Mobile Peer-to-Peer Application for Resource Sharing

BY

LEE JIA YI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMMUNICATION AND NETWORKING

Faculty of Information and Communication Technology

(Perak Campus)

JAN 2016

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: MOBILE PEER-TO-PEER APPLICATION FOR RESOURCE SHARING

Academic Session: JAN 2016

I LEE JIA YI

declare that we allow this Final Year Project Report to be kept in University Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_____ _____

(Author's signature) (Supervisor's signature)

Address:
15, Lebuh Perajurit 2,
Ipoh Garden East,
31400, Ipoh, Perak.

_____

Supervisor's name

Date: _____ Date: _____

DECLARATION OF ORIGINALITY

We declare that this report entitled "MOBILE PEER-TO-PEER APPLICATION FOR RESOURCE SHARING" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : _____

Date : _____

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Mr Tan Teik Boon who has given me this bright opportunity to engage in this final year project entitled Mobile Peer-To-Peer Application for Resource Sharing. His knowledge and experiences enables me to have a deep understanding on develop mobile application procedure which helped me a lot in developing this application. I would like to thank him for his support and understanding during these final year project development process.

Last but not least, I would like to thank my parents and friends for their supports during the process. Without their support and encouragement, this project would not be completed.

ABSTRACT

Most of the resources sharing application was using traditional client server approach which the central point of failure problem exists in this approach. Moreover, smart devices is the trend in the market nowadays, user tend to do resources sharing by using their smart devices which is portable instead of restricted to only share resources to others using their desktop. A peer-to-peer (P2P) over Mobile Ad Hoc Network (MANET) mobile application able to solve these problems. P2P approach allow sharing of resources to be done without going through a centralized server. A mobile P2P over MANET application allow users to do resources sharing using their smart devices in a wireless environment. Interest Group sharing and file update notification features added to this application to make it more convenient for users. User can share resources to an Interest Group and all the members in the group will receive the resources at the same time. File updated notification function will notify all the members in the same group whenever there is an existing file updated, this can help user to reduce the time of keep tracking on the file version.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

P2P	Peer-to-Peer
MANET	Mobile Ad Hoc Network
WBLE	Web-Based Learning Environment
DHT	Distributed Hast Table
OSN	Online Social Network
IM	Instant Messaging
LAN	Local Area Network
MOS	Mobile Operating System
iOS	iPhone Operating System
GUI	Graphic User Interface
SDK	Studio Development Kit

CHAPTER 1 INTRODUCTION

1.1 Problem Statement

1. Central point of failure

Server outage problem exists in the traditional Client Server approach used in resources sharing. For example, lecturers and students in UTAR normally using Web-Based Learning Environment (WBLE) to share files and learning materials to be used in class. They encountered problem to login to WBLE when WBLE is under maintenance or too much people accessing it at the same time, they will not be able to upload or download materials from WBLE. This caused the learning efficiency of student decreased as they unable to get the learning materials before attending the class.

2. User unable to share resources to multiple user based on interest

Lecturer would prefer to share learning materials to a group of students who taking the same subject at the same time instead of sharing it out one by one. It is time consuming and inefficient.

3. Documents updating time to time

Learning materials from lecturers keep updating time to time, students faced problem to keep track of their current documents is it the latest version. It not only take time for students to update their document manually, it also might cause unnecessary errors as well.

1.2 Background Information

Peer-to-peer (P2P) is an approach to share resources instead of using the traditional Client Server approach, it allow peers or nodes in a network connect with each other and share resources without the need of a central server. Each computer or devices in the network can act as both server and client with the bandwidth and resources are distributed independently. (Hui, Lui and Yau, 2006, pp.2727-2746)

Resources sharing in mobile wireless environment continuously gain popularity in society nowadays due to the rapid increase in the usage of smartphone and devices. However, most of the existing P2P resources sharing technology were developed in wired

network. Existing P2P resources sharing technology is not able to work in wireless environment due the high mobility of the peers in a wireless network. Therefore, P2P over MANET was developed. Mobile Ad Hoc Network (MANET) is a collection of wireless mobile nodes formed a self-organizing temporary network with the absence of a predefined infrastructure or centralized administration. (Pucha, Das and Hu, 2004, p.1) P2P and MANET sharing the similarities of without having a centralized administration and infrastructure. Moreover, they are both distributed as well. (Gottron, König and Steinmetz, 2010, pp.505-532)

The sales of smartphones around the world almost reaches 1250 million units in 2014 (Statista, 2014), the popularity of smartphone expected to be continuously increase due to its mobility and advanced functions. Users have the freedom to move around without any restrictions in terms of location and perform a lot of actions at the same time just by using their smartphones or tablets which come along with powerful processors, sensitive touch screen and sensors, high resolutions camera and so on. (Jabbar, Ismail and Nordin, 2013, p.1)

A mobile application using P2P over MANET concept allow user to share resources from one to another without going through a centralized server by their mobile devices in a wireless environment. They can move freely without any restrictions. Users nowadays also more concern on how they can share a newly-taken photos or an updated documents or information to another devices in a very short period. For the user in an office or company, if the documents able to share from one device to another device in a short time and any updates or changes made in the document able to synchronize automatically can be consider as of great interest.

1.3 Motivation

Resources sharing usually using the traditional client server approach, problems occurred when the server fail to provide services. User unable to upload the resources to the server and will not be able to get the file from server if server down or temporary offline problem happened.

Smart devices gain large popularity in the market since few years ago, it will be a great interest where user can do resources sharing among one another by using their smart devices such as smart phones and tablets as they can carry it anywhere they like. User no longer like to be restricted to only do resources sharing by using their desktop.

On the other hand, most of the existing P2P over MANET applications only allow peers to perform one to one for resources sharing, which means one user can only share resources to only one user at one time. It would be more efficient if user can send resources to a group of users which commonly interested to that particular resources at the same time.

Documents may be modify by user time to time to ensure the content of documents is up to date. Users might face some hard time to keep track on the document one by one to make sure it is the latest version. User may save a lot of time if the files in their devices able to automatically synchronize and update themselves into the updated version.

1.4 Objectives

1. To have a solution for resources sharing using P2P approach

- By developing a mobile application implementing P2P over MANET, lecturers and students in class can form a temporary network in class and share learning materials instantly without the need of a central server. They still can share resources among each other although WBLE is under maintenance or having network traffic congestion problem.

2. To allow resources sharing based on sharing scheme like interest

- Interest Group Sharing feature allow lecturer to create multiple Interest Groups in the application which represent the group of students from different subjects. When lecturer wants to share the learning materials to a group of students, lecturer can select the particular Interest Group and share the learning materials, all the students in the same Interest Group can receive the learning materials at the same time.

3. To provide updated information on availability of new files or resources

- Learning materials from lecturers keep updating time to time to make sure that it is well organized and can be easily absorb by students. It is very tedious for student to manually keep track on the file version uploaded by lecturer. If the every file updated in the group will notify to the all the member in the same group, then they do not have to manually keep track on the file version.

1.5 Project Scope

A P2P over MANET application will be develop in order to make the resources sharing process between lecturers and students can be done without the need of a centralized server. Structured P2P system will used to develop this project as the performance of Structured P2P system is better compared to the Unstructured P2P system. They will be connected to same Local Area Network (LAN) using their mobile devices to share resources among each other.

Interest Group sharing will be added in this resources sharing application. This creates an easy way for users to share resources with different groups of users. One user can have multiple interest group in this application. User can a send a file which is commonly interested by a group of users at the same time. Besides, file update notification will be added so that user do not have to keep track on the latest file version. Once a user modified a document, other users which in the same group will be notify so that they can download the latest version of the file.

Besides, this Structured P2P over MANET for resources sharing application will be develop into mobile platform to increase the mobility of lecturers and students in a class. The mobile application will be built in the Android platform, because it is the most common used smartphone platform in the market due to many different smartphone manufacturers are using it. It is also an open sources, it will be easier to develop and implement.

1.6 Proposed Solution

P2P technology will be implemented in this project to allow resources sharing without the existence of a centralize server. P2P over MANET approach will be develop in this project to allow UTAR lecturers and students to share resources among each other in a wireless environment.

Most of the existing P2P file sharing application does not include the Interest Group sharing feature. This project going to include the Interest Group feature so user can group the user who shared the same interest and always share resources among themselves. save the Interest Group configuration so that user do not have to recreate it every time when using the application and can have more than one Interest Group.

Besides, file update notification features does not included in any of the existing P2P file sharing system. With this feature added, students do not have to manually keep track on the files version.

1.7 Achievement

This project provides a solution for lecturers and students in UTAR to do resources sharing among each other. The main contribution of this project is to avoid the central point of failure problem. WBLE might have maintenance and the server will be temporary offline, this caused lecturers unable to upload and students unable to download the learning materials which is going to be used in class from WBLE. With this application, the resources will be still available among each other.

With this resources sharing P2P over MANET application, lecturers and students can share their learning materials just by running this application in the class by forming a temporary network. By developing this in to mobile application they no longer restricted to do resources sharing by only using desktop.

Moreover, the Interest Group sharing features allow users in the network to have auto self-management in resources sharing. Lecturer can divide students from different subjects into different interest group in the application. Once the lecturer share a file to an

interest group, all the students in the same interest group can receive the file at the same time. This will increase the efficiency of file sharing between lecturer and students.

Furthermore, students do not have to manually keep track on the file version and worry about they do not get the latest version of the file because they will receive a notification whenever the file is updated and remind them to download the latest version of the file.

1.8 Report Organization

There are total of six chapters in this report. The first chapter will be basically the problem statement and main motivation of this project, come along with the background information and project objectives of the project.

The second chapter will be literature review of the project. Some research and studies that related to the project was done. Besides, four existing system in the market which are quite similar to this project has reviewed as well. The basic functionality, advantages and disadvantages of the four systems has been listed down.

Chapter three will be the system design chapter. The functionality of the project was explant in details. Use case, Activity, Use case description, Class, Sequence and Object diagram are used to describe the functionality of the system in detail. Moreover, the interface design of the system also included in this chapter. The implementation issues and challenges of this project was listed now.

Chapter four is the methodology and tools used to develop this project. The chosen methodology and tool of used is described clearly in this chapter. The basic requirements needed in order to allow the system run well stated in this chapter as well.

The implementation and testing of the system was included in chapter five. The basic setup of the project and some important source code of the functions were documented in this chapter. The test cases of the system also constructed in this chapter.

Last but not least the final chapter will be the conclusion. It concluded the whole project and explained the possible of the future implementation of the system.

CHAPTER 2 LITERATURE REVIEW

2.1 Literature Review

2.1.1 Peer-to-peer approach

Resources sharing normally is using the traditional Client Server approach, which consist lots of limitations. The biggest weakness is central point of failure exists, once the main server is down, the whole network will not be able to function. For instances, Apple iTunes and App Store were suffered from server outage at this March of 2015. (Rosenfeld, 2015) Even some company which using cloud as the server also facing the cloud server outages problem, such as Windows Azure went down for 5 hours on the August of 2014 and also the Amazon Web Services went offline for 2 hours on the November of 2014. (Tsidulko, 2014) This is the reason Peer to Peer approach was introduced. There are two types of P2P network which are Structured and Unstructured P2P network. Unstructured P2P network are further sub-categorized into three categories: Centralized P2P, Pure P2P and Hybrid P2P system. (Gottron, König and Steinmetz, 2010, pp.505-532)

For Centralized P2P system of the unstructured network, the network still consists of a central server in the network, but this central server not used to provide file sharing service, it only stored the address of the files located. It will only return the files' location (IP address of a particular node) to the requested node if the query of the file is found, so that the requested node know where to download the file. (Gottron, König and Steinmetz, 2010, pp.505-532) Napster is one of the most popular Centralized P2P system.

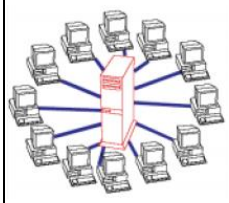
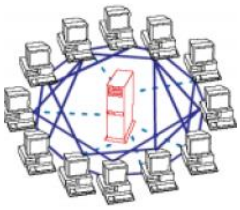
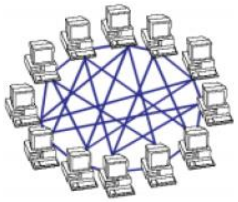
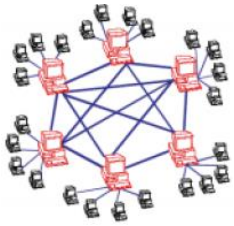
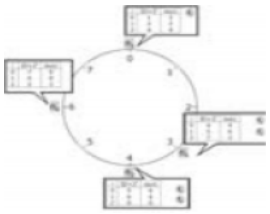
On the other hand, the Pure P2P system is completely decentralized, no any central entity is needed in the network. All nodes in the network have the equal responsibilities. Each node will only have the information of their neighbour node. Nodes in the network search for a particular requested file by flooding in the network. If the query file unable to match from their neighbour node, the neighbour will help to send out the query to other neighbours until the file is found. Examples of Pure P2P system are Gnutella and Freenet.

Hybrid P2P system nodes are separated into Super Node, act as the group leaders in the network and also normal node. A Super Node is only the "group leader" of a group of normal nodes. Super Node has more responsibility compared to normal node, they maintain the file lists and their respective location in their own group while normal node only request for files. KaZaA is one of the most well-known Hybrid P2P system.

Structured P2P networks using the Distributed Hash Tables (DHT) to assign each nodes in the network with a unique ID. Nodes in the network can be easily identify by using this ID. (Gottron, König and Steinmetz, 2010, pp.505-532) In Structured P2P the searching is based on the DHT to store the neighbours' information. The distribution of data is fair on each and every node, every nodes store almost the same amount of neighbours' information. The contribution of others active nodes in the network is required during file requesting process, as the routing table of each nodes only store the partial of the available active nodes in the network. (Gottron, König and Steinmetz, 2010, pp.505-532) The join and leave of nodes does not cause the whole list of address to remapping again, consistent hashing is the mapping method used by DHT. Chord, CAN, Tapestry are some popular Structured P2P system examples. Table 2.1 (Singh, Rizvi and Nainiwal, 2015, p.10) below summarized the Client Server network topology and the P2P network topology.

Besides file sharing, there are many applications are using P2P system in real life. P2P system is being applied in the IP Telephony and Instant Messaging (IM) field, Skype is one of the famous example. The bandwidth and the real time audio or video are shared as a resources. The server is only responsible for providing information and bill the user for charges calls. P2P system enable the IM communication to become more scalable and faster as the bandwidth was shared among users. (Readwrite.com, 2007) P2P system also contribute in video streaming field, such as P2PTV. Peers in the video streaming network will become the internet-connected host, they store and stream the video to the requested client. (Thampi, [no date], p.28)

Table 2.1: Summary of Client Server and P2P topology(Singh, Rizvi and Nainiwal, 2015)

<table>
<tr><th>Client Server</th><th colspan="4">Peer-to-Peer</th></tr>
<tr><td rowspan="4">➢ Central entity needed to store the resources
➢ Server have a higher performance</td><td colspan="4">➢ Peers can be client and server at the same time
➢ No central entity needed</td></tr>
<tr><td colspan="3">Unstructured P2P</td><td>Structured P2P</td></tr>
<tr><td>Centralized P2P</td><td>Pure P2P</td><td>Hybrid P2P</td><td>DHT</td></tr>
<tr><td>➢ Central entity is needed to keep track of location of peers
➢ Example: Napster</td><td>➢ No central entity needed
➢ Peers leave and join without any functionality lost
➢ Example: Gnutella, Freenet</td><td>➢ Consists of supernode and ordinary node
➢ Supernode have a higher performance
➢ Example: KaZaA</td><td>➢ No central entity needed
➢ Fixed connection in the overlay
➢ Example: Chord, CAN, Tapestry</td></tr>
<tr><td>  </td><td>  </td><td>  </td><td>  </td><td>  </td></tr>
</table>

The rapid increase of smart devices in the market leads to the increase of the resources sharing in the wireless environment. The existing P2P system was developed in a wired system and it is not suitable to apply in the wireless environment. Two challenges need to be concerned in P2P resources sharing in wireless network: the mobility of the peers in wireless environment and the leave and join of peers in P2P resources sharing network. (Chung-Ming Huang, Tz-Heng Hsu and Ming-Fa Hsu, 2007, pp.204-210) P2P over MANET was developed to overcome these problems.

2.1.2 Mobile Ad Hoc Network

MANET is defined by group of free moving nodes connecting each other through wireless connections without any existing infrastructure. The characteristics of MANET is it fully distributed and decentralized, independent from the predefined infrastructures, can perform multi-hop, low cost deployment and so on. End user devices was connected to the

network with fixed location traditionally. End user devices in MANET can free to move around, without staying in a fixed location. The routing infrastructure must be able to move following to the end user devices as the network topology and the addressing for the end user devices keep changing. (Corson, Macker and Cirincione, [no date], pp.64-65) Nodes in the MANET which is in the send range of each other can directly discovered and communicate with each other. (Hoebeke et al., [no date], pp.60-66) The MANET routing protocol is needed to indicate the responsible of every nodes in the MANET in order to discover the multi-hop paths for the data packets. (Mohanty, Thotakura and Ramkumar, 2014, pp.192-206)

2.1.3 P2P over MANET

P2P over MANET faces more challenges due to the MANET topology changes frequently because of the mobility of nodes in the network. (Klemm, Lindemann and P. Waldhorst, [no date]) P2P over MANET is in which the P2P from the application layer of the ISO model overlay on top of the MANET from the network layer of the ISO model. (Gottron, König and Steinmetz, 2010, pp.505-532) The specialized functions that are not possible in the IP layer of the ISO model justified this overlay network. The functions of the application layer is to share resources among devices in order to carry out distributed computations and fulfil needs of user. (Prof. Rekha and Ashwini, 2012, p.214)

2.1.4 Interest Group Sharing

Resources sharing in P2P environment usually do not share resources to multiple user based on interest. A great interest created where a peer can share a commonly interested resources to a group of peers, called Interest Group. Peers in the same Interest Group can receive the resources at the same time. This can be implemented by applying the hybrid architecture which combining both centralized and decentralized characteristics. (Felix, Sebastian and Axel, n.d.) The resources sharing between users using P2P approach and a server just used to store information of about the Interest Group such as Interest Group name, Interest Group member and others. The server does not involve in the resources sharing process.

2.1.5 Android Platform

Mobile Operating System (MOS) is the operating system that used by smart devices as a platform to run the application and program. (Mohd Shahdi et al., 2013) There are several types of MOS in the market: Android, iOS, Windows, BlackBerry OS and others, whereas Android and iOS are more dominated in the worldwide smartphone market nowadays. Figure 2.1 shows the worldwide smartphone operating system market share (in unit shipments) in first quarter of 2012, 2013, 2014 and 2015. 78% of the market in first quarter of 2015 is dominated by Android. The biggest factor is Android operating system not only used by one smart devices manufacturer. It used by Samsung, Sony, HTC, Huawei, XiaoMi, Lenovo, Asus and so on, but for others MOS such as iOS only used by iPhone and iPad, Windows only used by Nokia. Android having a larger market to target different types of smart devices users compared with other MOS. Android is also an open source operating system to allow developers to have more freedom in implementing their solution. This project will be develop into an Android based application in order to target more users and have more freedom in the development process.

Period	Android	iOS	Windows Phone	BlackBerry OS	Others
Q1 2015	78.0%	18.3%	2.7%	0.3%	0.7%
Q1 2014	81.2%	15.2%	2.5%	0.5%	0.7%
Q1 2013	75.5%	16.9%	3.2%	2.9%	1.5%
Q1 2012	59.2%	22.9%	2.0%	6.3%	9.5%

Figure 2.1: Worldwide Smartphone Operating System Market Share (in unit shipments) in first quarter of 2012, 2013, 2014 and 2015 (www.idc.com, 2015)

2.2 Critical Remarks of Previous Work

2.2.1 File Transfer

File Transfer is a peer-to-peer Android mobile application to send and receive files between smart devices such as smartphones and tablets. File Transfer allow user to transfer any types of files such as documents, pictures, video, audio files, apk files, and so on. Files are directly transferred within users in the same Local Area Network (LAN) without uploading it to a server and require user to download it from the server again. Sender can select the file that wish to transfer to other devices from the local storage, and the Receiver can just enter the "Receive Files" mode in order to retrieve the files. Figure 2.1.1a and Figure 2.1.1b show the interface of File Transfer.

