	10	-
2.1.5	Deployment integration	Wesley	3/21/21	3/24/21	3	2.1.4
2.1.6	Documentation	Wesley	3/25/21	3/26/21	2	2.1.5
2.2	Feedback and Review					
2.2.1	Create the Beta app on Staging	Fahim	3/11/21	3/15/21	3	2.1.1
2.2.2	Share the app wiht user base	Ammad	3/16/21	3/17/21	2	2.2.1
2.2.3	Collect Feedback and Improvements	Mayar	3/18/21	3/20/21	2	2.2.2
2.2.4	Implementation of improvements	Ayushi	3/20/21	4/6/21	12	2.2.3
2.3	Presentation and Marketing					
2.3.1	Create a to market strategy	Fayyaz	3/7/21	3/10/21	3	-
23.2	Prepare a demo deck	Fayyaz	3/11/21	3/13/21	2	-
2.3.2	Create posters and Advertisments	Ayushi	3/14/21	3/20/21	5	-
2.3.2	Launch the App	Fahim	4/7/21	4/8/21	2	2.2.4

1.4 Updated Plan - Phase 1

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUUE DATE	DURATION	DEPENDENCIES
0.1	App Development					
0.1.1	Splash / Sign up / Login	Fahim	1/11/21	1/24/21	10	-
0.1.2	Home Screen	Ammad	1/11/21	1/24/21	10	-
0.1.3	Profile / Notification panel	Mayar	1/11/21	1/24/21	10	-
0.1.4	Game feed	Fahim	1/24/21	2/4/21	9	-
0.1.5	Communities template	Ammad	1/24/21	2/4/21	9	-
0.1.6	Post template	Mayar	1/24/21	2/4/21	9	-
0.1.7	Leaderboard and Progress template	Ayushi	1/11/21	1/24/21	10	-
0.1.8	Documentation	Ayushi	1/24/21	2/4/21	9	-
0.2	Real Time Database					
0.2.1	Prepare the datapoints	Wesley	1/11/21	1/15/21	5	-
0.2.2	Define JSON structure	Ammad	1/16/21	1/23/21	5	0.2.1
0.2.3	Define DB security rules	Abis	1/16/21	1/23/21	5	0.2.1
0.2.4	Documentation	Ayushi	1/24/21	2/1/21	6	0.2.2
0.3	Firebase (GCP)					
0.3.1	Set up real time DB	Abis	1/11/21	1/15/21	5	-
0.3.2	Set up authentication	Abis	1/15/21	1/18/21	2	-
0.3.3	Set up Google Storage (firebase)	Fayyaz	1/18/21	1/22/21	5	0.3.2
0.3.4	Set up webhosting (firebase)	Abis	1/24/21	1/28/21	4	0.3.3
0.3.5	Documentation	Mayar	2/1/21	2/3/21	3	-

1.5 Updated Plan – Phase 2

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUUE DATE	DURATION	DEPENDENCIES
1.1	Machine Learning					
1.1.1	Test and scope custom models	Fayyaz	2/7/21	2/13/21	5	-
1.1.2	Firebase scoping of ML services	Fayyaz	2/7/21	2/13/21	5	-
1.1.3	Integeration of Facial Recognition	Fayyaz	2/14/21	2/20/21	5	1.1.1
1.1.4	Integeration of Landmark detection using ML	Ammad	2/20/21	2/27/21	5	1.1.3
1.1.5	Deploy and test the integrated models	Abis	2/27/21	3/6/21	5	1.1.4
1.1.6	Documentation	Wesley	2/27/21	3/6/21	5	1.1.4
1.2	App Development					
1.2.1	Feedback and Review	Fahim	2/7/21	2/13/21	5	-
1.2.2	Improvements and redesigns	Ammad	2/14/21	2/20/21	5	1.2.1
1.2.3	Integrations with RDS and testing	Mayar	2/20/21	2/27/21	5	1.2.2
1.2.4	Documentation	Ayushi	2/27/21	3/4/21	4	1.2.3
1.3	GPS and Geolocation					
1.3.1	Integrate API requests for Maps	Ammad	2/7/21	2/13/21	5	-
1.3.2	Integrate machine learning model for landmark detection	Fayyaz	2/14/21	2/17/21	3	1.3.1
1.3.3	Access Testing and Documentation	Ammad	2/17/21	2/19/21	3	1.3.2
1.4	API					
1.4.1	Scope and structure API calls	Fahim	2/14/21	2/17/21	3	-
1.4.2	Management and integration	Wesley	2/21/21	2/23/21	2	1.4.2
1.4.3	Testing and optimization	Wesley	2/24/21	2/27/21	3	1.4.2
1.4.4	Documentation	Fahim	2/28/21	3/2/21	2	1.4.3

1.6 Updated Plan – Phase 3

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUUE DATE	DURATION	DEPENDENCIES
2.1	Security					
2.1.1	Configuring hashing functions	Abis	3/7/21	3/10/21	3	-
2.1.2	Setting up debugging keys	Abis	3/11/21	3/13/21	2	2.1.1
2.1.3	O-Auth configuration (tokens)	Fayyaz	3/14/21	3/20/21	5	2.1.2
2.1.4	DDOS protection and downtime monitor	Abis	3/7/21	3/20/21	10	-
2.1.5	Android pro-gaurd implementation	Abis	3/21/21	3/24/21	3	2.1.4
2.1.6	Documentation	Wesley	3/25/21	3/26/21	2	2.1.5
2.2	Feedback and Review					
2.2.1	Create the Beta app on Staging	Fahim	3/11/21	3/15/21	3	2.1.1
2.2.2	Share the app wiht user base	Ammad	3/16/21	3/17/21	2	2.2.1
2.2.3	Collect Feedback and Improvements	Mayar	3/18/21	3/20/21	2	2.2.2
2.2.4	Implementation of improvements	Ayushi	3/20/21	4/6/21	12	2.2.3
2.3	Presentation and Marketing					
2.3.1	Create a to market strategy	Fayyaz	3/7/21	3/10/21	3	-
2.3.2	Prepare a demo deck	Fayyaz	3/11/21	3/13/21	2	-
2.3.2	Create posters and Advertisments	Ayushi	3/14/21	3/20/21	5	-
2.3.2	Launch the App	Fahim	4/7/21	4/8/21	2	2.2.4

