# Development of the Friend Finder - 6GZ1108 - Professional Development

Emily Chomicz, Robert Bowden, Yoanne Youmbi, Bradley Tabbron, Daniel Bethell

## Contents

**1 Abstract** **3**

**2 Introduction** **3**

**3 Research & Analysis** **4**
3.1 Existing Solutions . . . 4
3.1.1 Tinder . . . 4
3.1.2 Discord . . . 4
3.1.3 Facebook . . . 4
3.2 Market Research . . . 4
3.2.1 Demographics . . . 4
3.2.2 Case Study: Qamar . . . 5
3.3 Ethical and Legal Considerations . . . 5
3.3.1 General Data Protection Act . . . 5
3.3.2 Information Consent . . . 5
3.3.3 Moral Concerns . . . 5

**4 Design** **7**
4.1 Requirements . . . 7
4.2 Class Diagrams . . . 7
4.3 User Case Diagrams . . . 7
4.4 User Interface Designs . . . 7

**5 Implementation** **8**
5.1 Back-end . . . 8
5.1.1 WAMP . . . 8
5.1.2 SQL . . . 8
5.1.3 JSON . . . 8
5.1.4 PHP . . . 8
5.1.5 OKHttp . . . 8
5.1.6 Implementation Class Diagram . . . 8
5.2 Front-end . . . 8
5.2.1 Java . . . 8

**6 Manual** **10**
6.1 Log In . . . 10
6.2 Register . . . 11
6.3 Chat Tab . . . 12
6.4 Matching Tab . . . 13
6.5 View & Edit Profile Tab . . . 14

**7 Handover** **15**
7.1 Source Code Management . . . 15
7.1.1 GitHub . . . 15
7.1.2 Jenkins . . . 15
7.1.3 Code . . . 15

7.2 If we had more time . . . 15
7.2.1 Sockets . . . 15
7.2.2 Authentication . . . 15
7.2.3 Chat Feature . . . 15
7.2.4 Criteria . . . 16

**8 Conclusion** **17**
8.1 Next Goals . . . 17
8.2 Closing Thoughts . . . 17

**9 Appendix** **19**
9.1 Class Diagrams . . . 19
9.2 Code . . . 21
9.3 User Interface Mockups . . . 22
9.3.1 Initial Mockup . . . 22
9.3.2 Adobe XD . . . 25
9.4 User Case Diagram . . . 27
9.5 Qamar's Conversation . . . 28

# 1 Abstract

We were asked to create a friend finder application for mobile. We explored the different reasons why one may be needed, including social isolation and mental health. We briefly explored Tinder, Discord, and Facebook as alternatives and we found these to be inadequate for making friends. We then asked a user what they would be looking in an alternative app, and we designed the app based on these wishes. We then prototyped the application using Adobe XD and used PHP, Java and JSON in order to create the app. We briefly talk about how this app could be continued by future developers and where we see the app going next.

# 2 Introduction

We were asked by Manchester Metropolitan University to create a friend finder app to alleviate social isolation and boredom. (Crossley 2021) We thought this would be quite a good challenge not only because of it is our Year 2 assignment, but also it reflects on the nature of lockdown and societal isolation in general.

According to the ONS, the main statistical body of the UK government, up to 30.9% adults (7.4 million) have felt lonely over the statistical period of seven days. Of those, 50.8% of those were aged 16-24 (Office of National Statistics 2020). It is clear to us that, having friends who could be around them would greatly benefit that population.

Of course, mental health goes hand in hand with loneliness as well. Up to 34% of students felt that their well-being has declined in 2020. (Randstad 2020), and 55% have considered leaving their course. While 42% of students did go on to access counselling, sometimes one of the ways to help these struggling students is to help them make friends. According to one study in which people with mental illness were placed into intentional friendships, it was noted that self-esteem, self-worth, and self-confidence were improved drastically as a result. (Mccorkle et al. 2009)

Therefore, by being able to forge new friendships through this new app, we would be able to increase people's well-being, and also to alleviate loneliness.

# 3 Research & Analysis

We researched and analysed a number of solutions, as well as talked to potential users.

## 3.1 Existing Solutions

There are a selection of pre-existing solutions on the market that can be used to create new friendships similar to the solution we want to create. We decided to take a deep look at each one individually and scrutinize what they do well and where they are lacking in certain areas. The solutions we are looking at are the most popular currently and we are aware there are lots of other competitors as well.

### 3.1.1 Tinder

Tinder is the most popular online dating application on the market which allows users to like and dislike other people's profiles based on user information such as photos, a small bio and mutual interests. Their unique feature is the ability to swipe on people's profiles which allows the user to quickly match with other users quick and effortlessly making it perfect for a younger demographic. The effortless interface design is the main reason the app works so well for the younger demographic, with a plethora of competition the application with the easiest to use interface will come out on top as young adults will just move to a new platform if the interface isn't satisfactory.

One of the main gripes users have with Tinder is that it can make the dating experience become shallow (The Guardian 2013), this may be because the first thing a user will see is the profile's picture and this may make some users instantly dislike the profile without further looking into the user information. To see further information the user has to click on the profile and then read more about them, this is where we could improve the interface for our solution by not only showing the profile picture but a little extra information that let's a user know more about them. This may improve a few interactions and make the application experience more about the profile's personality rather than their physical attributes.

### 3.1.2 Discord

Discord is a chat application that allows users to connect with their friends or interact with communities that have similar interests to theirs whilst utilising voice, video and text features to make the most of their experience. The main use of Discord is for users to join a server of people with a similar interests or create one for others to join, this allows users to find people with similar interests really easy as they are all in one location and a single chat away.

The main issue some have with Discord is that it is not incredibly easy to find a server of people to join, there are so many for niche topics which can become overwhelming or daunting for some users as they will not know which one to choose. In the same way, the ability to join random servers can seem alien to some and not feel very invited to a community without a personal invite.

### 3.1.3 Facebook

Facebook is an application that allows users to create profiles, connect with friends and share pictures, thoughts or whatever they like. Users send friend requests to people they may know which, when accepted, will allow them to see what the other user posts. Being one of the original social networks, users love the platform for its ability to stay in contact with existing friends without seeing them in person evidently making communication between them a lot easier.

While Facebook makes it easy to connect with existing friends and also send requests to potential future ones, users can feel uncomfortable sending a stranger a request as some may not be using the platform for that reason or it may leave the user open to cyberbullying. Another big issue some have is with privacy, what a user posts can be seem by everyone on the platform unless their profile is set to private. Setting a profile to private restricts all of the users information to strangers which isn't very well for making new friends.

## 3.2 Market Research

### 3.2.1 Demographics

For the demographics of our app we decided to focus on university students who would therefore generally fall between the ages of 18 to 24. We chose this demographic for a number of reasons. One of them was that we figured a lot of students may have moved to an unfamiliar city or may have moved away from home for the first time. Now usually, student fairs and various open events would allow students to join societies in the things that they were interested in, however this option suddenly became unfeasible due to the COVID-19 pandemic. This is one of the reasons we targeted the app at university students, we felt that they may

be feeling lonely because of the pandemic and thought our app could help alleviate that loneliness somewhat.

#### 3.2.2 Case Study: Qamar

We spoke to User A, *Qamar*. We asked what *Qamar* would be looking for in a friendship application. The full conversation can be found in Appendix 9.5.

To briefly summarise, *Qamar* said that the current apps are more based towards dating, and he was looking for something that was more towards making friends. He thought that an app where similar people can meet each other that would be a cool idea, and that it would be effectively *'Tinder for lonely people'*.

*Qamar* said that one of the most important things in a friendship is to have a well-balanced match; not too serious, not too comedic, just about the right amount of maturity, and to share similar, but not the same viewpoints.

*Qamar* also said that current social media sites that he uses have limitations. For example, *Twitter* makes it very easy to talk to people, but it doesn't provide much in the form of friendships. On the other hand, Discord makes it too easy to connect with others, and this makes it hard to separate work and play. No opinion was given on Facebook, but it was implied that *Qamar* has a negative opinion about it.

We think *Qamar* is a good, typical user of social media, as he is within our age range (18-24), a student of another university, and generally tech savvy.

### 3.3 Ethical and Legal Considerations

We considered a number of potential considerations for legal and ethical dilemmas. Most of these were researched by Bradley.