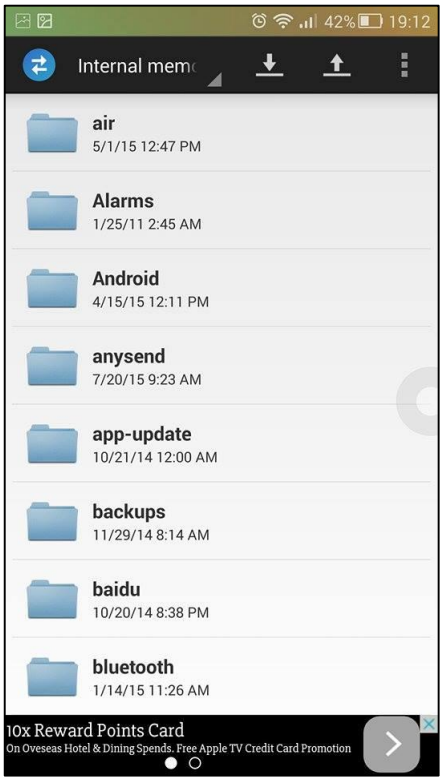


Figure 2.2.1a: Search files from internal storage page of File Transfer

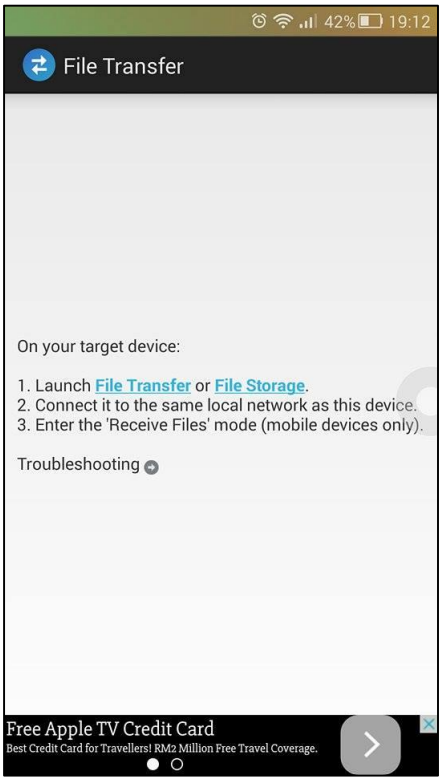


Figure 2.2.1b: Tutorial page of File Transfer

Strength

The files from internal storage were listed out clearly by folders in File Transfer, user can search the particular files that wish to be shared by entering into the folder which contain the file. There is no any complicated configuration steps needed regardless for sender or receiver to share or receive the files.

Weakness

The files in the internal storage are not sorted according to file type in File Transfer (document, photos, audio or video), it is not that convenient for user to search for a particular file to be share as user have to know which folder the particular file exactly located at. User each time can only select one file for sharing, it caused trouble when user wants to share few files at the same time. The Graphical User Interface (GUI) is too simple and not much details a user can modify such as user device name. File sharing among interest group and file synchronization function is not available in File Transfer.

2.2.2 Shoot

Shoot is an Android File sharing application launched by BitTorrent Sync's peer-to-peer technology. Shoot allows user to send photo and video in a large file size without going through any server and cloud. By using Shoot, only the first three sends are free, user have to purchase in order to enjoy the unlimited sending of files. There is free for receiving files. User can select any photo and video from the local storage and share it, once the file selected, a QR code will be generated. Receiver can just receive the files from sender by scanning the QR code from the device of the sender. Both sender and receiver does not required to connect to the same LAN. Figure 2.1.2a and Figure 2.1.2b show the interface of File Transfer.

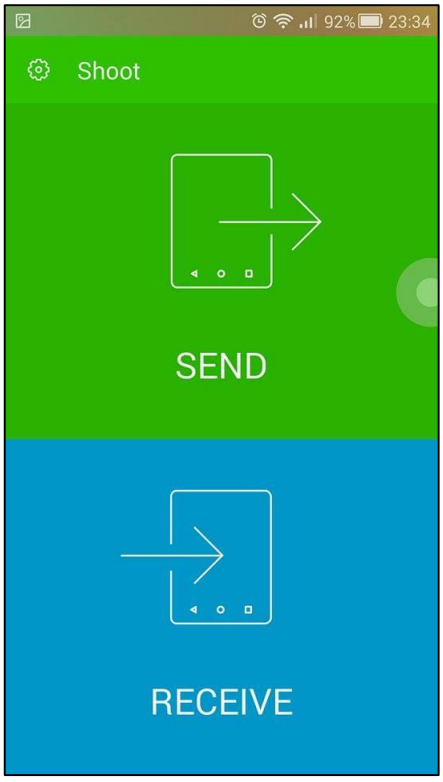


Figure 2.2.2a: Home Page of Shoot

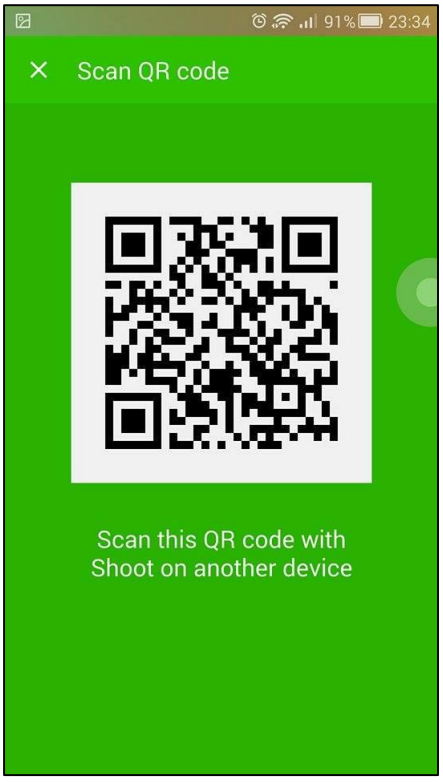


Figure 2.2.2b: Generated QR code when a Sender selected files to be share in Shoot

Strength

The GUI of Shoot is simple but it is clear and easy to allow user to use it without any confusion. User can just simple select the SEND and enter the select file page to select the file that wish to be share to other user and select the RECEIVE to scan QR code from sender in order to receive the file. The speed of transferring the files from one device to another device is fast although the file size of photo or video is large. User can also share more than one files to another device at a same time.

Weakness

Shoot only support sharing of photo and video. Besides, sender unable to share photo or video to few devices at a same time. Both sender and receiver must be physically near to each other as it required scanning of QR code. Shoot also required In-App Purchase in order to unlimited send photo and video. Moreover, Shoot does not include the file sharing in interest group and the file synchronization feature.

2.2.3 Send Anywhere (File Transfer)

Send Anywhere published by Estmob Inc. is also a peer-to-peer mobile application in Android based to securely send files between smart devices without passing through any server. Send Anywhere support any types of file for sharing. User can choose to create an account or not to create an account to use this application. Without creating an account, the user device name will be determined by using the google account email's username. User can select files that wanted to be share, then a 6 digit key code will be generated. All the nearby devices will be listed out once the 6 digit key code generated. Nearby devices can key in the 6 digit key code in the "Receive File" field to directly get the files from sender. Both sender and receiver required to be connected to a same LAN. For user for create an account in Send Anywhere, they can transfer files among themselves without keying in the 6 digit key code. Files transfer history is also available in Send Anywhere. Figure 2.1.3a and Figure 2.1.3b shows the interface of Send Anywhere.

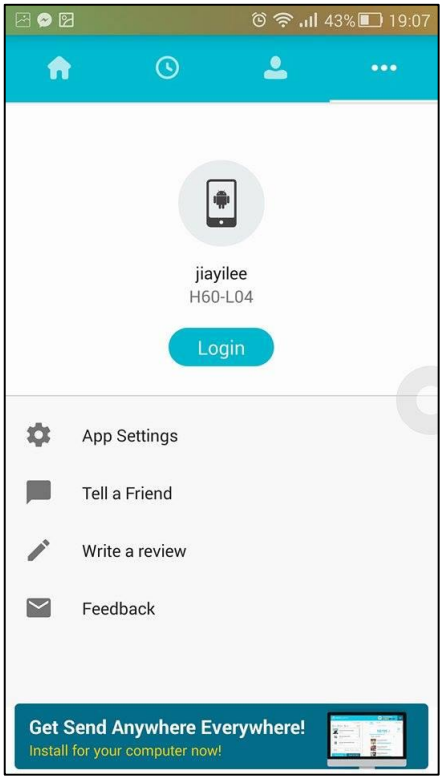


Figure 2.2.3a: User details page of Send Anywhere

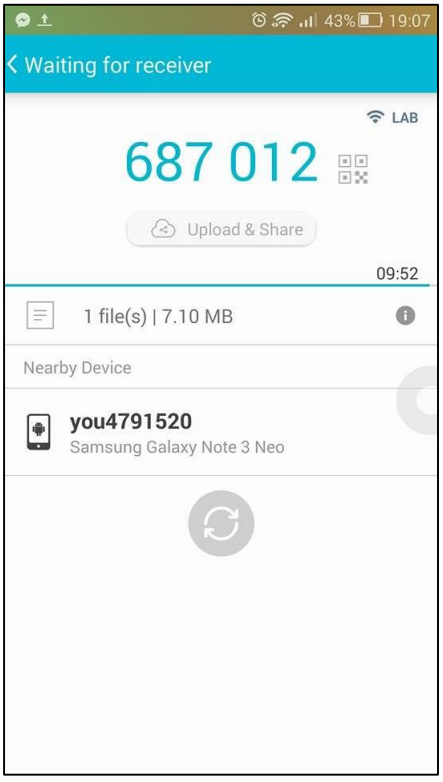


Figure 2.2.3b: 6 digit key code generated and nearby devices page of Send Anywhere

Strength

Send Anywhere provide a neat GUI for user to do files transfer efficiently. The files from the local storage were sorted according to file types, user can easily pick the files they wish to share based on the file type. Sender can select more than one files and share it at a same time. Moreover, sender can also share the files selected to multiple nearby devices which connected to the LAN at a same time. This reduced the time of file sharing among multiple user.

Weakness

Receiver who does not have a Send Anywhere account cannot receive the files shared by sender if he or she does not have the 6 digit key code. Besides, the file sharing in interest group and file synchronization features are not available in Send Anywhere as well.

2.2.4 Xender

This is a file sharing application released by Xender Team on 5th of March 2015. Xender can transfer documents, music, photos, videos and even application from one device to another device. File sharing between devices in Xender can be done without connecting all the devices in the same LAN. The main attraction of Xender is it can do group file sharing up to four devices, which mean sender can create a group up to four person and share them the file which the four of them interested with at the same time. The files from the local storage are arranged according to the file types, after sender select the files that wanted to be share, a group will be created and Xender will automatically turn on the Wi-Fi hotspot of the sender device. Devices nearby can join the sender group by connect to the Wi-Fi hotspot of the sender, once the receiver connected to the sender's Wi-Fi hotspot, the files will be automatically transfer from sender to receiver. The feature of file transfer history between devices us available in Xender. Figure 2.1.4a and Figure 2.1.4b shows the interface of Xender.

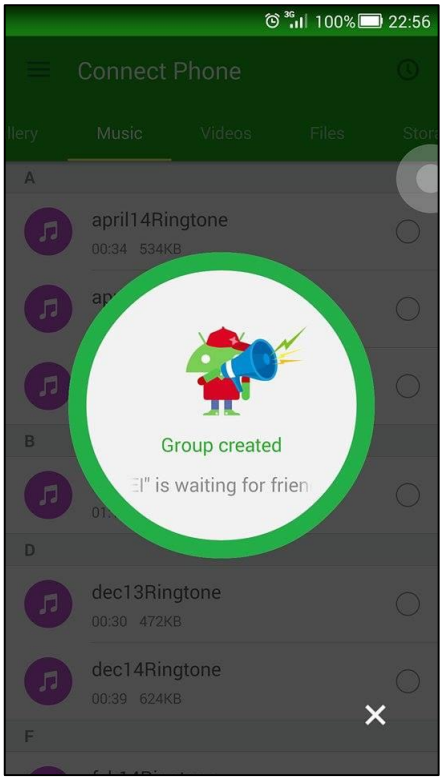


Figure 2.2.4a: Group created page when user selected the files that wanted to be share in Xender

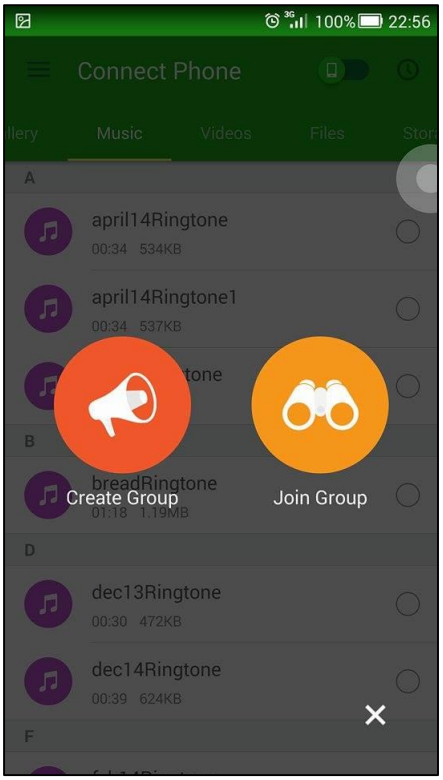


Figure 2.2.4b: User can create or join a group in Xender

Strength

File sharing in Xender does not required user to connect to the same LAN. Sender can create a group and allow up to four devices which interested to get the same share files to join the group. These devices able to get the files at the same time. The GUI of Xender is neat and user-friendly, user can create or join group without going through complicated steps.

Weakness

The group configuration does not save, which means sender have to create the group every time when using this application, and each time only one group which can fit up to four devices can be created. File synchronization features is not available in this application.

Table 2.2: Comparison Table for existing systems with proposed solution

Application / Features	File Transfer	Shoot	Send Anywhere	Xender	Mobile P2P Application
Free	✔	✘	✔	✔	✔
User Friendly	✘	✔	✔	✔	✔
Support ALL file type for file sharing	✔	✘	✔	✔	✔
Files from local storage sort according to file type	✘	✔	✔	✔	✔
Both sender and receiver have to be in the same LAN	✔	✘	✔	✘	✔
Interest Group Sharing	✘	✘	✘	✔	✔
File update notification	✘	✘	✘	✘	✔

CHAPTER 3 SYSTEM DESIGN

3.1 Requirement Specifications

3.1.1 Functional Requirement

1. User should be able to view the file directory of their device.
2. User should be able to select the resource that wanted to be shared from their file directory.
3. User should be browse the list of the shared resources available in the network.
4. User should be able to search a particular shared resources available in the network by using resource name.
5. User should be able to download the shared resources available in the network.
6. User should be able to create an interest group.
7. User should be able to manage the interest group.
8. User should be able to view the members in the interest group.
9. User should be able to share resources among group members.
10. User should be able to receive notifications when member in a particular interest group shared a file.

3.1.2 Non Functional Requirement

1. The system should be able to allow user to share resources with each other without using a server.
2. The system should be able to store the interest group information to Parse cloud database.
3. The system should be able to retrieve the interest group available to user from Parse cloud database.
4. The system should be able to verify the group owner before the user edit the interest group information.
5. The system should be able to update the details of the interest group to the Parse cloud database after user edited and save the information.
6. The system should be able to push a notification to all the members in a particular group when a member in the group shared a file.
7. The system should be able to notify all the members in a particular interest group when the group owner shared an updated file.

3.2 System Specification

The system design of this mobile P2P over MANET application for resources sharing will be explant more details in the section below. The interaction between user and the system will representing by using use case diagram, activity diagram, use case description and others.

3.2.1 Use Case Diagram

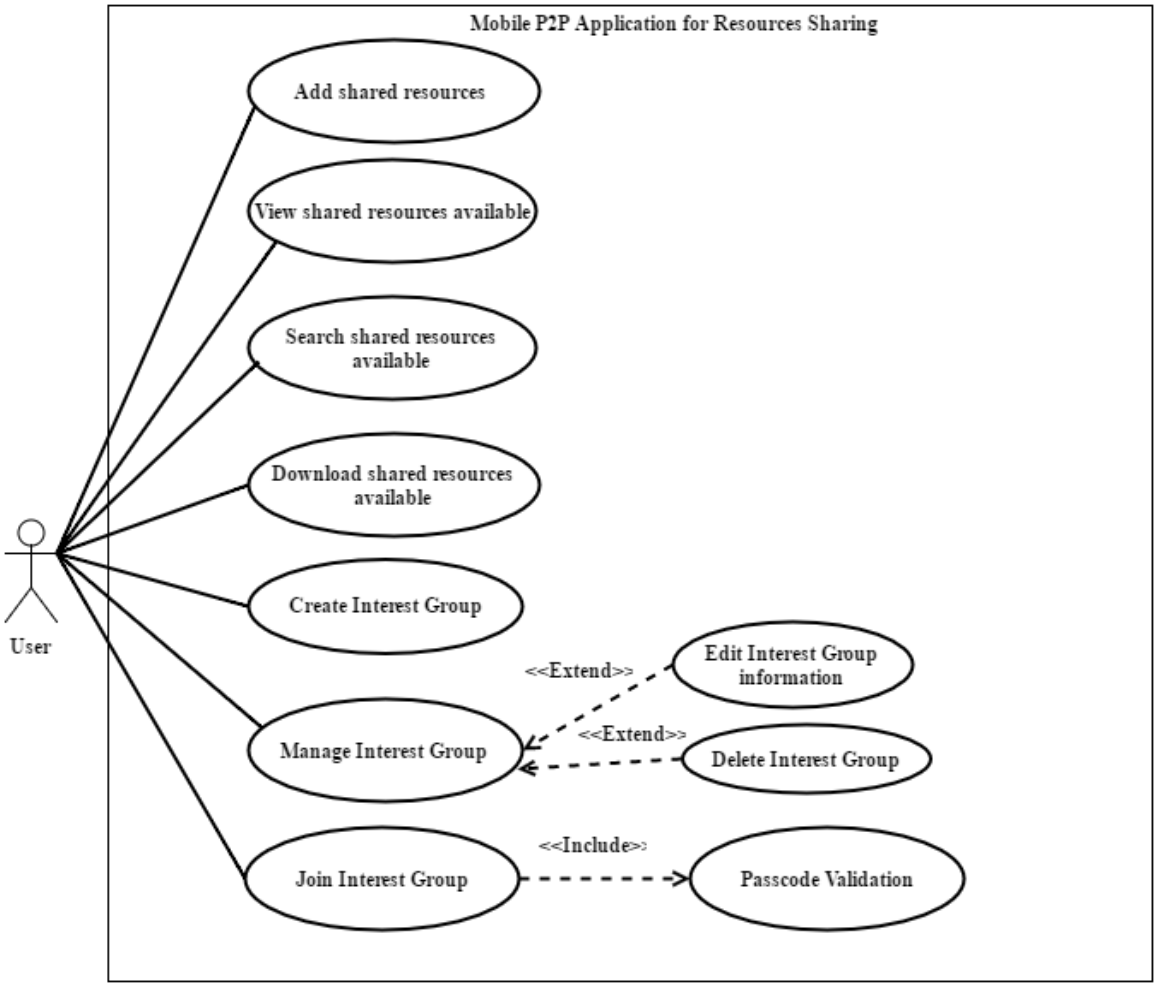


Figure 3.2.1: Use Case Diagram of Mobile P2P Application for Resources Sharing

3.2.2 Activity Diagram

Figure 3.2.2: Add Shared Resources Activity Diagram of Mobile P2P Application for Resources Sharing

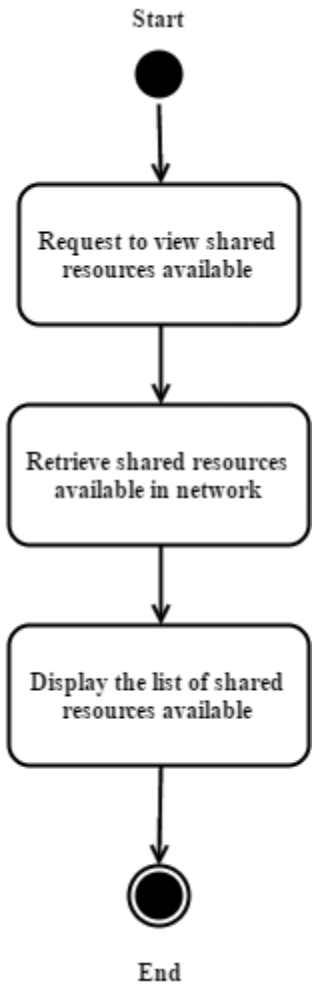


Figure 3.2.3: View Shared Resources Available Activity Diagram of Mobile P2P Application for Resources Sharing

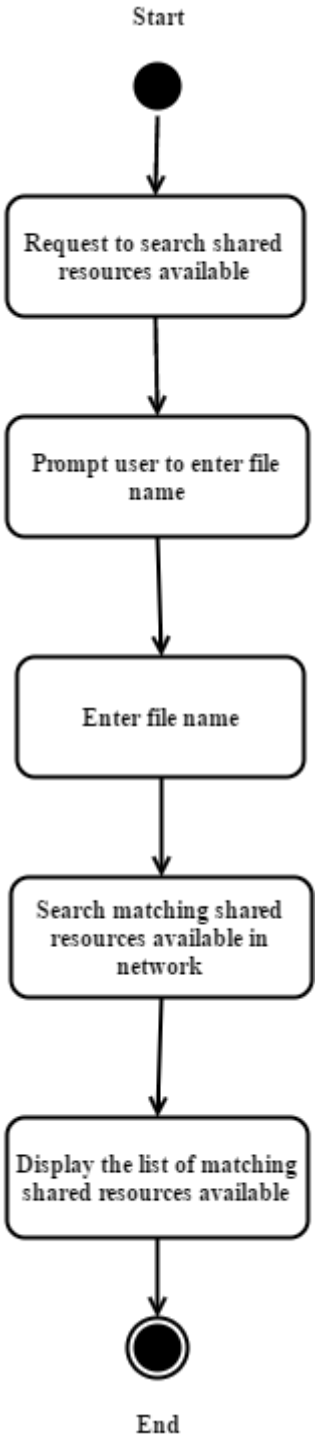


Figure 3.2.4: Search Shared Resources Available Activity Diagram of Mobile P2P Application for Resources Sharing