2. Landing Page

2.1 Company landing page (PC/iPad View)

The screenshot shows a professional company landing page with a light blue gradient background. At the top left is the company logo 'ZenTech Ltd.'. Below it is a headline: 'Building state-of-the-art applications using the latest technologies'. A brief description follows: 'ZenTech Ltd. is a leading mobile and web application development company, providing our services to turn your tech projects and ideas into a success!'. Two buttons at the bottom of this section are 'CONTACT US' and 'OUR PROJECTS'. To the right is a large image of two smartphones displaying various app interfaces.

Maximize your ROI
by building on your past software investments

Save Time
by working with a team that understands your ideas and industry

Activate Windows
Go to Settings to activate Windows.

Rest Easy
with experienced developers and frequent contact keeping you in control

Trust our devs
to create among the best applications on any platform

Check Out Our Projects

Partner with us to create seamless digital customer experiences whether you are looking to create a new application, need help with development or QA of your existing solutions.

EVERYTHING ABOUT US

From building intuitive apps to dynamic, reactive websites we have you covered!

We at ZenTech believe in delivering products of the highest quality with a strong team of developers who understand what your customers need.

Sell your product with confidence while strengthening existing customer relationships.

Keep your brand up-to-date by revitalizing your custom software.

MEET THE TEAM

CONTACT US

DISCLAIMER: This is a simulated example. All trademarks and service marks are the property of their respective owners.

OUR CLIENTS



THE TEAM



Still Have Questions?
Fill In The Form

Your Name

Your E-mail

Your Message

Agree to receive commercial information from
Tawfeer Inc.

CONTACT US

Activate Windows
Go to Settings to activate Windows.



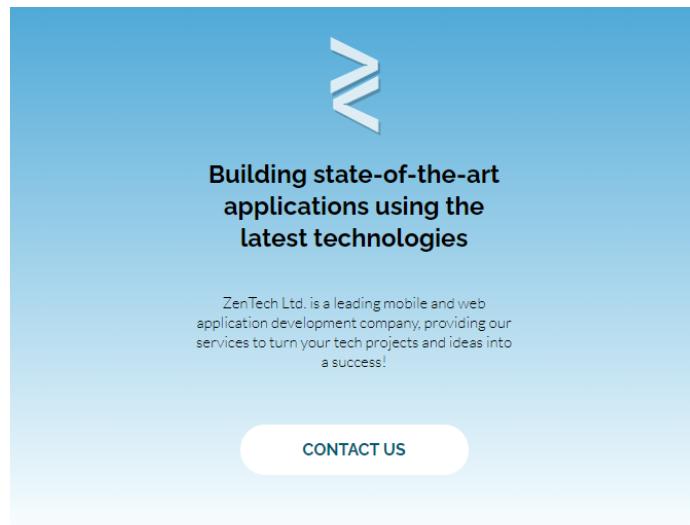
© 2019-2021 ZenTech Ltd.



Contact us via email
- wesley@zentech.com
- mayar@zentech.com

Activate Windows
Go to Settings to activate Windows.

2.2 Company landing page (Mobile View)



Save Time

by working with a team that understands your ideas and industry



Maximize your ROI

by building on your past software investments



Rest Easy

with experienced developers and frequent contact keeping you in control



Trust our devs

to create among the best applications on any platform

Check Out Our Projects

Partner with us to create seamless digital customer experiences whether you are looking to create a new application, need help with development or QA of your existing solutions.



EVERYTHING ABOUT US

From building intuitive apps to dynamic, reactive websites we have you covered!

Sell your product with confidence while strengthening existing customer relationships.

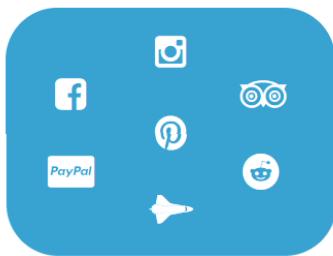
We at ZenTech believe in delivering products of the highest quality with a strong team of developers who understand what your customers need.

Keep your brand up-to-date by revitalizing your custom software.

[CONTACT US](#)

[MEET THE TEAM](#)

OUR CLIENTS



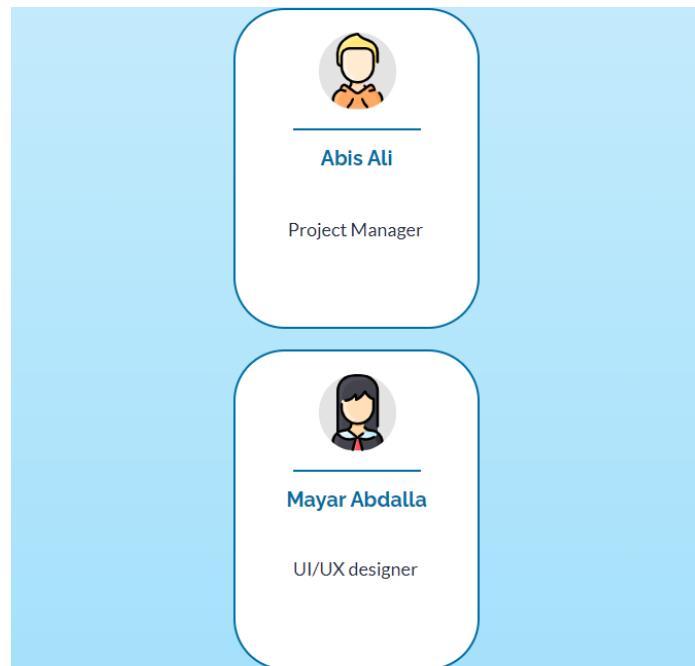
THE TEAM

A profile card for Fayyaz Ameen, featuring a circular placeholder for a photo, the name "Fayyaz Ameen" in bold blue text, and the title "Senior IT Manager" below it.

