

Individual Final Report: Co-runner

THE UNIVERSITY OF HONG KONG
DEPARTMENT OF COMPUTER SCIENCE



FYP

FINAL YEAR PROJECT

Individual Final Report

Project Title: Co-runner

Supervisor : Dr. Chung Ronald

Student : Choi Chi Hin (UID: 3035095145)

Submission Date : 16/4/2017

Individual Final Report: Co-runner

Abstract

In Hong Kong, many people are keen on running and around ten marathons are held each month^[1]. The fact that millions of money donated to charity organizations in Hong Kong exemplifies citizens' generosity for donation. However, currently there is little connection between these two sectors. The objective of this project is to create a digital platform which allows users to make sustainable donation can be conducted during workout.

The paper starts with the discussion of the project background and the need of integration between sports and charity, which the existing mobile application, such as Nike + Run Club, is in lack of. The deliverables and the functions of the application are also reviewed in this part.

In the second part, the team discusses the details of the methodology, which consists of design, implementation and testing phase. The setup of Android, Python, MySQL and Google Cloud Platform are carried out in this phase. In other words, the local server and database setup are conducted. In the implementation phase, the team moves to the workflow implementation to build the connection between the server and client.

The function implementation is another focus in this phase. Recording function and display function are manipulated to build the API (Application Program Interface) of the mobile application. With the test cases and feedback mechanism in the testing phase, a prototype with high utility, workflow management and efficient of the application is released in final stage.

Acknowledgement

I would like to express my great appreciation to Dr. Chung Ronald, my supervisor, for his patient guidance, useful advices, and critiques throughout the project. His help has been very much appreciated.

Individual Final Report: Co-runner

Table of Contents

Abstract	2
Acknowledgment	2
Table of Contents	3
List of figures and tables	4
Section I – Introduction	5-10
1.1 Background	5
1.2 Objectives	6
1.3 Existing Applications	7
1.3.1 Nike + Run Club	7-8
1.3.2 Charity Miles	8
1.4 Scope	9
1.5 List of features	9
1.6 Deliverables	10
1.7 Outline	10
Section II – Approach and Methodology	11-15
2.1 Overview	11
2.2 Design Phase	11
2.2.1 Mobile Platform Setup	11
2.2.2 Backend Setup	11
2.2.3 Database Setup	12
2.2.3.1 MySQL Database	12
2.2.3.1 Google Cloud Platform Database	12
2.3 Implementation Phase	13
2.3.1 Connection in client-server model	13-14
2.3.2 Function implementation	15-18
2.4 Testing Phase	19
Section III – Contributions and Project Deliverables	20-
3.1 Progress Summary	20
3.1.1 Frontend implementation	20-21
3.1.2 Database setup	22-23
3.1.3 Python coding	24-26
3.1.4 API documentation	27
3.2 Challenge and mitigation	28
3.2.1 Business partner cooperation	28
3.2.2 Recording function	28
3.3 Schedule and Milestones	29
3.4 Future Development	29
3.2.1 Business cooperation	29
3.2.2 Security	29

Individual Final Report: Co-runner

Section IV Conclusion	30
Section V Reference	30
Appendix	31-33

List of Figures

<i>Figure (1) - Overview of Co-runner</i>	6
<i>Figure (2) - Application of Nike + Run Club</i>	7
<i>Figure (3) - Application of Charity Miles</i>	8
<i>Figure (4) - Infrastructure of system component</i>	11
<i>Figure (5) – Process of data request (SQL query)</i>	13
<i>Figure (6) – Process of information retrieval</i>	14
<i>Figure (7) – Use case diagram for login function and screen-captioned image of Login Page and Homepage</i>	16
<i>Figure (8) – Use case diagram for function of charity selection</i>	17
<i>Figure (9) – Use case diagram for function of displaying workout data</i>	18
<i>Figure (10) – MainPage Figure</i>	21
<i>Figure (11) – ManageAccount Figure</i>	21
<i>Figure (12) – Previous Workout Figure</i>	21
<i>Figure (13) – Current Workout Statistics Figure</i>	21
<i>Figure (14) – ER diagram of database design</i>	23
<i>Figure (15) – Python coding (1)</i>	25
<i>Figure (16) – Python coding (2)</i>	26
<i>Figure (17) – Use case diagram for login function and screen-captioned image of Login Page and Homepage (Larger pixel)</i>	31
<i>Figure (18) – Use case diagram for function of charity selection (Larger pixel)</i>	32
<i>Figure (19) – Use case diagram for function of displaying workout data (Larger pixel)</i>	33

List of Tables

<i>Table (1) – Entitles in the database</i>	22
<i>Table (2) – Future Work Plan</i>	29

Table (3) – Test areas and the passing criteria of the trial test 30

Section I – Introduction

1.1 Background

With the advanced development of mobile technology in recent years, many integrations between different industries are facilitated. The integrations not only can add value to the business or industries stand alone, it can also contribute to society to reach a win-win-win situation.

The consistently high participation of the marathon and running competitions shows that people in Hong Kong are actively engaged in jogging. The fervency of Hong Kong people extends from workout to charity. People are crowned for donation. Statistics reveals that HK\$9.2 billion in 2010-11.^[2], which sees HK ranks 19th among the world.

While the two sectors gain much popularity in Hong Kong, there is little connection between sports and charity. Most of the existing mobile applications target at personal training, without bringing much social impacts. While charity groups rely much on traditional channels, such as booth promotion, flag-selling campaign, to receive donation, it is difficult for them to collect sustainable income. In fact, the high workout participation can lead to much donation. In view of this, Co-runner targets to make donation through running. Co-runner is a socially-driven, Android-based mobile application, connecting charities, individuals, and corporate sponsors to create social change. It enables users to make donation with the assistance of sponsors for charity whenever they work out.

Individual Final Report: Co-runner

1.2 Objectives

Co-runner aims to strengthen the prevalent trend of jogging and optimize citizens workout experience by integrating with charity and business corporations. Co-runner is an Android mobile application to create a social network to gather people and create more opportunities for donations

Business sponsors to make donation on behalf of the users to a charity chosen by the users. The application works as a fitness tracker to record runners' performance data including distance performed, time elapsed, calories burnt etc. Afterwards, business partners make donation to charity based on the miles recorded. For example, business partners donate \$1 to charity if runners run 10 kilometers. Ultimately, users may post their workout achievements and donation amount in social media to further promote this platform.

This project also has some long-term objectives. The final deliverable can achieve:

An account-based mobile application is delivered to record distance travelled or walk by the users. It can track money donated to chosen charities and displays advertisements and offers from participating businesses.

Apart from the application, the second deliverable is the backend server. It serves to record donation to charities and chooses proper ads based on information received.

An interim Report and interim presentation completed in mid January to present the progress and future plan of Co-runner. A final report and final presentation completed in mid April to showcase the final deliverable of Co-runner.

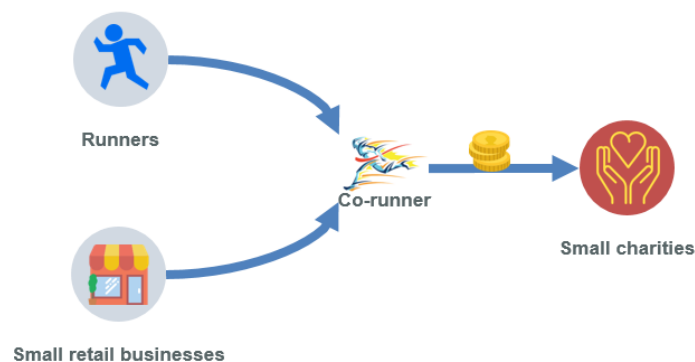


Figure (1) Overview of Co-runner

1.3 Existing Applications

Currently there is no mobile application which shares similar model as Co-runner in Hong Kong to blend sports and charity but there is a counterpart in Taiwan.

Currently in Hong Kong, there are hundreds of mobile applications which only offer assistance to individual during workout by recording a wide range of data. These kinds of features may provide advantages to runners, but this application may lack social impact. This will be further explained by discussing one of the local mobile applications- Nike+ Run Club.

In Taiwan, a mobile application called Charity Miles shares similar rationale as the model described above. It was established in 2012 and it has earned over 500k+ download rates in a rapid rate and cooperated with large brand partners and sponsors. As an initiative in Hong Kong, it is believed that Co-runner can highlight this case for reference.

1.31 Nike + Run Club

Nike +Run Club records users' performance details, ranging from distance to speed and running routes. As shown in Figure (2), Users can also retrieve the data from the history record. In addition, different targets or achievements can also be set according to individual physique. On top of that, users can schedule a week-based exercise schedule. It suggests everyday schedule according to the target for running distance, speed, number of workouts per day. This help cultivates long-term commitment and self-reflection.