#### 3.3.1 General Data Protection Act

It is fair to assume that a friend finding app will have sensitive information a user would like to be kept private linked to their profile. Thus, it will be imperative for the data handler of the app to be familiar with the GDPR and to have good encryption. A real-world example of the fallout that can happen when the issue of data not being secure enough and being compromised happened last year when around 845 gigabytes of data from dating and matchmaking apps was leaked when it was found that an amazon web service containing the data was accidentally made public. (Rotem & Locar 2020)

The data included sexually explicit photos and audio recordings as well as screenshots of private chats sent between users within the app as part of the relationships they were building. Data leaks like this bring with it a risk of extortion and/or psychological abuse and users of one of these apps didn't expect that others outside the app would be able to see and download the data." This raises the question of whether it is even ethical to keep the sensitive data of users knowing that it may not be safe.

A possible solution to this problem could be to establish a government body that would check in on data holders periodically to check that they are keeping their security to a good enough standard much like hygiene inspectors do at restaurants. This could help minimise the chance of data breaches such as the this.

#### 3.3.2 Information Consent

The issues surrounding information consent are issues that extend beyond the matchmaking app space and are applicable for the entire online world. A very common practice with technology companies is to sell or share the information they have about their users with third parties and other advertisers. Often users may not be aware that that their information is being used in this way or the extent of the usage.

Whilst there is a European law that makes websites show you which third party vendors they share your information with and should allow you to opt-out (ICO 2016), companies often deliberately try and make opting out very difficult and arduous. This raises an ethical question of whether it is okay to try and trick the user into granting uses of their information they otherwise may not have given if it could benefit the user experience.

That is the fine line, if you don't collect any data, it may be hard for a friend finding app to produce accurate matches but then again, people should have the right to not have their data used.

#### 3.3.3 Moral Concerns

Filtering is a common tool used to help whittle down the list to help users get the results they were looking for. Everything from online shopping to streaming services use filtering however, its use in matchmaking apps can bring about some questions regarding ethics.

More often than not friend finding and dating apps filter based on age, gender, height, race, education level, religion and political views amongst other things and whilst filtering in online shopping is always a good thing because it makes you more likely to get the item you want; in friend finding this could be seen as discriminatory given that humans who have feelings are involved.

You could say that you'd be likely to find the type of friend you are looking for on a friend finding app that uses filtering but on the other hand you could say that this could help encourage a closed mindset and this is worse than not having the best results.

Practically all friend finding and dating apps filter by age and gender, but one could argue that this information would be required for a user to find a friend who is in their own age group and (either) their own or the opposite gender. There are apps in use that filter by religious beliefs some examples being *Muzmatch* a friend finding and dating app exclusively for Muslims and *Christianconnect*, a matchmaking app exclusively for Christians. These apps could encourage users to not look outside of what they know when looking for friends a bad message to give out.

# 4 Design

We designed our app before we started developing.

## 4.1 Requirements

We considered what is and what isn't feasible within the timeframe in order to make a Minimum Viable Product (MVP).

We felt that basic functionality, such as finding a friend and to edit your profile, would be considered minimal required functionality, and that it **must** be included.

We settled on having on four different potential friend matches at any one time. In order to do that, we settled on an SQLite database on the back-end, and to communicate between the back/front-end using the JSON format.

Each profile had to show basic fields, such as name, age, bio, some pictures of each person, the location, and their hobbies and criteria.

The app had to be able to filter by these hobbies, and to disqualify any matches that do not fit within the criteria.

The app had to have a simple interface; swipe left to discard the match, and swipe right to accept the friend match. The other user should be notified of this.

We then considered some other possibilities; we considered a points system where you could assign points to criteria and if those criteria significantly differed, it would discriminate on the basis of these criteria. Such criteria could be *is an introvert/extrovert*, *likes to party*, and so on.

We also considered some other possibilities, such as implementing machine learning, implementing a chat feature, and having little games inside the app you could play. Unfortunately, due to time constraints, these features were scrapped and didn't make it into the final product.

Finally, we decided to not implement WebRTC for audio/video at this time, as it would take too much work, and we also decided that at this time, moderation tools and per-university registrations (for our target audience), would be too much work for a short development time and a small team.

## 4.2 Class Diagrams

We decided to make a class diagram in order to help us structure and frame our code. This class diagram can be found in Appendix 9.1

However, our code has changed significantly since the Class Diagram was made, as it was made with sockets in mind. More of an explanation on Sockets can be found in 7.2.1.

## 4.3 User Case Diagrams

In order to accurately plan out what a UX experience would look like, we planned out a User Case Diagram, which can be found in Appendix 9.4.

## 4.4 User Interface Designs

The original design of the user interface was done by Rob, who took our initial ideas and gathered them into a possible UI design. We unanimously agreed that we liked the design (Appendix 9.3.1) and we decided to take it to prototyping.

For prototyping, Dan used Adobe XD, which allowed us for swift prototyping of ideas, and to get an idea of how our program will work. These diagrams can be found in Appendix 9.3.2.

# 5 Implementation

## 5.1 Back-end

### 5.1.1 WAMP

In order to use the PHP server, we have decided to use WAMP. We decided to use WAMP because of its simple installation method and the fact it's just one click to launch the software, which greatly aided development speed.

### 5.1.2 SQL

SQL is a structured query language that allows us to use a database in order to store our user details, which has speed and reliability advantages over formats such as CSV files, as well as easier to manage and than just a load of JSON files. Some other aspects of the project we either modified or culled because of time include the change from using SQLite to MySQL. Whilst SQLite file sizes are typically smaller in size making for a more compact app, we found that during development we would rather have had the benefits of the full suite so made the chance. In addition, MySQL also works better with WAMP and XAMPP.

Our SQL database consists of one table which contains the name, password, bio, the image in Base64, and email and date of birth.

### 5.1.3 JSON

We decided to use JSON as our data interchange format, as this works better with Android as it is the defacto standard used in Android development, as well as it being easy to send over the internet.

We use JSON in order to send attributes about profiles in order to be able to search them and to show the profiles in the user interface.

### 5.1.4 PHP

We decided to use PHP based on the fact it is a stable and mature programming language used on the server side. It is used by millions of people and therefore it has been battle-tested, and with plenty of community support.

We have three PHP scripts to search for profiles, to log in and also to create a new person. Both scripts connect to the SQL database; the create a new person one runs a query to INSERT a new person given their attributes, and then returns whether it has been successful. The search one looks through the whole table, and adds each entry to an array, which is then returned in JSON format. Finally, we have a login script, that allows the user to register by selecting all profiles and seeing if the user and password exist, and filling in the UserProfile class if they do.

We've also got several other scripts for deleting a person, getting their details, and updating them; these did not take into the final product.

### 5.1.5 OKHttp

As an alternative to using sockets, we used the OKHttp library, which does all the socket and client networking stuff for us. We simply make a connection to the PHP server and we post the JSON to it using this library. We decided to use it based on the fact that it was easy to use, and it was easy to include in the Android project.

### 5.1.6 Implementation Class Diagram

A class diagram of the implementation can be found in Appendix 9.1.

## 5.2 Front-end

### 5.2.1 Java

Android Studio is the interactive development environment for native development of Android applications. We decided as a team that because of our group knowledge of Java, which had been a large focus of our programming studies at university, that developing for Android would be the best solution. This is not only due to Android applications being developed in Java (as well as Kotlin) but also because of the accessibility of mobiles in our modern times. We used Android SDK 29 for our base.

Android is an operating system that takes up a share of 72.57% of all mobile phones worldwide (Statcounter, GlobalStats 2020). This makes it a great choice for development when deploying an application attempting to reach a large majority of people, which we are attempting to achieve with our software solution.

When developing the front-end of an Android application XML is the mark-up language used. In an-

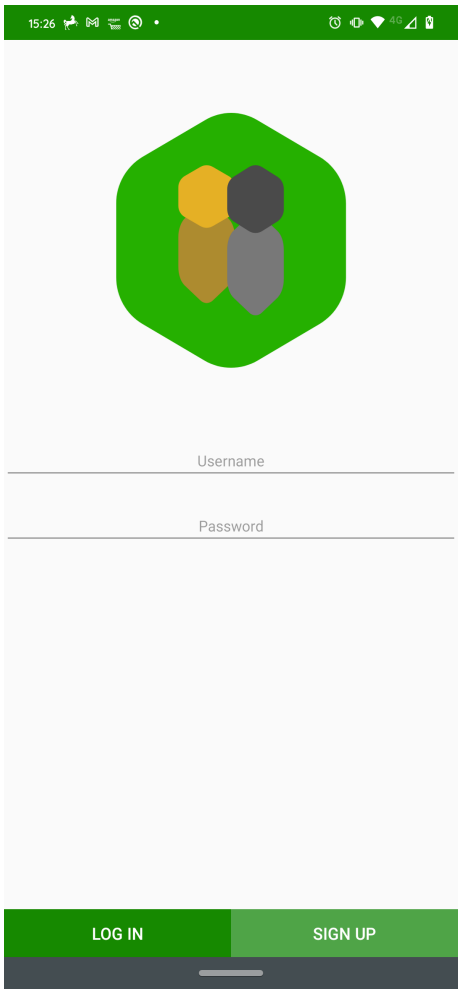
droid it is used to define a piece of utility that a user can interact with on their screen. Examples of this is 'Button', 'TextView', and 'EditText'. Without using XML there wouldn't be a front-end for the user to interact with.