A profile card for Ammad Ahmad, featuring a circular placeholder for a photo, the name "Ammad Ahmad" in bold blue text, and the title "Software Tester Head" below it.

A profile card for Ayushi Amin, featuring a circular placeholder for a photo, the name "Ayushi Amin" in bold blue text, and the title "Senior Software Developer" below it.

A profile card for Fahim Abdul Gani, featuring a circular placeholder for a photo, the name "Fahim Abdul Gani" in bold blue text, and the title "Chief Technology Officer" below it.



Still Have Questions?
Fill In The Form

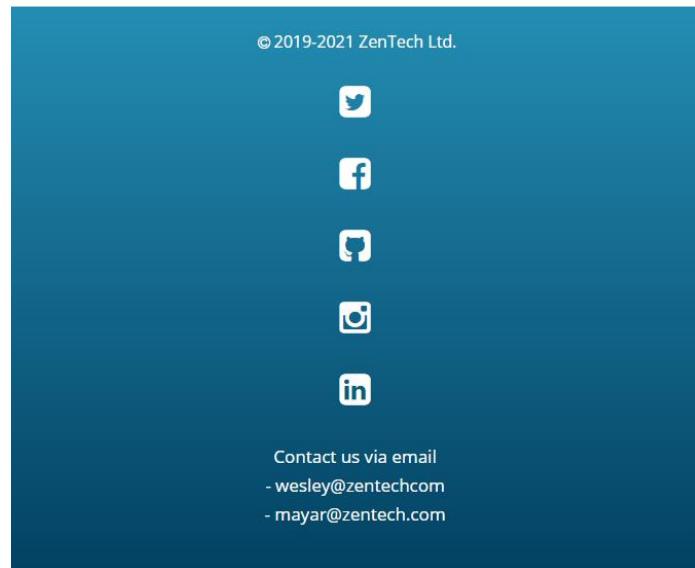
Your Name

Your E-mail

Your Message

I agree to receive commercial information from
Tasweer Inc.

CONTACT US



3. Meeting Minutes

Please note that the person who held the meeting represented by “meeting by” is automatically present and will not be in the list of attendees to avoid repetition of their name.

Minutes of Meetings 1

Time and Date: 15/10/2020 10:30am

Meeting By: Talal Shaikh

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Fahim Abdul Gani
5. Fayyaz Ameen Mohammed
6. Mayar Abdalla Mohamed Abdellatif Ahmed
7. Wesley Barreto
8. Zainab H Aibani

Discussion:

- Introduction to the project.
- Questions were asked by Prof Talal to each group member.
- Work ethic and group contribution advice was given by Prof Talal.

Minutes of Meetings 2

Time and Date: 15/10/2020 4:15pm

Meeting By: Fahim Abdul Gani

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Fahim Abdul Gani
5. Fayyaz Ameen Mohammed
6. Mayar Abdalla Mohamed Abdellatif Ahmed
7. Wesley Barreto

8. Zainab H Aibani

Discussion:

- Strengths and weakness of every member were discussed.
- Finalizing discussion on specification requirement.
- Creating a unified sharing location of documents. Github repository was created.
- Abis and Mayar are now responsible for minutes of the meeting.
- Everyone is expected to present their functional and non-functional requirements by Saturday. 17 Oct.

Minutes of Meetings 3

Time and Date: 17/10/2020 3:00pm

Meeting By: Fahim Abdul Gani

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Fayyaz Ameen Mohammed
5. Mayar Abdalla Mohamed Abdellatif Ahmed
6. Wesley Barreto
7. Zainab H Aibani

Action Items

- Compilation of all members' requirement into a single document.
Responsibility of Fahim. Deadline: Next Meeting.

Discussions:

- System requirements vs user requirements.
- We can have different people to moderate (a different group whose sole purpose is moderating – can be done at a later stage).
- Making a wiki page about the rules.
- Service requirements of the system such as cloud services.
- Types of non-functional requirements can include: verification of users.
- For ranking, the users will be divided within groups, so even if users drop down it's not all the way to zero but to the bottom of their rank in their division.

- There will be different types of ranking, e.g. global ranking.
- Discussion on how to move forward and tackle the remaining part of Stage 1.

Discussion Plans for Next Meeting

- Roles will be assigned to members regarding other parts of Stage 1.
- Review of the compilation of functional and non-functional requirements and discuss UML diagram with reference to it.

Minutes of Meetings 4

Time and Date: 19/10/2020 12:00pm

Meeting By: Fahim Abdul Gani

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Fayyaz Ameen Mohammed
5. Mayar Abdalla Mohamed Abdellatif Ahmed
6. Wesley Barreto
7. Zainab H Aibani

Discussions:

- The team members all went through the compiled version of requirement specification done by Fahim.
- On the matter of duplication of photographs being uploaded on the system.
- To prevent user from re-uploading a photograph that another user has captured, the database can be split into categories for efficient searching thorough to prevent uploading of a pre-existing picture in the database.
- On the possibility of taking pictures directly from Google (or any other search engine), use Google Image Search (or the other engines Image Search) to check if an exact replica/very identical picture exists and if it is found then it should not be allowed on system. (Google Images API)
- Logging in using Google account/Facebook account can be possibly integrated into the system.
- Where to store comments of users on each post?
- Adobe XD's Flutter support will make the entire UI design process much efficient so it is the likely candidate to be used in the implementation process.