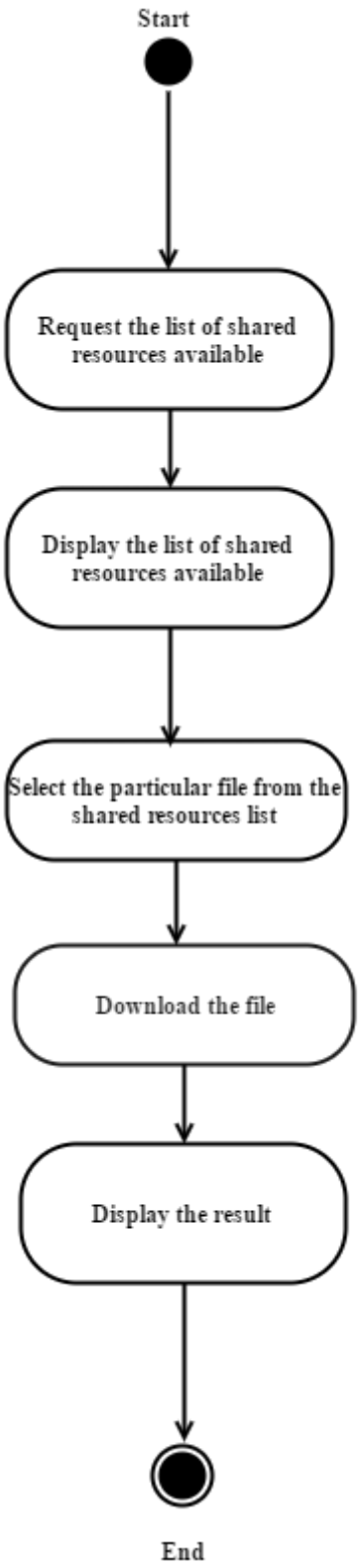


Figure 3.2.5: Download Shared Resources Available Activity Diagram of Mobile P2P Application for Resources Sharing

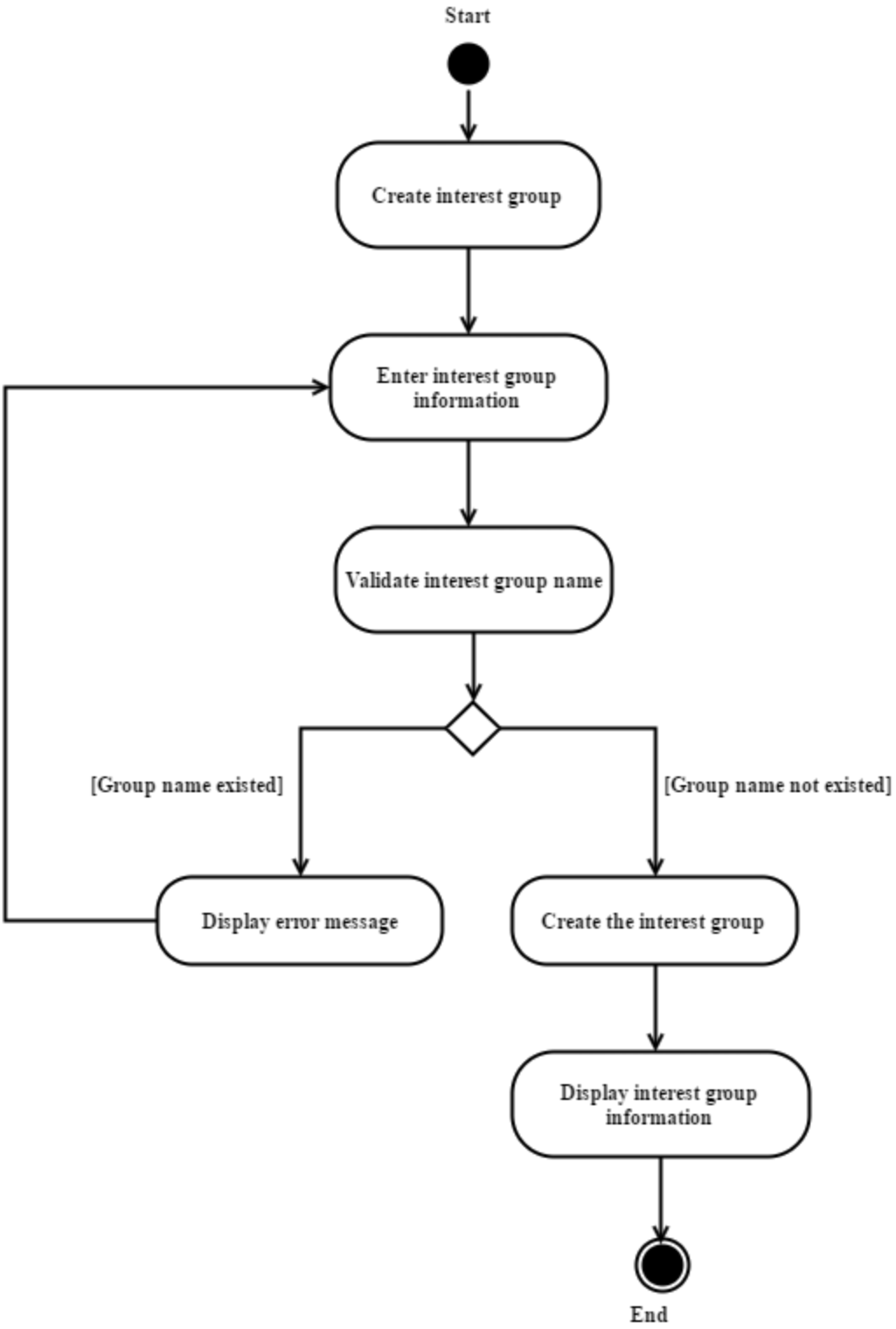


Figure 3.2.6: Create Interest Group Activity Diagram of Mobile P2P Application for Resources Sharing

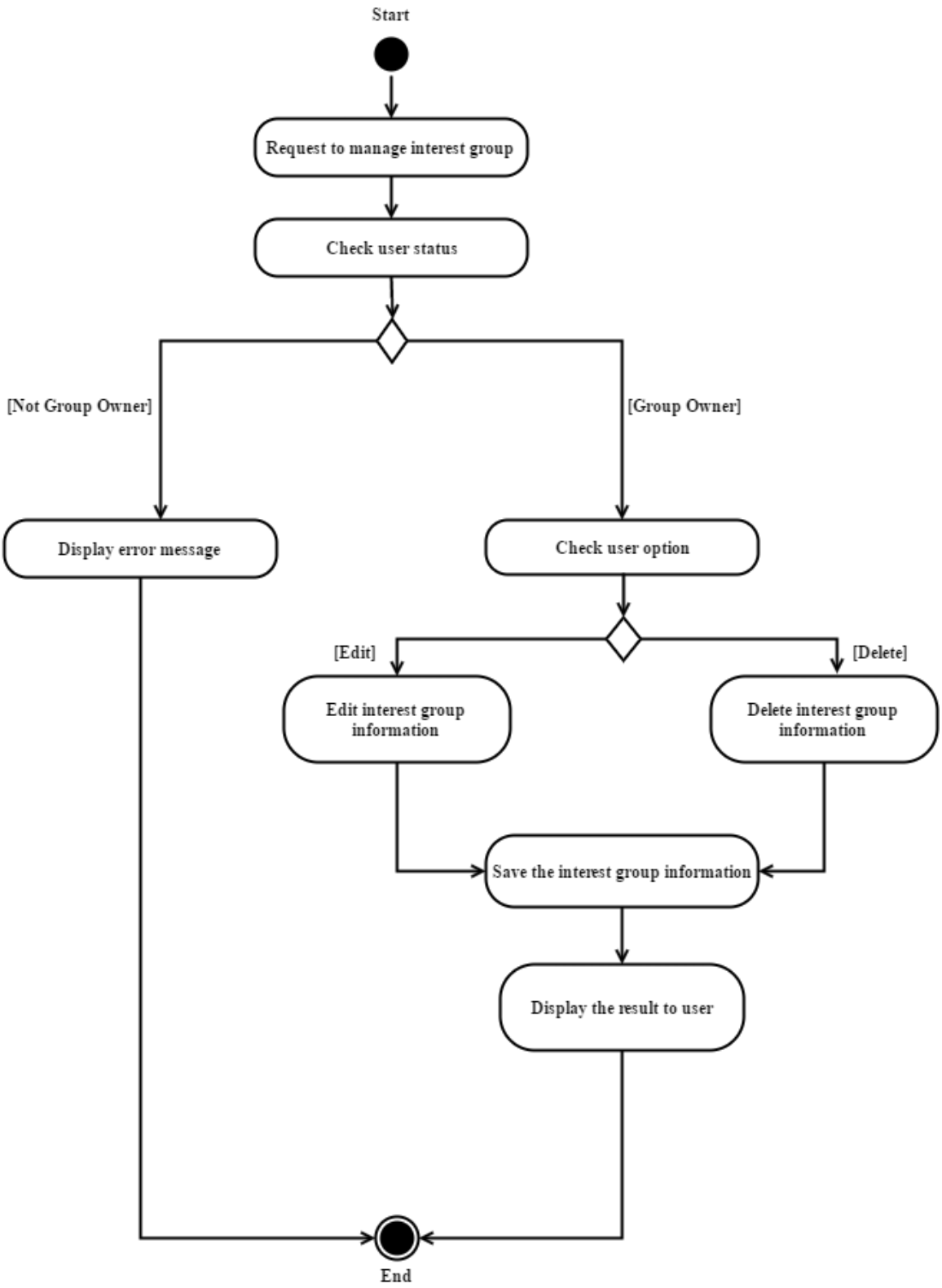


Figure 3.2.7: Manage Interest Group Activity Diagram of Mobile P2P Application for Resources Sharing

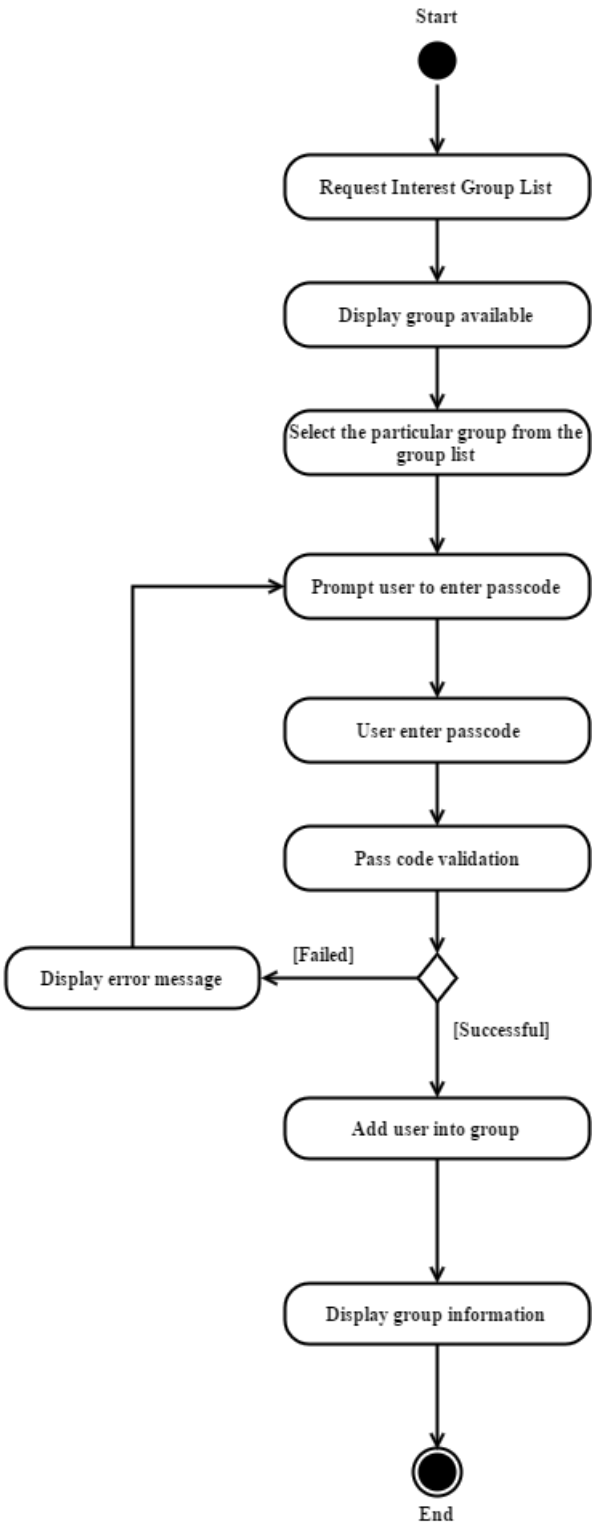


Figure 3.2.8: Join Interest Group Activity Diagram of Mobile P2P Application for Resources Sharing

3.2.3 Use Case Description

Use Case Name: Add Shared Resources	**ID:** 01	**Importance Level:** High
Primary Actor: User	**Use Case Type:** Details, Essential	
Stakeholders and Interests: User – select resources that wanted to share in the network		
Brief Description: This use case describes how the system handles the process of add the resources wanted to be shared into the network		
Trigger: User wants to add resources that wanted to be share into the network from local storage **Type:** External		
Relationships: **Association:** User **Include:** - **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. The user will enter the file directory. 2. The system will retrieve the file directory of the user. 3. The system will display the files directory that available in the local storage. 4. The user select the files that wish to be added into the shared resources list. 5. The system will add the selected resource to the shared resources list. 6. The system will display the file added results to user.		
Sub Flows: Not applicable.		
Alternate / Exceptional Flows: -		

Use Case Name: View Shared Resources Available	**ID:** 02	**Importance Level:** High
Primary Actor: User	**Use Case Type:** Details, Essential	
Stakeholders and Interests: User – view the shared resources by other users in the network		
Brief Description: This use case describes how the system handles the process of view shared resources by other users in the network.		
Trigger: User wants to view the shared resources by other users in the network **Type:** External		
Relationships: **Association:** User **Include:** - **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. The user request to view the shared resources by other users in the network. 2. The system retrieve the shared resources by all the user in the network. 3. The system display the list of shared resources available in the network to the user.		
Sub Flows: Not applicable.		
Alternate / Exceptional Flows: -		

Use Case Name: Search Shared Resources Available	**ID:** 03	**Importance Level:** High
Primary Actor: User	**Use Case Type:** Details, Essential	
Stakeholders and Interests: User – search the shared resources available in the network		
Brief Description: This use case describes how the system handles the process of search the shared resources available in the network.		
Trigger: User wants to search the shared resources available in the network. **Type:** External		
Relationships: **Association:** User **Include:** - **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. The user request to search shared resources available in the network. 2. The system prompt user to enter the file name in order to search. 3. The user will enter the file name that wanted to search. 4. The system search the matching shared resources available in the network. 5. The system display the list of matching shared resources available.		
Sub Flows: Not applicable.		
Alternate / Exceptional Flows: -		

Use Case Name: Download shared resources available	**ID:** 04	**Importance Level:** High
Primary Actor: User	**Use Case Type:** Details, Essential	
Stakeholders and Interests: User – download the shared resources available in the network.		
Brief Description: This use case describes how the system handles the process of download the shared resources available in the network.		
Trigger: User wants to download the shared resources available in the network. **Type:** External		
Relationships: **Association:** User **Include:** - **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. The user will request the list of shared resources available. 2. The system will display list of shared resources available in the network. 3. The user select the particular file from the shared resources list. 4. The system will download the file for user. 5. The system will display the result to user.		
Sub Flows: Not applicable.		
Alternate / Exceptional Flows: -		

Use Case Name: Create Interest Group	**ID:** 05	**Importance Level:** High
Primary Actor: User	**Use Case Type:** Details, Essential	
Stakeholders and Interests: User – create new interest group		
Brief Description: This use case describes how the system handles the process of user creating a new interest group.		
Trigger: User wants to create a new interest group. **Type:** External		
Relationships: **Association:** User **Include:** - **Extend:** - **Generalization:** -		
Normal Flow of Events: 1. The user request to create a new interest group. 2. The system prompt user to enter the new interest group information. 3. The user enter the interest group information. 4. The system validate the group name. 5. The system display the interest group information.		
Sub Flows: Not applicable.		
Alternate / Exceptional Flows: 4a. The system verify that the group name existed, display an error message and back to Step 2. 4b. The system verify that the group name not existed, the system will create the interest group.		

Use Case Name: Manage Interest Group	**ID:** 06	**Importance Level:** High
Primary Actor: User	**Use Case Type:** Details, Essential	
Stakeholders and Interests: User – manage the interest group		
Brief Description: This use case describes how the system handles the process of managing the interest group		
Trigger: User wants to manage the interest group **Type:** External		
Relationships: **Association:** User **Include:** - **Extend:** Edit Interest Group Information, Delete Interest Group **Generalization:** -		
Normal Flow of Events: 1. The user request to manage the interest group. 2. The system check for user status. 3. The system check for user option. 4. The system will save the interest group information. 5. The system will display the result to user.		
Sub Flows: Not applicable.		
Alternate / Exceptional Flows: 2a. The system verify that the user is a group owner, the system continue by Step 3. 2b. The system verify that the user is not a group owner and display error message. 3a1. The system will prompt user to edit the interest group information. 3b1. The system will allow user to delete the interest group information.		

Use Case Name: Join Interest Group	**ID:** 07	**Importance Level:** High
Primary Actor: User	**Use Case Type:** Details, Essential	
Stakeholders and Interests: User – request to join an interest group		
Brief Description: This use case describes how the system handles the process of user request to join an interest group		
Trigger: User request to join an interest group **Type:** External		
Relationships: **Association:** User **Include:** Passcode Validation **Extend: -** **Generalization: -**		
Normal Flow of Events: 1. The user will request the list of interest group. 2. The system display the list of available interest group. 3. The user will select the particular group that wish to join from the list. 4. The system will prompt user to enter the passcode. 5. The system will perform passcode validation.		
Sub Flows: Not applicable.		
Alternate / Exceptional Flows: 4a1. The system will verify that the passcode validation is successful. 4a2. The system will add the user into the selected group. 4a3. The system will display group information. 4b1. The system will verify that the passcode validation is failed. 4b2. The system will display error message to the user.		