Because of the large size of the project, as well as an attempt to have attractive features, the project required 22 XML files. Some of these files were used in the layouts of the fragments, where as a few of the files were used in the adapters which would inflate specific views like the RecyclerView and GridView to show multiple items of the same layout.

The XML layout files allow for the user to see other peoples profile's which directly results in the user viewing and creating their decision on whether the other user would be a viable friend.

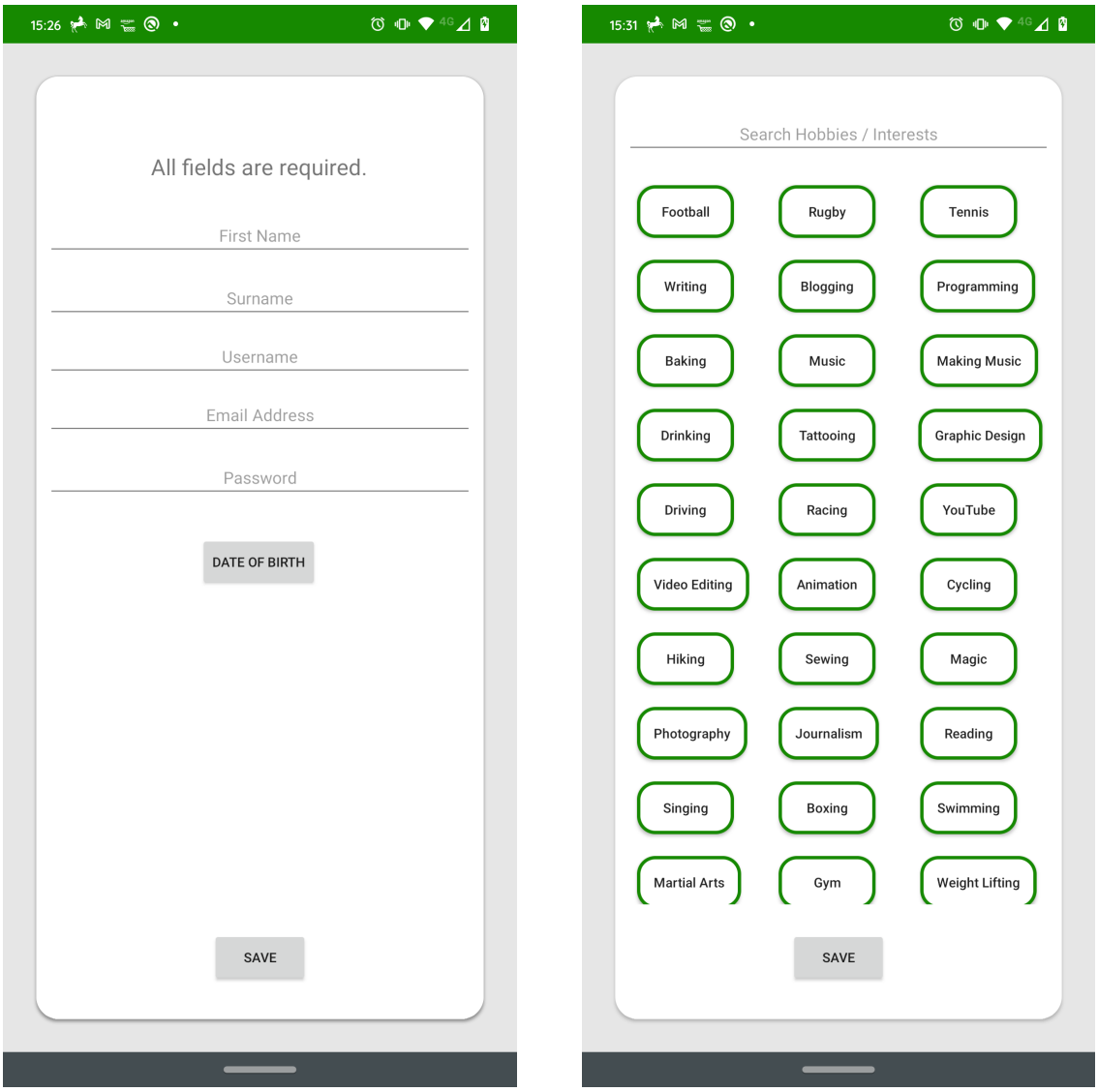
# 6 Manual

## 6.1 Log In




Type your login and password into the appropriate fields, and press *Log In*.

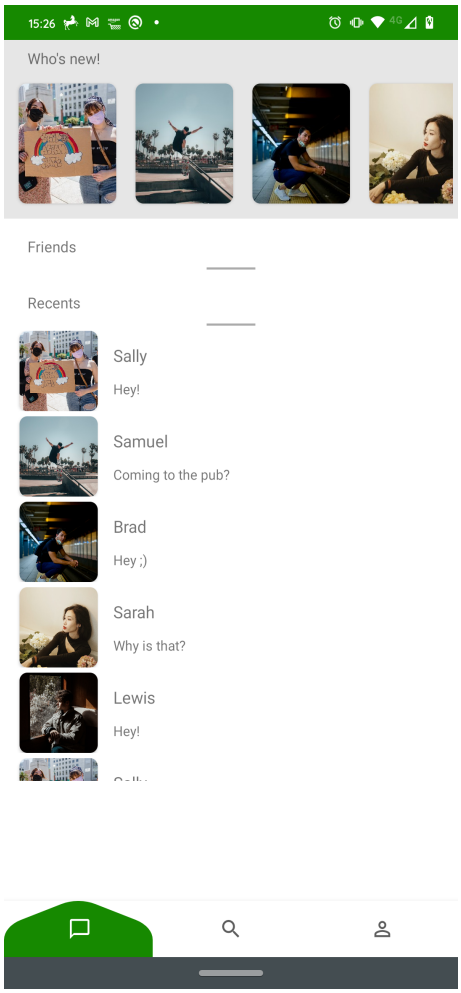
## 6.2 Register




(a) (b)

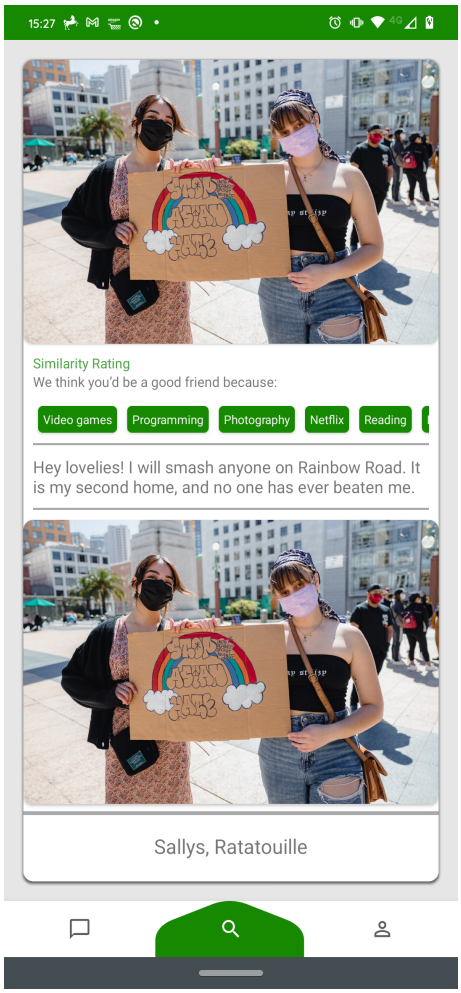
In order to register, click the Register button, then type your *First name*, *Surname*, *Username*, *Email address* and *Password* into the appropriate fields. Press *Date of Birth* to select your date of birth. Press Save. Then, select as many hobbies as you wish by clicking on them. Press save, and swipe to the left. Type in a *Bio* and press *Add Image 1* and *Add Image 2* in order to add images. Press Save.

### 6.3 Chat Tab




In order to see your friends, click on the *Chat* tab (speech bubble, the first item on the bottom), and you will be able to see your friends, contacts, and recents. In order to start a chat, click on a picture of a friend.

