THE CHINESE UNIVERSITY OF HONG KONG

FINAL YEAR PROJECT REPORT (TERM 2)

Collaborative Fact Check -

Power of the Crowd

Author:

Au Tak Tung

Supervisor:

Prof. Michael R. LYU

LYU1902

Department of Computer Science and Engineering

April, 2020

0. Abstract

In the era of information explosion, fact-checking is becoming more important for people to distinguish how trustworthy are the online resources. There are numerous fact-check sites available online nowadays, well-known examples include Snopes [1] and PolitiFact [2]. However, most of these fact-check sites mainly focus on political affairs and news of Western countries. There are few fact-check sites target in Hong Kong, for instance, AFP Hong Kong [3]. Nevertheless, these fact-check sites also focus on the news but not on daily issues. Also, these existing fact-check sites seldom allow users to create their own fact-check thread or being involved in the fact-checking process. Instead, they act as an authority to collect news and fact-check through different methods and channels.

In light of that, this project aims to build a fact-checking platform for all types of issues, such as news, daily affairs, rumours etc. Geologically, this project will focus on fact-checking issues at The Chinese University of Hong Kong (CUHK). The methodology of this fact-check platform will also be different from the ordinary approach introduced above. It is believed that crowd intelligence can be used as a collaborative fact-checking approach. In this project, the power of the crowd is applied in the collection and fact-checking of data. With this crowd intelligence approach, this platform is unique in several aspects. It is the only fact-check platform in CUHK, it will be a forum-based fact-check platform, it provides location-based fact-check feature, last but not least, the fact-checking can be done on different types of issues with a higher speed and accuracy.

0. Abstract **2**

1. Introduction **6**
1.1 Background 6
1.2 Motivation 7
1.3 Significance of the project 8
1.4 Objective 9

2. Background Research **11**
2.1 Related Work 11
2.1.1 Goop 11
2.1.2 CUHK Mobile App 13
2.1.3 Snopes 14
2.1.4 PolitiFact 16
2.1.5 AFP Hong Kong 17
2.1.6 Uniqueness and Superiority of CUgo 18
2.2 Technology Overview 19
2.2.1 Flutter 19
2.2.2 Android Studio 20
2.2.3 Firebase 21
2.2.4 Algolia 22

3. Development **23**
3.1 Features 23
3.1.1 Reply Format 23
3.1.2 Interactive Map 23
3.1.3 Rating/ Credibility 23
3.1.4 Privacy 24
3.2 Functionality 25
3.3 Architecture Design 28
3.4 Use Case Diagram 30
3.5 Data Flow Diagram 31
3.5.1 Authentication and Profile System 31
3.5.2 Main system 33
3.6 Database Schema 36
3.6.1 Overall Structure 36
3.6.2 Authentication 37
3.6.3 Post Database 38
3.6.4 Reply Database 40
3.6.5 User Database 42

3.7 Recommendation Function 44
3.8 Keyword Searching function 45

4. Fact-checking 46
4.1 Credibility System 46
4.2 Fact-check Scenario 47
4.3 Further application 48

5. Application Implementation 50
5.1 Login and User Profile System 50
5.2 Homepage, Menu and Category 51
5.3 Browse Topic 52
5.4 Reply Topic 54
5.5 Create Topic 55

6. Procedure Flow 56
6.1 Flow Chart 56
6.2 Scenario 57
6.2.1 Scenario A 57
6.2.2 Scenario B 62

7. Testing 67
7.1 Functional Testing 67
7.1.1 Specification 67
7.1.2 Result 68
7.1.3 Discussion 70
7.2 Usability testing 71
7.2.1 Specification 71
7.2.2 Result 72
7.2.3 Discussion 74
7.3 UI testing 77
7.3.1 Specification 77
7.3.2 Result 78
7.3.3 Discussion 80
7.4 Fact-checking 82
7.4.1 Specification 82
7.4.2 Result 83
7.4.3 Discussion 88

8. Limitation and Difficulties 89
8.1 Data Collection 89
8.2 Alpha Testing 89

9. Future Development **91**
9.1 User motivation 91
9.2 Public forensic Database 92

10. Division of labour **93**
10.1 Authentication 93
10.2 Text-based post 94
10.3 Location-based post 95
10.4 Database management 97
10.5 Search function 98

11. Conclusion **99**
11.1 Mobile Application 99
11.2 Fact-checking 99

12. Acknowledgement **100**

13. Reference **101**

1. Introduction

1.1 Background

It is always believed that the power of the crowd is a kind of collaborative intelligence that exists everywhere in our daily life. Especially in this digital era, it has become easier to make use of this collaborative intelligence for different aims, ranging from providing information to fact-checking. One famous example is Wikipedia, the online encyclopedia is built on collaborative effort and it is contributing and continuously amending the knowledge in the spirit of fact-checking. It has become a major online resource for the public. Making good use of the power of the crowd can enable effective data collection and accurate fact-check of data.

At the same time, a platform which can centralize the information is considered as a powerful tool which can make lives become more convenient. Without a platform which gathers information, people have to spend time reading different web-pages or using different applications just to retrieve part of the information from each page. The advantage of a centralized platform is that it can give an overall picture for the users by providing information from different sources.

Nowadays, when using most of the mobile applications, real-time location service is frequently requested. In several applications, such as Google Maps, users can receive the latest news nearby by looking at what is happening around them currently through the map.

To conclude, fact-checking, information gathering and location tracking are also powerful and important tools which can improve the quality of lives of users significantly. However, there is not a real-time location-based centralized fact-checking platform in Hong Kong, we will explain it more in detail in the next part: Motivation.

1.2 Motivation

Nowadays, with the ease of the internet, the spread of fake news has become an alarming issue. There is a tremendous amount of online information and it is hard for an individual to analyze the authenticity of it. Starting with our university CUHK, we aim to solve this problem by building a centralized platform to gather different kinds of information on the campus and make use of the power of the crowd to verify the credibility of data. For example, data of real-time traffic conditions nearby can be collected and fact-checked by crowd effort. This approach can efficiently gather information and increase the trustworthiness of data. To sum up, our fact-checking system is based on the data gathered on the platform.

Generally speaking, many applications request for accessing users' locations in the application policy. Location data is important nowadays. However, most of the users currently are just receiving the update news of the area nearby, they can rarely report a real-time update of their surroundings such as the breakdown of an elevator in a building. Therefore, we start to think about the possibility of working the other way round, that is user report and update the current situation of the area nearby. For CUHK, the area could be very

small or rather large. It could be as small as a fallen tree near Lake Ad Excellentiam or as large as the area of the whole Campus Circuit East.

After that, we began to have an idea of integrating a fact-checking system with location service to provide real-time updates to users in CUHK. This solution can solve the current problems in CUHK. The first problem is there is not a platform which allows real-time updates of CUHK happenings, users in CUHK can neither receive nor give out the up-to-date information. The second problem is that there is a spreading of fake news not only in CUHK but also in the city recently, users in CUHK cannot determine whether the piece of information could be trusted or not if they do not receive any comments from others.

1.3 Significance of the project

The project would like to provide a centralized platform to gather the information in CUHK. The replies in every topic must be in a standardized format which is customized for each topic. In such a case, different kinds of data can be collected in an appropriate structured format and users can view the information in a clear manner. We provide different choices of reply format for users to choose from such as radio button, scale and text field. Since the reply format in each post is determined by the author of the post, the replies under the same post will be the same. This also makes CUgo stands out from the other forum apps in the market because the format is restricted, it is not loose at all. We create a standardized reply format so as to let the users obtain the information in the replies that they need in just a glance.

There are numerous topics being discussed in CUHK every day. This project provides customized topic suggestions for each user according to different criteria such as their indicated interests, browsing history and the topics they saved as favourite. Therefore, users can reach the topics which they might be interested in easily.

This project also emphasizes on location. An interactive map will be implemented to show replies on a location-based topic. Take 'lost and found items in CUHK' as an example, users need not scroll through all the lost and found posts on the Facebook page anymore as they can view all the lost items near the suspected locations on the map. Details such as image and text descriptions will also be shown.

The real-time feature can allow CUHK users to access information faster than before. Be it the post created or the reply submitted, the changes could be shown in CUgo immediately. All changes will be made and saved to the real-time database. The abuse case or report received from other users will be handled immediately.

1.4 Objective

The goal of the entire project is to design and implement a collaborative mobile application for gathering and fact-checking information in a real-time manner. It could be deployed on the Android platform.

We want to establish a platform which is built by collaborative efforts. It can be sustainable since the users give their continued support for CUgo by either creating meaningful posts or

submitting valuable replies. Most importantly, fact-checking could be achieved with the help of collaborative intelligence. The rating system is also known as the credibility system could become an important indicator for users to determine if the information is trustworthy or not. We hope CUgo would become a trustworthy application that users will go to CUgo if they want to search for any queries about CUHK.

We would like to deploy CUgo on the Android platform. Since we use Flutter for development, it allows deploying the application directly on the Android platform without extra work. As an alpha version of CUgo, our target audience at this stage will be CUHK staff and students using the Android platform.

2. Background Research

2.1 Related Work

2.1.1 Goop

Goop [4] is a forum in which its target users are university undergraduates in Hong Kong. It has both web and mobile versions. In the web version, users could browse through all the topics under different universities, however, the function of posting a comment could only be done in the mobile version. In Goop, topics are shown in the list view with the rating of each topic.

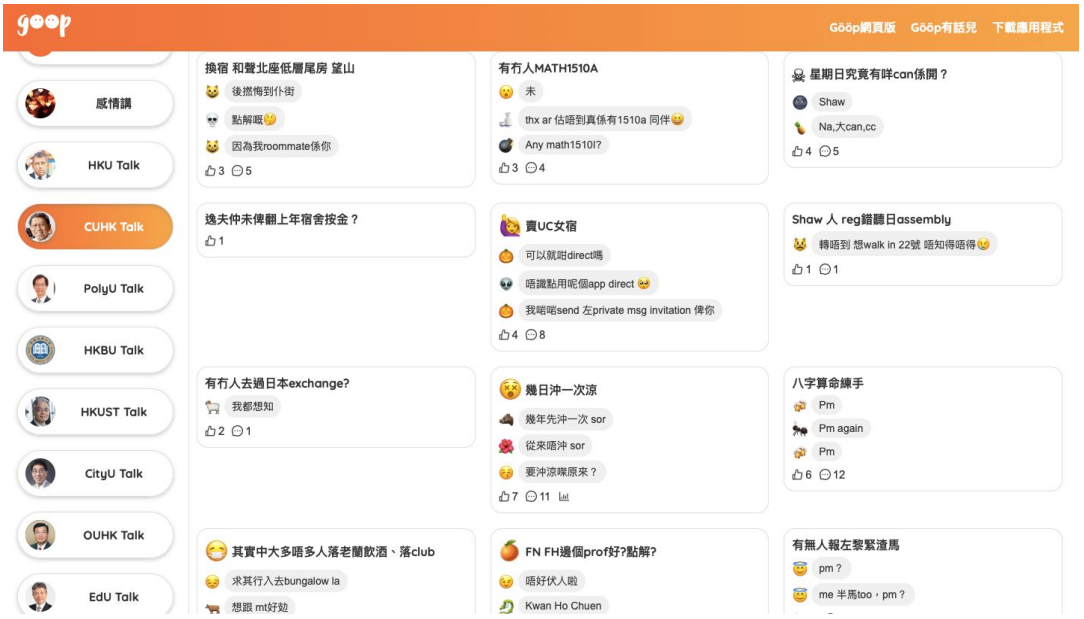


Figure 2.1.1.1 Layout of Goop web version

Figure 2.1.2.1 Example of a discussion which exchanges information

Students could exchange their views and ideas on different topics, especially for the topics about CUHK. The reply format is loose which allows text, images, videos and GIFs, there is no specific restriction upon the format of commenting in each post.

2.1.2 CUHK Mobile App

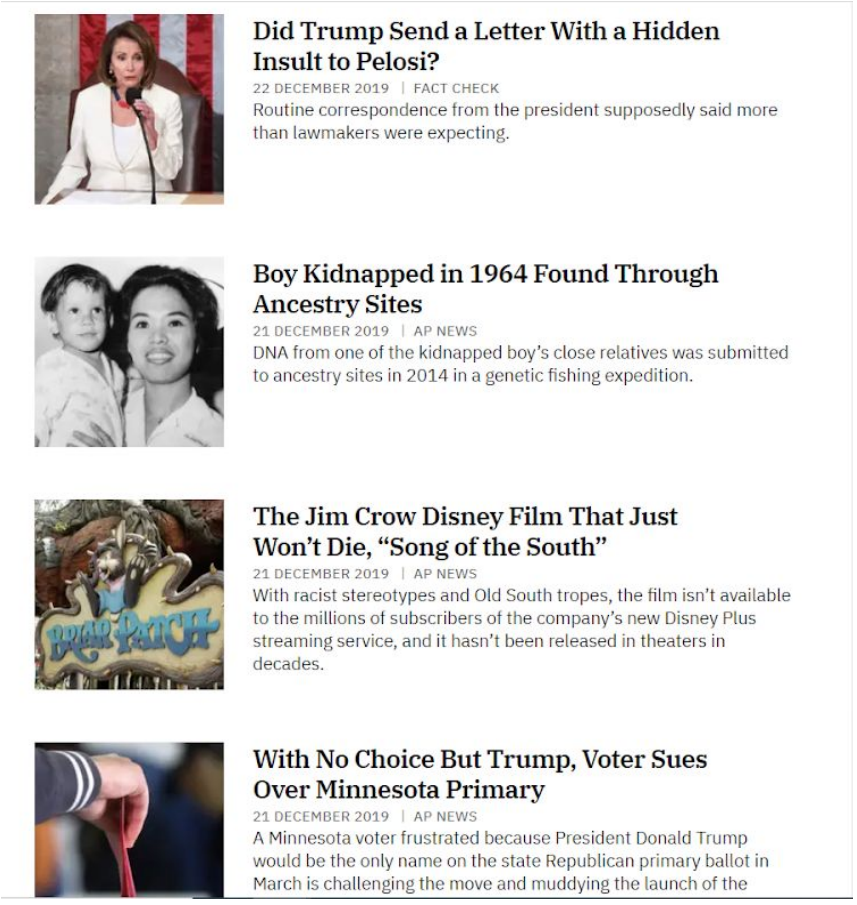
CUHK Mobile App is an official mobile application which is developed by CUHK. It provides a lot of useful information for the CUHK campus. The data in there is static and not real-time. Users cannot interactive with the App but just read and browse the information in there.

Fig 2.1.2.1, 2.1.2.2 The types of information provided by CUHK, which includes the locations of water dispenser on campus

Apart from showing university almanac and shuttle bus timetable, the app also shows the locations of different facilities on the map. However, no photos are attached and the user may need more time to find where are the dispensers placed exactly.

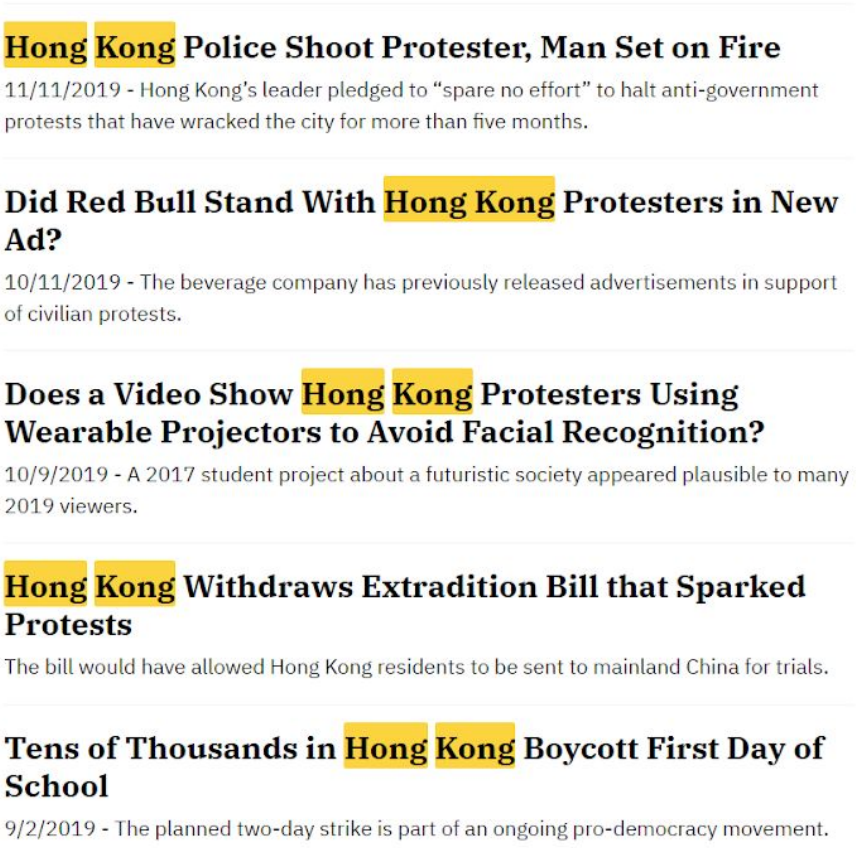
2.1.3 Snopes

Snopes is a fact-checking platform focusing on Western urban legends, news and political issues. It accepts a wide range of sources for fact-check including news, legends and rumours collected from the internet or submitted by users. It is a well-established and comprehensive fact-check site with a long history and high accuracy [5]. The fact-check methodology used for Snopes is to assign their experts to do research on that topic, for example contacting the primary source provider and use non-partisan resources to analyse the topic [6].



2.1.3.1 Snopes homepage

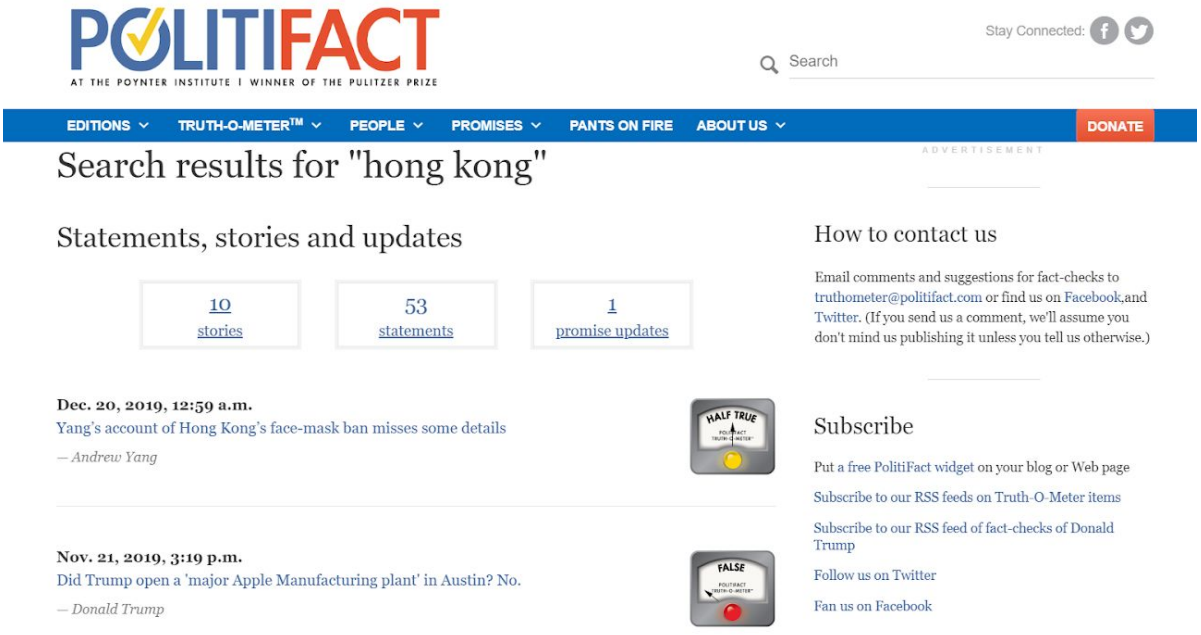
However, Snopes seldom updates issues about Hong Kong, ranging from 1-2 topics per month. Topics about Hong Kong are mostly news and political issues. Some examples are shown below.



2.1.3.2 Search result of "Hong Kong" in Snopes

2.1.4 PolitiFact

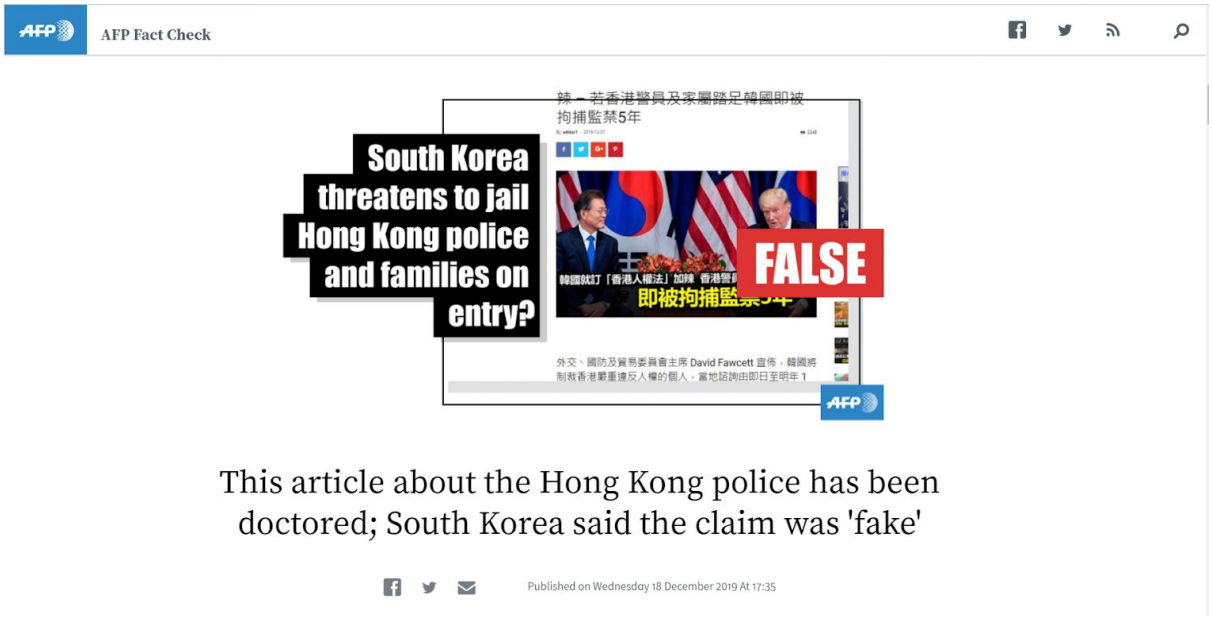
PolitiFact is a fact-checking website founded by the Tampa Bay Times. Its main focus is on news and political affairs of the U.S. PolitiFact collects issues for fact-check from news reports, social media, speeches, reader contribution etc., then send their experts to do research through various means such as primary source verifying, searching the online database and referencing previous fact-checked sources [7]. As PolitiFact has its focus on American issues, issues in Hong Kong are seldom being reported.



2.1.4.1 Search results for “Hong Kong” in PolitiFact

2.1.5 AFP Hong Kong

AFP Hong Kong is operated by the Agence France-Presse (AFP). The AFP fact checks Hong Kong is a branch of the AFP fact check focusing on Hong Kong local issues. The experts in AFP Hong Kong fact-check sources from the internet, news stories etc. through different methods such as tracing the source and verifying images and videos [8]. AFP Hong Kong team is still in a developing stage with limited experts, it becomes an obstacle to verify a wide range of issues. The website mainly focuses on verifying news and political rumours recently.



2.1.5.1 AFP Hong Kong homepage

2.1.6 Uniqueness and Superiority of CUgo

CUgo gathers all kinds of information about CUHK. Be it official news or some private information provided by other users, it can also be found in CUgo. Most importantly, after collecting sufficient information from the crowds, fact-checking can be carried out. Users can determine whether that piece of information is trustworthy or not by different parameters such as the rating of that reply. By using CUgo, users become both a viewer who enjoys the fact-checked result and a contributor who helps to provide information.

2.2 Technology Overview

2.2.1 Flutter

Flutter is a development platform targets at building cross-platform mobile applications for both iOS and Android. Dart language is used in which its syntax is similar to C-style languages and it could be trans-compiled into JavaScript. It provides various user interface widgets for these two different platforms respectively according to their own user interface.

There is also a key advantage of Flutter. It provides a live debug function called "hot reload" [9] which users can redeploy the changes made in the code while having the debugging process, this does not require recompilation and redeployment.

Apart from Flutter, React Native is another platform that is popular among software developers. In the first phase of implementation, we have chosen React Native as well. However, we found that Flutter provides high compatibility with its own database which is called Firebase. By using Firebase, user authentication [10] could be performed easily.