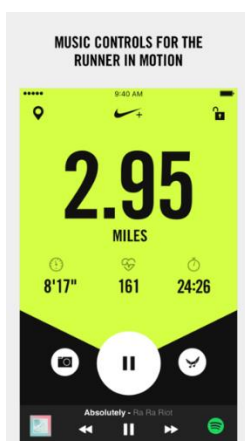


Figure (2) - Application layout of Nike + Run Club

Some may praise this application as handy and useful for exercise as some features can provide tailor-made arrangements. On the other hand, this application lays much emphasis on individual exercise and improvement, which in turn lack connection to other parties. Considering this perspective, the proposed project can complement in this area.

Individual Final Report: Co-runner

1.32 Charity Miles

Charity Miles is a free mobile application established in Taiwan. Not only does it collect the workout statistics for individual improvement, it also creates social impacts. Corporate sponsors make donation on users' behalf to the charity chosen by the users. Charity Miles co-operates with 20+ charity bodies and some renowned sponsorship to promote this mobile application. As shown in Figure (3), the application record and display the distance performed by the users, together with the corresponding amount of donation generated.

Charity Miles shares some similarities with the purposed model of Co-runner. For the technical development, it is realized that they focus more on the security problem and database management system while Co-runner may focus more on the implementation of functionalities of the mobile application.



Figure (3) - Application layout of Charity Miles

Individual Final Report: Co-runner

1.4 Scope

The mobile application only covers outdoor usage. It is difficult to implement indoor running route as it requires three-dimension data to construct indoor routes. It may be difficult to collect three-dimension data and it may lead to unprecise data and thus uncomplete running routes. In this account, Co-runner decides not to cover the running routes for indoor cases.

1.5 List of features

The mobile application records four workout data, namely distance performed, speed calories burnt and donation amount. These data are also kept in database system.

With the assistance of GPS system and Google map API, Co-runner is designed to capture the running routes and they are also kept in the database system

Users are allowed to view their all past performance in their personal profile. The mobile application displays the workout data of the past performance by retrieving the data in the database.

Individual Final Report: Co-runner

1.6 Deliverables

The team executes the project with four deliverables, namely - Android mobile application, Python backend server, MySQL Database and Google Cloud Platform.

An Android mobile application that tracks the workout statistics by using GPS system and Google API. It also indicates the amount money donated to chosen charities, and advertisements from participating businesses may be displayed.

Python backend server is built to handle the data request from the application(client) side by retrieving data in MySQL database and display in client side.

MySQL database is used to store the data of users, charities and business corporations. There is real-time update for all the data when users finish workout.

Google Cloud Platform is designed to satisfy the wants of Google Search and Youtube. It is in a vicinity of the Google for Work package. Google Cloud Platform consists of fifty services and half-dozen online world knowledge center. It is chosen to be a backend database as it has large storage system but with lower security, compared with MySQL database.

1.7 Outline

In Ch.1, the paper discusses the background and motivation of Co-runner, the Android-based mobile application supported by Python as backend system and MySQL as database system. In Ch.2, the function and the methodology – design, implementation, and testing phase are explained. The workflow of the client-server model is discussed. MySQL database is the server to receive client (The Android Application) and retrieve data from MySQL database. The current progress and challenges are discussed in Ch.3, followed by a conclusion in Ch.4

Individual Final Report: Co-runner

Section II– Approach and Methodology

2.1 Overview

The project follows a software development cycle framework. The cycle is divided into three phases - design phase, implementation phase, and testing phase.

As the application is account based, Co-runner decides to employ a client-server model. As the server only needs to store users' profile data and workout data, it is no need to use a complex backend setup. MySQL database and Python server are chosen. The database stores all the users' data, including running statistics, favorite charities and past record. Google Cloud Platform is used as backup database.

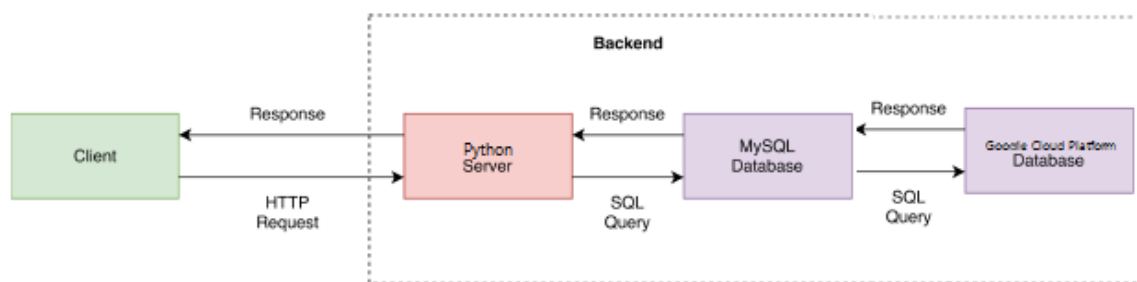


Figure (4) - Infrastructure of system components

2.2 Design Phase

The design phase completed in October. At this stage, the project team focuses heavily on the environment setup, which consists of Android studio, MySQL, Python and Google Cloud Platform as shown in Figure(4). The four platforms are established independently in Design Phase and are integrated in the implementation phase. The design phase includes the following works:

2.2.1 Mobile Platform (Frontend) setup

Android studio is the standard development IDE for develop android application. At present, more than 80% of the smartphone devices use Android as their operating system^[3]. Android powered devices have become the foremost need of all the tech-savvy people across the world and the prime reason is being an open source platform for the development of mobile application and allows application developers to immediately publish them.

The team first constructed a prototype version of the mobile application and check if it satisfies the functionality requirement. Afterwards, the prototype mobile connects the backend and database system.

Individual Final Report: Co-runner

2.2.2 Backend Setup

Python makes use of strict indentation enforcements that is very convenient for developers. This causes Python being more readable than other programming languages with a simpler syntax and code, which is easier to understand in Python. Another feature is that Python it is simple in structure and very well documented for developers to implement different functions.

2.2.3 Database Setup

2.2.3.1 MySQL Database (backend – database)

MySQL Database is an open source relational database management system which is used to store user information, for example past record, workout data. Compared to other major open-source relational database management systems like, SQLite and PostgreSQL, MySQL is the most popular one with many features. Also, MySQL is more secure as it provides security measures. For example, it implements SSL library to provide connection encryption. Clients are required to use their own set of private keys and certificate to use the encryption.

2.2.3.2 Google Cloud Platform (backend – database)

Google Cloud Platform is used for backup storage of the user data. It is an internet-based platform to store and manage data using a remote server hosted on the Internet which make use of technology of cloud computing. Cloud computing ensures data is available anytime and anywhere. Also, cloud computing is cheaper than other alternatives as no purchase equipment is required. Currently, Google drive and Google Cloud are the two most popular cloud storage system. The reason to choose Google Cloud Platform is that it is enterprise platform which supports advanced data management features and better security. It also provides durable storage with fast performance.

Individual Final Report: Co-runner

2.3 Implementation Phase

The implementation phase has completed in March. There are two main tasks in this phase –

1. Construct connection between the four software in the client-server model to handle data request from client.
2. Function implementation

2.3.1

Connection in client-server model

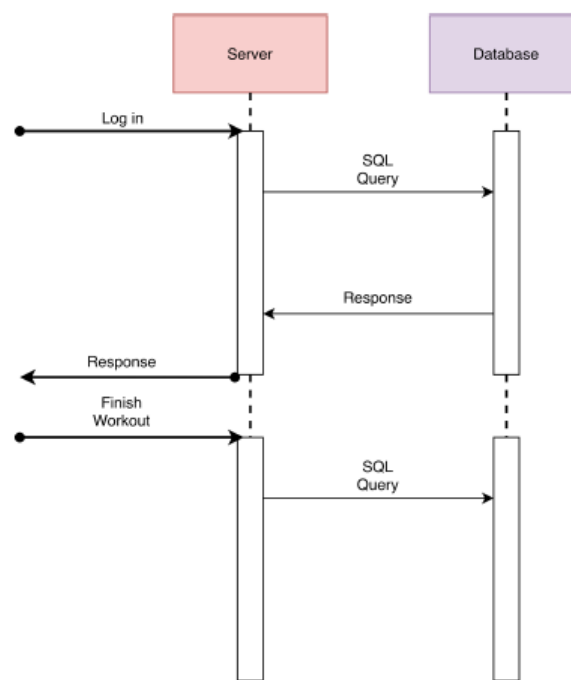


Figure (5) – Process of data request (SQL query)

With reference to Figure (5), communication between the client and the backend is kept minimal as there are only two scenarios to conduct SQL Query- Login stage and completion of workout

Upon logging in the application, the client sends an HTTP GET request to the server to retrieve the user's profile data. The server parses this request, and makes an SQL query to access the user's profile. The data of the user's profile is then returned to the server, and in turn, the client.

