T.C.
MARMARA UNIVERSITY
FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

CSE497 – Analysis Design Document

Push-to-Talk Application for Drivers via Wi-Fi Direct

Group Members

150113841 – Sezin GÜMÜŞ

150111061 – Yasin ORHAN

Supervised by

Assoc. Prof Müjdat SOYTÜRK

February 20th, 2018

CONTENTS

1. Introduction

1.1. Problem Description and Motivation

Most of the people check the traffic conditions on their routine routes before getting on the road. If they are lucky, then there will not a traffic jam on their route but what if traffic condition goes bad, then these people will need to learn the reason and how much time will take this problem, to decide to change or not to change their routes. However, people usually cannot have these kinds of information instantly.

In this project, our goal is to provide that real-time information to needy person from someone who is already in traffic jam inside a specific range without any cost by using Wi-Fi Direct technology. So, people can talk each other without taking any risk by only speaking rather than looking at the screen, which is the action that you have to do in most of the navigation apps to get information about the traffic while driving. Also, our project can help drivers who are in trouble on the road to get assist from someone nearby.

So, our biggest motivation is the demand for the instant information about the traffic, which is needed by drivers. And there are some sub benefits of our project, which supports our motivation, such as being free for communication, usability for not only being informed about traffic but also calling help when needed.

1.2. Scope of the Project

Our project will have several phases. At first, we will connect two peers via Wi-Fi Direct and we will manage the connection between them. By this step, we will have a knowhow about usage and management of Wi-Fi Direct technology.

When we complete the first step successfully, we will improve our application for multiple peer environment. We will connect more than 2 peers at a time and we will manage the connection according to the distance between them and ranges of each peer.

In the third step, we will implement VoIP features such as error correction, noise reduction on signals etc.

Finally, we will work for adding some extra features to our project, such as end to end encryption, extended range via relay nodes etc.

In these steps, the main technology that we will use is Wi-Fi Direct and even if it is a good technology, its short range is a disadvantage and one of our constraints. In addition, making a Voice over IP (VoIP) call without internet connection is a constraint, too because to make a traditional VoIP call, an intermediate server is needed, and we assume that our application will not use internet connection.

1.3. Definitions, acronyms, and abbreviations

VoIP: Voice over Internet Protocol

SIP: Session Initiation Protocol

P2P: Peer-to-Peer

UI: User Interface

2. Literature Survey

Android: Android is a mobile operating system which is developed by Google. The first release was in September 23rd, 2008 and it was developed for mainly mobile devices but now many kinds of smart devices operates with Android. [1]

Android Studio: It is the official Integrated Development Environment (IDE) which is used to develop applications for Android operating system. It uses IntelliJ IDEA environment. [2]

Wi-Fi Direct: Wi-Fi Direct, in other words Wi-Fi Peer-to-Peer, is a Wi-Fi standard which allows devices to easily connect and communicate with each other with no need an access point or a wireless router. [3]

Voice over IP: Voice over Internet Protocol (VoIP) provides voice communications and multimedia sessions between devices over Internet Protocol (IP) networks, which means that it uses packet switching rather than circuit switching. [4]

Session Initiation Protocol: The Session Initiation Protocol (SIP) is a widely used communications protocol for signalling and controlling multimedia communication. It is used to start, finish and control multimedia communication sessions of applications for video or voice calls in a private network. [5]

SQLite: SQLite is a widely used relational database management system which has been written in C/C++ programming languages. Unlike many other database management systems, SQLite is not a client–server database management system. It is an embedded database which means that there is no server process. [6]

3. Project Requirements

3.1. Functional Requirements

- Each user who wants to broadcast should select at least one interest and other users should be able to see these interests
- Users should be able to see other users that is in the range
- Users should be able to send request to connect any broadcaster user
- Users should be able to add any unwanted user to blocked users list
- Users should be able to add any user to their favourite list
- Users should be able get notification when a favourite user is in their range
- Time for push-to-talk should be limited for each user not to block other users
- Users should be able to communicate with minimum delay time

3.2. Non-functional Requirements

- Users should have a device which operates with Android 4.0 (Ice Cream Sandwich) or a higher version of Android operating system
- The device should support Wi-Fi Direct technology
- Wi-Fi Direct connection of the device should be enabled

4. System Design

4.1. UML Use Case Diagram(s) for the main use cases

In this part, we are going to explain how Push to Talk over Wi-Fi Direct application runs briefly with Use Case diagram.