Considering the styling, Flutter definitely does a better job than React Native. The latter only has basic components so users have to put a lot of effort into styling. Fortunately, the former provides adaptive widgets and the user could customize their design easily by using the same widget on both iOS and Android platforms.

Lastly, Flutter ranks higher in the list when it comes to the most wanted framework for developers to develop on it continuously.

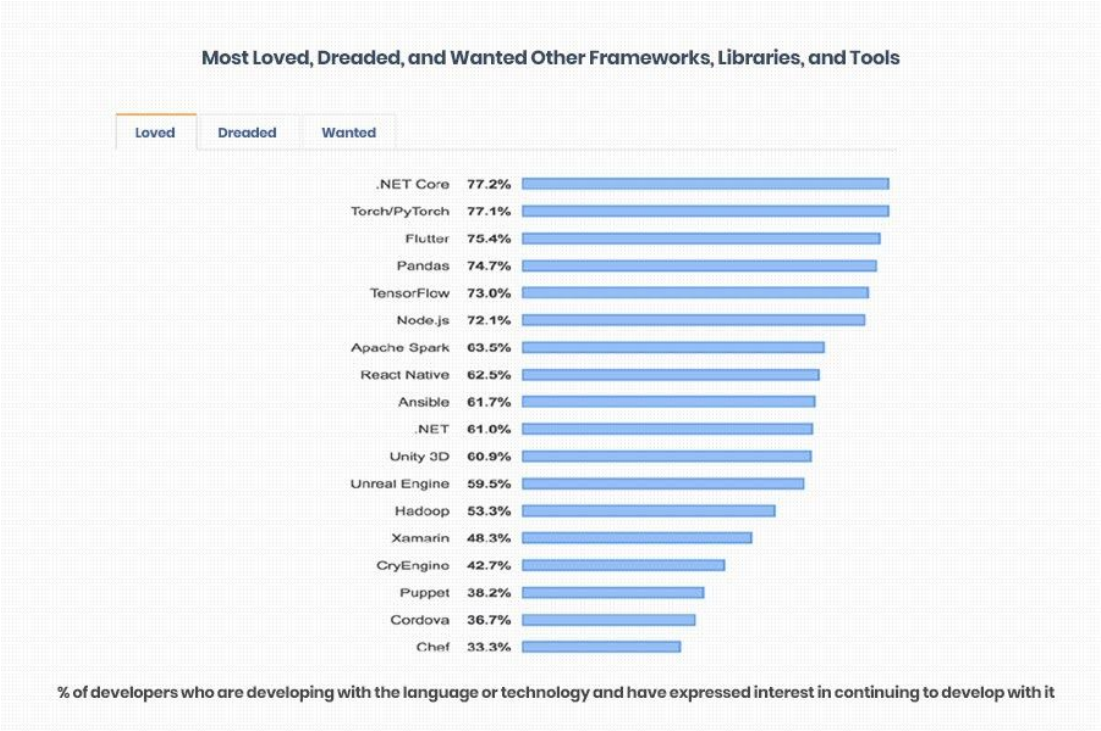


Figure 2.2.1.1 Percentage of developers who express their interest in continuing to develop the framework

2.2.2 Android Studio

Android Studio is developed by Google, it is an official Integrated Development Environment (IDE) for the Android system. Tools from Android Studio help to build mobile applications in a faster and efficient manner.

2.2.3 Firebase

Firebase is a Backend-as-a-Service (BaaS) backend service platform developed by Google. It provides backend services such as realtime database, cloud storage and built-in authentication. As both Flutter and Firebase are developed by Google, they are highly compatible and many built-in functions can be used with ease.

Comparing with AWS, Firebase is younger. fast-growing backend solution. It is more user-friendly to small-scale projects in terms of server management, ready-to-go functionalities and APIs as it avoids complicated setup and server-side management.

In this project, Cloud Firestore is employed as the backend database. It is a cloud-hosted NoSQL database that automatically synchronizes all data in realtime to every user. It supports cross-platform application and offline access [11].

Built-in authentication simplifies the development process of the user authentication system. Cloud Storage is used to store different kinds of data in this project, such as images. Google Analytics [12] is also a featured service provided by Firebase, which enables analyses of user behaviour to enhance application experience.

2.2.4 Algolia

Algolia is a search-as-a-service platform that provides powerful search API supporting all types of text search. The search API is ready to use and highly customizable. After connecting Algolia with Firebase, data will be imported and indexed for search automatically. Self-defined rules such as searchable attributes, ranking and sorting and typo-tolerance can be added according to need [13].

3. Development

3.1 Features

3.1.1 Reply Format

The format of replying in each post is determined by the post creator. For example, if he wants the replies below to be a photo together with some description. Then all the replies in that post would be a photo plus some text. Replies are restricted in a specific format requested by post creator could allow the creator himself and other users to browse the post and retrieve information in a more systematic and clear way.

3.1.2 Interactive Map

The posts which are location-based would be shown in map view. The location appears as a flag marked on the map. The reply will be represented by a marker in the map view. By clicking it, the controller view will be changed accordingly to the card which shows the details of the point of interest. Users can click to view further details.

Also, CUgo asks for the user's current location and the user can expand or minimize the view of replies in the surroundings of the user by changing the radius in the scroll bar.

3.1.3 Rating/ Credibility

A user could give a rating to posts, either a thumbs up or thumbs down. Other users could have a picture on whether the post is useful or not by checking the rating of the post. Post

with high ratings would be considered as a popular and hot topic that would be pushed to the top of the page so that more users could take a look at it.

3.1.4 Privacy

Each user is anonymous while using the app. The personal identity information of a user such as email address will never be disclosed to other users. The user could interact with each other by using his or her username. Anonymous user system allows users to comment and discuss as freely as possible without being traced by others. Inside a post or a reply, only username and timestamp will be shown for recognition.

If the user refuses to allow the app to access the current location services, part of the functions on the map would be disabled. For example, the user cannot locate his/ her own location on the map while browsing a location-based post.

3.2 Functionality

Functions	Description
Search hashtags/keywords in posts	Users can search for desired posts by inputting single or multiple keywords on a search bar after clicking the search button at the top search bar. Hashtag function is added in each post for filtering out the unwanted posts when the user searches for certain keywords.
Create a post	Users can create a post by clicking the add button at the bottom right corner. The application will redirect to the create post page to allow users to input the post's title, content, hashtag and reply format.
Customize reply format	Users need to specify a customized reply format when creating a new post. The application provides various components such as text field, image and location. Users can assemble the reply format by simply clicking "+" to add, "-" to remove the component.

Reply a post	For text-based posts, users can reply to a post by completing the reply form where users need to input according to the standardized reply format of that post. For location-based posts, users can reply to the post by clicking the “add reply” button on top of the map view. Users will then be redirected to a place autocomplete page to fill in a location. After that, the user will need to complete the reply form according to the standardized reply format of that post.
Browse a post	Posts are classified into categories. Users can browse posts from different categories by clicking the button in the top navigation bar. Replies of the post will be displayed in the standardized format of that post.
Vote for authenticity	There is an authenticity voting system for each topic and reply. Users can vote for the authenticity of that topic or reply according to their own knowing. This makes use of the collaborative effort and crowd wisdom to determine the authenticity of the source.

Recommend Topic	The “For You” category provides a list of recommended posts tailor-made for each user. System analysis user’s indicated interest to generate the set of recommended posts.
Add to Bookmark	Users can add posts into My Favourite by clicking the star button on each post. Users can always refer to their list of favourite posts in the “Bookmark” inside the side menu.

3.3 Architecture Design

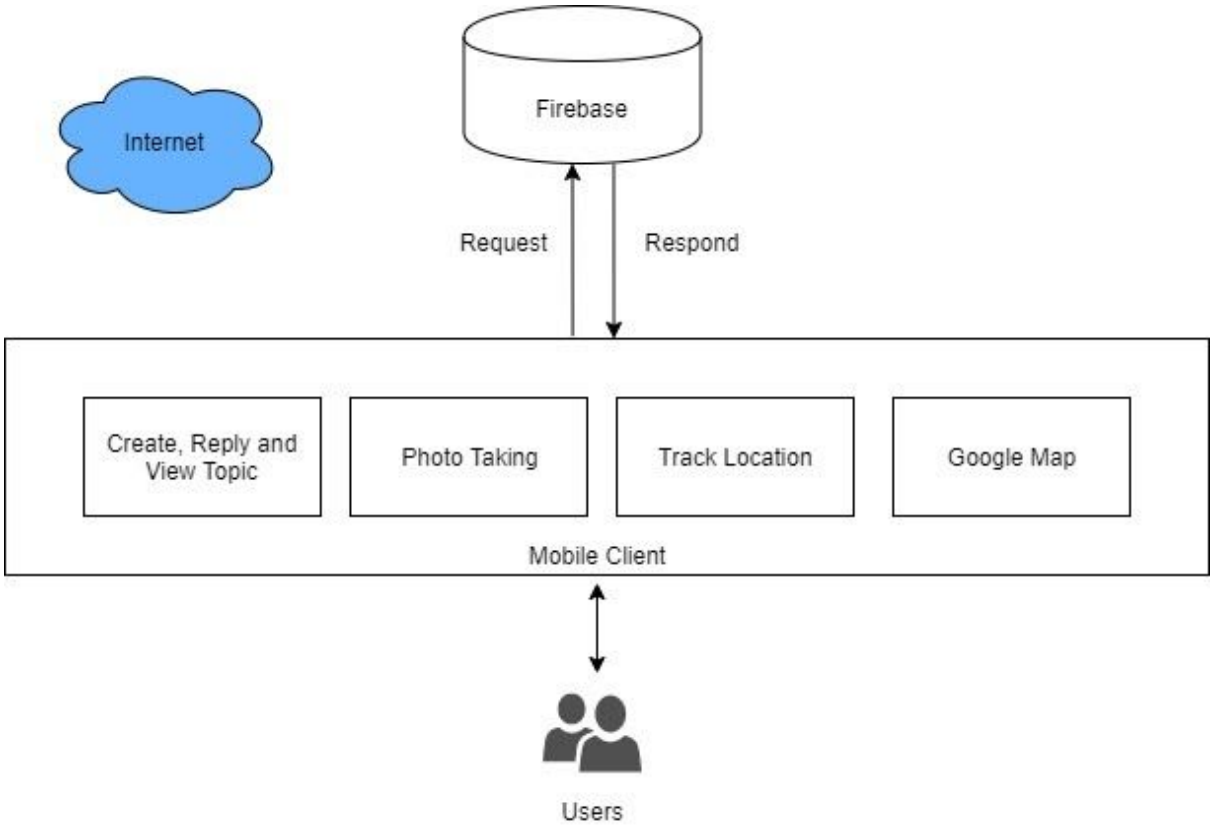


Fig. 3.3.1 Architecture Design Diagram

CUgo consists of two main parts, the mobile client and the database. Main functions such as create, reply and view topics require communication between database and mobile client. Because CUgo is a forum-like fact check platform, data are expected to be updated frequently, users are expected to be connected to the internet most of the time for synchronization of data. Meanwhile, Firebase also supports offline data persistence by caching the data from the database and synchronizes the data once connected [14].

The application also makes use of the photo-taking function in mobile clients to upload images as proof in related topics for the fact check purpose.

One of the features of CUgo is location-based fact checking, with the use of Google Map and track location function in mobile clients, it enables the application to show nearby replies and pin a new reply on the map on a topic.

3.4 Use Case Diagram

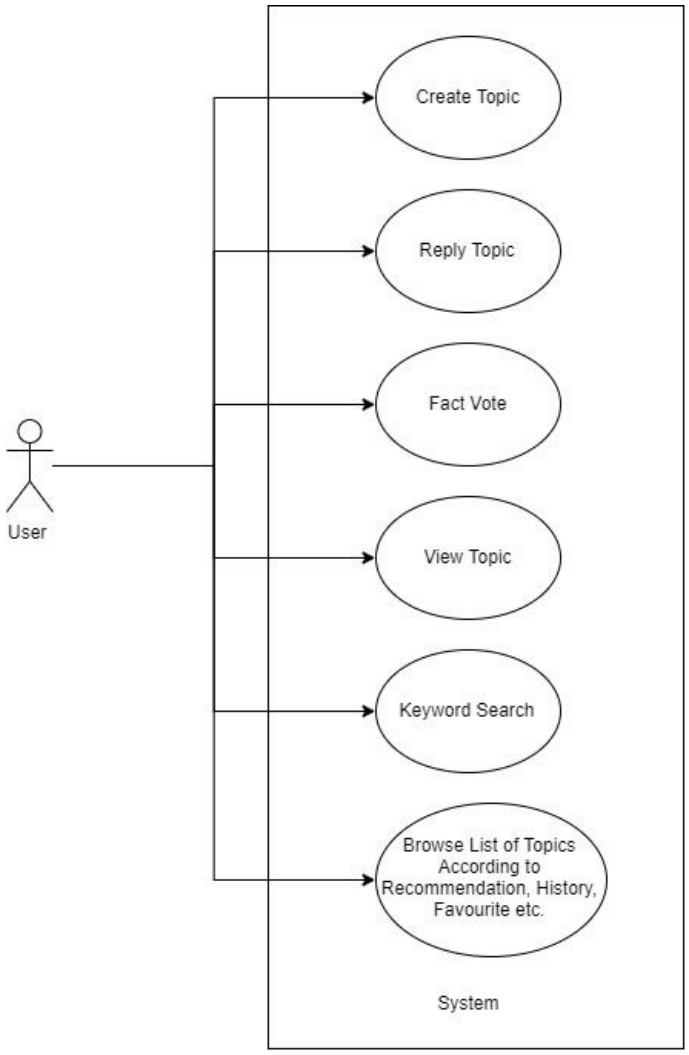


Fig. 3.4.1 Use Case Diagram

There are several main functionalities in CUgo, such as create, view and reply topics, fact vote for the credibility of a reply, perform keyword searching and browse the list of topics according to different criteria like categories, recommendation, recent history, my favourite etc. These functionalities act as basic components to build up the entire fact-check platform.

3.5 Data Flow Diagram

3.5.1 Authentication and Profile System

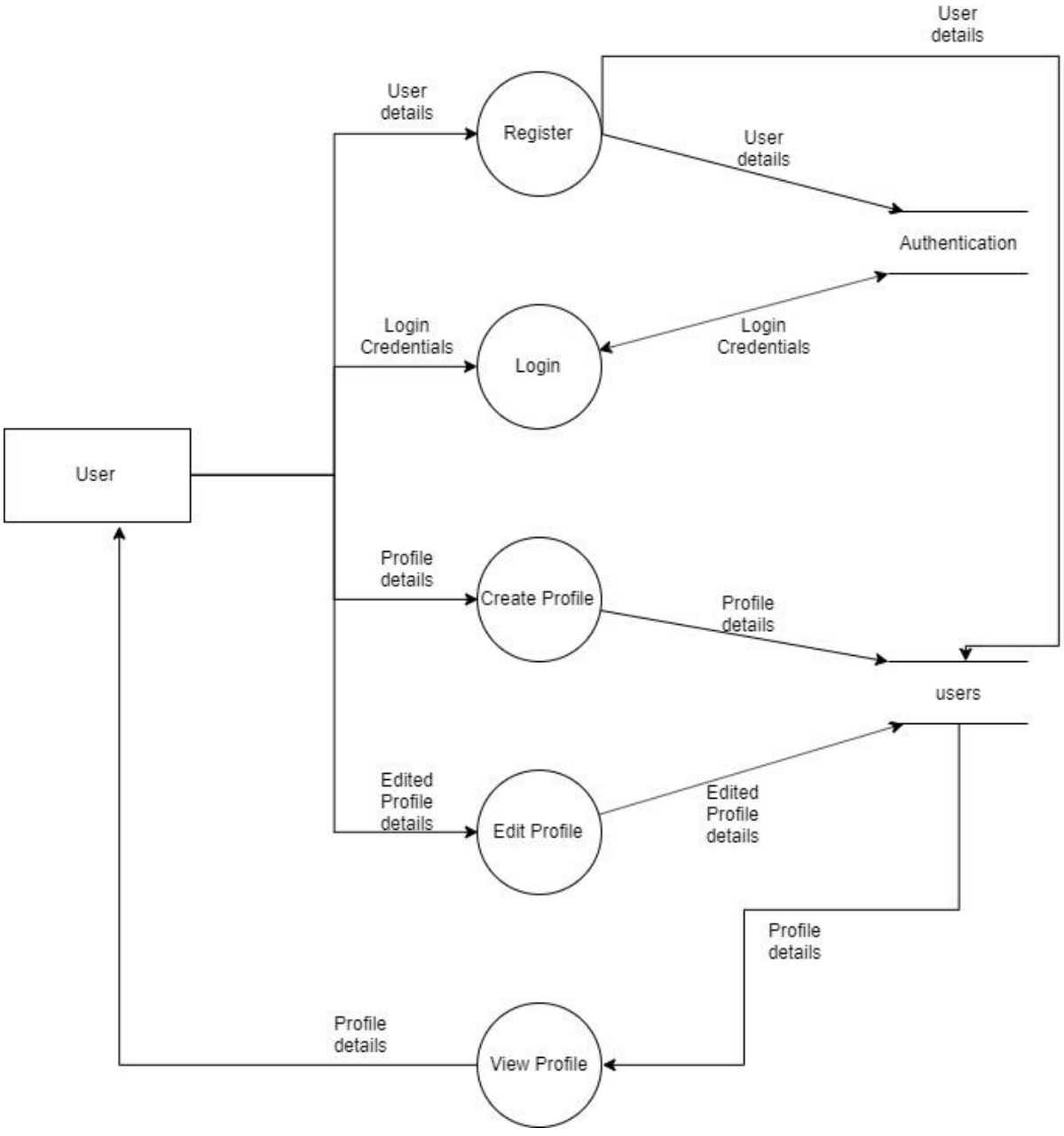


Fig. 3.5.1.1 Authentication and profile system data flow diagram

This diagram demonstrates the data flow of authentication and profile system. It is worth mentioning that the authentication database is separated from the users database as the application makes use of the built-in authentication system of Firebase which will

automatically store the login information in an authentication database, which is separated from all other databases.

1. Register

   Register function requires the user to input user details such as username, email and password. After passing them to the authentication database and passing the email authentication step, a new account is set up. User details such as username and email will also be sent to the users database to set up a new user profile.

2. Login

   Login function requires the user to input login credentials such as username and password. This information will be passed to the authentication database for authentication check, the result will be passed back indicating whether the login is successful or not.

3. Create a Profile

   Create a Profile is performed after a successful registration. Users need to input profile details such as the topics they are interested in. Data will be stored in the users database.

4. Edit Profile

   Edit Profile allows the user to update user profile information into the users database. Users are allowed to change their username and their interests declared.

5. View Profile

   In view profile function, profile details will be retrieved from users database.

3.5.2 Main system

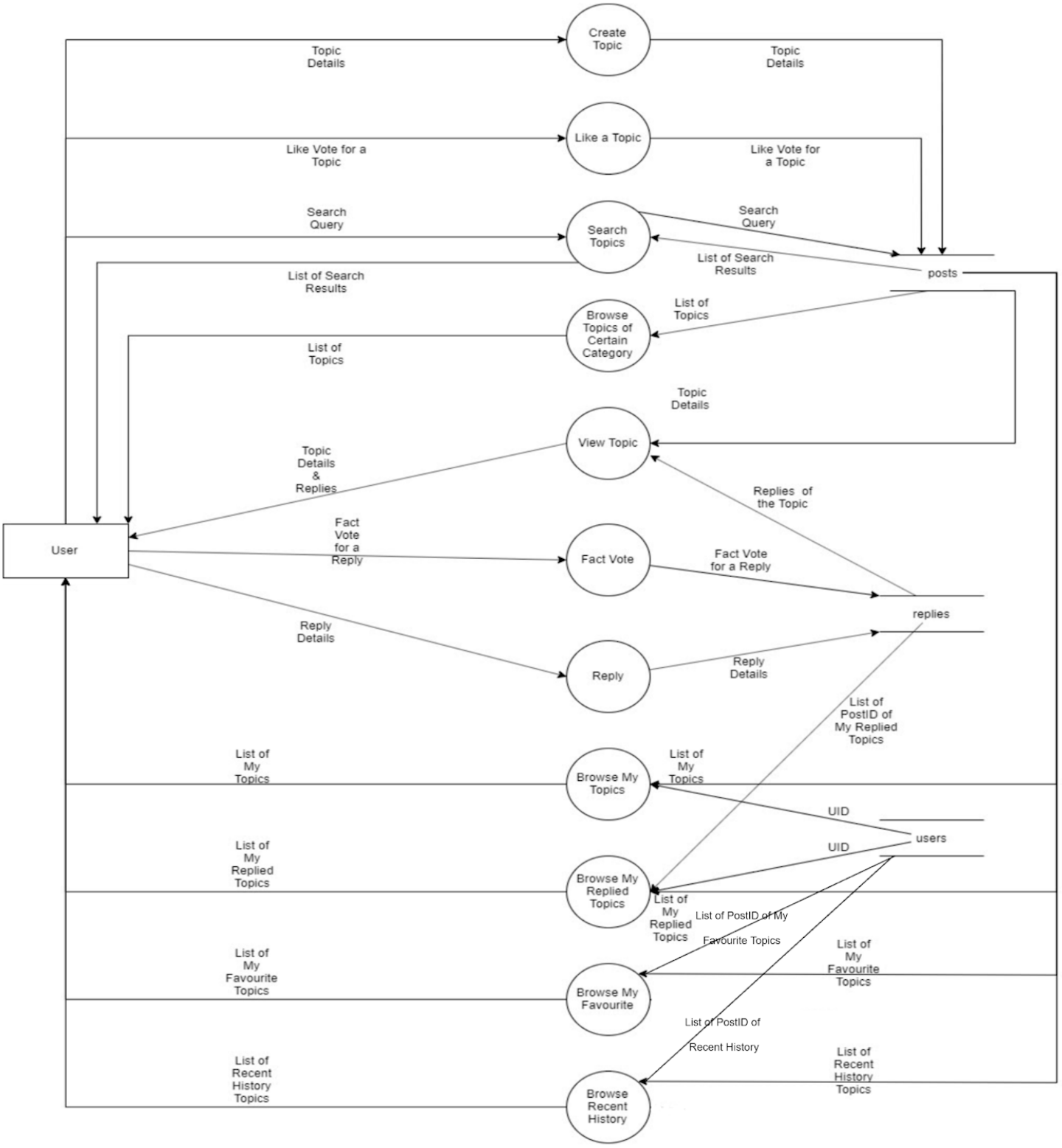


Fig 3.5.2.1 Main system data flow diagram

This diagram is to show the data flow in the main system of CUgo. There are six databases used in total. The main system is basically classified into eleven main functions.

1. Create a Topic

   When a user creates a new topic, topic details input by the user such as topic title, body, reply format and other automatically generated data such as author, create time will be passed to posts database.

2. Upvote a Topic

   Users can upvote or downvote a topic. The vote count will be stored and updated in the posts database.

3. Search Topics

   Users can perform a keyword search on the topic title. A search query will be passed to the posts database. After performing the search query, a list of search results will be passed back.

4. Browse Topics of Certain Category

   When the user browses topics in a certain category, the query will be done in the posts database to retrieve the list of topics in that category.

5. View Topic

   When a user views a topic, topics details from posts database and replies of that topic from replies database will be shown.

6. Fact Vote

   Users can vote for each reply by clicking the upvote or downvote button. The vote will then be stored and updated in the replies database. This function serves as an important component for the fact check feature.

7. Reply

   User inputs along with generated information such as author, reply time will be passed into replies database.

8. Browse My Topics

   Users can browse the topics he/she created in this function. The query will be done between users database and posts database by matching the authorID with UID.

9. Browse My Replied Topics

   Users can browse the topics he/she replied in this function. The query will be done between users database and replies database by matching the authorID of replies with UID to generate a list of postID. The list of postID will be used to retrieve the replied topics from the posts database.

10. Browse My Favourite

    When a user browses the My Favourite section, the system will get a list of postID of My Favourite topics from users database and query with posts database to return a list of favourite topics.

11. Browse Recent History

    When a user browses the Recent History section, the system will get a list of postID of Recent History topics from users database and query with posts database to return a list of recently visited topics.

