

Chatfer



An app to connect universities' students together

Independent Project
Jørgen Fosseng Duley
Spring 2017

Supervisor
Prof. David Rossiter

This work was jointly done with **Benson Ng** as well as server-side development by **Tijmen van Dien**

Contents

Abstract	2
1. Introduction	2
2. Chatfer	3
3. Design	4
4. Features and Implementation	5
4.1 Frontend (Xamarin (Mono))	5
4.2 Backend (Microsoft Azure (.Net Core))	5
4.3 Registration	5
4.4 People Page (Landing Page)	6
4.5 Chat	7
4.6 Multi-Text Chat	8
4.7 Interests (Categories)	9
4.8 Summary	9
5. Conclusion	10
5.1 Results from signup form	10
5.2 Closing Remarks	11
6. Appendix	12
6.1 Minutes of the 1st meeting	12
6.2 Minutes of the 2nd meeting	12
6.3 Minutes of the 3rd meeting	13
6.4 Minutes of the 4th meeting	13
7. References	14

Abstract

Social network changes our lives, we are somehow addicted to the media. Everyday we are using our smartphone or computer, seeking news or updates from friends/family. It has become an inevitable part of our life. Sadly, these kind of interaction does not only bring prosperity to our generation.

Nowadays, youngsters fear physical socialising when dealing with direct interaction among people. Hence, the paradox emerges; if technology brings benefits for convenient and wider social connectivity, why is it such an issue to draw people from the virtual world back into reality? Can we reverse this rising trend?

I. Introduction

As a believer in direct interaction with people, I would rather have a phone call than messaging. One day, my classmate Benson was eating alone in the school canteen, he looked around and saw multiple people eating alone, staring at their phone. Ironically, he thought, these people might have a great opportunity to be friends, putting down their phone and talk with each other. He forwarded this observation to me, and it struck me that I did not pay attention to this important detail before. It got me intrigued.

Social barriers make it strange for people to say hello to each other. People are complaining about how difficult and awkward it is to meet new friends in HKUST. However, interpersonal connections through internet, removes this immediate obstacle. Ergo, this would allow for a less troublesome first encounter among each other, especially introvert students. Not only meeting friends, some would want a person to share their interest with, which most of his or her friends does not share. Additionally, some people would like to achieve something and want to find his companions.

Having an app to put the burden of first encounter is not something new. There are multiple examples out there, like Tinder, Match, Facebook, and so on and so forth. However, some of these apps rely on dating, or require a number of personal information. What if someone just wants a simple, convenient and fast way to meet new people whom also shares their interest?

With the above thinking in mind, the idea of **Chatfer** transpired.