As we explain the implementation; firstly, clients have to enable Wi-Fi Direct feature from settings of their smart devices for being able to use our application then they can register with their name and interests. After clients register, they can prefer either to check or update their profile in home page of the application or to see available users who are in a specific range of Wi-Fi Direct to chat. They can also reach available clients from home page. If they want to chat with someone in available clients, they have to choose from clients list where shows online (available users) to send request for communication. If user's who receives the request response is okay, client who send request can send his/her voice message. The use case diagram (see Figure 4.1) on the below shows that this application is implemented. (Actor: A client or user)

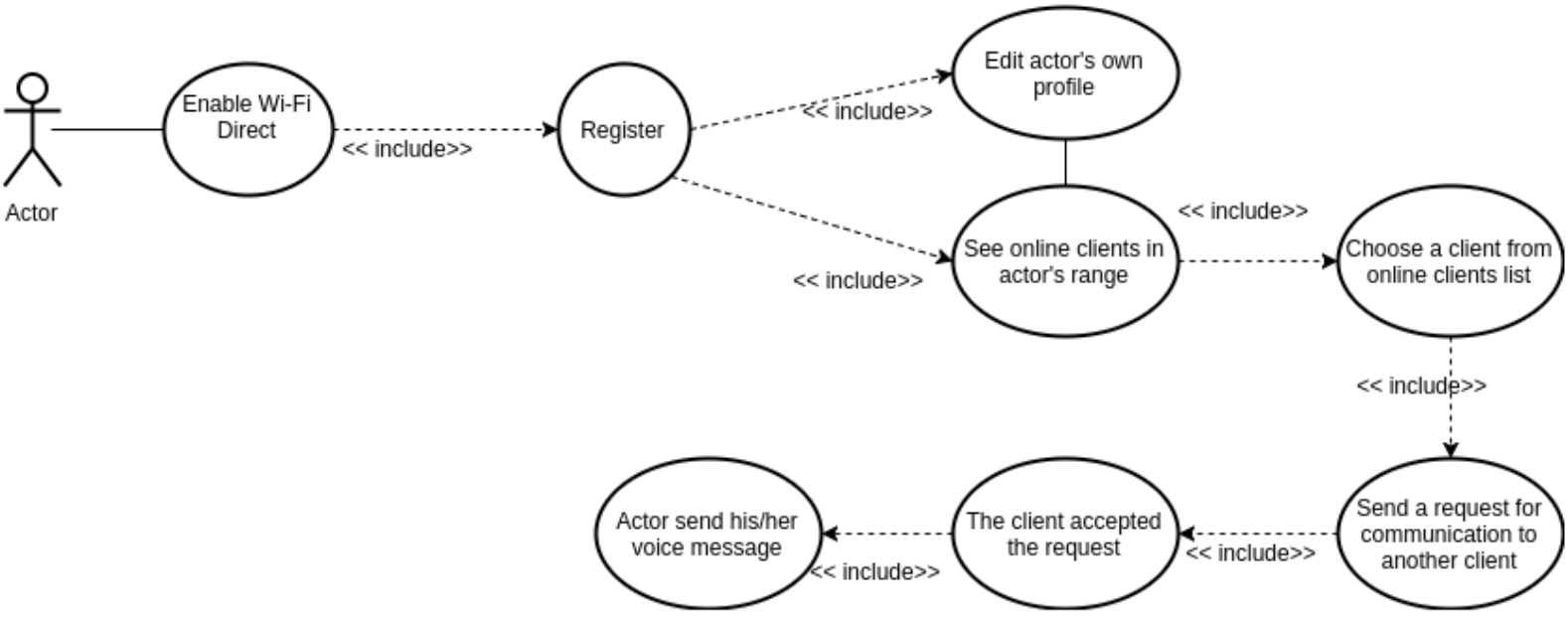

Figure 4.1: Use Case Diagram of the Application

4.2. UML Class and/or Database ER diagram(s)

Push to Talk over Wi-Fi Direct application has three classes which are Wi-Fi Direct Class which handles Wi-Fi Direct connections, FileTransfer that processes voice message like sending or receiving, Client which stores data (name, interest). The following UML class diagram (see Figure 4.2) shows classes of this application.