3.6 Database Schema

3.6.1 Overall Structure

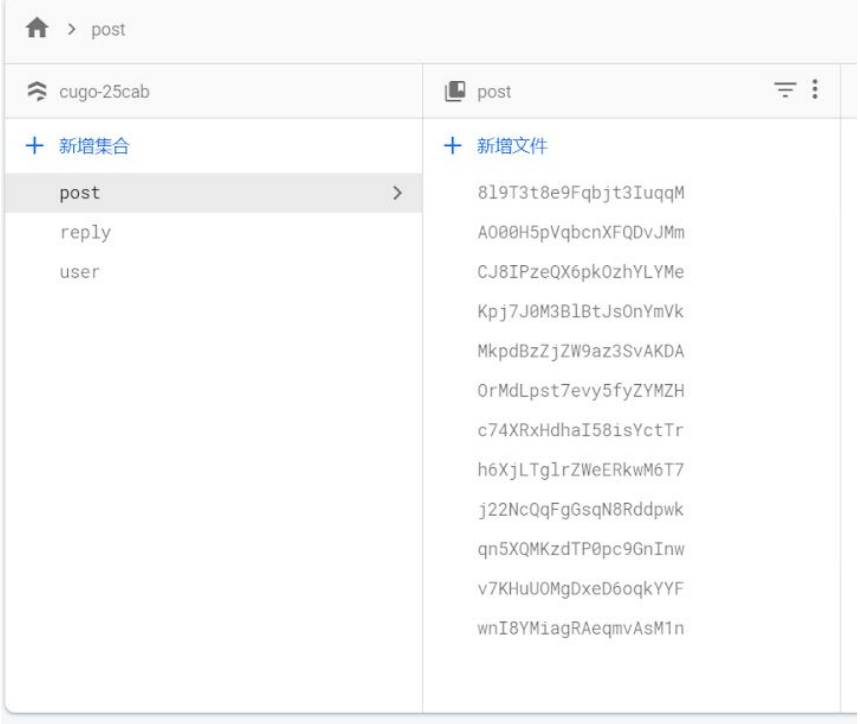


Fig. 3.6.1.1 Database schema

There are four main databases in the database system of CUgo. Posts database stores the topic details such as title, content, author, reply format, hashtags, etc. of each topic. Replies database stores reply details of every reply such as content, author, etc., grouped by each topic. Users database stores user details of each user including username, email, interests, recent history, bookmark, etc. While the last database is the authentication database.

3.6.2 Authentication

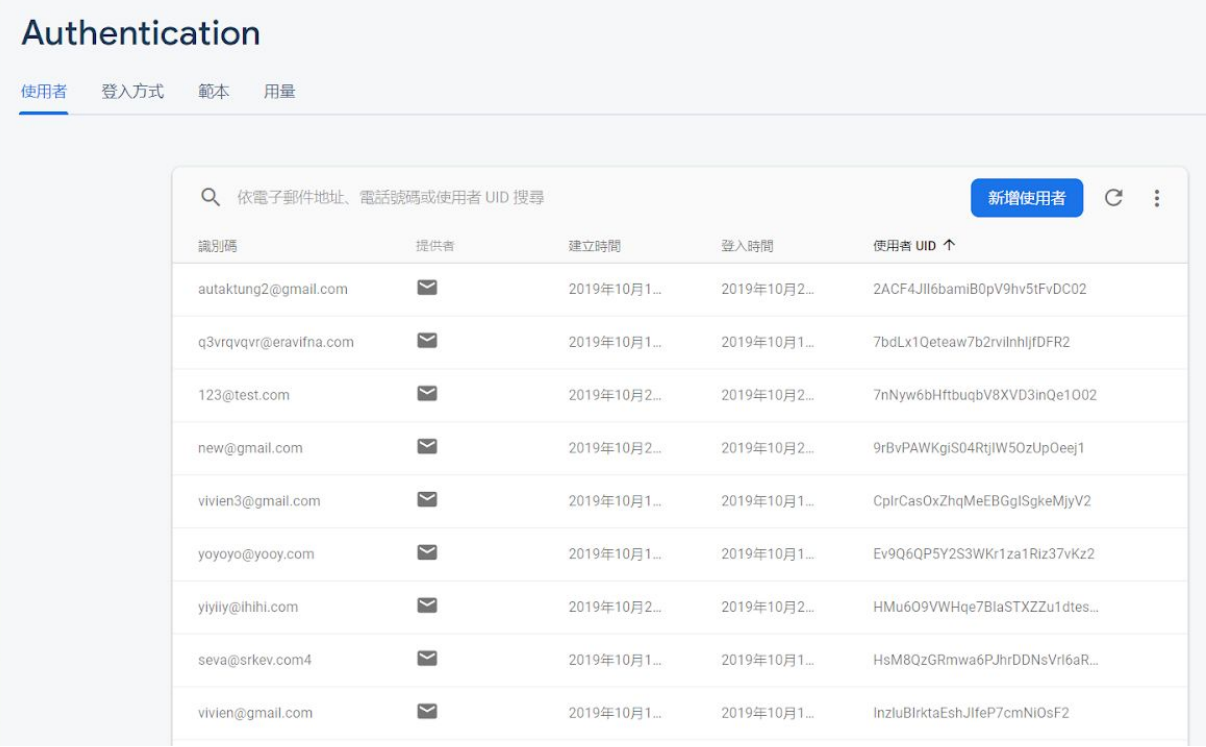


Fig. 3.6.2.1 Authentication database schema

Authentication database stores the email, authentication provider, created time, last login time, and UID of an account. It is a component of the Firebase built-in authentication system.

3.6.3 Post Database

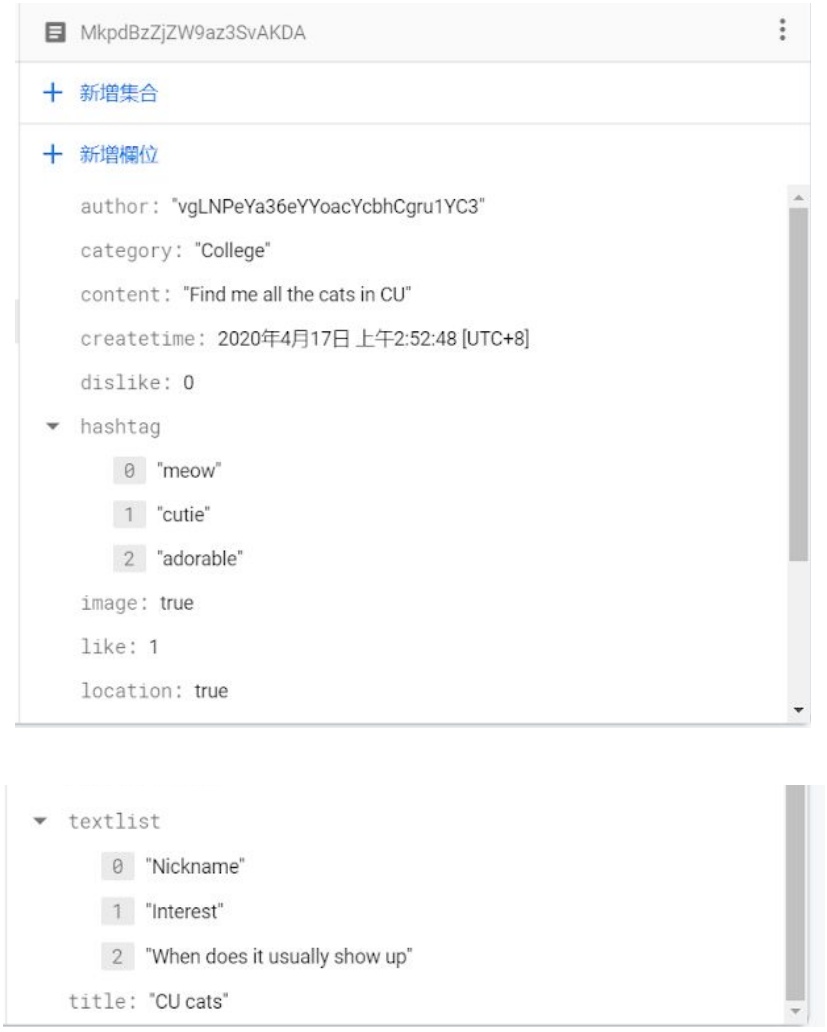


Fig. 3.6.3.1 Post database schema

There is a collection of posts in the post database, each with a unique post ID. In each post record, the author, category, title, content, create time, upvote, downvote, hashtag, and reply format of that post will be stored respectively. Reply format refers to a combination of image, location, and text list.

Text list is a string array storing the field name of the corresponding text field, the length of the array indicates how many text fields will be shown on the reply form of that post. Image

is a boolean value indicating whether a reply in this post should include an image. If the value is true, an image uploader will be shown on the reply form of that post. Location is a boolean value indicating whether a reply in this post should include a location. If the value is true, a google map will be shown inside that post, and users are required to input a location when they reply.

For example, the above post is a location-based post and each of its replies contains a location, an image, and three text fields.

3.6.4 Reply Database

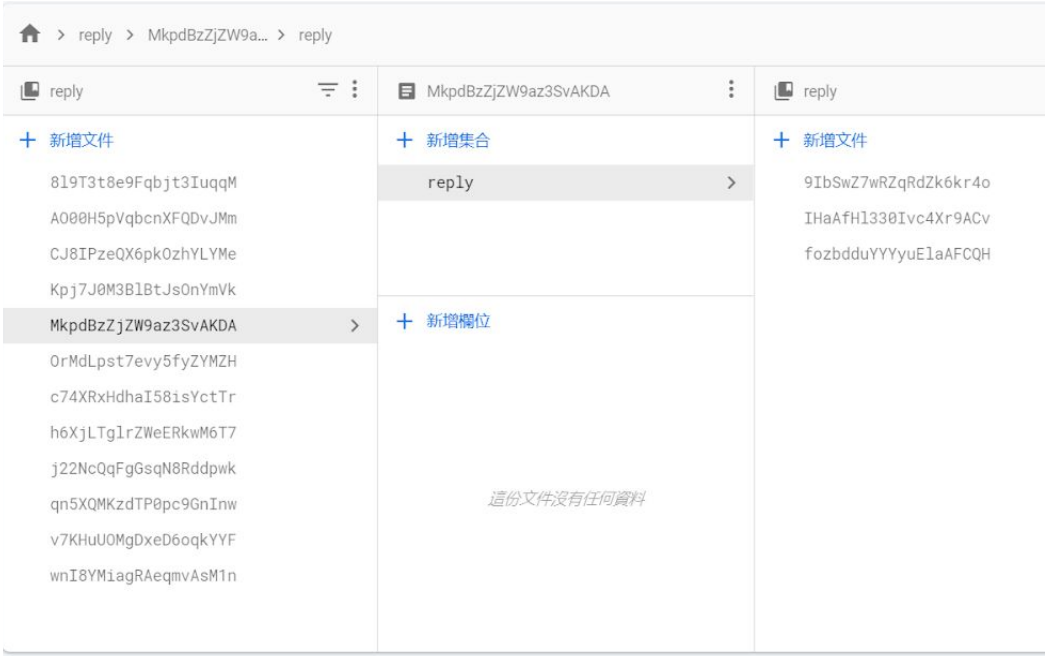


Fig. 3.6.4.1 Reply database schema

Replies of each post are being stored in the Reply database. Each post is represented by a post ID, a collection of replies of that post are being stored under the corresponding post ID. Each reply has its own unique replyID.

Fig. 3.6.4.2 Reply database schema

Inside each reply, it stores the author name, author ID, time of creation, number of upvotes and downvotes, and the formatted reply body as explained in the reply format section in 3.6.3. For instance, this is a reply under the post in 3.6.3, in which the reply format is three text fields, an image, and a location. This reply body includes three text replies, an image URL, and a location represented by latitude, longitude, and place ID which fit in the reply format of the corresponding post.

3.6.5 User Database

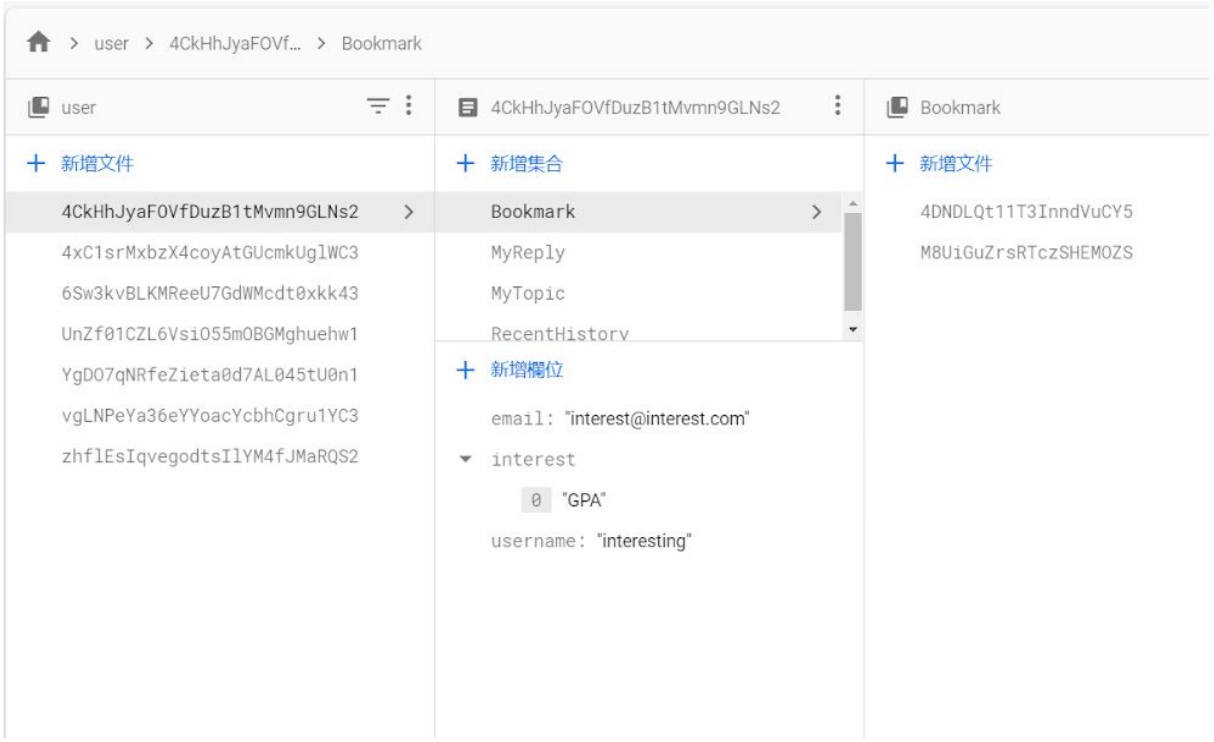


Fig. 3.6.5.1 User database schema

The User database stores user details for each user. In the database, each user is represented by a UID. Within each UID, email, interest, username, bookmark, my reply, my topic, and the recent history are being stored. Note that a string array is used to store user interest. While bookmark, my reply, my topic, and the recent history of each user are being stored in four collections under each user record.

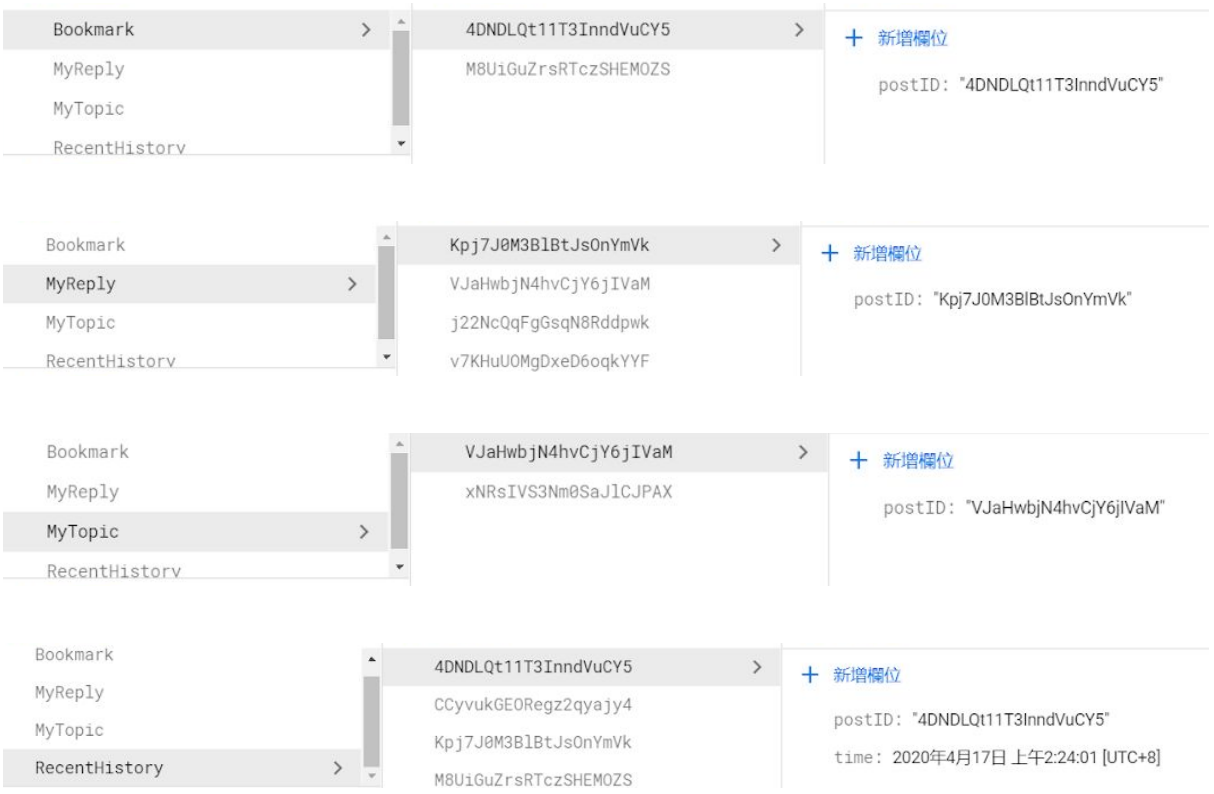


Fig. 3.6.5.2 Schema of bookmark, my reply, my topic, and recent history collection

The schemas of bookmark, my reply, my topic, and recent history collection are similar. Each of the bookmarks, my reply, and my topic records stores a post ID, while a recent history record stores a post ID and the browsing time. For example, this user has two bookmarked posts, replied to four posts, created two posts, and recently browsed four posts.

3.7 Recommendation Function

CUgo suggests posts for users according to their interest. Upon successful registration, the interest of a user is set to “Not selected” originally. Users can edit interest by accessing the Profile page. After the change in interest was declared, the category “For you” under the home page will show the posts related to the interest selected.

The recommendation is made based on the string-matching method. The interest selected will be matched to one of the categories listed below.

Interest	Category
Seminar	Study
Study Location	
GPA	

College Culture	College
Orientation Camp	
Hostel Life	

CU cats	Facilities
Photo Spots	
Flowers at CUHK	

Intern	Career
Grad Job	
Apt Test	
Job Hack	

3.8 Keyword Searching function

The keyword search function is powered by Algolia search engine. The keyword search function enables post title, content and hashtags to be the searchable attributes used for keyword searching. When the keyword appears in the title or content or hashtags of a specific post, that post will be returned as a search result. List of search results will then be ranked and sorted according to several criteria as follows.

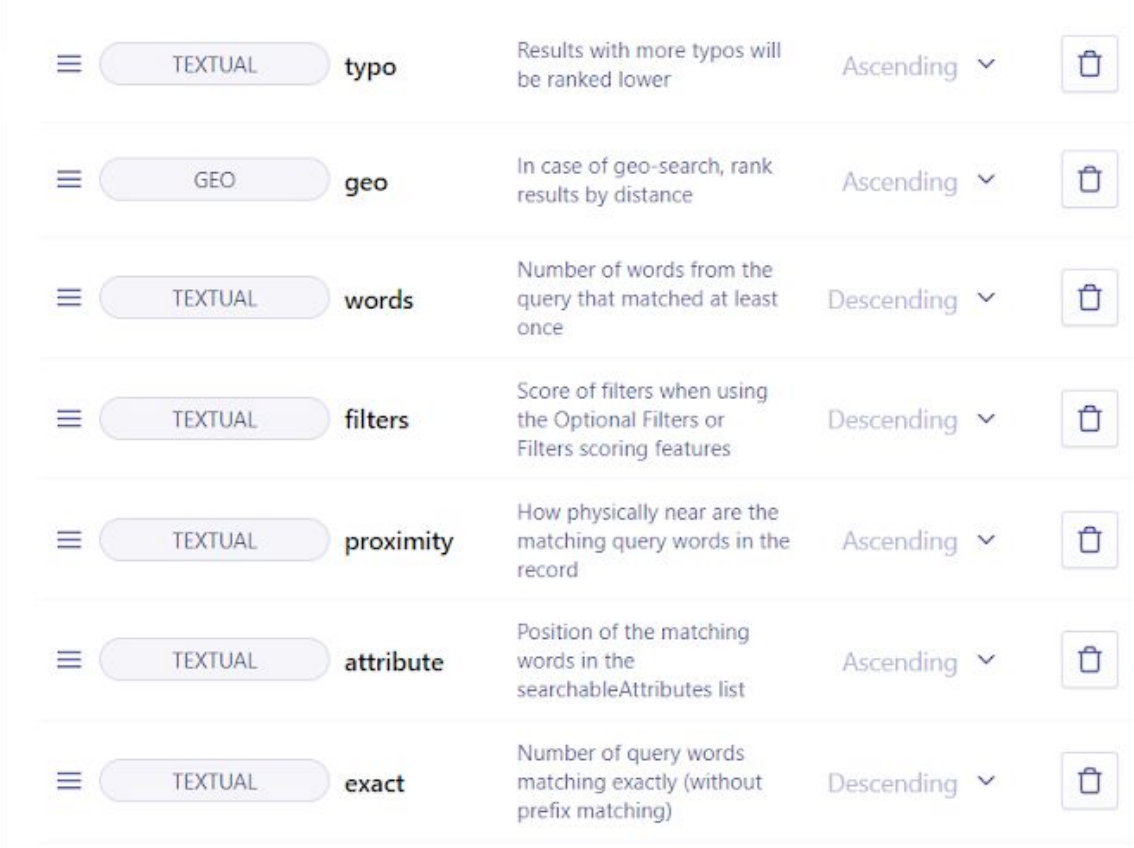


Fig. 3.8.1 Algolia ranking and sorting

The list of results will be ranked and sorted according to their relevance. Algolia is also typo-tolerance in the keyword search. Through implementing keyword search, users can search for their desirable posts and browse posts with ease.

4. Fact-checking

4.1 Credibility System

CUgo determines the credibility or trustworthiness of a certain post or reply by the credibility system. It is a rating shown in each post and reply. In the homepage of CUgo, under the tab "All", the posts will be displayed according to the credibility of each post. Same for the reply in each post, they rank according to the number of upvotes. Posts and replies with higher credibility will be shown at the top.

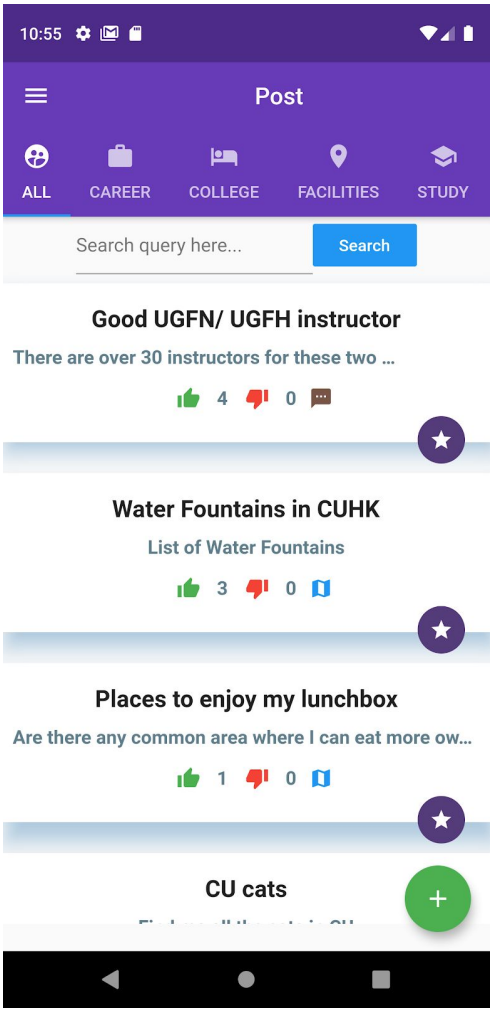


Fig. 4.1.1 Rating in a post

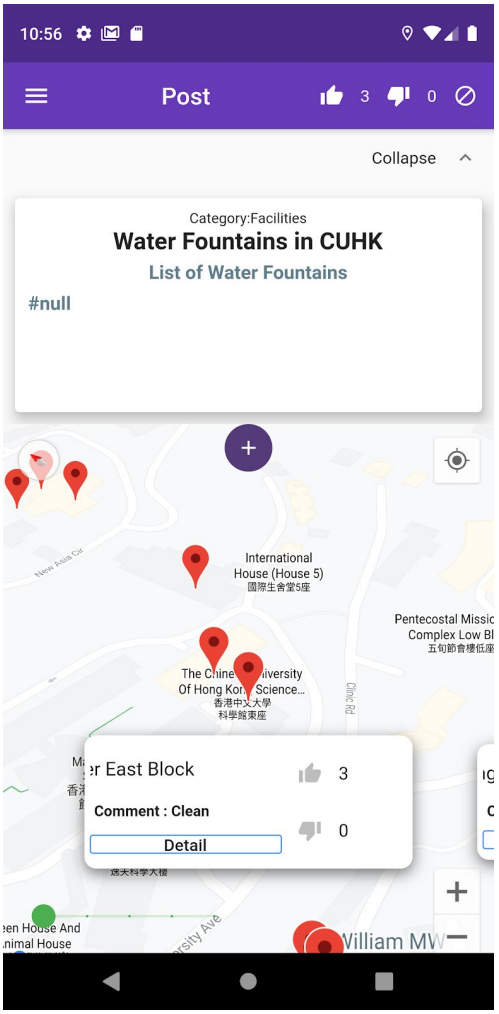


Fig. 4.1.2 Rating in a location-based reply

4.2 Fact-check Scenario

In this scenario, a user wants to look for the information of CU cats. He is neither the owner nor one of the commenters in this post. He clicks into the post in order to find the information he needs.