2. Chatfer

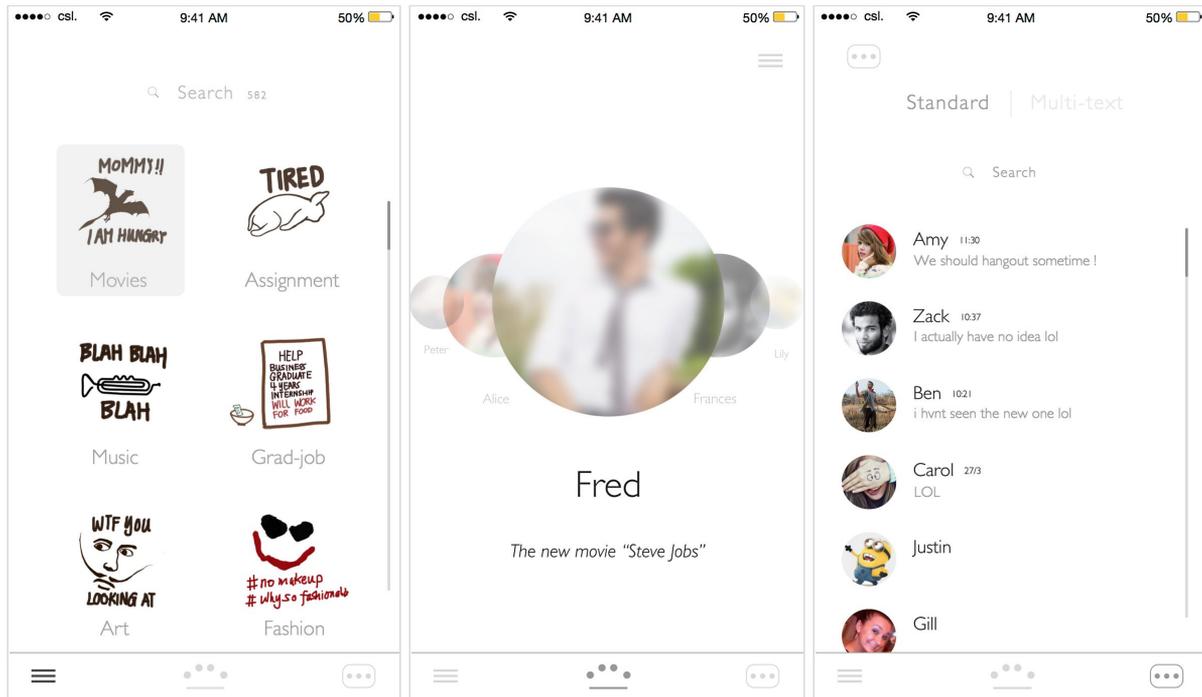
Chatfer is an app aiming to bring an interpersonal connection to people with similar interests and goals together. Our vision is not only about finding friends, we want Chatfer to be an app that people can use for varieties of reasons. For example, a student who wanted to try to build an app can use the app to find a partner who is excel in programming; a student who wants to make a short film can use the app to search for cameramen/director; a student who wants to consult a senior student for advice in a certain company he or she had done internship in, and so on.



The target audience is initially meant for student, although we hope that the app can be used among the public as to increase the diversity for the users. In short, our vision is that Chatfer can be an app that effectively changes the way people connect.

3. Design

The design for the app is minimalistic, as we understand that apps nowadays are not as much attractive as before. A streamlined, minimal and intuitive app will grant a more enjoyable experience for the user. The following will show the user interface design for the app.



The main page for the app (center) contains the function for searching potential students with similar interest. Users can view other students by swiping. The users profile pictures are blurred because we do not want people to connect only because of appearance. Once a user connects with another user, the blurring function will decay in a rate that the nearer to the meet-up, the clearer the profile picture will be.

The category page (left) allows users to choose different kinds of categories they enjoy or find interesting. The category function can help users to find better matching results with other students.

The messaging function (right) is basically the same as other apps, it is made for users to communicate before their meet-up, to locate and get to know each other. Yet we have added an exciting function. As we understand first message is usually awkward and people are often too nervous on the opening line. Hence, we will create a break-the-ice message generated from our database which contain thousands of funny text messages. We would like to use this function to break the ice and give users a enjoyable and funny experience to start off their conversation with their connected person.

4. Features and Implementation

As already mentioned in the design, the app features several social functions. For this to work seamlessly and with high performance cross-platform, the app is relying in several technologies. I will go into detail for the most important here.

4.1 Frontend (Xamarin (Mono))

The foundation of the app is built using the Xamarin framework for the frontend mobile development. The main benefit achieved is cross platform native performance on iOS, Android and Unified Windows platform. This is all done using .NET and C# shared codebase. Hence the logic for the app only needs to be written once. Everything in the app, including the visual elements is coded using C# and as much shared code as possible.

As Xamarin is relying on standard visual elements, to achieve our design choices, there had to be written specific native iOS and Android code using Xamarin.iOS and Xamarin.Droid. Both is still written in C#, but does only affect the respective platform. Specifically, the use of these in Chatfer, is as custom renderer.

4.2 Backend (Microsoft Azure (.Net Core))

The backend runs on Azure using the Dotnet core framework with an MSSQL database for data storage. The Azure Notification hub is used to send push notifications to all users. Using a full-Microsoft stack for development has many advantages that allow all services to integrate very well.

