CECI1

FYP Final Report

UMatch : A Mobile Friendship App for University Students

By

Chan Yin Wan, Ho Hau Wo, Wong Wai Hong, and Zhang Shao

CECI1

Advised by

Dr. Chan Ki Cecia

Submitted in partial fulfillment of the requirements for CPEG 4901

in the

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

2021-2022

Date of submission: 2022-04-20

Abstract

University is a new stage for all university freshmen. In order to adapt to university life, new friendships take an essential role. The main motivation of our project comes from the challenges in finding appropriate project group mates in the university, especially the current mobile friendship apps on the market that are not designed for university students. UMatch provides a simple user-friendly interface for users to make friends without excessive effort. Instead of filling up a list of information, users solely input limited but essential data in their profile. Based on those collected data, the system uses the designed algorithm to find potential users with some commonality between them. Furthermore, UMatch also provides a platform for users to engage in different activities. By joining activities, not only do users expand their social network, but they are also motivated to engage in various activities that they have never experienced before. Additionally, there is a trading platform for users to get or exchange goods such as course materials or second-hand laptops. During the purchase process, users can meet people who are interested in similar things. This report will first evaluate similar existing apps on the market to show the main difference with this project. The design and implementation are explained in detail to provide description of the frontend, backend, and the recommendation system. Testing and evaluation are demonstrated to show how our application is being tested. Future development is discussed to include our possible future works and improvements. It is believed that UMatch can help users to find friends on campus more easily.

Table of Content

1. Introduction

1.1 Overview

Undoubtedly, university is a new stage for all university freshmen. In order to adapt to university life, new friendships take an essential role. There was research [1] indicating a strong association between the quality of new friendships and adjustment to university. Especially in the freshman year, a good network of friendships encourages students to engage in university life.

Compared to primary and secondary school, there is no fixed course route in the university because of the curriculum flexibility. Plus the online learning mode during the pandemic, there is less time to meet classmates face-to-face. Consequently, it is difficult to know each other well as well as to make friends, which will be detrimental to their social development. It is believed that university students are still at an age when they are relatively free and eager to develop interests and seek self-worth. Therefore, when it comes to making friends, the university should provide a good platform for students to find people who are equally passionate about the same thing.

Back to the practical problem, university students always come across challenges in finding appropriate project group mates. In university, most of the students need to form groups on their own. This is a great challenge for students, especially difficult for first-year students who are used to following teacher guidance. Commonly, they lack the time and communication to understand their classmates. It seems that strangeness is the main reason they feel anxious when forming groups.

In the 2020s, mobile phones became indispensable for everyone, especially for younger generations. Among all the apps, social media and dating apps are the two categories that involve connecting and reconnecting people. Most of the social media apps allow users to search for a person, join groups, express their views; the latter allows users to match with potential romantic partners/friends and chat with them through the in-build messenger function. When comparing these two types of apps, social media is more for friends or acquaintances to develop the relationship further, whilst dating apps are for strangers to start a new relationship. Due to the reason that university students are overcoming strangeness by establishing

relationships with others, the following paragraphs are focused on discussing dating apps.

Most teenagers meet new friends, not only romantic partners, through mobile dating apps. According to a survey conducted by Rakuten Insight [2] in 2020, nearly half of Hong Kong dating app users claimed that they frequently use dating apps to meet new people. As a result, an excellent mobile app for making friends could help university students to adapt to university life.

However, current solutions have deficiencies. According to our survey, more details will be shown in Section 1.3, almost half of the HKUST students are dissatisfied with the current solutions. For example, most mobile dating apps are not dedicated to university students. These apps do not take study-related factors into account. Besides, some apps for college students do not suggest friends to users.

Our project is to create a mobile friendship application in IOS and Android platforms that aims to provide an intelligent matching function, live chat, a second-hand goods online market, and an event pair platform. It suggests related friends based on the users' profile profile pictures and description.

1.2 Objective

Our main goal is to develop a mobile friendship application in order to provide a platform for university students to integrate into Campus. Therefore, our objectives are as follows:

The first objective is to provide a simple user-friendly interface for users to make friends without excessive effort. Instead of filling up a list of information, users solely input limited but essential data in their profile. Our application minimizes the manual efforts by automatically collecting user data, such as their feedback in the recommendation system and their profile picture. Based on those collected data, the system uses the designed algorithm to find potential users with some commonality between them.

The second objective is to facilitate social activities and communications among university students in an integrated app. Our system provides different ways of matching for users to widen their social circle. Users can find lists of personalized recommendations of people generated by the system while using the user-specified matching to find someone through criteria. Thus, students can conveniently meet and connect with like-minded people on campus. Furthermore, users can find various activities in this application. By joining activities with other users passionate about the same interests and collaborating to achieve the same goals, not only do users expand their social network, but they are also motivated to engage in diverse activities that they have never experienced before. Additionally, there is a trading platform for users to get or exchange goods such as course materials or second-hand laptops. During the purchase process, users can meet people who are interested in similar things.

The last objective is to improve the study life of university students. The chat room in the application provides an additional platform for users to interact with their classmates after the lesson so that users can discuss course-related topics together. Furthermore, our system would smartly suggest group mates and study partners to users in accordance with study-related factors such as their course. The algorithm would provide a more accurate result if users indicate their interest in specific courses or import their timetable into the application.

1.3 Survey of Social Platforms for HKUST Students

In June 2021, a survey was conducted among 42 students at HKUST. The purpose of the survey was to discover the students' habits in using social platforms and dating apps.

The survey was conducted by means of a questionnaire given to the students to complete, referred to Appendix D. There are two sections in the questionnaire. The first section is about the experience of making friends in university. The second part is concerned with reviews of using existing mobile apps.

The most important findings are as follows. During the pandemic, 86% of the students found it more difficult to meet friends. Almost half of the participants were willing to meet friends in university by using mobile dating apps, while a quarter of participants may probably use mobile dating apps to meet friends in university. Among 6 favorite social apps and dating apps, Gööp is the most popular app in HKUST. 61% of students have used Gööp. Coffee Meets Bagel and Tinder are the second and third popular apps. A quarter of students have used them.

In the second section, in terms of user experience, 40% of students found that the existing social and dating apps were just average, while almost one-third of students were dissatisfied with the apps. Almost half of the students would like to meet new friends using existing social and dating apps. Regarding the most enjoyable features of social and dating apps, almost half of students considered the chatting function, while about one-third of students liked the matching function.

If any conclusions may be drawn from the data, they are as follows. The majority of HKUST students found difficulties in meeting new friends. They would like to use social and dating apps to explore new friendships in university. Most of the students do not enjoy using existing social and dating apps.

1.4 Literature Review

To our knowledge, university-oriented friendship apps are not universal in Hong Kong. On the other hand, there are numerous relevant social media apps and dating apps, which have been widely studied and evaluated in academia. Considering that the nature of these apps is related to our project, they help us to understand their matching and business practices so that we can apply them in the university environment. Therefore, this review focuses on two types of apps: widely used dating apps in Hong Kong and the most popular app connecting students in the university.

1.4.1 Tinder

Tinder is a global online dating app where users can expand their social network via finding a potential matching created by its own algorithm. Figure 1 demonstrates the user, which is recommended by Tinder. Users can anonymously swipe right to accept someone or swipe left if they are uninterested. To boost the success rate, users can use the Super Like function to notify the target that they are interested. If both users swipe right on each other, they are regarded as matched up. Tinder then provides a chatbox for them to communicate. Regarding the matching algorithm, Tinder used to rank people by Elo score narrowly. This algorithm adjusts the score of each user, depending on those who swiped right on the user. Tinder tends to assign other users with a similar score to the user. After 2019, Tinder no longer considers the Elo algorithm as their unique factor [3]. Other parameters such as the behavior in the past 24 hours and activeness are included in the current algorithm.

Although Tinder has a robust algorithm for users to find someone, it does not provide any searching method to specify the matching result based on customized criteria. Furthermore, Tinder users are from all walks of life, which is challenging for students to have a match in the same university, not to mention the same major.

Figure 1. A snapshot of Tinder with the UI of the Matching Screen

1.4.2 Meetup

Meetup is a social media platform that allows users to find and build connections via participating in various activities [4]. By searching for keywords and locations or looking for an event that will start soon as shown in Figure 2, users can discover a myriad of events which allow users to connect with people who share the same interest or goals. Similar to other dating apps, Meetup provides internal messaging options for users to stay in touch. One outstanding feature of Meetup is its group formation system. Not only do users interact with a group of people with common interests, but they can also receive alerts once other groupmates schedule new events.

Since Meetup considers location as a mandatory field for every search, users have no choice but to search the identical keywords with different positions multiple times to explore more results.

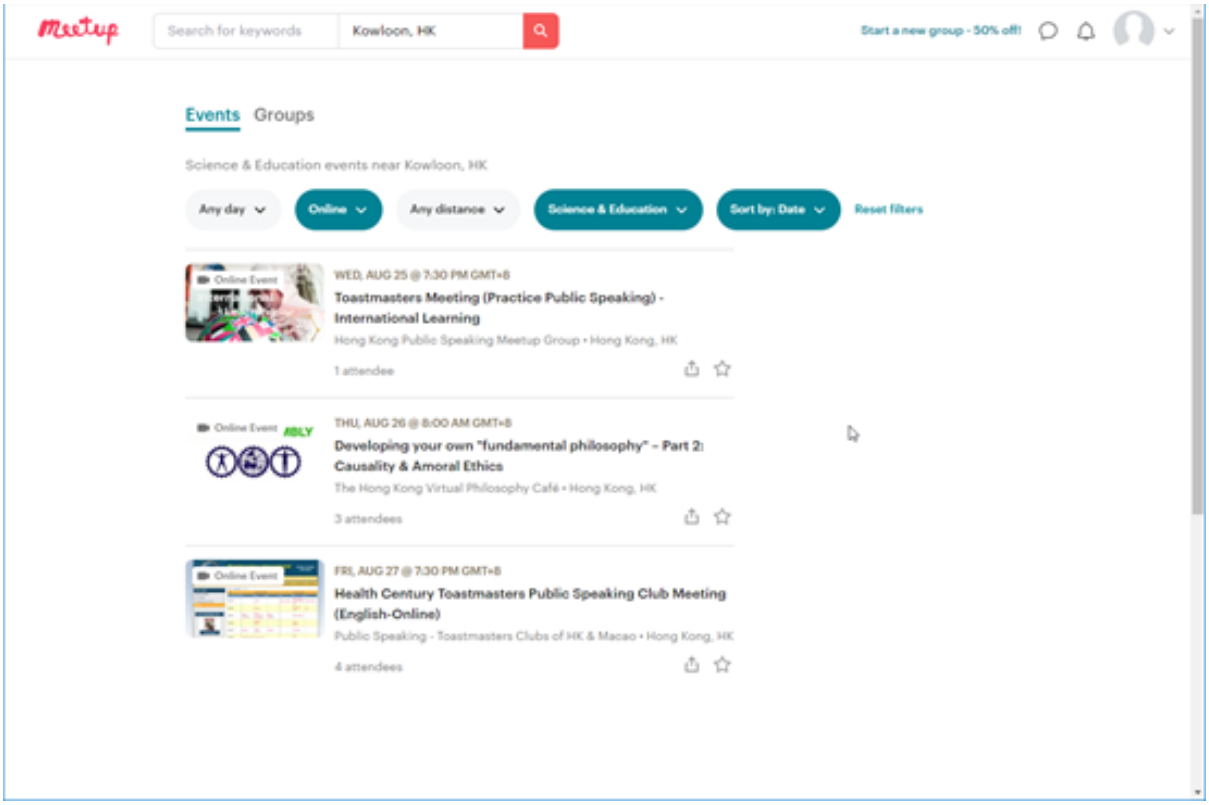


Figure 2. A snapshot of MeetUp, showing the searching result filtered by location and category

1.4.3 Happn

Happn [5] is a mobile dating app that connects users to the people they cross paths with every day. The app collects the GPS information of the users and recommends people to users. Whenever the users cross paths with another Happn in the street, their profiles will appear in both of their Timelines. Figure 3 demonstrates the matching and chatbox interfaces of Happn. Happn provides an opportunity for users to catch up with the people they met. Moreover, Happn could synchronize with some popular social media accounts which saves the time of editing profiles in the app. Besides, users could edit their states as dining, drinking or movie, etc. Users could find some people with the same interest nearby.

However, Happn only shows the people that users have crossed paths with. There are not enough criteria to make sure the matching is suitable. If the app is used in universities, more parameters should be taken into account.

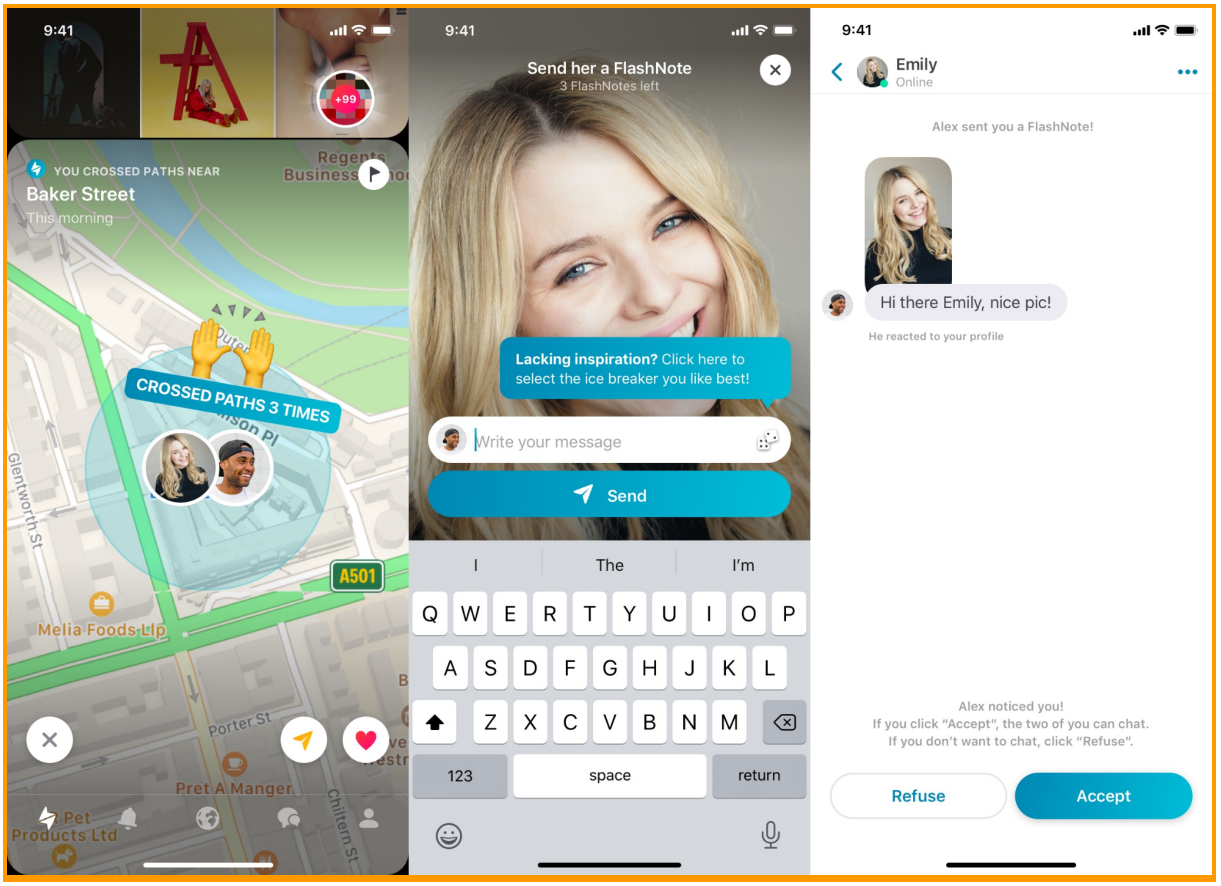


Figure 3. Snapshots of Happn with the UI of the Matching Screen and chatbox

1.4.4 Gööp

In Hong Kong, Gööp [6] is the most famous app which connects people in universities. There is a certain number of university students who meet friends or seek help in Gööp's forum. Figure 4 demonstrates the forum of Gööp.

Gööp is a discussion platform that provides a space for live chat. Unlike general forums, Gööp divides different communities into their dedicated discussion forums, creating sub forums under these forums. For example, the University of Hong Kong has HKU Talk, including sub-forums such as dating, part-time work, and living in Hall. In addition, Gööp also has discussion areas such as tourism lectures, advancement lectures, and football lectures for members who have common interests to discuss together. Moreover, Gööp provides high privacy for discussion. Some discussion areas are hidden unless the users have an invitation code. Especially in universities' forums, the users cannot comment in the forums unless they are verified by email. Besides, Gööp provides anonymous comments and messages. Users have different identifiers in different posts. Therefore, users do not worry about being "doxed".

Nonetheless, the app lacks an intelligent recommendation system, users need to open a new post listing their requirements and wait for the reply, which is harder for users to find a new friendship.

Figure 4. A snapshot of Goop‘s forum

In summary, Tinder, Happn and Meetup aim to connect people from different areas, which is not specific to university students. Being a forum-based platform, Gööp lacks a function for users to find friends effectively. Tinder matches users based on their profile while Meetup gathers people by holding events. Happen makes use of GPS to connect different users. The use of GPS allows users to find friends within a close area conveniently. Therefore, we will consider using it as a reference in developing our mobile application. Eventually, all four applications provide a chat room for users since it is a simple method to communicate with others.

2. Methodology

2.1 System Design

Since our app is designed for university students, we want to verify the user identification through emails in registration. In this stage, we focused on HKUST students. But our app can be extended and customized to be used by other local and international universities in the later stages. After registration, we need some personal information of users to generate the matching result. Users can update their school timetable, profile picture and personal information in the Profile Function. After updating the profile, we will analyze the profile of users and recommend some matched people to the users. The matched result will be shown in the Match Function. Users can view, accept and decline requests on the page of Match Function. Moreover, in order to facilitate social activities in the university, our app provides an event platform. In the Event Page, users can host events, view and join the events created by others. In addition, university students usually have many reusable materials such as course materials, second-hand laptops, and furniture in the residential hall. In our app, we provide a user-friendly way for students to trade second-hand goods. We have a Trade Function for students to launch, view the products. No matter if the users match a friend, join an event or target a product, they need to contact the person. Therefore, we provide a Chat Function to facilitate communication among the users.

The detailed system flow is shown in Figure 5. There is a Login Page in our application, including the Forget Password and Sign Up functions in the beginning. Users can create an account on the Sign Up Page for the new users by giving basic information such as ITSC email and password. The email address must not duplicate with other users, and the server will send a confirmation email to the user for verification. It will be a verification stage for checking if the email is legitimate or not. After registration, the users will go back to the Login Page and jump to the Chat Page after logging in. Registered users who forget their password can retrieve their password on the Login Page by email and validation code.

Users can access the Chat Page, Match Page, Trade Page, Event Page and Setting Page through the navigation bar. When the user uses the Match Page, the application will send a request to the server for processing. The server will decide what kind of result should be displayed and then extract relevant data from the database. A new chat room will be generated when the user clicks on the icon from the results list and waits for the acceptance of the opponent. When the user uses the Event Page, the application will send a request to the server, and the server will list the events chronologically and send it back to the application. Also, they can add, edit or delete their own events and all actions will be recorded into the database. When the user uses the Trade Page, the application will send a request to the server, and the server will list the goods chronologically and send it back to the application. Moreover, they can add, edit or delete their own goods and all actions will be recorded into the database. When users update their profile in the Setting Page, they can manually type in their information or upload their timetable. The server will extract their enrolled class from the timetable or keywords from their description and then store the information into the database to further use in the recommendation algorithm.

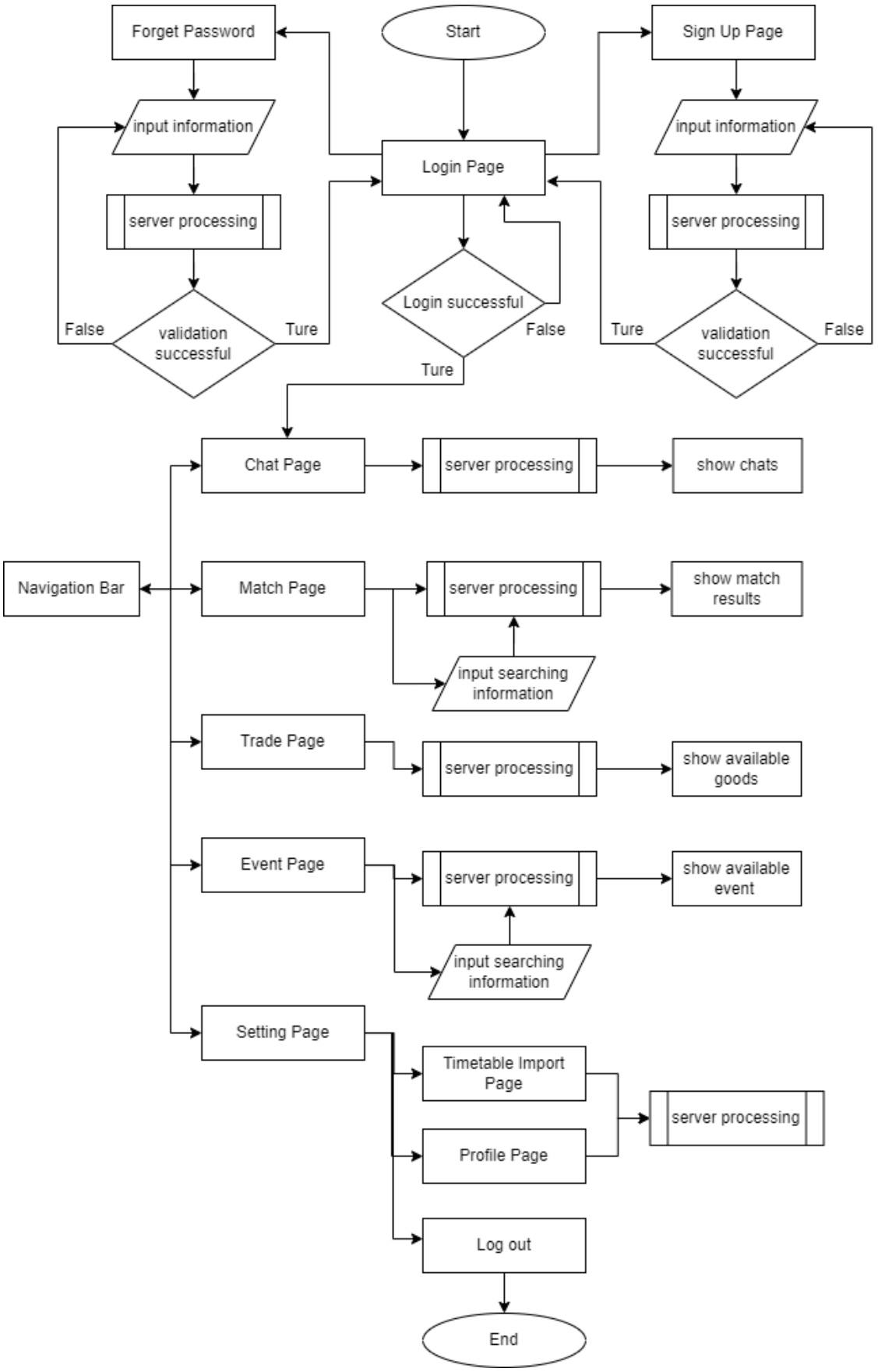


Figure 5. The system flow of the application

2.2 Functionality Design

2.2.1 Splash/Onboard Screen

Our application is designed for all university students so the application name and logo are designed with the signature “U”. Splash screens are typically used to notify the user that the program is in the process of loading or running. There are dynamic elements added to the logo which make it alive. Then, the application will check whether you have installed this app before. If you are a new user, an onboard screen with a brief introduction to our application will be shown to let users get ready for our apps. If users have clicked the "Remember me" button before, it will guide them to the home page (Chat Page) after the Splash Screen. The screenshots of the Splash Page and Onboard Pages are shown in Figure 6.