- All the team members chose the tasks they feel comfortable tackling from the next part of the Stage 1 documents (namely: project decisions and plan, risk analysis and UML diagrams).
- The inclusion of more mini games apart from the guessing game.
- Things to confirm with Professor Talal:
 - What do we add the source of the requirement to be? (We came up with some original requirements and adapted the rest from the spec sheet in accordance to our choice of implementation of the system).
 - Adding the priority for the requirements
 - Is it to be individually for each requirement or it enough to have the priority written within the sentence to be bolded.
 - Cross check on system requirements (are the system requirements correct or are they more suitable to be functional requirements instead?)
 - Integrating different APIs into the system to give the system different features (Google API discussed above).

Discussion Plans for Next Meeting

- Receive confirmation (or have a meeting with professor Talal) regarding the requirement specification.
- UML Diagrams (specifically Use Case diagrams).

Minutes of Meetings 5

Time and Date: 25/10/2020 12:00pm

Meeting By: Fahim Abdul Gani

Attendees:

1. Abis Ali
2. Ayushi Amin
3. Mayar Abdalla Mohamed Abdellatif Ahmed
4. Wesley Barreto
5. Zainab H Aibani

Action Items

Item	Person Responsible	Deadline
To tackle Use Case Diagram	Ayushi, Mayar, Zainab	2-11-20
To tackle Activity Diagram	Fahim, Wesley	2-11-20
To tackle Project Planning and Risk Analysis	Abis, Ammad, Fayyaz (+ Fahim, Wesley)	5-11-20
To tackle UI Design	Ayushi, Mayar, Zainab	N/A

Discussions:

- Regarding Use Case Diagrams:
 - Splitting the tabs, homepages, profile into different use cases
 - Update account type
 - Personal page, add communities
 - Search page, requesting follow and not
- Pinterest idea
 - Unregistered users can view pictures until a certain part

- So you can still see things in the background, but you can't click on it or interact with the picture
- Naming ideas for the system
 - Notice: Naming the system should be based on business focus -- our focus is gamification of photography sharing.
 - *Capture*
 - *Featurised*
 - *Click N Climb*
 - *Unkr* (انقر)
 - *Pentso* (picture in latin)
 - و
 - *Bokeh*
 - *Exposure*
 - *Focus*
 - *Shutter*
 - *Lens* (عدس)
 - *ShutterFocus*
 - *LensFlair*
 - *Tasweer*
 - *Poto*
 - Frame? Frame it?

Minutes of Meetings 6

Time and Date: 27/10/2020 10:30am

Meeting By: Fahim Abdul Gani

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Fayyaz Ameen Mohammed
5. Mayar Abdalla Mohamed Abdellatif Ahmed
6. Talal Shaikh
7. Wesley Barreto
8. Zainab H Aibani

Discussions:

- For source of requirements
 - Anything from the specification document should be stated.
- For user requirement and system requirement
 - Both are a requirement to be present in the final document.
- Use case diagram can be split up for simplicity (while writing the use case spec, making the activity diagram), but needs to be one diagram with all use cases in the final document.
- For activity diagram, show for the most critical parts of the system.
- The core fundamental features needs to be developed by the team.
 - For reverse search, it was not specifically mentioned in the specification, it is an extra functionality for the purpose of unique picture.
 - E.g. user posts on one platform, tries to cross-post on our system but the reverse search detects it.
 - Solution:
 - Add something within the platform/system that once somebody uploads nobody else can upload that picture. (Suggested by the team.)
 - Such as: someone can dispute the ownership of the picture. (Suggested by the Dr. Talal.)
 - E.g. an option to flag the picture.
 - There needs to be a process to allow users to dispute Such as signature, or meta data. If there is an issue optionality needs to be there.
 - The report made by users goes to the central admin.
 - The inclusion of this feature could be a unique selling point of the application.
 - Using the restriction of the picture containing buildings (and no people), a simple AI algorithm can be used so that each time a picture is being processed, it searches for faces/people. If it is a human, it will not allow it to upload.
 - Solution if the system detects faces/humans:
 - Crop the image so the areas that contain humans is not included.
 - Blur the people out of it.
 - The usage of an AI system in the implementation of the application will result in marks of 80+.
- Consider:
 - Structuring how the application will be eventually.
 - How will it be deployed, i.e. system architecture.
 - Example:
 - Connect it to a DevOps pipeline (non-functional requirement).
 - The code gets tested via our test cases.

- This allows the testing cycle to be automated.

Minutes of Meetings 7

Time and Date: 01/11/2020 12:00pm

Meeting By: Fahim Abdul Gani

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Fayyaz Ameen Mohammed
5. Mayar Abdalla Mohamed Abdellatif Ahmed
6. Wesley Barreto
7. Zainab H Aibani

Discussions:

-
- Use Asana (a project management tool) to be able to track our progress (will happen next meeting).
 - Log In System changed to Log In/Out system.
 - Extend adds extra steps and cannot terminate the use case whereas alternative flow allows for termination.
 - E.g. Signing out will terminate the entire system so the option to sign out can be an alternative flow.
 - "Edit Choice" can be written in the use case description.
 - One big Use Case diagram as a high-level view and the rest of the small components can be represented in Use Case Specification.
 - Log In
 - Log Out
 - The above two points will be represented as 2 separate bubbles/use cases.
 - To add alternative flow for "Daily Bonus Points" Use Case
 - "Daily Bonus Points" + "XP on Rare" to be combined into one aspect such as presented below.
 - "Give Additional XP"
 - Then the system will check if it is a rare picture or a daily bonus and then the system will allocate it effectively.
 - Use case for communities and admin can be presented together instead of having a separate diagram for admin access.
 - 2 admins are available.