The backend of the project is designed to run as efficiently as possible on a system that scales very well. Using the *hamming distance* between two users to calculate matches allows us to compare 50.000 users in about half a second on the cheapest tier of the Microsoft Azure cloud.

It should be noted that the backend side is almost solely developed by Tijmen van Dien.

4.3 Registration

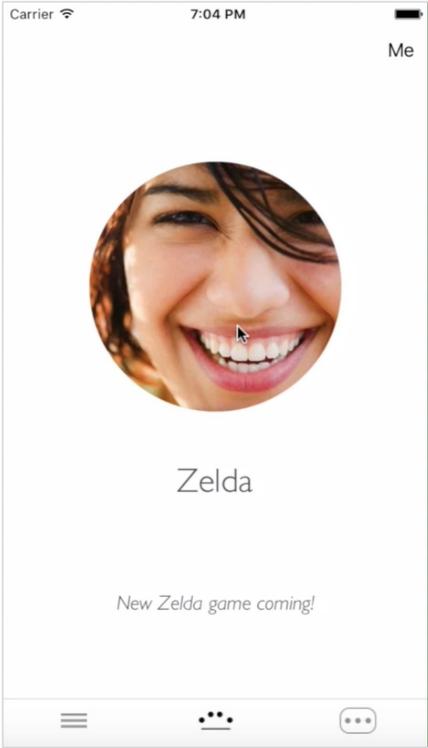
When the app launches, the app will look up the unique device ID which will be used as the unique identifier for the user. The main benefit of this, is there is no need to have a password, as well as it is impossible to have more than one account. It is similar to how WhatsApp works. The server (which resides in Netherlands), returns a unique token based on the device ID which will be used throughout each session. Every communication is done through asynchronous secure HTTPS calls over the internet.

Following the creation of an account in the first run, the user is asked to select or take a photo. The photo is then automatically compressed and converted into a byte stream to avoid heavy server load. When the user is presented the chosen photo, which has been applied a circle transformation, the user will be shown a simple slider which applies a blurring transformation corresponding to the slider value. The user will then continue and the photo will be uploaded to the server as a byte stream using a HttpClient with a JSON header as a string content.

The following steps are adding the user name, status, interests and email address. All the steps in the registration phase would not have been possible without a number of open source libraries (Json.NET, FFImageLoading, etc.) provided as NuGet packages for Xamarin.

4.4 People Page (Landing Page)

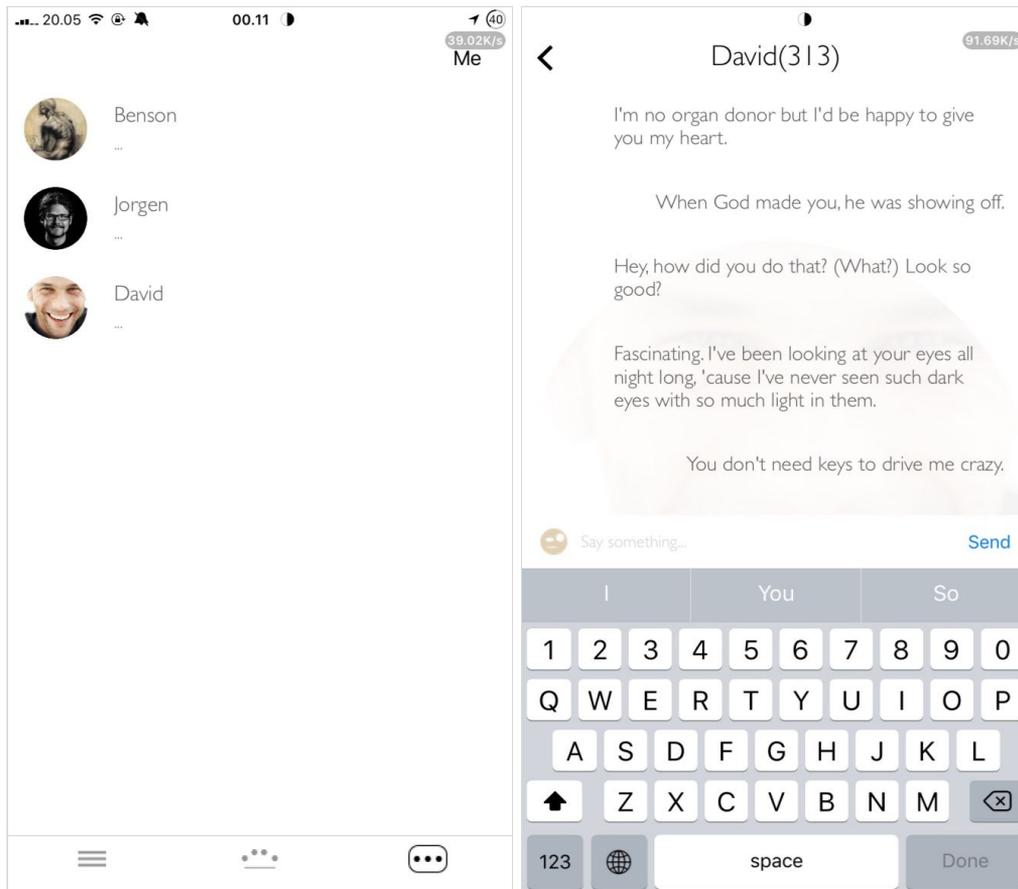
Once the registration is completed, the user will be shown the users based on the interests chosen in the registration. The results are returned based on a ranking scale. The users whom have the most similar interests, are given a rank of 0 and is presented first. The users with less matching interests are then given a lower (higher in terms of numbers) ranking. 10 users are being returned by the server each time the user lands on this page (People Page).