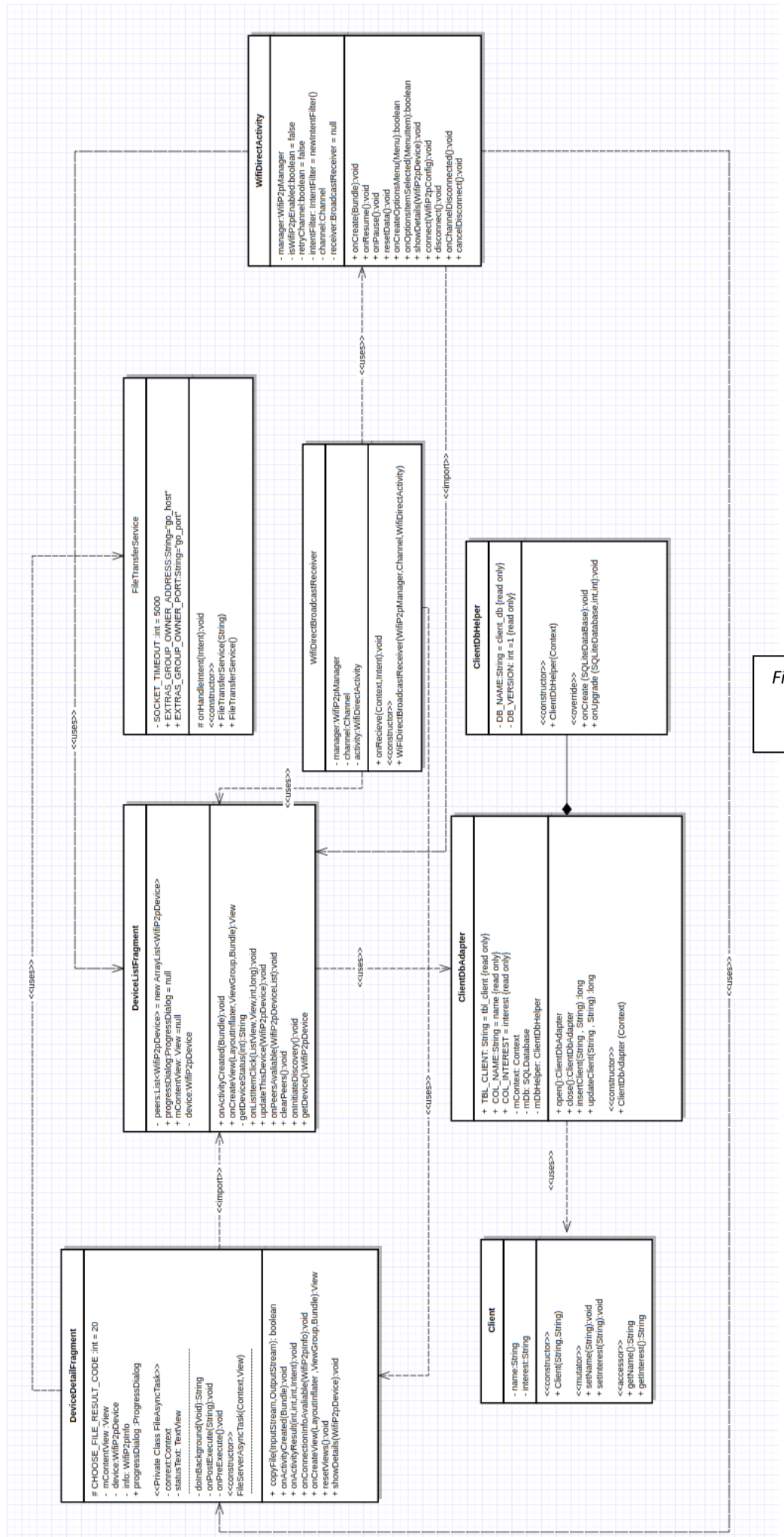

Figure 4.2: UML Class Diagram of the Application

4.3. User Interface

In this section, we are going to talk about user interface of Push to Talk over Wi-Fi Direct. We build and design our project in Android Studio. On below you can see preliminary version of application.