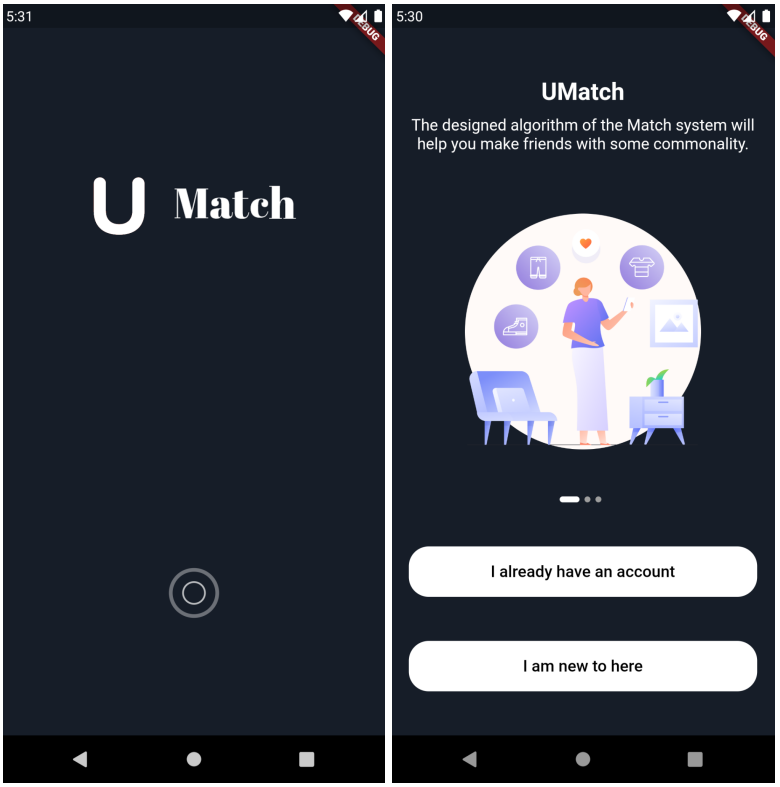


Figure 6. The screenshots of Splash Page and Onboard Pages

2.2.2 Login/Signup System

In the early stage, we only customize our app for HKUST students first. Our application is supposed to support university email sign-in methods. In the process of registration, we will ask users to fill in their HKUST ITSC email and password during registration. After finishing the registration, users need to verify their account through their HKUST email. We want to make sure that all users come from HKUST. After the verification, users can login to our application. Our Login system will store the Login information to simplify the Login procedure. When the users leave the application and re-open it, the Login Page will be skipped and directly entered to the home page (Chat Page). The screenshots of the Login system are shown in Figure 7.

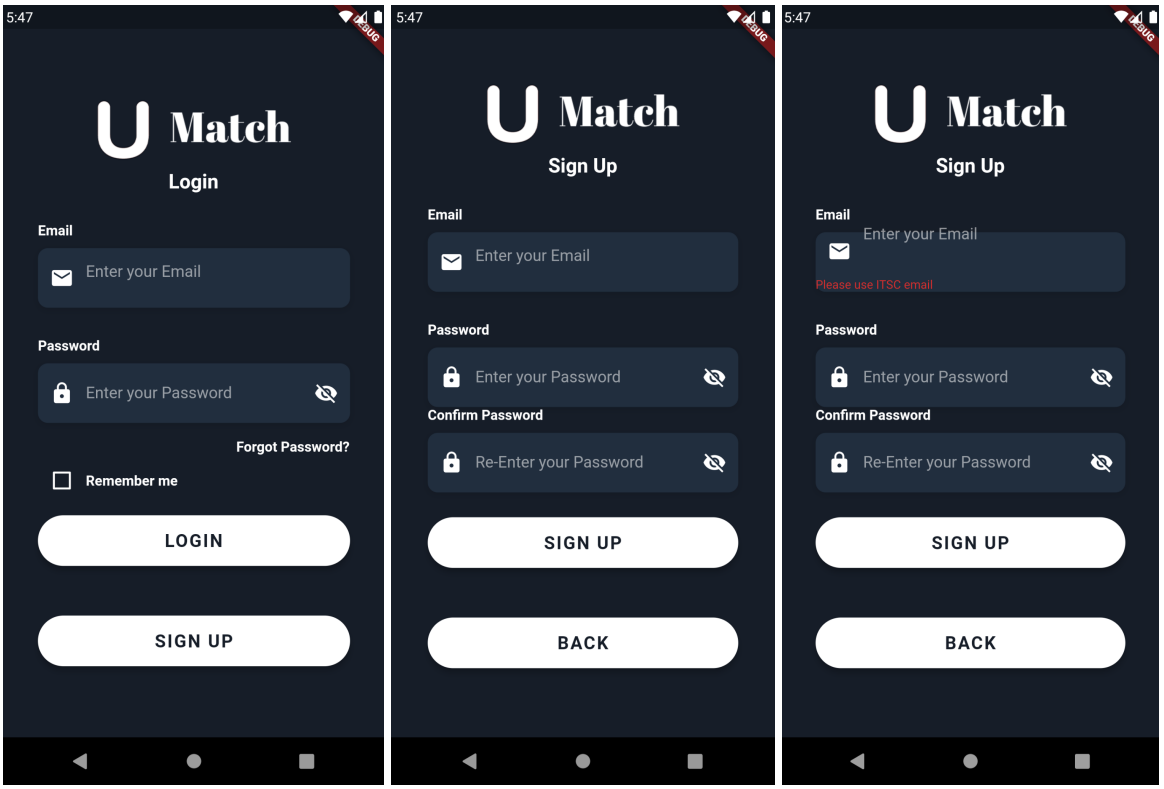


Figure 7. The screenshots of Login and Register Pages

2.2.3 Edit Profile

2.2.3.1 Building Up the Profile

Users can build up their personal information as shown in Figure 8. Users can manually input their username, description, gender, major, courses and hashtags on the Profile Page. When users input the hashtags, the system will provide a list of suggested hashtags sorted by alphabetical order. Moreover, users can select their interests on Profile Survey Page, which will be mentioned in Section 2.2.3.2, and update courses by the Import Timetable function, which will be mentioned in Section 2.2.3.3.

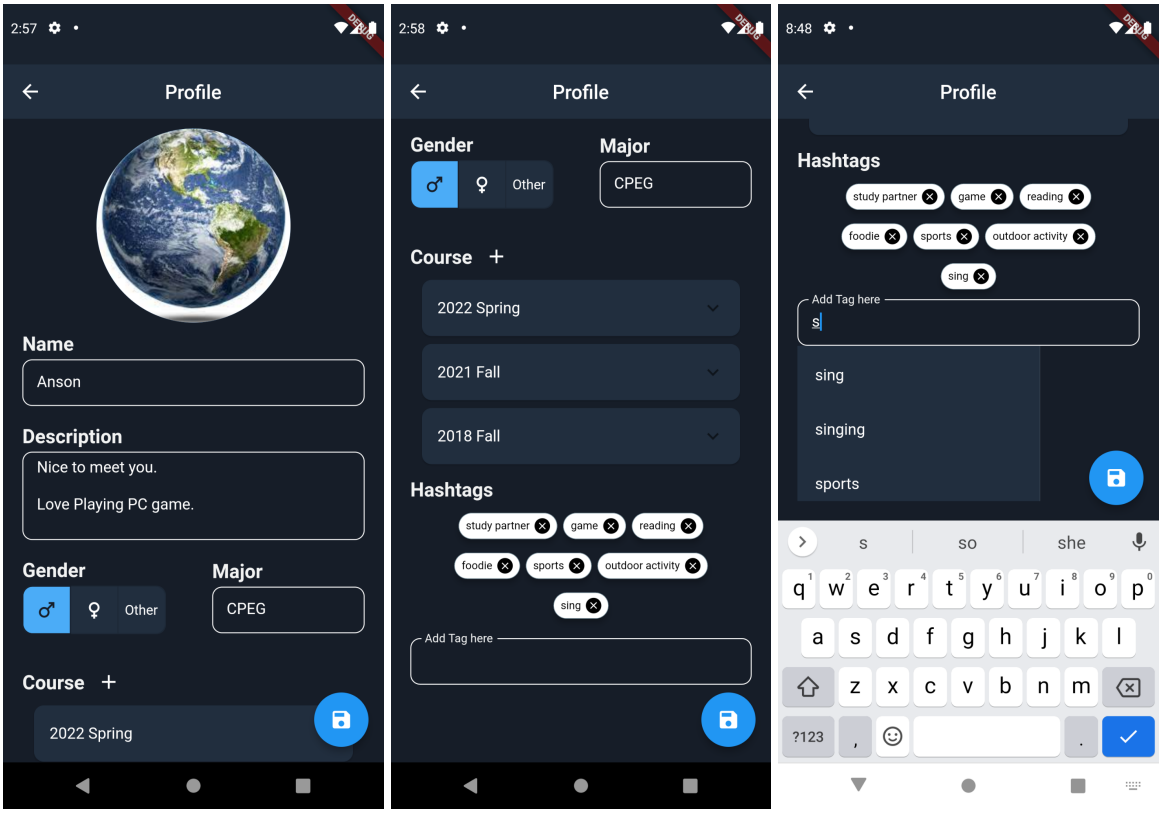


Figure 8. The first two screenshots are the Profile Page and the right screenshot shows the suggestion list when inputting hashtags

2.2.3.2 Profile Survey Page

When users first go to the Profile Page, there is a survey to collect the interests of users. They can choose to answer or skip the survey questions, which will be used for the recommendation system. Users can complete the survey the next time if they press the “Select later” next to the “That’s all” button. When a user indicates its interest, our algorithm can generate more accurate results based on more factors. When a user skips the survey, the matching will randomly assign matching on the Match Page, or users can find the preferable matching by typing specific keywords. The screenshots of the Survey Page are shown in Figure 9.

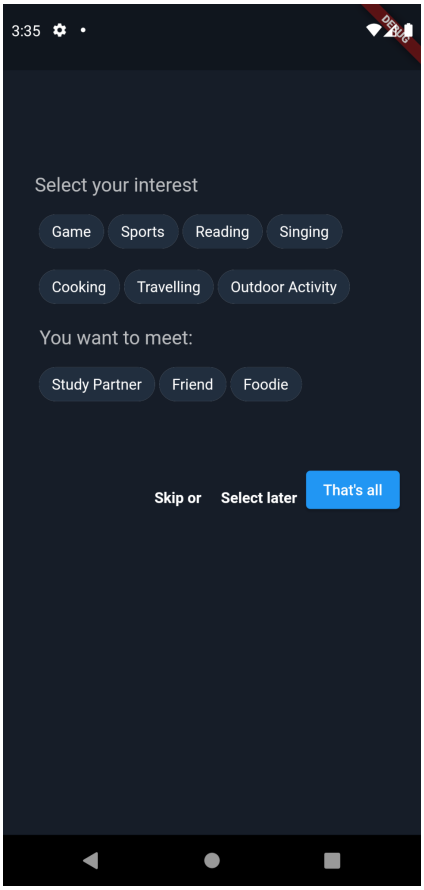


Figure 9. The screenshot of the Survey Page

2.2.3.3 Import Timetable

Users can upload their timetables by importing the .ics file which can be downloaded from HKUST Timetable Planner (https://admlu65.ust.hk/). A .ics file stores the information of different courses such as the description, the start and end time, and the location of the course. In this project, the course code, section, and year of study will be extracted and stored in the database for the algorithm used. Users can view and edit their course information grouped by year and semester on the Profile Page as shown in Figure 10.

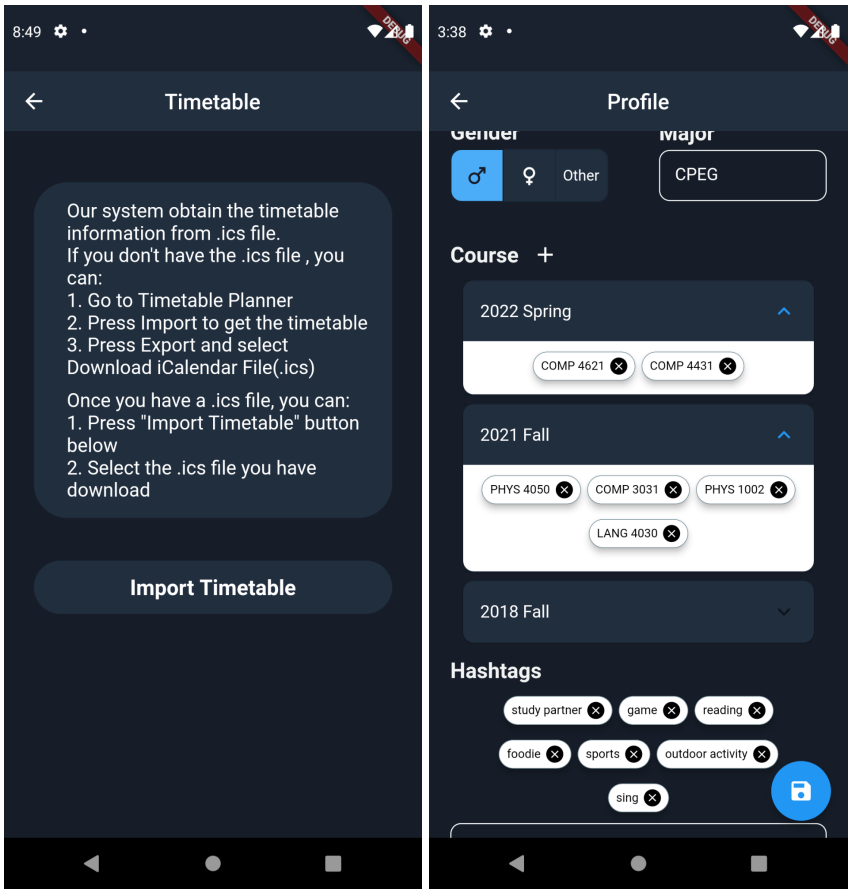


Figure 10. The screenshot of the Import Timetable and Profile Page showing course information

2.2.4 Chat Function

2.2.4.1 Chat Room

One of the main objectives of our application is to connect people. Therefore, communication takes an essential role. Our application provides a chat function to users for better communication. After the user matches a friend on the Match Page, joins an event on the Event Page or wants to purchase a product in Trade Page, our application will connect users by providing them a chat room. To support Event Function, we will provide a group chat room for each event. Participants can talk to multiple users at the same time. To support Trade Function, our application will automatically generate an offer greeting message to the seller. On the Chat Page, users can see their friends and choose to chat with. By tapping the name of users' friend, users will enter the chat room. On the chat room, users can send and receive messages to their friends in real time. The screenshots of the Chat Page and chat room are shown in Figure 11.

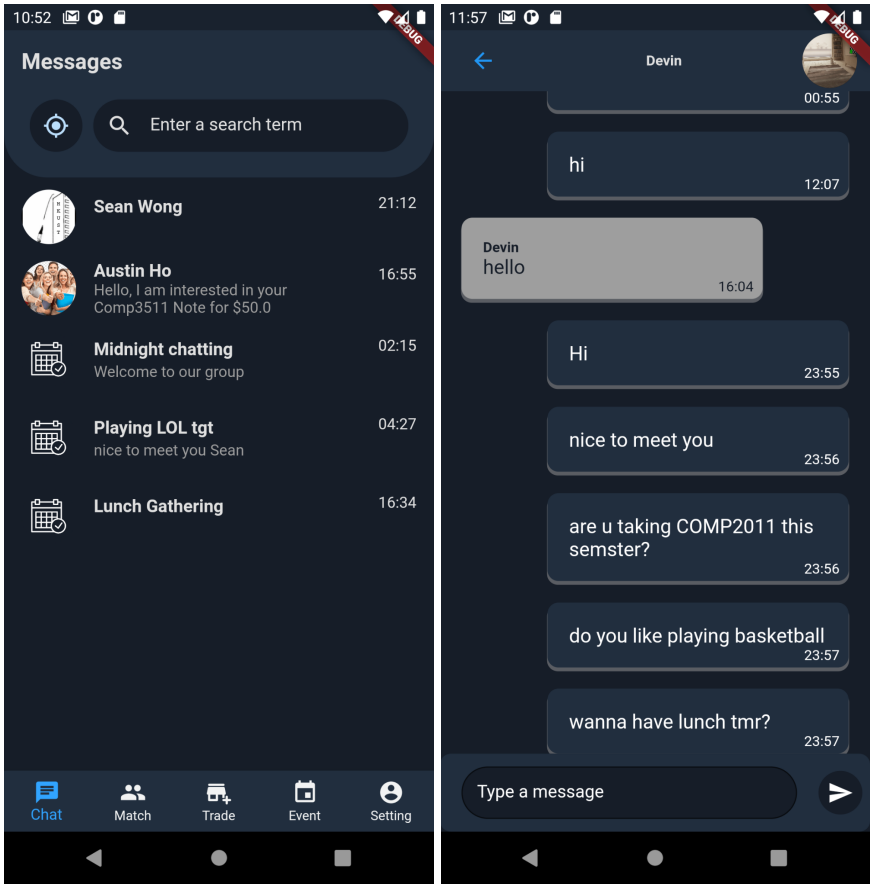


Figure 11. The screenshots of Chat Page and chat room

2.2.4.2 Group Chat

Our application will create a group chat for all participants in the event. Therefore, our Chat Function supports users to chat in groups. When a user successfully joins an event, our application will automatically put the user into the group of that event. In the group chat, all users can communicate at the same time. It facilitates the process of communication. Moreover, when a user taps the event icon in the top right corner of the chat room page, our application will show a list of all users in the group. Users can view the profile of other group members and send them friend requests. The screenshots of the Group Chat page, chat room member list and profile of member are shown in Figure 12.

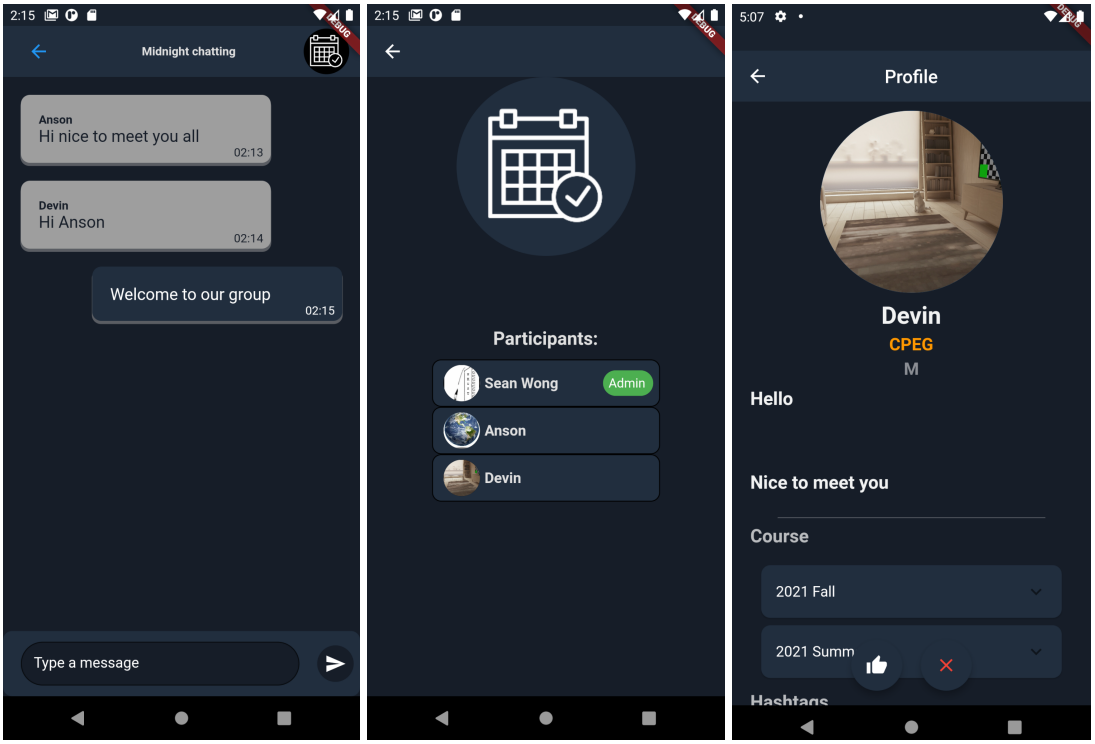


Figure 12. The screenshots of Group Chat page, chat room member list and profile of member

2.2.4.3 Search Nearby Users

Sometimes, students meet friends in the lectures and tutorials. However, it is time consuming to exchange the phone numbers. Hence, our application provides a function to chat with the nearby users. With our application, users can get the contacts of each other in an instant. First, users need to connect with other users by tapping the “plus” button in the top left corner of the Chat Page. One of the users taps “Wait” to wait to be searched by others. Meanwhile, another user taps “Search” to search for this user. After that, both of them can see the names of each other. After the confirmation of being friends, the two users are connected. Then they can chat with each other on the Chat Page. The screenshots of “Searching” and “Waiting for search” are shown in Figure 13.