3.2.4 Class Diagram

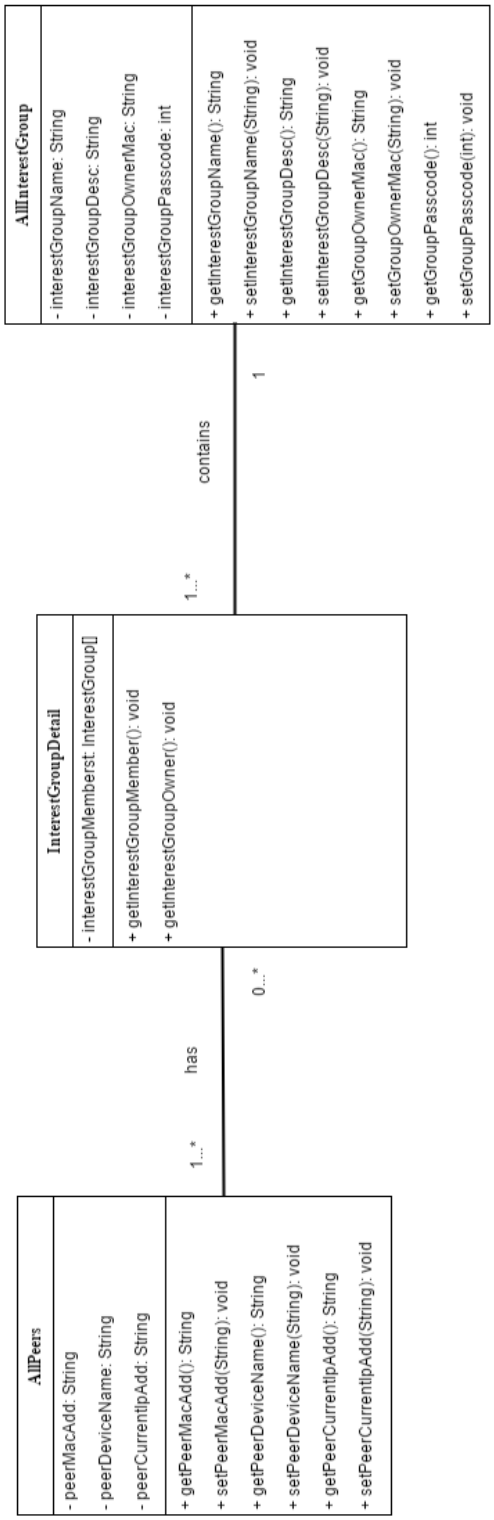


Figure 3.2.10: Class Diagram

3.2.5 Sequence Diagram

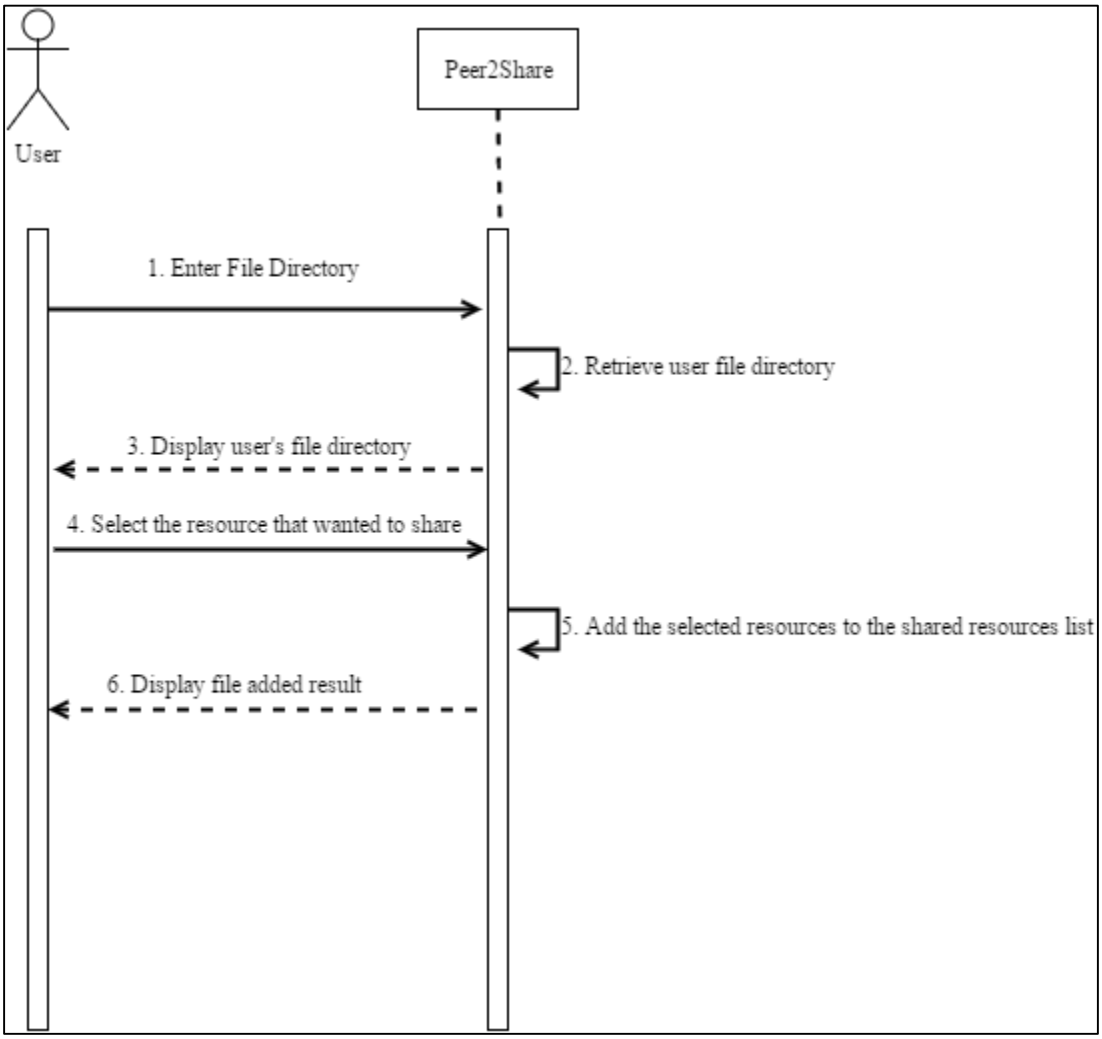


Figure 3.2.11: Sequence Diagram of Add Shared Resources

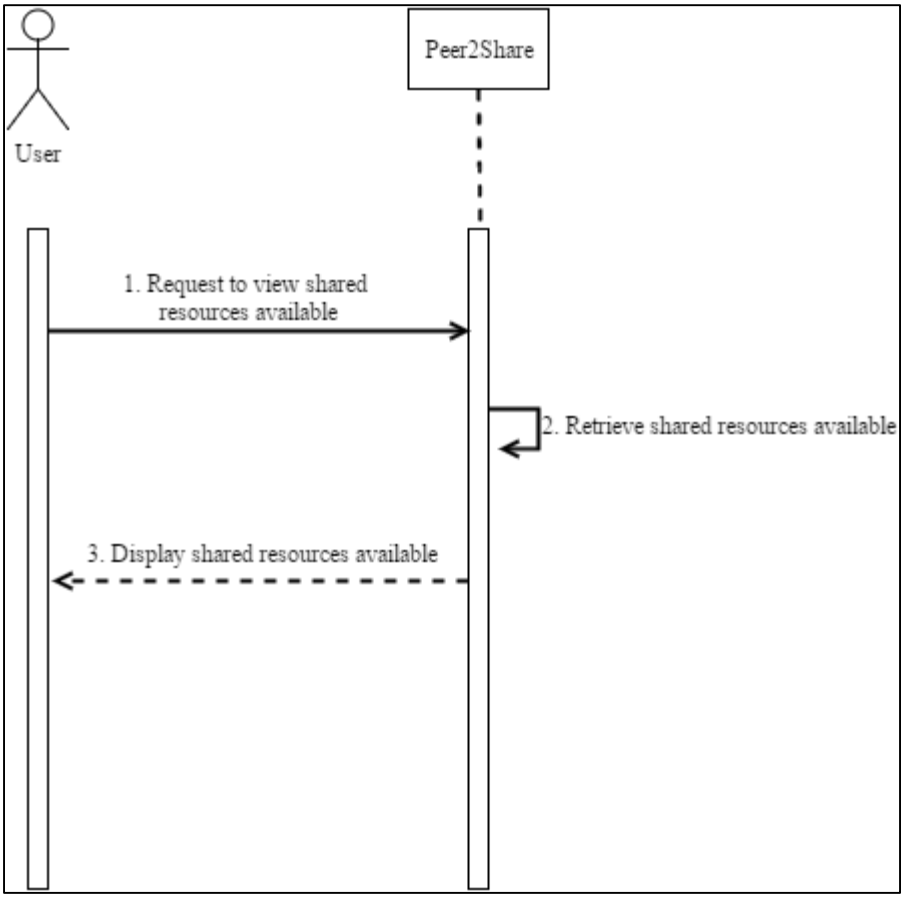


Figure 3.2.12: Sequence Diagram of View Shared Resources Available

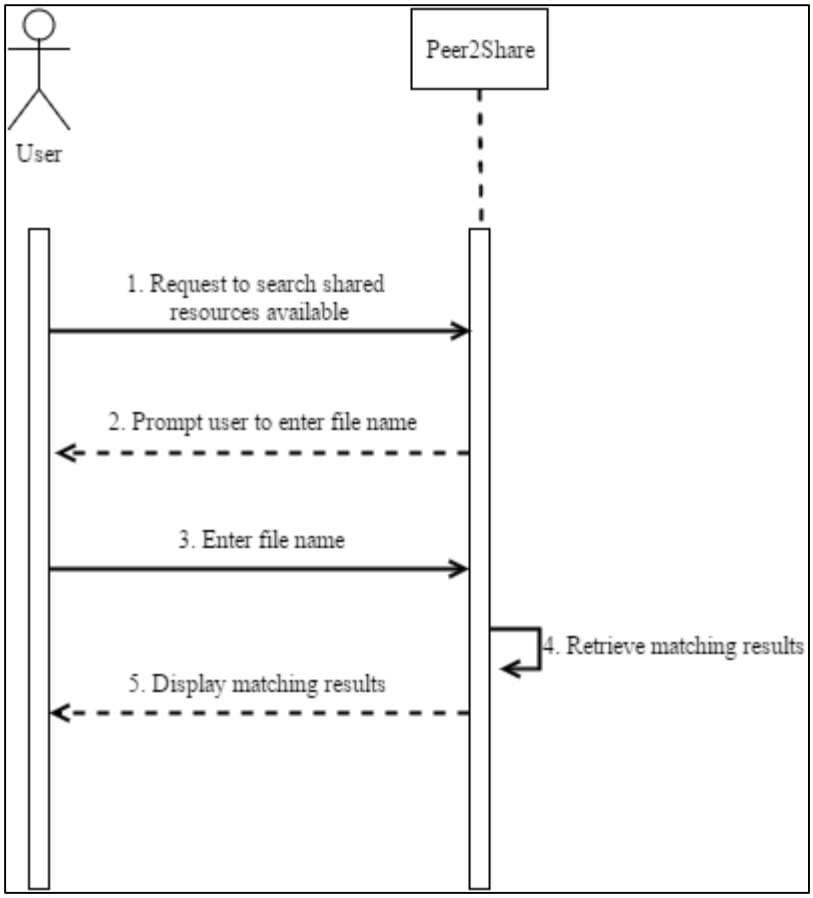


Figure 3.2.13: Sequence Diagram of Search Shared Resources Available

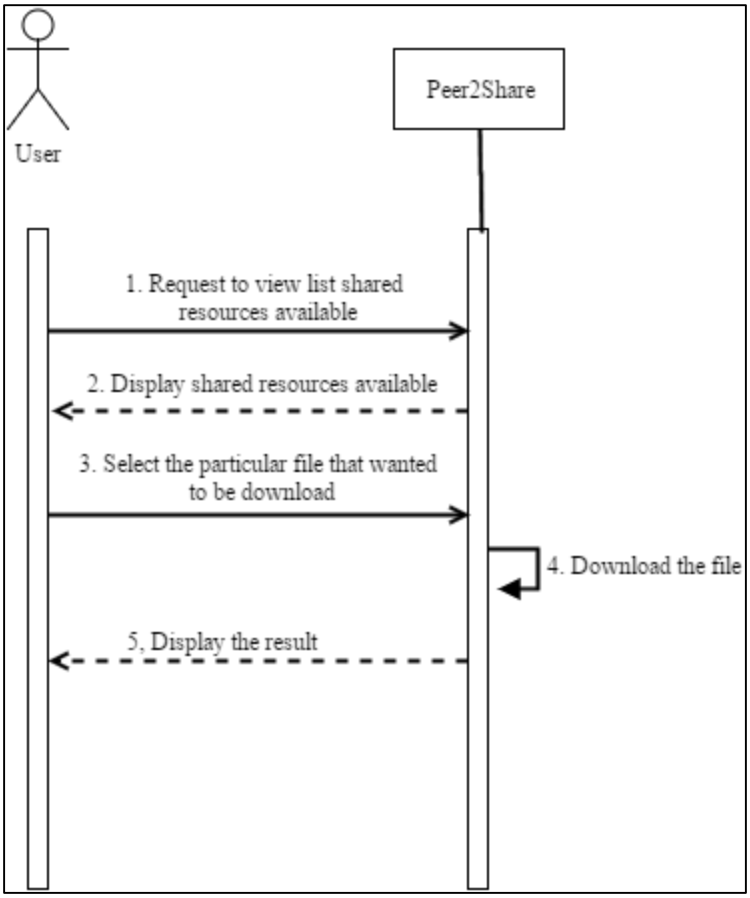


Figure 3.2.14: Sequence Diagram of Download Shared Resources

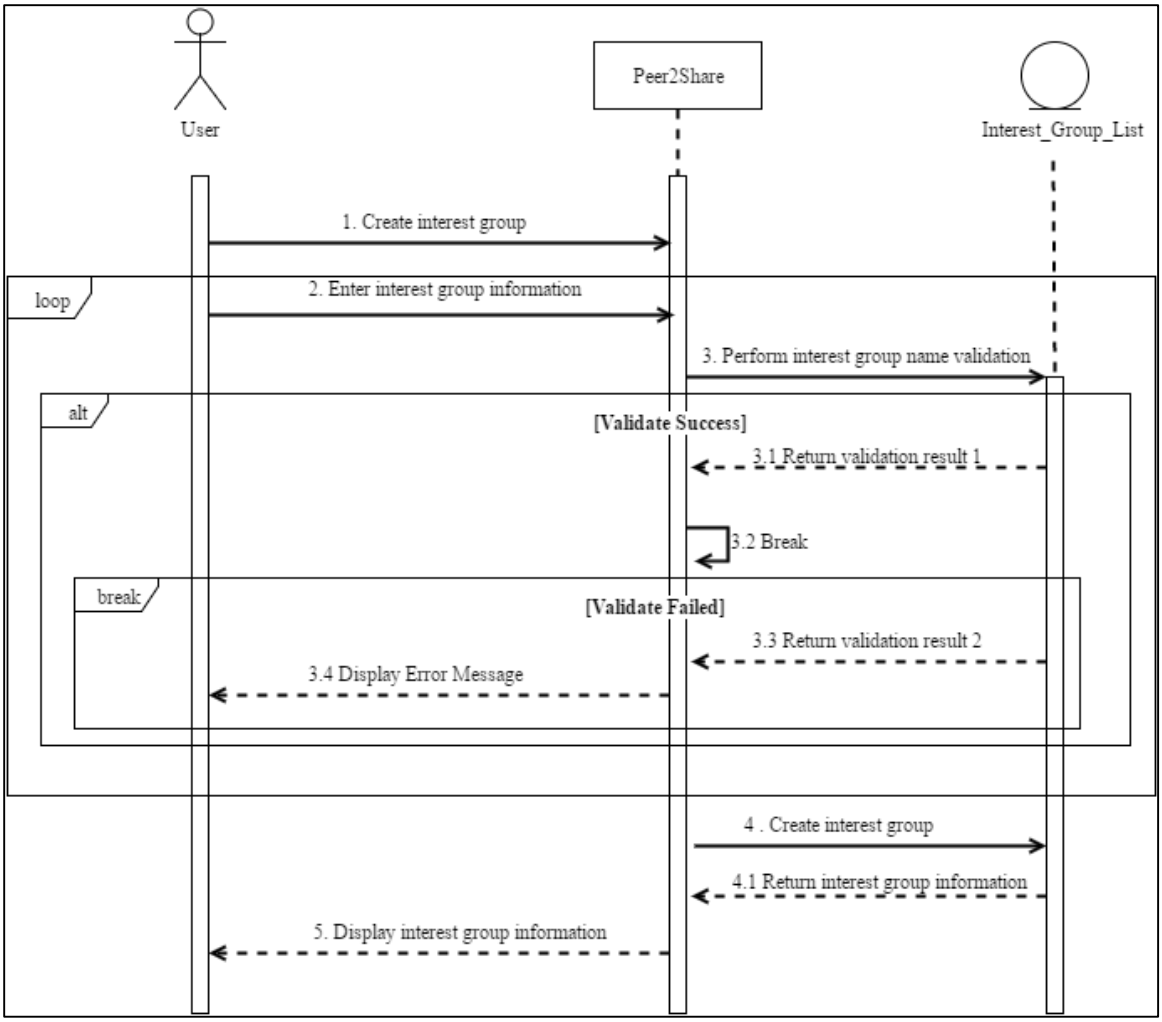


Figure 3.2.15: Sequence Diagram of Create Interest Group

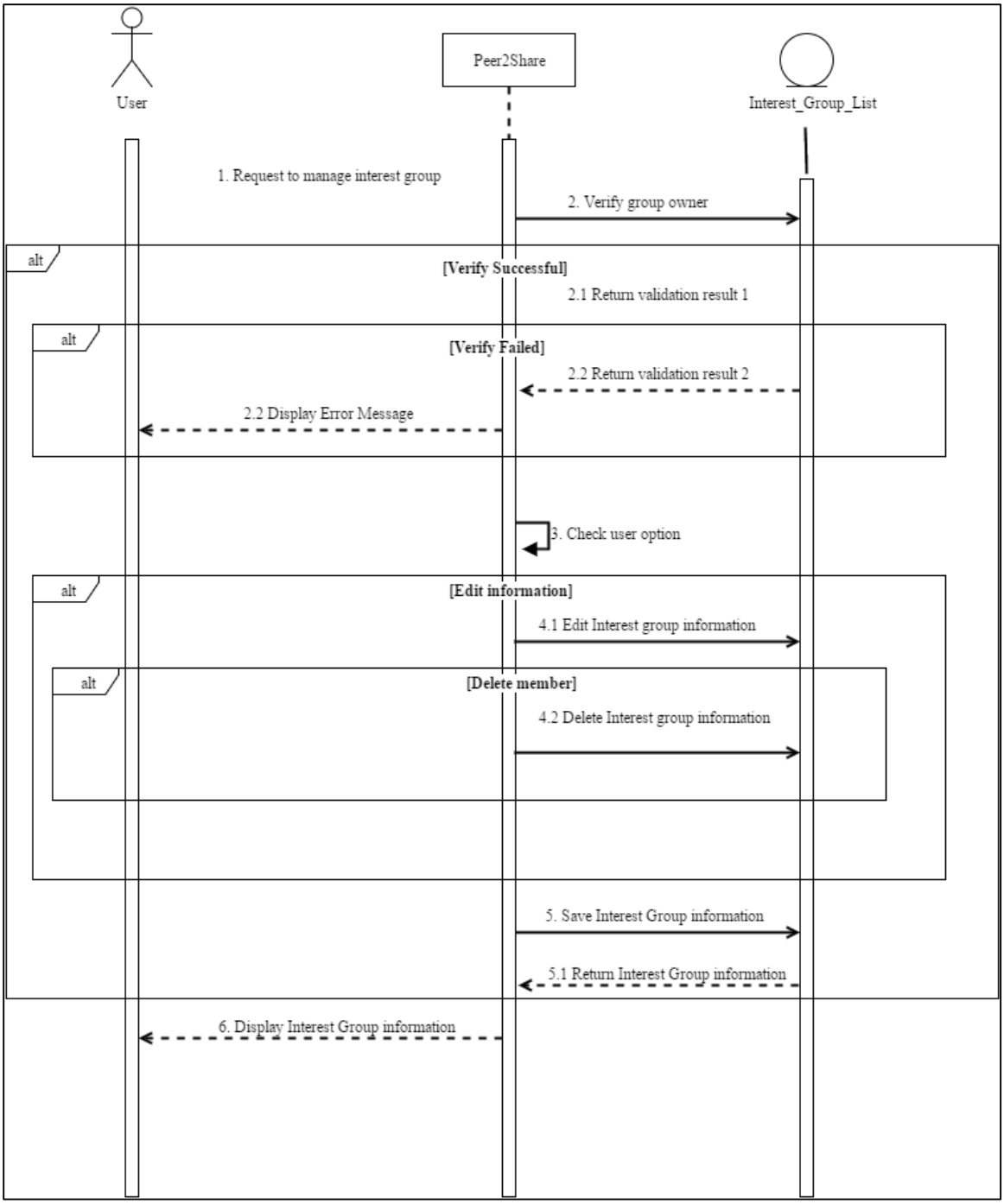


Figure 3.2.16: Sequence Diagram of Manage Interest Group

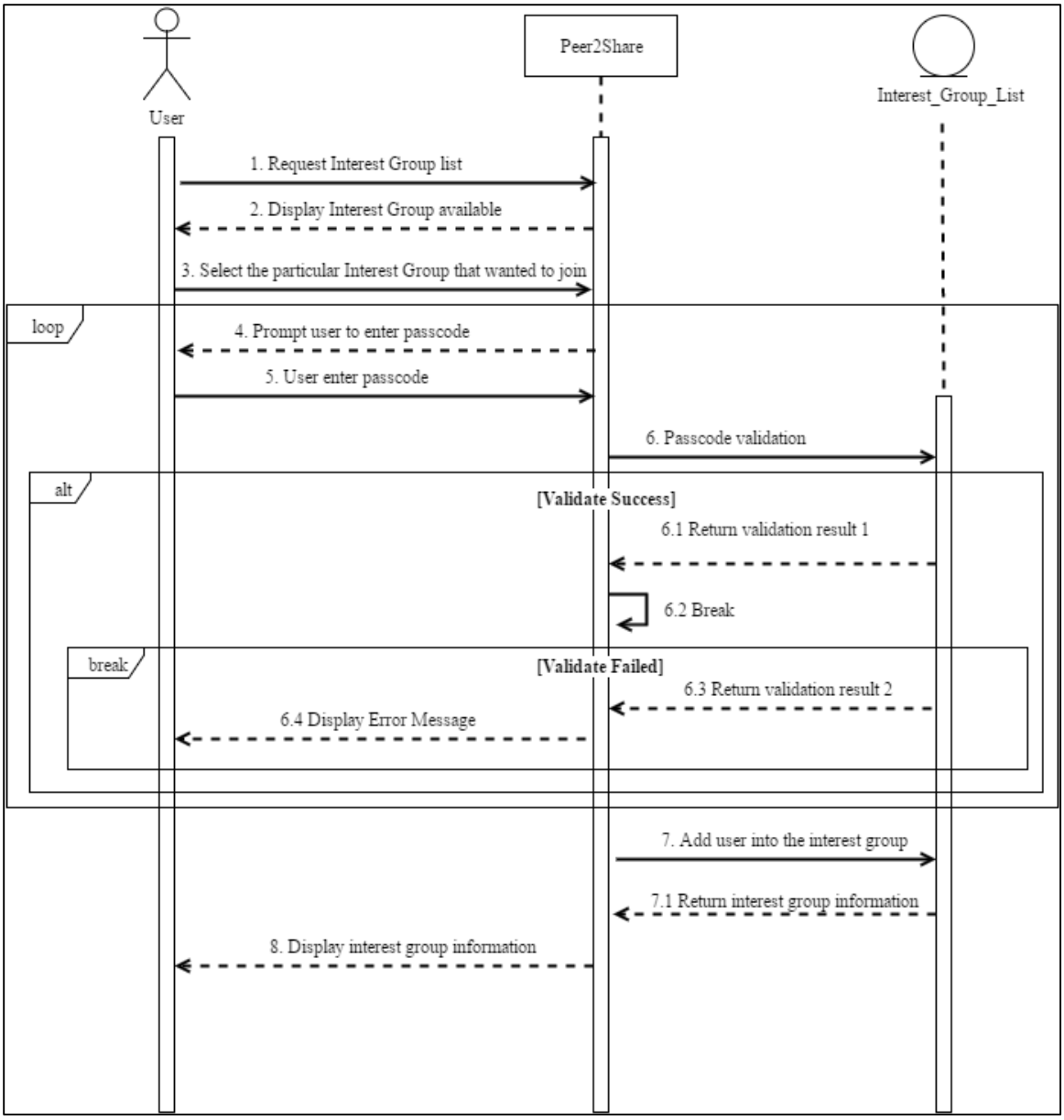


Figure 3.2.17: Sequence Diagram of Join Interest Group

3.3.6 Object Diagram

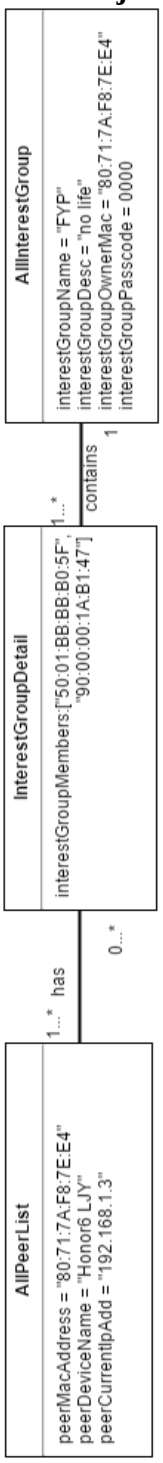


Figure 3.2.18: Object Diagram

3.3 System Architecture Design

The system basically allow user to select files that wanted to be shared to other in the network, and other users will be able to see the list of resources shared by user. They can selected the particular file they want and the file will be download into their device. Resources sharing in the system does not involve any server as any user in the network can act as client and server at the same time. In order to do resources sharing with a specific group of people, user can create an interest group, the user who created the particular interest group will be the group owner of that group. Group owner can edit and manage the interest group. During the process of creating the new interest group, group owner is requested to enter a 4 number passcode, and this passcode can be pass to other user who interested to join the group. Other user who interested to join a particular interest group can request to join the group by entering a passcode set by the group owner. Whenever the system detected the group owner shared a file which already existed in the shared resources list, it will consider it as an updated file and will notify the group members in order to download the updated version of the file.