## 6.4 Matching Tab

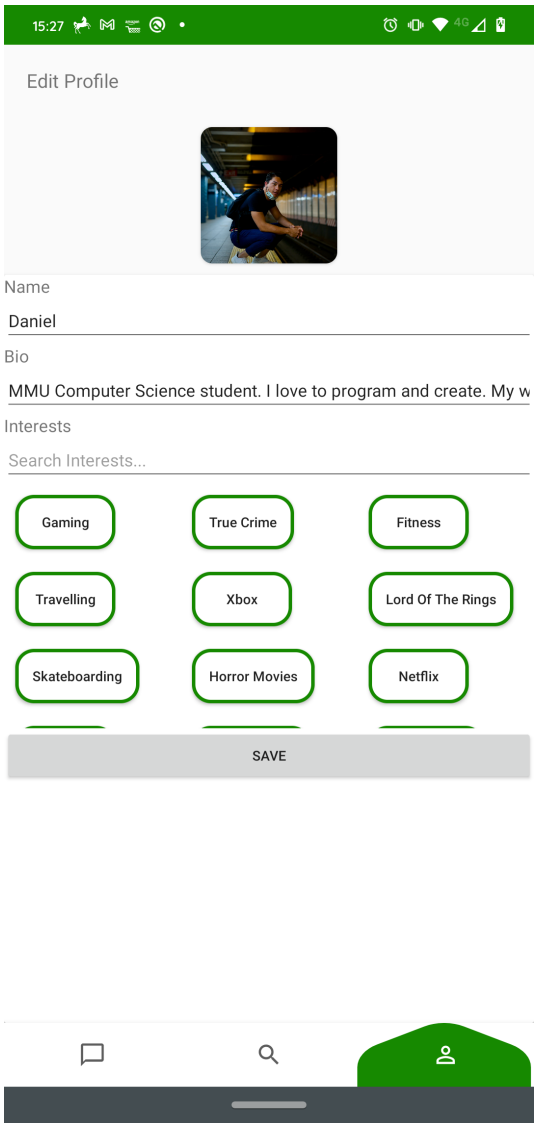



In order to see your friends, click on the *Chat* tab (speech bubble, the first item on the bottom), and you will be able to see your friends, contacts, and recents. In order to start a chat, click on a picture of a friend.

In order to be matched to new friends, press the *Match* tab (magnifying glass, middle bottom button). You will then see a series of suggested friends. In order to be matched, swipe the tab to the right. In order to reject a match, swipe left. You can do this until you run out of friends for the day to be matched with.

## 6.5 View & Edit Profile Tab


(c)




(d)

In order to view and edit your profile, click the *Profile* tab (the person icon, right bottom button). You will then be able to see your profile. In order to edit your profile, click the pencil on the top right. You can then change your *Name*, *Bio*, or your *Interests*. Click Save in order to save.

# 7 Handover

Information for future developers.

## 7.1 Source Code Management

### 7.1.1 GitHub

We decided to use GitHub for our source version management system as GitHub has a proven track record and it allows for multiple users to collaborate.

In order to do this, we set up two distinct repositories; one for the back-end, and one for the front-end. The back-end repository was handled by Emily, and the front-end was handled by Rob.

One advantage of using Git was that our team could effectively submit code to untested branches, not polluting the main, working branch, and these changes could be merged. Each *commit* required an explanation of what was changed, why it was changed, and how it helps our project.

### 7.1.2 Jenkins

In order to test our code and to see if it works and compiles, Emily set up Jenkins, an open source automation server. Unfortunately, we ran out of time and didn't manage to get JUnit testing to validate our code, but the theory would have been that each branch would go through a rigorous procedure in order to stop non-functioning code getting into main branches.

### 7.1.3 Code

As our GitHub repositories are private, our code can be found here on OneDrive: Source Code (this link is clickable)

## 7.2 If we had more time

Due to time constraints, we were not able to implement all the criteria we wanted.

### 7.2.1 Sockets

Midway through the project, we realised that network sockets would be too much work and we switched to a simple PHP server instead. However, the proper, real way of doing this would be to use a Socket/ServerSocket.

This means that all communication would be multi-threaded and allow many clients to connect at any one time. This would then pave the way to PUSH notifications. It would also mean more efficient code, rather than the mess that we have at the minute with PHP, and less resource usage overall, as you would not have to run a fully fledged HTTP server such as Apache or Nginx.

We decided to leave our socket code in for future developers, and it can be found in Appendix 9.2.

### 7.2.2 Authentication

We decided to not go with any encrypted password based authentication or security features at this time. While this is imperative to a dating app, we simply did not have enough time in order to make it work.

The way it would have worked is that it would have encoded the password using SHA256 and a salt, and stored it in the database. In order to get the password out, all we would have had to do is to ask the user for the password and then when the user types in a correct password, the salt would get added on and then the password gets encoded into SHA256. If these two hashes match, we would have a successful login. We did write the code for this, which can be found in Appendix 9.2.

Equally, we also decided not to go with TLS security at the time. This is partly because we never managed to implement sockets. We would send an HTTP request with the public certificate and the server would have the private certificate using SSLSocket, however this was far too complicated in this short amount of time to implement and ultimately not necessary in a MVP.

### 7.2.3 Chat Feature

Initially we had planned to include a text based chat feature and potentially a video chat feature however we decided that given the time-frame we were working with this feature was of a lower priority. The chatting feature of the application was a great idea, it would allow for users to communicate between one another, improving the ability to befriend others. However, this concept would be extremely time consuming and without prior knowledge or previous work completed on this topic, it would have been too time consuming.

With that said, the chat feature was not necessary in achieving our requirements. It would have been an added feature to our application on top of our main requirements. We figured that everyone already has their preferred method of communicating online and it wouldn't be sensible to try and compete with the likes of Discord, WhatsApp and Instagram, plus a chat feature is not vital to the app in order for it to function.

Above we had spoken about Socket Programming as being a way of communication between client and server. And after reading articles and watching tutorials on socket programming with Android, it revealed that it would be helpful in the implementation of the chat feature when handing it to developers in the future. Because of server-side programming languages such as Java and Python being easily portable (used on any operating system and hardware), it would be

a good solution for developing the chat feature and client server communication between multiple clients, as well as being multi-threaded allowing multiple users to use the same server.

### 7.2.4 Criteria

Criteria were also depreciated in favour of hobbies. We did this because it made more sense to base matches off of the things people like rather than physical traits, as this is something that is more typical of dating applications; this was decided after a discussion with the group in one of our group meetings. Thus, the filtering and points based system also went with this change.

# 8 Conclusion

## 8.1 Next Goals

Currently, the profiles displayed for matching are just profiles retrieved from the database. We were thinking of a feature where we would use machine learning in order to match people based off thousands of profiles, however a quick look at the ethics show that could be a really complicated rabbit hole (GDPR's automated decision making provisions, etc).

Implementing a chat feature, that could also allow for group chats, would create a greater atmosphere for befriended individuals or groups to become more acquainted. While also allowing for the possibility of university society groups to access the plethora of students who may be wanting to find a fun and socially connected society of students with similar interests. Such as a fencing society, or the Future Entrepreneurs Society.

Taking into consideration the mobile operating system market share, there is still a percentage of users who are using the iOS software and own apple devices. No matter how little iOS devices share the market according to statistics, they still share the market. Which means part of our next steps would be developing for iOS devices as well as continued development for the Android devices.

## 8.2 Closing Thoughts

Overall, we overestimated the amount of time that we had for this project. Mostly this was down to the enthusiasm of the team. Our project, while not everything was implemented as originally planned, still provides a viable product as it demonstrates the technological aspects that could be used in a friend finding app.

Provided that this app is developed further towards a finished deployable product, this app achieves the aim of friend finding by having a simple, easy to use user interface that allows people to talk to and find friends. We believe that we have made something that could be commercially viable in the future.

## References


Crossley, M. (2021), '6G5Z1108 - Professional Development: Project Options'.
**URL:** *https://moodle.mmu.ac.uk/course/view.php?id=122839*

ICO (2016), 'Guide to the General Data Protection Regulation (GDPR)'.
**URL:** *https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/lawful-basis-for-processing/consent/*

Mccorkle, B. H., Dunn, E. C., Wan, Y. M. & Gagne, C. (2009), 'Compeer friends: a qualitative study of a volunteer friendship programme for people with serious mental illness', *International Journal of Social Psychiatry* **55**(4), 291–305.
**URL:** *https://doi.org/10.1177/0020764008097090*

Office of National Statistics (2020), 'Coronavirus and loneliness, great britain: 3 april to 3 may 2020'.
**URL:** *https://www.ons.gov.uk/peoplepopulationandcommunity/wellbeing/bulletins/coronavirusandlonelinessgreatbri*

Randstad (2020), 'Student mental health report'.
**URL:** *https://www.randstad.co.uk/employers/areas-of-expertise/student-support/student-mental-health-report-2020/*

Rotem, N. & Locar, R. (2020), 'Breach exposes 100,000+ users on niche dating apps'.
**URL:** *https://www.vpnmentor.com/blog/report-dating-apps-leak/*

Statcounter, GlobalStats (2020), 'Mobile vendor market share worldwide'.
**URL:** *https://gs.statcounter.com/vendor-market-share/mobile*

The Guardian (2013), 'Tinder: the shallowest dating app ever?'.
**URL:** *https://www.theguardian.com/lifeandstyle/2013/nov/23/tinder-shallowest-dating-app-ever*

# 9 Appendix

## 9.1 Class Diagrams


Figure 1: Back-End Class Diagram (for Design) - this did not get used.