Due to limitations in visual design provided by Xamarin, the way to see different users is simply by swiping the the right, which is achieved using a Carousel Page, with each user as a child Content Page. Every user is clickable, allowing the user to advance into the user profile to see their interest and choose to send a coffee or meal request. There will also be some interesting and funny guidance to get a user starting a conversation after the first connection.

4.5 Chat

The chat is mostly like any other chat client out there. And this is also where Xamarin showed its limitations in a greater scale. The chat is implemented using a list of child views with separate templates for receiving and sending messages. The tricky part is handling the view when the keyboard is showing on the screen. The ideal way is the entire view to move up, while having the entry field to be pushed up accordingly.



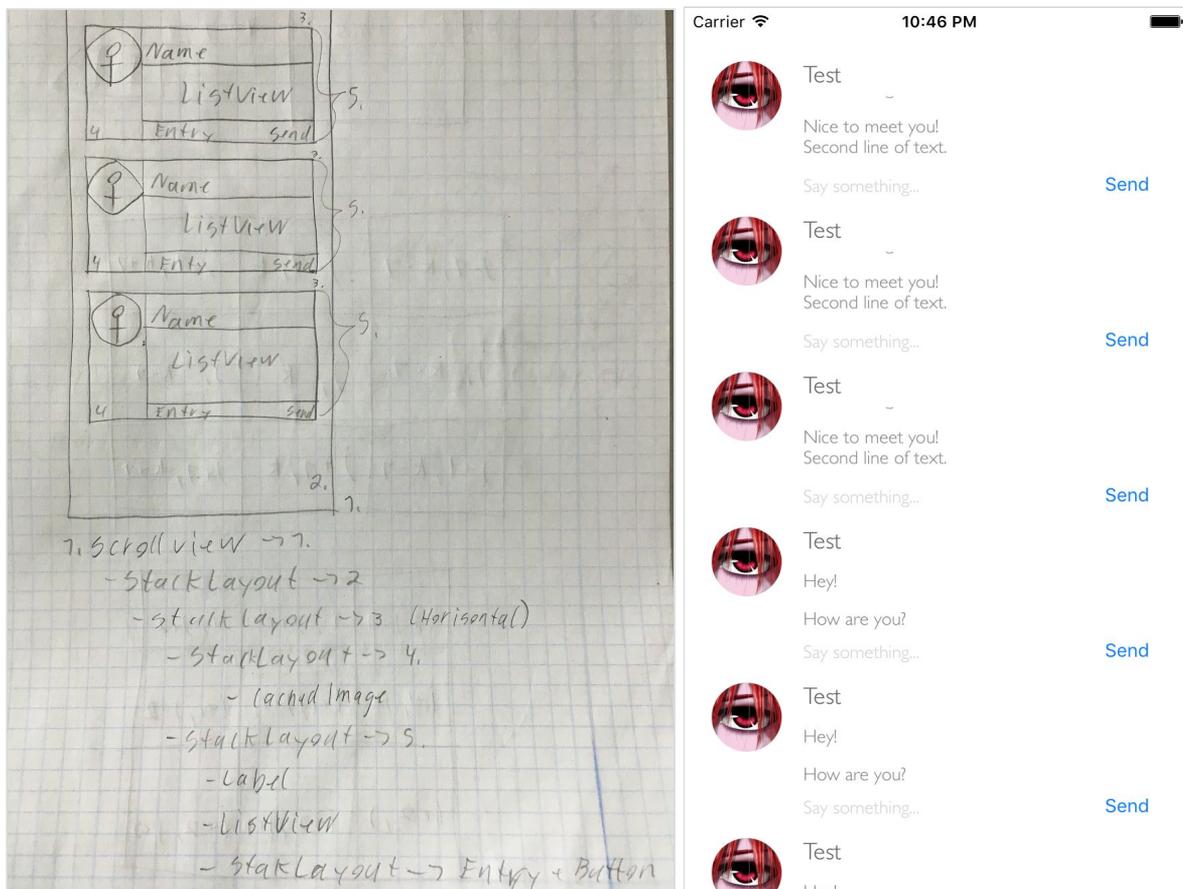
