Title: A cross-platform mobile application for schedule-based matchmaking

Author: Ng Helen Hoi Ling 3035269629
Wong Tin Chun Jeremy 3035298151
1. Project Background

People has always been looking for ways to help them find love and companionship. Matchmaking applications are the latest manifestation of human beings doing what our ancestors have been doing. With a smartphone in possession, it is sensible for us to create an online profile, in hope of finding new friends and relationships. Online matchmaking websites are very popular nowadays, with an estimation of over 200 million active users worldwide[1] and generating revenue which amounts to USD $1,383 million dollars in 2018 [2].

Existing matchmaking applications are time consuming with a low success rate. In 2012, Tinder, an online dating application was created. It is one of the most successful matchmaking applications. Tinder created a social phenomenon of swiping left and right to indicate whether the user would to be paired with this person. While Tinder is proven to be successful, it focuses a lot on the user’s profile and pictures instead of actually getting people to meet one and another in real life. An average of 3,000 swipes [3] are required for a user to successfully match up then went out to meet with one another. Three thousand swipes, at two seconds per swipe, translates to a solid one hour and 40 minutes of swiping to go on a single date. This is time consuming and would take a significant more amount of time if the user would like to look at different users’ profiles.

Matchmaking applications should focus more on getting people to meet each other in real life. The problem with existing applications is that they only provide a chat box for both users to chat and discuss a time for them to go out. Some people have a regular schedule or are busy even after work. It may be difficult for them to find a date which both of them are free. For example, Bumble is a social application which only female users can make the first contact with matched male users. This application also requires users to upload a selfie to ensure the users are not using a fake photo. These measures make the application more secure and increases its success rate. Bumble’s primary goal is to help users find “empowered and lasting connections”, however, only 25% of users went on a date with someone they met on the application last month[4]. This implies that 75% of users only chat with their matched users on the application and did not see them in real life. Matchmaking applications should facilitate users to meet up with one another.

The two main reasons to make our schedule-based matchmaking application is to optimise matchmaking results to make the process less time consuming and increasing the meet-up rate. We would like to make a matchmaking application that is not solely for dating but mainly to provide a platform for people to make new friends outside of their social circle. There are many situations where we would like to have a friend accompany us to achieve a goal within our schedule. It could be finding a friend to have lunch with during the one hour break or someone to do exercise with after work.
We hope that this application could encourage users to go offline and have conversations with people in real life. As the matchmaking process is based on users’ schedule, they would finally be able to go out with their matched users without changing their original plans or the fuss to come up with a date and time which both of them are free. This application is desirable as it minimizes time for users to pick and choose.


2. Project Objective

In this project, we want to achieve the followings:

1. To develop a mobile application which:

   1.1 **The mobile application supports both iOS and Android operating system**

   According to a research, in 2018, iOS has 51.61% market share and Android has 47.93%, which they have a combination of 99.54% market share in Hong Kong. That means, if we have a mobile application that support both platform, majority smartphone user in Hong Kong can use our app.

   1.2 **The mobile application serves a schedule-based matchmaking function**

   A schedule-based matchmaking is defined as a matchmaking application which use a user defined schedule as a major parameter of the matchmaking algorithm. The algorithm shall pair two candidates whilst both of their available time slots are matchable and they should have a nearby physical location.

   The general flow of the application is planned as the following:
   1) Users submit their schedules to server
   2) Server receives schedules and do the pairing
   3) When there is a pairing result, notify the user

   1.3 **The mobile application collects user’s feedback and information**

   The application shall collect feedbacks after each matchmaking event. Feedbacks will be in a format of text comment and a rating, these data will further analyze to improve the quality of the matchmaking. For instance, if we analyze that matchmaking pair of a engineering student and business student often have a higher rating, we then adjust the algorithm so that engineering student will be more likely to match with business student.

2. To study the relationship between user’s schedule and their corresponding behaviour

   2.1 **Motivation**
In general, we want to find out an answer of this question ‘In what time, they prefer what?’ The answer of this question would give us an insight of how people behave. Here is a simple example, we can study what people prefer for their lunch during lunch time. What is more popular? Or what if we do more in the application, like give some promotions and suggest them to some restaurant, would they behave differently? Able to answering such question would give a huge potential business value.