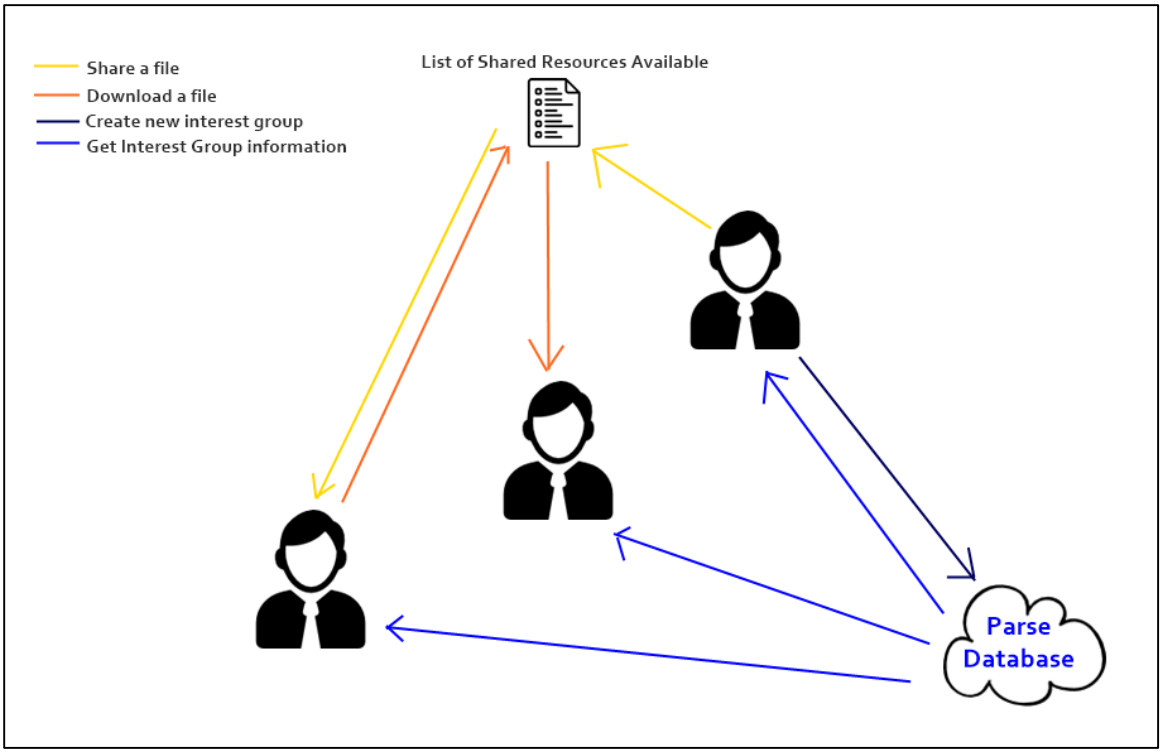


Figure 3.3: Expected final system architecture

3.4 System Flow Chart

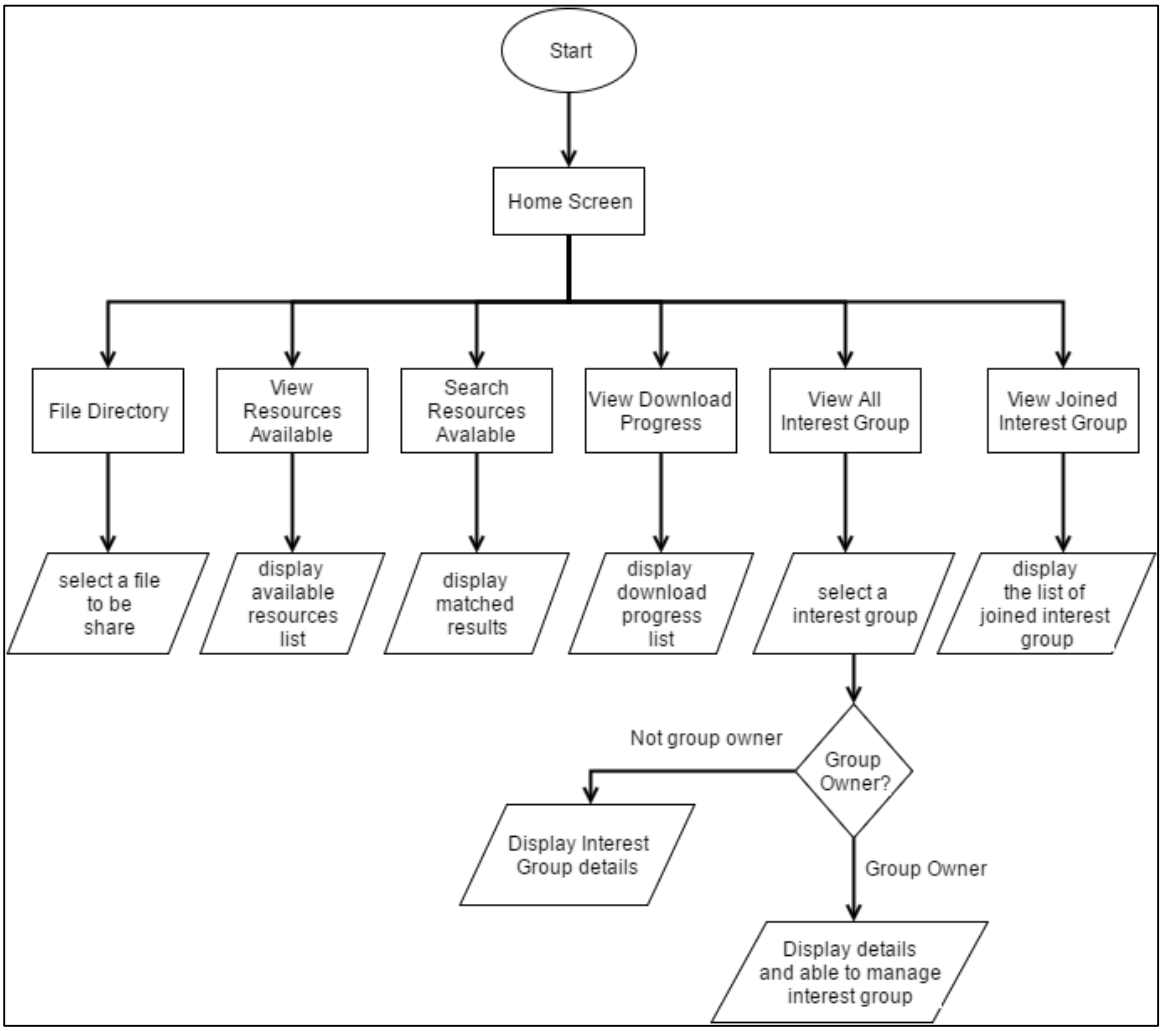


Figure 3.4: System Flow Chart

3.5 Application User Interface Design

3.5.1 Home Page

Figure 3.5.1 shows the interface design of the Home Page of PeerToShare. There are three main component in the Home Page. First component is the user device information, it consists of the user device's name that user set it in the system setting, MAC address and the IP address of the user device. There are four options available in the second component: File Directory, user will be directed to browse and select the resources that wanted to be share to other peers in the network by selecting this option. Shared Resources Available, user will be directed to list of shared resources in the network. Search Resources Available, user will be directed to search resources available in the network by typing the resources name. Download Progress, user will be directed to see the download progress of the resources selected. The third component will be some tips of using this application.

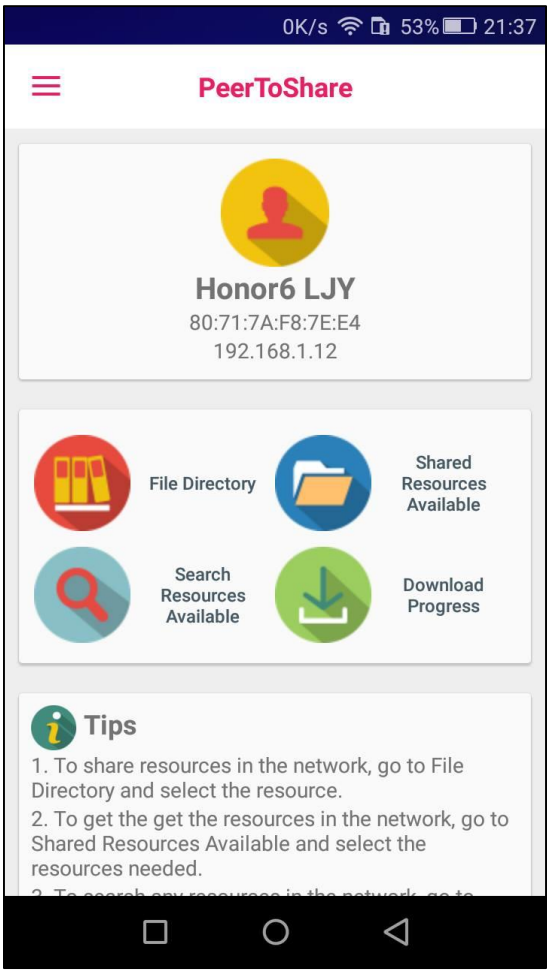


Figure 3.5.1: Interface Design of PeerToShare Home Page

3.5.2 File Directory

The File Directory screen shown in Figure 3.5.2 below. User can browse all the files available in the device's file system regardless of internal or external storage. In order to select the resources that wanted to be shared on the network, user just simply have to press on the file that wanted to be share, and the file will be automatically added to the shared resources list.

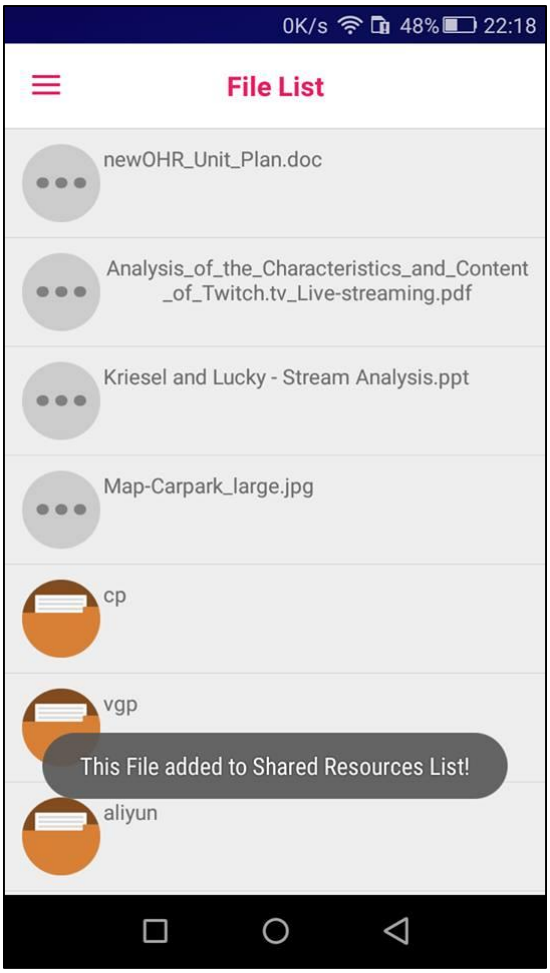


Figure 3.5.2: Interface Design of File Directory screen

3.5.3 Shared Resources Available

Figure 3.5.3 shows the Shared Resources Available in the network. User who selected the resources to be shared in the network will be shown in this list. The IP address of the user who shared the resources and the total file shared will be shown in the list. User can select the IP address that want to get the particular resource from, then the resources name shared by that user will be shown. User just simply select that particular resource from the list and the resource will be downloaded to the user device.

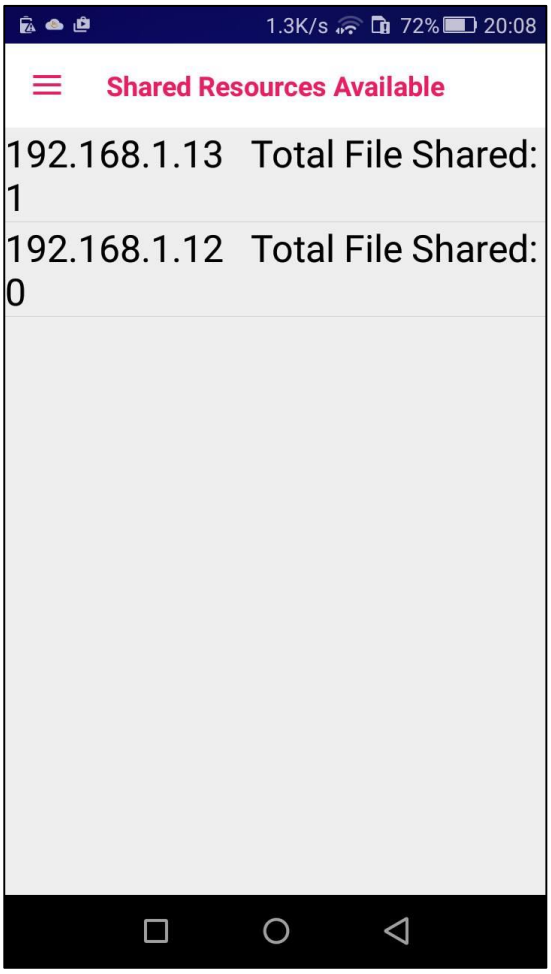


Figure 3.5.3a: Interface Design of Peer with Shared Resources Available

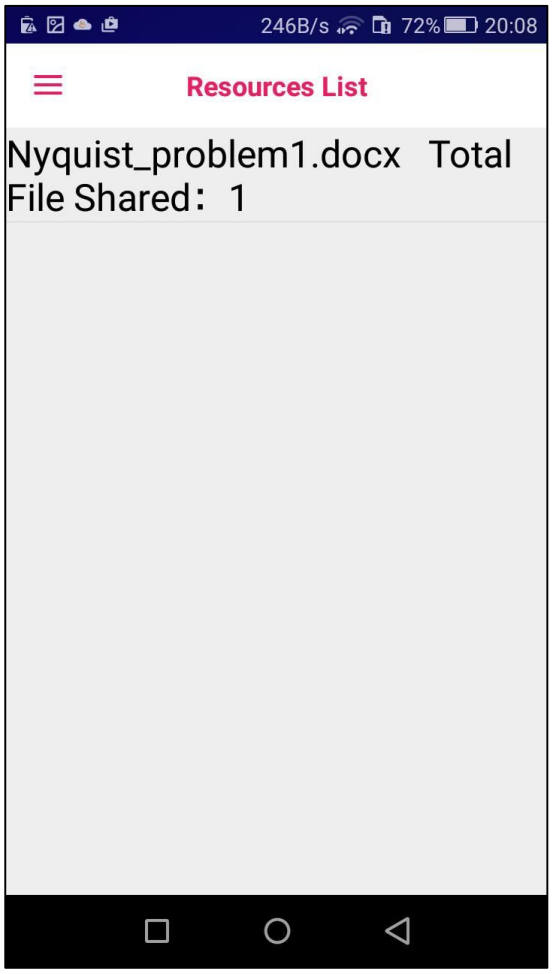


Figure 3.5.3b: Interface Design of Shared Resource

3.5.4 Search Resources Available

Search Resources Available screen was shown in Figure 3.5.4 below. User can get a particular resource not only by viewing the whole list of shared resources available in the network, if the list of shared resources available is too lengthy and this will take some time for user to search for a particular resource. User can use this search resources available function to search for a particular resource by tying the name of the resources. If the name entered by user matched any of the resources from the list, it will be shown to the user in order allow user to select it.

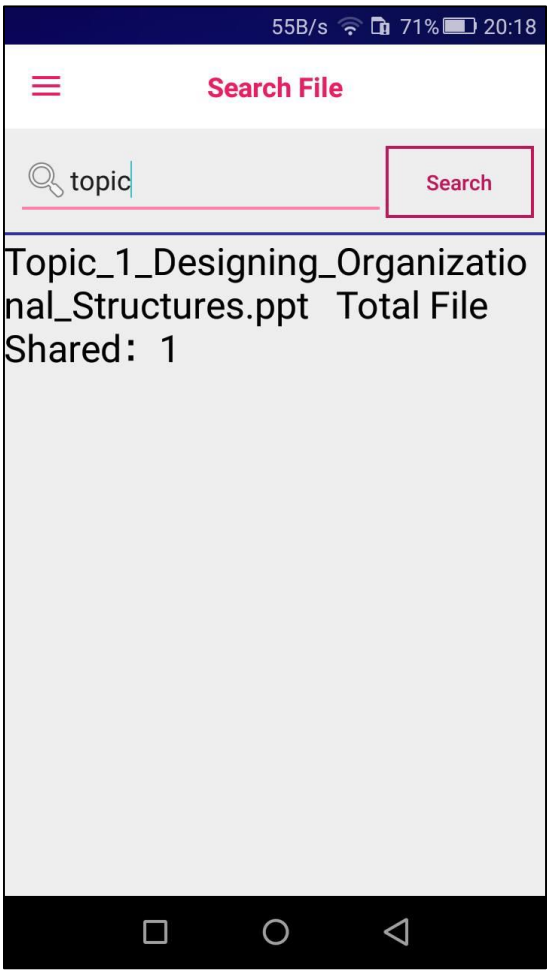


Figure 3.5.4: Interface Design of Search Resources Available

3.5.5 Download Progress

The Download Progress interface design shown in Figure 3.5.5. After user selected the resource they want to download, they will be directed to this screen. It will show the download progress of the resource. If it is still in download progress the status will be shown as Downloading and if it is download completed the status will be Download Completed.

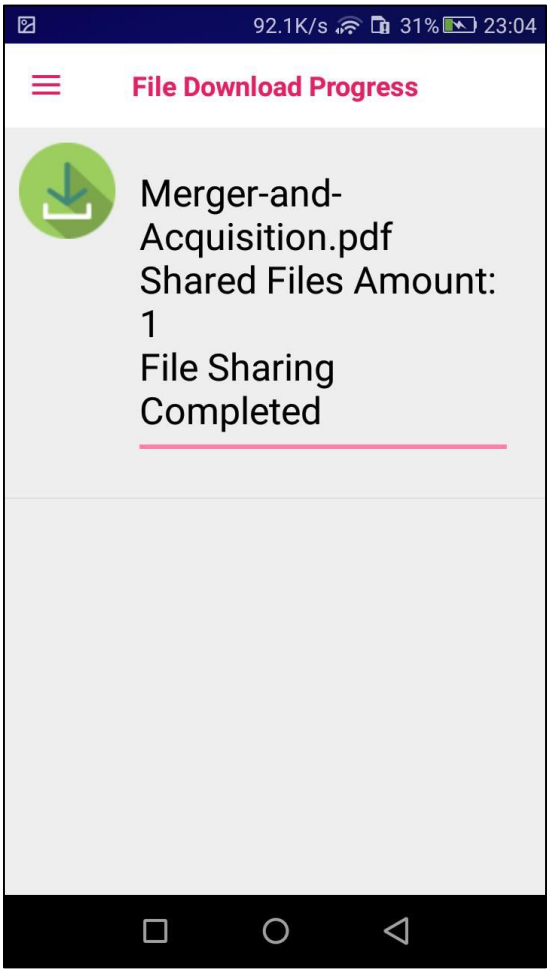


Figure 3.5.5: Interface Design of Download Progress

3.5.6 Interest Group

Figure 3.5.6a and Figure 3.5.6b show the Interest Group interface design. The list of interest group created by all peers available in the network will be shown as in the Figure 3.5.6a. In order to add a new interest group, user have to press on the pink button on the bottom right corner, and the dialog of inserting the interest group details will be shown as in the Figure 3.5.6b. User have to key in the new interest group name, description and a 4 numbers passcode in order to create a new group. If user leave even one of these fields blank, error message will be shown and user will not be able to proceed.