However, the only way to achieve this, is by having the entry field inside a scrolling view, as to scroll/push/move it up when the keyboard is shown. This means there needs to be a parent scrollview who has a child of views inside a list that is also scrollable. The result is not optimal, as this pushes both views up. Also, when scrolling through messages, you are scrolling through two views which inherit scrolling. This causes some visual artifacts and partly ruins the streamlined experience.

Aside from this issue, the chat is working decently.

4.6 Multi-Text Chat

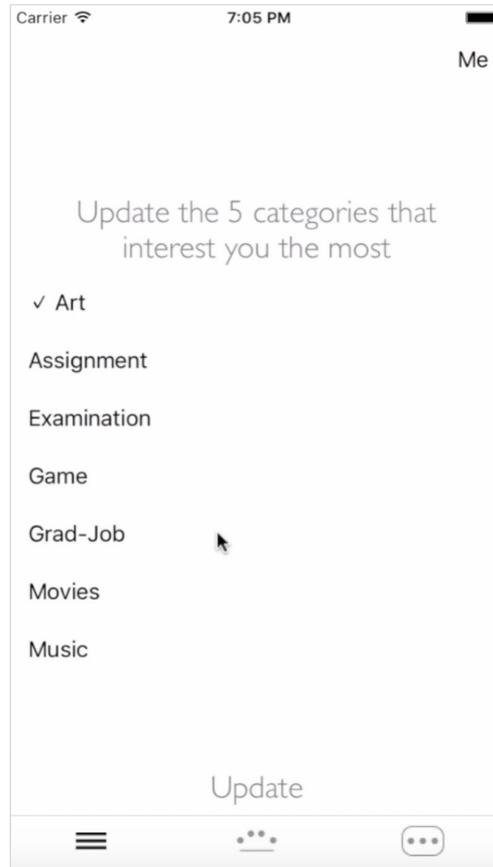
Apart from the standard chat, I experimented with implementing a new chat function that we called the Multi-Text chat. It's different from the standard chat functions in most app in the way that every chat is available in one page. Meaning there is no need to press into the specific chat to communicate.

Messages can be sent to multiple recipient quickly using this approach. Interestingly, I had less difficulties implementing this than the normal chat as well as fewer bugs. The sketch for this and the actual way I followed for implementation can be seen below.