Upon workout, the client sends a POST request with the updated profile information to the server, and the server updates the database.

Individual Final Report: Co-runner

Client (Android Studio)

The client is an Android application. It deals with account creation, workout tracking, and other functions interact with the server. When the users' information is updated, it updates its locally stored profile as well as send the new information to the server.

Server (Python)

The server is written in Python, and its function is relatively simple. The main function is to receive and interpret the HTTP requests from the client (Android application), then access the database to retrieve requested data, and generate the appropriate response to send back to client. In most cases, the server only needs to retrieve data from MySQL database server as most of the data are stored in MySQL as data frequent users is stored in MySQL. In order to prevent MySQL database from suffering overload due to excess data storage, for users who have not used the application for one month, their data are stored in Google Cloud Platform.

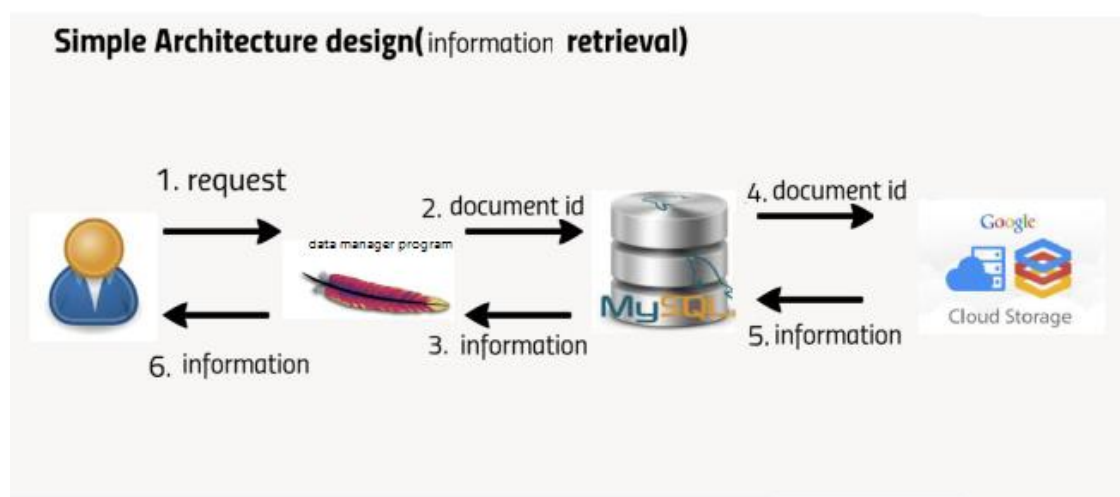


Figure (6) – Process of information retrieval

In an architecture point of view, Figure (6) shows the interaction between different software to achieve information retrieval. When clients send a request from the Android application to client side, data manager program handles the request by sending the document ID of the request to the database, MySQL database in most of the cases. MySQL database then returns the detailed information of the profile like past record and favored charity, to the data manager, and thus the client. If users have not used the application more than one month, the information is retrieved in Google Cloud Storage, which requires a further

Individual Final Report: Co-runner

request. The data is also cached in MySQL in order to have faster response for the data request next time.

2.3.2 Function implementation

Another goal in implementation is function implementation. Our team targets to achieve the following functionality requirements.

1. Create an account

- a. Each runner will need their own account to use the application
- b. Runners will log in with their Facebook account
- c. View profile: total distance, total sessions, charity points, money earned for charity, favorite charities

2. Workouts

- a. Review past workout information
- b. Create a new workout: select activity and charity project
- c. View current workout information: time, calories burned, distance, points earned, money donated

3. Charities

- a. Choose a charity to run for during a workout
- b. See how much has donated to this charity through the app
- c. Mark favorite charities
- d. Remove favorite charities from list

4. Businesses

- a. View information about donating businesses
- b. Address, phone, hours

5. Menu

- a. Start a workout
- b. View participating charities
- c. View participating businesses
- d. Redeem coupons
- e. View ranking based on total amount donated
- f. View profile
- g. Search profiles, charities, and businesses
- h. Logout

Individual Final Report: Co-runner

The following use-case diagrams are served as step-by-step guideline to explain the main functions work well, including login function, function for charity selection and display function.

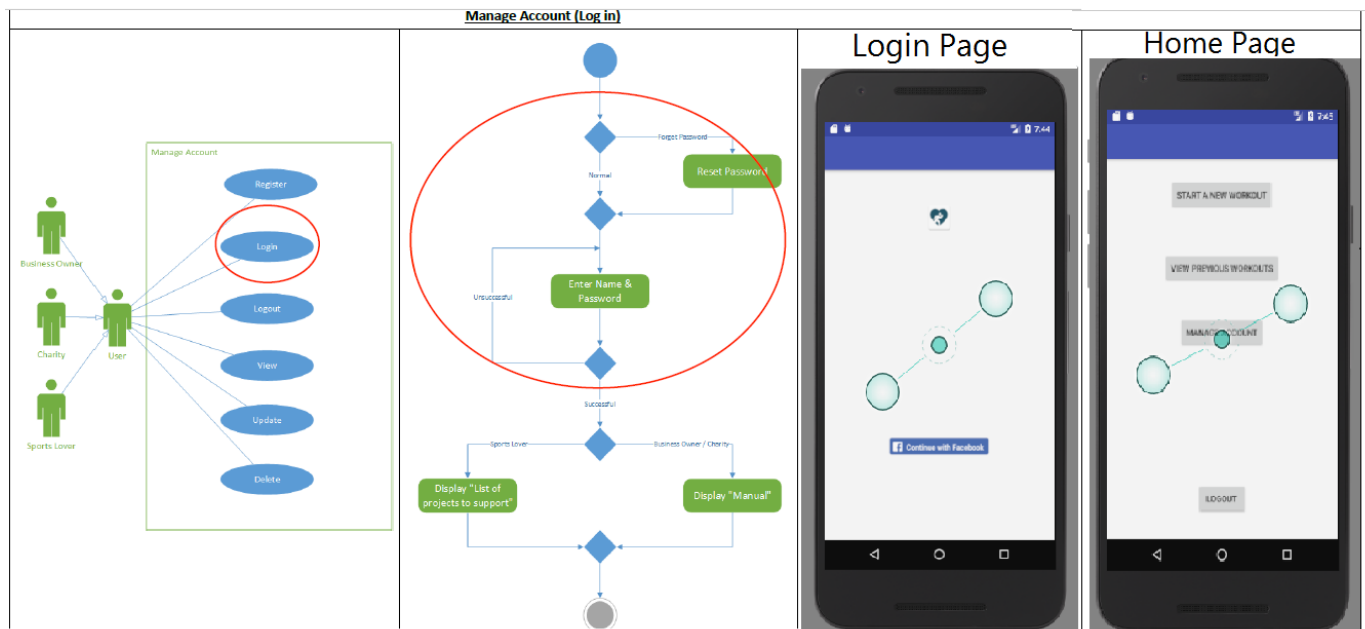


Figure (7) – Use case diagram for login function and screen-captioned image of Login Page and Homepage

(The Use case diagram with larger pixel is available in Appendix)

1. In the Login Page, when users login the application via Facebook or Twitter account*, system checks whether they are new users. If so, they are required to create a new account. If no, users enter the Homepage and the data is retrieved from the database in the meantime. Every time users enter their account again, the application is able to retrieve basic information of the user from social media the user has chosen to authorize.
2. In the Homepage, users can choose to start workout, view past workout record, and manage the account.

* With today's myriad of security threats, developing our own secure login system in a short timeframe may not be possible. Because of that, using reputable login APIs would be a better idea. Since we want our users to be able to share activities to Facebook from our app and to connect with their friends, Facebook login APIs would be the best choice for us.