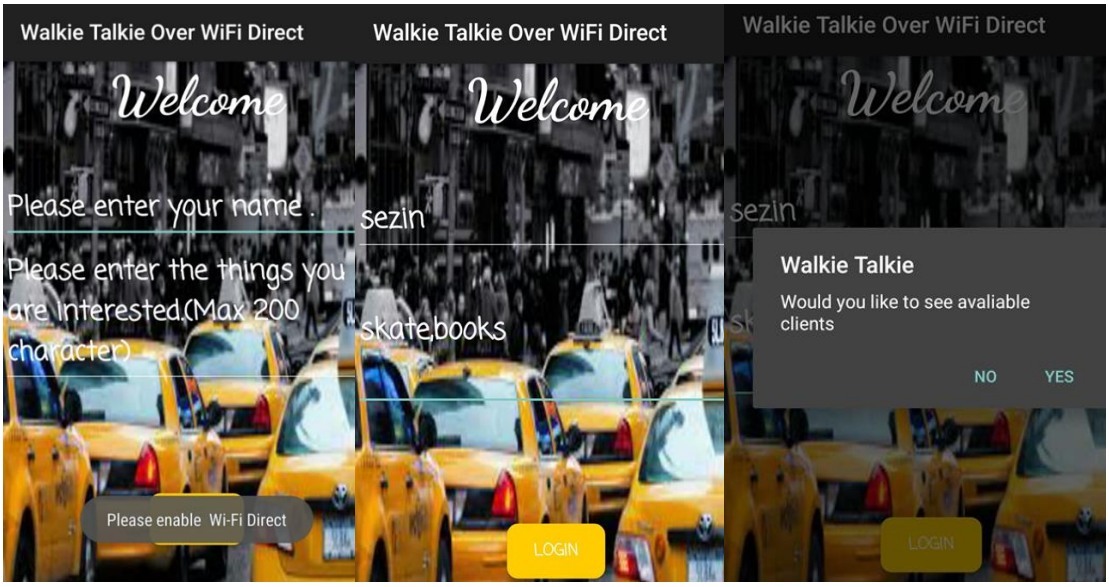

Figure 4.3: Login Screens of the Application

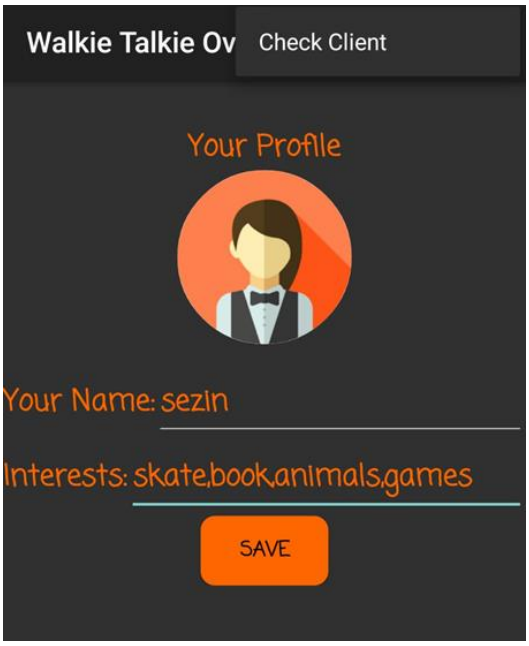

Figure 4.4: Profile Screen

Figure 4.5: Available Clients Screen

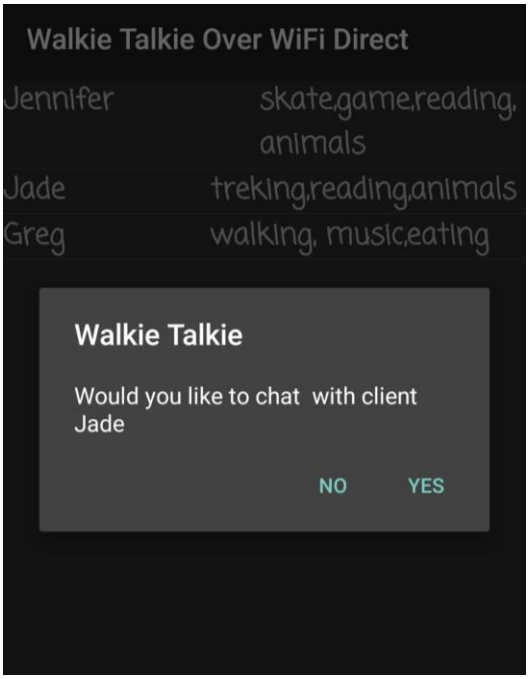

Figure 4.6: Send Request Screen

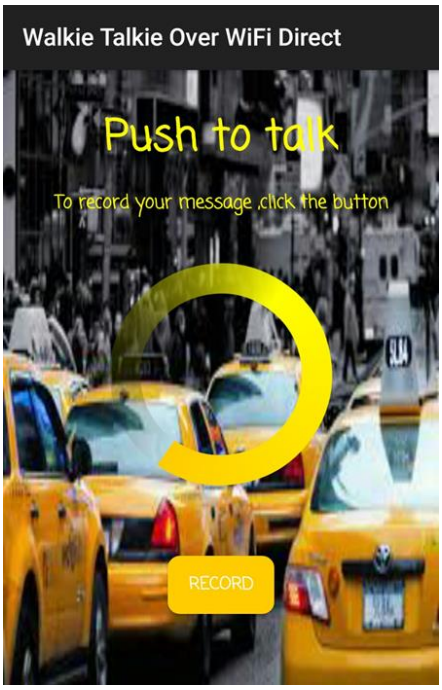

Figure 4.7: Record Message Screen

4.4. Test Plan

In this section, we are going to explain test plan of our project. As we recall application, this project implements push to talk (a.k.a. Walkie Talkie) by using VoIP (Voice over Internet Protocol) over Wi-Fi Direct. According to this description, here are test plan of this app:

- We are going to check that application can receive Wi-Fi Direct signal or not.
- After step 1 is done successfully, we are going to try to see available devices in Wi-Fi specific range.
- If all these two steps are done, we are going to send a request to one of the available devices. If this request can be responded successfully, we are done with step 3.
- We are going to check the data that entered in login screen is displayed correctly in home screen and another client's available devices list.
- We are going to send a file over Wi-Fi Direct to other device and check that this file is sent correctly and can be opened or not from other device in a small Wi-Fi Direct area. If we are done this step, then we will increase distance between two devices and again try same processes.
- After completed previous stage, we are going to do same process as step 5 however for this time, we will send a sound file to the other device.
- If we do the previous process successfully, we are going to try recording our voice message. The client will click record button in record message window then he/she will speak (she/he can speak max 30 seconds), after he/she is completed with recording her/his message, then this client will send this voice message. The other client will get voice message and listen to it.
- If all processes are completed successfully, then we will complete our testing process.