4.7 Interests (Categories)

The final page is the Interest Page that allows the user to update its interests as mentioned in the design. Ideally, the categories should follow a flowlist design, but due to several technical difficulties, I settled on a more simple and workable approach, using a listview with strings.



4.8 Summary

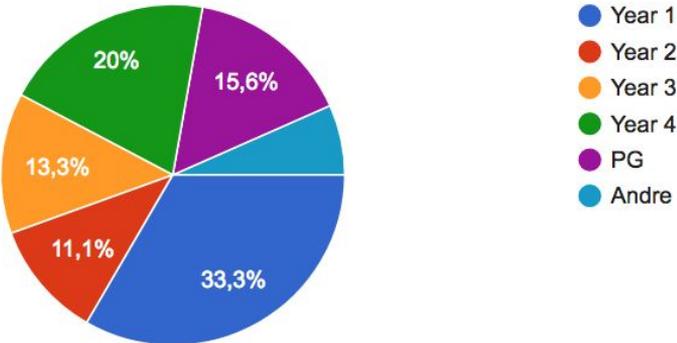
In general, Chatfer has been designed and implemented from ground up using professional sketching tools, multiple drawings and testing of several implemented prototype designs to attain a streamlined and intuitive user experience with native platform performance.

5. Conclusion

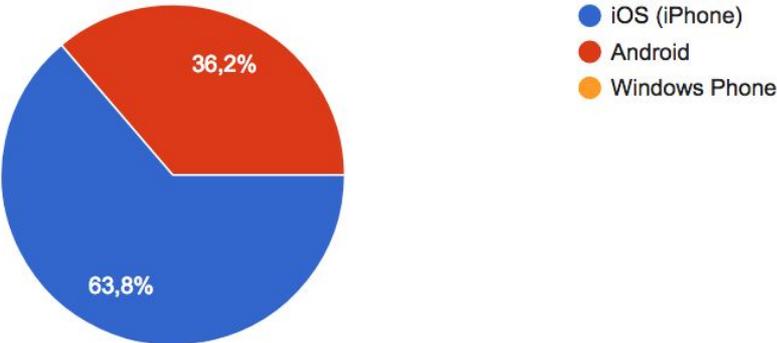
5.1 Results from signup form

We put up posters at random locations in the HKUST campus, including the canteen in LG7 and LG1. In just under a month, we had more than 50 signups, and this gave us some interesting data as can be seen below:

Which year are you in?



Which platform are you using?



As can be seen from the sign-ups above, the majority of interested students are either year 4 (33.3%) and year 1 (20%). As well as mostly iOS (iPhone) users (63.8%), and no Windows Phone users. Even though Chatfer is also deployable on unified Windows platforms due to the .Net support, it seems unlikely that there will be demand for a Windows Phone version.

5.2 Closing Remarks

Chatfer has been a major learning experience on multiple levels. On the technical side, although there might have been time savings on implementing Chatfer using a cross-platform approach, it also presented some extra work. The most time-consuming part has been by far to implement all the visual elements and animations. Making this work cross-platform required a lot of research and eventually lead to undesirable workarounds as well as having to write native code for both, including shared code, which arguably caused more work than to simply write specifically for each platform outside of Xamarin.

Although it is helpful to write native code in Xamarin, you are left out on the open source libraries available for each platform, which for instance would have been helpful in both the implementation of the interest page, people page and the chat.

The takeaway message, is that Xamarin and cross-platform solutions are excellent for enterprise apps or apps with standard design patterns. If I had to redo the project, I would have started writing Swift code in Xcode and launched the iOS version first, as the greatest user base is residing here (from the signup form).

Aside from the above mentioned, Chatfer has also required me to understand the target audience and the difficulties related to it. We have received multiple possible feedback on the idea and the app itself from early testers of the app, and we hope to launch it outside of a closed testing environment. Reaching the Microsoft Imagine Cup National Final in Hong Kong also served as a motivational boost.

Ultimately, we hope that Chatfer can ease the life of students.