- 1 admin for communities
 - 1 admin for general
 - For "Admin Approval" Use Case
 - Required in private and public communities
- User gets reports that represent the following information:
 - Hours the system has been used by user
 - User's accumulated points
 - User's place on leaderboard
 - Etc.
- For "Type of Report" Use Case
 - Is available only to admin.
 - Will include all the reports available to user + admin-specific reports
- Change use case location of:
 - "Automated Tagging" -- to be shifted to Posts Use Case diagram
 - "View Posts" -- to be shifted to Posts Use Case diagram
 - Create an alternate flow for unregistered users
- Make a XP or XP + Gaming Use Case Diagram
 - This simplifies the entire process as the system will check:
 - If the picture is rare
 - For daily bonuses
 - How much XP should be given to a user E.g. if user posts a rare picture + they have been posting for consecutive days
 - Incremental XP is crucial here (XP increases)
- Mention Actors in User Specification
 - For each of the actors we can have a set of steps
- To Consider:
 - Users can post picture on the app AND communities (allowed function)
 - Mini games (for users posting)
 - It has been established that a guessing game will be part of the system for points-awarding. To include more or is this sufficient?
 - Making a Mini Game Use Case for a clearer representation of what the mini games section of the system includes.
 - Users general posting or getting involved in the biweekly competitions
 - Users can have the choice of posting on their own page and communities/biweekly competitions
 - Simplification of system for an effective approach in consideration of system implementation and how the system will pan out in the long run.

Discussion Plans for Next Meeting

- Using Asana for efficient management of tasks allocated to each member of the team.
- For the Project Plan:

- Tasks will be broken down by each member tackling it and the estimated time of completion of each task will be noted.
- Deadlines will be set for each task to minimize risk of incomplete work.

Minutes of Meetings 8

Time and Date: 02/11/2020 12:00pm

Meeting By: Fahim Abdul Gani

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Fayyaz Ameen Mohammed
5. Mayar Abdalla Mohamed Abdellatif Ahmed
6. Wesley Barreto
7. Zainab H Aibani

Discussions:

- Deadlines have been set for each part of the project on Asana.
- Regarding Gantt Chart
 - Does it need to include every single task beginning from the start till the end of the project?
 - Sectioning it in terms of sprints?

Discussion Plans for Next Meeting

- Merging all the Use Case Diagrams into one.

Minutes of Meetings 9

Time and Date: 02/11/2020 3:00pm

Meeting By: Fahim Abdul Gani

Attendees:

1. Ayushi Amin
2. Mayar Abdalla Mohamed Abdellatif Ahmed
3. Wesley Barreto

4. Zainab H Aibani

Discussions:

Use Case Diagram Points

- Create account will extend sign in instead of separating the two.
 - Sign in from a different device means verify using two factor therefore might need to include mobile number when doing the two-factor authentication.
 - Check usernames for uniqueness
 - "Verification Email" Use Case will be written in the Use Case Specification only
 - "Customize Account" Use Case will include:
 - Choose account type
 - Choice of categories
 - Delete account
 - "Valid User Details" Use Case will be done by the database and can be written within the specification.
 - "Store Details" Use Case is not needed as creating account will automatically have users' details.
- "Edit Picture" Use Case will include:
 - GIF generation
 - GPS location
 - Photo manipulation
 - Cropping
 - Filters
 - Stamp Overlay (can be implemented)
- "Tagging" Use Case will include:
 - Automated Tagging using GPS.
 - Users can tag objects manually.
 - "Customise Feed" Use Case
 - Delete picture
 - Do we delete points if user deletes picture?
- "Post Picture" Use Case will include:
 - Users can post either in community or on their own feed.
 - Each post users put up can be presented on their feed even if they post it in their community.
 - Users can take a picture from within the app.
 - All communities can be available
 - Check box
 - Default profile will be checked
 - Picture profanity check
 - i.e. check for non-offensive pictures

- "Comments" Use Case will include:
 - Offensive remarks
- "Weekly Competition" Use Case will include:
 - Leaderboard
 - XP
 - Includes "View Posts" to take care of the likes etc.
 - Extend to global ranking of a user
- "Rank" Use Case
 - "Global Ranking" Use Case
 - XP includes the global ranking.
 - Users are divided into separate ranks based on the global XP they have.
 - When you post in competition your global competition user's XP gets effected
 - Reaching a specific rank allows rewards such as gold, bronze, silver etc.
- Unregistered users are allowed to:
 - View posts and search on the platform
- There can be different levels of rarity
 - Uncommon (+25 points)
 - Rare (+50 points)
 - Ultra rare (+100 points)
- For admins
 - Access to certain aspects of the application
 - Has a higher level view of the entire system
 - They can see for example issues reported with system
 - E.g. bugs

Points to consider:

- Do we delete points if user deletes picture?
- For GDPR Compliance
 - Check if they are over the age of 18.
 - Cross reference to check from their email (age declared there).
 - When creating a report about the app usability:
 - Users we are experimenting on are above 18 or not.
 - Is everyone allowed to use the app? (i.e. kids or just people above 18).
- Database of system:
 - One database vs multiple databases?
 - Can include:
 - User data
 - Picture data
 - Likes (and who liked which picture)

- Metadata can include all the list of users who liked it.

Discussion Plans for Next Meeting

- Reviewing all of the work completed by the members.

Minutes of Meetings 10

Time and Date: 08/11/2020 12:00pm

Meeting By: Fahim Abdul Gani

Attendees:

1. Abis Ali
 2. Ammad Ahmed
 3. Ayushi Amin
 4. Fayyaz Ameen Mohammed
 5. Mayar Abdalla Mohamed Abdellatif Ahmed
 6. Wesley Barreto
 7. Zainab H Aibani
- The automated tagging can be done by the system.
 - i. Can be done by the system when posting.
 - ii. Found at the project requirements document: S-UR 12.
 - There are tiers in global rank.
 - i. E.g. Level 0-5 is a rank, Level 5-10 is a higher rank, etc.
 - ii. System will calculate the points.
 - iii. Users will not be allowed to like pictures of other users in the competition page.
 - iv. If a user likes picture posted by another user, XP will get added to them for liking the picture.
 - User on the receiving end of the like will have XP added to them for having their picture liked.
 - v. Add a limitation to the number of points users can get.
 - This is done to prevent the possibility that users will not engage in any other features of the system with exception of liking posts to gain XP.
 - Solution can be in the form of making the "liking XP" be very marginal.
 - E.g. 0.1/0.5 points for liking other users' photos.
 - vi. Higher rank people can have more pictures posted hence more chances to get more XP.
 - Found at the project requirements document: F-UR 11.
 - If user deletes a post:

- i. Some points gets deducted.
- ii. System will notify user before deletion can occur.
 - E.g. "Are you sure you want to delete this post? You will lose the XP you gained from it."
- iii. An archive feature can be implemented to store the pictures. Although the idea may be subject to change because, for example, YouTube removes views if it detects bots. Thus, if users delete their posts, the people who liked their posts can face deduction of points in the same fashion views get deducted from videos where bot behaviour is detected.
- For the system mock ups:
 - i. Add a trophy button in the profiles so users can view each other's achievements.
- Things to inquire about:
 - i. Profile Pictures
 - Can they will include human faces?
 - If profile picture includes human faces then it will be in the database and so the search that the system will conduct to maintain the database free from human faces can be jeopardized.
 - ii. Planning Document:
 - The implementation part of the project needs to clarified. Specifically, allocating the tasks to be conducted and completed in Stage 2 to the members at this stage. Is it a requirement for it to be added in Stage 1's final document and be followed to a tee in Stage 2, or can they be estimates/subject to change?

Minutes of Meetings 11

Time and Date: 10/11/2020 11:00am

Meeting By: Fahim Abdul Gani

Attendees:

1. Abis Ali
2. Abrar Ullah
3. Ayushi Amin
4. Mayar Abdalla Mohamed Abdellatif Ahmed
5. Wesley Barreto
6. Zainab H Aibani

Discussions

- The reader should be able to understand the system while reading the requirement specification and the UML diagrams.
- The use of Use Case Specification will be mainly to complement the requirements, if they are missing the requirements should cover it.
 - Describe the use cases in the diagram.
 - Other diagrams to be added?
 - Design decision that should be taken and justified.
 - If another diagram can improve the visual representation, the adding it would be okay however avoid repetition of points.
 - E.g. adding data flow diagram might increase readability.
- Actors:
 - Platform manager
 - Owner/admin of system
 - View reported posts, managerial tasks, etc.
 - Channel creator
 - Is not admin, but has more privileges than a user.
 - Channel manager
 - Moderate their own channel vs moderate all channels.
- Assumptions can be written/established in the final document.
- Using 2 different classification of requirements:
 - Use MoSCoW to label priority.
 - For prioritization, it is usually represented via tables, with the requirements mentioned alongside their respective priorities.
- The source label of the use cases:
 - Is the source from where the information came from, in this case it is the requirement specification document given but as the specification is vague we can expand on it in our document. Hence then, the source can be attributed to be "the team".
- Profile pictures are allowed but sharing human pictures are not allowed.
 - Other users can see each other's profile pictures.
- Sharding can be used for the databases.
- For the planning document:
 - There are a number way to describe the project plan.
 - Include a brief description.
 - Categorize the tasks into top categories, into a small level of tasks.
 - Insta Gantt on Asana works really well for that purpose.
 - For allocating tasks to be done in the later stages:
 - Identify indicative dates for different tasks.
 - Include contingency in the document too.
 - E.g. putting a day or two contingency for different tasks, to meet any foreseen delays or situations.

Minutes of Meetings 12

Time and Date: 10/11/2020 2:00pm

Meeting By: Fahim Abdul Gani

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Fayyaz Ameen Mohammed
5. Mayar Abdalla Mohamed Abdellatif Ahmed
6. Talal Shaikh
7. Wesley Barreto
8. Zainab H Aibani

Discussions:

- Two factor authentication:
 - Is a non-functional requirement and falls under the category of security.
 - As it is security, do we include it in the Use Case Diagram?
 - If it is to be highlighted as a special feature, then it can be put it in the Use Case Diagram.
 - You can split the requirement to show some in Use Case Diagram and some in Activity Diagram.
- Game leaderboard:
 - Will include the top pictures.
 - Will include a dynamic chart.
 - How will the points be allocated to the users of the system?
- Usability Study
 - The users need to figure out how to use the application.
 - If they find it difficult to work with, then before the implementation of the system, the feedback needs to be taken into account and the design can be changed accordingly.
- Project Planning
 - Need to show the big milestones achieved every 3-4 days, or on weekly basis.
- User Interface
 - Add an option that if the system detect humans then a pop up will appear to prevent posting it.
 - Categories photos from the map, so when users click on it, it will direct them to a page for all the pictures taken in a specific location.
- Communities
 - General page should be the first thing the user sees in communities.

- Suggestion to join different communities can be added.

Minutes of Meetings 13

Time and Date: 10/11/2020 6:00pm

Meeting By: Fahim Abdul Gani

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Fayyaz Ameen Mohammed
5. Mayar Abdalla Mohamed Abdellatif Ahmed
6. Wesley Barreto
7. Zainab H Aibani

Name of minute taker: Wesley (+ Mayar)

Discussions

- User Profile will include:
 - The ranking in current competition.
 - Profile Picture.
 - Username.
 - Account Level.
 - Awards (like instagram highlights).
 - User's posts.
- Picture:
 - Tags on top of picture.
 - Picture.
 - Likes and comments.
- New gaming features:
 - From main page, random pictures appear.
 - Each picture will have:
 - Likes
 - Comments
 - Points
 - Any picture that appears randomly and liked will gain points.
 - The game section will have a filter feature where you can see the top rated posts.
 - Any picture liked after filtering will not get points but will get likes.
- Awards that are weekly/monthly:

- Users get awards which are badges that will stay on their profile after the competition ends.
- Users get 0.1 points for liking a picture. Maximum number of pictures that can be liked is 50 pictures per day.
 - 1 like = 1 point for recipient.
 - 1 post = 5 points.
- Users can only see the ranking of another user and not their points.
 - Points are relevant to the admins/managers of system, users would not be able to see the points they have.
- Discussed about writing down what we will need to implement the system.