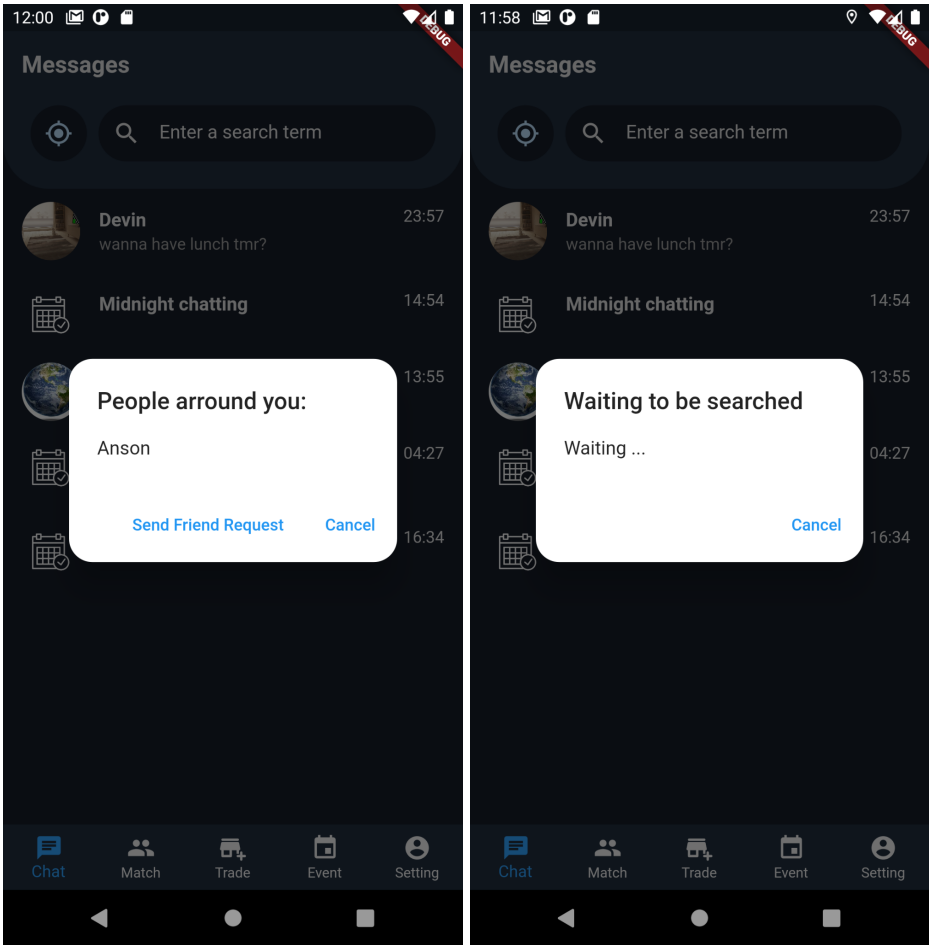


Figure 13. The screenshots of Searching and Waiting for search

2.2.5 Match Function

When the user uses the match function, the app will read the database and display the recommended results. The recommended results will be shown in 2 separated lists, “Match By Courses” and “Match By interests”, which are based on 2 different algorithms mentioned in 2.4.4. Furthermore, users can make use of the filter to find other users with specific keywords. Opening the filter as shown in Figure 14, users can obtain suggestions regarding the major and courses when inputting keywords. A new chat room will be generated when the user clicks on the icon from the results list and waits for the acceptance of the opponent. The screenshots of the Match Page and the filter are shown in Figure 14.

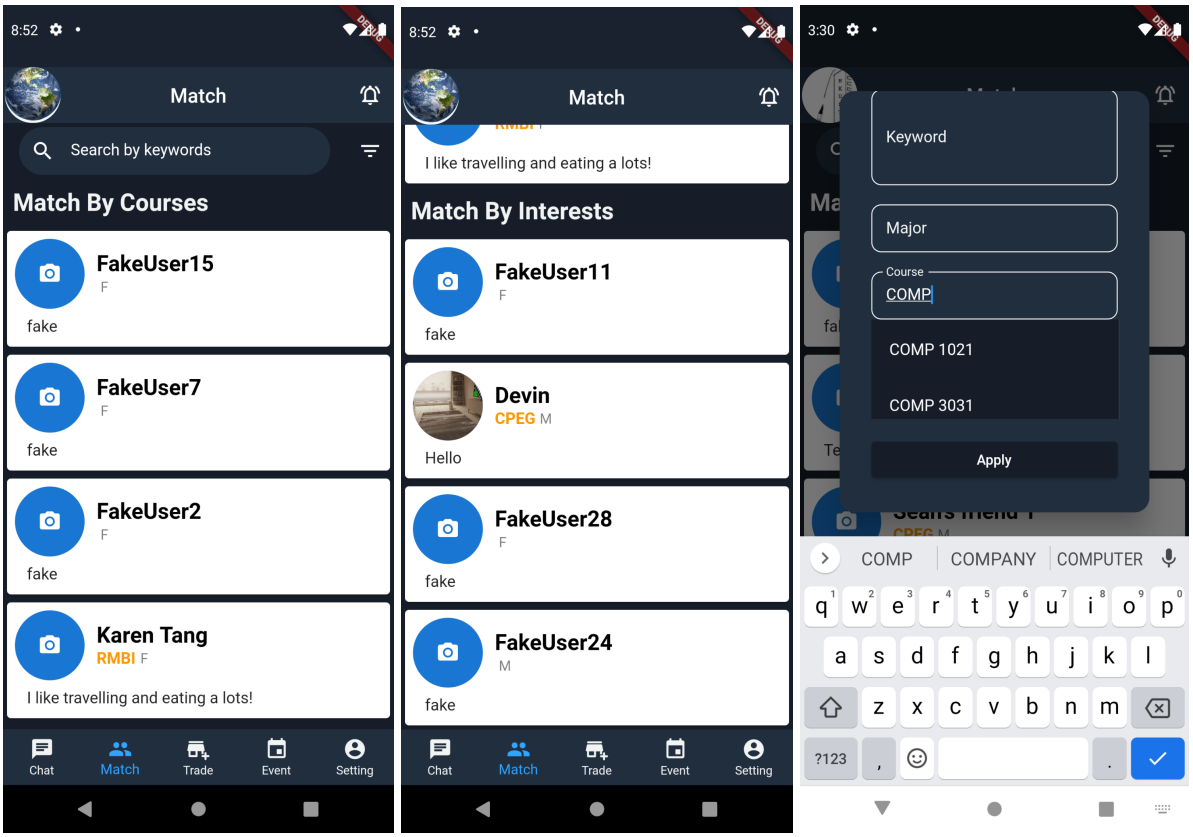


Figure 14. The first two screenshots is the Match Page and the left screenshot is the filter suggesting courses

2.2.6 Event Function

Our application provides a platform for users to engage in different activities. Users can view a list of events in chronological order where events having the earliest start day will be displayed on the front of the Event Page. Users can also use the search bar to filter the result. After clicking on the particular event to see the details, users can press the “Contact Me Now!” at the bottom of the page to apply if they are interested. Users can create their events on the Event Edit Page with the left floating “plus” button. Users are required to fill in the title, the number of participants, date, start time and end time of the event. They can include description, location, and category if they want to. After pressing the “home” button on the top-right corner of the Event Page, users can manage events they have applied, edit the events they have created, and view the events they have participated in. Figure 15 demonstrates the default page of the Event Function, the page showing the details of the event ”Midnight chatting” and the Event Edit Page.

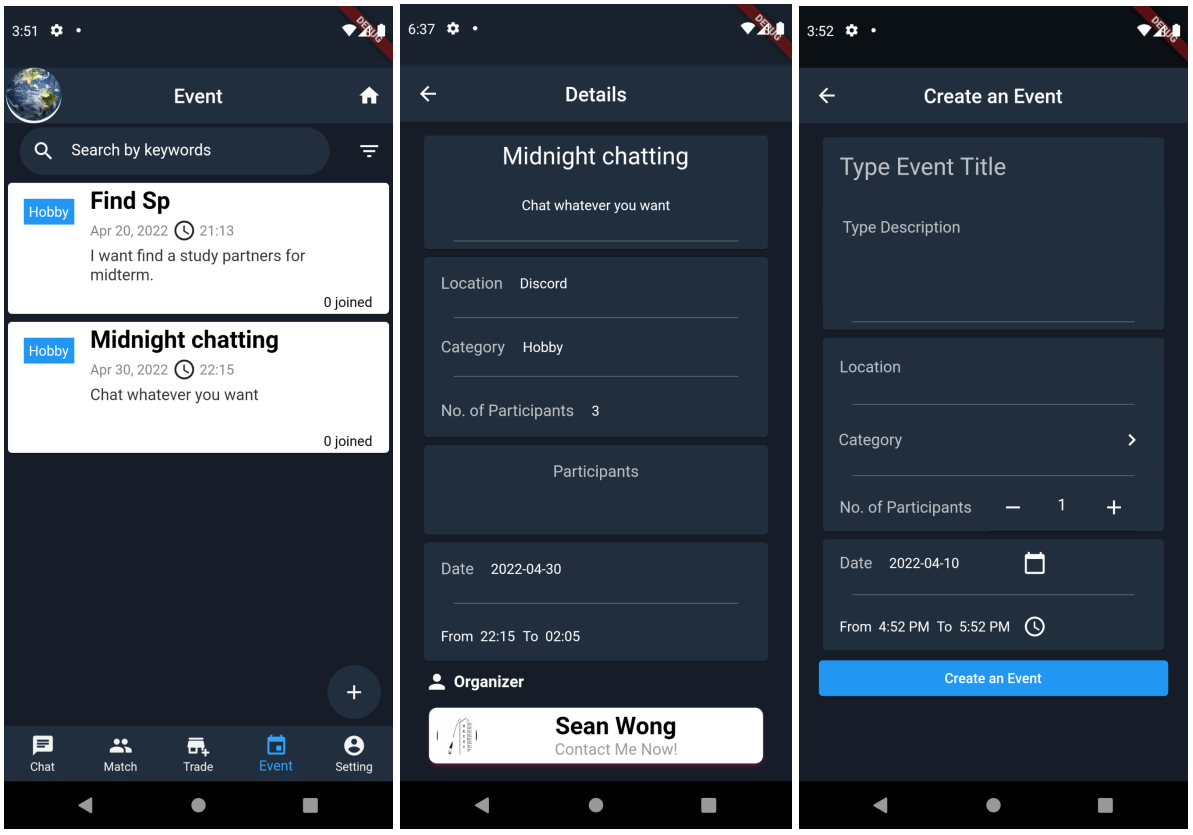


Figure 15. The left screenshot is the Event Page, the middle screenshot is the detail page of Event “Midnight chatting” and the right screenshot is the Event Edit Page

2.2.7 Trade Function

Our application provides a trading platform for users to mainly sell idle goods such as second-hand textbooks or clothes. When users go to the Trade Page, our system will generate product previews in chronological order. There is a search bar, and filter button containing different sorting functions for users to facilitate their product searching. For example, users can directly type keywords such as the product name or course ID in the search bar to search for related products. Clicking the preview, users can go to the Product Details Page which contains all the information about the products and the recommendation related to the product category users currently see. Inside the Product Details Page, users can click the “heart” button to like the product and contact the seller. Users can click the heart button on the top right corner of the Trade Page to find out what they have liked. Also, “notification” buttons are next to the “heart” button to retrieve announcements when the products they like were updated or your products were liked. Users can click the notification to link to the Product Details of the products. Users can upload products to the trading platform by clicking the floating “add” button and it will guide you to the Sell Your Product Page. After filling in all the information, you can click the “manage your uploaded product” button on the Setting Page to edit your uploaded products. Figure 16 demonstrates the Trade Page, the Sell Your Product Page, the Product Details Page, the Notification Page, and the Edit Your Product Page.

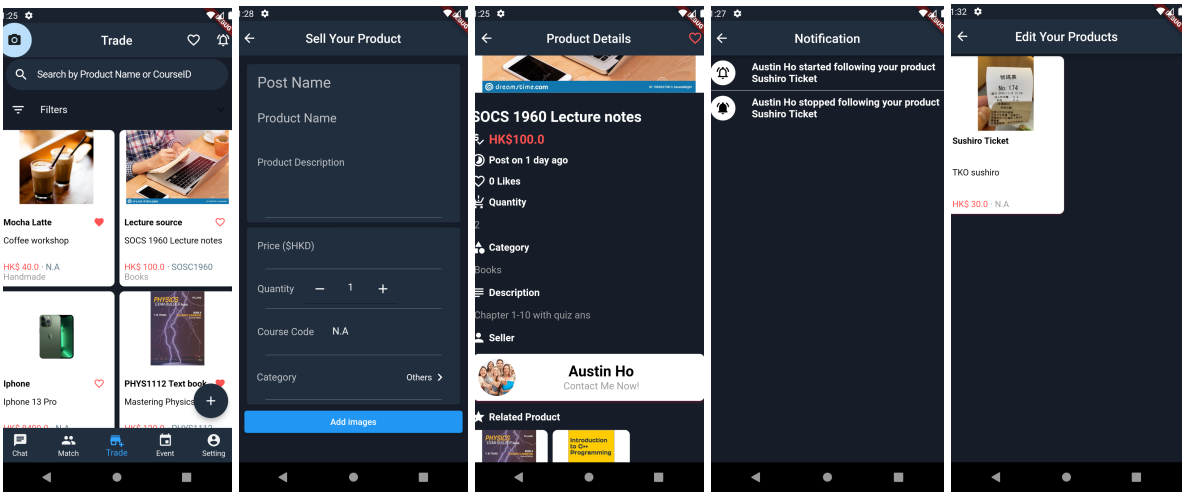


Figure 16. The left screenshoot is the Trade main page, the middle first is the Product Details Page, the middle second is the Sell Your Product Page, the middle third is the Notification Page and the right screenshot is the the Edit Your Product Page

2.3 Database

We adopt Firebase Cloud Firestore, which is a NoSQL database, as our database. It is a non-relational database, so it is hard to describe data relationships using ER diagrams. The relationship and purpose are shown in Table 1 below. Table 2 describes the database algoMatchResult, which is used by the recommendation system and the Match Function. Regarding the Chat Function, Table 3 to Table 4 illustrate the chatRoom database, which stores the chat information. And Table 5 stores the GPS information of the users. The event database which will be used in the Event Function is listed in Table 6. Table 7 to Table 8 show the notifications, while Table 9 stores the product information. These three databases are applied in the Trade Function. Table 10 stores the user information, which is used in all functions. Table 11 stores the suggestion list of major, course and hashtag which will be used in Match Function and Profile Function.

Table Name	**Purpose**
algoMatchResult	Store the result of the algorithm for users
chatRoom	Store the chat information and messages of the users
searchNearby	Store GPS address of users
event	Store the information of event
notifications	Store the notification of the users
products	Store the product information
user	Store user information
autocomplete	Store the suggestion list of hashtag,major and course

Table 1. Database tables

Column name	Data Type	Description	Default value
resultByCourse	array	The string array of the user ids generated based on course	
resultByTag	array	The string array of the user ids generated based on hashtags	

Table 2. Collection: Schema of Document in algoMatchResult

Column name	Data Type	Description	Default value
chatRoomId	string	The unique ID of Chat Room	
Users	array	The string array of the emails of two users in the Chat Room	
Chats	collection	Store the message information of the Chat Room	

Table 3. Collection: Schema of Document in chatRoom

Column name	Data Type	Description	Default value
currentemail	string	the email of sender	
message	string	the text message	
senttoemail	string	the email of receiver	
time	number	the time of sending message	

Table 4. Collection: Schema of Document in Chats (under ChatRoom)

Column name	Data Type	Description	Default value
user_wait	string	The email address of the user who is waiting to be searched	
name	string	The name of the user who is waiting to be searched	
lat	number	The latitude of GPS of the user who is waiting to be searched	
long	number	The longitude of GPS of the user who is waiting to be searched	
user_search	string	The email address of the user who sent friend request to this user	
user_search_name	string	The name of the user who sent friend request to this user	

Table 5. Collection: Schema of Document in searchNearby

Column name	Data Type	Description	Default value
category	String	The type of the event	
created_by	String	The id of the user creating the event	
description	String	The description of the event	
from	Timestamp	The start date and time of the event	
location	String	The Location that events take place	
name	String	The name of the event	
noOfParticipants	number	Total number of participants	“1”
participantsList	array	The string array of the user id that takes part in the event	
to	Timestamp	The end date and time of the event	
waitingList	array	The string array of the user id that waiting for the owner to accept	

Table 6. Collection: Schema of Document in event

Column name	Data Type	Description	Default value
activities	collection	store the notification of all the users	

Table 7. Collection: Schema of Document in notifications

Column name	Data Type	Description	Default value
activityDate	timestamp	the creation time of the notification	
content	string	the content of the notification	
isRead	boolean	the notification has been seen	“False”
productID	string	the related product of the notification	

Table 8. Collection: Schema of Document in activities (under notifications)

Column name	Data Type	Description	Default value
createDate	timestamp	the date of post creation	
updateDate	timestamp	the date of post update	
followers	array	The followers' UID	
isDeleted	boolean	The post has been deleted	"False"
isSold	boolean	The product has been sold	"False"
isVisible	boolean	The post can be seen	"True"
postName	string	The name of the post	
courseID	string	The related course code of the product	
description	string	The description of the product	
productCategory	string	The category of the product	
productPrice	number	The price of the product	
productQuantity	number	The quantity of the product	1
likes	number	The quantity of the followers	0
productImages	array	The product pictures	
userID	string	The UID of the seller	
commentName	string	The name of the commenter	

commentUid	string	The uid of the commenter	
content	string	The content of the comment	
score	number	The rating towards the sellers from buyers	

Table 9. Collection: Schema of Document in products

Column name	Data Type	Description	Default value
dataForAlgo	map	A map of course information containing course code, section and year of study	
declined	array	The string array of users' UID that the current user decline in the Match Function	
description	string	The description of the user	
descriptionHistory	array	The string array of descriptions in the past	
email	string	The ITSC email of the user	
friends	array	The string array of emails of friends	
major	string	The major of the user	
name	string	The name of the user	
pending	array	The string array of users' UID that the current user send request in the Match Function	
sex	string	The gender of the user	
tag	array	The string array of hashtags describing user	
tagid	bool	User has done the survey or not	"False"
uid	string	The UID of the user	

Table 10. Collection: Schema of Document in user

Column name	Data Type	Description	Default value
course	array	A string array of course codes	
major	array	A string array of majors	
taglist	array	A string array of hashtags	

Table 11. Collection: Schema of Document in autocomplete

2.4 Implementation

Section 2.4.1 introduces the tools we used to implement the project. Section 2.4.2 describes the Login system. Section 2.4.3 describes the Profile Function and the Import Timetable Function which will be used to collect the user's data. Section 2.4.4 describes the recommendation system we used in the Match Function. Section 2.4.5 describes the workflow of the Chat Function. Section 2.4.6 discusses the implementation of the Match Function. Section 2.4.7 explains the system flow of the Event Function and Section 2.4.8 describes the Trade Function.

2.4.1 Chosen Tools

We have decided to use the Google Cloud Platform (GCP) for deploying our recommendation system while Firebase as the backend system and Flutter is used in building the application. Moreover, we adopted Algolia to implement a better search function for Match Function and Event Function. Meanwhile, we use Fuzzy to implement search function in Trade Function.

2.4.1.1 Google Cloud Platform (GCP)

We deployed a recommendation system on the Google Cloud Platform (GCP) [7] to serve recommendations. The architecture of the production system is shown in Figure 17. Table 12 below shows the role of each component in GCP. Once GCP receives the request from the end user, the app engine will read the input data from Firestore and return the recommendation result to the Firestore. After that, the end users can read the recommendation result from the Firestore.

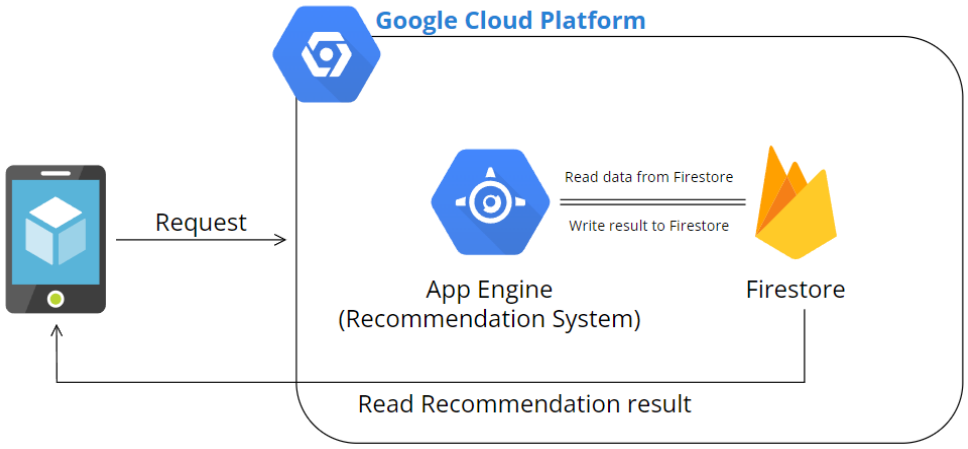


Figure 17. The architecture of the deployed recommendation system

Components of Google Cloud Platform	Service in GCP
Firestore	Integrated with firebase to store the data.
App Engine	Deploy an App Engine endpoint to serve recommendations. The engine will base on the input from firestore and return the recommendation result.

Table 12. The usage of the components in Google Cloud Platform

2.4.1.2 Firebase

Firebase [8] is a software development platform that helps in building web and mobile applications with its 18 services. The most useful services for us are authentication, cloud database, and cloud storage. Especially the cloud database, it is different to the traditional relational database management system (RDBMS). Cloud databases are more flexible for us to make changes to our database. Besides, Firebase provides many purposeful APIs for both IOS and Android Application development. On top of that, Firebase works well with Flutter since both of them belong to Google. Originally, we proposed to use SQLite as a database system in our application. Due to the provided cloud services, high flexibility of the database management system and the high compatibility with Flutter, we decided to use Firebase instead of using RDBMS.

2.4.1.3 Flutter

Flutter is Google's portable UI toolkit for crafting beautiful, natively compiled applications for mobile, web, and desktop from a single codebase [9]. Flutter works with existing code, is used by developers and organizations around the world, and is free and open source. We decided to make our application available on both IOS and Android platforms. By using Flutter, we only need one set of code to build both IOS and Android Application. Therefore, it reduced the code development time and hence increased the time-to-market speed. In addition, the coding technique of Flutter is relatively simple. One of the biggest reasons for using Flutter is the ability to customize anything we see on the screen, regardless of how complex it may be. It was an incredible experience for using Flutter to build the UI design.

2.4.1.4 Algolia

As a NoSQL database, Firebase has limitations in querying documents. For example, there is no “like” operator in Firebase. Hence, we adopted Algolia [10] to conduct the search function in this project. Algolia provides a fast search and easy-to-implement search engine for our app to do a full-text search on the Match Page and the Event Page. Furthermore, the free version of Algolia can store 10000 records and provide 10000 search requests per month, which is sufficient for this project.

2.4.1.5 Fuzzy

Fuzzy [11] search is done through a fuzzy matching procedure, which returns a list of results based on possible relevance even though the search parameter word and spelling may not match exactly. Exact and highly relevant matches appear near the top of the list. A subjective relevance rating, usually in the form of a percentage, can be given. If the user has only a vague or general idea of a topic, or doesn't know what to look for, a fuzzy search is much more powerful than an exact search. Hence, we adopt Fuzzy search to conduct the search function in Trade.