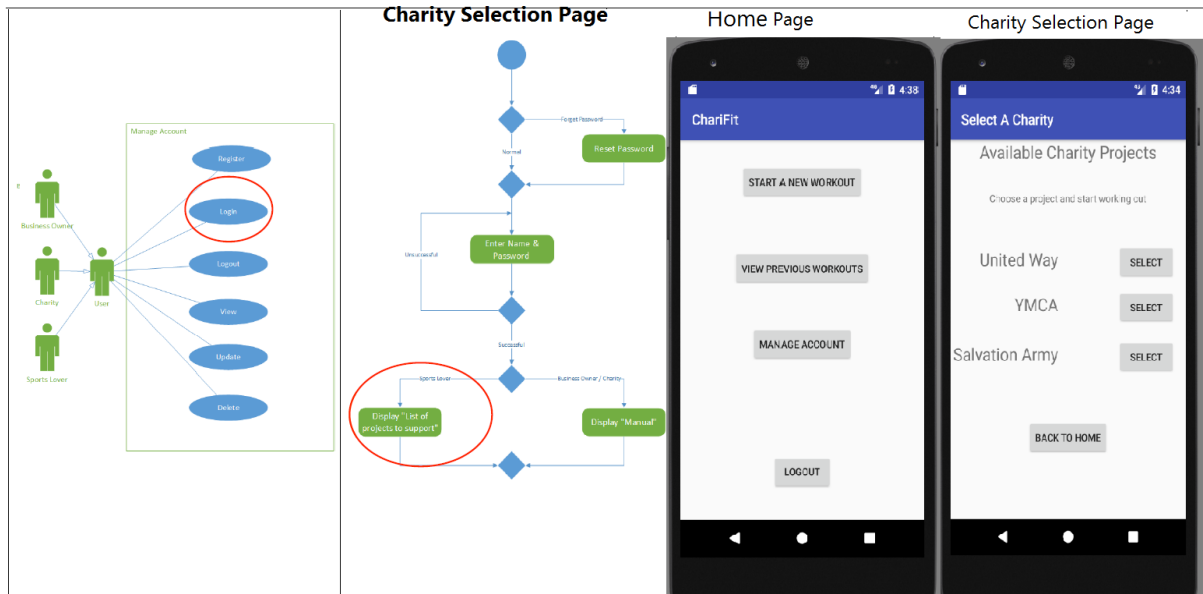


Figure (8) – Use case diagram for function for charity selection
 (The Use case diagram with larger pixel is available in Appendix)

1. After users choose to workout (jogging) in the Homepage, the system retrieves the list of charity from the database and displays in client side.
2. Users can pick their favored charity and start exercise.

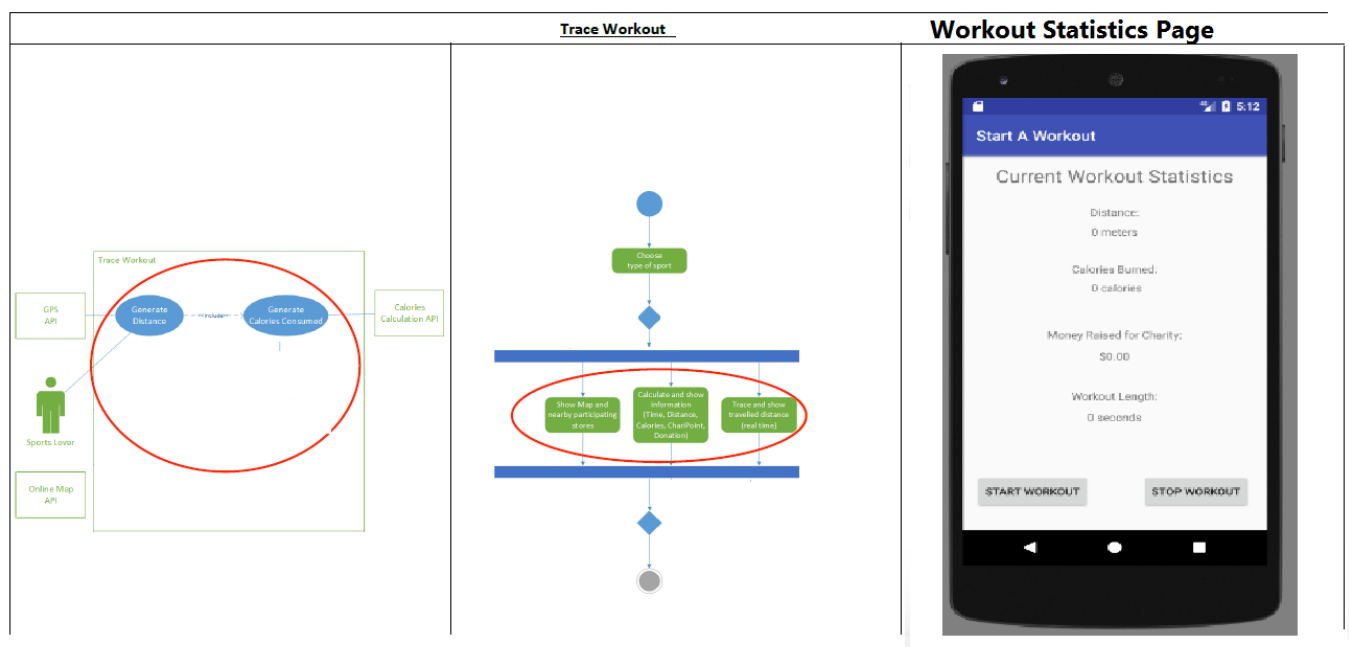


Figure (9) – Use case diagram for function of displaying workout data
(The Use case diagram with larger pixel is available in Appendix)

1. The system automatically retrieves their option in the last workout. After users click the “start” icon, the application starts recording the workout data of the performance
2. After user finishes exercise, the workout data, including distance covered, time elapsed are shown in client side. The database also updates with the latest runners’ workout statistics. The amount donated to the charity is also shown based on kilometers run.

Individual Final Report: Co-runner

2.4 Testing Phase

The application is accessed mainly for the performance of the functionality and the connection between different platforms.

In the final milestone, black-box testing was applied for the final product. The application was tested in Android Phone of the Android Emulator. The test cases are designed for standard scenario. Some criteria are observed to check whether the loading time and recording time is acceptable.

Section III– Contributions and Project Deliverables

In the first semester, the team has finished the research of all the technical and business requirement. The team also completed the environment setup, including Android platform, Database and Python backend in implementation stage. The team faces some unsolvable challenges in the following aspects - including business cooperation and data update between client side and server side. The deliverable of co-runner allows basic functions which is meet the team expectation.

3.1 Progress Summary

3.1.1 Frontend implementation

For application interface and request handling, the platform adopts Android studio as the development framework. Android Studio is the most common platform for writing Android application.

The layer of the coding design is quite simple. There are few main pages, which are “MainPage”, “ManageAccount”, “Previous Workouts” and “WorkoutPage”.

As indicated by Figure(10), the “MainPage” serves as the homepage of the application after successful login, it allows users to start workout, view previous workout record and manage account.

As indicated by Figure (11), the “ManageAccount” page allows users to change the account name or delete account. The team has implemented some logic to prevent deuplicate username and add API function to change and update the name

As indicated by Figure (12), the “Previous Workouts” page allows users to view the previous workout record.

As indicated by Figure (13), the “WorkoutPage” page allows users to start workout and with reference to ch.2.3.2, the users may choose different charity and perform workout.