Minutes of Meetings 14

Time and Date: 12/11/2020 11:00am

Meeting By: Fahim Abdul Gani

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Fayyaz Ameen Mohammed
5. Mayar Abdalla Mohamed Abdellatif Ahmed
6. Wesley Barreto
7. Zainab H Aibani

Name of minute taker: Wesley and Mayar

Updates:

- Updated UI was shown by Fahim.
- Some of the use case specifications were showed by Wesley.
- The team went through the project tasks sheet and created database schema sheet.
- Few tables have been added to the database schema.

Discussions

- Add mobile number & gender in the signing up page.
- Fix the going back button, when liking and commenting, so it does not undo the like/comment.
- Fix the "Search Location" page to link to proper searching pages.
- "Upload To" option:

- The ability to select where to post the photo for the non-compulsory options.
- Have a filter option in the main feed to see featured posts, posts from people that users follow.
- There will be a separate games page which will show the uploaded pictures in the competition. These will be displayed randomly to different users.
- Manual tagging gives names to objects, whereas automated tagging gives the category (nature, travel, etc.)
- Picture posted to game will be posted to profile by default.
- Possible feature to be added:
 - Community chat.
- Rating of pictures will be done in the game page. The game page is independent of the home page.
- Can picture be involved in multiple category?
 - E.g. building in nature? so the possible categories of this photo will be "nature" and "buildings"
- Issues:
 - Issues with 1 to many and many to many relationships
 - Implementing the recommended system.
- To confirm:
 - Follow requests
 - Users need to send the request and they can't view the XP and stuff if the account is private.

Minutes of Meetings 15

Time and Date: 12/11/2020 5:00pm

Meeting By: Wesley Barreto

Attendees:

1. Ayushi Amin
2. Mayar Abdalla Mohamed Abdellatif Ahmed
3. Zainab H Aibani

Name of minute taker: Wesley

Discussions

- Use Case Specification Discussion:
 - The main feed, game feed and explore page will have filter buttons and on clicking the button you get the following options for each page:
 - Main feed:
 - a. Posts from people you follow

- b. Posts from communities that user is a part of. (Random order of recent posts from communities)
- Game feed:
 - a. Hottest Posts
 - b. Recent Posts
 - c. Random Posts (This filter will account for user points)
- Explore Page:
 - a. Featured Posts (Hottest posts in last 24 hrs)
 - b. Recommended Posts - Based on categories chosen, pics liked, based on who users follow.
- When a user creates an account for the first time, they get a mixture of featured and recommended posts based on choice of categories.
- Just keep 1 user in use case diagram.
- Maximum number of user defined tags in picture is 5.
- Ability to report accounts, pictures, comments. (What about communities?)
- Users can comment maximum 5 times on a post.
- The system gives an award in the form of a badge and XP to a user at the top 10 at the end of a weekly competition.
- Removed WeeklyCompetition use case.
- Users get XP for posting pictures that is not in the game.
- Must add user level and XP to user profile.
- Questions:
 - What part of other user's profile can't you see if you haven't followed that user?
 - Do we have a "Create Communities" button? (Do we need the platform manager to approve a community?)
 - Is the main flow of the InteractWithPost use case fine or do we change it?
 - How can we make sure that a post appeared randomly to the user in the game section?

Minutes of Meeting 16

Date and Time: 21/11/2020

Meeting By: Fahim Abdul Gani

Name of Minutes Taker: Wesley Barreto

Attendees

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin

4. Fahim Abdul Gani
5. Fayyaz Ameen Mohammed
6. Wesley Barreto
7. Zainab H Aibani

Discussions

Discussion about Usability Evaluation of Mock-Ups and parts of stage 1 that are left to do.

Usability Evaluation of Mock-Ups

Task Scenarios to be performed by subjects:

At Home Screen:

- Like a picture.
- Report a comment.
- View notifications.
- Use map view to see pictures taken at Downtown Dubai.

At Search Screen:

- Search for a user named Alex and follow him.

At Profile Screen:

- Delete Picture from profile.
- View all the awards(badges) the user has.

At Post Picture Screen:

- Post Picture

At Communities Screen:

- Request to join a community.

At Game Screen:

- View top 3 users in current competition

Have a list of goals that users will check when they've completed.

Questions For Users:

1. On a scale of 1-10 how easy was it to perform the above scenarios (for each scenario)? (Likert Scale)
2. How would you rate the aesthetic of the app? (Likert Scale)
3. Any suggestions?

For Analysis:

1. Compare results (ease of use) by people who have no experience of photo sharing apps vs those who do.
2. Comparison based on age.
3. Compare the difficulty ratings for each scenario.

Hypothesis: Our application is easy to use.

Get average Likert scale response for each scenario, if the average of each is greater than 8, we say that that part of our system is usable.

Stuff to include in UML Diagrams:

- Data flow diagram.
- Traceability Matrix.
- Activity Diagram, sequence, state machine (If we have time).

Action Items

Item	Person Responsible	Deadline
Experimental Protocol and Plan	Ayushi, Mayar, Zainab	22-11-20
Gantt Chart	Fayyaz, Abis, Fahim, Wesley	24-11-20
Traceability Matrix	Ammad	24-11-20
Additions to UI Mock-Ups	Fahim, Mayar	22-11-20
Data Flow Diagram	Wesley	21-11-20
Sequence Diagrams	Fahim, Wesley	24-11-20

Minutes of Meeting 17

Date and Time: 22/11/2020 11:00am

Meeting By: Fahim Abdul Gani

Name of Minutes Taker: Wesley Barreto

Attendees

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Fayyaz Ameen Mohammed
5. Mayar Abdalla Mohamed Abdellatif Ahmed
6. Wesley Barreto
7. Zainab H Aibani

Discussions

- Reviewed:
 - Traceability Matrix
 - Reviewed the usability report
- Discussed:
 - Gantt Chart