2.4.2 Login/Signup System

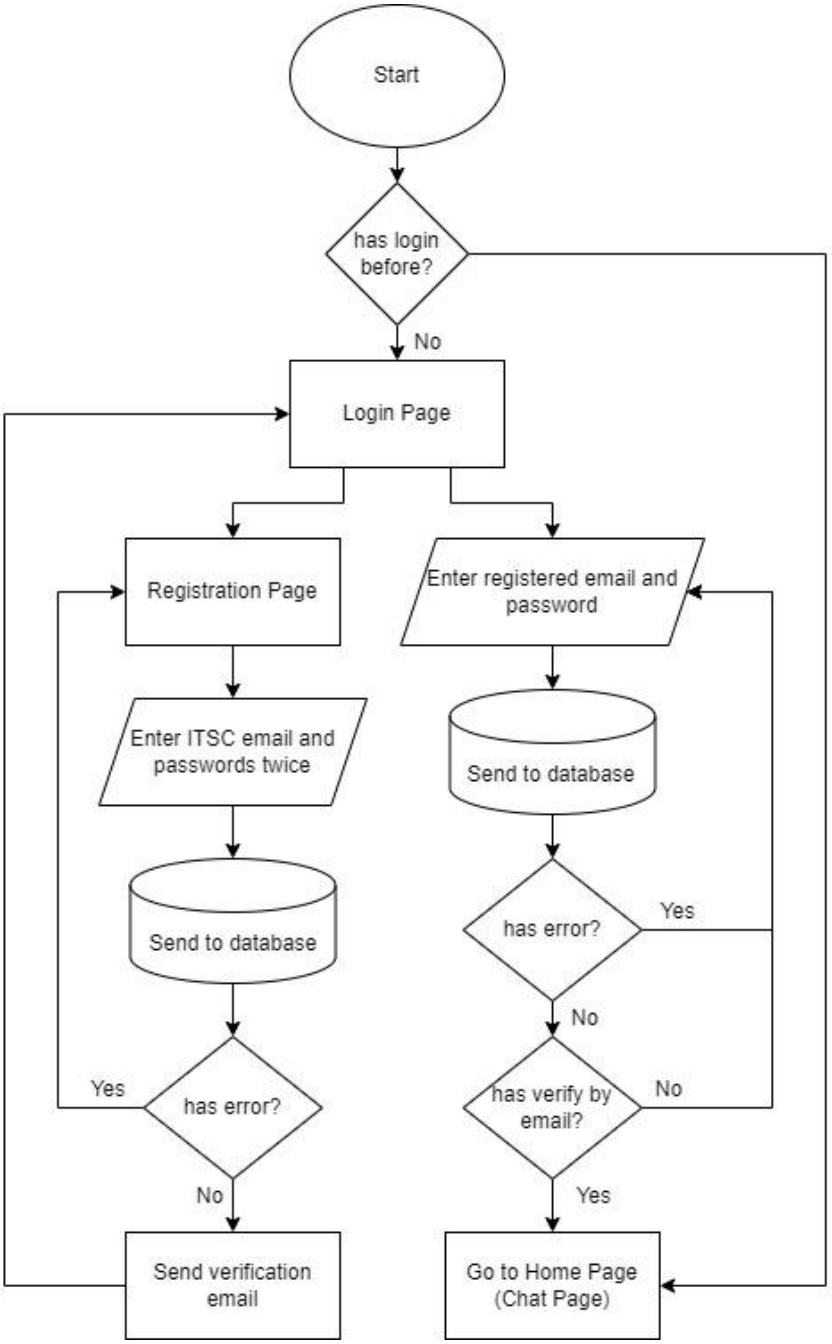


Figure 18. The system flow of the Login and Logout system

Figure 18 demonstrates the system flow of the Login and Logout system. When users launch our application, it will check if the users have logged in before. If yes, it will directly go to the home page (Chat Page). If not, users will enter the Login Page. On the Login Page, users can choose to log in or sign up.

In the Sign Up system, our application requests users to enter their ITSC email and validate the passwords twice. After the registration, a verification email will be sent to the ITSC email of the user. Users need to verify their emails.

In the Login system, our application requests users to enter their registered email and password. If there is no problem with the account information and the email has been verified, the users will get into the Home Page(Chat Page). If not, users will need to enter their email and password again or verify their email.

Our application supports the Logout function, users can log out their account anytime. Once the users actively Log out the account, our application will not save their login information for fast launch.

Our application adopts the authentication system provided by Firebase. To implement the Login/Sign Up system, we import the Firebase Authentication API Libraries to our flutter project.

2.4.3 Profile Display and Editing

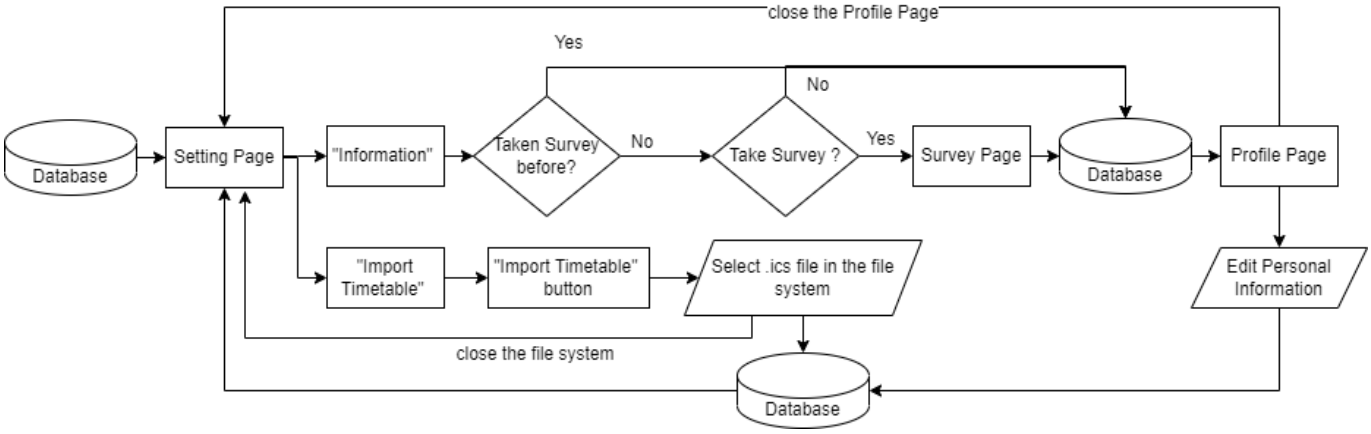


Figure 19. The system flow of building up the user's profile

Figure 19 shows the process of building up the user's profile. Users can click the "Profile" on the Setting Page. The system will then check the status of the users. If they did not take the survey before and are willing to take it, they would be redirected to the Survey Page to indicate their interests. The answer will then be stored in the database. Whether the user takes the survey or not, the system will grant the user data and the suggestion list of hashtags from the database and redirect the user to the Profile Page.

On the Profile Page, users can further enter their personal information such as their username, major or profile picture. After users click the "save" button, the changes in the Profile Page will be stored in the user collection in the database. Meanwhile, the major, course and hashtag will be stored in the autocomplete collection in order to enrich the suggestion lists for searching in Match Page and inputting the hashtag in the Profile Page. And users will be redirected to the Setting Page.

In order to import timetables, users can click the "Import Timetable" on the Setting Page. Users can then click the "Import Timetable" button to open the file system and select the .ics file. The system obtains the course information, namely course code, year of study and section, and stores these data in the user collection. The course information will be used for the Weighted Euclidean Distance Algorithm in Section 2.4.4.3. Meanwhile, course code will be updated to the autocomplete collection for the searching function in Match Page.

2.4.4 Recommendation System

2.4.4.1 The Design and the Flow of the Recommendation System

The main feature that requires implementation in the recommendation system is automatically providing suggested matching results in the Match Function. We implemented two algorithms in this recommendation system. The Weighted Euclidean Distance Algorithm is for users to find a study partner by courses, which are extracted by Import Timetable Function. The result will be shown under “Match by Courses” in the Match Page. The Cosine Similarity Algorithm is for users to find friends by hashtags, which are generated by AI models. The result will be shown under “Match by Interest” in the Match Page. Their relationship is shown below in Figure 20.

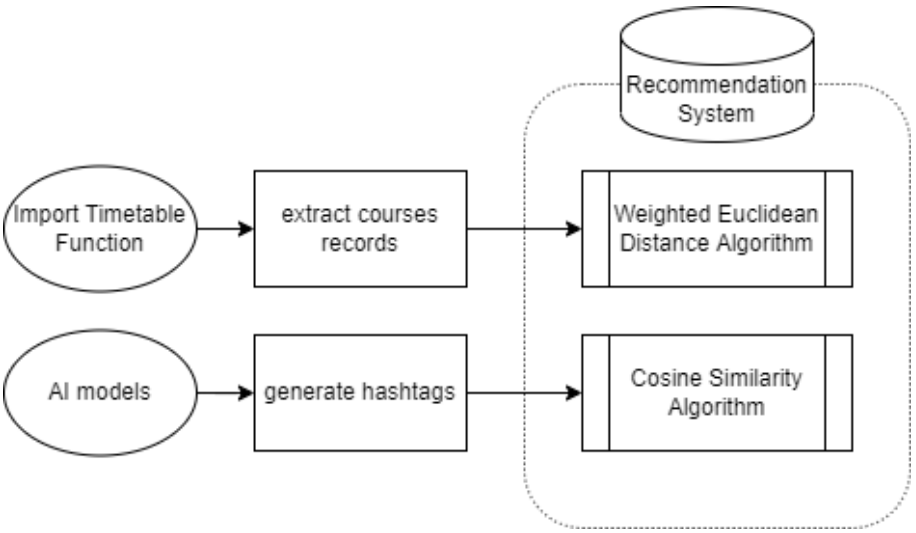


Figure 20. The relationship between Import Timetable function, AI models, and the recommendation system

The algorithms are written in Python. They are stored on the Google Cloud Platform, and they can be run by the App Engine on this platform. We considered that the redundancy of the matching process and multiple users simultaneously running the matching process might lead to the server crash. We have two responding methods to avoid these problems. Firstly, the recommendation system will store the matching results in the database for the mobile app to read instead of giving out the results directly to the app. Secondly, the algorithm will be triggered only when the user successfully logs into the mobile app or edits his or her profile. As a result, if a user

goes to the Match Page and Chat Page repeatedly, no pressure will be given to the system since there is no request sent to the recommendation system.

2.4.4.2 AI Model

Most social media application users will upload their profile pictures, and the profile pictures could be important to analyze the similarity of users. Since the like-minded people should have a similar sense of art as well as the shooting style of their profile picture. Therefore our recommendation system analyzes the profile pictures provided by the users. In this project, the application analyzes the shooting style of the profile.

Similarly, most social media application users will give a brief description to themselves. We believe that users write their description to give a brief statement about themselves. Therefore,we want to make use of the description to analyze the users. In this project, we implemented an NLP model for extracting hashtags from description. To implement these features, we built the following models:

2.4.4.2.1 Model for Analyzing the Style of the Profile Pictures

We want to build a model to analyze the shooting style of the profile pictures. In general, a profile picture could be classified as a full-body photo, half-body photo, big-face photo or no face photo. Moreover, the number of people in the profile pic will be analyzed.

To implement, we adopt an existing convolutional neural networks (CNN) model [12] for face recognition. The CNN model can return the points (A, B, C, D) of 4 corners of all faces in the photo.

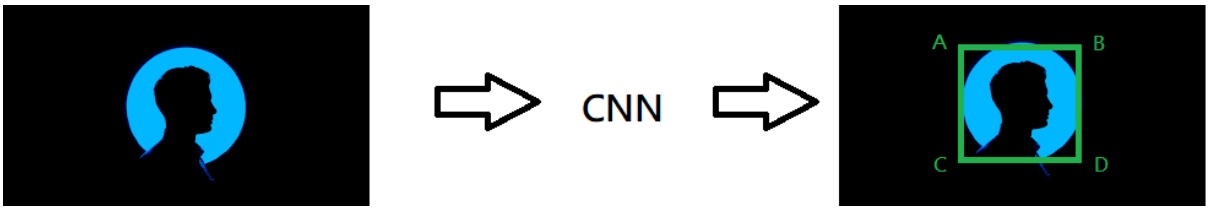


Figure 21. Return values of our CNN model

After finding the coordinates of all corners of all faces, our model will calculate the number of faces and the scale of the face.

We convert the number of faces into hashtags as demonstrated in Table 13.

No. of faces	Represent Categories	1st Return Hashtag
0	No Face photo	#NoFacePhoto
1	One person photo	#OnePersonPhoto
2	Two people photo	#TwoPeoplePhoto
more than 2	Group of people photo	#GroupPeoplePhoto

Table 13. Table of conversion from number of faces to hashtags

Calculation of scale:

$$Scale = \frac{(A - C) \cdot (B - A)}{ImageHeight \cdot ImageWidth}$$

After computing the scale, our model will bin the scale into 4 categories as shown in Table 14.

Face Scale Percentage	Represent Categories	2nd Return Hashtag
0%	No Face photo	#NoFacePhoto
0% < scale <= 2%	Full Body photo	#FullBodyPhoto
2% < scale <= 7%	Half body photo	#HalfBodyPhoto
7% < scale	Big Face photo	#BigFacePhoto

Table 14. Table of binning the scales

Therefore, the model got the 1st hashtag and the 2nd hashtag value. The model combines 2 hashtags into a feature vector. The feature vector will be used to compute the similarity in our recommendation algorithm in Section 2.4.4.3.

2.4.4.2.2 Model for Extracting Hashtags from the Description

We want to build a model to extract the hashtags from the description. We believe that the description of the users may contain the keywords of their hobbies, thoughts or personal information. These keywords are then used as hashtags to represent the interests of the users.

To implement, we adopt an existing python library - RAKE [13] for hashtags extraction. This packet will return a list of possible hashtags based on the Rapid Automatic Keyword Extraction (RAKE) algorithm. This algorithm makes use of stop words to build up a list of candidate keywords. It then develops a co-occurrence graph and calculates the word score based on different metrics. In this project, we use the ratio of the word degree to word frequency to generate the results. Word degree depends on the number of combinations the word has with other candidate keywords while word frequency counts the number of times that the word exists in the text. If more than 1 candidate word appears together in the text, they will become a key phase and the score of each phase will be the sum of the individual score of words. For each keyword extraction, we store the top 5 hashtags in the database.

The RAKE algorithm considers the frequency of word appearance and the number of co-occurrence with other words in a text when analyzing the text. However, users usually write one to three sentences with limited words in the description. There is a high chance that every candidate keyword appears once in the description. Hence, the system combines all previous descriptions and the new description together to form a long paragraph and extract hashtags from it.

Below is an example of the RAKE algorithm. User A has inputted 2 descriptions: "Want to find friends who love basketball" and "Playing basketball every Friday". The system will first combine 2 descriptions into a long paragraph with a ".". Then, based on the stop-word list [14], the paragraph will be separated into

["find", "friends", "love", "basketball", "Playing", "Friday"]

These candidate keywords will be used to build a co-occurrence graph as shown in Table 15.

	find	friends	love	basketball	Playing	Friday
find	1	1	0	0	0	0
friends	1	1	0	0	0	0
love	0	0	1	0	1	0
basketball	0	0	1	2	1	0
Playing	0	0	0	1	1	0
Friday	0	0	0	0	0	1

Table 15. The example co-occurrence graph of user A

In the co-occurrence graph, each row means the number of times the word appear with other content words in the text. The word degree of each word is equal to the sum of each row. And we can obtain the word frequency from the value of the cell where row name and column name are the same. The score of each word will be displayed in Table 16.

	word degree	word frequency	word degree / word frequency
find	2	1	2
friends	2	1	2
love	2	1	2
basketball	4	2	2
Playing	2	1	2
Friday	1	1	1

Table 16. The score of each word calculated by the RAKE Algorithm

In this example, there are 3 key phases, namely “find friends”, “love basketball” and “Playing basketball”, and the score of all combinations is 4. Therefore, the keyword list generated is [“find friends”, “love basketball”, ”Playing basketball”, ”Friday”].

For each keyword extraction, the top 5 hashtags generated will be stored in the tag field in user collection, which will be used to compute the similarity in our recommendation algorithm in Section 2.4.4.3.

2.4.4.3 Weighted Euclidean Distance Algorithm

The first algorithm used in the recommendation system is to match by courses that users have taken with Euclidean distance. The minimum distance of the users will determine the results. The recommendation system gives the most matching result $User_r$ based on the taken courses for a user, $User_t$ with the following formula:

$$User_r = argmax_{i \neq t} || C_t - C_i ||^2$$

$$where\ C_t\ and\ C_i\ are\ the\ course\ list\ vectors\ of\ user_t\ and\ user_i,$$
$$C_t, C_i \in \mathbb{R}^M,\ i \in \{1, \dots N\},$$
$$M\ is\ the\ number\ of\ courses,\ and\ N\ is\ the\ number\ of\ users.$$

$$For\ each\ entry\ k\ in\ vector\ C_i,\ C_i[k] = 4 - t_{ik}/365$$
$$where\ C_i[k]\ is\ the\ score\ for\ course_k,$$
$$t_{ik}\ is\ the\ time\ (unit\ in\ day)\ since\ user_i\ toke\ course_k.$$

Firstly, the recommendation system gets all course records and users' course history from Firebase and creates a course array. For each element in the array representing one course, if the user has not taken this course, the element will be marked as integer 0. Otherwise, it will be marked as integer 1 to 4 depending on the time interval. Weighting 4 means most recently, and weighting 1 means most previously. We give weighting to the elements $C_i[k_1]$ to distinguish two users if they had taken the same course but with different time intervals. After that, the system will compare the course array of the target user with all other users by the euclidean distance measure of vectors. Below is an example of the first algorithm, the course array of all users is shown in Table 17 and the results of this algorithm are shown in Table 18.

	Course_A	Course_B	Course_C	Course_D
User_A	4	3	2	1
User_B	3	2	1	0
User_C	2	1	0	0

Table 17. The example course array of users

	User_A	User_B	User_C
User_A	-	2.000	3.606
User_B	2.000	-	1.732
User_C	3.606	1.732	-

Table 18. The example results of the Weighted Euclidean Distance Algorithm

In the course array table, each row means an array of a user. Each column means a course that users have or have not taken. Weighting 4 means that the user is taking this course now, weighting 3 means that the user took this course last year, and so on. For the results in this example, in User_A's view, the system will recommend User_B to him or her since User_B has a shorter distance (2.000) than User_C (3.606). Besides, in User_B's view, User_C has a shorter distance (1.732) than User_A (2.000), so the system will recommend User_C to him or her.

2.4.4.4 Cosine Similarity Algorithm

The second algorithm used in the recommendation system is to match by hashtags of users with cosine similarity. The maximum similarity of the users will determine the results. The recommendation system gives the most similar $User_r$ based on hashtags for a user, $User_t$ with the following formula:

$$User_r = argmax_{j \neq t} \frac{H_t \cdot H_j}{||H_t|| \cdot ||H_j||}$$

$$where\ H_t\ and\ H_j\ are\ hashtag\ vectors\ of\ user_t\ and\ user_j\ ,$$
$$H_t\ , H_j \in \mathbb{R}^V,\ j \in \{1,\ \dots\ N\},$$
$$V\ is\ the\ number\ of\ hashtags,\ N\ is\ the\ number\ of\ users,$$
$$||H_t||\ and\ ||H_j||\ are\ the\ L2-norm\ of\ vector\ H_t\ and\ H_j\ .$$

$$For\ each\ entry\ in\ vector\ H_j,\ H_j[l]\ \in\ \{0, 1\}$$
$$where\ H_j[l]\ represents\ the\ existence\ of\ hashtag\ l\ of\ user_j\ ,$$
$$0\ is\ representing\ False,\ and\ 1\ is\ representing\ True.$$

Firstly, the recommendation system gets all hashtags and users' hashtags from Firebase and creates a hashtags array. For each element in the array representing one hashtag, if the user has this hashtag, the element will be marked as integer 1. Else, it will be marked as integer 0. After that, the system will compare the hashtags array of the target user with all other users by the cosine similarity measure of vectors. Below is an example of this algorithm, the hashtags array of users are shown in Table 19 and the results by this algorithm are shown in Table 20.

	Hashtag_A	Hashtag_B	Hashtag_C	Hashtag_C
User_A	1	1	1	0
User_B	1	0	1	0
User_C	0	1	1	1

Table 19. The example hashtag arrays of users

	User_A	User_B	User_C
User_A	-	81.65%	66.67%
User_B	81.65%	-	40.82%
User_C	66.67%	40.82%	-

Table 20. The example results by Cosine Similarity Algorithm

In the hashtags array table, each row means an array of a user. Each column means a hashtag that users have or not. For the results in this example, in User_A's view, the system will recommend User_B to him or her since User_B has a higher similarity (81.65%) than User_C (66.67%). Besides, in User_C's view, User_A has a higher similarity (66.67%) than User_B (40.82%), so the system will recommend User_A to him or her.

2.4.4.5 The Differences Between Two Algorithms

The recommendation system gives the most matching user for the first algorithm by picking the vector with the shortest Euclidean distance. And for the second algorithm, the recommendation system provides the most matching user by the maximum similarity. A rough differentiation for these two algorithm methods is that the results of the first algorithm are normalized due to the giving weight in it, but the results of the second algorithm are not normalized. Also, Euclidean distance measurement is sensitive to absolute values, so it is suitable for those data with objective facts like what courses the users took and which year they took. But the hashtags measured in the Cosine Similarity Algorithm are more subjective, and they can be added or removed by users casually. Hence, the second algorithm is more suitable for hashtags, which is not sensitive to absolute value as the first algorithm.

2.4.5 Chat Function

2.4.5.1 Create Chat Room of Two Users

Figure 22. Flowchart of creating a chat room of two users

When the user joins an event or makes a friend, our application will automatically create a chat room for the users. To create a chat room of two users, our application will get the emails of both users and combine them as the chat room ID. The ID format will be “Auser@connect.ust.hk_Buser@connect.ust.hk” or “Buser@connect.ust.hk_Auser@connect.ust.hk”. Then, our application will create a document with the chat room ID under the chatRoom collection in the database. After creating the document in the database, our application will redirect the user to the Chat Page. Figure 22 demonstrates the flowchart of creating a chat room for two users.

2.4.5.2 Display Recent Chat on the Chat Page

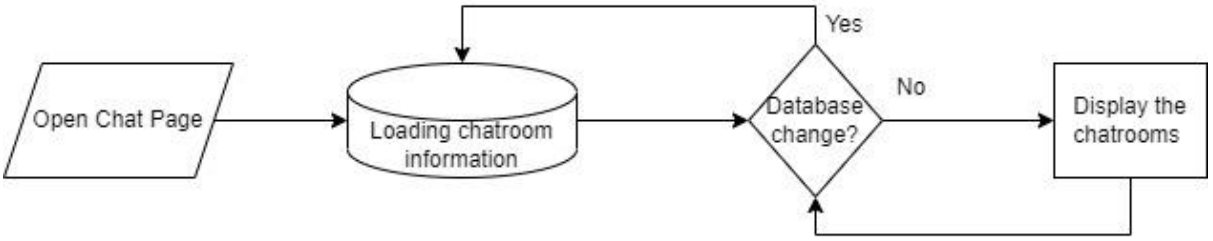


Figure 23. Flowchart of displaying Recent Chat on the Chat Page

Unlike traditional database I/O, displaying recent chat functions need to be in real time. To implement this feature, we used the stream builder function in Flutter. In the stream builder, our application will get the information from the database when the database has changed. It means that our application always gets the latest information from the database. In the flow of displaying recent chat, our application will read the data from the database when the user enters the Chat Page. After that, the latest version of the chat room list is displayed. Our application will keep

observing the change of database. Once the database has changed, our application will get new data from the database to ensure all information is the latest. Figure 23 demonstrates the flowchart of displaying Recent Chat on the Chat Page.