5. Software Architecture

In this section, we show control flow of the Push-to-Talk via Wi-Fi Direct Application. (See the Figure 5.1 on the next page)

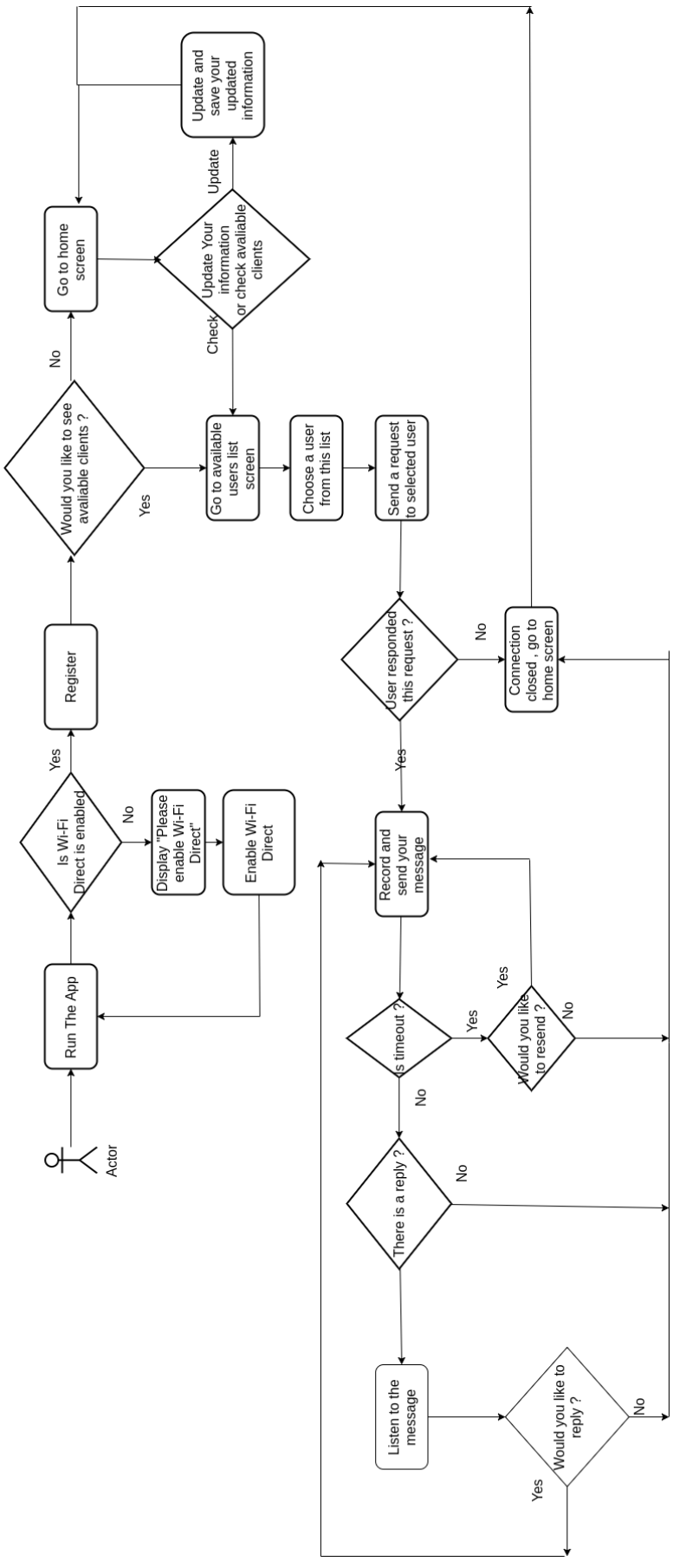

Figure 5.1: Control Flow of Push-to-Talk via Wi-Fi Direct Application

6. Tasks Accomplished

6.1. Current State of the Project

In this section, we are going to explain that what we have done so far.