Minutes of Meeting 18

Date and Time: 23/11/2020 11:00am

Meeting By: Fahim Abdul Gani

Name of Minutes Taker: Wesley Barreto

Attendees

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Mayar Abdalla Mohamed Abdellatif Ahmed
5. Wesley Barreto
6. Zainab H Aibani

Discussions

- Reviewed:

- The UI
- The questionnaires

Minutes of Meeting 19

Time and Date: 12/01/2021 12:00 PM

Meeting By: Fahim Abdul Gani

Meeting Minutes By: Mayar

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Fayyaz Ameen Mohammed
5. Mayar Ahmed
6. Wesley Barreto

Action Items

Item	Person Responsible	Deadline
1. To tackle front end	Ayushi, Mayar, Wesley	N/A
1.1 Use Xd to flutter plug in to work on the splash screen/login screen/interest page	Ayushi, Mayar, Wesley	15-01-2021
2. To tackle back end	Abis, Ammad, Fayyaz	N/A

3. To tackle web development	Fahim	N/A
------------------------------	-------	-----

**Discussions*

- Landing page of the website will include:
 - Download options such as "App Store" and "Play Store"
 - Showcase of the prototype
 - GIFs/screenshots
- Split into 3 teams
 - Front End
 - Back End
 - Web Development

Minutes of Meeting 20

Time and Date: 15/01/2021 3:00 PM

Meeting By: Wesley Barreto

Meeting Minutes By: Mayar

Attendees:

1. Abis Ali
 2. Ammad Ahmed
 3. Ayushi Amin
 4. Fahim Abdul Gani
 5. Fayyaz Ameen Mohammed
 6. Mayar Ahmed
- AWS Amplify
 - Allows the flutter code be on the cloud itself.
 - If a user wants to use the app they can either:
 - Install via app store (default option)
 - Directly access it from the web itself (hence it becomes like a web app)

Minutes of Meeting 21

Time and Date: 18/01/2021 12:00 PM

Meeting By: Fahim Abdul Gani

Meeting Minutes By: Mayar

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Mayar Abdalla Mohamed Abdellatif Ahmed
5. Wesley Barreto

Discussions

- Everyone showcased their progress
- Using Telegram's API to allow users to chat between each other within communities.
 - Telegram's API would be used because it is open source.

Minutes of Meeting 22

Time and Date: 19/01/2021 10:30 AM

Meeting By: Fahim Abdul Gani

Meeting Minutes By: Mayar

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Mayar Abdalla Mohamed Abdellatif Ahmed
5. Talal Shaikh
6. Wesley Barreto

Discussions

- The core of the application should be available by Stage 2 deadline
- The website design and the website in general is more towards IS students, hence it should not be the main focus of stage 2.

Minutes of Meeting 23

Date and Time: 20/01/2020 11:00am

Meeting By: Fayyaz Ameen

Name of Minutes Taker: Abis Ali

Attendees

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Wesley Barreto

Discussions

- Reviewed:
 - The website
 - GIT intro

Minutes of Meeting 24

Time and Date: 27/01/2021 12:00 PM

Meeting By: Fahim Abdul Gani

Meeting Minutes By: Mayar

Attendees:

1. Ayushi Amin
2. Mayar Abdalla Mohamed Abdellatif Ahmed
3. Wesley Barreto

Discussions

- The core of the application should be available by Stage 2 deadline
- The website design and the website in general is more towards IS students, hence it should not be the main focus of stage 2.

Minutes of Meeting 25

Time and Date: 31/01/2021 3:15 PM

Meeting By: Wesley Barreto

Meeting Minutes By: Mayar

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Fahim Abdul Gani
5. Fayyaz Ameen Mohammed
6. Mayar Abdalla Mohamed Abdellatif Ahmed
7. Talal Shaikh

Discussions

- Regarding the website
 - Change the company name from Tasweer Inc. because Tasweer is the product being offered to client.
- Regarding the application and the demo
 - Show what has been planned and what has been done.
 - Show the aspects you could have completed or some aspect you did not start.
 - Show what infrastructure you have acquired, the testing done, etc.
 - During the demo
 - Showcase the functionality of the application developed.
 - Prompt sentence: "this is what we managed to do, this is our progress thus far."

Minutes of Meeting 26

Time and Date: 22/02/2021 7:00 PM

Meeting By: Wesley Barreto

Meeting Minutes By: Wesley Barreto

Attendees:

1. Abis Ali
2. Ammad Ahmed
3. Ayushi Amin
4. Fahim Abdul Gani
5. Fayyaz Ameen Mohammed
6. Mayar Ahmed

Discussions

- Members reviewed their progress.

Minutes of Meeting 27

Time and Date: 13/03/2021 6:30 PM

Meeting By: Wesley Barreto

Meeting Minutes By: Wesley Barreto

Attendees:

1. Abis Ali
2. Ayushi Amin
3. Fahim Abdul Gani
4. Mayar Ahmed

Discussions

- Members reviewed their progress.
- Tasks to fix:
 - Home screen (Fix bug when posting image)
 - Remove SizeConfig File from project.
 - Map View: For the map view, we can have a marker that splits itself into many markers when the map is zoomed in.
 - Integrate backend and frontend for likes, comments, chat.

Minutes of Meeting 28

Time and Date: 23/03/2021 8:00 PM

Meeting By: Wesley Barreto

Meeting Minutes By: Wesley Barreto

Attendees:

1. Abis Ali
2. Ayushi Amin
3. Fahim Abdul Gani
4. Fayyaz Ameen Mohammed
5. Mayar Ahmed

Discussions

- Members reviewed their progress. Tasks for Front-End:

1. Make a better search bar (Cross check with previous stage's implemented search bar)
2. Make a notification page. The page will read from a list of notification objects (fields will be username, profile pic, message, etc.).
3. Make the explore page show posts from user interests.
4. Make the buttons in the settings section functional.