Step 1

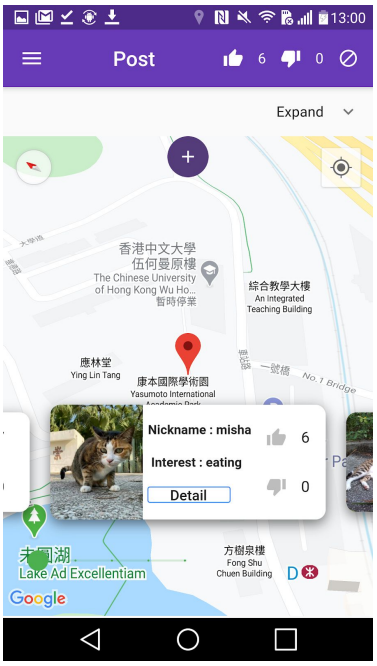


Fig 4.2.1 Map view of the post CU cats

Step 2

Then, he realises that the first three cats have 80-100% of upvote. While the others receive a high proportion of downvote. He can assume that the first three cats are actually the cats that live in CUHK since the downvote represents that the reply does not actually echo with the topic.

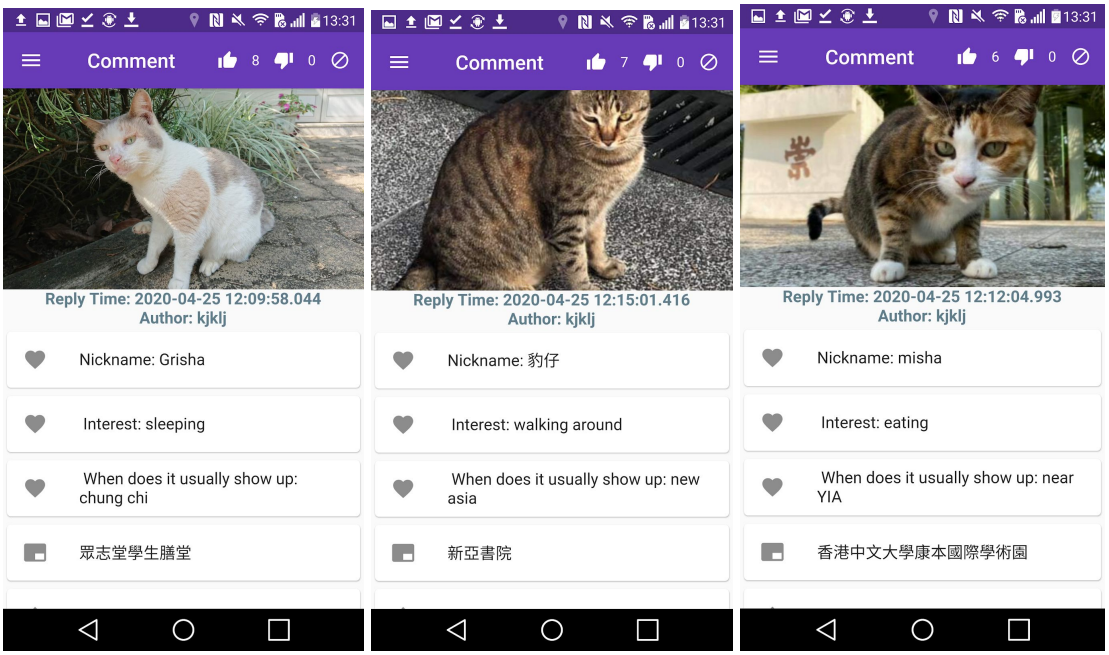


Fig 4.2.2, 4.2.3, 4.2.4 The detailed reply page of the three cats with the most upvote

4.3 Further application

There are a number of real-life scenarios that can be solved with the aid of CUgo. However, due to the limitation of full access to CUHK during this COVID-19 period, the scenarios are kept without an actual experiment.

What makes CUgo stand out from the crowd is the fact that it allows fact-checking with photos. Therefore, news which can be verified by photo is particularly useful and suitable for CUgo. There are several further examples of application which could be verified more easily with the help of CUgo. One of the examples is taking a photo of a certain location. For example, it is heard that there are more suspect cases of illegal felling of trees in CUHK, including valuable Aquilaria. Users can help to take photos of the location where they know

there is an Aquilaria planted. Then, a database can be set up and users will know whether there is an Aquilaria missing in CUHK.

Other examples like taking photos of the interior situation of a lift and the construction site which caused temporary close-off of the road will help to verify rumours as well. The photo function of CUgo comes into handy.