2.4.5.3 Chat in Real Time

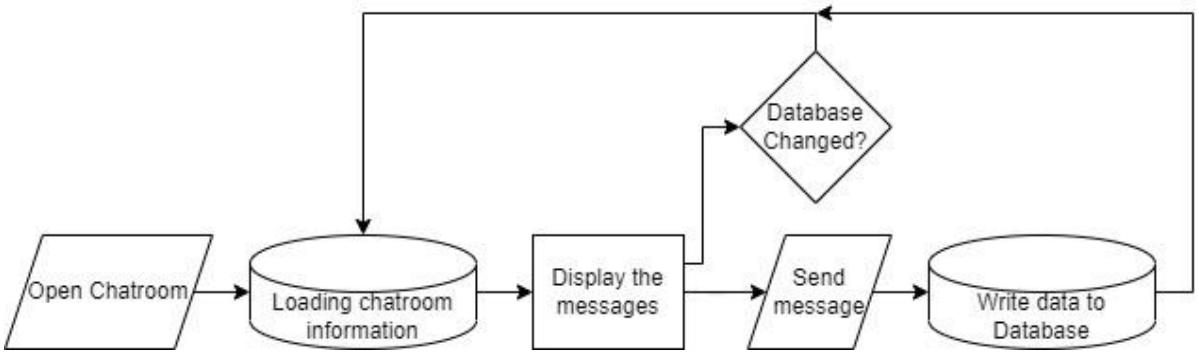


Figure 24. Flowchart of chat in real time

When users open the chat room, the application will first download the messages from the database and display the message on the chat room screen. Then, our application will keep observing the change in the database. Once the database is changed (received a message from friends), our application will download the chat messages again and display them. On the other hand, if the user sends a message to their friend. The application will write data to the database. This action will update the database data and trigger the application to reload the chat room messages again. Therefore, users always see the latest message and chat in real time. Figure 24 demonstrates the flowchart of chat in real time.

2.4.5.4 Search Nearby Users by GPS

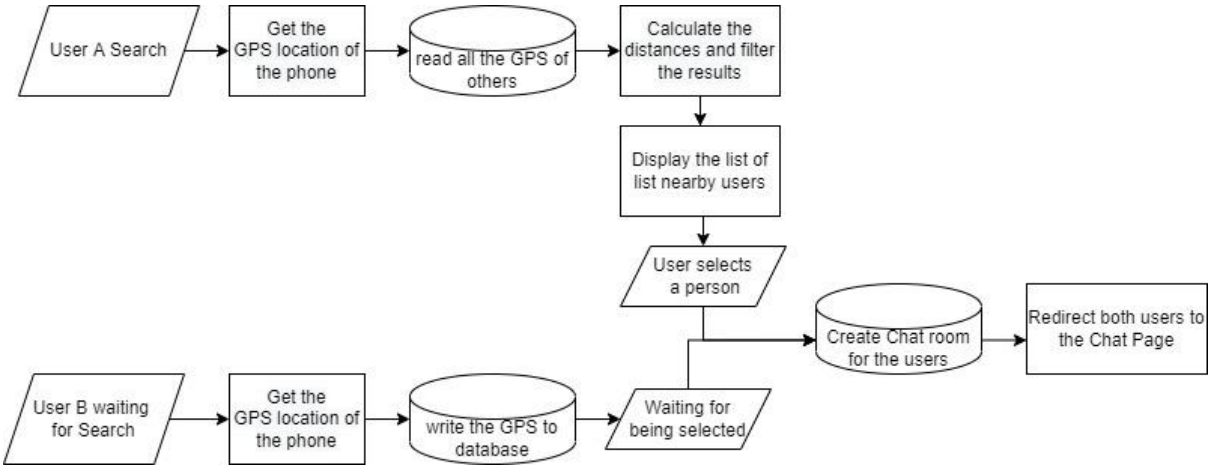


Figure 25. Flowchart of Search Nearby Users by GPS

When a user A is waiting to be searched, our application will get the GPS location of the phone and upload it to our database. At this moment, user A is in a waiting state, which will be interrupted by other users. Meanwhile, another user B is supposed to stay next to user A and search nearby friends. When user B searches the people around, our application will get the GPS location of the phone and get all the GPS location of others in the database. After that, our application will calculate the distance between this user and all other users. Furthermore, the closest users will be displayed. At this moment, user B can send friend requests to user A. After a consensus, our application will create a chat room for users A and B and redirect them to the Chat Page. As a result, user A and B are connected. Figure 25 demonstrates the flowchart of Search nearby users by GPS.

2.4.6 Match Function

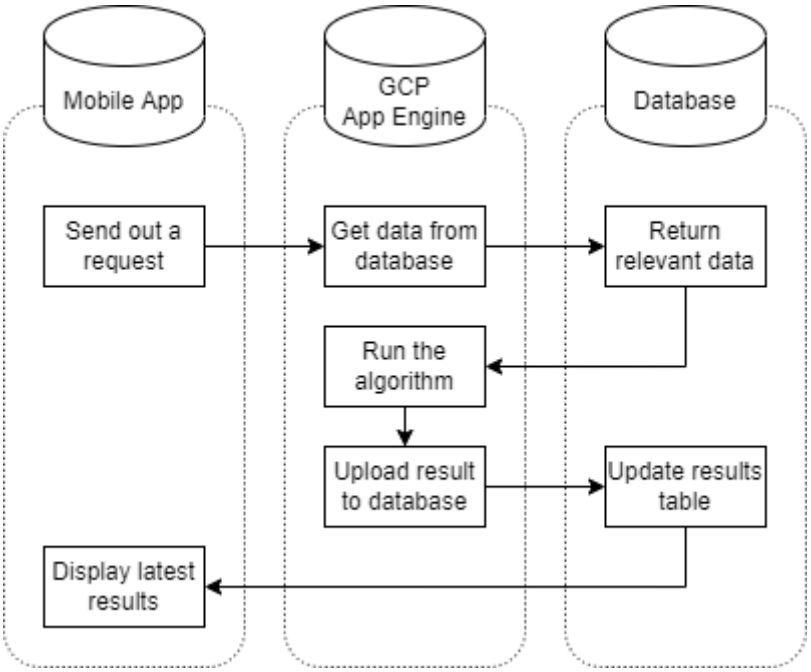


Figure 26. The system flow of the Match Function

Figure 26 shows the system flow of the Match Function. When the user successfully logs into the app or edits his or her profile, it will trigger the app to send a request to the Google Cloud Platform. The App Engine on the Google Cloud Platform will get relevant data from the Google Firebase database and then run the algorithms. After that, the process will update two result tables to the database on Firebase. Finally, the Match Function on the Match Page will display the latest results from the result tables on the database, which are “Match By Courses” and “Match By Interests”.

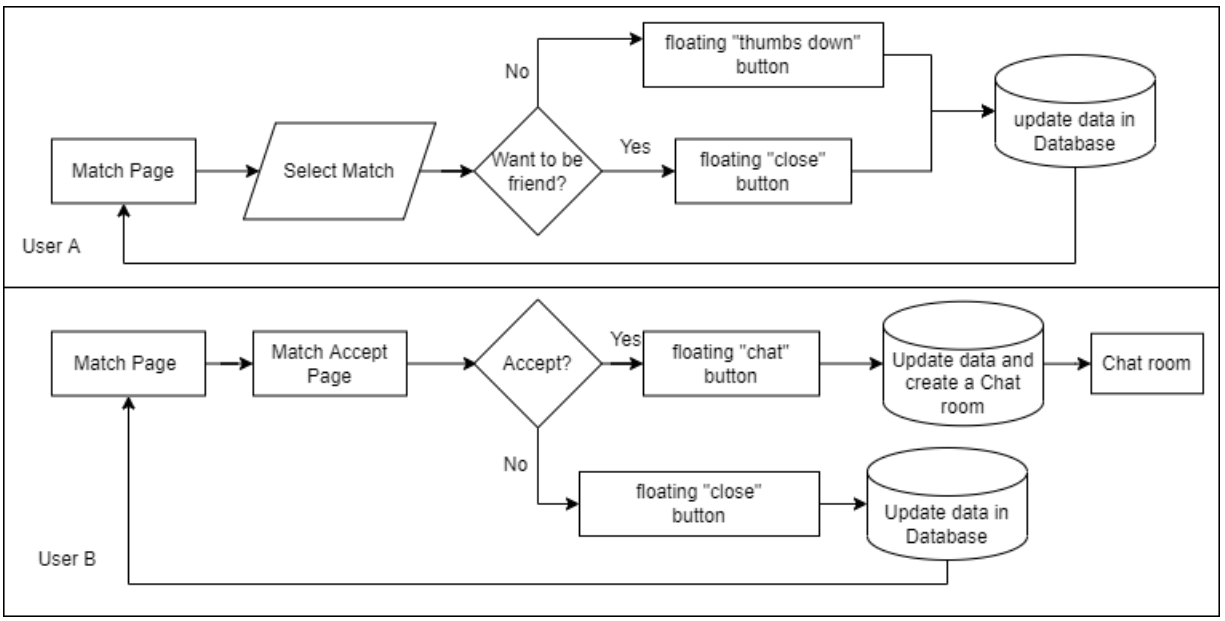


Figure 27. The system flow of matching a friend

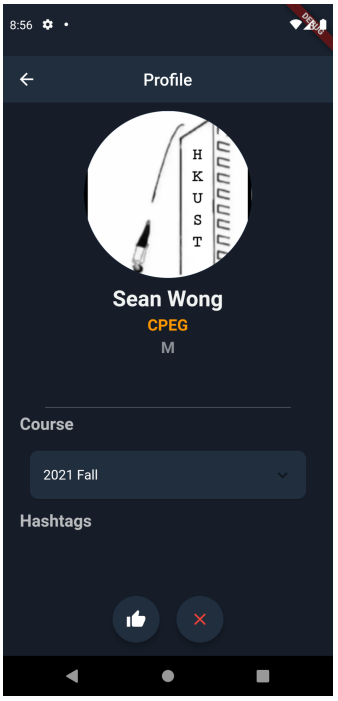


Figure 28. The screenshot of the Match Detail Page of User “Sean Wong”

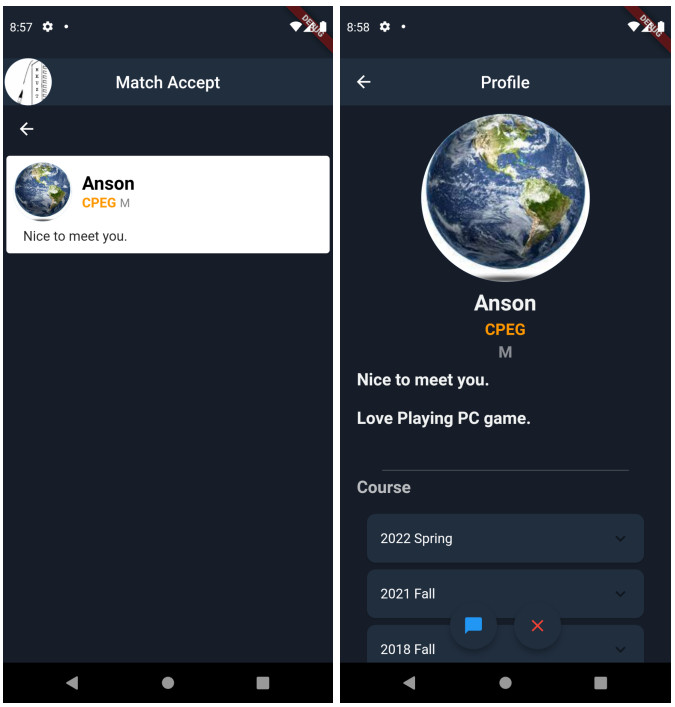


Figure 29. The screenshots of the Match Accept Page and the Match Detail Page after clicking a user on the Match Accept Page

Figure 27 shows the flow of matching a friend. On the Match Page, two lists of matched users generated by the recommendation system in Section 2.4.4 will be displayed to users. Users can click the target users to view their details on the Match Detail Page as shown in Figure 28. If users are uninterested in the match, they can press the floating “close” button. The uid of the uninterested user will be removed in the algoMatchResult collection and added to the declined array in both documents in user collection. Once users get interested, they can press the floating “thumbs up” button and wait for approval. The system will remove the uid of the interested user in the algoMatchResult collection and update it to the pending array in the user’s document.

Users can go to the Match Accept Page as shown in Figure 29 to view the friend requests from other users. To accept or decline requests, users can click the particular user to view their details on the Match Detail Page. If users are uninterested in the match, they can press the floating “close” button. The uid of the uninterested user will be added to the declined array in both documents in user collection. If users want to accept the friend request, they can press the floating

“chat” button. The system will update the friends array in the document, create a chat room, and redirect the users to the chat room.

2.4.7 Event Function

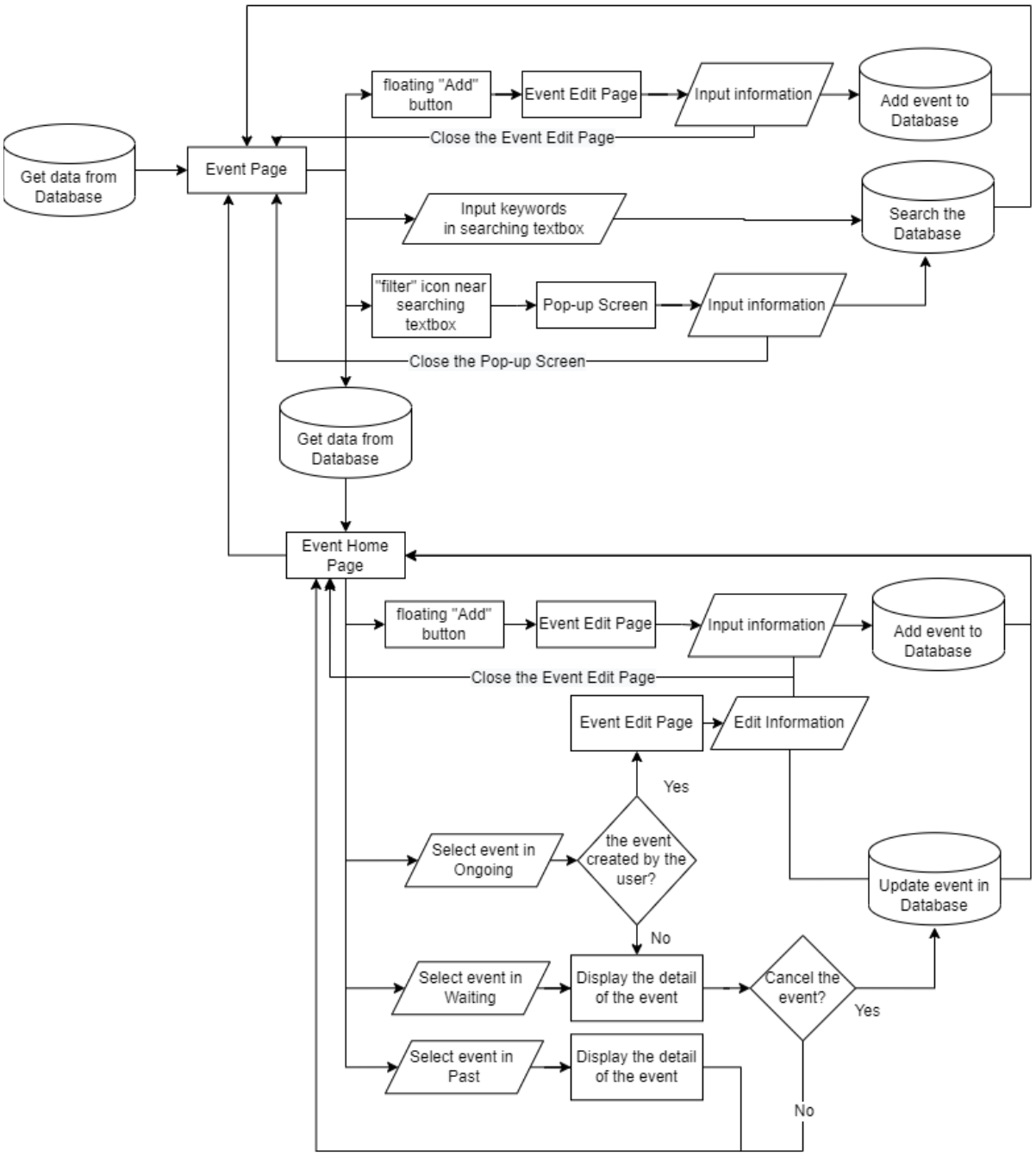


Figure 30. Flowchart of Event Function

Figure 30 describes the workflow of the Event Function. When the user clicks into the Event Page, all the event data sorted by the start day in ascending order firstly fetched from the Firebase Cloud Firestore. Different functions in the Event Page will be introduced below.

2.4.7.1 Search Events

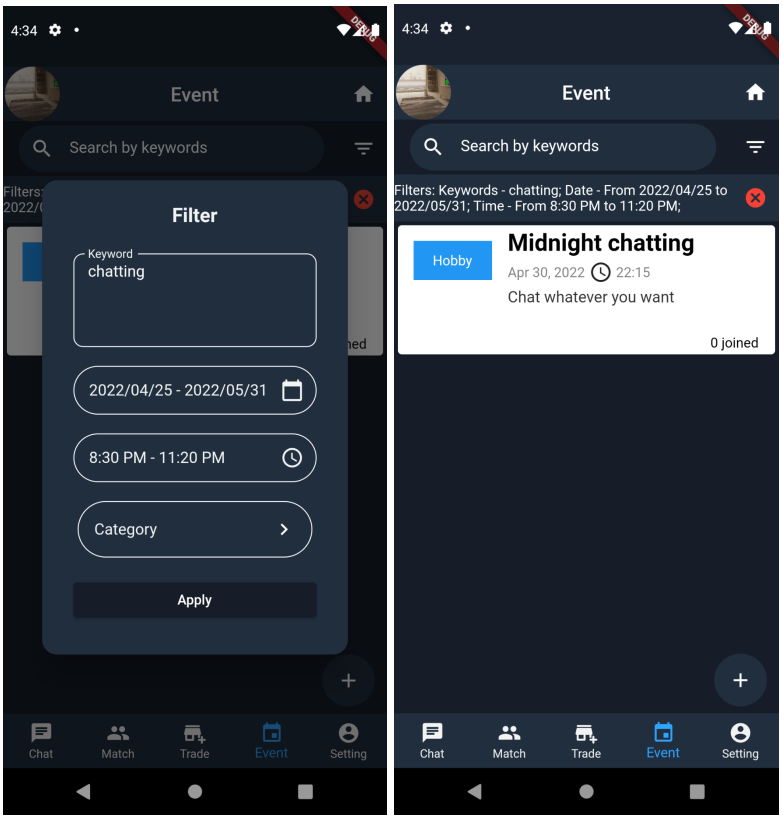


Figure 31. The screenshot of the search function and the Event Page displaying the searching output

Users can directly type keywords in the textbox on the top of the Event Page to search for specific events. Furthermore, users can click the “filter” button near the textbox to open the pop-up screen with more filters as shown in Figure 31. Users can input their criteria and press the “Apply” button to request for searching. Once the system receives the result, it redirects the users to the Event Page and displays the selected events. Users can view the current applied filter under the search bar. Once the user wants to cancel the filter, they can press the close icon. The system then obtains data from the database with no query.

2.4.7.2 Create Events

In order to create a new event, users can add an event by pressing the floating “add” button at the bottom-right corner of the Event Page. The system will redirect the users to the Event Edit Page. Users have filled in critical data, namely name, date, start time, and end time. After filling all this necessary information, users can either continue to input description, location, and category to provide more information for the search function, or press the “Create an Event” button to add the event to the database.

2.4.7.3 Organize Events

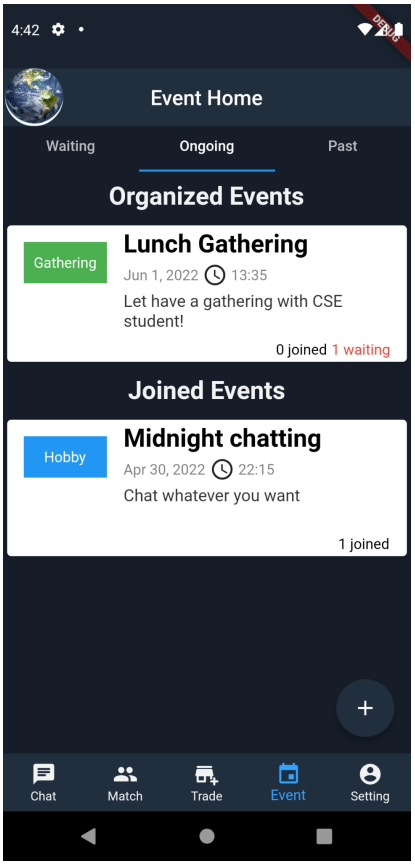


Figure 32. The screenshot of Event Home Page

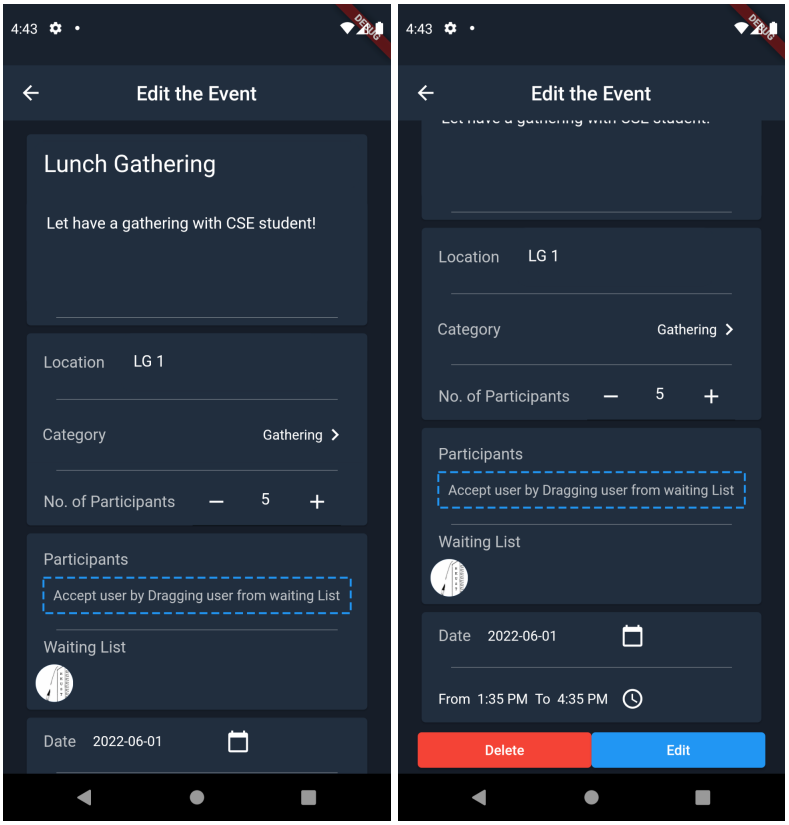


Figure 33. The screenshots of the Event Edit Page when editing an existing event

To manage the user's event, the user can click the "home" icon button on the Event Page to go to the Event Home Page as shown in Figure 32. The Event Home Page is split into three states based on different conditions as shown in Table 21.