2.2 Study targets

Although this application is supposingly open to public and anyone could download it from app store, to limit the study scope is rather important in a limited time project. We shall focus to study the behaviour of university students as they will be our first category of user that will use our app.
3. Project methodology

1. Mobile App development

There is lots of app development framework which support both iOS and Android, and they required only some minor changes to the source code and native settings to build the app.

In this project, we are planned to use Flutter as our mobile app frontend framework. Flutter is developed by Google and its stable is released at 9th March 2018. It uses Dart as the programming language and it has a build-in UI framework (which use Google’s Material UI). General developer tools like debug logger, hot update function… are included. So we believe Flutter would be good enough to be used in our development.

Framework compare

React-Native is another cross platform framework that serve very similar function as Flutter and it is quite popular nowadays. Based on the React framework, React-Native uses javascript and require a packager to maintain the application. Libraries of React-Native are mature and it does have a large variety (since React and javascript libraries are also supported). There are also many choices of UI framework, including Material UI, that can be work with.

Objectively, both framework would qualify enough to be our pick, but there are some reasons that support us to prefer Flutter over React-Native.

It is all about the developing experience. During development, React-Native require the computer to run a Node packager to maintain the app state, so when we test the app with the real machine, our machine need to connect, through network socket, to the hosting computer. Whenever there is hot update, the packager will push the update to the device. It means that if we take our device away from the hosting computer network, the app cannot run.

Flutter uses another approach, the ‘packager’ (not sure how is it called) is run in the device and if you want hot update, you need to connect the device to the computer (physically). Flutter will figure out what is your recent changes and only update them accordingly. After the update, you can disconnect the device from the computer, which mean you can test the app wherever you want. So, in terms of the experience, we think Flutter would give us a better feeling.
2. Server development

We will need a database to store all users’ information and a server to handle all of the matchmaking requests.

In this project, we will use Node.js to develop our server. Node.js is a javascript based framework and it allows the server to run the javascript. We know that Node.js would not be the best choice since this project may involve a large volume of data process if there are many users. However, we believe that in this project scope, Node.js server is good enough to handle such data flow. Time and resource is limited in this project, so we pick a framework we are more familiar with.

3. Machine learning

Machine learning is required in this project in order to study the user’s behaviour. We planned to study the possibility of using Amazon Web Service’s machine learning module. If it is possible, we will try to use it, or we will do it ourselves by using Python.

There are lots of data we have to collect as it is required to train a machine learning model. Besides the user’s schedule and basic information, we need to collect some information such as location, how often they use the app, when they use the app and what they usually do in the app.
4. Project Schedule

Phase 1 is to make a usable application with user interface and server interaction. A server should be set up and it should be able to do some simple matchmaking.

Phase 2 is to implement machine learning which optimises the matchmaking results. Data should be collected for the machine learning process. The application should also be able to give suggestions for matched users regarding the activities they could do nearby or restaurants. An optional feature of random, immediate matchmaking could be implemented. For example, a user is free at the moment and would like to match up with any person nearby who happens to also be free. This matchmaking is totally random with an immediate matchmaking result.

<table>
<thead>
<tr>
<th>Month</th>
<th>Task Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>October 2018</td>
<td>Research and design all functionalities that the application would provide</td>
</tr>
<tr>
<td>November 2018</td>
<td>Develop the user interface and finalise the functionalities the application would provide</td>
</tr>
<tr>
<td>December 2018</td>
<td>Set up the server so it could do simple matchmaking and collect data for phase 2 machine learning</td>
</tr>
<tr>
<td>January 2019</td>
<td>Complete Phase 1 and first deliverable</td>
</tr>
<tr>
<td>February 2019</td>
<td>Collect data for machine learning and develop phase 2 features</td>
</tr>
<tr>
<td>March 2019</td>
<td>Do testing and debug and consider making the optional random feature</td>
</tr>
<tr>
<td>April 2019</td>
<td>Do testing and debug and finalise the product</td>
</tr>
</tbody>
</table>