Individual Final Report: Co-runner

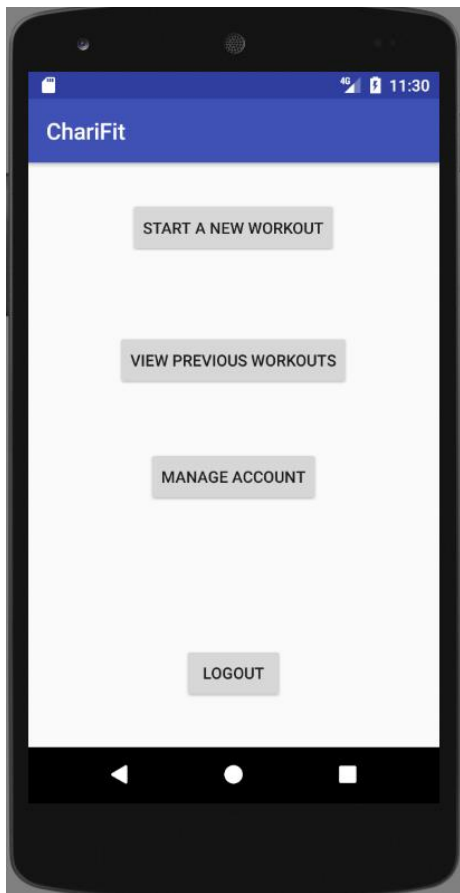


Figure (10) MainPage Figure

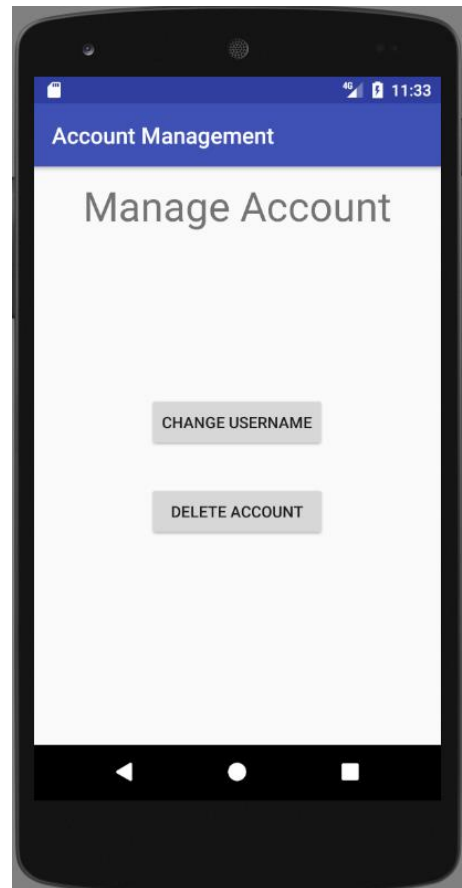


Figure (11) ManageAccount Figure

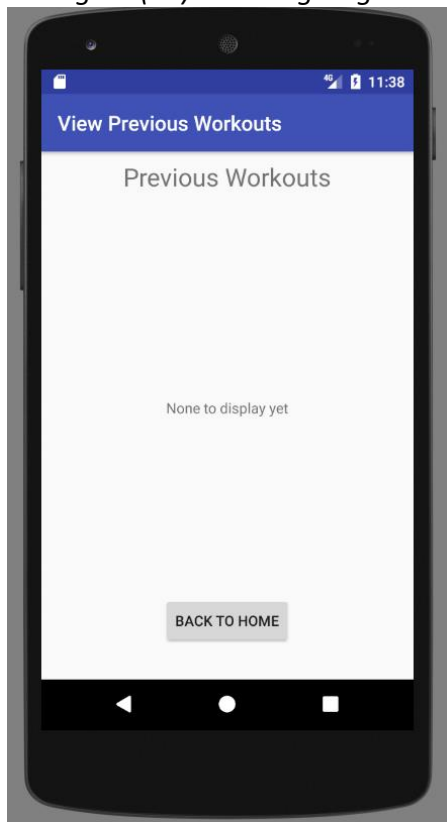
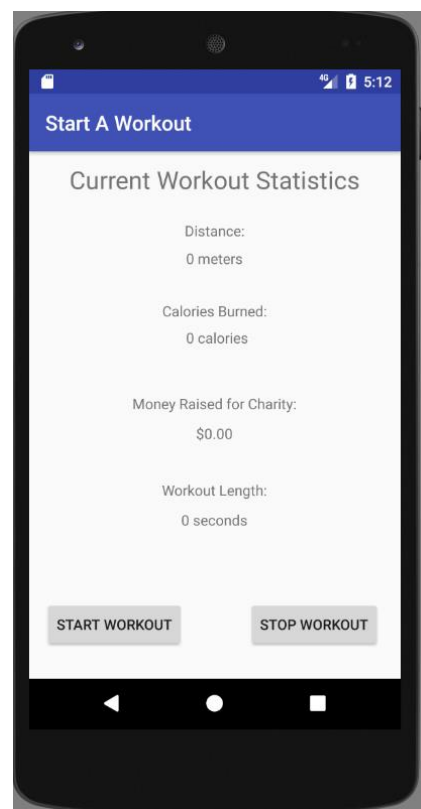


Figure (12) Previous Workout Figure



Figure(13) Current Workout Statistics Figure

Individual Final Report: Co-runner

3.1.2 Database setup

Concerning the current database design, the team has created 15 entities, which cater for all users from companies, users, charity company. The entity servers many purposes, including the authentication right of different information for different users, different information stored in the database and past information recording.

Entity	Usage
odm_data	contains the information of the document stored in the server
odm_user_perms	contains permission of a user
odm_settings	contains the settings of the system
odm_rights	contains the kind of rights in the system
odm_udf	contains user defined field for adding custom meta information to the documents
odm_user	contains the information about a user
odm_odmsys	contains the information about the system
odm_filetypes	contains the types of file supported in the system
odm_category	contains the kind of categories in the system
odm_dept_reviewer	contains the user information who is the department reviewer
odm_log	contains the log of each document
odm_access_log	contains the access log of the system
odm_department	contains the kind of categories in the system
odm_dept_perms	contains permission of a department
odm_admin	contains the information about an admin

Table 1. 15 entities in the database

Individual Final Report: Co-runner

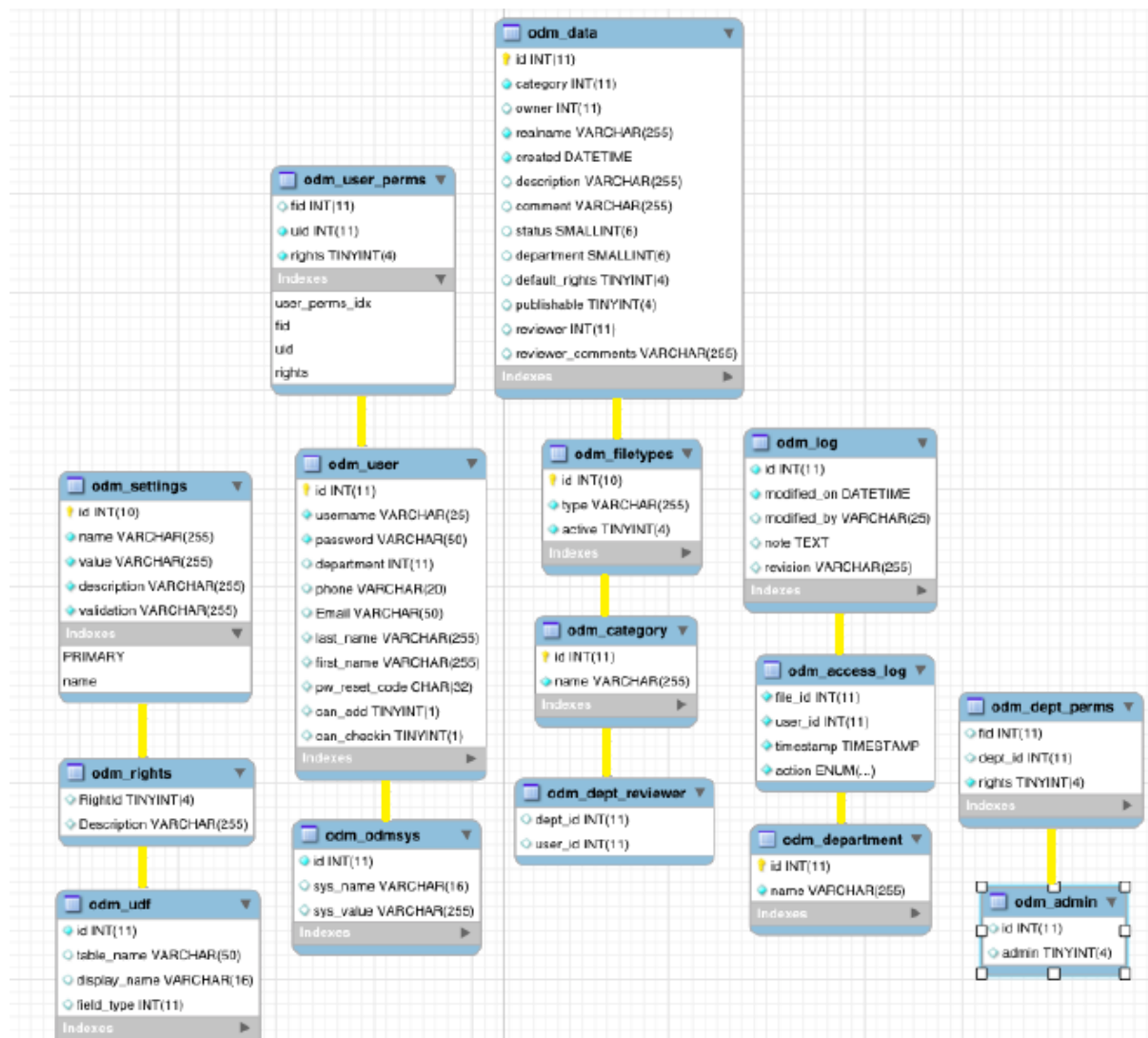


Figure 14 ER diagram of database design

Figure (14) shows the ER Diagram- which shows the hierarchy of the 15 entities, with odm_data (storage of workout data) and odm_user_perms (Permission right of the database) being the parent node / root node.