State	**Condition**
Waiting	1. User is on the waiting list. 2. The start day of the event is after today.
Ongoing	1. User is the organizer or 2. User is on the participant list. 3. The start day of the event is after today.
Past	1. User is the organizer or 2. User is on the participant list. 3. The start day of the event is before today.

Table 21. Three states in Event Home Page based on different conditions

Users can view the details of their joined event or manage their event by clicking that event on the Event Home Page. As participants, after the users click the event in the Waiting and Ongoing state, they can view the details of the event. If they do not want to join the event anymore, they can press the floating “Cancel” button to cancel the event. If the event was created by the user, The system will request the event data and redirect the user to the Event Edit Page as shown in Figure 33. When the user presses the “Edit” button, the changes will be uploaded to the database. The event will be removed from the database if the user presses the “Delete” button. If the users click the event in the Past state, they can only view the details of the event.

2.4.7.4 Flow of Joining an Event

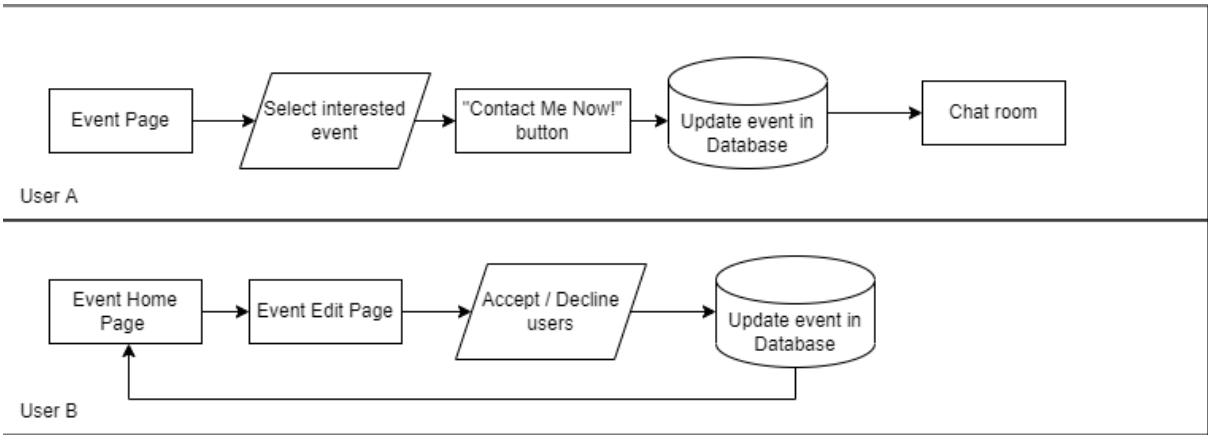


Figure 34. Flowchart of joining an event

Figure 34 shows the flow of joining an event. On the Event Page, a list of events will be displayed to users. Users can click the desired event to view their details. Once users get interested, they can press the “Contact me Now!” at the bottom of the page. The system will update the waiting list in the event’s document and create a chat room for the user to chat with the organizer. If the organizer wants to accept the participation requests, he can go to the Event Edit Page of that event as shown in Figure 35 to accept the users by dragging the user’s icon under the “Waiting List” to the “Participants". If the organizer regrets to invite a participant, he can drag that user’s icon under the “Participants" back to the “Waiting List” to remove his participation. If the organizer wants to reject the users, he can double click the users icon in the “Waiting List” to remove the user from the event. The organizer should press the “Edit” button to update the changes to the database.

2.4.8 Trade Function

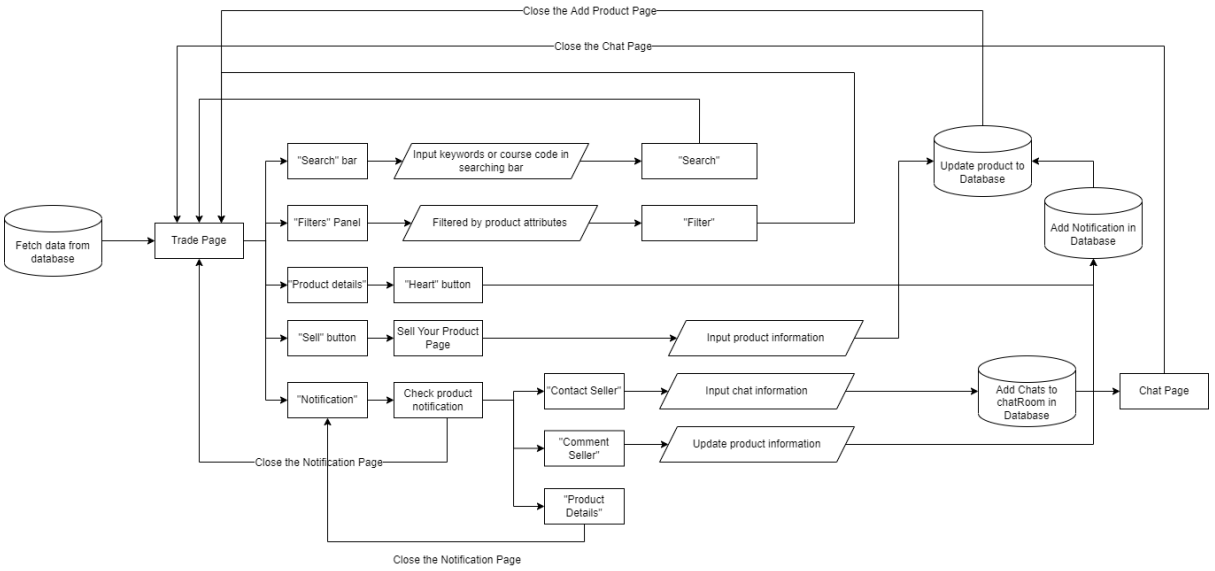


Figure 35. Flowchart of Trade Page

Figure 35 describes the overall workflow of the Trade Function. When the user clicks into the Trade Page, all the product data is firstly fetched from the Firebase Cloud Firestore and first 10 of them will be shown in chronicle order based on the attributes updateDate and isVisible of each product in the database. There will be a “load more” button for users to see more products without fetching data from the database.

2.4.8.1 Search Products

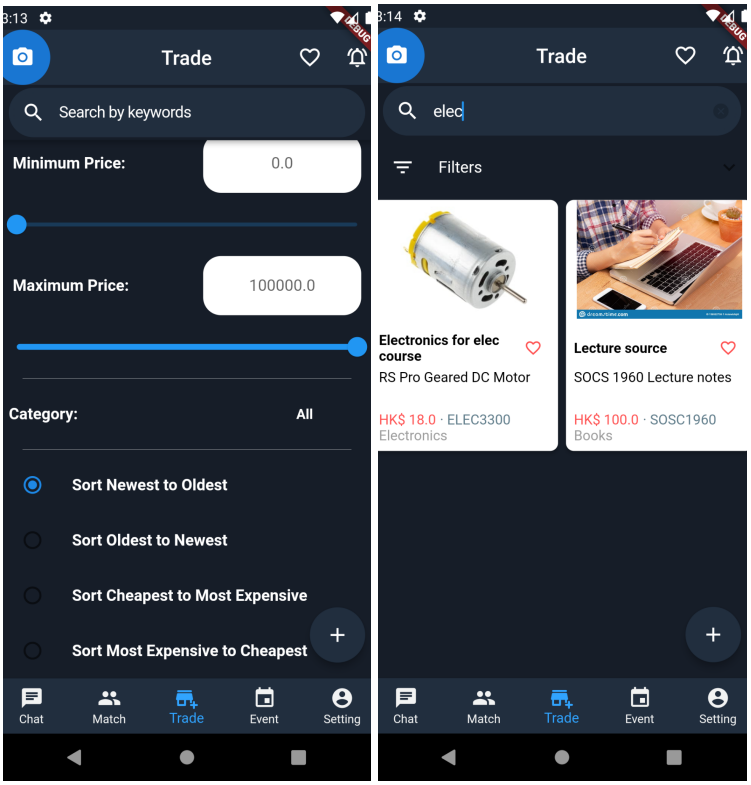


Figure 36. The screenshot of the Filter Function and the search result

Using the plugin Fuzzy, the search function is provided for users to find products by relevance. Performance of relevance is the result of distance. An exact letter match which is 'distance' characters away from the fuzzy location would score as a complete mismatch. A distance of '0' requires the match be at the exact location specified, a threshold of '1000' would require a perfect match to be within 800 characters of the fuzzy location to be found using a 0.8 threshold. Furthermore, users can click the “filter” button near the textbox to open the drop down menu with more filters. In Figure 36, If the word "elec" is entered, popular products such as "electronics", "lecture" and "ELEC course" are returned. Users can input their range or select the criteria of filters, the results will be displayed while searching.

2.4.8.2 Add Products

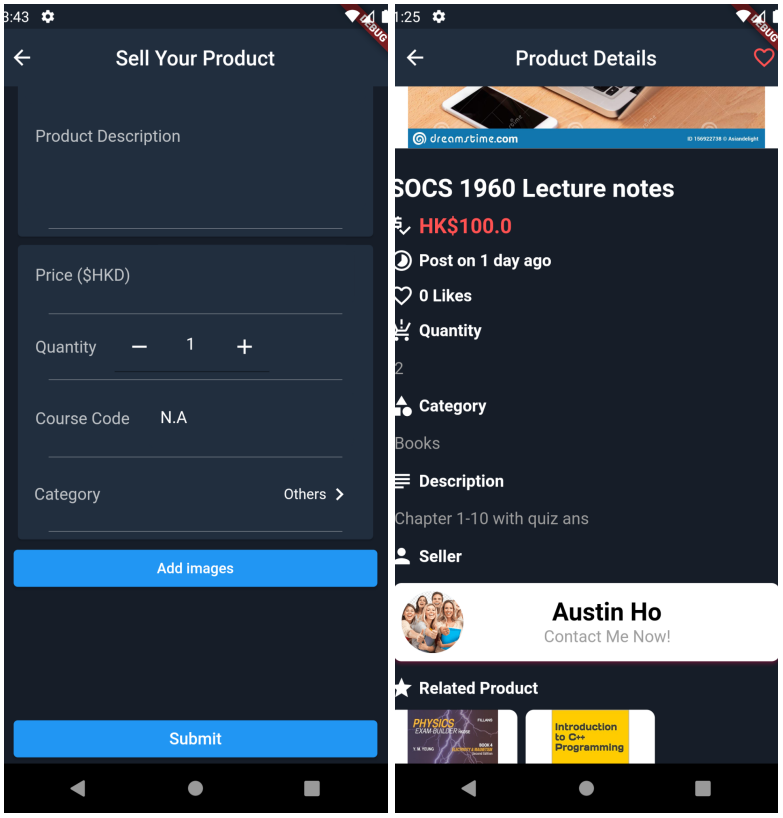


Figure 37. The screenshot of the Sell Your Product Page and the Product Details Page

According to Figure 37, all the text controllers of the product such as post name, product name, product description, product image, product price, product quantity, and product category needed to be filled in before it was submitted. Otherwise, there will be a warning coming out to alert the user. After it is submitted, the system will create a document with the product ID under the Products collection in the database and store all the data from the user input followed by a user ID. The user needed to pull down the screen to refresh the trading system by fetching data from the database again. Opening the Product Details Page in Figure 37, apart from the fundamental information, "Likes" is also provided based on the number of followers and it will determine how hit the product is. The post date is calculated as the time difference between the current time and the most updated date of the product inside the database. There are comments of the seller fetched from all the products he sold to let the consumer grasp a look at the seller. Moreover, the related product provides the suggested products that are the same as the current state by comparing the category field of different products.

2.4.8.3 Edit Products

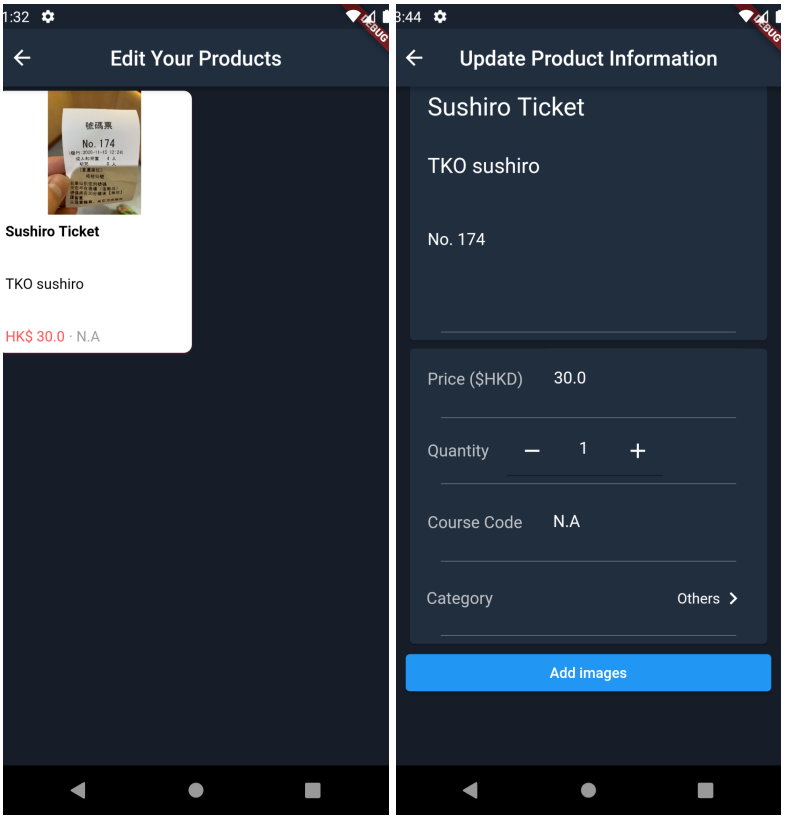


Figure 38. Screenshots of the Edit Your Product Page and Update Product Page

Product Condition	**Attribute State**

Add	Visibile = True;
	isSold = False;
	isDeleted = False;

Hide	Visibile = False;
	isSold = False;
	isDeleted = False;

Sold	Visibile = False;
	isSold = True;
	isDeleted = True;

Deleted	Visibile = False;
	isSold = False;
	isDeleted = True;

Table 22. Product's attribute states based on different conditions

If it is the user's own product, the user can click the “Manage Your Product” button on the Setting Page to update product information according to Figure 38. It displays all your own products, the targeted product information will be loaded when you click the preview. editing, it will update the corresponding attributes of the product in the database. Apart from the product information, users can also adjust the condition of the product to "Add", "Hide", “Sold”, or "Deleted" according to Table 22. Whether it was selected, it returns the value to the database and it will refresh the products on the Trade Page.

2.4.8.4 Notification

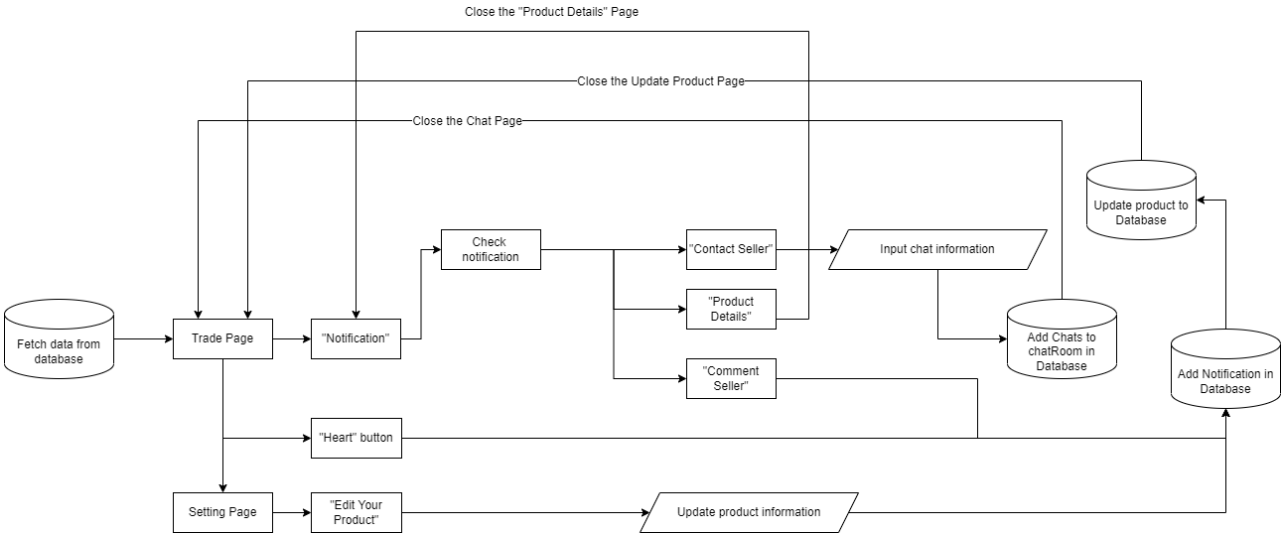


Figure 39. Flowchart of Notification Page

The screenshots of the Notification Page is shown in Figure 39. As shown in Figure 40, Users can become a follower of the products by clicking the “heart” button. Once the button is clicked, the user ID of the followers will be updated to the database collection of the Products and the system will add the user ID to the Followers Field. Once the liked product is updated, sold or deleted, it will make a request to the system and send a notification to users. Every notification record is saved in the database followed by their own user ID. If the notification has been seen by the users, it will return a true value of the attribute, isRead to the database immediately shows that it was readed and the icon will change to notify users whether it was read. Once the favourite was sold, the seller needs to input the name of the buyers and send the comment request through notification.

2.5 Testing

Several fundamental parts of the app have to be tested to ensure the functionalities, usabilities, and consistencies.

2.5.1 Test for the UI design

For UI testing, several components need to be tested. For example, the screen resolution, screen size, UI elements, multi-touch, long/short press, etc. This could be done by automation or manual testing. We test it in both ways. We first test it manually and then adopt Selendroid [15] for automatic testing. Selendroid is a mobile app UI testing tool. It could be used for Android and IOS apps.

2.5.2 Test for the Multi-platform Compatibility

We decided to make our app applicable to both IOS and Android devices. However, there would be conflicts between these two operating systems. Therefore, we need a different test strategy to test the consistency of our app. The consistencies will be tested in 2 phases. In the first phase, we found some keys to keep alert to the issues caused by hybrid (IOS and Android) app elements. In the second phase, we conducted automation to test the compatibility of our app.

2.5.3 Test for the Recommendation System

The recommendation system needs to be tested in terms of its accuracy, robustness, and scalability. Our recommendation system is tested under two conditions. The first condition is to test whether the app will keep recommending the same set of people if the users have declined them. The second condition is to test if the app will randomly recommend matching when no filtering result was found.

2.6 Evaluation

In the user acceptance tests, we obtained interviews with university students. In the interviews, we showed the interviewees all the features and functions of our application. For some interviewees, we also helped them to create their own account for testing. After that, we gave them a Google Form survey to collect their feedback.

App Target Users

In the current phase, the app is designed for HKUST students only. Hence, our target user group is focused on HKUST students.

Survey Overview

As mentioned previously, the purpose of our application is to provide an integrated app for students to develop a friendship with others having similar interests on campus. The survey invited 13 students to test our application. We use the survey to check whether the users feel the app followed the objectives we mentioned and collect feedback for further improvements. The list of the survey questions is shown in Appendix B.

Survey Summary

In general, the overall user feedback to the app performance is positive as shown in Figure 41. Most of the users reported that they are satisfied with the functions we have developed. In addition, most of the users reported that our application is useful to facilitate social activities and communications with other students as shown in Figure 42.

Figure 41. Ratings about the functionalities in our application

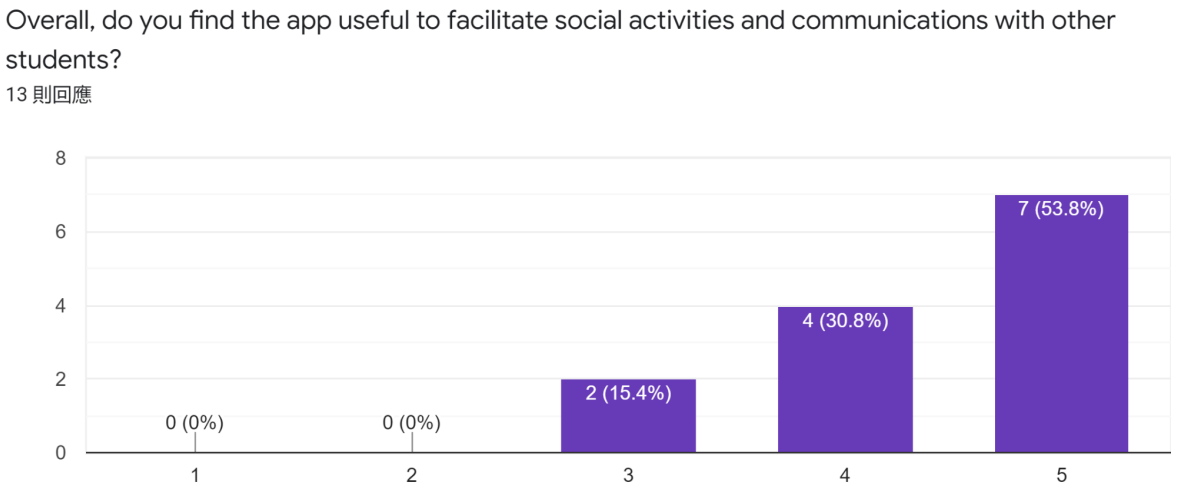


Figure 42. Rating about the usefulness of our application

Questions about whether the user has encountered any problems or suggestions were asked. The main feedback from users is to provide a guide for using our application. Some users reported that they do not know how to use our application. In addition, some users reported that the UI such as searching bars look different in different pages. We should make them consistent for pages. For some other useful

suggestions, our team decided to place it into future works. Moreover, most of the users think the recommendation results are useful to them with no negative response as shown in Figure 43.

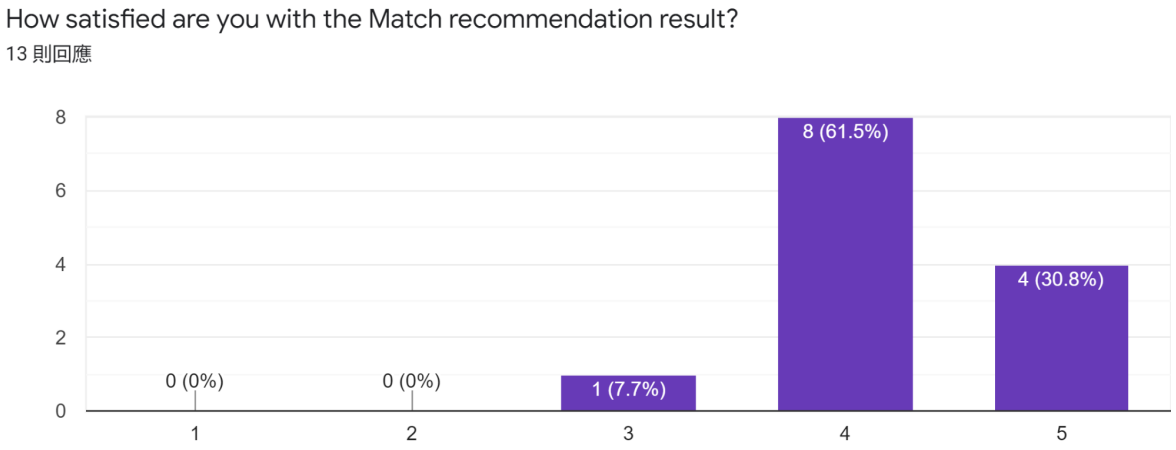


Figure 43. Ratings about the recommendation results

Through the user acceptance testing, we have explored the problems of our application and we have continuously improved our app based on their feedback to provide a better user experience.

