This assignment is an implementation of an Android mobile app called **MyMonashMate** to help

students with finding new university friends**.** A current study shows1 that majority of international

students experience problems of loneliness or isolation, especially in the early months, and the

creation of new bonds between students in the educational settings is the key to this difficult period.

The **MyMonashMate** aims to provide a starting point for social bonding within the educational

scope and learning environment, particularly based on their common course and the corresponding

units.

This application will be implemented following a mobile distributed model. The Android application

will be the mobile client and consumer of a RESTful Web Service that you develop in NetBeans. The

RESTful web service will provide services such as registering/signing new members, storing profiles,

matching profiles, and others.

Using this app, first each student needs to sign in and provide some study-related and personal

information (profile). Then they can start using the app to locate other students nearby on a map

(e.g. within the campus) who match their profile and share the common interests (e.g. both enrolled

in FIT5046!). The map initially shows a very brief description of each matched student on the map

such as name but when the user taps on it, it opens up another screen/view that displays the more

information about the student (the visible/public parts).

The assignment MUST be implemented in **Android 4.4.2** (API 19) to be consistent with the lab

machines and tutorials.

**Task 1 – Database (5 marks):**

You need to create a database that stores the profiles of students.

**Database**

The profile will include a number of relevant attributes such as names (e.g. first name, surname,

nickname), course, units\*, geographical coordinates latitude, longitude, nationality, native

language, second language, living suburb, favourite food, favourite movie, favourite unit,

favourite programming language, current job, previous job, etc. The number of attributes is not

limited and you can add more. However, the attributes of id (PK, name, course, latitude,

longitude and units are compulsory. **(1 mark)**

\*Students are enrolled in one course but in multiple units. Therefore, there is a one-to-many

relationship between these fields. There is a need for an appropriate schema to implement this

relationship (this will require adding additional table/s and creating relationship between the

tables). The users/students need to sign in and sign out each time. The username and passwords

also need to be stored in the database. **(2 marks** [this includes designing the best possible

schema and its implementation]**)**

All the tables need to be populated at least with 15 records and all data should be meaningful

and close to real world values. Two of the important attributes are geographical coordinates

(latitude and longitude). This has to be hardcoded for all users/members but they need to

hardcoded according to the real locations in within or around the Caulfield campus **(1 mark)**

You need to implement a two level visibility for all the attributes. These levels include public and

restricted. Each attribute has to be assigned with a public or restricted visibility value. NOTE, the

attributes of Id, name, altitude, longitude, course, and unit SHOULD have public visibility and this

cannot be updated. **(1 mark)**

**Task 2 – RESTful Web Service (6 marks):**

You need to create **a RESTful web service** from the profile database (Task 1). **(1 mark)**

The RESTful web service needs to be **enhanced by adding extra methods or modifying** existing

code to **support different types of queries.** This means that from the Android client application

the user can enter different search keywords for matching and retrieving data, and the web

service should support them. The first screen after signing up should allow the user to customise

the matching based on a list of criteria such as matching based on the course, nationality,

language, suburb etc. The effectiveness and accuracy of your matching code and the quality of

SQL code are considered here. (**2 marks)**

NOTE, the query can access the data (e.g. about units) from more than one table. This requires

you to add extra code or modify the existing code in the web service. **(1 mark)**

For search, you need to create a second level of search such that the matching will be performed

based on multiple criteria (e.g. first based on course and then based on suburb) **(2 marks)**

**Task 3 Android Client of RESTful WS(2 marks):**

You need to create an android application client that will connect to the server and

consume the RESTful web service created in Task2. **(1 mark)**

Accessing data and executing all the queries from the server side (and the web service)

should be achieved using the AsynchTask approach **(1 mark)**

**Task 4 Screens, Interface and Navigation (9 marks):**

The Android client application will have several screens. The main screens that have to be

implemented are as follows and marking is based on achieving high quality user interface and

design.

- The first Screen to sign in and create account for the first time user **(1 mark)**

- The screen to enter the data for the profile (for the first time user) **(1 mark)**

- The screen for updating profile (after the user is registered) **(1 mark)**

- The main screen that displays the map and students who match the user’s profile. **(1 mark)** This

screen also should allow the user to set the matching criteria using appropriate widgets and

components. **(1 mark)**

- The screen to display the detailed profile of students. When the user taps on a marker on the

map, a new screen or view should appear with extra information. The detailed profile can be

also displayed in the map screen using an appropriate widget or component. A standard bubble

or simple widget has (1 mark) but if you customise and improve it **(2 marks)**

Between all screens a logical and easy navigation needs to be implemented **(1 mark)**

Data entry validation implemented and error messages are provided where necessary. **(1 mark)**

**Task 5 Using MapQuest (3 marks):**

In this assignment, since it is limited to the use of emulators, we will use MapQuest. In the

main/map screen the application should display the students who match the selected criteria and

those who are closer to the location of the user based on latitude and longitude (hardcoded in the

database) **(2 marks)**

To differentiate between the matched students based on who is a closer match, markers on the map

will use a colouring scheme so that if the matching results shows a student is a closer match the

colour can be darker or red (or any other colouring scheme) and the ones with less matching score

will have a different colour. **(1 mark)**

**Task 6 An additional feature (3 marks):**

This task requires you to enhance the existing Android application further with implementing an

additional feature. The examples include:

i) linking the application to the Facebook,

ii) enabling communication between the user and any of the students displayed on the

map. This can be achieved by implementing a customised messaging system and

integrate it with the web service, and storing the messages [i.e. from to content] in

another table ,or using existing technologies such as Google Cloud Messaging

service for Android and socket.io

iii) using an ontology to store the user profile and apply ontology reasoning to enable

semantic search.

iv) enhancing security of the app by implementing an authentication mechanism into

the RESTful web service using e.g. shared secret key (different for each client).

**Additional Marking Criteria**

**Only completing a task does not yield a HD mark.** There are 6 other criteria for marking each task.

For a full mark of any task, all the following criteria should be achieved:

**1. High quality of design and programming/coding (e.g. high cohesion, loose coupling,**

**,separation of concerns, bug free, solid exception handling, following coding standards, proper**

**and meaningful naming of variables and methods, efficient use of variables, etc)**

**2. Excellent GUI interface and layout and navigation**

**3. Evidence of originality and creativity of student (going beyond the assignment specification),**

**4. Full functionality of all the operations (during the interview),**

**5. Handling of all the input errors and exceptions properly,**

**Related mobile apps:**

- the Monash Univeristy App. It provides access to important information and campus services such

as eMaps with layers of points of interest e.g. libraries or lecture theatres and parking location, staff

Directory, Monash News, Searchable unit entries from the official University Handbook, and Shuttle

Bus Timetables

(https://play.google.com/store/apps/details?id=com.monashuniversity.monashuniversity&hl=en)

- Find My Friends app: https://play.google.com/store/apps/details?id=com.fsp.android.friendlocator