5. Application Implementation

5.1 Login and User Profile System

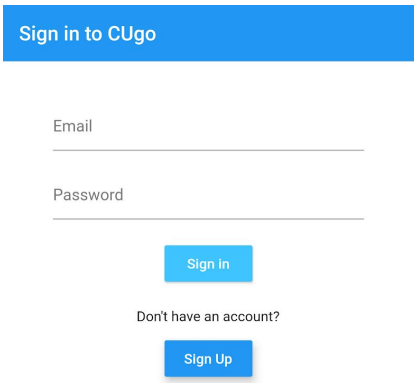


Fig. 5.1.1 Sign-in page of CUgo

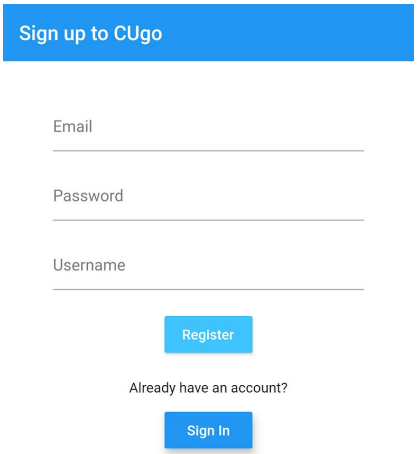


Fig. 5.1.2 Create New Account

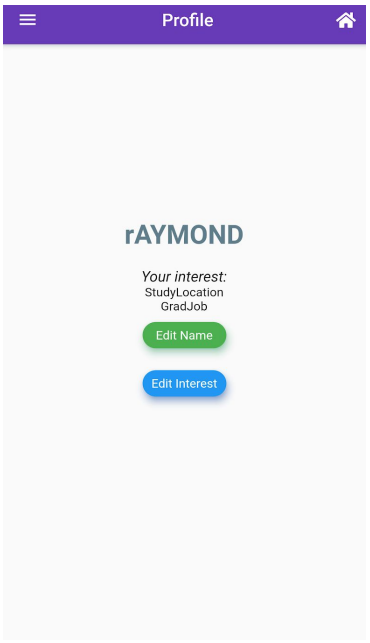


Fig. 5.1.3 Profile of a user

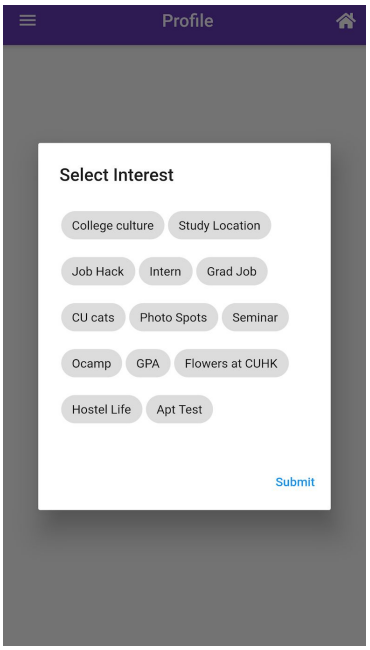


Fig.5.1.4 Select/ Change indicated interests

5.2 Homepage, Menu and Category

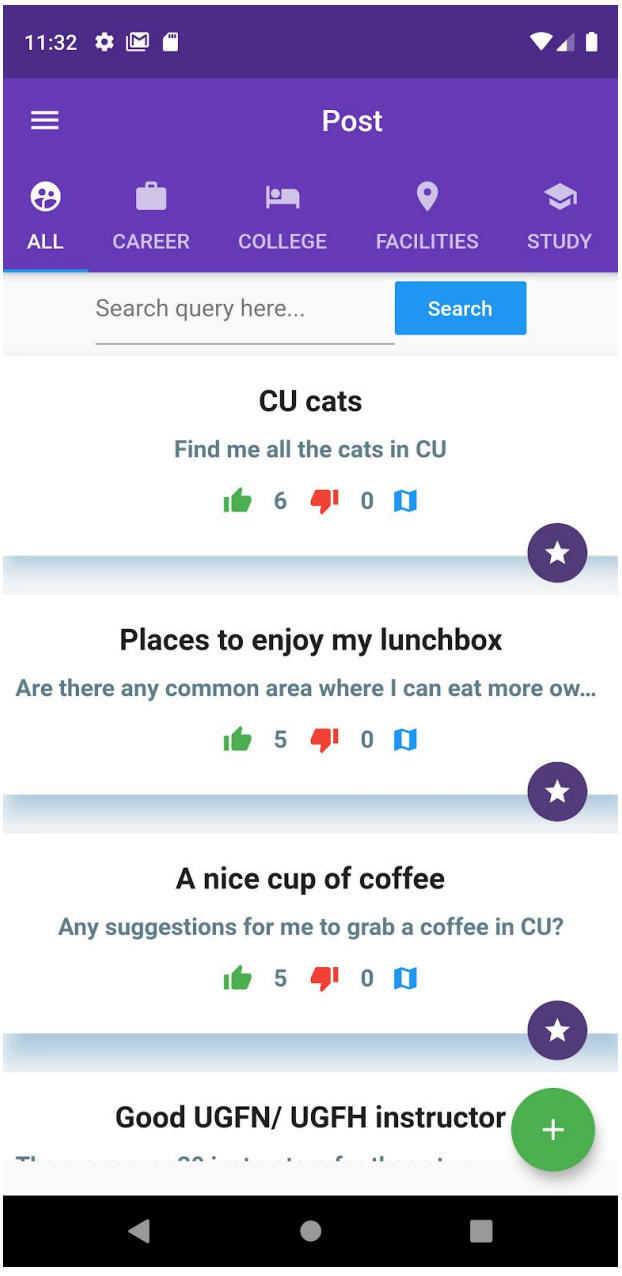


Fig. 5.2.1 Homepage and category

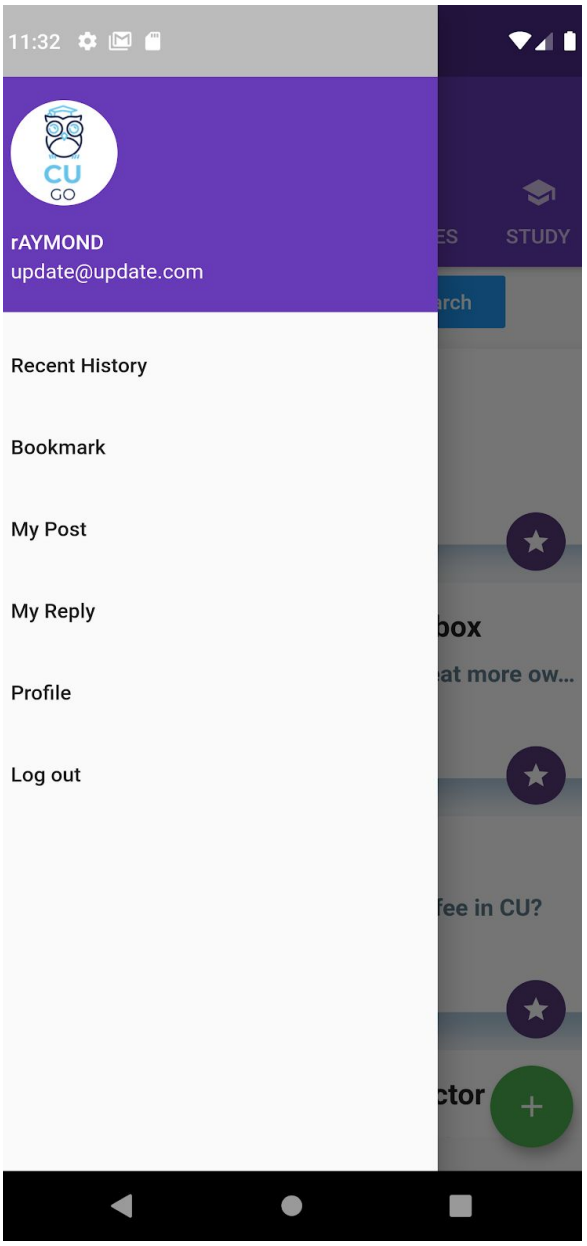


Fig. 5.2.2 Sidebar Menu

5.3 Browse Topic

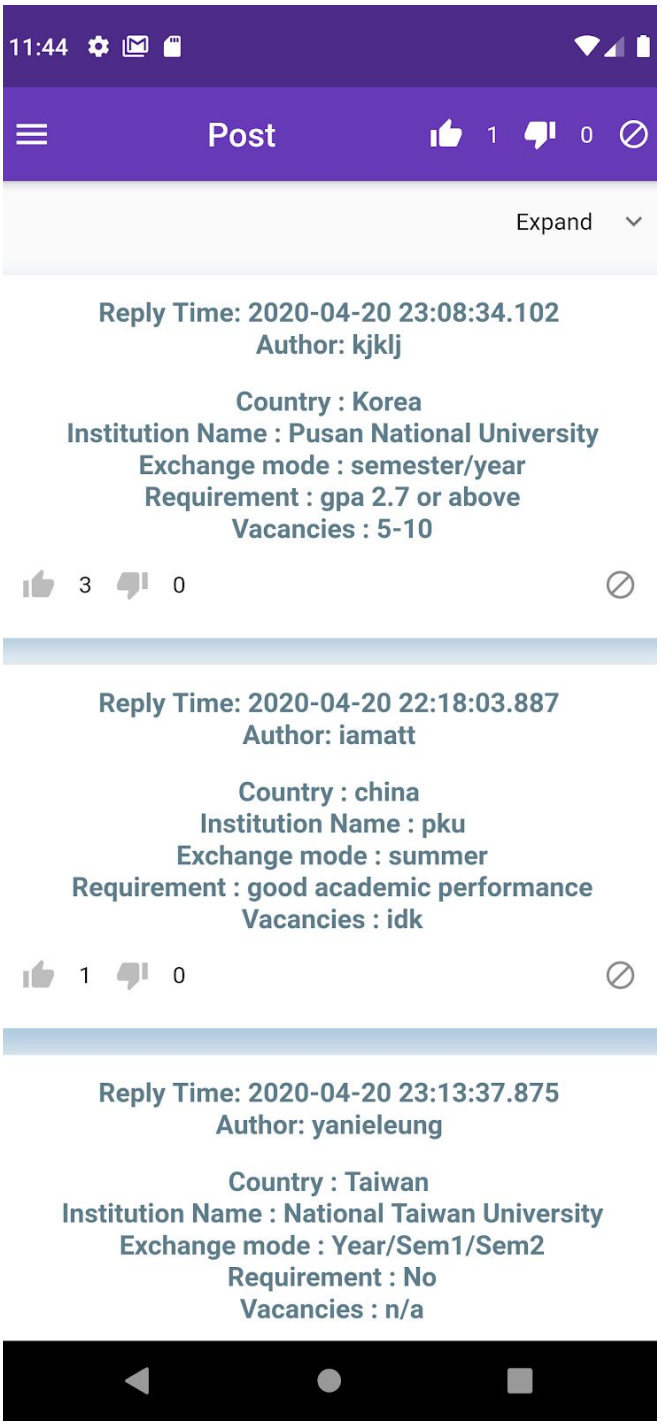


Fig. 5.3.1 Browse a non-location-based topic

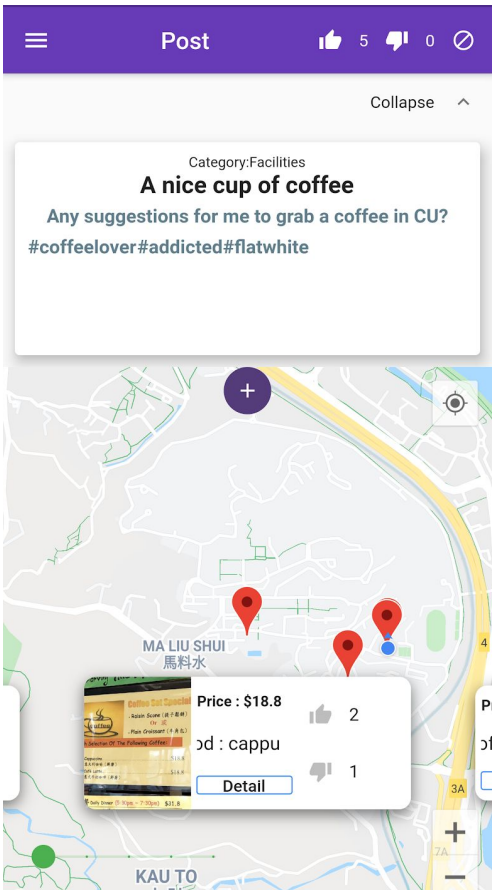


Fig. 5.3.2 Browse location-based topic

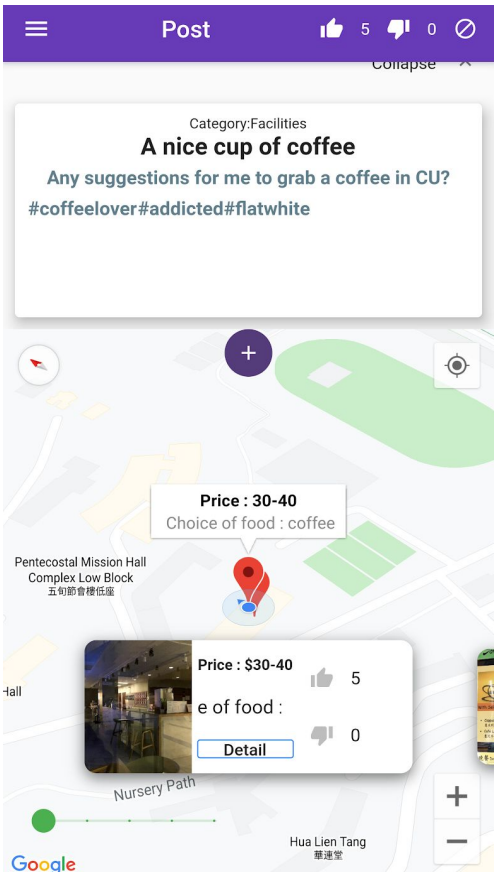


Fig. 5.3.3 View information window after clicking the marker

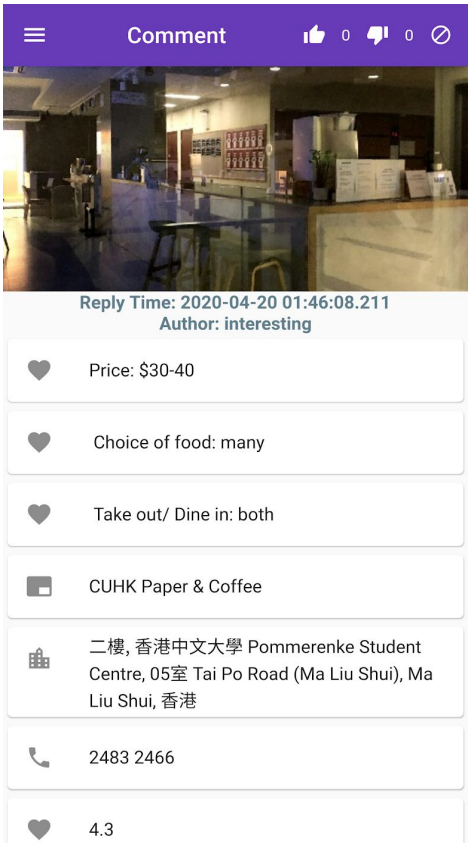


Fig. 5.3.4 Browse further details of the spot

5.4 Reply Topic

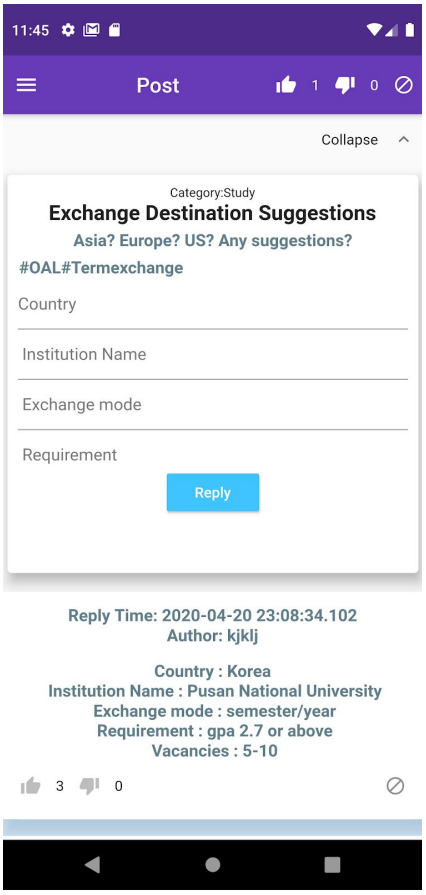


Fig. 5.4.1 Add a new reply in non-location-based post

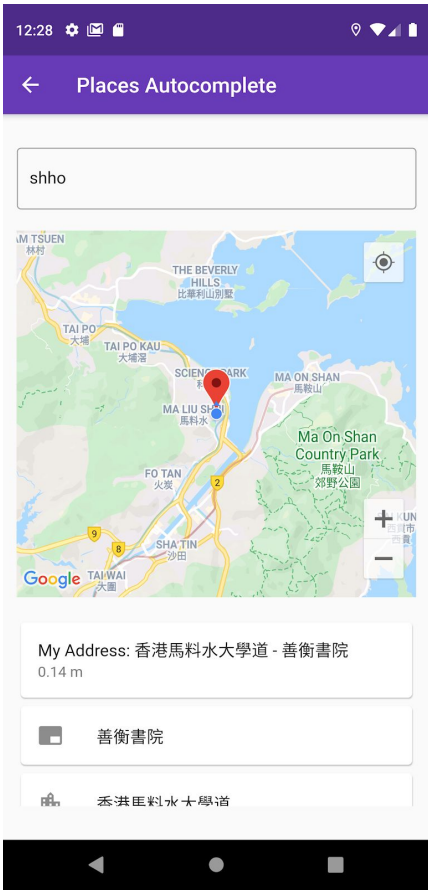


Fig. 5.4.2 Add a new location

5.5 Create Topic

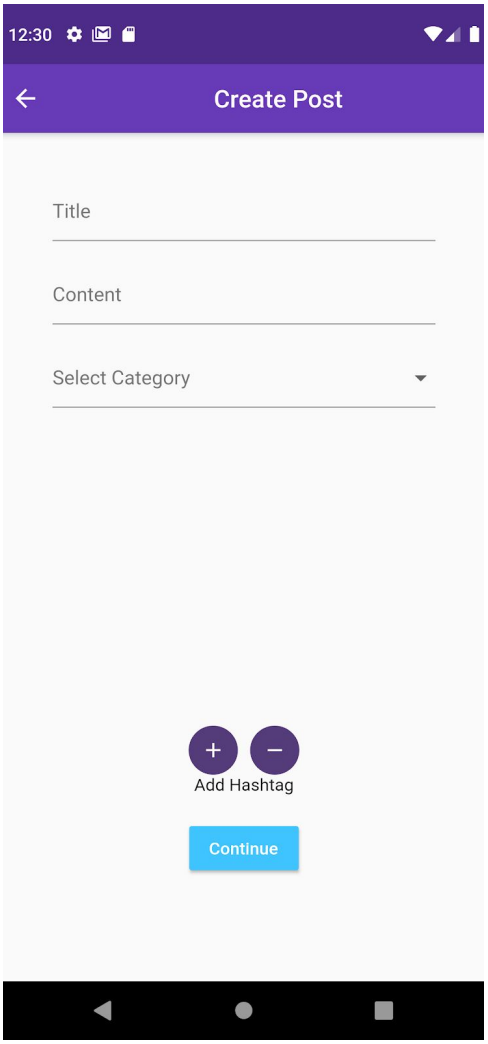


Fig. 5.5.1 Create new post

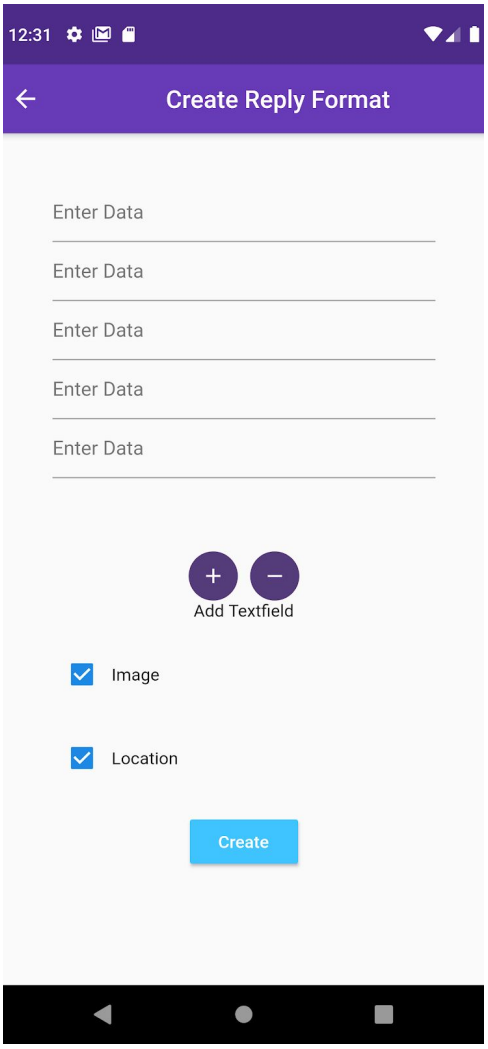


Fig. 5.5.2 Drag-and-drop components for constructing reply format

6. Procedure Flow

6.1 Flow Chart

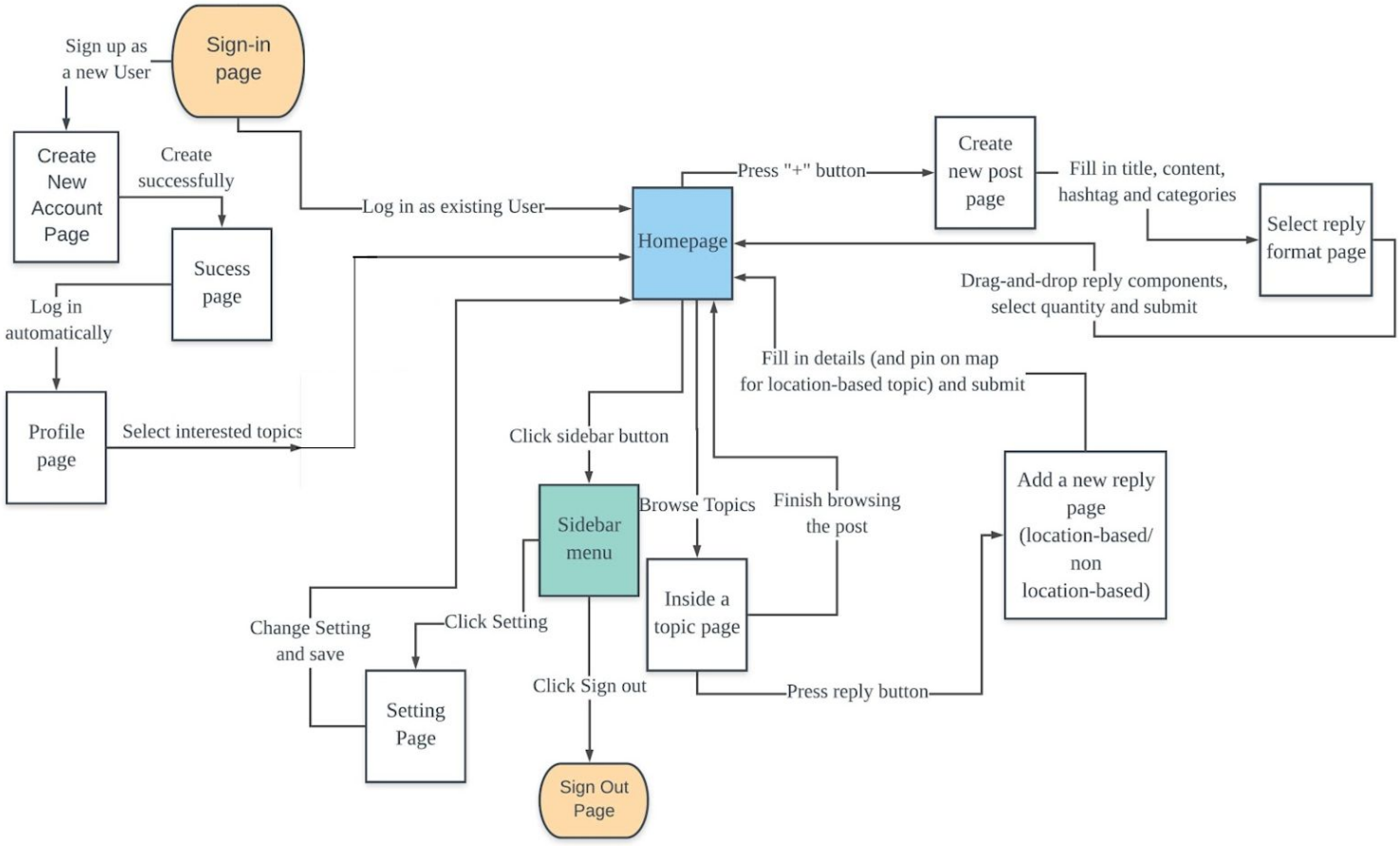


Fig. 6.1.1 Page flow of CUgo

The complete flow of pages start from our sign-in page and end with sign-out page. Upon executing the action named on the arrow, the user will be directed to the pages pointed by the corresponding arrows. The homepage is coloured as blue as it is the main page of our application. Most of the paths in this chart contain the homepage. Also, the page of the

sidebar menu is coloured in green since there are many buttons in the menu. Users can be redirected to different pages such as My history and My topic apart from setting page and sign out page mentioned in the chart. The details of the sidebar menu are omitted here.

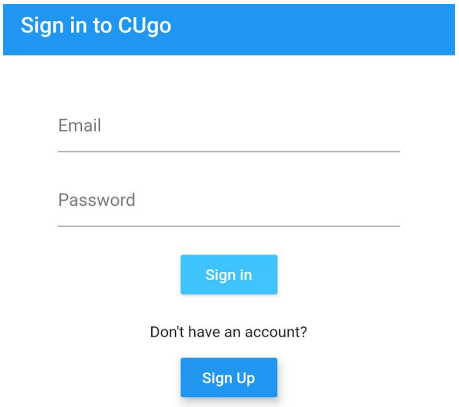
6.2 Scenario

6.2.1 Scenario A

In this scenario, user Tom registers as a new user. Upon successful registration, he browses his profile and edits his interest first. He then creates a post to ask for information about the UG course instructor suggestion.

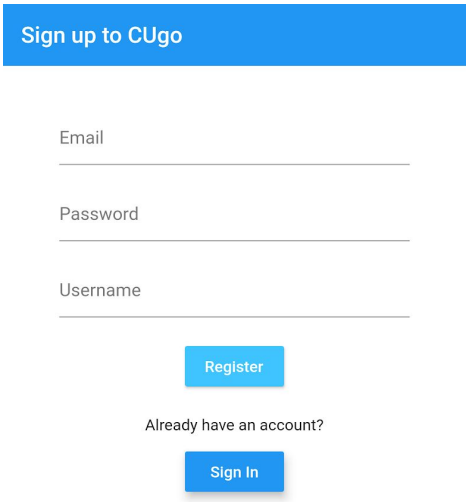
Step 1

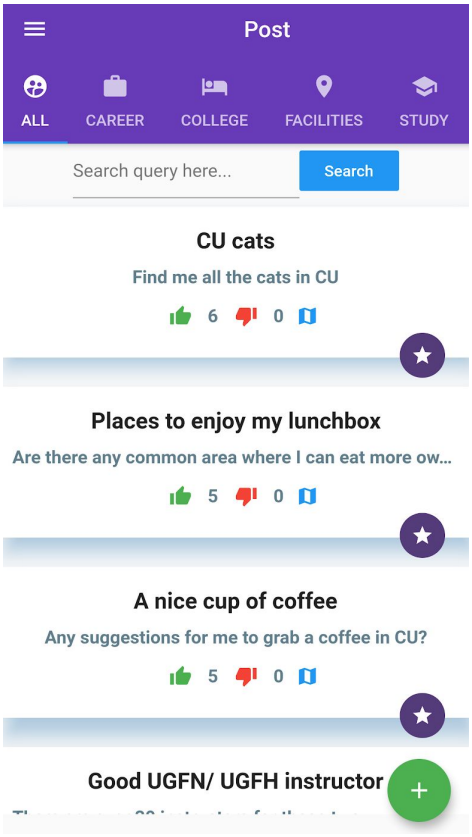
On the Sign In page of CUgo, Tom presses the "Sign Up" button at the bottom to create an account.



Step 2

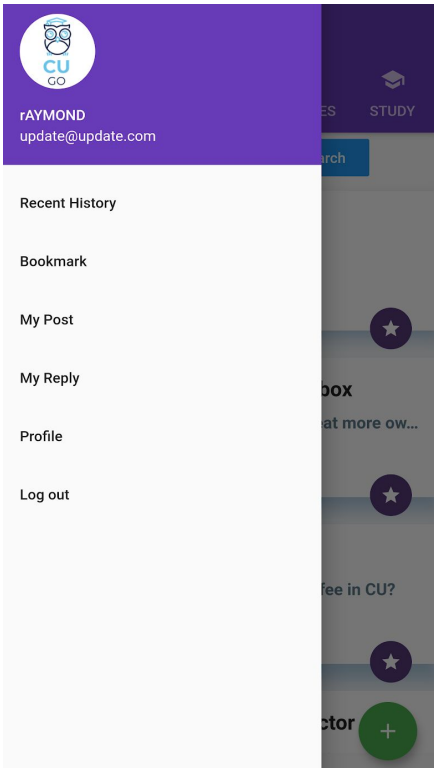
After inputting the required information, Tom creates an account successfully. CUgo helps him sign in automatically.

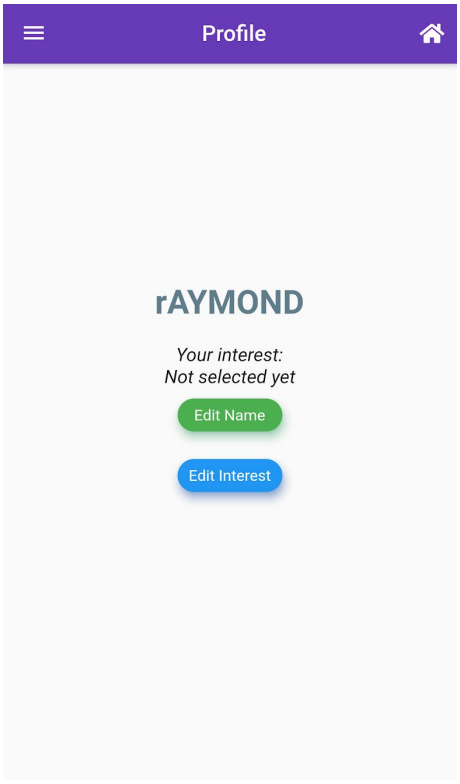


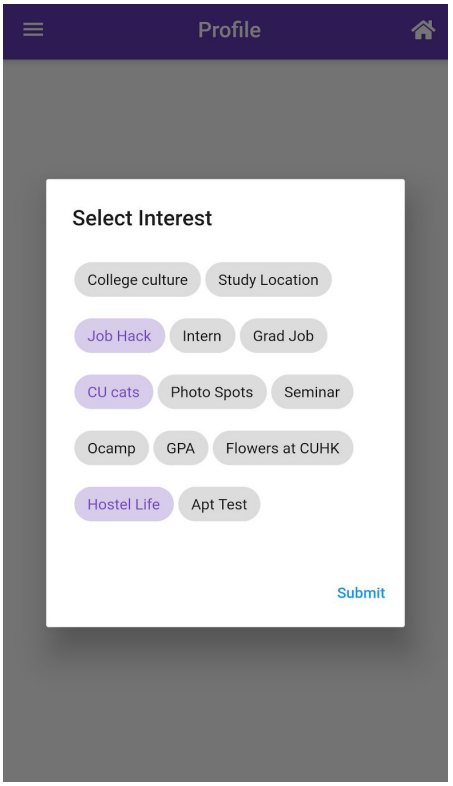


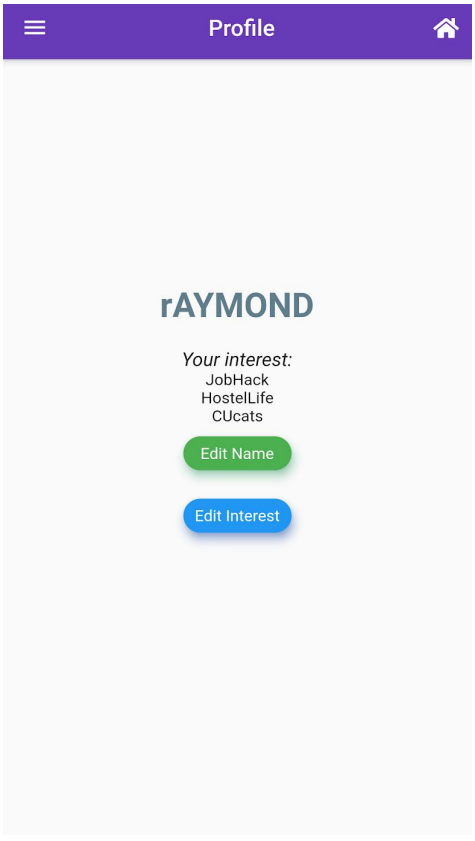
Step 3

Tom browses the side menu, he clicks on the profile page and selects the topics he is interested in, the chosen ones are in purple colour. After confirming his interests, Tom's profile is set up.



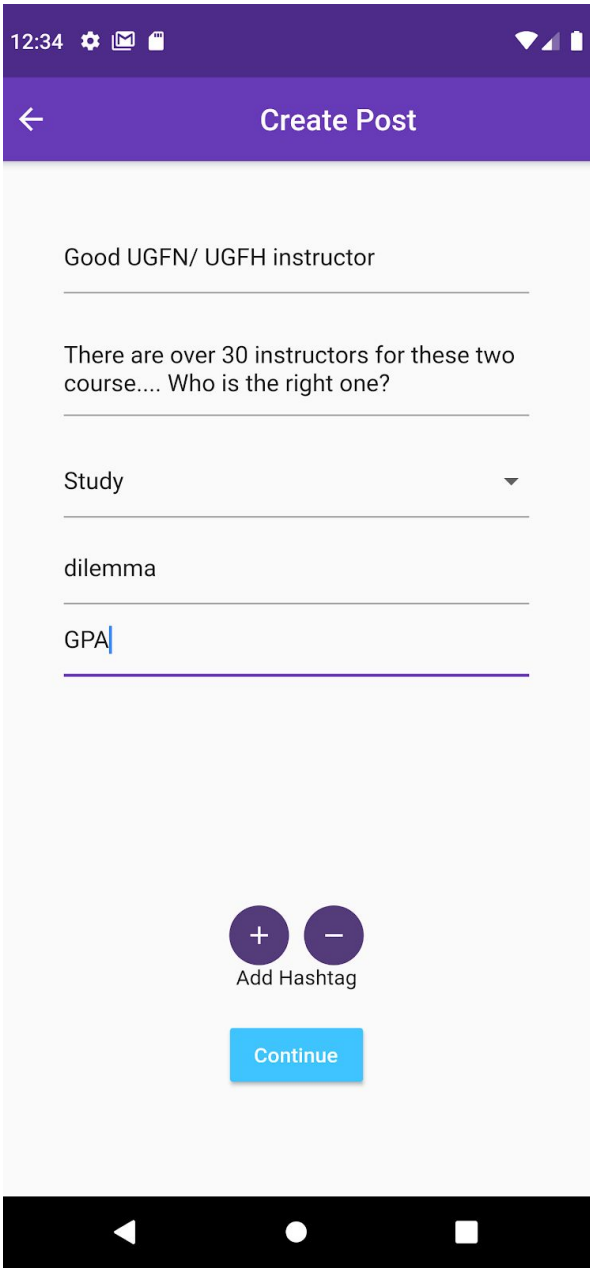






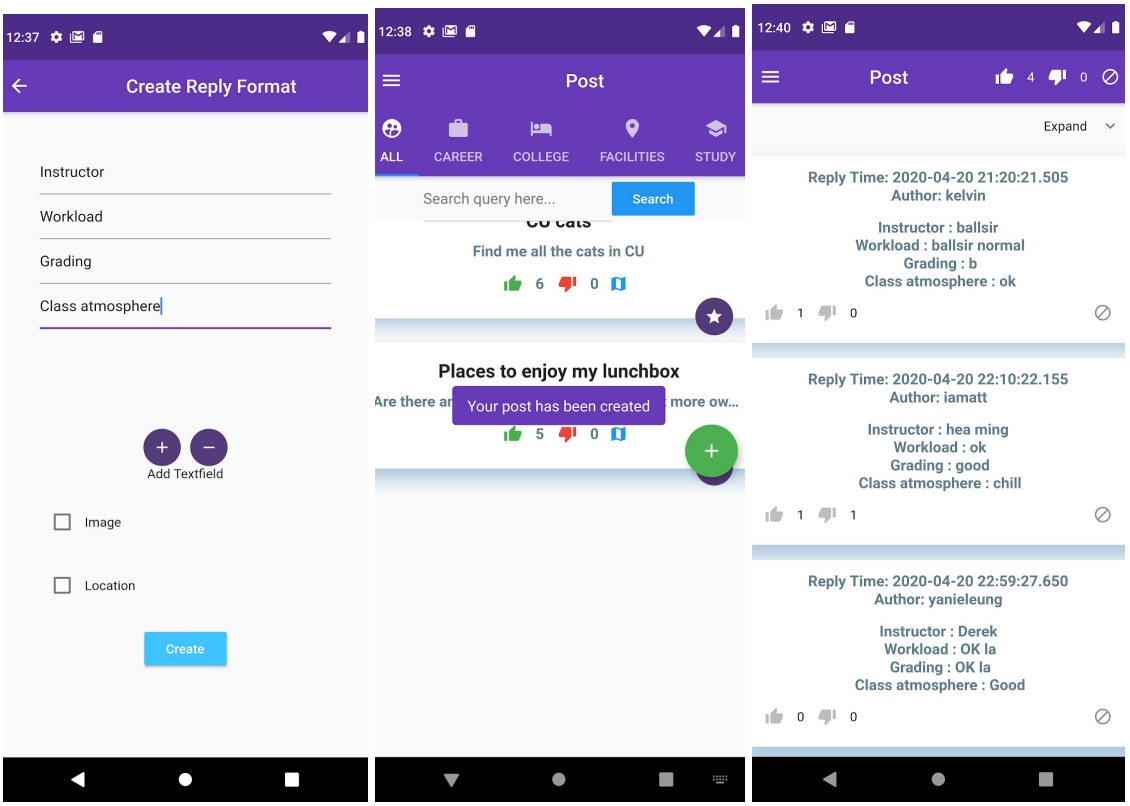
Step 4

Tom goes to the home page then press the "+" button at the bottom right to create a post. He wants to ask about "Good UGFN/UGFH instructor". After inputting the description and the hashtags, Tom can go to the next step.



Step 5

Tom has to choose the format of the reply to this post. He leaves the boxes "Image" and "Location" blank. He sets the standardized format of the reply to include three text fields. They are namely "Instructor", "Workload", "Grading" and "Class atmosphere". Next, he will be redirected to the main page and at the same time an information notification pops up to remind him that his post has already been posted. He can see his topic now. After some time, Tom has received a reply, he can click into his post to view.

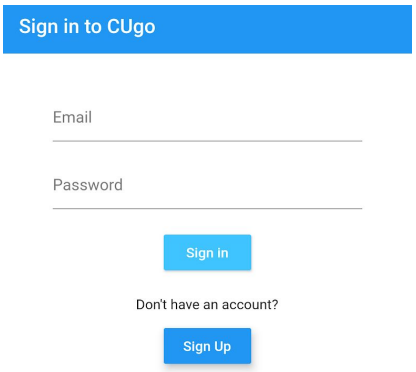


6.2.2 Scenario B

In this scenario, Kate is an existing user of CuGo. After logging in, she takes a look at the posts that she has bookmarked. Then, she browses a topic called "CUHK Water Fountains". She replies that there is a new location of the water fountain in CUHK. Lastly, she reports an inappropriate post.

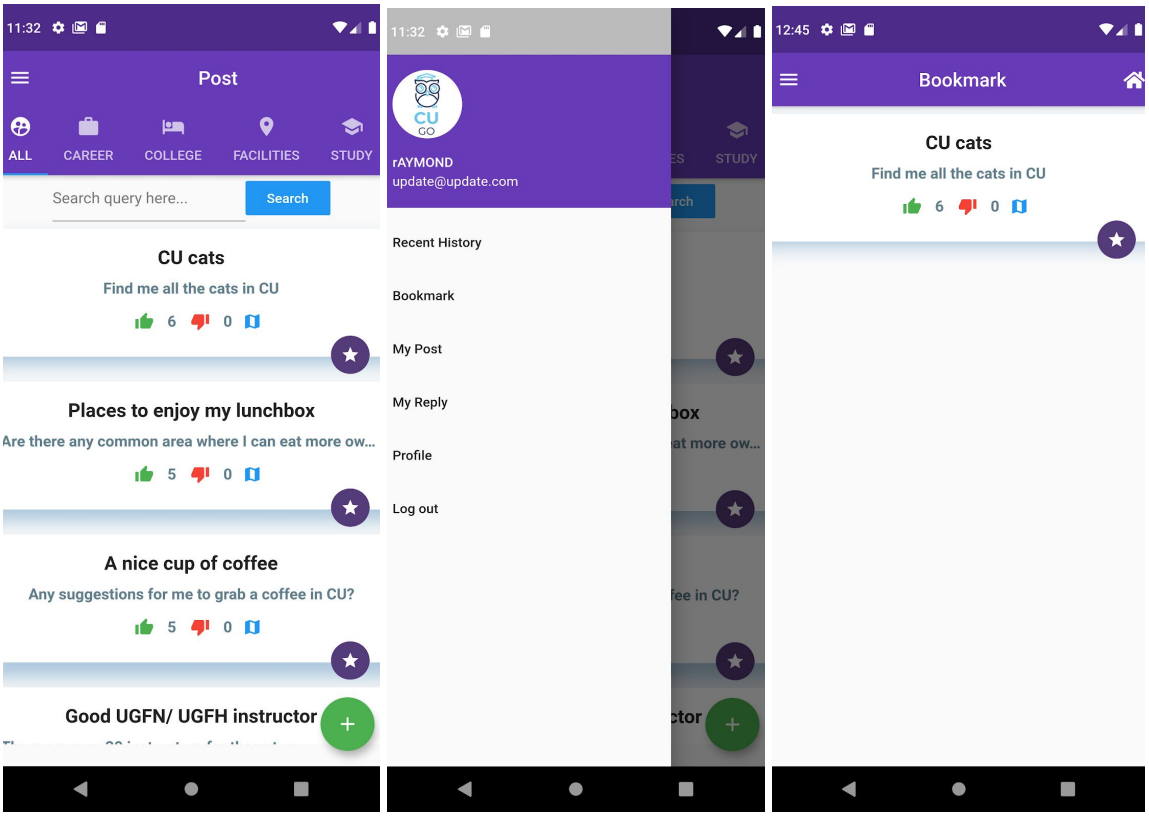
Step 1

On the Sign In page of CUgo, Kate inputs her username and password.