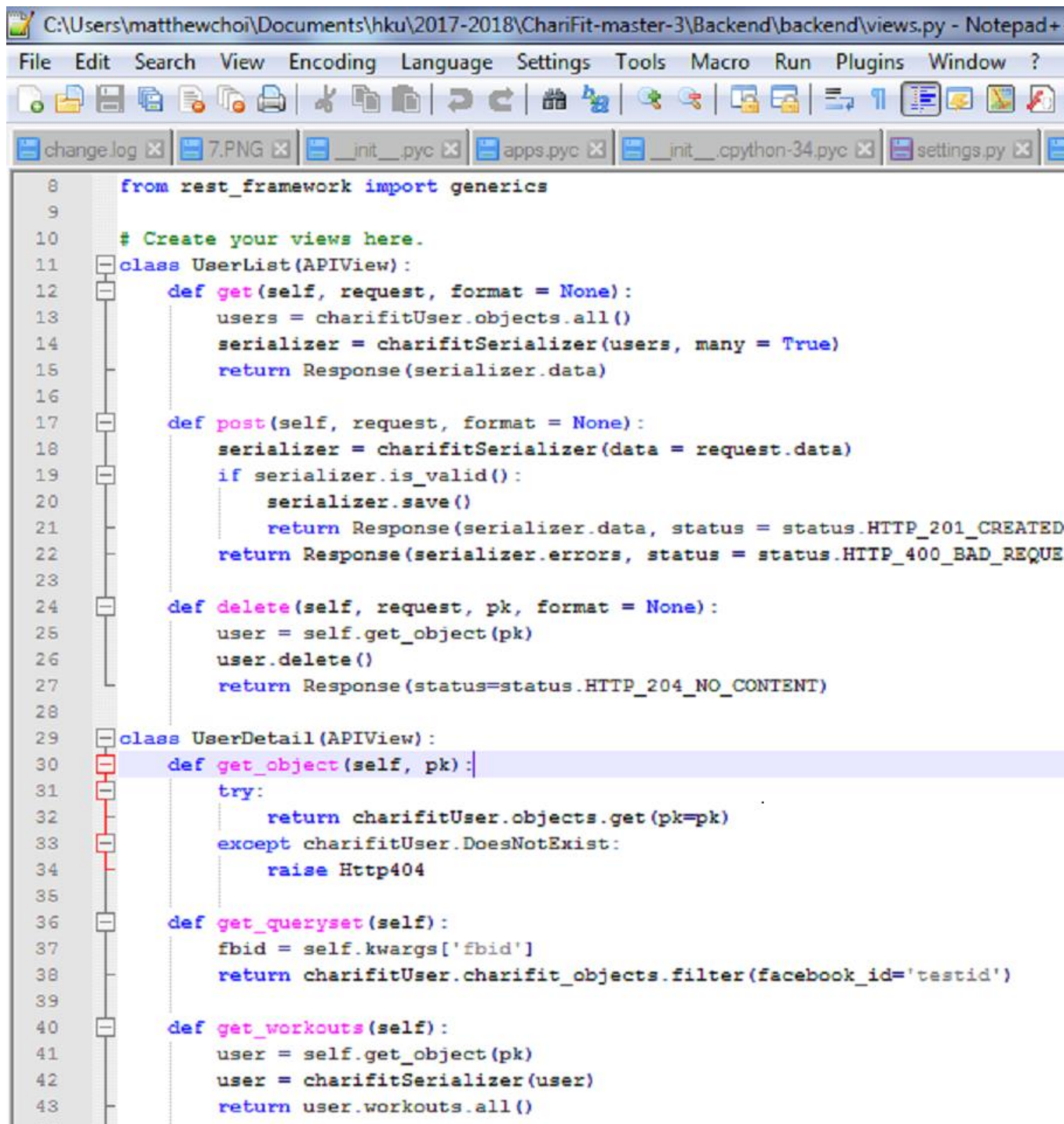
The database structure is simple as there is only 1 to 1 relation for the 15 entities, as indicated by the connection highlighted in yellow.

Currently, as the mobile application still not available in the market, no true data is collected and recorded. The database setup is for trial purpose and have not connected to backend development which is regarded as advanced function.

Individual Final Report: Co-runner

3.1.3 Python coding

In regard of the backend design, the team has implemented the GET and POST function to receive the data request from the client side (as indicated in line 12-15 and line 36-38 in Figure (15)). These two functions are used to handle request from the application so as to display the past workout record in the client side.

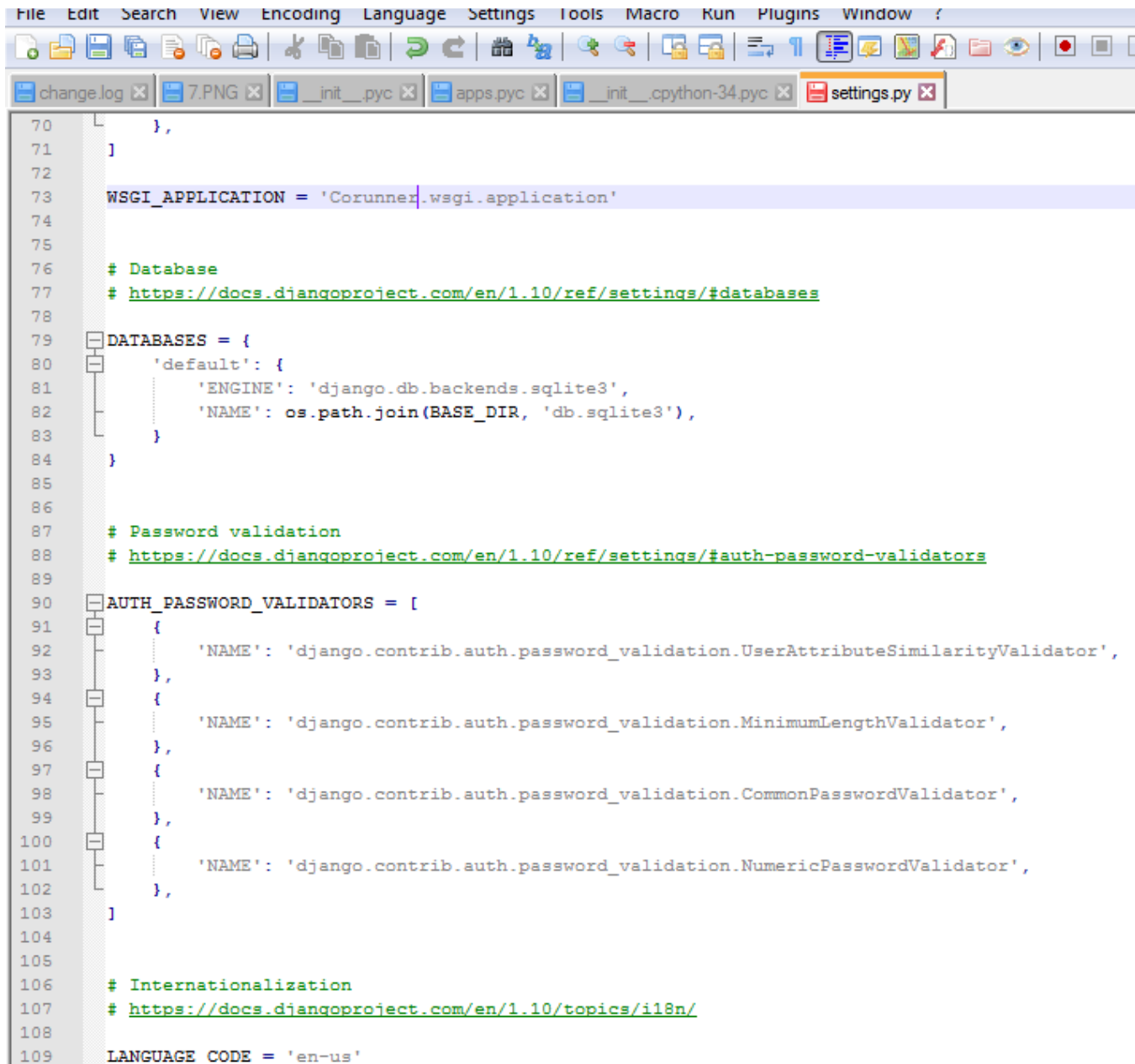


```
8 from rest_framework import generics
9
10 # Create your views here.
11 class UserList(APIView):
12     def get(self, request, format = None):
13         users = charifitUser.objects.all()
14         serializer = charifitSerializer(users, many = True)
15         return Response(serializer.data)
16
17     def post(self, request, format = None):
18         serializer = charifitSerializer(data = request.data)
19         if serializer.is_valid():
20             serializer.save()
21             return Response(serializer.data, status = status.HTTP_201_CREATED)
22         return Response(serializer.errors, status = status.HTTP_400_BAD_REQUEST)
23
24     def delete(self, request, pk, format = None):
25         user = self.get_object(pk)
26         user.delete()
27         return Response(status=status.HTTP_204_NO_CONTENT)
28
29 class UserDetails(APIView):
30     def get_object(self, pk):
31         try:
32             return charifitUser.objects.get(pk=pk)
33         except charifitUser.DoesNotExist:
34             raise Http404
35
36     def get_queryset(self):
37         fbid = self.kwargs['fbid']
38         return charifitUser.charifit_objects.filter(facebook_id='testid')
39
40     def get_workouts(self):
41         user = self.get_object(pk)
42         user = charifitSerializer(user)
43         return user.workouts.all()
```

Figure 15– Python coding (1)

Individual Final Report: Co-runner

Second, the team completed integration between Python (Backend) and Database by connecting to the MySQL database (as indicated in line79-84 and line 90-103 in Figure (16)).