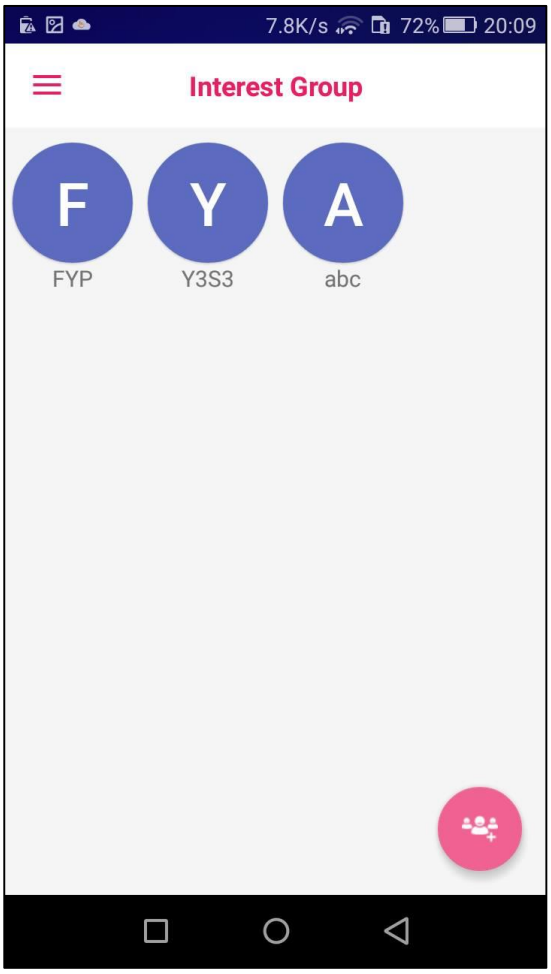


Figure 3.5.6a: Interface Design of Interest Group

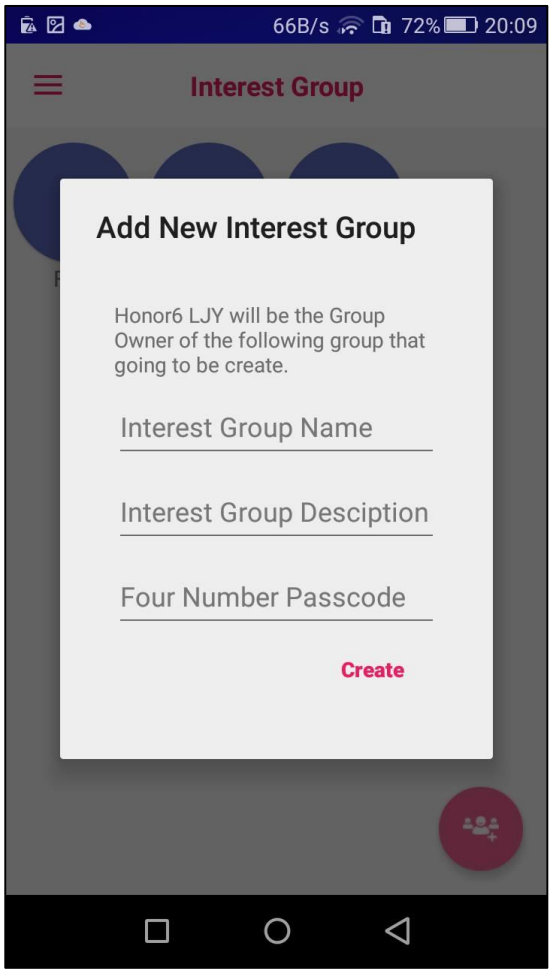


Figure 3.5.6b: Interface Design of create new interest group

3.5.7 Interest Group Details

After the user selected a particular group on the Interest Group screen, it will be directed to this interface shown in the Figure 3.6.7a below. The Interest Group name and description show on the top of the screen. Followed by the group members in the particular group, the group members list can be scrolled horizontally. If the user intended to join a group, he or she can press on the Request Join button and a dialog will prompt out to ask for passcode which shows in Figure 3.6.7b.

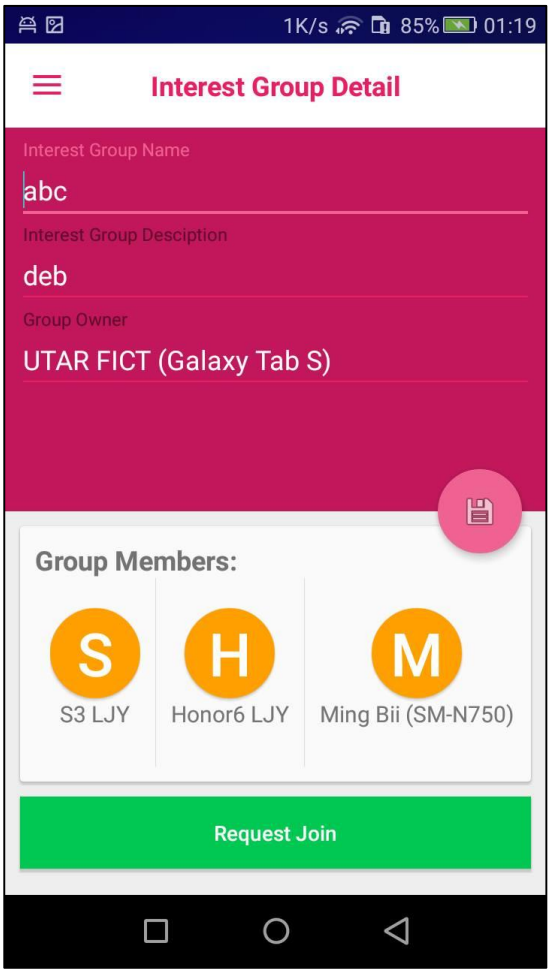


Figure 3.5.7a: Interface Design of Interest Group detail

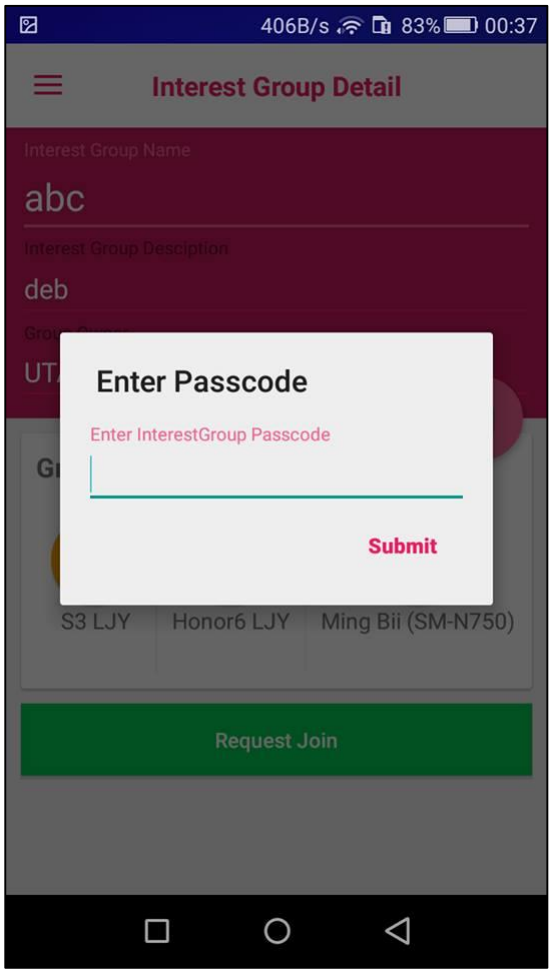


Figure 3.5.7b: Dialog of requesting user to enter passcode

Figure 3.5.7c shown the interface design of Interest Group Detail when the user is the group owner of the group. The Request Join button will not appear when the user is a group owner or the user is a member in the particular group. Besides, there is an extra field shown in the top component of the screen, which is the group passcode set by the group owner. Figure 3.5.7d shows the confirmation dialog when group owner trying to long tap on a particular member in the group to delete the member. When user press on the Share Files button, they will be directed to the File Directory screen and when user press on the View Resources Available button will be directed to the Shared Resources Available screen.

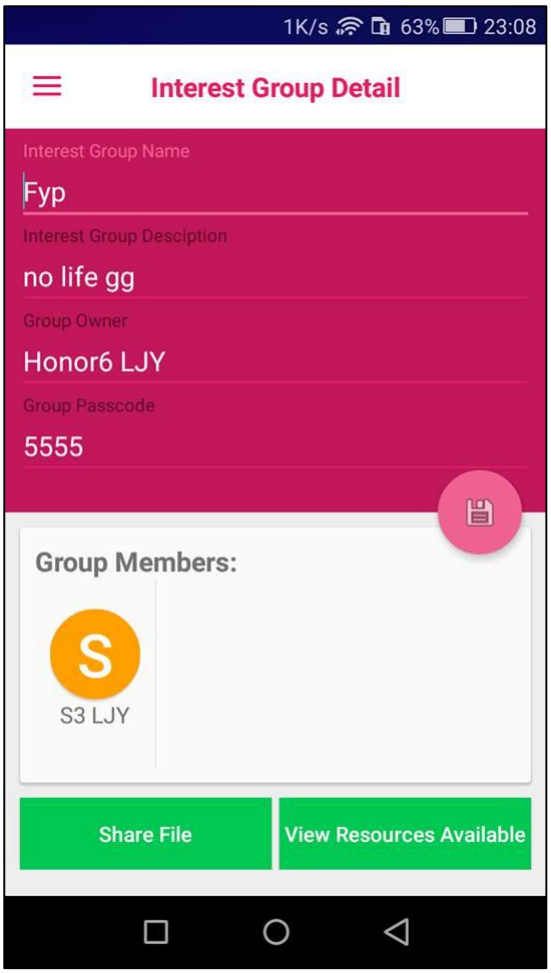


Figure 3.5.7c: Interface Design of Interest Group Detail for group owner

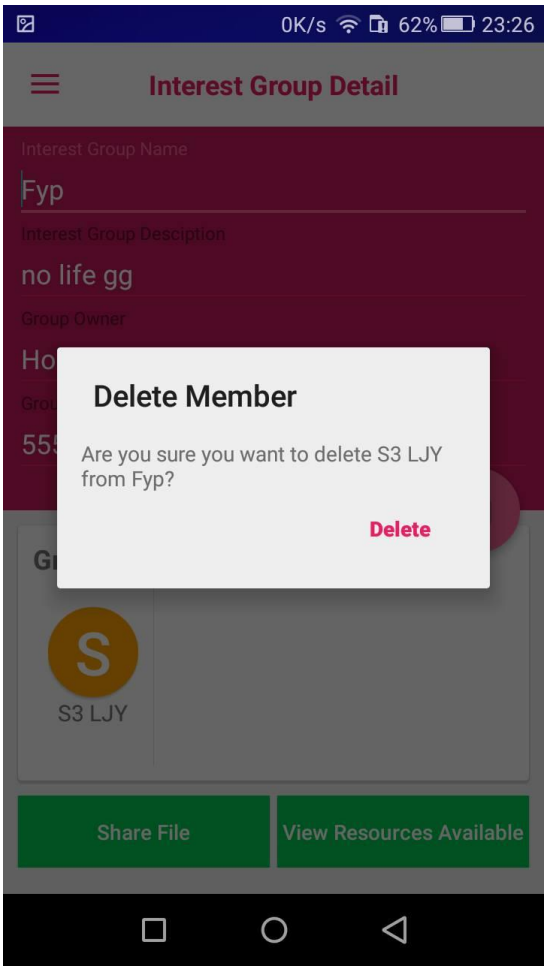


Figure 3.5.7d: Confirmation Dialog of deleting a member

When a group member in the group selected a file to share in a group, a notification will be sent to all the group members in the group to notify them there is a new file added in the group. Figure 3.5.7e shows the notification screen.

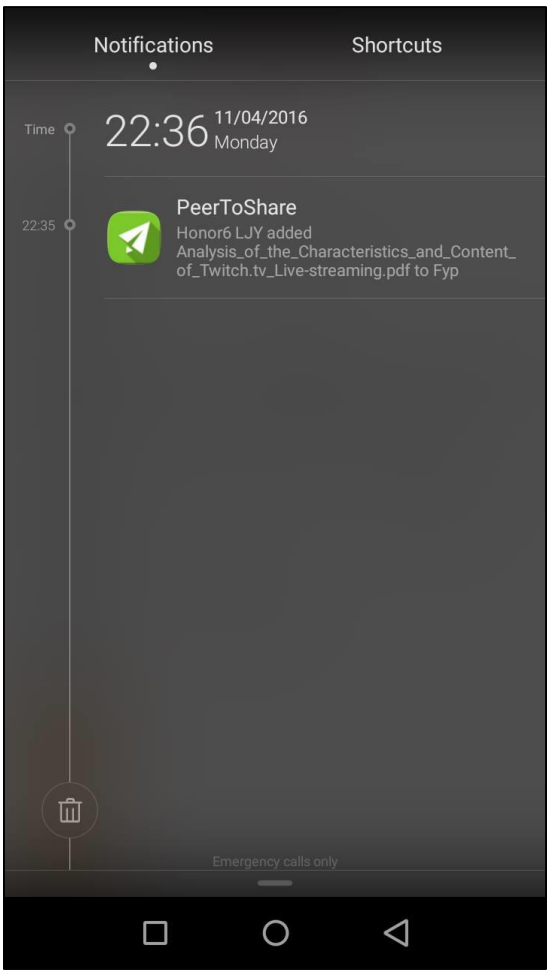


Figure 3.5.7e: Notification received when group member shared a file in the group

3.5.8 Navigation Drawer

By pressing the hamburger icon on the top left corner or swiping from left to right in the application, the navigation drawer will be shown. User can directly navigate to another screen by using this navigation drawer.

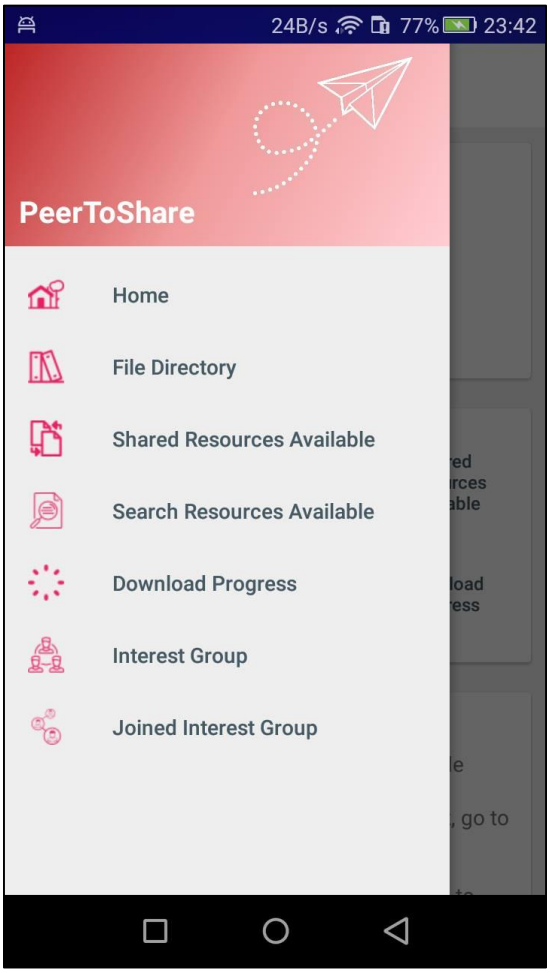


Figure 3.5.8: Interface Design of Navigation Drawer

3.6 Timeline

The timeline of the project is scheduled by using Gantt Chart as shown as below. For Table 3.6.1 the Gantt Chart for previous semester and Table 3.6.2 is the Gantt Chart scheduled for the this semester. This project is estimated to be complete in two semesters which is around 14 weeks. There is two project report submission. After Project I report submission, a prototype presentation will be conducted, whereas after Project II report submission, the real system demonstration will be conducted.

Figure 3.6.1: Gantt Chart for previous semester

Task Name	Duration	Start Date	End Date	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Final year project 1	**94 Days**	**8/6/2015**	**11/9/2015**							
1. Planning	**17 Days**									
1.1 Identifying Problem	2 Days	8/6/2015	10/6/2015							
1.2 Define The Project Objective	3 Days	15/6/2015	17/6/2015							
1.3 Define The Project Motivation	3 Days	27/6/2015	29/6/2015							
1.4 Research on previous system	5 Days	1/7/2015	5/7/2015							
1.5 Determining The Project Scope	3 Days	18/6/2015	21/6/2015							
1.6 Schedule in grant chart	1 Days	25/6/2015	25/6/2015							
2 Analysis	**11 Days**									
2.1 Review existing system	6 Days	6/7/2015	11/6/2015							
2.2 Proposed solution	5 Days	9/7/2015	13/7/2015							
3 Design	**20 Days**									
3.1 Studies on system methodology	3 Days	14/7/2015	16/7/2015							
3.2 Chosen methodology	2 Days	17/7/2015	18/7/2015							
3.3 System Design	15 Days	17/7/2015	1/8/2015							
Final report for FYP 1	**58 Days**	**8/6/2015**	**7/8/2015**							

Task Name	Duration	Start Date	End Date	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14
4 Implementation	**36 Days**									
4.1 System prototyping	21 Days	2/8/2015	28/9/2015							
4.2 System testing	5 Days	28/9/2015	1/9/2015							
Presentation	**1 Days**	**7/9/2015**	**7/9/2015**							
Submit Final report FYP 1	**1 Days**	**14/9/2015**	**14/9/2015**							

Figure 3.6.2: Gantt Chart for curent semester

Task Name	Duration	Start Date	End Date	Week 15	16	17	18	19	20	21	22	23	24	25	26	27	28
Final year project 2	**91 Days**	**18/1/2016**	**29/5/2016**														
1. Project analysis	5 Days	18/1/2016	22/1/2016														
2. Project implementation	75 Days	23/1/2016	7/4/2016														
2.1 System Development	75 Days	23/1/2016	7/4/2016														
2.2 System Testing	75 Days	23/1/2016	7/4/2016														
2.3 System Refinement	75 Days	23/1/2016	7/4/2016														
2.4 Project evaluation	75 Days	23/1/2016	7/4/2016														
Final report for FYP2	20 Days	8/4/2016	27/4/2016														
Presentation	1 Days	22/5/2016	22/5/2016														
Submit Final report FYP 2	**1 Days**	**29/5/2016**	**29/5/2016**														

CHAPTER 4 METHODOLOGY AND TOOLS

4.1 Design Specifications

Evolutionary development (prototyping) approach is choose to develop this project. Figure 4.1 (Elfira Nureza Ardina, 2010) shows the model of the evolutionary development.

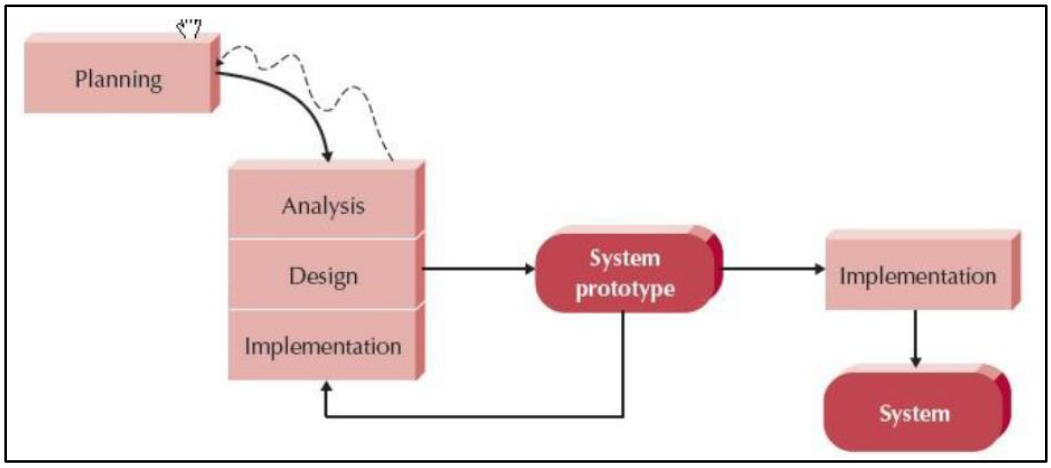


Figure 4.1: The model of the evolutionary development (Elfira Nureza Ardina, 2010)

Evolutionary development approach is selected because this project can be develop by into prototype which is a system which is not fully functioning based on latest requirement. User can try to use the prototype and provide feedbacks in order to improve the system so the final outcome can fulfil all the user requirements. Moreover, by using this method to develop the system can reduce the risk of system failure because errors can be detected in the process of prototype evaluation.