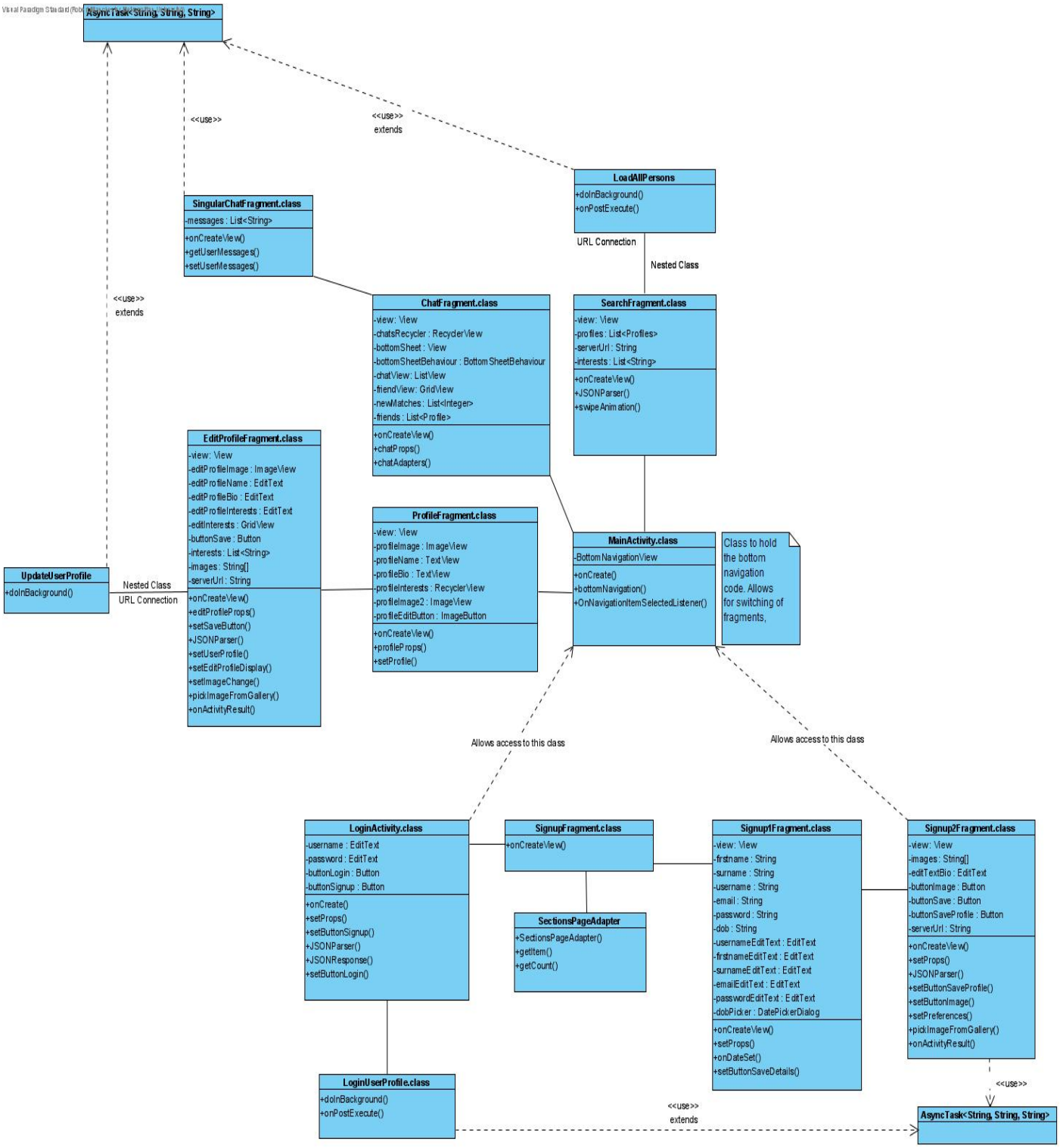



Figure 2: Front-End Class Diagram (as Implemented) - please zoom in if you can't read it.

### 9.2 Code

```
public boolean checkPassword(String password, String repeatPassword)
{
    if (password == repeatPassword)
        if (encodePasswordToSHA(password) == this.PHashedPassword)
        {
            this.authenticated = true;
            return true;
        }
    return false;
}

public boolean setPassword(String password, String repeatPassword)
{
    if (password == repeatPassword)
    {
        this.PHashedPassword = encodePasswordToSHA(password);
        return true;
    }
    return false;
    }

public String encodePasswordToSHA(String text)
{
    text += "SecureString42$";
      MessageDigest digest;
    try {
        digest = MessageDigest.getInstance("SHA-256");
          byte[] hash = digest.digest(text.getBytes(StandardCharsets.UTF_8));
          return hash.toString();
    } catch (NoSuchAlgorithmException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return null;
}
```