6. Appendix

6.1 Minutes of the 1st meeting

Date: Monday, 6 Feb 2017

Time: 10:45 am

Place: Room 3512

Attending: Prof. Rossiter, Jørgen Duley, Benson Ng

Absent: None

Recorder: Jørgen Duley, Benson Ng

Approval of minutes

The minutes of the last meeting were approved without amendment.

Report on Progress

Showing of the idea and design/implementation choices

Discussion Items and Things to Do

Design changes and further implementation of prototype

Meeting adjournment

The meeting was adjourned at 11:15am

6.2 Minutes of the 2nd meeting

Date: Wednesday, 15 Feb 2017

Time: 5:00 pm

Place: Room 3512

Attending: Prof. Rossiter, Jørgen Duley, Benson Ng

Absent: None

Recorder: Jørgen Duley, Benson Ng

Approval of minutes

The minutes of the last meeting were approved without amendment.

Report on Progress

Progress on developing cross-platform elements, in particular photo capturing and selection, as well as handling client-server communication.

Discussion Items and Things to Do

User-selected blurring and artistic effects (i.e. cartoonization) on photos to preserve anonymity. After two users connect, the blurring should gradually go away, as to let the users see whom they have connected to once the meeting time approaches.

Meeting adjournment

The meeting was adjourned at 5:30pm

6.3 Minutes of the 3rd meeting

Date: Friday, 17 Mar 2017

Time: 4:30 pm

Place: Room 3512

Attending: Prof. Rossiter, Jørgen Duley, Benson Ng

Absent: None

Recorder: Jørgen Duley, Benson Ng

Approval of minutes

The minutes of the last meeting were approved without amendment.

Report on Progress

Working prototype. Main focus on iOS platform, as Android caused several development issues using a cross-platform approach. Server-side, using Microsoft Azure, mostly finished, with scalable matching architecture relying on *hamming distance* on interest match.

Discussion Items and Things to Do

Simplification and better user interaction for improved intuition. Focus on getting user feedback.

Meeting adjournment

The meeting was adjourned at 5:00pm

6.4 Minutes of the 4th meeting

Date: Wednesday, 19 Apr 2017

Time: 4:00pm

Place: Room 3512

Attending: Prof. Rossiter, Jørgen Duley, Benson Ng

Absent: None

Recorder: Jørgen Duley, Benson Ng

Approval of minutes

The minutes of the last meeting were approved without amendment.

Report on Progress

Reached the national final in the Microsoft Imagine Cup 2017. Great progress on the app on all aspects. Memorystream issues on Android platform on certain devices. Issues in push notification and visual elements in chat on both devices. Experimental multi-text chat has been successfully implemented, but not deployed.

Discussion Items and Things to Do

Prepared presentation to use in the Microsoft Imagine Cup 2017 National Final. Received valuable continuous feedback used to improve the presentation as well as aspects of the app.

Meeting adjournment

The meeting was adjourned at 6:00pm

7. References

<https://compete.imagine.microsoft.com/en-us/team/117018>
<https://developer.xamarin.com/guides/xamarin-forms/application-fundamentals/navigation/carousel-page/>
<https://developer.xamarin.com/api/type/Xamarin.Forms.ContentPage/>
<https://developer.xamarin.com/guides/xamarin-forms/user-interface/layouts/scroll-view/>
<https://developer.xamarin.com/guides/xamarin-forms/user-interface/listview/>
<https://developer.xamarin.com/guides/xamarin-forms/application-fundamentals/custom-renderer/>
<https://github.com/luberda-molinet/FFImageLoading>
<https://github.com/aritchie/deviceinfo>
<https://github.com/daniel-luberda/DLToolkit.Forms.Controls/tree/master/FlowLayoutView>
<https://www.xamarin.com/>
<https://azure.microsoft.com/en-gb/>
<https://www.microsoft.com/net/>
<https://www.microsoft.com/net/core>
https://en.wikipedia.org/wiki/Hamming_distance
<https://www.sketchapp.com/>
<https://www.nuget.org/>
<http://www.newtonsoft.com/json>