In the initial planning phase, the problems faced by UTAR lecturers and students was determined. The scope of the project also been set after some researches being done. The project scope is to develop a mobile application implementing Structured P2P over MANET so that lecturers and students can instantly share learning materials in the class by using their smart devices without going through a central server. They will still be able to share learning materials in the class although WBLE is down for maintenance sometimes. Interest Group sharing feature included in this project to increase the efficiency of resources sharing between lecturer and students, as lecturer can share the learning materials to a group of students at a same time instead of sharing it one by one. Furthermore, file

synchronization is also part of the project scope, as lecturers will be updating the learning materials they use in class time to time, it is hard for student to keep track for their file version is it the latest version.

In the analysis phase, some existing P2P file sharing applications has been reviewed and compared about their strengths and weaknesses. The comparison table has been constructed to clearly identify that what types of method the existing system was using to develop the project and what improvement or features should be added in order to make the system more complete. An improved solution is proposed to overcome the weaknesses in the existing application. Structured P2P system will be used to implement in this project as it is complexity is lesser compare to Unstructured P2P system. After reviewing the existing similar application in the market, most of the applications did not include the Interest Group sharing feature and none of them implemented the file synchronize feature.

Development process started at the design phase, a user-friendly GUI of the resources sharing P2P over MANET will be develop on the mobile devices. Some simple and basic function of the proposed solution will be develop as a first prototype. The function of sharing resources without passing through a server between mobile devices is added to the mobile application. The file synchronization features will be added into the mobile application at the second stage.

For the implementation phase, it is separated into 2 phase which is the Prototype implementation phase and the Final System implementation phase. Prototype implementation phase will be further divided into 2 stages. At the first stage, the function of sharing resources without passing through a server between mobile devices is added to the mobile application. Interest Group sharing functions will be included in the application at the second stage of implementation. Then, the file synchronization features will be added into the mobile application at the third stage. A prototype will be develop and deliver to users in order to know their feedback at the end of each stages. The feedback is collected and the prototype will keep reversing back to the design and implementation phase in order

to improve and fulfil user expectations. This will continue repeat until the end of the design and implementation phase.

Lastly will be the final system implementation stage, the last version of prototype which already go through multiple testing and accepted by user will be develop into the final system. In the final system, smart devices should be able to connect to each other in the wireless environment, and do resources sharing by implementing the P2P over MANET approach by using the mobile application. User can also share resources to different Interest Group so that a group of users can receive the resources at a same time. The entire development process should be well documented in order to maintain the system easily and have further improvement in future.

4.2 Tools to use

Software:

- Android Studio with Studio Development Kit (SDK)
    - IDE used to develop and debug the Android Application and link it to the database which used to store the Interest Group information
- Parse
    - Cloud database used to store the Interest Group Information

Hardware:

- Notebook
    - Specification: Asus K551LN with the specifications of Intel® Core™ i5-4200U CPU @ 1.60GHz 2.30GHz processor, 8.00GB RAM, 64-bit Operating System and x64-based processor
    - Used to develop the mobile application
- Android-based smart devices
    - Huawei Honor 6, Android 5.1.1 Lollipop
    - Samsung Galaxy Tab S, Android 5.0.1 Lollipop
    - Samsung Galaxy S3, Android 4.1.1 Kikkat

4.3 System Requirements

The basic requirements need in order to make sure the application function well are listed as below:

1. The Android version of the device must be at least Android Kikkat 4.2.2 (API 17) and above.
2. The device must be connected to a Wi-Fi network in order to share resources, regardless of Wi-Fi direct from another device or a wireless internet router.
3. The device must connected to the internet in order to get Interest Group information.

This minimum Android version used to develop this project is Android Kikkat 4.2.2 (API 17) and maximum Android version is Android Marshmallow 6.0 (API 23), for devices which having Android version lower than the minimum version will face problem during installation as the version is not compatible. To share resources among devices, the device is not necessary connected to the internet. By connecting to another device using Wi-Fi Direct and they can start to do resources sharing among each other. Alternatively, devices that connected to the same internet access point can share resources among each other as well. In order for user to get interest group details, user have to connect to the internet and retrieve the interest group information from the Parse cloud database. By connecting to the Parse database, user can manage, edit the interest group information and the data will be updated in real time.

CHAPTER 5 IMPLEMENTATION AND TESTING

5.1 Implementation

The process of implementation and development of this project will be describe in the following part. The basic installation and configurations steps are listed down in the following sections. By documenting the development process, the future implementation, update and maintaining this application will be much easier.

5.1.1 Android Studio Setup and Installation

Android Studio is the main software used in this developing this project, it is the IDE used to code, develop and debug the Android application. The Android Studio installation file can be downloaded from http://developer.android.com/sdk/index.html

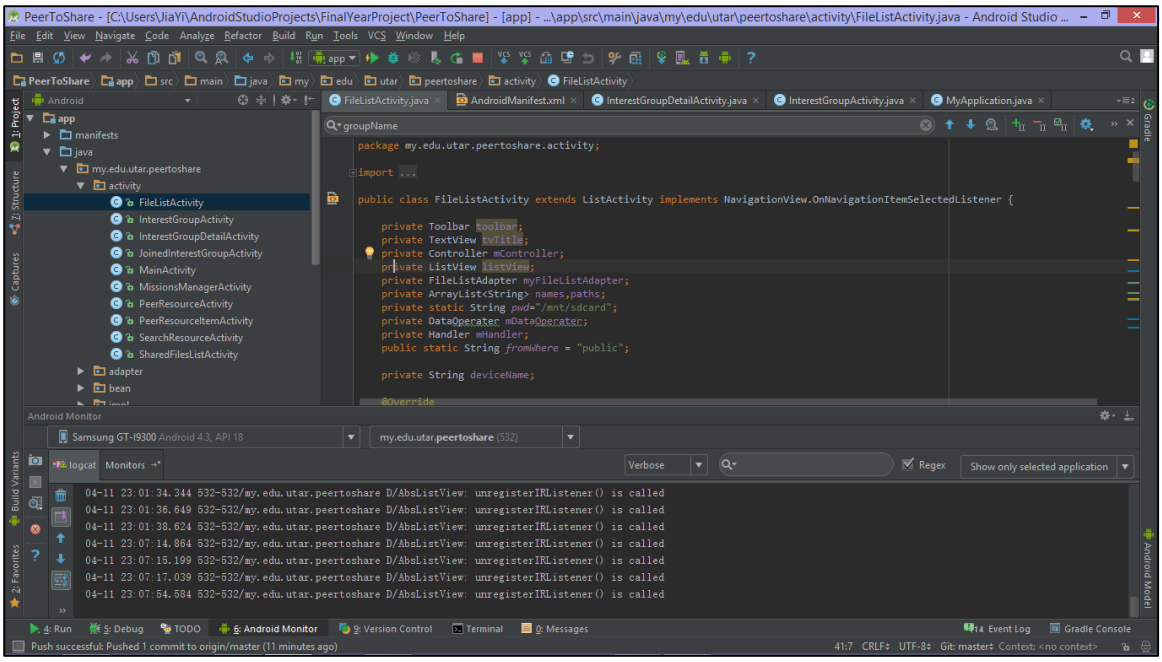


Figure 5.1.1a: Android Studio 2.0

Besides installing Android Studio, Android SDK Manager need to be downloaded as well. These are all the SDK platforms that are going to be used to develop the application.

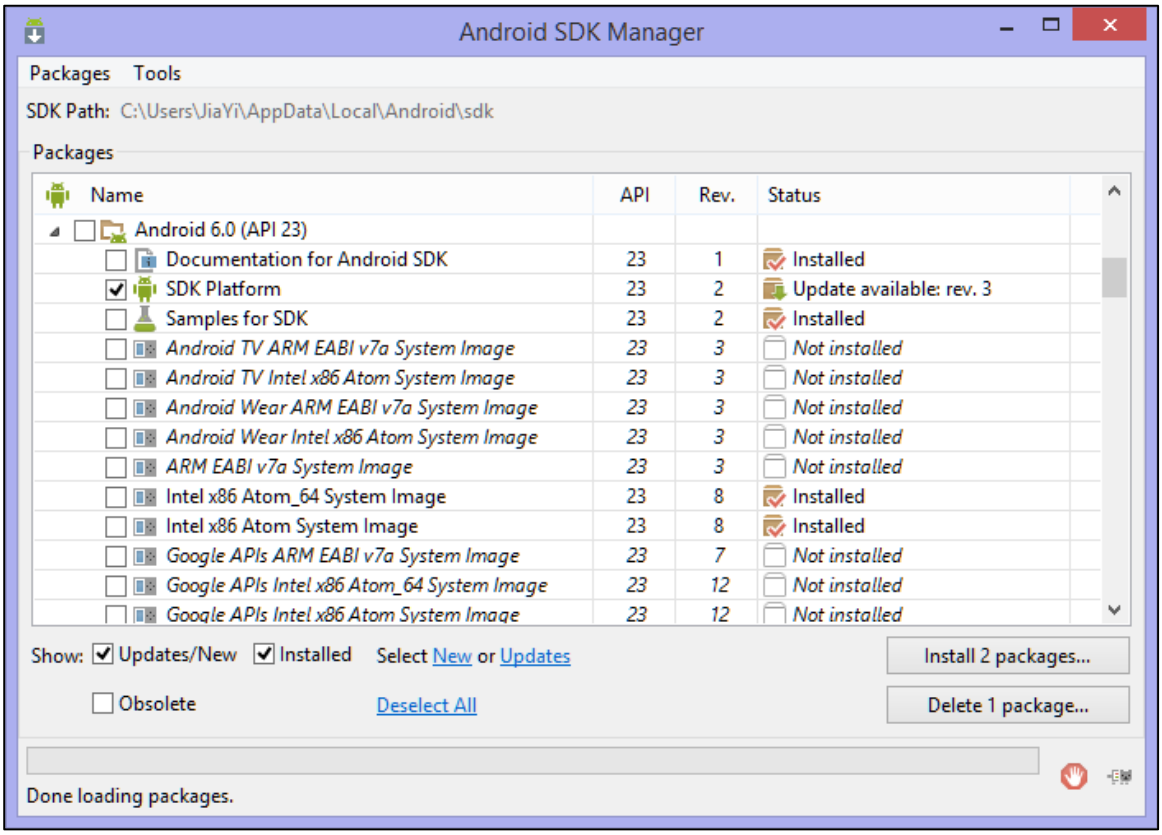


Figure 5.1.1b: Android SDK Manager

5.1.2 Parse Cloud Database

Firstly, create a user account in Parse and start to create an application in Parse. P2Share is the application created in Parse that link to this project.

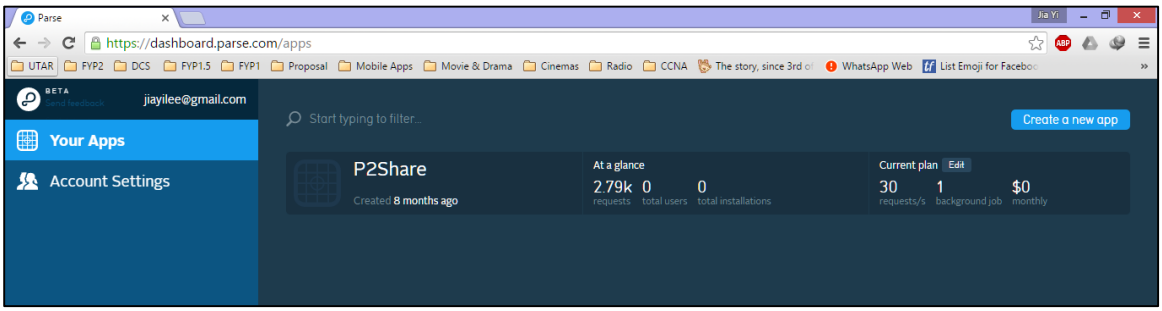


Figure 5.1.2a: Parse application dashboard

The tables created in Parse can be view in a table form. It is quite convenient for user to view the data in the tables.

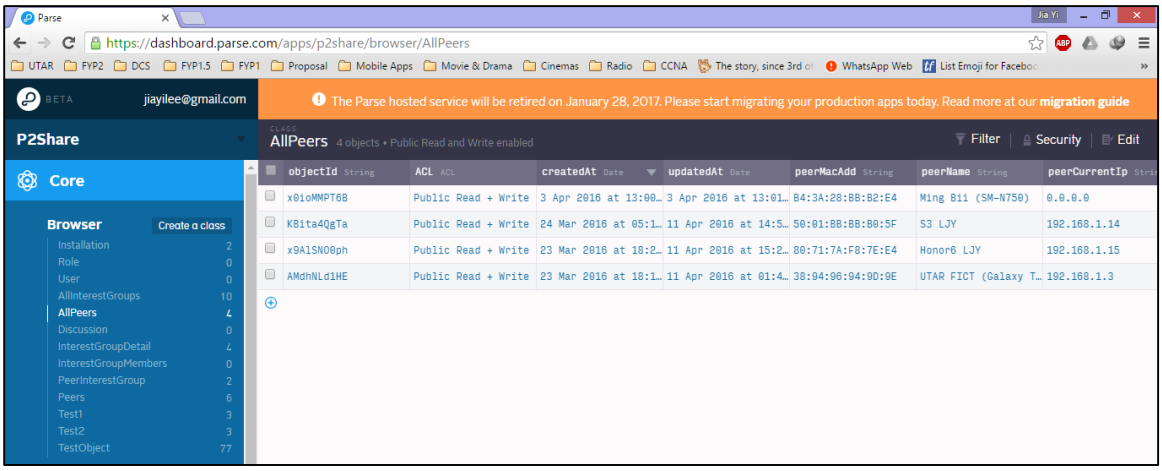


Figure 5.1.2b: Database tables in Parse

The following code needed in order to connect the application to the Parse cloud database. The *app_key* and *client_key* need to be replaced based on the user account created in Parse, it is auto assigned by Parse.

```
Parse.enableLocalDatastore(this);
Parse.initialize(this, "app_key", "client_key");
ParseInstallation.getCurrentInstallation().saveInBackground();
```

Figure 5.1.2c: Code connecting the application to Parse

Moreover, the Parse library need to be included in the build gradle file of the application. By including the `compile 'com.parse:parse-android:1.+'` the entire Parse library will be imported into the project.

```
android {
    compileSdkVersion 23
    buildToolsVersion "23.0.2"

    defaultConfig {
        applicationId "my.edu.utar.peertoshare"
        minSdkVersion 17
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.1.1'
    compile 'com.android.support:design:23.1.1'
    compile 'com.android.support:cardview-v7:23.1.1'
    compile 'com.parse:parse-android:1.+'
}
```

Figure 5.1.2d: Build gradle of the application

Figure 5.1.2e shows the how the data from Parse cloud database can be query and store in a local variable in the project. In this example, the data from the InterestGroupDetail table is queried and store inside a local List of Object variable name listOfInterestGroup.

```
publicInterestGroups = new ArrayList<>();

try {
    ParseQuery<ParseObject> query = new ParseQuery<>("InterestGroupDetail");
    query.orderByAscending("groupName");
    interestGroup = query.find();

    for (ParseObject interestGroupList : interestGroup) {
        InterestGroupBean interestGroupBean = new InterestGroupBean();
        interestGroupBean.setGroupName((String) interestGroupList.get("groupName"));
        interestGroupBean.setGroupDesc((String) interestGroupList.get("groupDesc"));
        interestGroupBean.setGroupPasscode((String) interestGroupList.get("groupPasscode"));
        interestGroupBean.setGroupOwnerId((String) interestGroupList.get("groupOwner"));
        publicInterestGroups.add(interestGroupBean);
    }

    for(int a=0; a<publicInterestGroups.size(); a++){
        InterestGroupBean temp = new InterestGroupBean();
        temp.setGroupName(publicInterestGroups.get(a).getGroupName());
        temp.setGroupOwnerId(publicInterestGroups.get(a).getGroupOwnerId());
        listOfInterestGroup.add(temp);
    }

} catch (com.parse.ParseException e) {
    Log.e("Error", e.getMessage());
    e.printStackTrace();
}
```

Figure 5.1.2e: Code of query data from Parse cloud database

For the push notification for user when group member in the interest group shared a file, Parse push notification is used. Some uses permissions is required to set in the AndroidManifest.xml file in the project in order to allow the push notifications.

```
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />


<permission android:protectionLevel="signature"
    android:name="my.edu.utar.peertoshare.permission.C2D_MESSAGE" />
<uses-permission android:name="my.edu.utar.peertoshare.permission.C2D_MESSAGE" />
```

Figure 5.1.2f: User permission for push notification in AndroidManifest.xml

```
<service android:name="com.parse.PushService" />
<receiver android:name="com.parse.ParseBroadcastReceiver">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
        <action android:name="android.intent.action.USER_PRESENT" />
    </intent-filter>
</receiver>
<receiver android:name=".parse.MyCustomParsePushReceiver"
    android:exported="false">
    <intent-filter>
        <action android:name="com.parse.push.intent.RECEIVE" />
        <action android:name="com.parse.push.intent.DELETE" />
        <action android:name="com.parse.push.intent.OPEN" />
    </intent-filter>
</receiver>
<receiver android:name="com.parse.GcmBroadcastReceiver"
    android:permission="com.google.android.c2dm.permission.SEND">
    <intent-filter>
        <action android:name="com.google.android.c2dm.intent.RECEIVE" />
        <action android:name="com.google.android.c2dm.intent.REGISTRATION" />
        <category android:name="my.edu.utar.peertoshare" />
    </intent-filter>
</receiver>
```

Figure 5.1.2g: Services for push notification in Android Manifest.xml

After setting up the user permission, Figure 5.1.2g shows the code of building the notification. Notification icon, title and the notification content should be set here.

```
JSONObject json = new JSONObject(intent.getExtras().getString("com.parse.Data"));
Log.d(TAG, json.getString("alert").toString());

final String notificationTitle = json.getString("title").toString();
notificationContent = json.getString("alert").toString();
final String uri = json.getString("uri");

Intent resultIntent = null;
TaskStackBuilder stackBuilder = TaskStackBuilder.create(context);


resultIntent = new Intent(context, MainActivity.class);
stackBuilder.addParentStack(MainActivity.class);

stackBuilder.addNextIntent(resultIntent);
PendingIntent resultPendingIntent =
        stackBuilder.getPendingIntent(0, PendingIntent.FLAG_UPDATE_CURRENT);

NotificationCompat.Builder builder =
        new NotificationCompat.Builder(context)
                .setSmallIcon(R.mipmap.ic_launcher)
                .setContentTitle(notificationTitle)
                .setContentText(notificationContent);
//              .setContentIntent(resultPendingIntent); //intent to open the activity while onclick

int mNotificationId = 001;
NotificationManager mNotifyMgr =
        (NotificationManager) context.getSystemService(Context.NOTIFICATION_SERVICE);
mNotifyMgr.notify(mNotificationId, builder.build());
```

Figure 5.1.2h: Code of building notification

5.1.3 Resources sharing without using a server

Some uses permission need to be set in the AndroidManifest.xml before resources sharing between devices can happen. The uses permission of read and write on the user device storage, the uses permission to access to internet and wifi state has to be set.

```
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Figure 5.1.3a: Uses permission in AndroidMenifest.xml

The segment of the code below shows how the file being shared from one device to another device without using a server. The port number 33333 is defined to allow devices share resources using this port and the Datagram Socket is used to share the file from one device to another.