Figure 3: SHA256 Password Encoding

### 9.3 User Interface Mockups

#### 9.3.1 Initial Mockup

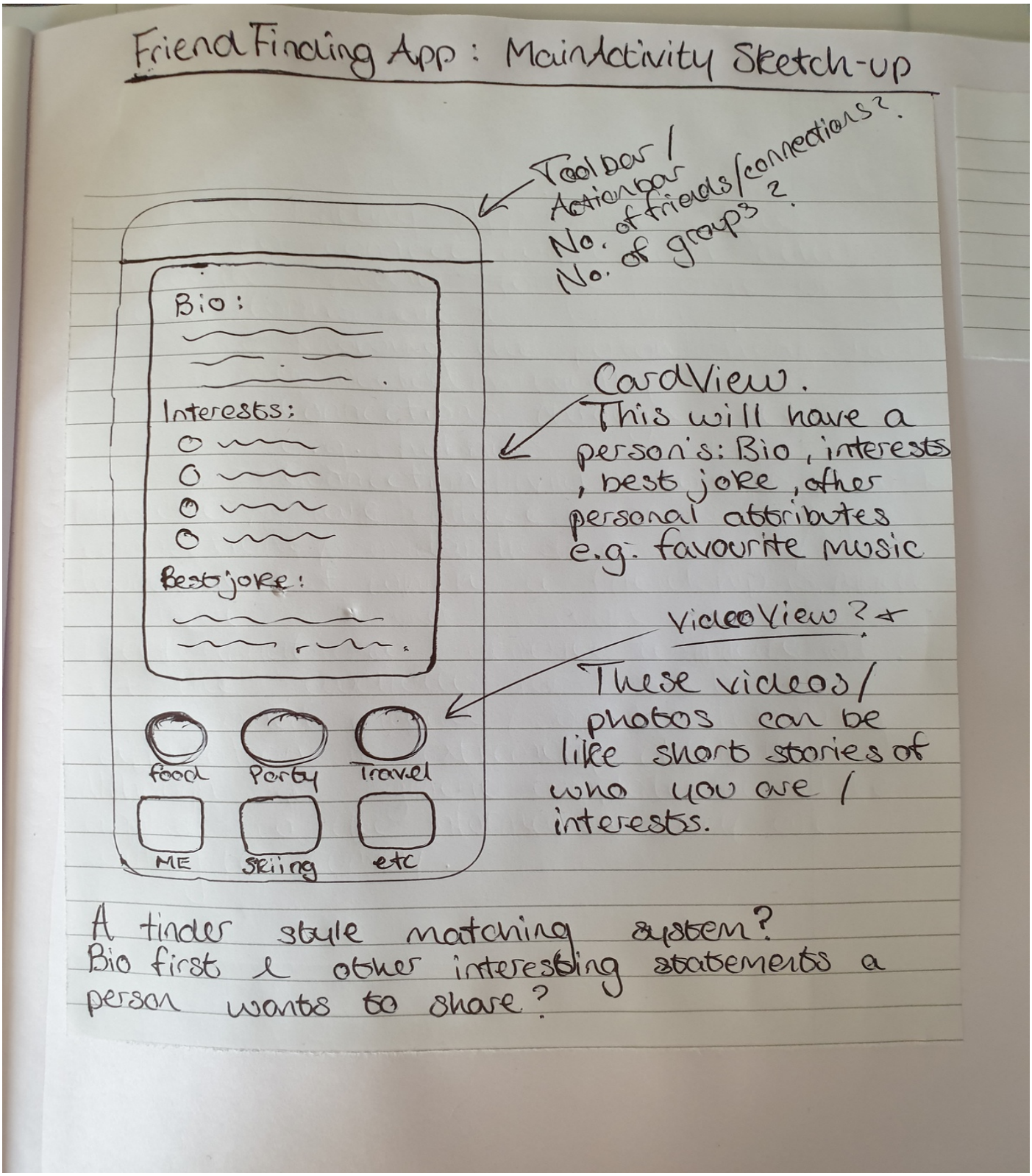



Figure 4: Initial Mockup done by Rob

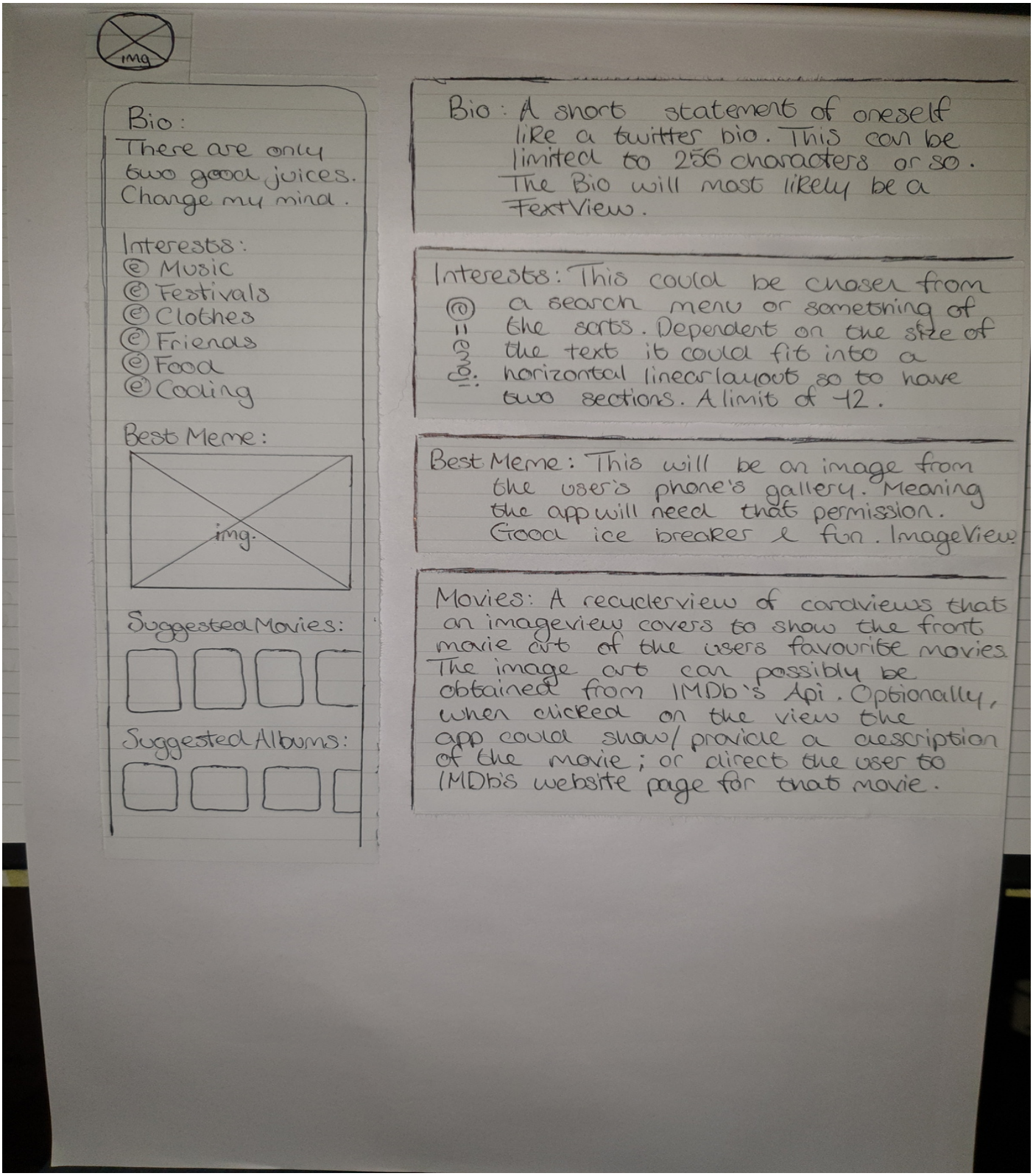
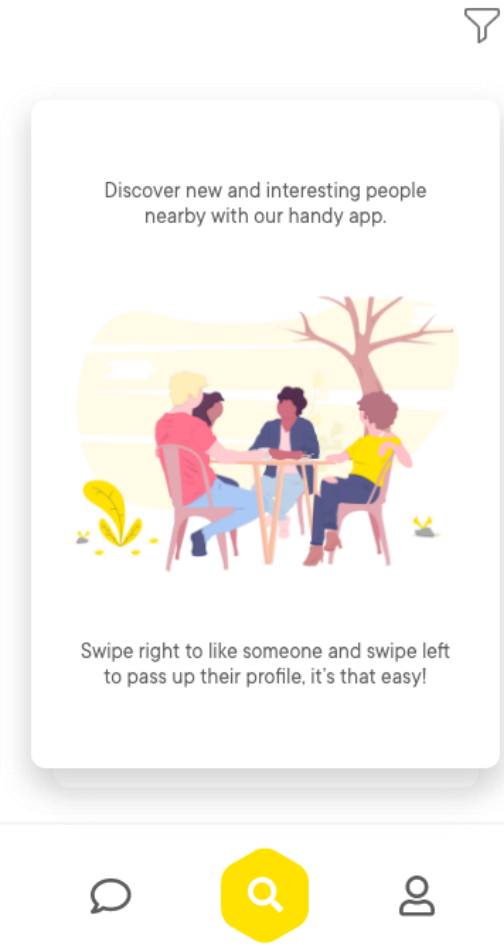
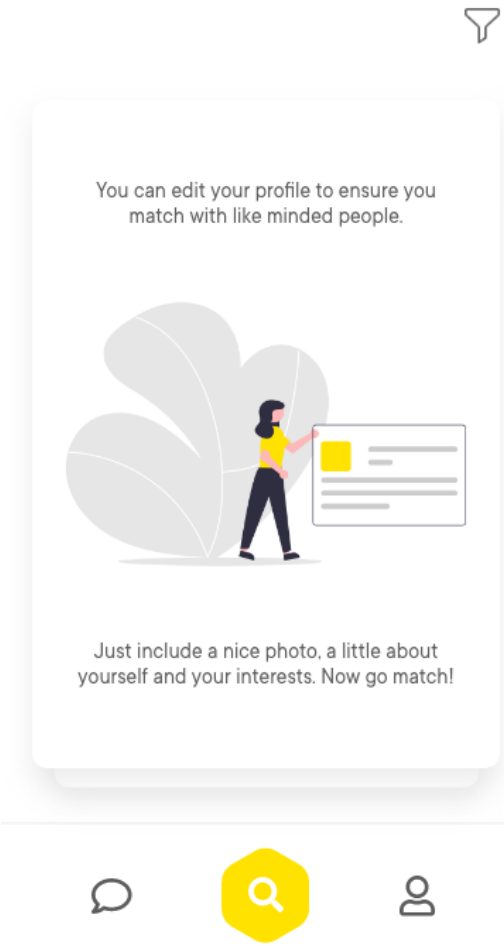