A screenshot of a Python IDE window showing the 'settings.py' file. The window title bar includes 'File', 'Edit', 'Search', 'View', 'Encoding', 'Language', 'Settings', 'Tools', 'Macro', 'Run', 'Plugins', and 'Window'. The file explorer shows several open files: 'change.log', '7.PNG', '__init__.pyc', 'apps.pyc', '__init__.cpython-34.pyc', and 'settings.py'. The code in 'settings.py' is as follows:

```
70     },
71 ]
72
73 WSGI_APPLICATION = 'Corunner.wsgi.application'
74
75
76 # Database
77 # https://docs.djangoproject.com/en/1.10/ref/settings/#databases
78
79 DATABASES = {
80     'default': {
81         'ENGINE': 'django.db.backends.sqlite3',
82         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
83     }
84 }
85
86
87 # Password validation
88 # https://docs.djangoproject.com/en/1.10/ref/settings/#auth-password-validators
89
90 AUTH_PASSWORD_VALIDATORS = [
91     {
92         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
93     },
94     {
95         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
96     },
97     {
98         'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
99     },
100    {
101        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
102    },
103 ]
104
105
106 # Internationalization
107 # https://docs.djangoproject.com/en/1.10/topics/i18n/
108
109 LANGUAGE_CODE = 'en-us'
```

Figure 16– Python coding (2)

Individual Final Report: Co-runner

3.1.4 API Documentation

Title: Show All Users

Shows all users stored in database

- URL: /users/
- Method:
GET, POST
- URL Params:

None

- Data Params:

```
{  
facebook_id : [integer]  
first_name: [string]  
last_name: [string]  
email: [string]  
weight: [integer]  
height: [integer]  
}
```

- Response Codes:
Success (200 OK), Bad Request (400), Unauthorized (401)
- Title: Get User By ID
Shows only the user which matches the ID provided
- URL: /users/[integer]
- Method:
 - o GET, PUT, DELETE
- URL Params:
 - o Integer value of ID to select
- Data Params:

```
{ facebook_id : [integer]  
first_name: [string]  
last_name: [string]  
email: [string]  
weight: [integer]  
height: [integer]  
}
```
- Response Codes:
 - o Success (200 OK), Bad Request (400), Unauthorized (401)

Individual Final Report: Co-runner

3.2 Challenge and mitigation

In the current stage, the team has faced challenges in three aspects –business partner cooperation and recording function

3.2.1 Business partner cooperation

Runners, business partners and charity are all equally vitally important for Co-runner to publish in market. Although with predication in early stage, the team is not still able to find any business partners.

Regarding business partners, much more effort may be required as they are expected to make donation to charity in a newly established mobile application platform. In order to attract more business sponsors, the team may consider cooperation with some small business partners which are more eager to build company images through different digital platforms. Also, Co-runner may provide more incentives, such as setting lower donation rate in early stage to attract more publication. Also, Co-runner can be a marketing platform for the business sponsors as advertisements can be put in the application.

3.2.2 Recording function

Recording the workout data is one of the important functions in the application. It was developed in the implementation stage, during which major problems of the function may occur. In the current stage, the recording function cannot synchronize with the update of the Database and cannot display the data in client side.

Individual Final Report: Co-runner

3.3 Schedule and Milestones

Task	Status	Deadline
Research and Environment Setup (Database, Python, Android studio)	Finished	Early November
Backend environment development	Finished	Late January
Application layout	Finished	Late December
Functions implementation	Finished	Late Feb
Testing	Finished	Mid April

Table 2 – Schedule and Milestones

There are five big phases. Firstly, the team implemented the environment setup which built the base of the system. In the second milestone, the team design and develop the logic of the backend, which is mainly in Python environment. Then, the team worked on the layout and logic flow of the application, followed by function implementation. And finally testing phase was carried out in April.

3.4 Future Development

3.4.1 Business corporation

To achieve the aim of the project – integrate daily workout (in particular jogging) with charity, it is necessary to find partners to work out the application. Currently, the team only consists of one person, which offer limited technical knowledge and business mindset. This may limit the potential development of co-runner. In future, this may better to cooperate with other partners in order to develop co-runner in larger scale.

3.4.2 Security

Security is one of the future developments. The login system of application can apply the SSL to tighten security. As the deploy of server to Google Cloud is quite late, there is a risk that there is no enough time to load all functions of application in reasonable time.

Individual Final Report: Co-runner

Section IV – Conclusion

Co-runner targets to build connection between sports, business sector and charity by encouraging business corporations to make donation on users' behalf through running.

Regarding the methodology sector, the Android-based application is a client-server model with the mobile application being the client. For the server part, Python is served as the backend system and MySQL and Google Cloud are the database system. The data manager program in Python receives request from client and retrieves data from database system and give to client

In the final stage, the team has completed the function implementation, connection of software environment. This allow the mobile application work with basic functions, for example, data recording and different pages. In this progress, the team made some alterations for the functions, including some function elimination, for example, Google API implementation.

Although the team cannot solve some problems expected before, for example, business partnership, data update between clients and servers. Still, the mobile application can achieve the basic functions which under the team expectation.

It is anticipated that the application can encourage more individual to work out and charity can receive sustainable donation.

Section V – Reference

Section VII –Reference

- [1] Sportsoho.(2017).[Online].Available: <https://www.sportsoho.com/pg/match>
- [2] K.Y.Lee (2013,July 18). Up to donors to make the most of their charity dollars [Online].Available:http://hkcss.org.hk/e//cont_detail.asp?type_id=2&content_id=821
- [3] M.Mikado; (2016, Sep 19). 7 Stats That Will Make You Rethink Your Document Management Strategy [Online]Available: <http://www.business.com/document-management-software/7-statistics-that-will-make-you-rethink-your-document-management-strategy/>

Individual Final Report: Co-runner

Appendix

The following figures are the use case diagrams which describe function implementation in Ch.2.4.2