```
private final int port = 33333;
private DatagramSocket datagramSocket;
private int count;
```

```
byte[] buf = receivedatagramPacket.getData();
byte[] send=null;
int len=buf.length;
String ip = receivedatagramPacket.getAddress().toString();
ip = ip.substring( ip.lastIndexOf("/") + 1 );
int port = receivedatagramPacket.getPort();
String msgString=new String(buf);
Log.v("test", "resolvecommand:"+msgString);

String[] msgSeg=msgString.split(":");

String dateString=msgSeg[1];
```

```
sendDatagramPacket=new DatagramPacket(send, send.length, receivedatagramPacket.getAddress(), 33333);
datagramSocket.send(sendDatagramPacket);
datagramSocket.close();
```

Figure 5.1.3b: Code of sharing resources without using a server

5.2 Testing

Black Box testing is used to examine the functionality of this project. Black Box testing is a software testing method where the function, design, implementation of the item being test is not known to the tester. Black Box testing is used because the test can be conducted by a person who is independent from the developer, this can avoid to arise the problem of developer-bias. Besides, tester do not need to have any programming language background which related to the system. They do not need to know how the software being developed and implemented as well.

5.2.1 Resources Sharing Module

5.2.2.1 Home

Case	Test Action	Test Result	Status
1	Tap on *File Directory* button	Directed to the File Directory screen	Pass
2	Tap on *Shared Resources Available* button	Directed to the Shared Resources Available screen	Pass
3	Tap on *Search Shared Resources Available* button	Directed to the Search Shared Resources Available screen	Pass
4	Tap on *Download Progress* button	Directed to the Download Progress screen	Pass
5	Launch the application without connecting to a Wi-Fi	Popup alert shown and mentioned user did not connect to a Wi-Fi network	Pass
6	Launch the application by connecting to a Wi-Fi	No popup alert shown	Pass

5.2.1.2 File Directory

Case	Test Action	Test Result	Status
1	Tap on a *folder*	Enter the folder directory	Pass
2	Tap on a *file*	Popup alert is shown mentioning that the file is added to Shared Resources list	Pass

3	A member in an interest group tap on a *file*	Notification about someone shared a file in the group being sent to all group members in that particular group	Pass
4	The group owner tap on a *file* which already shared previously	Notification about the group owner updated the file being sent to all group members in that particular group	Pass

5.2.1.3 Shared Resources Available

Case	Test Action	Test Result	Status
1	Tap on a *particular IP address* of peers	Directed to the Resources Item list	Pass
2	Tap on a *resource* in Resources Item list screen	Directed to the Download Progress screen	Pass

5.2.1.4 Search Shared Resources Available

Case	Test Action	Test Result	Status
1	Tap on the *Search* button without entering file name	Popup alert is shown by asking user to enter file name	Pass
2	Tap on the *Search* button by entering a file name which matched the shared resources available list	The list of matched results shown	Pass
3	Tap on the *Search* button by entering a file name which does not matched the shared resources available list	"No match results" message shown	Pass
4	Tap on a *resource* in Resources Item list screen	Directed to the Download Progress screen	Pass

5.2.1.5 Download Progress

Case	Test Action	Test Result	Status
1	File successfully downloaded	"[*filename*] Download Completed" message is shown	Pass
2	File downloading	"[*filename*] Downloading" message is shown	Pass
3	Long tap on a *file* and tap on the *pause* button	The file download progress paused	Pass
4	Long tap on a file and tap on the *delete* button	The file download progress deleted	Pass
5	Long tap on a *file* and tap on the *resume* button	The file download progress resume	Pass

5.2.2 Interest Group Module

5.2.2.1 Interest Group

Case	Test Action	Test Result	Status
1	Tap on a *particular interest group* from the list	Redirected to the Interest Group Detail screen	Pass
2	Tap on the *add new interest group* button	Popup dialog shown with the user device name and create new interest group form	Pass
3	Tap on the *Create* button without entering the new interest group name, description or passcode	Warning alert shown that cannot leave any of the field blank	Pass
4	Tap on the *Create* button without entering the new interest group name, description and passcode	Warning alert shown that cannot leave any of the field blank	Pass
5	Tap on the *Create* button by entering the new interest group name, description and passcode	Popup alert shown that the new interest group successfully created	Pass
6	Tap on the *Create* button by entering the new interest group name which already existed in the list	Warning alert shown that the group name is repeated and ask user to change another group name	Pass

5.2.2.2 Joined Interest Group

Case	Test Action	Test Result	Status
1	User is not a member of any interest group	"No interest group available" message shown	Pass
2	User is a member of one or more than one interest group	The list of interest group that user joined as a member shown	Pass
3	Tap on a *particular interest group* from the list	Redirected to the Interest Group Detail screen	Pass

5.2.2.3 Interest Group Detail

Case	Test Action	Test Result	Status
1	Tap the *Save* button by without entering the interest group name and description	Warning alert shown that cannot leave any of the field blank	Pass
2	Tap the *Save* button by without entering the interest group name or description	Warning alert shown that cannot leave any of the field blank	Pass
3	Tap the *Save* button after editing the interest group information but user is not a group owner	Popup error message and mentioned that user is not a group owner is shown	Pass
4	Tap the *Save* button after editing the interest group information and user is the group owner	Popup alert shown the information is successfully updated	Pass
5	Long tap on a *particular member* from the member list	Popup alert shown that the new interest group successfully created	Pass
6	Tap on the *Delete* button to member but user is not the group owner	Popup error message and mentioned that member cannot be deleted as user is not the group owner	Pass
7	Tap on the *Delete* button to member and user is the group owner	Popup alert shown that the particular member has been deleted successfully	Pass

8	Tap on the *Request Join* button	Popup dialog shown and request user to enter the passcode	Pass
9	Tap on the *Enter* button without entering the passcode	Warning alert shown that cannot leave the field blank	Pass
10	Tap on the *Enter* button by entering the wrong passcode	Popup error message and mentioned that passcode is wrong and user unable to join the group	Pass
11	Tap on the *Enter* button by entering the correct passcode	Popup alert shown the user joined the interest group successfully	Pass
12	Tap on the *Share File* button	Directed to the File Directory screen	Pass
13	Tap on the *View Resources Available* Button	Directed to the Resources Available screen	Pass

5.2.3 Others Module

5.2.3.1 Navigation Drawer

Case	Test Action	Test Result	Status
1	Tap on *Home* button	Directed to the Home screen	Pass
2	Tap on *File Directory* button	Directed to the File Directory screen	Pass
3	Tap on *Shared Resources Available* button	Directed to the Shared Resources Available screen	Pass
4	Tap on *Search Resources Available* button	Directed to the Search Resources Available screen	Pass
5	Tap on *Download Progress* button	Directed to the Download Progress screen	Pass
6	Tap on *Interest Group* button	Directed to the Interest Group screen	Pass
7	Tap on *Join Interest Group* button	Directed to the Joined Interest Group screen	Pass

CHAPTER 6 CONCLUSION

6.1 Project Review

Before developing this project, some research related to this project has been done to further understand the concept and methodology needed to be implement in this project such as P2P approach, Mobile Ad Hoc Network, P2P over MANET, interest group sharing and etc. After the research on related concepts, the critical remarks on similar existing applications has been done as well. The strengths and weaknesses of all the existing application was listed down and this project is improved from the existing application.

To allow resources sharing without having a centralize server by using P2P approach is the core function of this project. Moreover this function should be develop in Android platform to allow user to install the application in their smart devices and share resources without any restriction on mobility. Interest group feature added into this project to allow user to classify their friends into different group based on interest, and the interest group details will be save so that user do not have to recreate the group every time when using the application.

In the interest group, when the group owner shared a file which already existed in the shared resources available list, it will be consider as a updated file and notify the members in the group about the file is updated so that group member do not need to keep track on the file version time to time.

6.2 Discussion

This project implementing structured P2P for resources sharing. Once a user send a resources to a group of users, each user in the group has a copy of the resources, so that if the initial sender went offline, the resources will still be available in the network.

In order to implement Interest Group Sharing feature in this application, a database server needed to store the Interest Group details such as Interest Group name, Interest Group members and others. The database server does not involve in any of the resources sharing process. By doing in this way, user do not have to recreate the Interest Group each time they use the application and user can create more than one Interest Group in the application.

For any existing file which has a newer version, the group owner can shared the updated version of the file and other members in the group will receive a notification and reminded them to download the latest version of the file.

6.3 Contribution

Mobile P2P application for resources sharing is to provide a P2P resources sharing environment without using a centralized server whereby the end to end connection is not stable due to the mobility of users. Resources sharing in traditional central client approach will not be available when the network is congested or the server failed to function. By applying P2P approach in resources sharing, the availability of resources will be increased.

The use of smart devices rapidly increasing in these few years due to its portability and mobility. This gives the motivation of developing this project in a mobile application version instead of desktop version, so that users no longer only do resources sharing using their computer, they can easily share resources among each other in anywhere and anytime. In order to target more users, this project will develop in the Android platform.

Existing P2P resources sharing application do not have a sharing mechanism that allow multiple users to share resources based on Interest. This project includes Interest Group feature in the application. User can create multiple Interest Groups in this application which have different group of users in each Interest Group. This is able to increase the performance of resources sharing as the files shared by a user can received by a group of users at the same time.

Moreover, this application will also include the file update notification function to allow when the existing document in a group is updated, the application will notify the user which in the same group. This function able to reduce the time of user keep tracking the file version by themselves manually.

In a nutshell, this application able to provide user a server-less resources sharing environment with more freedom in term of mobility as they are no longer restricted to share resources in front their computer. The Interest Group Sharing and file update notification function can increase the performance of resources sharing and provide a more efficient way in resources sharing.

6.4 Future Work

The biggest challenge in this project is to increase the availability of resources although the sender went offline or leave the network. For further development, resources which having bigger file size, the files can be chunk into smaller pieces and distribute evenly among the peers in the network instead of distributing the entire file among themselves. By doing in this way, the storage used by to store the chunked pieces of the file will be smaller compared to storing the entire file in their smart devices.

Besides, for the interest group module can be improve in such a way that user who wish to join a particular interest group will send a request to the group owner and the owner reply with a accept or decline message. In this way group owner can have a full control on the joining of members.

BIBLIOGRAPHY

Chung-Ming Huang, Tz-Heng Hsu, and Ming-Fa Hsu, (2007). Network-aware P2P file sharing over the wireless mobile networks. *IEEE J. Select. Areas Commun.*, 25(1), pp.204-210.

Clayton, S., Nicolas, B., Frederic, S. and Lara, D. (2014). Efficient File Synchronization: Extensions and Simulations.

Corson, M., Macker, J. and Cirincione, G. [no date]. INTERNETBASED MOBILE AD HOC NETWORKING. *IEEE Internet Computing*, pp.64-65.

Elfira Nureza Ardina, 2010. *Chapter 1 Introduction To System Analysis And Design*. Available from: <http://ekohandoyo.blog.undip.ac.id/2010/01/17/chapter-1-introduction-to-system-analysis-and-design/> [28 March 2015]

Felix, B., Sebastian, G. and Axel, K. (n.d.). Towards a Three-tiered Social Graph in Decentralized Online Social Networks.

Gottron, C., König, A. and Steinmetz, R. (2010). A Survey on Security in Mobile Peer-to-Peer Architectures—Overlay-Based vs. Underlay-Based Approaches. *Future Internet*, 2(4), pp.505-532.

Hoebeke, J., Moerman, I., Dhoedt, B. and Demeester, P. [no date]. An Overview of Mobile Ad Hoc Networks: Applications and Challenges. pp.60-66.

Hui, K., Lui, J. and Yau, D. (2006). Small-world overlay P2P networks: Construction, management and handling of dynamic flash crowds. *Computer Networks*, 50(15), pp.2727-2746.

Jabbar, W., Ismail, M. and Nordin, R. (2013). Peer-to-Peer Communication on Android-Based Mobile Devices: Middleware and Protocols. p.1.

Klemm, A., Lindemann, C. and P. Waldhorst, O. [no date]. A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad Hoc Networks.

Mohanty, S., Thotakura, V. and Ramkumar, M. (2014). An Efficient Trusted Computing Base for MANET Security. *Journal of Information Security*, 05(04), pp.192-206.

Mohd Shahdi, A., Nur Emyra, M., Rathidevi, N., Rosilah, H. and Nor Effendy, O. (2013). Comparison Between Android and iOS Operating System in terms of Security. 2013 8th International Conference on Information Technology in Asia (CITA).

Prof. Rekha, P. and Ashwini, D. (2012). Cross Layer Design for P2P Overlay Network over MANET. *International Journal of Emerging Technology and Advanced Engineering*, 2(9), pp.212-215.

Pucha, H., Das, S. and Hu, Y. (2004). Ekta: An Efficient DHT Substrate for Distributed Applications in Mobile Ad Hoc Networks. *In Proceedings of the 6th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2004)*, p.1.

Readwrite.com, (2007). [online] Available at: http://readwrite.com/2007/03/29/p2p_introduction_real_world_applications [Accessed 31 Mar. 2015].

Rosenfeld, E. (2015). *Apple: Services back after major outage*. [online] CNBC. Available at: http://www.cnbc.com/id/102495735 [Accessed 31 Mar. 2015]

Sarmady, S. (2007). A Survey on Peer-to-Peer and DHT. pp.1-5.

Singh, A., Rizvi, T. and Nainiwal, V. (2015). Peer to Peer File Sharing on Local Area Network. p.10.

Statista, (2014). *Global smartphone sales to end users 2007-2014 | Statistic*. [online] Available at: http://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/ [Accessed 1 Mar. 2015].

Thampi, S. [no date]. A Review on P2P Video Streaming. p.28.

Tsidulko, J. (2014). *The 10 Biggest Cloud Outages Of 2014*. [online] CRN. Available at: http://www.crn.com/slide-shows/cloud/300075204/the-10-biggest-cloud-outages-of-2014.htm/pgno/0/9 [Accessed 31 Mar. 2015]

Wang, J., Yin, J. and Jiang, G. (n.d.). ARDC: An Improved Remote File Synchronization Algorithm.

www.idc.com, (2015). IDC: Smartphone OS Market Share. [online] Available at: http://www.idc.com/prodserv/smartphone-os-market-share.jsp [Accessed 25 Jul. 2015].

APPENDIX A – WEEKLY REPORT

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** Week 2 & 3
Student Name & ID: Lee Jia Yi 1203946	
Supervisor: Mr Tan Teik Boon	
Project Title: Mobile Peer-to-Peer Application for Resources Sharing	

1. WORK DONE

FYP I report, FYP I prototype has been refined. Further discussed with supervisor on the system requirements.

2. WORK TO BE DONE

Development on the system.

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

Need to speed up the progress of the system development.

_____ _____

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** Week 5 & 6
Student Name & ID: Lee Jia Yi 1203946	
Supervisor: Mr Tan Teik Boon	
Project Title: Mobile Peer-to-Peer Application for Resources Sharing	

1. WORK DONE

Working on file sharing module of the system. Link the application to the Parse cloud database.

2. WORK TO BE DONE

Complete the file sharing module and using Parse cloud database to perform the store the interest group information.

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

Need to speed up the progress of the system development.

_____ _____

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** Week 7 & 8
Student Name & ID: Lee Jia Yi 1203946	
Supervisor: Mr Tan Teik Boon	
Project Title: Mobile Peer-to-Peer Application for Resources Sharing	

1. WORK DONE

Completed the file sharing module and testing on the function. Working on the interest group module. A navigation drawer is created for the application. Reorganized the format of the report based on the FYP II guidelines. Modified Uses Case Diagram, Activity Diagram and Uses Case Description.

2. WORK TO BE DONE

Work on the interest group module. Refine FYP II report, add in sequence diagram, class diagram and object diagram.

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

Need to speed up the progress of the system development.

_____ _____

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT
(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** Week 9 & 10
Student Name & ID: Lee Jia Yi 1203946	
Supervisor: Mr Tan Teik Boon	
Project Title: Mobile Peer-to-Peer Application for Resources Sharing	

1. WORK DONE

Code refactor for file sharing module. Interest group module 60% completed, can query interest group information from Parse, can add, update and delete interest group. Added sequence diagram, class diagram and object diagram into report.

2. WORK TO BE DONE

Work on the interest group module. Refine FYP II report.

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

Need to speed up the progress of the system development and FYP II report.

_____ _____

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** Week 11 & 12
Student Name & ID: Lee Jia Yi 1203946	
Supervisor: Mr Tan Teik Boon	
Project Title: Mobile Peer-to-Peer Application for Resources Sharing	

1. WORK DONE

Code refactor for file sharing module. Working on integrating file sharing function into interest group. FYP II report draft is done.

2. WORK TO BE DONE

Work on the interest group module. Refine FYP II report.

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

Need to speed up the progress of the system development.

_____ _____

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Sem 3	**Study week no.:** Week 13
Student Name & ID: Lee Jia Yi 1203946	
Supervisor: Mr Tan Teik Boon	
Project Title: Mobile Peer-to-Peer Application for Resources Sharing	

1. WORK DONE

Code refactor for file sharing module. File sharing function integrated into the interest group module. Testing on interest group module. Complete FYP II report.

2. WORK TO BE DONE

Work on the notification function in interest group module.

3. PROBLEMS ENCOUNTERED

-

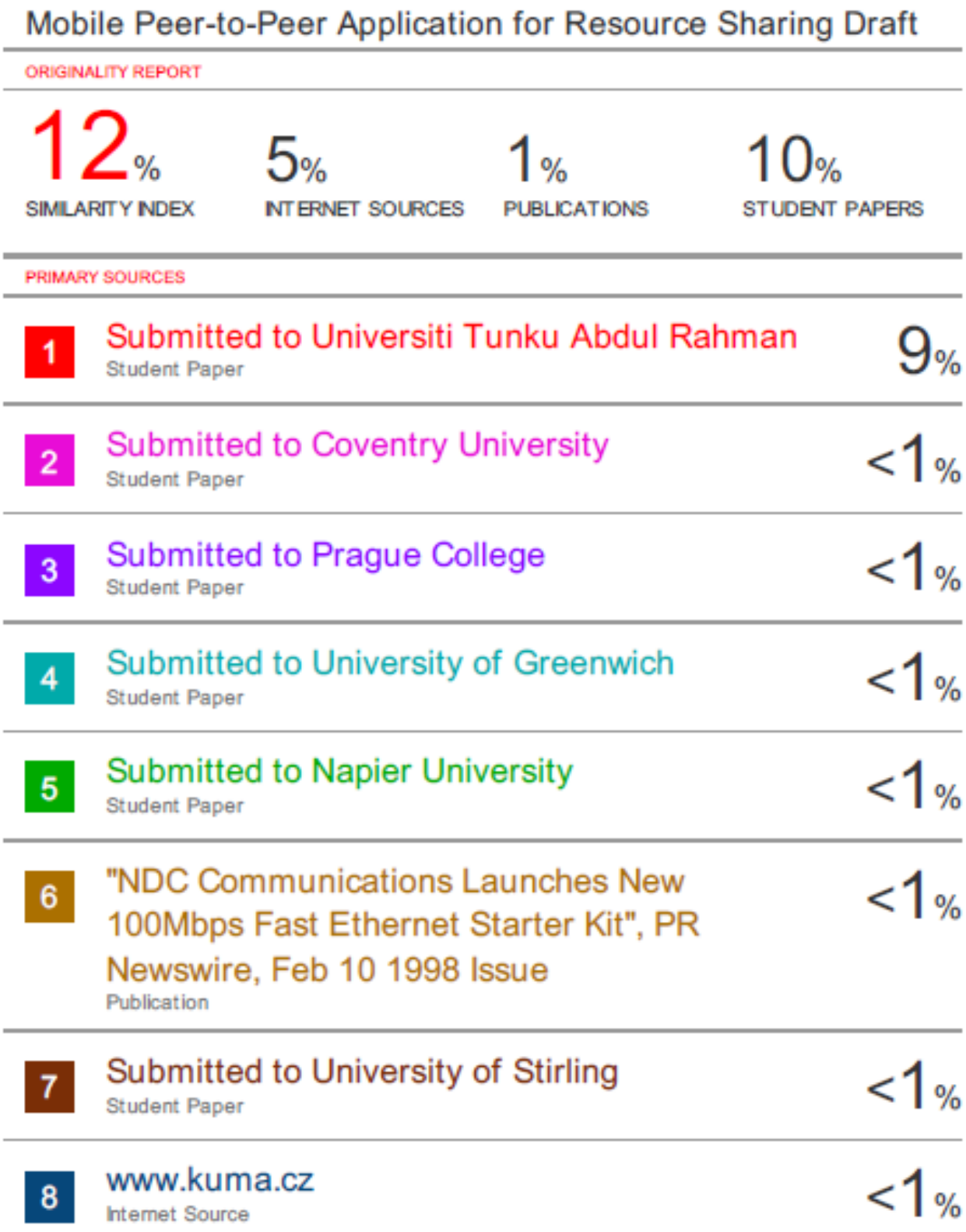
4. SELF EVALUATION OF THE PROGRESS

Need to speed up the progress of the system development.

_____ _____

Supervisor's signature Student's signature

APPENDIX B – TURNITIN SIMILARITY REPORT



Mobile Peer-to-Peer Application for Resource Sharing Draft

ORIGINALITY REPORT

12%	5%	1%	10%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Tunku Abdul Rahman Student Paper	9%
2	Submitted to Coventry University Student Paper	<1%
3	Submitted to Prague College Student Paper	<1%
4	Submitted to University of Greenwich Student Paper	<1%
5	Submitted to Napier University Student Paper	<1%
6	"NDC Communications Launches New 100Mbps Fast Ethernet Starter Kit", PR Newswire, Feb 10 1998 Issue Publication	<1%
7	Submitted to University of Stirling Student Paper	<1%
8	www.kuma.cz Internet Source	<1%

9	Submitted to The Hong Kong Polytechnic University Student Paper	<1%
10	www.csie.ncku.edu.tw Internet Source	<1%
11	Karnakewicz, Mark, Ola Sjoelund, Ayman Kanan, T. G. Beck Mason, Rao Gudimetla, Kai Hui, Jim Wang, J. M. Geary, and Julie Ji. "", Active and Passive Optical Components for WDM Communication, 2001. Publication	<1%
12	www.zdnet.co.uk Internet Source	<1%
13	Carbou, . "Service Management", Digital Home Networking Carbou/Digital Home Networking, 2013. Publication	<1%
14	www.znlww.com Internet Source	<1%
15	www.naldotech.com Internet Source	<1%
16	黒田 義之. "Formation of Inorganic Hierarchically Structured Materials by Utilizing Assemblies of Various Colloidal Particles", [出版者不明], 2012. Publication	<1%

17	Wang, Yi-da, Zhi-yong Li, and Ji-yong Li. "Monitoring system of the resource sharing network", 2012 International Conference on Systems and Informatics (ICSAI2012), 2012. Publication	<1%
18	www.daejanggan.org Internet Source	<1%

EXCLUDE QUOTES	ON	EXCLUDE MATCHES	OFF
EXCLUDE BIBLIOGRAPHY	ON		