Figure 5: In-Depth Mockup done by Rob

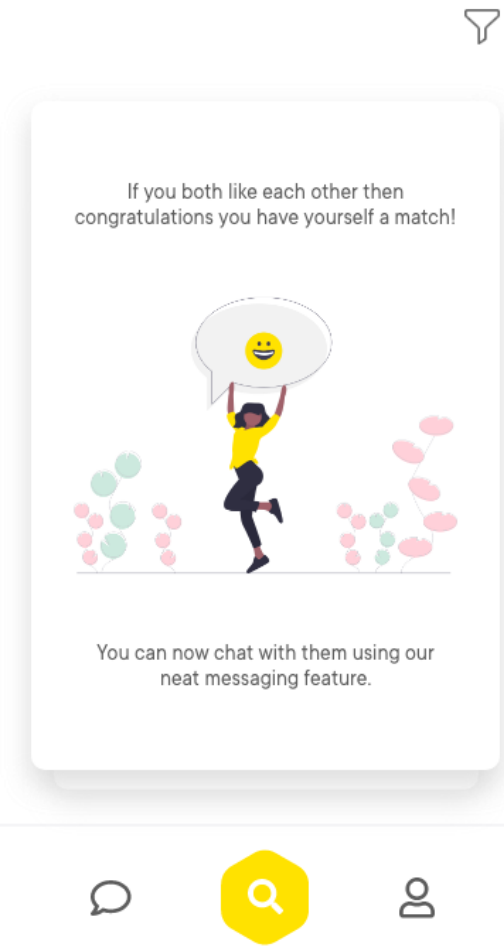
### 9.3.2 Adobe XD



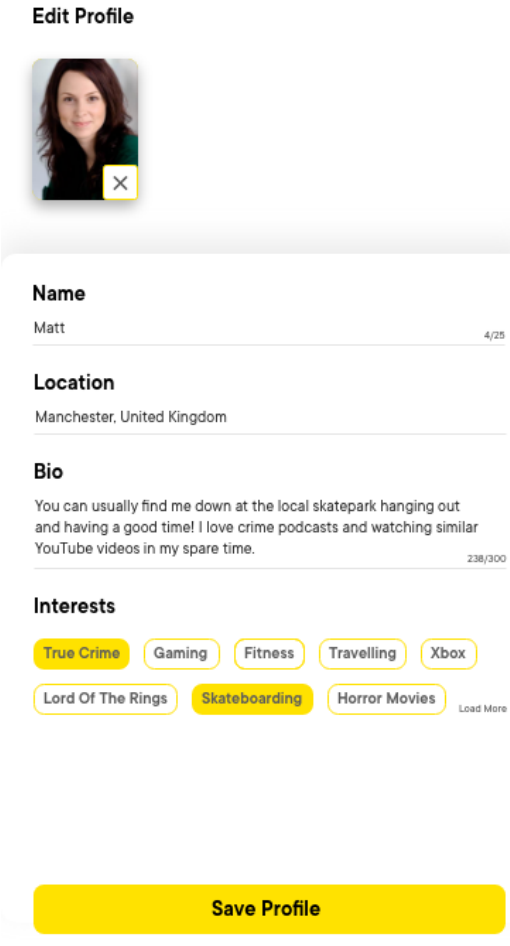

(a) Tutorial Screen 1



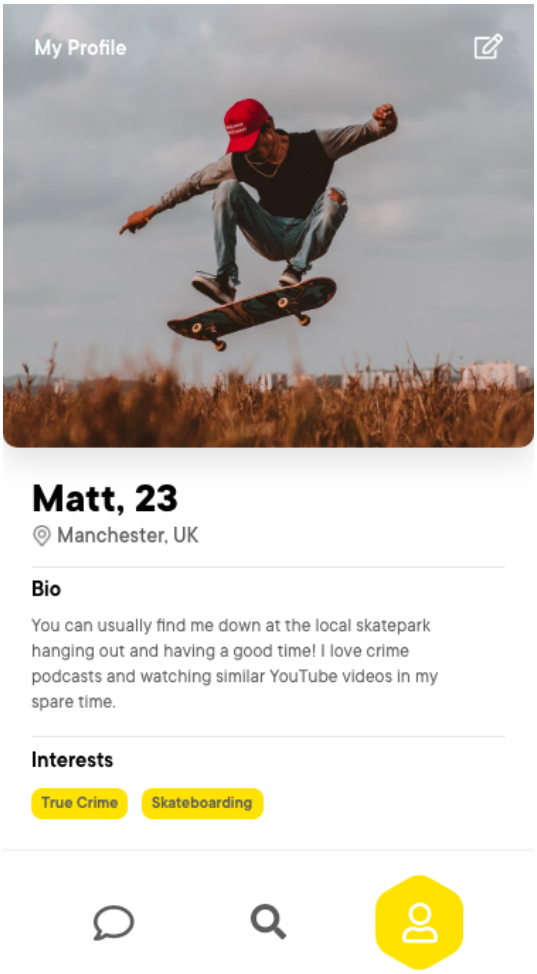

(b) Tutorial Screen 2



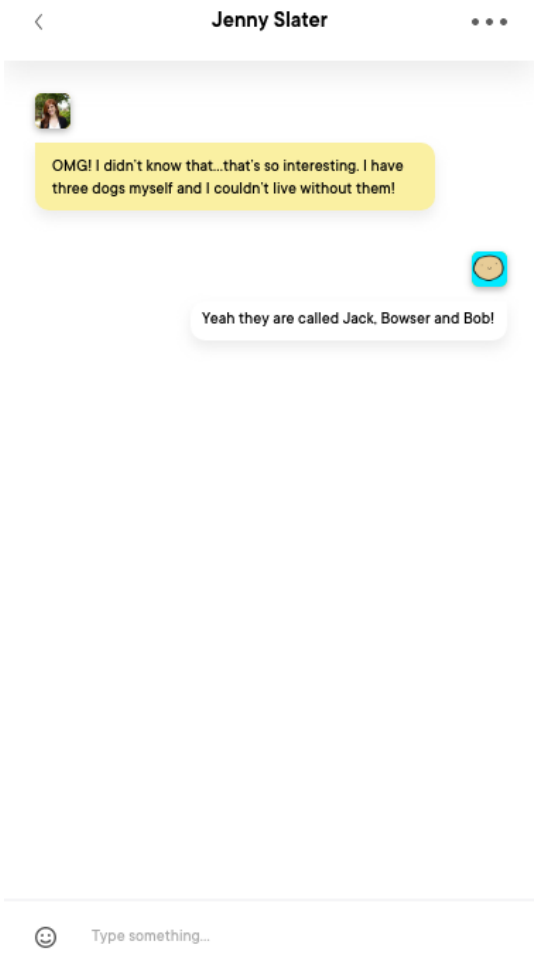

(c) Tutorial Screen 3



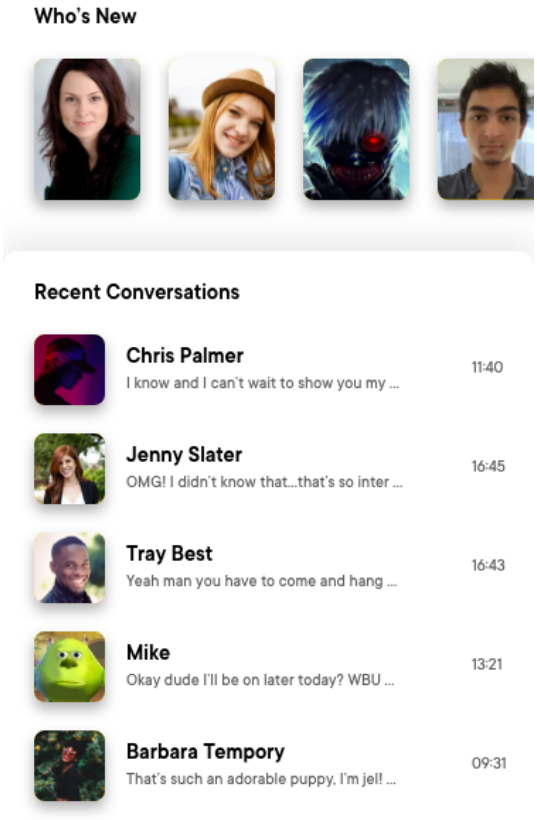

(d) Edit Profile



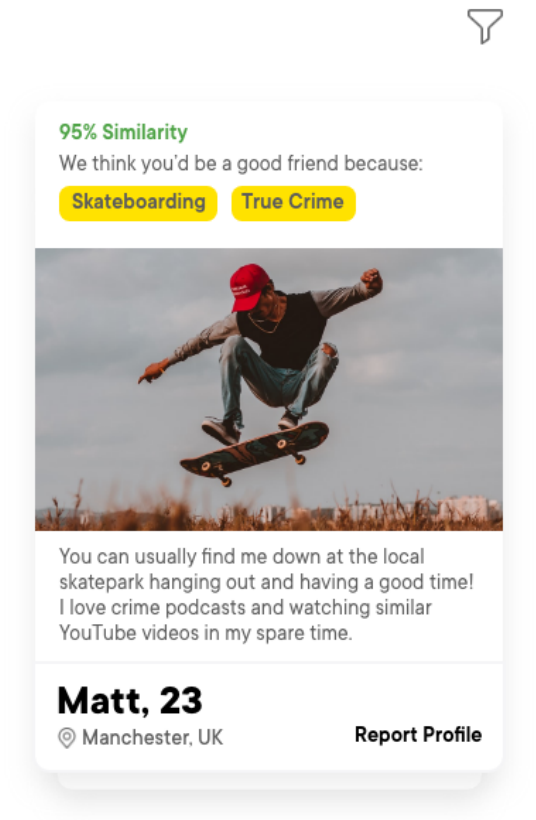

(e) Profile Screen




(f) Chat Screen




(g) Chat Conversations




(h) Profile Matching Screen

### 9.4 User Case Diagram

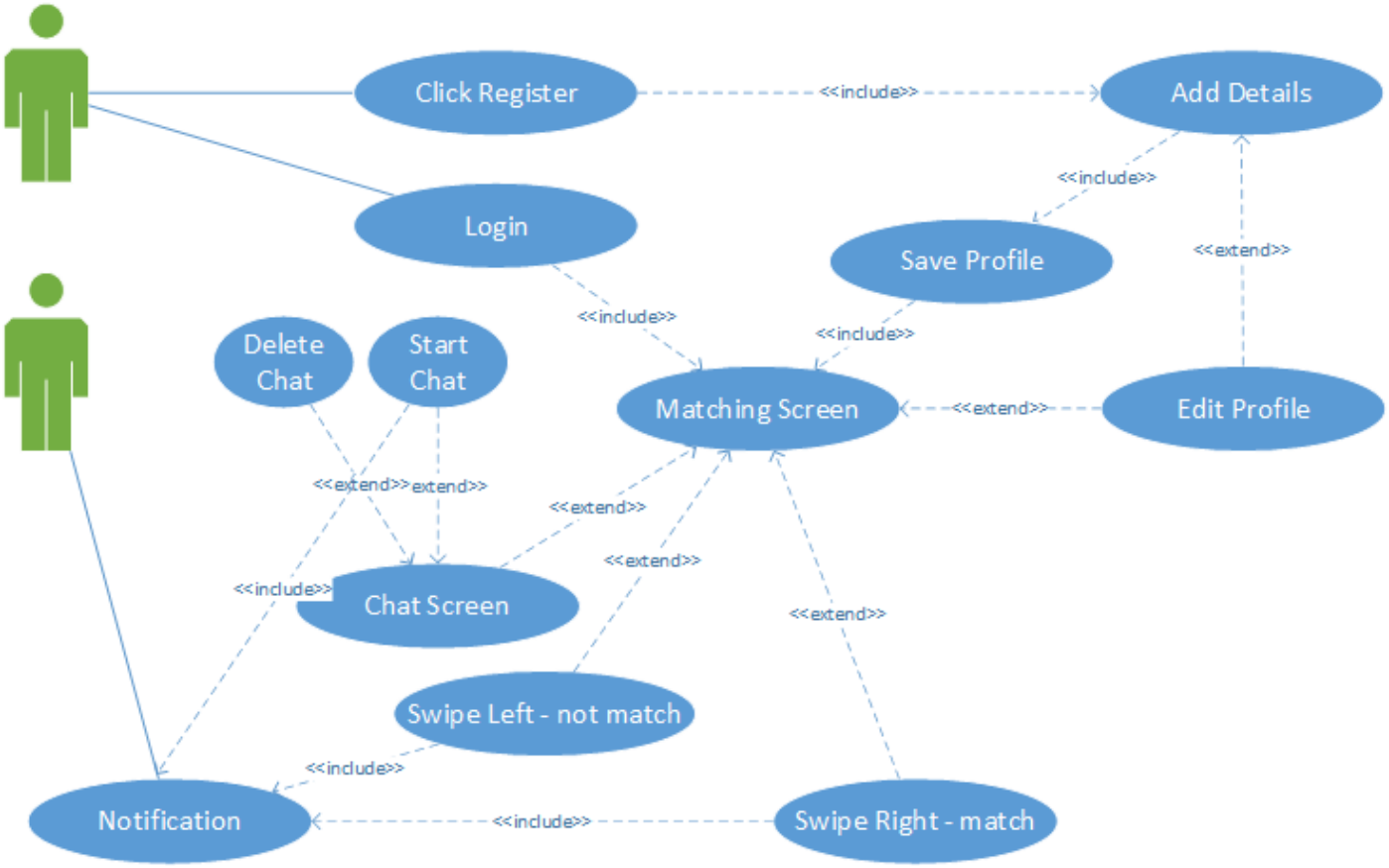



Figure 7: Diagram of Possible UCD for the App.

### 9.5 Qamar's Conversation

Emily: so an example of some requirements you may ask for "it must be easy to use" or like "it must match me with at least 5 different friends at once for me to pick from" there's really no right or wrong answer on this

Qamar: this sounds like tinder for lonely people, was that the intention

Emily: kind of so like when looking for friends what do you look for

Qamar: usually how much we can connect and what similarities we have

Emily: and with these similarities, how similar do you think a person has to be?

Qamar: similar enough where we have some things we can relate and aren't too strongly differing. like usually i feel more comfortable with someone that has the same political beliefs as me

Dan: what about area, so you can meet up?

Qamar: that could be important too but i'm not opposed to virtual friends either

Emily: what do you like doing with your friends?

Qamar: mostly hanging out like eating out together.. but going to places like the beach or similar places can be fun too

Emily: how do you find your friends normally?

Qamar: usually through different groups or [Discord] servers. there's usually a common ground in them and i meet different people in them

Bradley: Hello Qamar. I was wondering what you find is the most important factor in finding a friend (what should be weighted more). At the moment I've got stuff like similar interests (if you both like walking and horror films you may make good friends, that kind of thing) , location (are they going to be able to meet you or will it be a long distance online kinda deal) and the stuff such as age, sex, race, religion? Thanks in advance

Qamar: i think our common ground and the ways we connect are the most important factors in a friendship honestly. i've never been turned away by a friendship because of other factors but i know other people do so that'd probably be a second important thing (or first depending on the person)

Emily: how do you keep your friends?

Qamar: if we connect really well and just "click" that's usually how it keeps going. usually we need to have a level of seriousness and humor that balances each other where they're not too immature, but also not too serious or afraid to be weird at times too. i also think a lot of the time we have to share a lot of the same viewpoints but it's not something that'd break a friendship depending on what it is

Emily: how important is a person's sexuality :smile:

Qamar: i could not give a shit (am i allowed to swear?) but i would like to meet more gay people