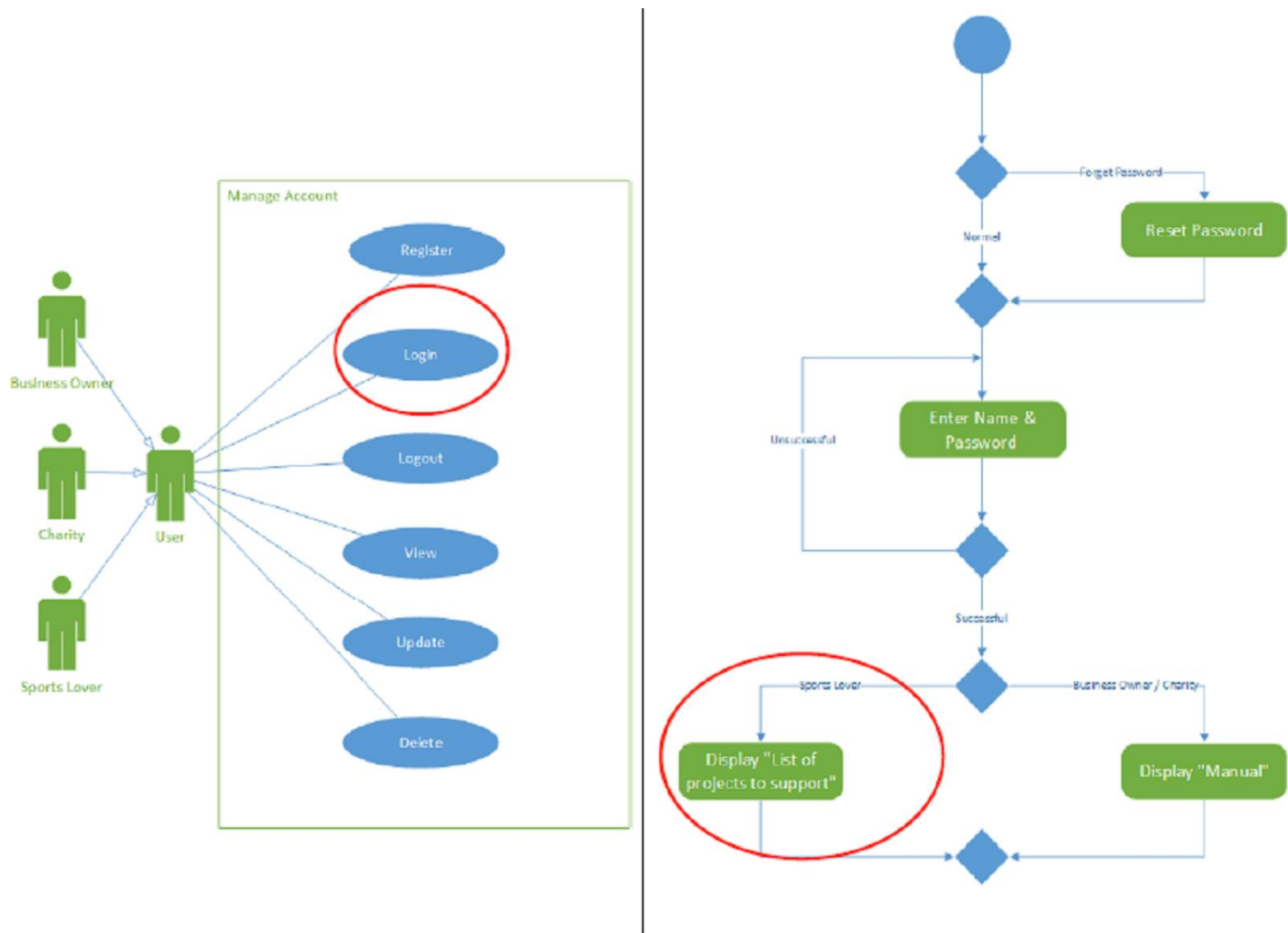


Figure 17- Use case diagram for login function and screen-captioned image of Login Page and Homepage (Larger pixel)

Individual Final Report: Co-runner

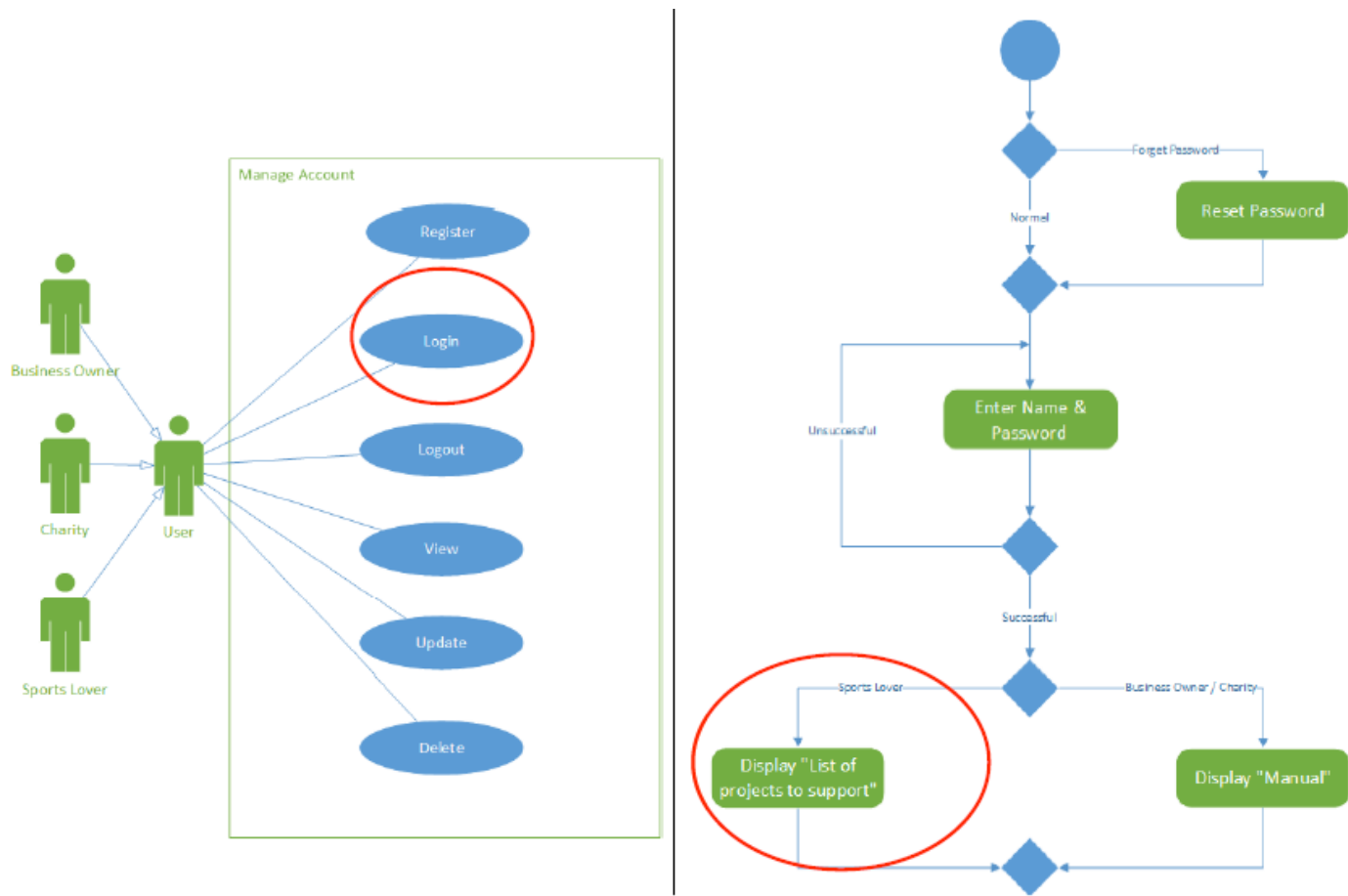


Figure 18- Use case diagram for function of charity selection (Larger pixel)

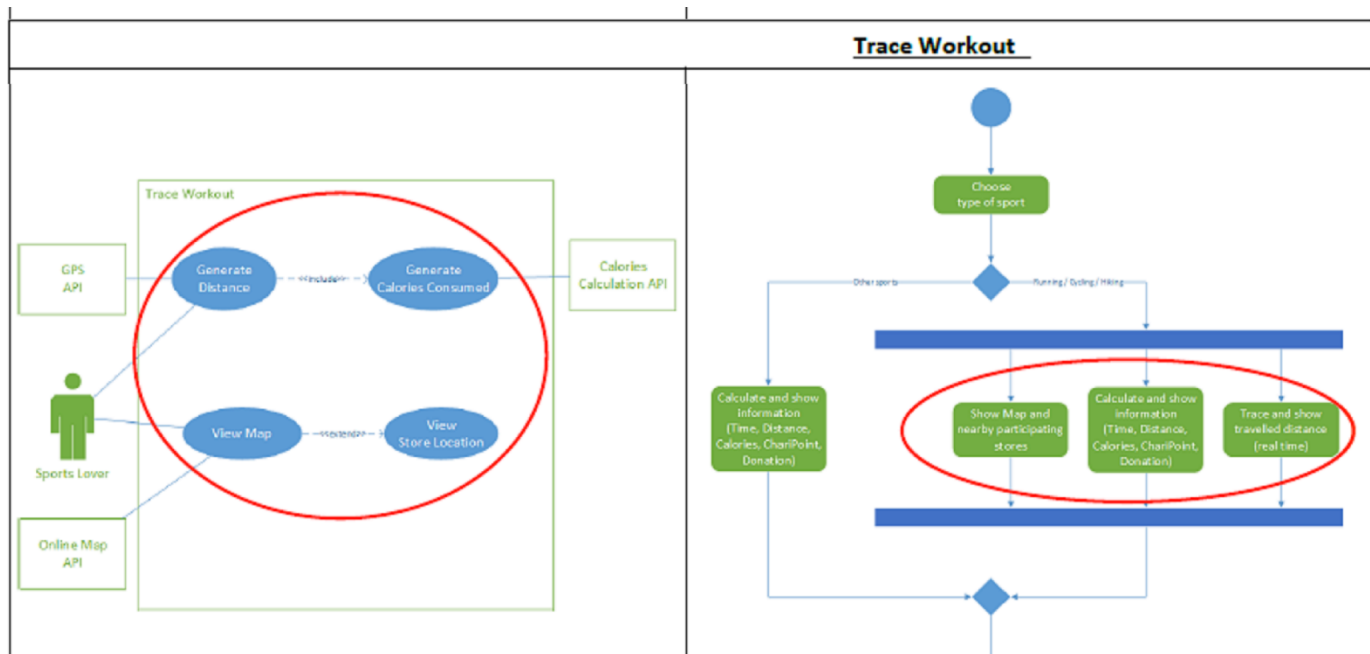


Figure 19– Use case diagram for function of displaying workout data (Larger pixel)