First of all, after deciding our project topic, we specified that what is aim of our project, what we are going to do, which technologies we are going to use, which challenges we have during the implementation of our project. We decided to use VoIP, Wi-Fi Direct thus, we made a lot searching about them.

Secondly, we looked some android applications which are similar to our applications. However, unfortunately, we found a few this kind of applications. Thus, we decided to study android programming. Luckily, we both are familiar with android programming; hence we did not need to study android tutorials from beginning. We began seeking about how to use Wi-Fi direct in android programming. While we were making searches, we found some source codes about file transfer over Wi-Fi direct.

Now, we are done with our searches and we made our class, use case diagrams, control flow and started the implementation. After final exams, we are going to discuss about these diagrams for a week then we will continue with implementation.

6.2. Task Log

- 12 September 2017 13:00 We decided the topic of our project.
- 21 September 2017 15.30 – 16:00 We talked about things we are going to do during this year and VoicR app which is similar to our application and developed by Continental with our co-advisor.
- 28 September 2017 15.30 – 16:00 We talked about which technologies can be useful for our application with co-advisor
- 12 October 2017 15.30 – 16:00 We talked about which challenges we have for our project
- 14 October 2017 13:00 We sent an e-mail to Continental to get more information about VoicR

- 16 October 2017 17:00 -20:00 We started to our PSD document.
- 11 November 2017 14:00 – 19:00 We studied to Wi-Fi Direct tutorials in android programming
- 16 November 201715:30 -16:00 We talked about VoIP with TCP and UDP connections, SIP
- 23 November 2017 -15 December 2017 we made research about VoIP, Wi-Fi Direct.
- 30 November 2017 18:30 We talked about our A.D.D document
- 15 December 2017 – 20 December 2017 we prepared 497 presentation slides
- 24 December 2017 - 10 January 2018 we prepared A.D.D.

6.3. Task Plan with Milestones

In this section, we are going to explain our task between January and May.

Software design: We are going to check our UML Class Diagram, Use Case Diagram and Control Flow. If there is a necessary change, we will update our diagrams in a week.

Software Development: After, we finish our software design, we are going to start our software development. We are going to create our database, classes (if necessary), send voice messages over Wi-Fi Direct between clients in our application.

Testing: While we are developing software, we are going to make tests according to our test plan. Testing and software development will be take three months.

CSE-498 Report: From the last step of the software development, we will be writing CSE-498 report. Every step we do will be written in this report.

	January	February	March	April 1st - 15th	April 15th - 30th	May
Software Design						
Software Development						
Testing						
CSE 498 Report						

Table 6-1: Timetable for our project

7. References

[1]: Wikipedia, Android (operating system) Page [Online]. Available: https://en.wikipedia.org/wiki/Android_(operating_system) (Date of Access 09/01/2018)

[2]: Android Studio, User Guide Page [Online]. Available: https://developer.android.com/studio/intro/index.html (Date of Access: 09/01/2018)

[3]: Wikipedia, Wi-Fi Direct Page [Online]. Available: https://en.wikipedia.org/wiki/Wi-Fi_Direct (Date of Access: 09/01/2018)

[4]: Wikipedia, Voice over IP page [Online]. Available: https://en.wikipedia.org/wiki/Voice_over_IP (Date of Access: 09/01/2018)

[5]: Wikipedia, Session Initiation Protocol Page [Online]. Available: https://en.wikipedia.org/wiki/Session_Initiation_Protocol (Date of Access: 09/01/2018)

[6]: Wikipedia, SQLite Page [Online]. Available: https://en.wikipedia.org/wiki/SQLite (Date of Access: 09/01/2018)