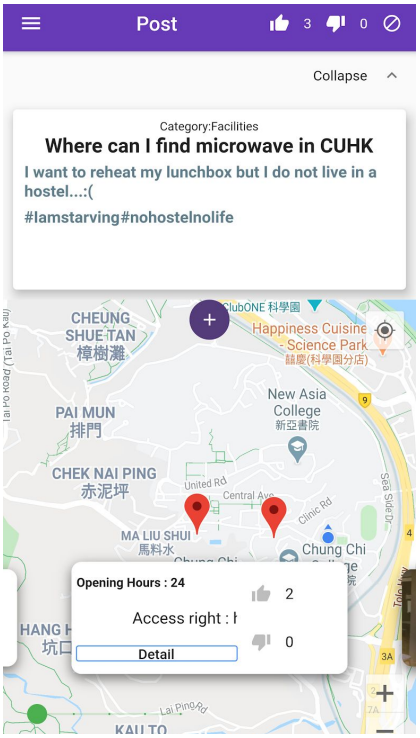
Step 2

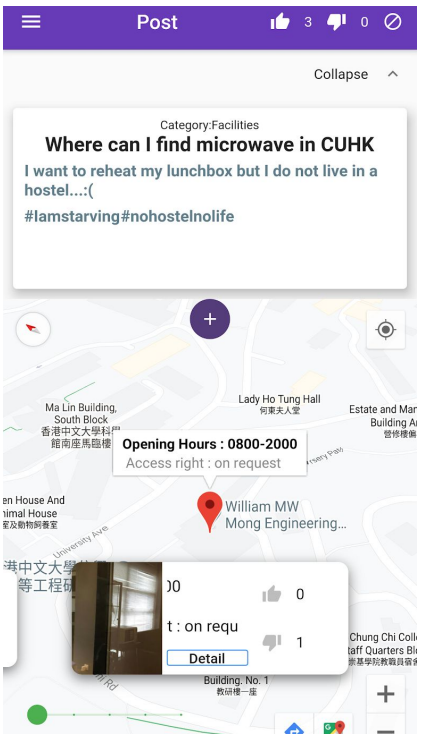
After signing in successfully, Kate sees the list of topics and different categories on the homepage of CUgo. She clicks on the leftmost button on the top bar, a sidebar menu is shown. Kate goes to the Bookmark and views what posts she has bookmarked.

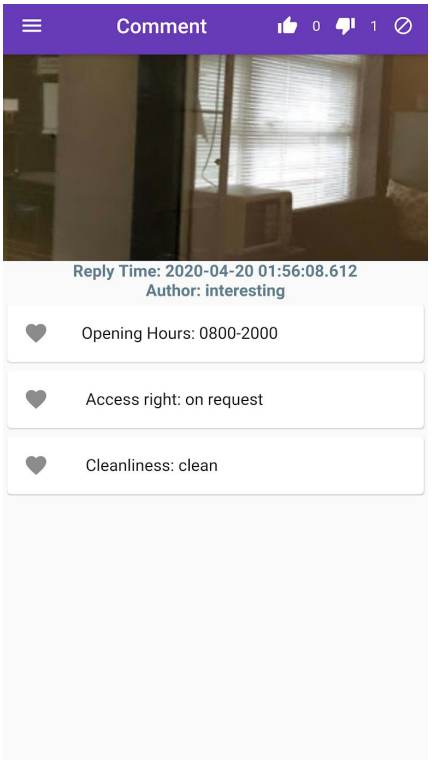


Step 3

Kate goes back to the homepage and clicks into a post called "Where can I find a microwave in CUHK", it is a located-based topic so the replies are shown on the google map as different markers. She clicks on one of the markers and the information window pops up. The information is shown on the card list as well. She then clicks on "details" to view further details.

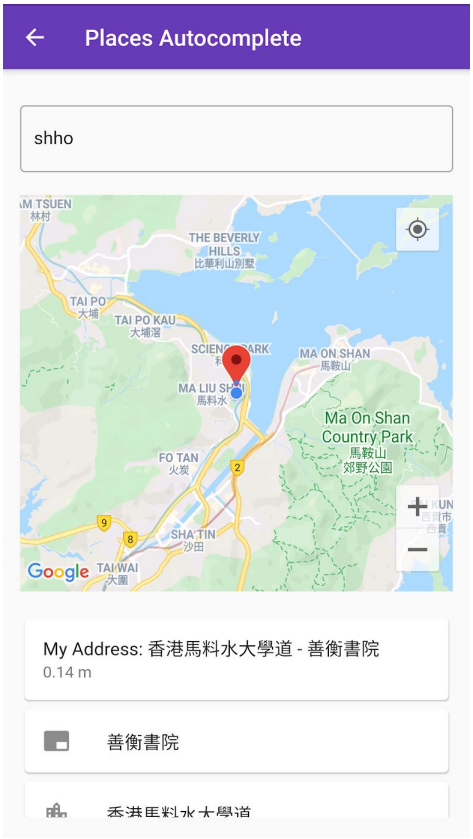


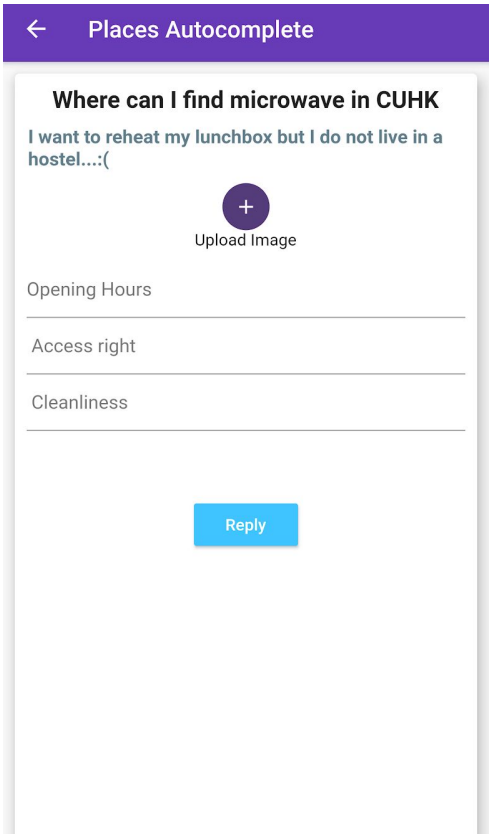




Step 4

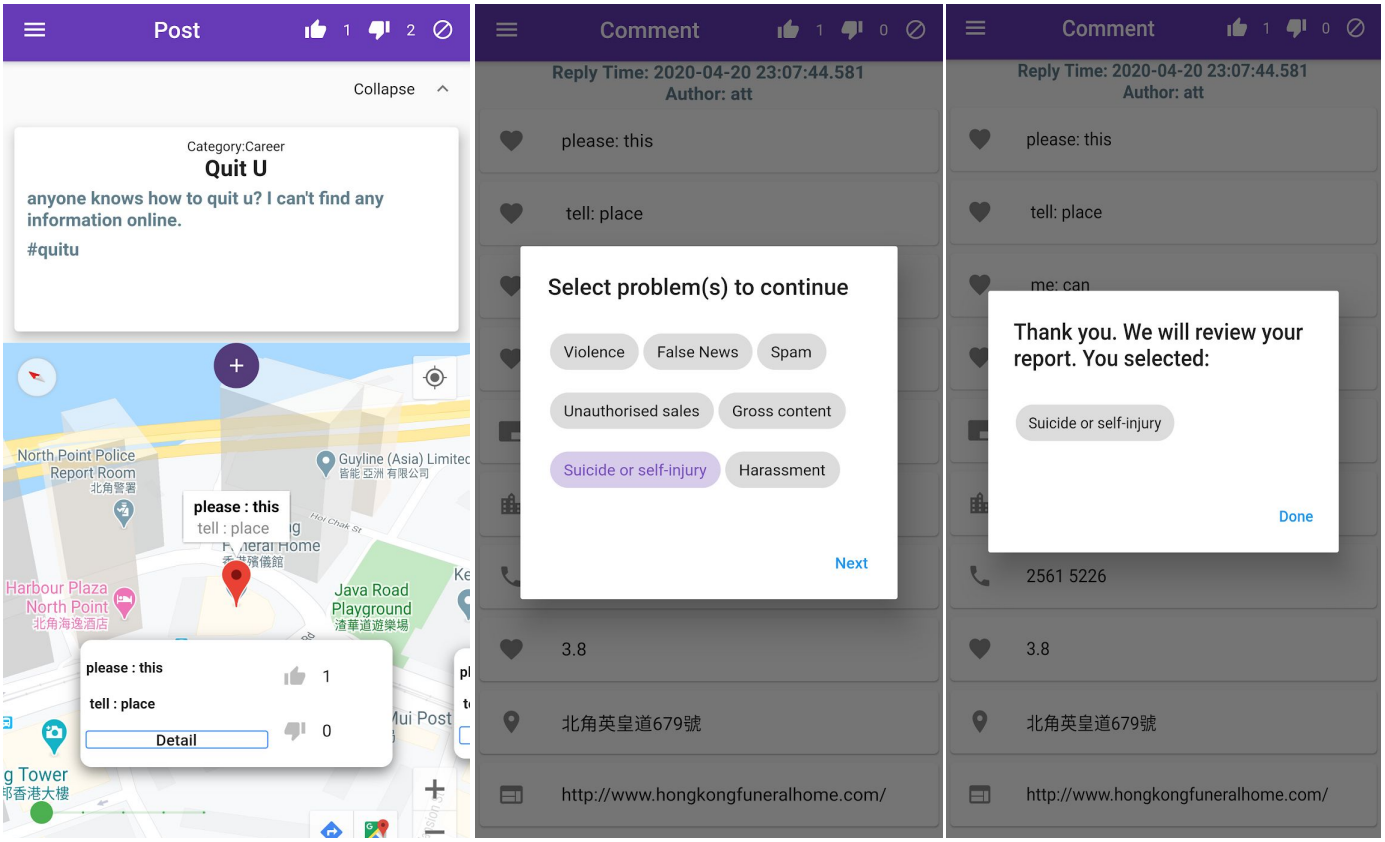
Kate wants to add an unreported microwave location, she presses the "+" add button, then she searches for the location and confirms it. Then, she fills in the details. Finally, she can submit her reply.





Step 5

Kate continues the experience in CUgo. She finds out there is a weird reply which is not meaningful and helping for the post. It is actually persuading others to harm themselves. Kate reports this reply as inappropriate.



7. Testing

7.1 Functional Testing

7.1.1 Specification

Functional testing is a type of black-box testing for testing every function without knowing the internal structure. Functional testing is to examine whether functions are generating the desired outcome given certain input test cases. By comparing the generated result with the expected result, we can conclude whether the function is working properly and perform as per the planned requirements [15].

The application is divided into six main sections, with 3 components in Authentication, 11 components in Homepage, 10 components in Text-based post, 16 components in Map-based post, 5 components in Create post, and 8 components in Side menu functions. There are 53 test components in total, graded with Accept or Slow Response or Error according to the generated result. Accept means the result is as expected. Slow Response implies the result is as expected, however, there is a significant delay (>2 seconds) in delivering the result. While Error means the result is not as expected or the component is not working. The results are generated based on normal usage test cases.

7.1.2 Result

Authentication	
Registration	Slow Response
Login	Slow Response
Switch Login and Register page	Accept
Homepage	
Browse Homepage	Accept
Browse 'Career' category	Accept
Browse 'College' category	Accept
Browse 'Facilities' category	Accept
Browse 'Study' category	Accept
Browse 'For You' category	Accept
Browse 'New' category	Accept
Search query	Accept
Side Menu	Accept
Add Bookmark	Accept
Real time update	Accept
Post (text based)	
UI display	Accept
Collapse/Expand reply form	Accept
Like/Dislike a post	Accept
Report a post	Accept
Like/Dislike a reply	Accept
Report a reply	Accept
Add reply - upload photo	Accept
Add reply - textfield	Accept
Reply submittion	Slow Response
Real time update	Accept

Post (map based)	
Google Map display	Slow Response
Marker display on map	Accept
Reply thumbnail display	Accept
Reply Detail page display	Accept
Collapse/Expand reply form	Accept
Map view changes when scrolling reply thumbnails	Slow Response
Distance scale bar	Slow Response
Like/Dislike a post	Accept
Report a post	Accept
Like/Dislike a reply	Accept
Report a reply	Accept
Add reply - place autocomplete	Slow Response
Add reply - upload photo	Accept
Add reply - textfield	Accept
Reply submittion	Slow Response
Real time update	Accept
Create Post	
Title, Content and select category	Accept
Add Hashtag	Accept
Add Textfield	Accept
Select Reply Format	Accept
Post creation	Accept
Side menu functions	
Recent History	Accept
Bookmark	Accept
My Post	Accept
My Reply	Accept
Log out	Accept
Profile	Accept
Profile - Edit name	Accept
Profile - Edit interest	Accept

The result shows that 8 out of 53 (15%) of the components have a slow response, while 45 out of 53 (85%) of the components are accepted. We can see for components in Homepage, Create Post, and Side menu function, the acceptance rate is 100%. While for Authentication, 2 out of 3 (67%) of its components have a slow response. For Text-based post and Map-based post, 1 out of 10 (10%) and 5 out of 16 (31%) of the components have a slow response respectively.

7.1.3 Discussion

Components that result in slow response are mainly actions related to uploading action, authentication, and Google Map services. For uploading action, especially image upload, it takes time for uploading the image to Firebase Storage and causing a significant delay for reply submission. Authentication requires communication with Firebase to pass the credentials to the Firebase Authentication SDK. Firebase backend services will then verify those credentials and return a response to the client [16]. These processes cause delays for registration and login procedure. Lastly, accessing Google Map API requires access to Google Maps servers, data downloading, map display, and response to map gestures [17]. While using other map services such as Places API for place autocomplete needs to issue requests and wait for the response from google web service, which causes significant delay in real usage [18].

7.2 Usability testing

7.2.1 Specification

We carried out a survey of user experience to collect feedback from users. The questions range from multiple choices to open-end questions in order to gather the review from users as detailed as possible.

When users are using CUgo, they may encounter difficulties. Therefore, we aim to examine whether they have a problem or confusion by asking them to complete the survey. Apart from the difficulties they may face, we encourage users to leave recommendations and suggestions for improvement as well.

We consider the survey as the major channel that we receive the feedback from. As usability testing focuses on how easy CUgo is to use. We put more emphasis on how easy a user finds when they attempt to accomplish each of the functions.

13 users are invited to try out CUgo at home. Due to the suspension of the face-to-face lesson, all of the users access the functions of CUgo at home, but not in CUHK. They. After that, each of them completed the user experience survey to review their experience. Functions such as taking a photo of a location in CUHK using phone camera cannot be accomplished. It is qualitative testing since we do not collect feedback from a large sample.

7.2.2 Result

Majority of the survey questions are extracted for the analysis.

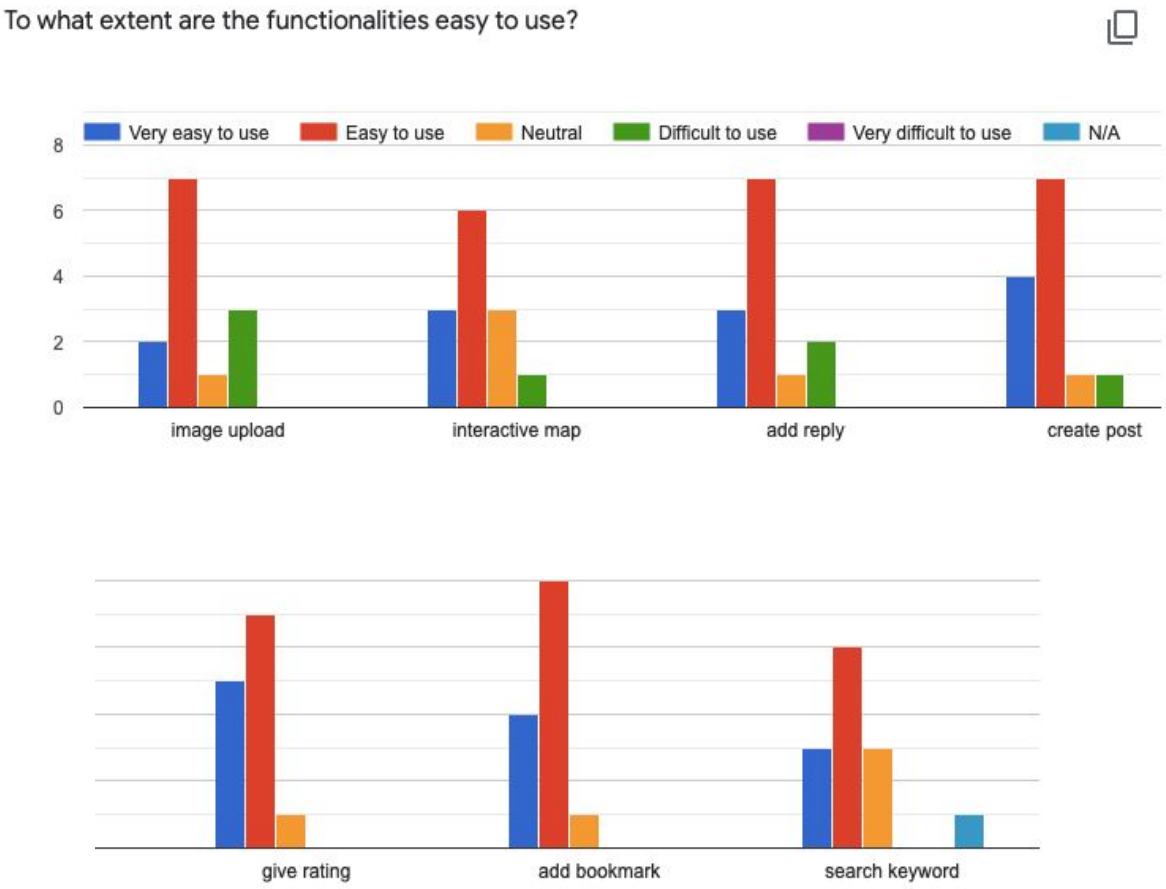


Fig 7.2.2.1 Result of "To what extent are the functionalities easy to use"

The question examines the ease of using each function. Major functions are listed out. "Easy to use" implies that users can perform the action very smoothly without confusion. While "Difficult to use" indicates that users may encounter difficulties or hesitation while attempting the function.

Majority of the users find that most of the functions are easy to use. While 3 out of 13 (23%) of the users find that it is difficult to use the image upload function. Other major functions

including an interactive map, add a reply and create post are also reported as difficult to use by the users. The survey result also shows that search function is not used frequently since some of the users have not explored this function during the testing.

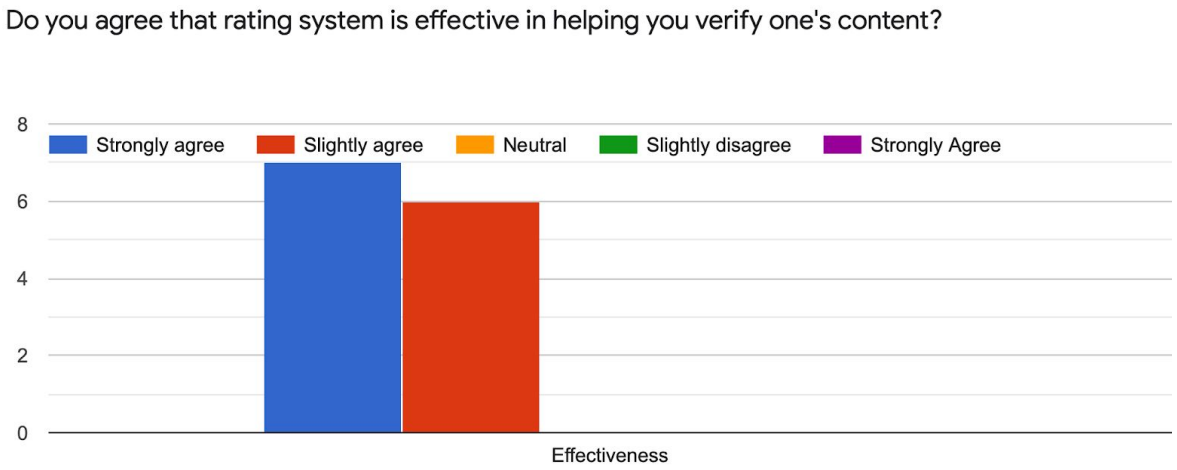


Fig 7.2.2.2 Result of the effectiveness of the rating system in verifying the content

This question is designed to find out the feeling of users towards the effectiveness of the rating system as a tool of fact-checking. 100% of the respondents considered the method is effective. Over half of the respondents strongly agree that it is effective in verifying the content.

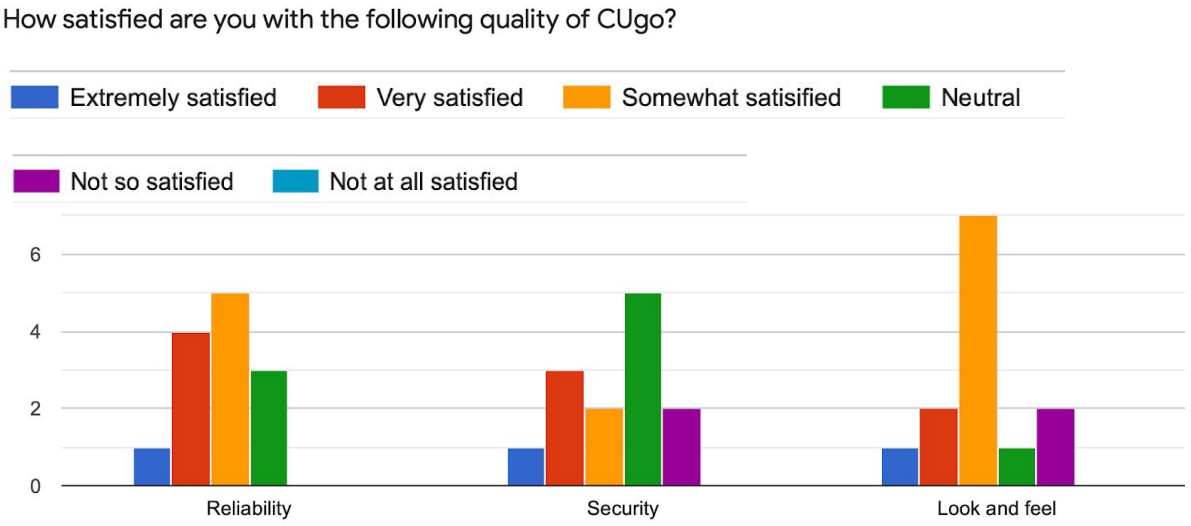


Fig 7.2.2.3 Result of the satisfaction of CUgo

This question is asked to evaluate whether users are satisfied with the following qualities of CUgo, namely reliability, security and look and feel.

In terms of reliability, 100% of respondents are satisfied with the performance CUgo. However, in terms of security and look and feel, both of them have 2 out of 13 (15%) respondents indicate that they are not satisfied with either of these two aspects.

7.2.3 Discussion

A usability test is better than traditional testing since the former involves actual users and target users rather than just developers. We received feedback from a different perspective we might have missed out.

The result of the ease of using different functions reveals the ability of the user to manipulate the function [19]. The answer is diverse throughout the spectrum from "very easy to use" to "difficult to use". Take a step back to see this result in a wider view, there is a need to clarify on how to perform the functions especially the major 4 functions which are image upload, interactive map, add a reply and create a post. A user guide could be added to guide the user along using the major functions. Apart from that, the search function may be overlooked by some of the users that they did not attempt this function. The search bar could be added in more pages rather than just under the "All" tab in the home page.

When it comes to the evaluation of the effectiveness of the credibility system, it is generally considered as effective in CUgo. It indicates that the credibility system contributes to fact-checking to a great extent.

Regarding the qualities of CUgo. This result shows that CUgo is of high reliability, there is no failure of performance reported by the user. The security problem of CUgo may be related to a simple registration process which does not require two steps authentication. However, apart from the email address, CUgo did not ask for other credentials such as the name of the users. The username in CUgo is the nickname typed by the users instead. Also, CUgo asks the user to grant the location request before CUgo retrieves the location data from the user. The privacy policy statement may need to be added in CUgo during registration to comfort the worries from users. The quality of “Look and feel” is more related to the user interface and layout of CUgo. The interface may look too simple without much aesthetic design to appeal to the users. The interface should be improvised to make it more appealing.

After analyzing the result, some of the potential issues of CUgo are revealed. The results reminded us there is still a lot of room for improvement for CUgo. All in all, as an evaluation, this usability testing is successful upon providing many insights for improving CUgo.