3. Discussion and Future Development

3.1 Achievement

The feedback of the users has reflected to us that UMatch was successfully developed and provides a platform for college students to integrate into the campus. The system provides a simple user-friendly interface and collects data so that users do not need to fill in too much information. The system uses a designed algorithm to find successful potential users who have some commonalities between them and meet their life needs.

3.2 Future Development

Further Customize For All University Student

UMatch is open to HKUST students for registration. After passing the test and acquiring a certain number of users, it is planned to launch to all university students in Hong Kong in the future. They can register through their school emails. Our goal is to develop an application to unite all Hong Kong college students to facilitate their life quality. Especially during the epidemic, they have no chance to contact more people even in their own social circle, so we hope that the software can develop the possibility for college students.

Optimize the App and Provide Guidance

As user-friendly apps, It is very important that an app can be read and used smoothly on any mobile device. Whether the size or user interface of the application, is an indicator for visitors to judge the quality of the app. In the future, we plan to minimize the size of apps and provide the most simple user interface. Also, we will add the tutorial or guidance inside the apps when users first log in. Thus, they can easily and faster master this app and find joy in it.

Provide a Recommender for Trade and Event

The search algorithm is an important element of the apps. In the future, we hope that by analyzing the user's usage habits and user input, the system can facilitate users to search and even recommend what they want. For example, we have collected the

preferences (tags) of users and what they study, and the program can arrange some related products and activities for them to choose from.

4. Conclusion

In this project, we have developed UMatch, a mobile friendship app exclusively for university students. This project aims to provide an integrated app for students to develop a friendship with others having similar interests on campus. It is achieved by developing different functions, namely the Chat Function, the Match Function, the Event Function, and the Trade Function. Users can now chat with others, find other potential users based on the recommendation systems, create and participate in different events, and purchase goods in our app. It is believed that we have implemented the recommendation systems in the Match Function based on different algorithms and AI models to help users make friends without excessive effort.

This project develops a foundation of a university-oriented mobile friendship app. It is initially designed for HKUST students, with a development plan to further customize for students from other local universities in later stages. More functions such as the guidance page, recommendation systems in the Trade Function, and the Event Function can be implemented to improve user experience.

5. Project Planning

5.1 Division of work

	Chan Yin Wan	Ho Hau Wo	Wong Wai Hong	Zhang Shao
Research				
Exploration of related technology	A	A	L	A
Literature Survey	A	A	A	L
Literature review	L	A	L	A
Prototype and UI Design	L	A	A	L
Implementation				
Development of Login System	A	A	L	A
Development of database	A	A	A	L
Development of Chat Page	A	A	L	A
Development of Match Page	L	A	A	L
Development of Event Page	L	A	A	A
Development of Trade Page	A	L	A	A
Testing	L	A	A	A
Recommendation system	A	A	A	L
Documentation				
Proposal	L	A	A	A
Monthly Report	A	A	L	A
Progress Report	A	A	L	A
Final Report	L	A	A	A
Presentation				
Poster Design	A	A	A	L
Presentation	A	A	L	A
Video making	A	L	A	L

L = Leader A = Assistant

5.2 Gantt Chart

Task	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr
Research										
Literature Survey										
Literature Review										
Research, analysis and selection of mobile framework										
Research, analysis and selection of backend framework										
Prototype and UI Design										
Implementation										
Develop Login System										
Develop Database										
Development of Chat Page										
Development of Match Page										
Development of Event Page										
Development of Trade Page										
Testing										
Recommendation system										
Documentation										
Proposal										
Monthly Report										
Progress Report										
Final Report										
Presentation										
Prepare for the presentation										
Project Poster										
Video making										

6. HARDWARE AND SOFTWARE REQUIREMENT

6.1 Hardware Requirements

Mobile Type	**Model**
iPhone	iPhone 6s or later
Android Phone	Android 6 higher

6.2 Software Requirements

Software name	**Function**
Flutter	Programming languages
Visual Studio Code	Compiler
Google Cloud Platform	Server
App Engine	Algorithm implementation
Adobe XD	For designing the UI

7. Reference

[1] V. M. Buote, S. M. Pancer, M. W. Pratt, G. Adams, S. Birnie-Lefcovitch, J. Polivy, and M. G. Wintre, “The 3 importance of friends,” *Journal of Adolescent Research*, vol. 22, no. 6, pp. 665–689, 2007.

[2] “To have a meaningful impact,” *Rakuten Insight*. [Online]. Available: https://insight.rakuten.com/ . [Accessed: 18-Aug-2021].

[3] Powering Tinder® — The Method Behind Our Matching, Tinder. [Online]. Available: https://blog.gotinder.com/powering-tinder-r-the-method-behind-our-matching/.[Accessed: 25-8-2021]

[4] ”MeetUp”. [Online]. Available: https://www.meetup.com/en-US/ [Accessed: 25-8-2021]

[5] “Find the people you've crossed paths with,” *happn*. [Online]. Available: https://www.happn.com/en/ . [Accessed: 01-Sep-2021].

[6] “Gööp - for the group you love,” *Gööp Web*. [Online]. Available: https://goop.ai/ . [Accessed: 01-Sep-2021].

[7] Google Cloud Platform.Google Cloud.[Online] Available: https://cloud.google.com/gcp. [Accessed: 16-9-2021]

[8] Firebase Documentation, Firebase. [Online] Available: https://firebase.google.com/docs/firestore/rtdb-vs-firestore.[Accessed: 16-9-2021]

[9] “FAQ | flutter,” *Flutter*. [Online]. Available: https://docs.flutter.dev/resources/faq. [Accessed: 25-Jan-2022].

[10] Algolia, *Algolia*.[Online] Available:https://www.algolia.com/. [Accessed:25-Jan-2022]

[11] “Fuzzy: Dart package,” *Dart packages*, 07-Jan-2021. [Online]. Available: https://pub.dev/documentation/fuzzy/latest/data_fuzzy_options/FuzzyOptions-class.html [Accessed: 20-Apr-2022].

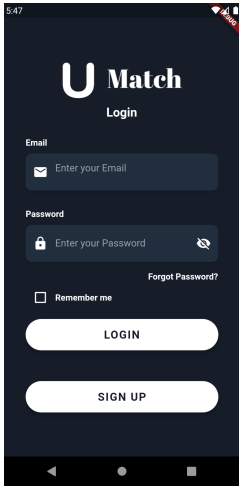
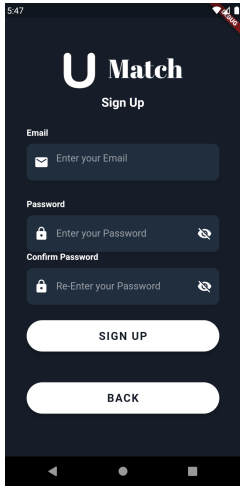
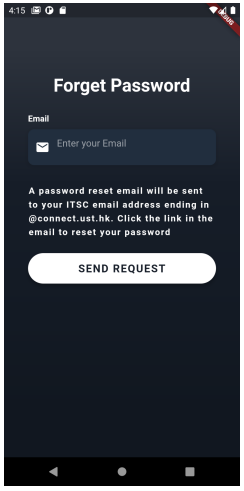
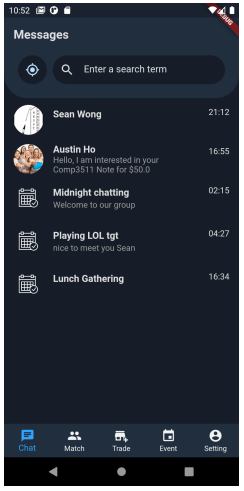
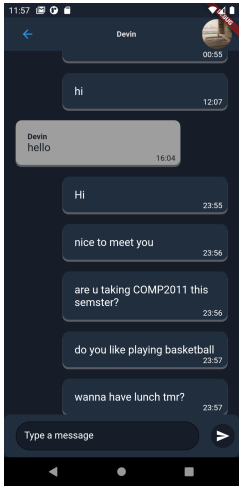
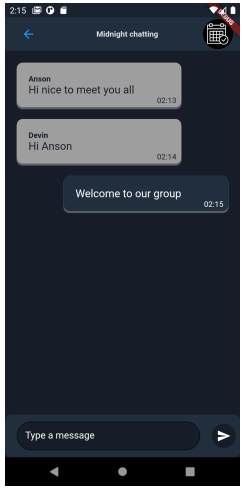
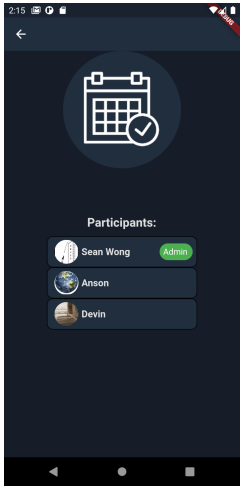
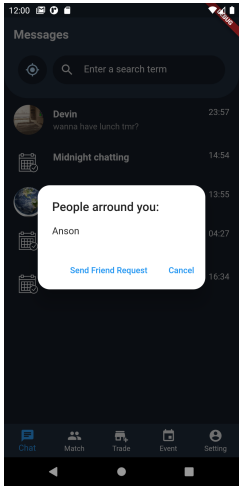
[12] Adarsh2021 (2021) FaceDectection [Source Code]. https://github.com/adarsh1021/facedetection/

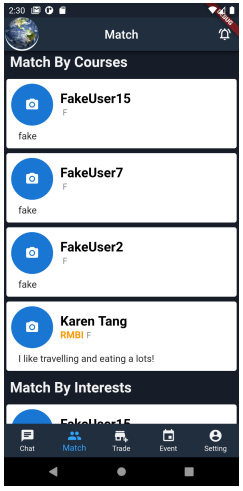
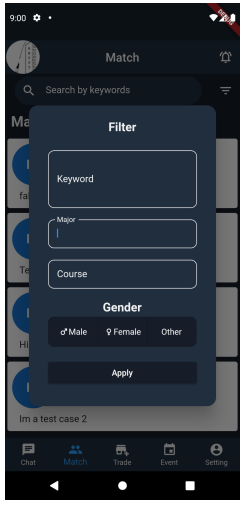
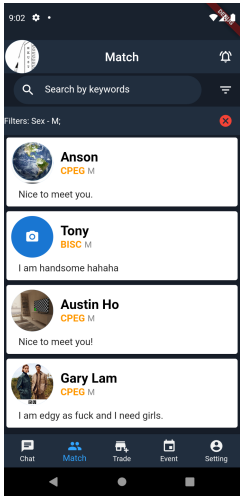
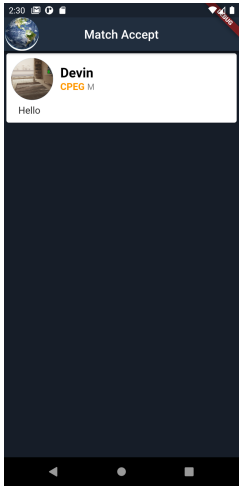
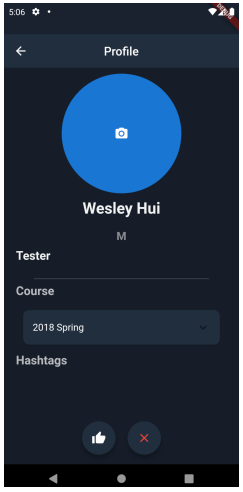
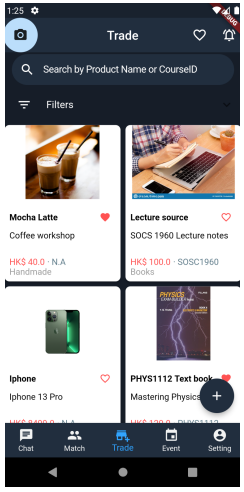
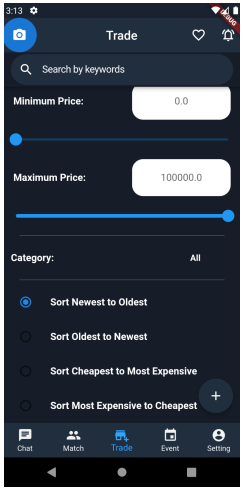
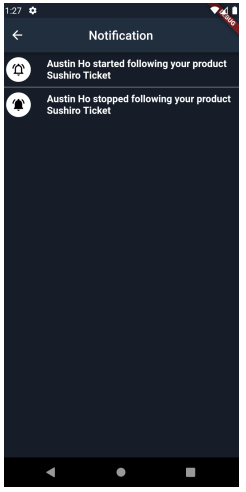
[13] python-rake.[Source Code].https://github.com/fabianvf/python-rake

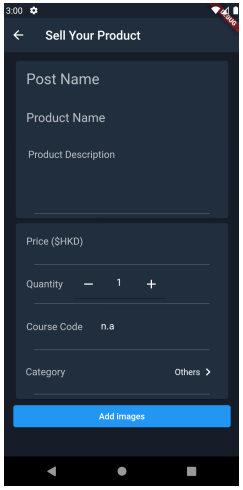
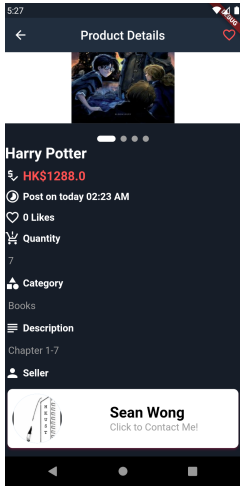
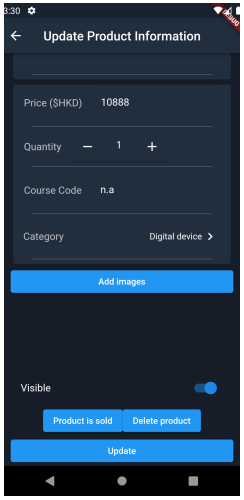
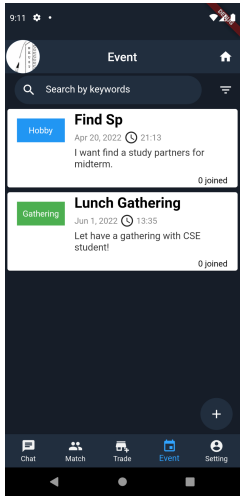
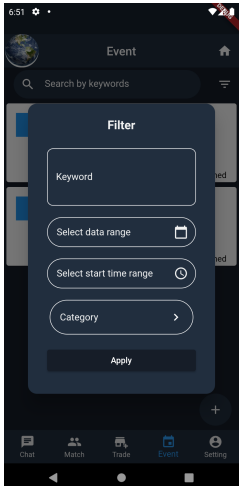
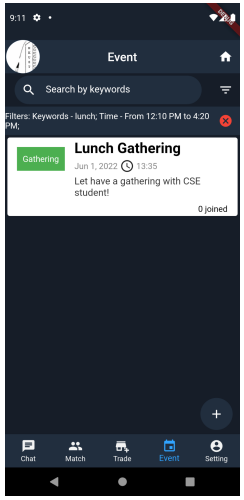
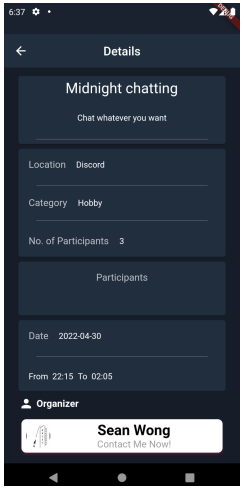
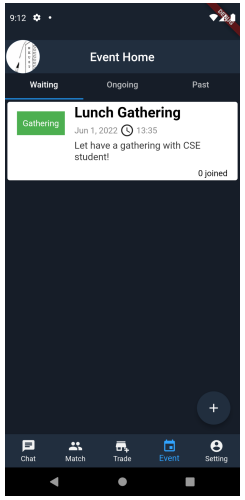
[14] Salton,smartstoplists (1971), SMART.[Source Code]. https://github.com/fabianvf/python-rake/blob/master/RAKE/stoplists/SmartStopList.py

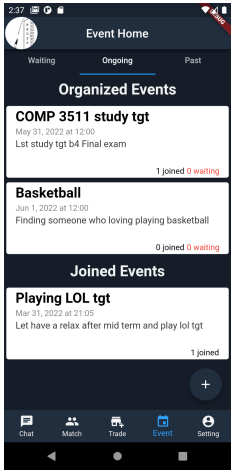
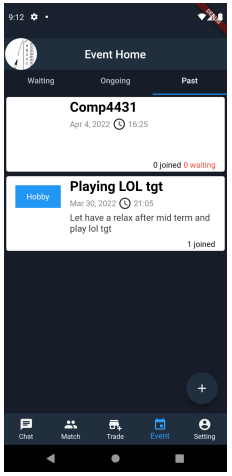
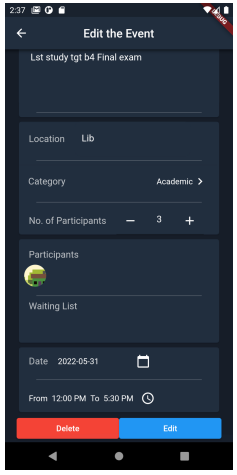
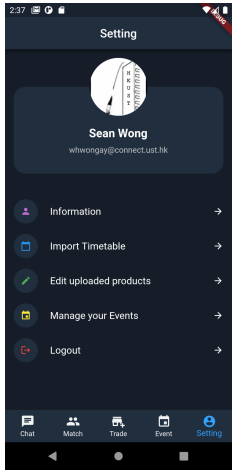
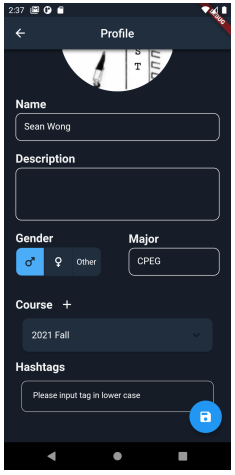
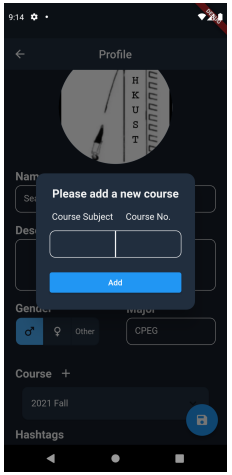
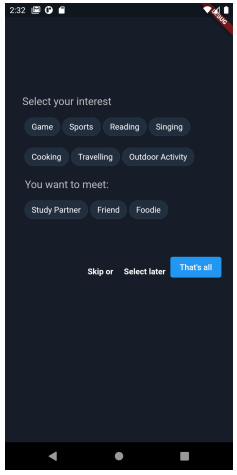
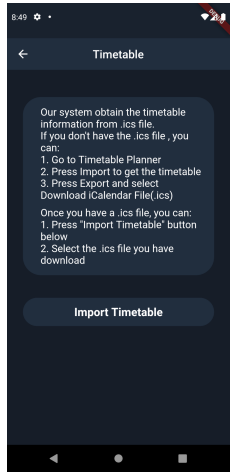
[15] D. Dary, “Selenium for Android,” *Selendroid*. [Online]. Available: http://selendroid.io/ . [Accessed: 01-Sep-2021].

8. Appendix A - Page Naming Descriptions

Login Page	Sign Up Page	Forget Password Page	Chat Page
			
Chat Room	**Group Chat Room**	**View Group Members**	**Search Nearby Person**
			

Match Page	Match Search	Match Page after search	Pending Request
			
View Profile	**Trade Page**	**Trade Filter**	**Trade Notification**
			

Sell Your Product Page	Product Details Page	Update Product Page	Event Page
			
Event Search	**Event Page after search**	**View Event Details**	**Event Home (Waiting)**
			

Event Home (Onging)	Event Home (Past)	Edit Event	Setting Page
			
Edit Profile	**Edit Profile (Add Course)**	**Profile Survey Page**	**Import Timetable**
			

9. Appendix B - Evaluation Survey Question and Survey Result

CECI1 Evaluation Survey

Our main goal is to develop a friendship application in order to provide a platform for students to integrate into Campus. Therefore, our objectives are as follows:

1. To provide a user-friendly interface to make friends without excessive effort.
2. To facilitate social activities and communications among university students in an integrated app.
3. To improve the study life of UST students

How satisfied are you with the following functions?

	Very Dissatisfied	Dissatisfied	Neutral	Satisfied	Very Satisfied
Chat function	○	○	○	○	○
Match function	○	○	○	○	○
Trade function	○	○	○	○	○
Event function	○	○	○	○	○
Profile function	○	○	○	○	○
Overall	○	○	○	○	○

Do you find the system useful to build up your profile?

E.g Import Timetable, Retrived hashtag from profile picture/description

	1	2	3	4	5	
Not Useful at all	○	○	○	○	○	Very Useful

How satisfied are you with the Match recommendation result?

	1	2	3	4	5	
Very Dissatisfied	○	○	○	○	○	Very Satisfied

Overall, do you find the app useful to facilitate social activities and communications with other students?

	1	2	3	4	5	
Not Useful at all	○	○	○	○	○	Very Useful

Overall, do you find the app user-friendly?

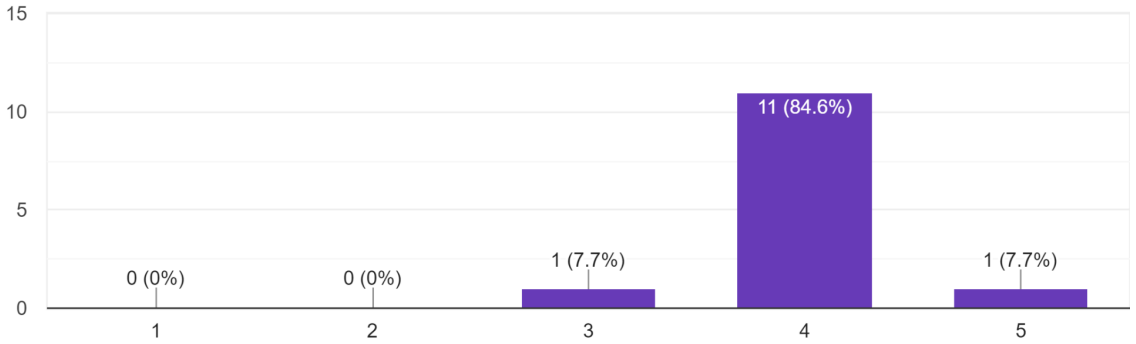
	1	2	3	4	5	
Not User-friendly at all	○	○	○	○	○	Very User-friendly

Survey Result

How satisfied are you with the following functions?