Emily: what about what they look like?

Qamar: that also doesn't matter too much to me but i'll admit i'm more tempted to be friends with someone that actually tries to look somewhat presentable and takes care of themselves

Rob: Hello Qamar! Do you find that video or photos best describe a person and their characteristic traits? (Including their bios/others)

Qamar: sometimes, but not always. people easily fluff up photos and videos to make them look better for others to see so i've never seen them as good ways to tell their characterizes. it can maybe help with some things like explaining what things they like but maybe not too much. i think bios are better because they're shorter and to the point about what a person is like

Emily: also, qamar, if you were to find a friend, would you like an app that runs in the background and tells you about friends every so often?

Qamar: i wouldn't mind an occasional reminder but nothing overboard where it feels like i'm being hunted down and forced to find a friend. i like being encouraged to find that on my own

Rob: Qamar, what social medias do you use? And what are some simple pros and cons of each?

Qamar: the ones i use the most is twitter and discord. occasionally i might also use instagram and i only use facebook because my job forces me to. discord is pretty great with its servers and chatting and whatnot because it's so easy to connect with others, but i think that can also be a con sometimes because it's a little too easy with twitter it's easy to connect with people too but i don't think it's really a friend-making kind of site. it's more for sharing how you feel with other people but you won't have a clue who those people are - you just relate to what they say. instagram has a bit more friendship-making but i don't use it enough to have a good opinion on it

Qamar: and facebook, i won't give my opinion on that

Rob: And is it the group aspect of Discord and Twitter that you enjoy?

Qamar: yes, i do like the group aspect and how easy it is to find people that are similar to you. like if you want to find a server that's dedicated to just food, you'll find it

Rob: Would you prefer to read a bio before or after seeing what a person walks, talks and looks like, or beforehand?

Qamar: probably before honestly. it gives you an idea of who the person is but enough where it might turn you away from a person. i feel like seeing the bio after is a bit odd since the goal is to help you understand the most important details about them before getting to know them

Bradley: One more quick question: what information do you think is acceptable to be on your profile. I think name, photos, bio are pretty much required but do you think that things such as your age and your location is suitable for a public profile or do you feel it oversteps the mark? :)

Qamar: name, photos, bio are all good. age possibly depending on the site and safety concerns and maybe country but i don't think more than that. i also think gender should be shared too but school, more location detail, birthday, not really needed. that's a bit too personal to be seen publicly