7.3 UI testing

7.3.1 Specification

CUgo relies heavily on graphical user interface interaction. It is crucial to ensure the GUI displays and works properly on different Android devices to broaden our target audience. Manual testing on devices of different resolution and OS versions will be performed on three interfaces including homepage, text-based post and location-based post. Three different Android devices are chosen to perform the UI test, their specifications are as follows.

Type	Name	Resolution ▲	API	Target
	4.65 720p (Galaxy Nexus) API 24	720 × 1280: xhdpi	24	Android 7.0 (Google APIs)
	Pixel 3a API 28	1080 × 2220: 440dpi	28	Android 9.0 (Google Play)
	Pixel C API 26	2560 × 1800: xhdpi	26	Android 8.0 (Google APIs)

Fig. 7.3.1.1 Android devices specifications

7.3.2 Result

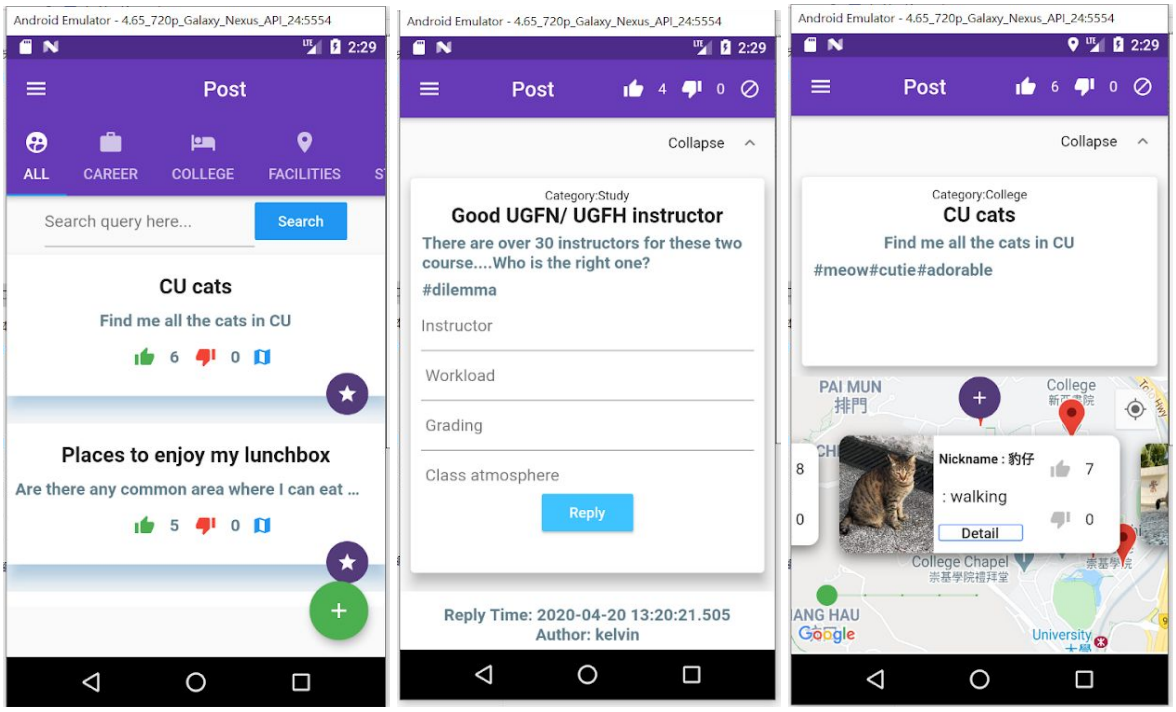


Fig. 7.3.2.1 Galaxy Nexus screen captures

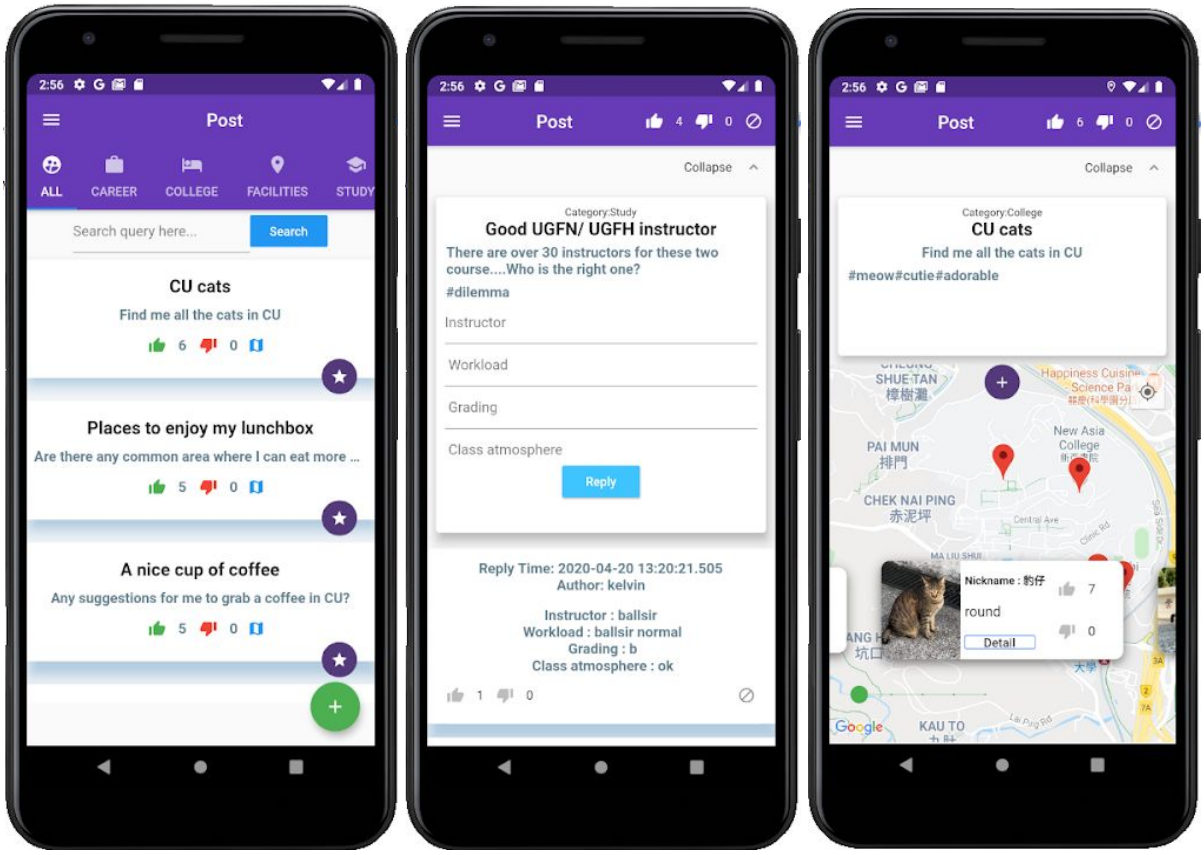


Fig. 7.3.2.2 Pixel 3a screen captures

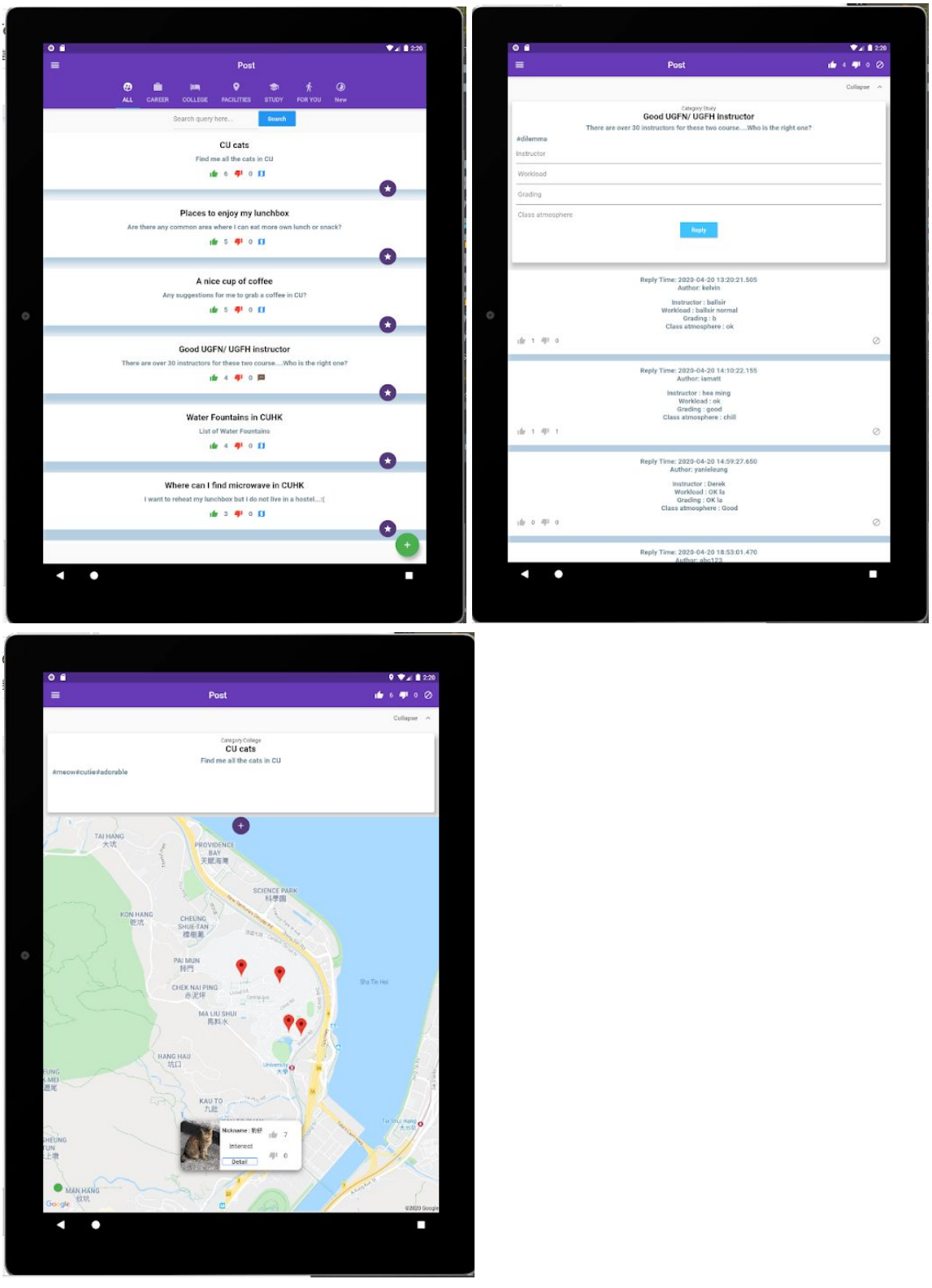


Fig. 7.3.2.3 Pixel C screen captures

7.3.3 Discussion

Flexibility and scalability of GUI are one of the design concerns when developing CUgo. The performance is as expected and no error occurred in different selected devices of screen resolution ranging from 720*1280 to 2560*1800, and Android version from 7.0 to 9.0.

It is worth mentioning that when a user first enters a text-based or location-based post, the reply form or post content will be shown and occupied a large portion of the screen in device with relatively low resolution such as Galaxy Nexus with 720*1280 in our test case. A button is added on the top-right corner to expand or collapse the reply form or post content section when they are not needed. This can increase the readability of post replies and enhance usability.

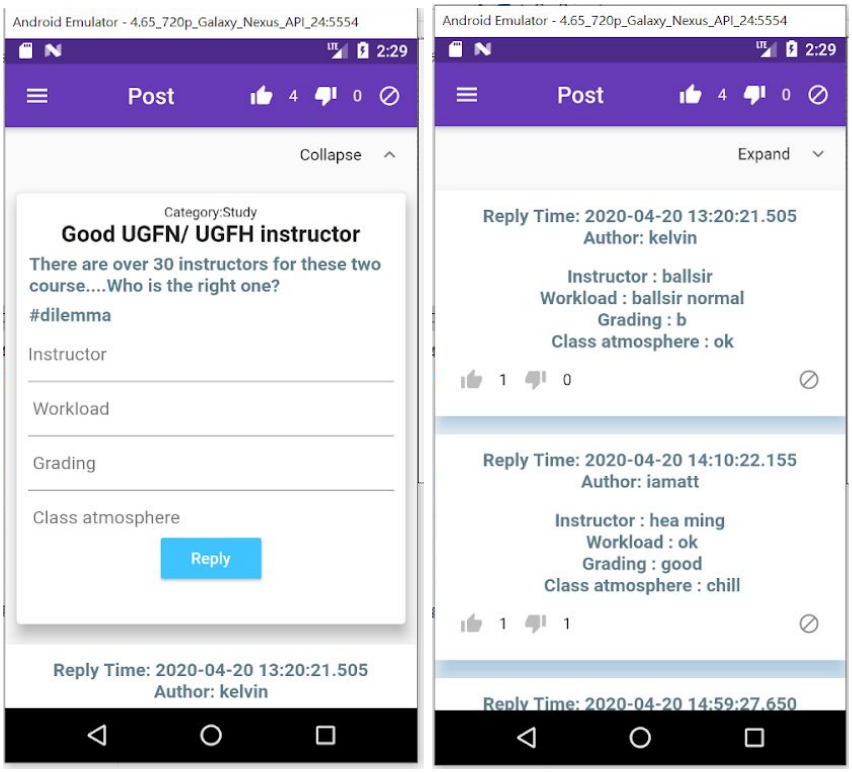


Fig. 7.3.3.1 Galaxy Nexus text-based post screen captures

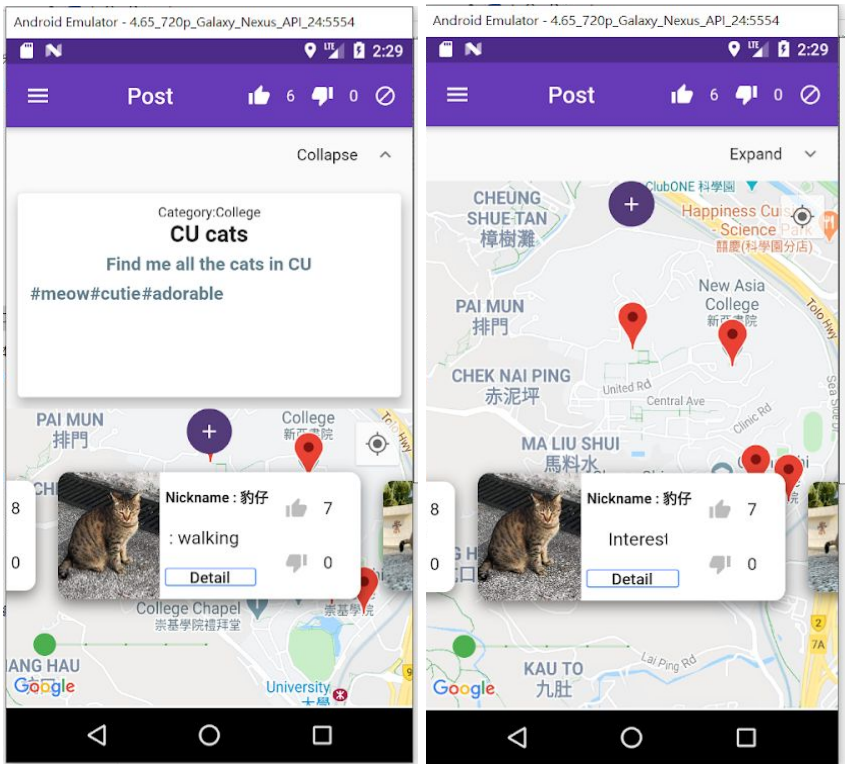


Fig. 7.3.3.2 Galaxy Nexus location-based post screen captures

7.4 Fact-checking

7.4.1 Specification

CUgo has used a collaborative approach to collect data and determine the credibility of the collected data by crowd effort. We believe that crowd intelligence can be used to distinguish facts and frauds and eventually build up a crowd fact-check database.

13 participants are invited to reply on different posts and vote for replies and posts according to their credibility. The results will then be analysed to evaluate the effectiveness of this collaborative approach in terms of fact-checking. As the amount of participants is small, we decided to analyse the top three upvoted posts as there are more replies and votes within these posts. The focus of the analysis will be the correlation between votes and degree of truth.

7.4.2 Result

“CU cats”, “Places to enjoy my lunchbox” and “A nice cup of coffee” are the top three upvoted posts with 6, 5, and 5 upvotes. There are 6, 8 and 4 replies inside the post respectively.

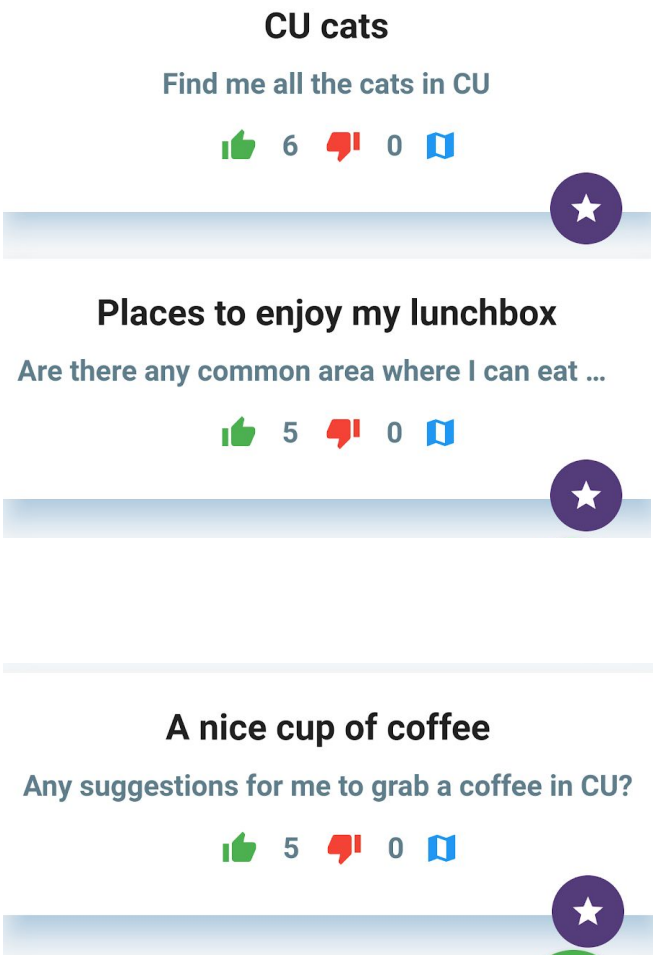


Fig. 7.4.2.1 The top three upvoted posts

For the post “CU cats”, the replies and result of votes are as follows.

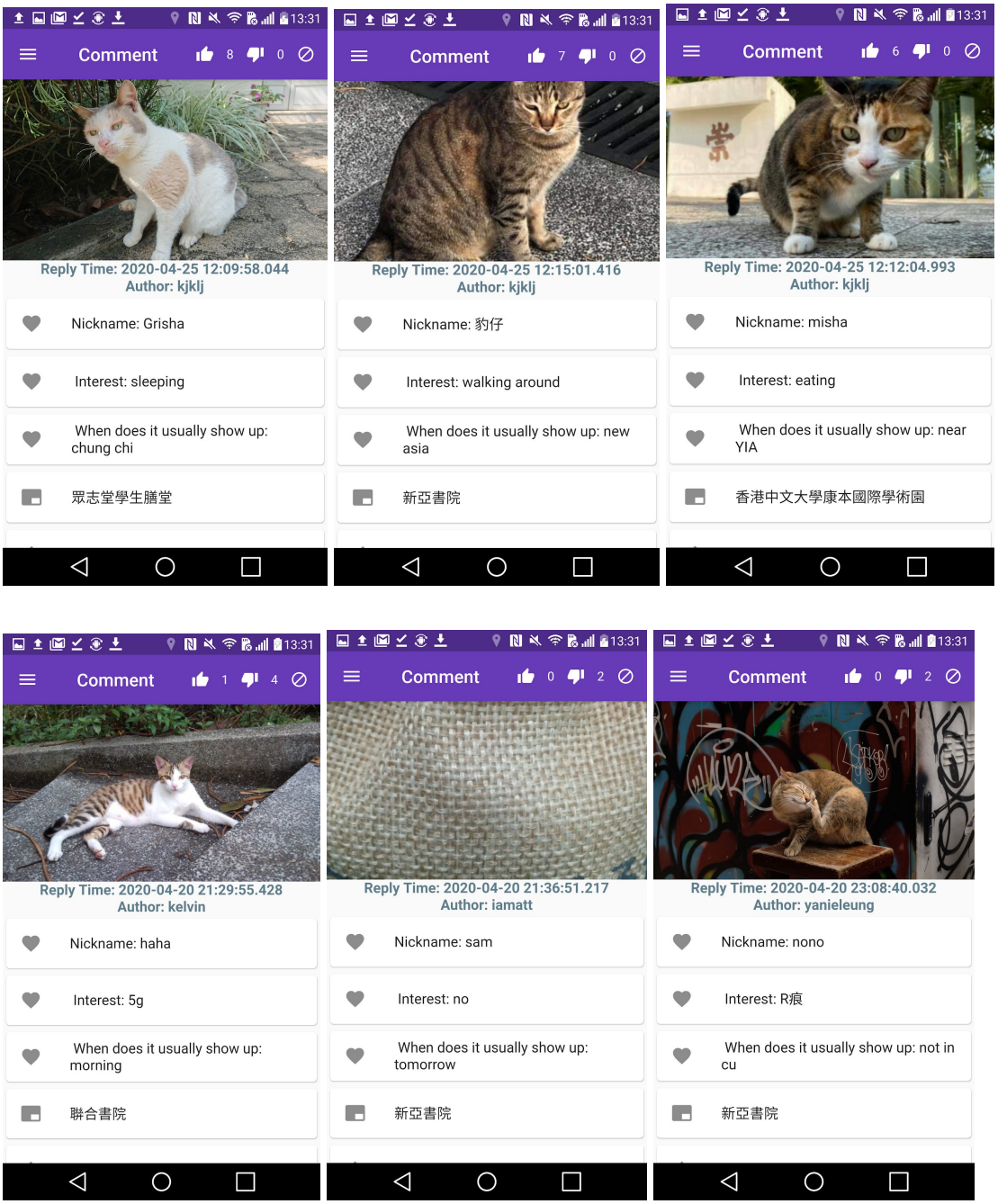
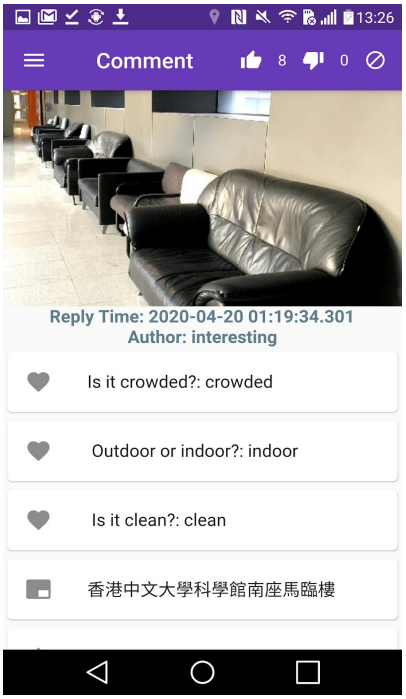
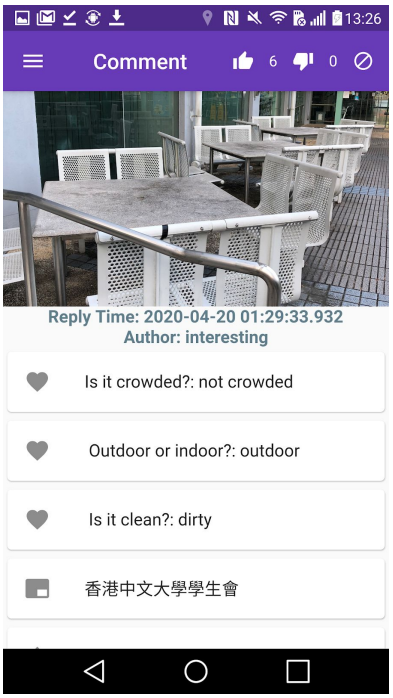


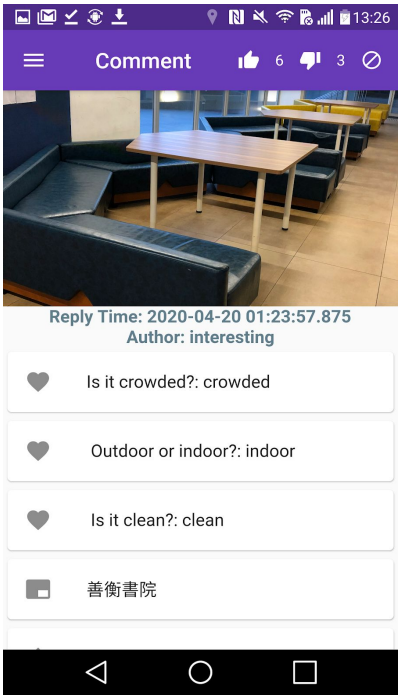
Fig. 7.4.2.2 Replies in the post “CU cats”

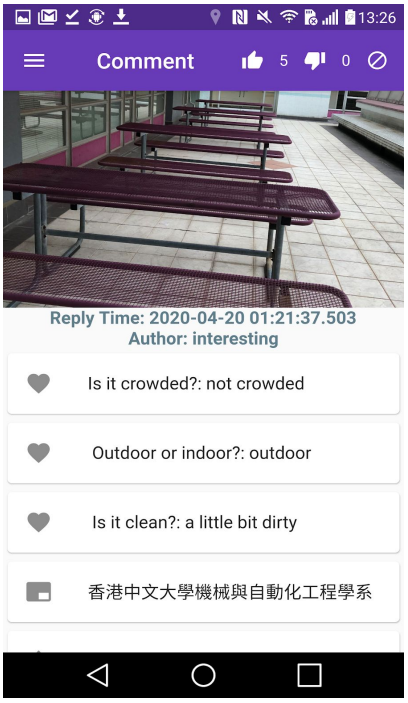
The first three cats receive 100% upvote while the remaining receive 80-100% downvote. The first three replies are the cats that actually live in CU, while the remaining replies either provide irrelevant cat information or false context. It is confident that the voting actually reflects the credibility of information and does fact-check on these replies in such cases.