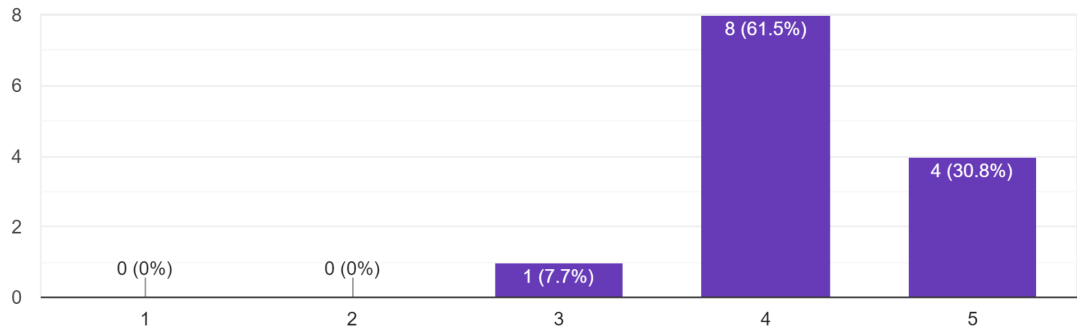
Do you find the system useful to build up your profile?

13 則回應



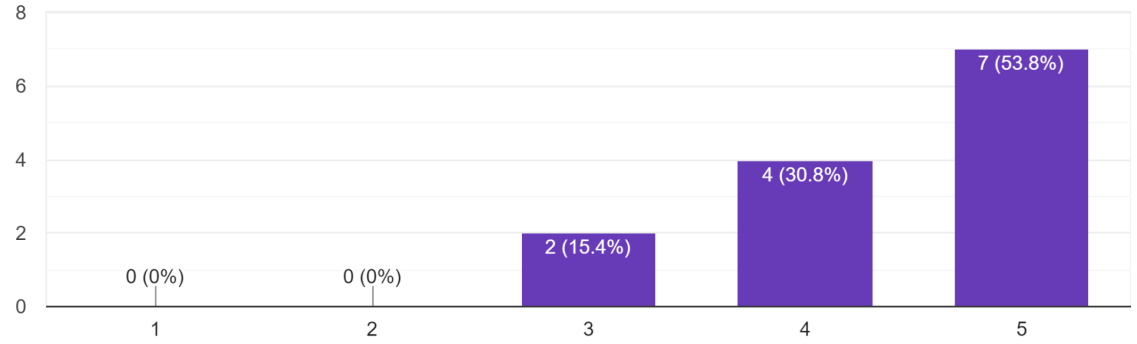
How satisfied are you with the Match recommendation result?

13 則回應



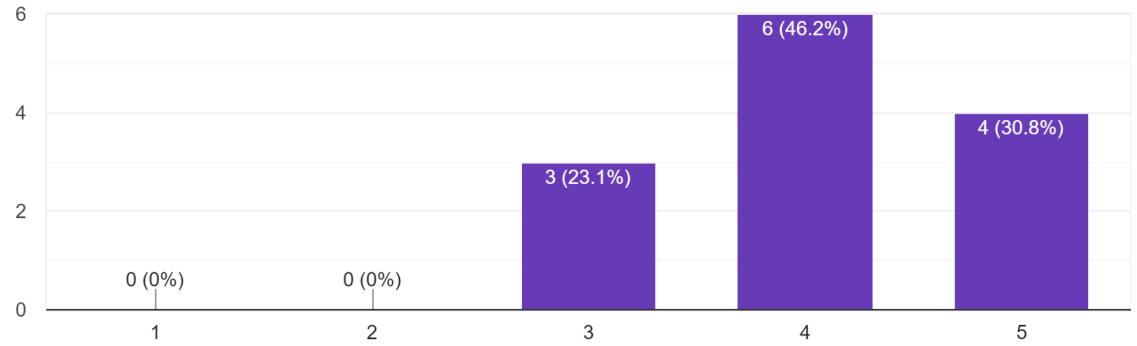
Overall, do you find the app useful to facilitate social activities and communications with other students?

13 則回應



Overall, do you find the app user-friendly?

13 則回應



10. Appendix C - Meeting Minutes

10.1 Minutes of the 1st Project Meeting

Date:	July 4, 2021
Time:	12:00 noon
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Ho Hau Wo(Austin), Zhang Shao(Devin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Chan Yin Wan (Anson)

1. Discussion items

   1.1 Details of Recommendation System

   1.2 Special Feature

   1.3 Objective

2. Remark Item

   2.1 The important day of FYP (Report due date)

3. Goals for the coming week

   3.1 Literature review

   3.2 Research related development tools

4. Meeting adjournment and next meeting

The meeting was adjourned at 13:00 PM.

The next meeting will be on July 7th at 12:00 noon on Zoom

10.2 Minutes of the 2nd Project Meeting

Date:	July 18, 2021
Time:	12:00 noon
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Chan Yin Wan (Anson)

1. Report on progress

    1.1. Identified the disadvantage of the existing application

    1.2 Decided to use Flutter as the programming language to implement both IOS and Android platform

2. Discussion items

    2.1. Feature Discussion

        - Matching system

        - Second-hand book Trade System

        - IOT

3. Goals for the coming week

    3.1 Define Objective

    3.2 Define Feature with detail

4. Meeting adjournment and next meeting

The meeting was adjourned at 12:45 PM.

The next meeting will be on Aug 1st at 12:00 noon on Zoom

10.3 Minutes of the 3rd Project Meeting

Date:	August 1, 2021
Time:	12:00 noon
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Ho Hau Wo(Austin), Zhang Shao(Devin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Chan Yin Wan (Anson)

1. Report on progress

    1.1 Decided to workflow of the mobile application

    1.2 Decide the detail of the matching system

    1.3 Designed the UI of Login, matching screen

2. Discussion items

    2.1. Feature Discussion

        - Manual Matching System

        -Recommendation Matching System

    2.2 Front-end/ back-end Design

3. Goals for the coming week

    3.1 Database design

    3.2 Recommendation system research

    3.3 Timetable Import attempt

4. Meeting adjournment and next meeting

The meeting was adjourned at 12:45 PM.

The next meeting will be on Aug 15th at 12:00 noon on Zoom

10.4 Minutes of the 4th Project Meeting

Date:	August 15, 2021
Time:	12:00 noon
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Ho Hau Wo(Austin), Zhang Shao(Devin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Wong Wai Hong(Sean)

1. Report on progress

    1.1 Designed the structure of the database

    1.2 Design the method to import the timetable

2. Discussion items

    2.1 Feature Discussion

        - Obtain keyword from description

    2.2 Proposal Report

3. Goals for the coming week

    3.1 Proposal Report (send draft to Cecia on 1st Sep)

    3.2 App Implementation

4. Meeting adjournment and next meeting

The meeting was adjourned at 12:30 PM.

The next meeting will be on Sep 3rd at 9:00 PM on Zoom

10.5 Minutes of the 5th Project Meeting

Date:	September 3, 2021
Time:	09:00 pm
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Chan Yin Wan (Anson)

1. Report on progress

    1.1 Proposal Report

2. Discussion items

    2.1 Proposal Report

    2.2 Meeting Time (Confirmed to hold face-to-face meeting on Mon at 10:00am)

3. Goals for the coming week

    3.1 Improve Proposal Report based on the discussion (send draft to Cecia before 10 Sep)

    3.2 App Implementation

4. Meeting adjournment and next meeting

The meeting was adjourned at 10:10 PM.

The next meeting will be on Sep 11th at 11:00 AM on Zoom

10.6 Minutes of the 6th Project Meeting

Date:	September 11, 2021
Time:	10:30 am
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Wong Wai Hong(Sean)

1. Report on progress

    1.1 Proposal Report

2. Discussion items

    2.1 Proposal Report

3. Goals for the coming week

    3.1 Improve Proposal Report based on the discussion (send draft to Cecia before 17 Sep)

    3.2 App Implementation

4. Meeting adjournment and next meeting

The meeting was adjourned at 11:12 AM.

The next meeting will be on Sep 27th at 10:00 AM.The place will be set as the CSE Conference Room.

10.7 Minutes of the 7th Project Meeting

Date:	September 27, 2021
Time:	10:00 am
Place:	Room 3520
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Zhang Shao(Devin)

1. Report on progress

    1.1 basic UI of the app

    1.2 basic Login system

2. Discussion items

    2.1 The important day of FYP

    2.2 Preparation for the presentation

3. Goals for the coming week

    3.1 Plan Schedule

    3.2 App Implementation

4. Meeting adjournment and next meeting

The meeting was adjourned at 10:30 AM.

The next meeting will be on Oct 11th at 10:00 AM. The place will be set as the CSE Conference Room.

10.8 Minutes of the 8th Project Meeting

Date:	October 11, 2021
Time:	10:00 am
Place:	Room 3520
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Chan Yin Wan (Anson)

1. Report on progress

    1.1 Timetable import function

    1.2 improved Login system

2. Discussion items

    2.1 evaluation method

    2.2 data processing

    2.3 Algorithm design

3. Goals for the coming week

    3.1 App Implementation

4. Meeting adjournment and next meeting

The meeting was adjourned at 10:45 AM.

The next meeting will be on Oct 25th at 10:00 AM. The place will be set as the CSE Conference Room.

10.9 Minutes of the 9th Project Meeting

Date:	October 25, 2021
Time:	10:00 am
Place:	Room 3520
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Chan Yin Wan (Anson)

1. Report on progress

    1.1 Recommendation Algorithm

    1.2 UI

2. Discussion items

    2.1 evaluation method

    2.2 UI design

    2.3 Algorithm design

3. Goals for the coming week

    3.1 App Implementation

    3.2 Algorithm Implementation

4. Meeting adjournment and next meeting

The meeting was adjourned at 10:45 AM.

The next meeting will be on Nov 8th at 10:00 AM. The place will be set as the CSE Conference Room.

10.10 Minutes of the 10th Project Meeting

Date:	November 8, 2021
Time:	10:00 am
Place:	Room 3520
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Chan Yin Wan (Anson)

1. Report on progress

    1.1 UI

2. Discussion items

    2.1 UI design

    2.2 Algorithm design

3. Goals for the coming week

    3.1 App Implementation

    3.2 Algorithm Implementation

    3.3 Monthly report

4. Meeting adjournment and next meeting

The meeting was adjourned at 10:45 AM.

The next meeting will be on Dec 1st on Zoom.

10.11 Minutes of the 11th Project Meeting

Date:	December 1, 2021
Time:	6:20 pm
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Chan Yin Wan (Anson)

1. Report on progress

   1.1 UI

2. Discussion items

   2.1 Timetable import

   2.2 Arrangement in Winter Sem

3. Goals for the coming week

   3.1 Short-term goals/ Long-term goals

4. Meeting adjournment and next meeting

The meeting was adjourned at 18:40 PM.

The next meeting will be on 21/12 at 14:00 PM. The place will be set as the CSE Conference Room.

10.12 Minutes of the 12th Project Meeting

Date:	December 21, 2021
Time:	2:00 pm
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Chan Yin Wan (Anson)

1. Report on progress

   1.1 Arrangement in Winter Sem(Short-term goals/ Long-term goals)

2. Discussion items

   2.1 Improvement in Arrangement in Winter Sem

3. Goals for the coming week

   3.1 Please according to the schedule

4. Meeting adjournment and next meeting

The meeting was adjourned at 14:23 PM.

The next meeting will be on 1/3 at 14:00 PM. The place will be set as the CSE Conference Room.

10.13 Minutes of the 13th Project Meeting

Date:	January 3, 2022
Time:	2:00 pm
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Chan Yin Wan (Anson)

1. Report on progress

    1.1 UI of the app(Chat Page, Event Page, Trade Page)

    1.2 Design of algorithm

2. Discussion items

    2.1 Equipments for presentation

    2.2 Algorithm design for Event, Trade and Match

    2.3 Timetable

3. Goals for the coming week

    3.1 Please according to the schedule

4. Meeting adjournment and next meeting

The meeting was adjourned at 15:00 PM.

The next meeting will be on 1/21 at 16:00 PM. The place will be set as the CSE Conference Room.

10.14 Minutes of the 14th Project Meeting

Date:	January 21, 2022
Time:	4:00 pm
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Chan Yin Wan (Anson)

1. Report on progress

    1.1 UI of the app(Chat Page, Event Page, Trade Page)

    1.2 Algorithm for image similarity

2. Discussion items

    2.1 Improvement on image similarity (color style)

    2.2 Algorithm design for Trade

3. Goals for the coming week

    3.1 Please according to the schedule

    3.2 Progress report 1st draft submitted on January 28

4. Meeting adjournment and next meeting

The meeting was adjourned at 16:35 PM.

The next meeting will be on 1/31 at 11:30 AM. The place will be set on Zoom.

10.15 Minutes of the 15th Project Meeting

Date:	January 31, 2022
Time:	11:30 am
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Wong Wai Hong(Sean)
Absent:	Ho Hau Wo(Austin)
Recorder:	Wong Wai Hong(Sean)

1. Report on progress

    1.1 Progress Report

    1.2 GPS function

2. Discussion items

    2.1 Improvement on Progress Report

3. Goals for the coming week

    3.1 Progress report 2nd draft submitted on February 10

4. Meeting adjournment and next meeting

The meeting was adjourned at 12:55 PM.

The next meeting will be on 2/11 at 18:00 PM. The place will be set on Zoom.

10.16 Minutes of the 16th Project Meeting

Date:	February 11, 2022
Time:	18:00 am
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Zhang Shao(Devin)

1. Report on progress

    1.1 Progress Report

    1.2 GPS function

2. Discussion items

    2.1 Improvement on Progress Report

3. Goals for the coming week

    3.1 Progress report 3rd draft submitted on February 13, 14

4. Meeting adjournment and next meeting

The meeting was adjourned at 18:40 PM.

The next meeting will be on 2/28 at 12:00 noon. The place will be set on Zoom.

10.17 Minutes of the 17th Project Meeting

Date:	February 28, 2022
Time:	12:00 noon
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Zhang Shao(Devin)

1. Report on progress

    1.1 Group Chat

    1.2 Manage Tag and Course in Profile Page

2. Discussion items

    2.1 Evaluation Date and Process

    2.2 Presentation Flow

3. Goals for the coming week

    3.1 App Implementation

    3.2 Algorithm Implementation

4. Meeting adjournment and next meeting

The meeting was adjourned at 12:33 PM.

The next meeting will be on 3/16 at 12:00 noon. The place will be set on Zoom.

10.18 Minutes of the 18th Project Meeting

Date:	March 16, 2022
Time:	12:00 noon
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Wong Wai Hong(Sean)
Absent:	Ho Hau Wo(Austin)
Recorder:	Chan Yin Wan (Anson)

1. Report on progress

    1.1 UI improvement

    1.2 GCP connection

2. Discussion items

    2.1 UST elements

    2.2 Server

3. Goals for the coming week

    3.1 App Implementation

    3.2 Algorithm Implementation

4. Meeting adjournment and next meeting

The meeting was adjourned at 12:31 PM.

The next meeting will be on 3/28 at 12:00 noon. The place will be set on Zoom.

10.19 Minutes of the 19th Project Meeting

Date:	March 28, 2022
Time:	12:00 noon
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Chan Yin Wan (Anson)

1. Report on progress

    1.1 GCP connection

2. Discussion items

    2.1 Evaluation Detail

3. Goals for the coming week

    3.1 App Implementation

    3.2 Algorithm Implementation

4. Meeting adjournment and next meeting

The meeting was adjourned at 12:31 PM.

The next meeting will be on 4/4 at 12:00 noon. The place will be set on Zoom.

10.20 Minutes of the 20th Project Meeting

Date:	April 4, 2022
Time:	12:00 noon
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Zhang Shao(Devin)

1. Report on progress

    1.1 UI

2. Discussion items

    2.1 Improvement on UI

3. Goals for the coming week

    3.1 Evaluation

    3.2 Final report

4. Meeting adjournment and next meeting

The meeting was adjourned at 12:44 PM.

The next meeting will be on 4/13 at 12:00 noon. The place will be set on Zoom.

10.21 Minutes of the 21st Project Meeting

Date:	April 13, 2022
Time:	12:00 noon
Place:	Zoom
Present:	Dr. Cecia Ki CHAN (supervisor), Chan Yin Wan (Anson), Zhang Shao(Devin), Ho Hau Wo(Austin), Wong Wai Hong(Sean)
Absent:	None
Recorder:	Chan Yin Wan (Anson)

1. Report on progress

    1.1 Final Report

2. Discussion items

    2.1 Presentation detail

3. Goals for the coming week

    3.1 Final report Improvement

4. Meeting adjournment and next meeting

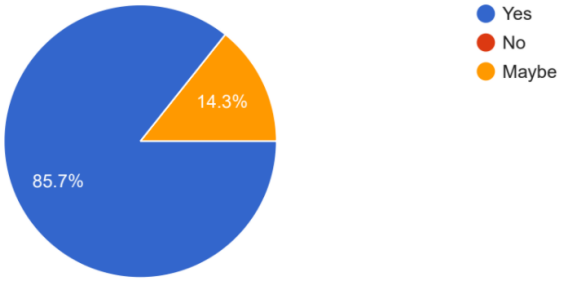
The meeting was adjourned at 13:02 PM.

The next meeting will be on 4/25 at 12:00 noon. The place will be set on Zoom.

11. Appendix D - Survey of social platforms for HKUST students

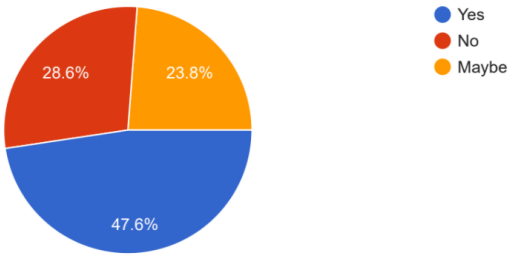
1. Under the epidemic, is it more difficult to make friends at school?

42 則回應



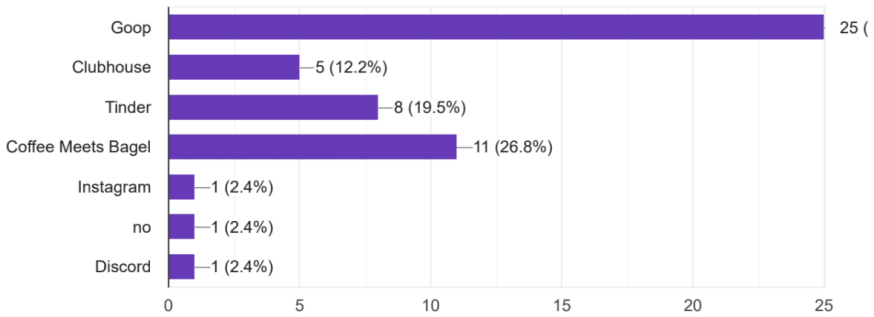
2. Will you use social platforms or dating apps to meet school friends? (Examples listed in Q3)

42 則回應



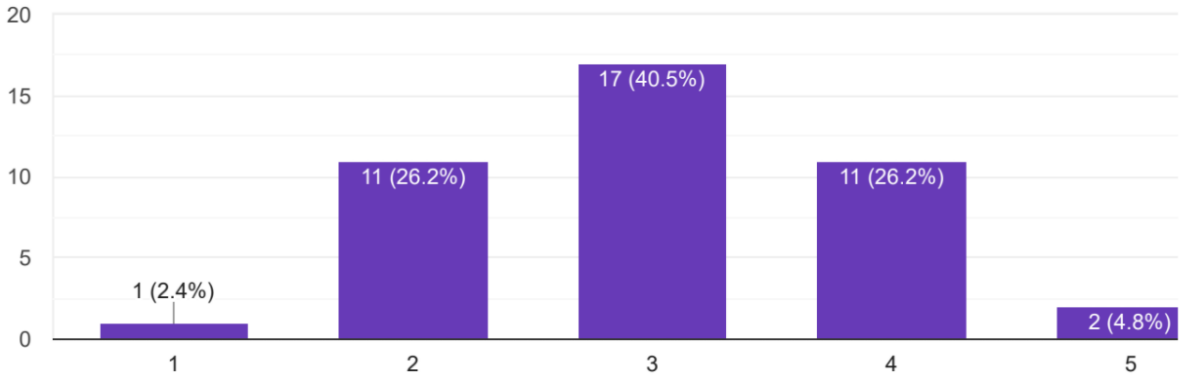
3. Have you ever used any of the following platforms?

41 則回應



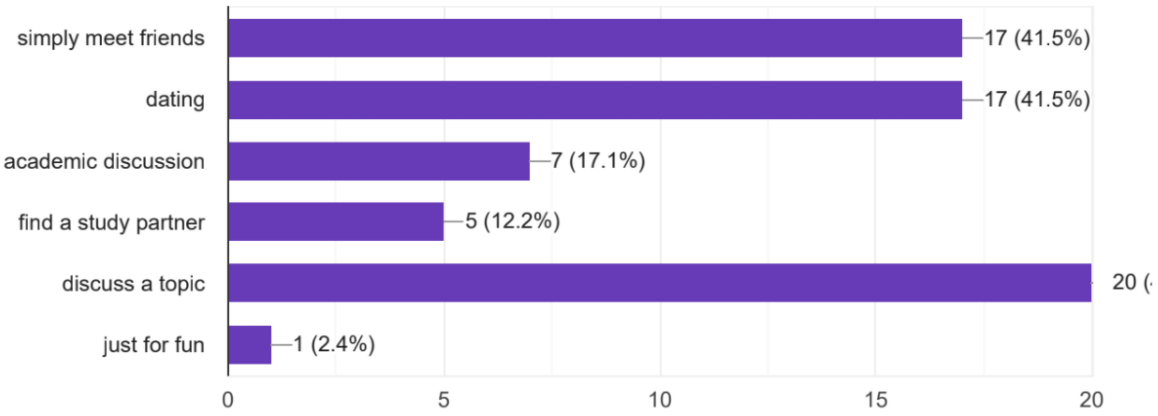
4. How would you rate these apps?

42 則回應



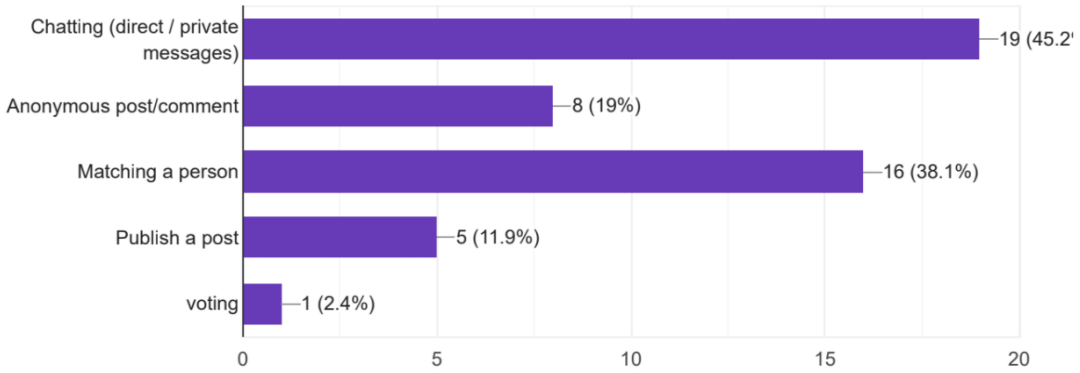
5. What are your main purpose(s) for using these apps?

41 則回應



6. What features of a social platform do you enjoy most?

42 則回應



7. What function do you need most but is lacking?

9 則回應