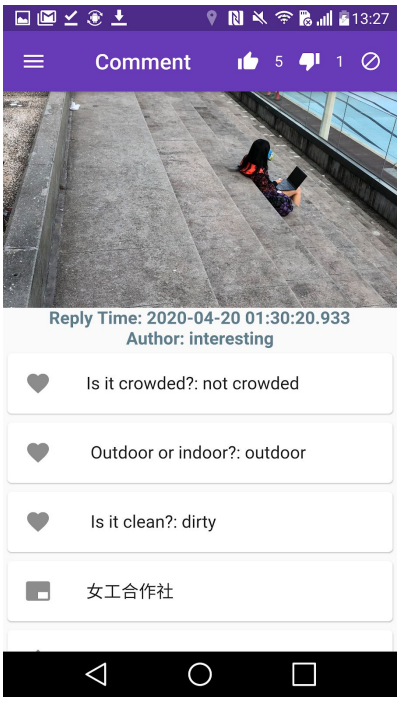
For the post “Places to enjoy my lunchbox”, the replies and result of votes are as follows.

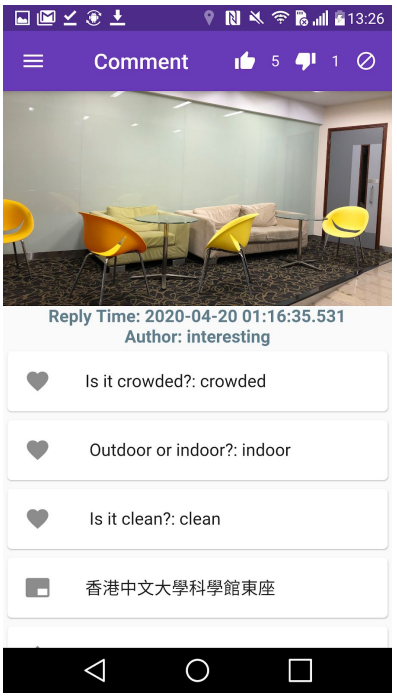












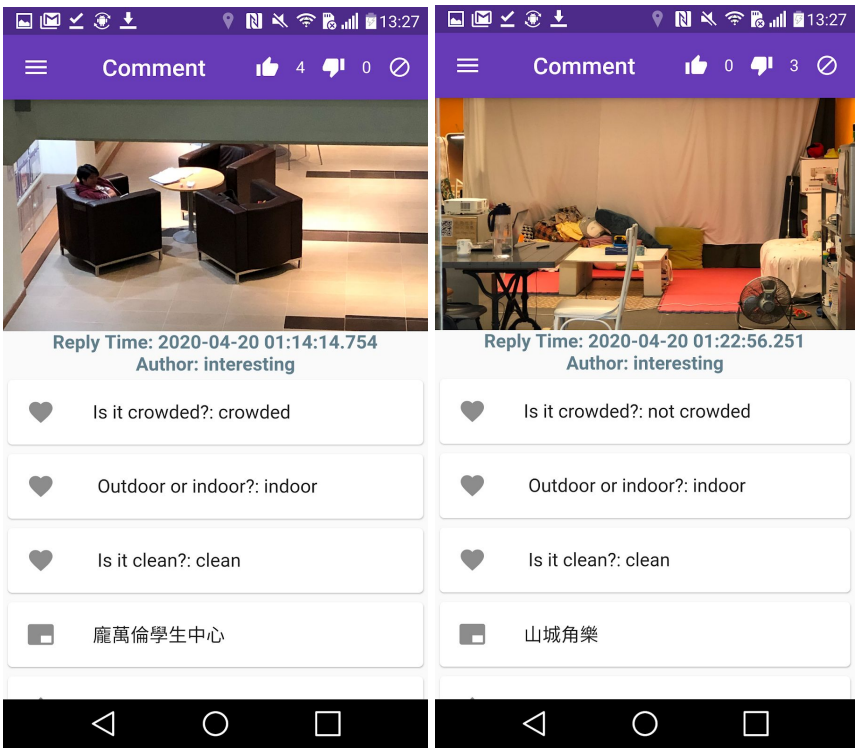


Fig. 7.4.2.3 Replies in the post “Places to enjoy my lunchbox”

The first 7 replies receive 67-100% upvote while the last one receives 100% downvote. The first 7 replies are the places that staff and students can enjoy their lunchbox, while we discover that the last place does not allow eating in there after checking manually. We can tell from the voting that it correlates with the credibility of information and perform fact-check this post.

For the post “A nice cup of coffee”, the replies and result of votes are as follows.

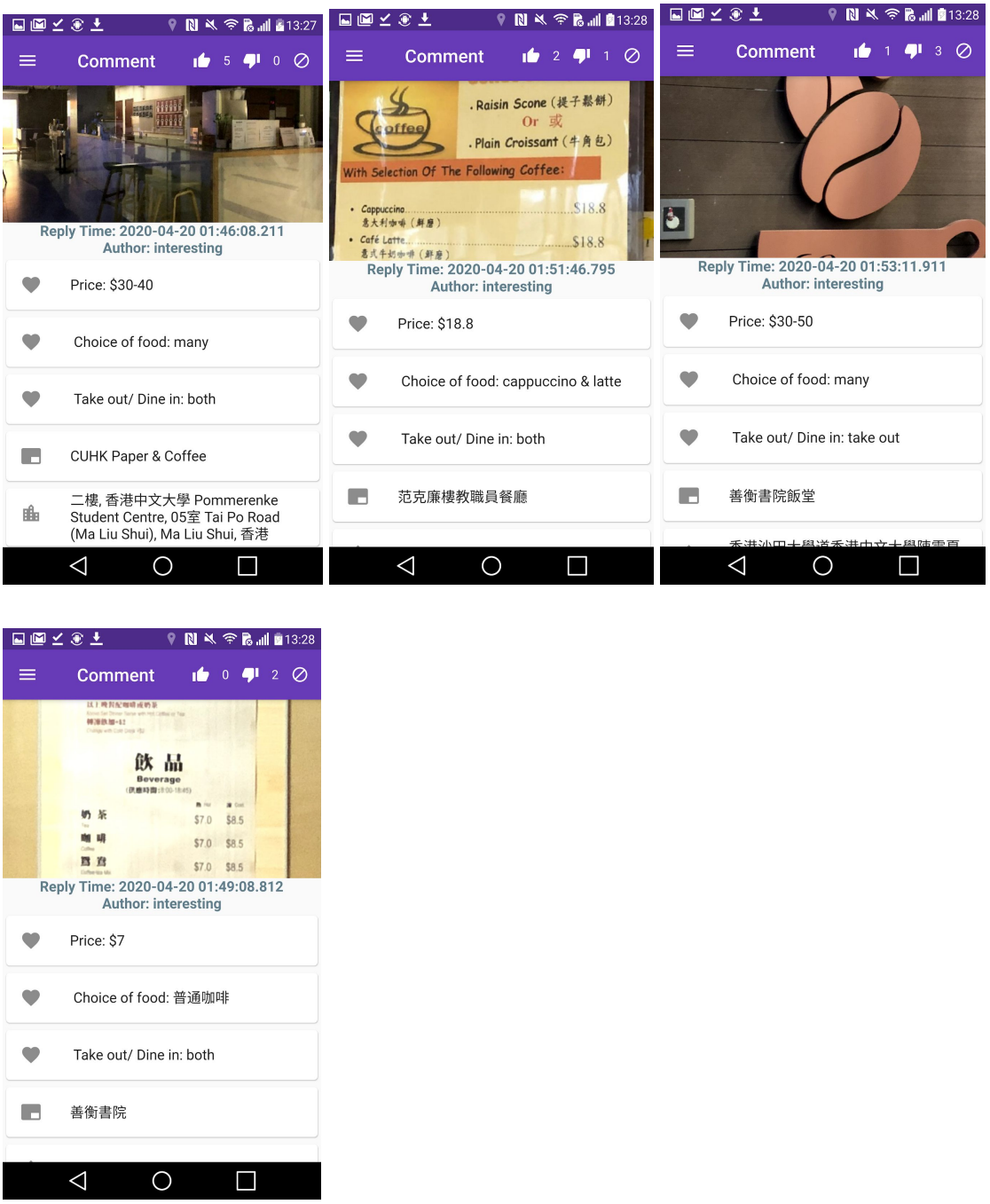


Fig. 7.4.2.4 Replies in the post “A nice cup of coffee”

The percentage of upvote is 100%, 50%, 25%, 0% for the four replies respectively. As the topic of this post is subjective, users may have different opinions on "a nice cup of coffee”. Users can use the votes as a reference when they are buying a coffee. However, this is more like a rating and recommendation other than performing fact-check.

7.4.3 Discussion

The effectiveness of collaborative fact-check depends on two main criteria, number of responses and the nature of the post. The testing result shows that posts that are not popular and have few replies and votes usually are more biased and cannot reflect fact effectively. This is also the reason for us to analyse the top three most upvoted posts because they are more likely to have more replies and votes. Only with sufficient amount of user contribution, either replying or voting, the power of collaborative fact-check can be reflected.

Another important criteria is the nature of the post topic. If the post topic is subjective, replies are more on personal opinions than fact-checking, the fact-checking result can be biased and meaningless. It is crucial for a post topic to be objective in order to enable fact-checking.

8. Limitation and Difficulties

8.1 Data Collection

As the feature of CUgo is the fact that it supports creating a post and adding a reply with photos. Also, the current version only targets CUHK. However, under the social distancing measures imposed by CUHK, students are not encouraged to go back campus. Therefore, the majority of photos are taken earlier before the virus outbreak.

With the lack of photos, we tried our best to balance the proportion of posts with and without an image in CUgo to weigh equivalently.

8.2 Alpha Testing

CUgo has undergone the alpha testing stage. As almost all of the users test CUgo at their homes, many functions may not be fully performed comparing to the situation where they can move around freely in CUHK.

Furthermore, the original aim of CUgo is to allow users to create a post at any time when they encounter any enquiry regarding the matters at CUHK. Therefore, when almost all the users are staying at home to receive online teaching, they are accustomed to the learning lifestyle at home and most of them are detached from the CUHK atmosphere. It is difficult for them to create posts about the matters they concern at CUHK.

For the location-based posts, CUgo asks for user location at first and then several replies which are marked at the user's current location nearby will be shown. Users can also change the scope from the radius of 200m to 1km by dragging the scroll bar at the bottom of the map. However, this function is poorly tested due to the fact that most of the users are not at CUHK while every reply is marked at CUHK. Users can enjoy the suggestion of nearby points of interest during the testing phase.

Despite the limitation, we attempted to create some posts for the users first before the testing. Therefore, during the testing phase, there are some posts which have already been created for users to reply freely. Users are still able to create their own posts.

9. Future Development

9.1 User motivation

Maintaining a collaborative platform of rapid contribution from users is not an easy job, user motivation is one of the important factors that we have to take care of.

In order to encourage users to post their queries and submit their facts to CUgo continuously, we aim at implementing a reward scheme for users to enhance their motivation to contribute to this collaborative platform. Just like the membership scheme of different forums and games, users are divided into different levels. The level of each user takes into account the number of posts and replies he made and also the credibility of his posts and replies on average.

Every effort counts, we value each contribution. The higher the credibility earned by the user and the higher the level he can achieve.

9.2 Public forensic Database

CUgo has built a small database for the replies and posts made by users. In the long term, it is possible for CUgo to develop into building a public forensic database which can help with fact-checking and giving out the truth if there are enough active users and the knowledge exchange among users are meaningful and useful.

For example, for the post which collects the photos of CU cats and their last seen location. If the posts become popular within the CUHK community, the contribution from the users will increase significantly and it probably can set up a detailed database to record the activities and location of each of the cats in CUHK according to the reply added by the user. The database will be useful when it comes to missing the cat or there is an accident happening on them. The database may successfully be able to trace where the last seen location of that cat is and who is feeding the cat at that time.

10. Division of labour

10.1 Authentication

The authentication system of CUgo is implemented through Firebase. Firebase provides us with several options of login methods such as email and password, phone number, google, facebook, anonymous etc.

I have chosen email and password as our authentication method. Email of each individual user will be stored on the server. However, as we hope to provide an anonymous fact-check platform for users to contribute freely, personal identity such as email will never be disclosed to other users.

10.2 Text-based post

Text-based post consists of three main parts, post title and content, reply form and view replies.

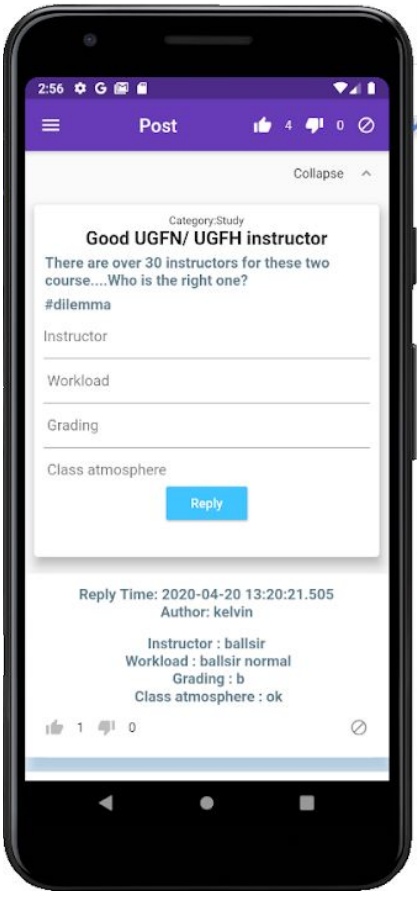


Fig. 10.2.1 Text-based post

Users can view, add a reply, upvote or downvote a post, upvote or downvote a reply, report a post or report a reply. All contents are being stored and retrieved from Firebase. Replies and number of upvotes and downvotes will be updated in Firebase and reflected on screen immediately with response to user actions as streambuilder is used, the user does not need to reload the page to see new changes.

10.3 Location-based post

Location-based post consists of three main parts, post title and content, Google Map and reply cards.

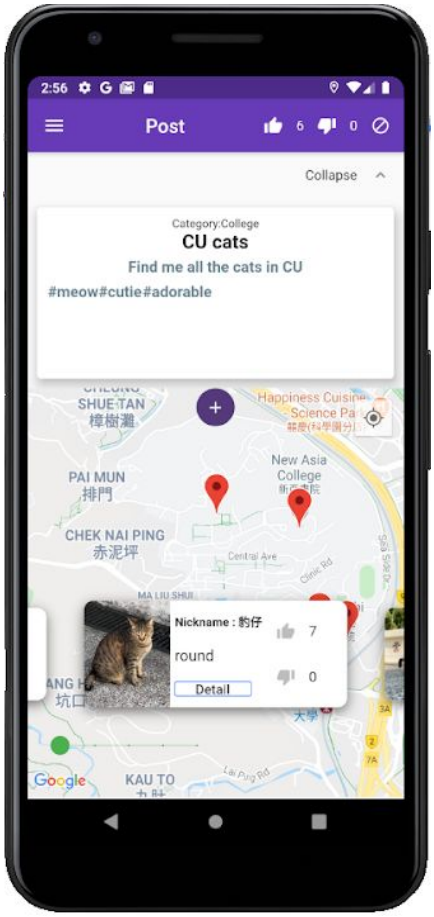


Fig. 10.3.1 Location-based post

Users can view the map, reply cards and reply details, add a reply, scroll the reply card list, move camera to current location, upvote or downvote a post, upvote or downvote a reply, report a post or report a reply. Same as text-based post, any updates or changes will be reflected immediately by streambuilder. The map view is implemented using Google Map API, markers will be pinned according to the replied locations. Users can scroll the reply card list to view other replies and the map camera will move to the location of that reply

automatically. The reply card list provides preview information such as thumbnail, number of upvotes and downvotes and text information of each reply. Users can click “detail” on the corresponding card to view the reply detail.

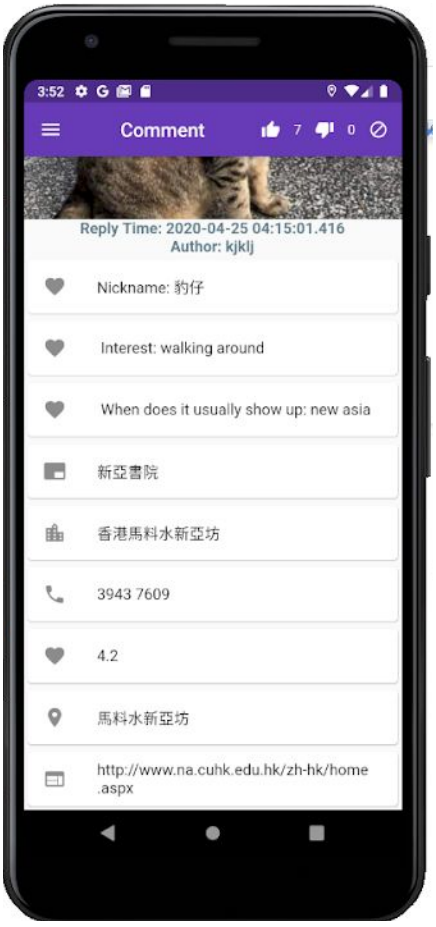


Fig. 10.3.2 Location-based post reply detail

Information in reply details such as place name, address, telephone number, rating etc. are provided by Google and are not stored in our backend. Only the place ID, longitude and latitude of that location is stored in the database, other information of this location are being retrieved real time by the use of Google Place API.

Fig. 10.3.3 Location-based post reply database schema

10.4 Database management

Database management is the backbone of CUgo. Most functions such as creating a post, adding a reply, upvoting or downvoting, bookmarking a post etc. are supported by create, read, update and query action.

```
Future replyPost_map(List<String> reply, String postid, double lat, double lng, String placeID, String username) async {
  await replyDatabase.document(postid).setData({});
  return await replyDatabase.document(postid).collection('reply').add({
    'authorid': this.uid,
    'author': username,
    'reply': reply,
    'createtime': DateTime.now(),
    'like': 0,
    'dislike': 0,
    'latitude': lat,
    'longitude': lng,
    'placeID': placeID
  });
}
```

Fig. 10.4.1 A database management function to upload location-based reply

It is worth noting that for storing images in the database, the actual image is stored in Firebase Storage. After uploading the image onto Firebase Storage, we then get its URL and store the URL in the database. Images are displayed on screen by the means of network image.

10.5 Search function

I have integrated Algolia search engine into CUgo to perform the keyword search function. I have set up the connection between Firebase and Algolia for retrieving data from Firebase, indexing the data and configuring the custom ranking and rules. Finally, codes are added into the application to perform search using Algolia.

```
_search() async {
  print("ENTER");
  setState(() {
    _searching = true;
  });

  Algolia algolia = Algolia.init(
    applicationId: 'DXHJ3JTQGA',
    apiKey: '946b9e2d2f4cb2226cfa583f0d14b733',
  );

  AlgoliaQuery query = algolia.instance.index('posts');
  query = query.search(_searchText.text);

  _results = (await query.getObjects()).hits;

  setState(() {
    _searching = false;
    search=true;
  });
}
```

Fig. 10.5.1 Part of the code in search function

11. Conclusion

11.1 Mobile Application

CUgo is a product of software engineering. It is a mobile application in which the target users are people in CUHK. CUgo has several functionalities which include creating a post, adding a reply, browsing a map, editing an interest, adding a bookmark, just to mention a few. All of the functions add up together to build a platform for users to exchange their knowledge.

11.2 Fact-checking

With the help of credibility system, replies and posts with higher positive ratings will be considered as more close to a fact. CUgo itself is not a means of fact-checking, but it is more like a supplementary way of supporting fact-checking. Many of the fact-checking channels nowadays are more likely to focus on news fact-checking which is usually text-based. There is a saying that “A picture is worth a thousand words”. Pictures can help reveal a true story and it makes the seemingly real information more convincing to the users. This is the reason to make CUgo different from other fact-checking platforms. CUgo supports image and location and most importantly, fact-checking is not solely by one authority or organisation but it is by the power of the crowd.

12. Acknowledgement

Lastly, we would like to express our gratitude to Prof. Lyu Rung Tsong Michael for his guidance and suggestions. We would also like to special thanks to Mr. Edward Yau for giving us a helping hand in technical aspects. Their constructive opinions provide valuable insight on our project.

13. Reference

[1] “Snopes.com - The definitive fact-checking site and reference source for urban legends, folklore, myths, rumours, and misinformation,” Snopes Media Group Inc. [Online]. Available: https://www.snopes.com/.

[2] “Fact-checking U.S. politics | PolitiFact,” Poynter Institute. [Online]. Available: https://www.politifact.com/.

[3] “AFP Hong Kong | AFP Fact Check,” AFP. [Online]. Available: https://factcheck.afp.com/afp-hong-kong.

[4] “Gööp U-Life: header.web,” Gööp Web. [Online]. Available: https://hk.goop.ai/feed?ref=landing.

[5] “About Us,” Snopes Media Group Inc. [Online]. Available: https://www.snopes.com/about-snopes/

[6] “Question: What is Snopes' fact-checking process?,” Snopes Media Group Inc. [Online]. Available: https://www.snopes.com/faq/fact-checking-process/.

[7] “The Principles of the Truth-O-Meter: How we fact-check,” Poynter Institute. [Online]. Available: https://www.politifact.com/truth-o-meter/article/2018/feb/12/principles-truth-o-meter-politifacts-methodology-i/.

[8] “Fact checking: how we work,” AFP. [Online]. Available: https://factcheck.afp.com/fact-checking-how-we-work.

[9] “Hot reload,” Flutter by Google. [Online]. Available: https://flutter.dev/docs/development/tools/hot-reload.

[10] “Firebase Authentication | Firebase,” Google. [Online].

Available: https://firebase.google.com/docs/auth.

[11] “Cloud Firestore | Firebase,” Google. [Online].

Available: https://firebase.google.com/docs/firestore.

[12] “Google Analytics | Firebase,” Google. [Online].

Available: https://firebase.google.com/docs/analytics.

[13] “Search API for speed, relevance for modern UX,” Algolia. [Online].

Available: https://www.algolia.com/products/search/.

[14] “Enabling Offline Capabilities on Android,” Flutter by Google. [Online].

Available: https://firebase.google.com/docs/database/android/offline-capabilities.

[15] ISO/IEC/IEEE International Standard - Systems and software engineering.

ISO/IEC/IEEE 24765:2010(E). 2010. pp. vol., no., pp.154, 15 Dec. 2010.

[16] “Firebase Authentication,” Google. [Online].

Available:

https://firebase.google.com/docs/auth/?gclid=CjwKCAjw-YT1BRAFEiwAd2WRtrInPLao8s

At-vZMNa7gXJOd_NNF4uNa4TqMV7Ryalcp2CqlCFoFXRoCX88QAvD_BwE.

[17] “Overview | Maps SDK for Android | Google Developers,” Google. [Online].

Available: https://developers.google.com/maps/documentation/android-sdk/intro.

[18] “Overview | Places API | Google Developers,” Google. [Online].

Available: https://developers.google.com/places/web-service/intro.

[19] “What is usability testing?,” Experience UX. [Online].

Available: https://www.experienceux.co.uk/faqs/what-is-usability-testing/.