# Introduction: Small Ranged Chat for easy conference (Bluetooth/Wi-Fi hotspot) UMeChat

A simple chat system that uses built-in mobile facilities to provide short distance and costless communication with additional file sharing features.

Purpose/Justification of my project

I am more open with words I type than what I say .According to a recent survey conducted by Open Market, when given the choice between being able only to text versus call on their mobile phone, a whopping 75 percent of millennials chose texting over talking. These people are able to express their thoughts by writing/typing more than they can express themselves verbally. Although it can have a negative impact as well but having an alternative way to express ourselves can be quite handy. Since a network (hotspot or Bluetooth connectively) requires a user defined password or Pairing matches as well, conversation can be carried out in a secret manner. My application would use a system that would use the full accessibility of phone's capability to improvise short ranged communication in an effective and costless manner. Also, private Conference can be held within a company for confidential data transferring. This application might not be much of a use but with new improvised ideas it can be upgraded furthermore. I would want it to be better as much as my thoughts could exceed furthermore so I might have to look at it again (redesign or additional features included) and probably java as my language as this application would be for mainly androids due to the wide use of android users so I want most of the people to be able to use this.

Background of my project

I am uncertain if this idea would work or even if such application exists already. But I have been inspired with this idea from the offline games we play through Wi-Fi-hotspot and Bluetooth pairing. So I am engaging my thoughts into making a system that would be beneficial to a certain small groupies. Providing an alternate source of communication medium somehow I want my system to be helpful to those who are lacked with self-confidence and are more open to typing rather than speaking. Being an introvert there are lots of things that goes through my mind which are unheard and unspoken so I think providing a simple application would also be helpful for some people.

Problems associated with my project

Firstly we have been dealing with C# as a programming language in my current prior college lessons and with due respect since I have no basic knowledge about Java the troubles I have to go through gathering information through different alternatives (i.e. YouTube/google) which would be time consuming and tiresome as well. Also I have no certainty if it is even possible regards with my current status. Beside personal problems negative aspects that would be on the mental health by using this application and possibly a social threat to socialism may occur.

Problem solving methodology

I would consider to take help from all the Medias as well as my experienced teachers. Since it is an individual project I just have to start off with this idea a lot quicker than others so that I can finish it in due time. Also I am planning on to take some additional classes on learning Java as well.

Description of my project

This project includes an alternate communication platform between two users which is co-relatively controlled. This project has two major types of users each representing one who creates a coverage area and ones within a coverage area that are relatively connected with each other through sources. By this connection they are allowed to transfer messages and files with each other.

Features of my project

-Alternative communication medium

-Costless network communication

-Secure communication

-Data sharing

-Feature to shared data backup

- (Additional features to be added)

Overview of my application

My application might not be useful in most context but for some people it might be precious and I think for developers like us we look for every possibilities we can come up with even if it is beneficial for some people only.

Scope

Scope of my Project

This project is created for connecting users in a small coverage area in a costless manner. Basically it includes chatting system for android device featured with Bluetooth and Wi-Fi-Hotspot accessibility. Also it creates an alternate platform for data privatization.

Limitation of my Project

The Limitation of my project are illustrated below:

* This system does not covers a wide range area making long ranged communication not possible without prior equipment’s.
* This system is not on practice making it vulnerable on demands.
* This system might bring conflict in ones day to day life.
* This system demotivates people from social collaboration verbally.

Aims of my Project

The aims of my project are illustrated below:

* To facilitate users with costless communication medium platform.
* To help respective user’s to fully potential communicate with each another.
* To promote confidential data sharing facility in a costless manner.
* To promote privatization of user confidential messages.

Objective of my Project

The objective of my project are illustrated below:

* This chat system encourages confidential message transfer in a costless manner.
* This chat system fully facilitates the functionality of owner’s devices.
* This chat system encourages collaborative groupies.
* This chat system ensures data protection.

Overview of my Scope

A well-defined Scope mainly helps in completion of task efficiently within assumed time rate in a distinctive way. Thus my Scope would help me cover-up all the things included above accordingly systematically and in effective manner keeping in mind my main aim is costless communication.

Development Methodology

Description of Methodology

Software development methodologies are the frameworks that are basically used to plan, structuralize, and control the processes in developing of information system wholly.

Agile Software Development Methodology

Agile Software Development is an alternate approach of software development in which phases are carried out in a non-respective manner where solutions and required facilities can be furthermore enhanced by collaboration evolved by enabling pre remodeling and re-featuring. This method mainly emphasize on real time communication and working software as the primary measure of progress in software development. This method compare to others produce very small amount of written documentation due to this emphasized methods.

Advantages of using Agile Methodology

* It produces high product quality.
* It provides customer with higher satisfaction.
* It provides transparency and higher project manageability.
* It has comparatively lower chances of Risks.
* It provides features which are delivered incrementally.

Agile Methodology over Waterfall Methodology

Waterfall Method being very different to agile method especially because it’s not iterative method, Waterfall is more about a process where you can see the progress route through the difference phases. It’s a sequential model defined from requirement analysis, design, implementation, testing to maintenance. Whereas agile development tends to deliver adaptability, visibility and value in the beginning of the process and reduces a lot the risks during the project development.

Design Pattern

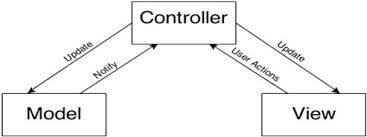
MVC (Model-View-Controller Pattern) is a design pattern associated with my project.

This Pattern involves three major criteria which are as listed below:

**- Model**: It represents an object carrying data which can also have logic to update controller if its data alters.

**-View**: It represents the visualization of the data that model contains and usually has the UI logic.

**-Controller**: It controls references both Model and view. It controls the data flow into model object and updates the view whenever data changes accordingly and keeps View and Model separately.

****

**Fig: MVC Design Pattern**

**Advantages of MVC**

- It is comparatively faster development process **as** it supports rapid and parallel development. With MVC, multiple person can work on different sector.

- It provides multiple views**.** As you can create multiple views for a model it is very helpful. Code duplication also limited in MVC.

-MVC also supports asynchronous technique, which helps developers to develop an application that works very quickly.

- Any Modification does not affects the whole entire model asbecause model part does not depend on the views part. Thus, any changes in the Model will not affect the whole architecture.

- MVC pattern returns data without applying any formatting so the same components can be used and called for use with any interface available.

**Disadvantages of MVC**

-MVC has increased complexity.

-It has inefficiency of data access in view.

-It is difficult to use MVC with modern user interface.

-It requires multiple programmers.

-It requires knowledge on multiple technologies.

System Architecture

The interface, features and structure of the system is categorized into an architecture model referred as system architecture. These can be further categorized into three different tier models which are as listed below:

* Presentation tier: Top most appearance layer of the application.
* Logic tier: Logical decisions and evaluations based layer of the application.
* Data tier: Data Storage layer of the application.

Presentation Tier

Data Tier

Logic Tier

Fig: System Architecture

Why use 3 Tier Architecture?

There are lots of benefits for using a 3-layer architecture which includes speed of development, scalability, availability, and performance.  Modularizing different tiers of an application gives development teams individually the ability to develop and enhance a system with greater speed than developing a singular code base because a specific tier can be changed with minimal impact on the other layers.

Project Planning

Work Breakdown Structure

Introduction: WBS (Work Breakdown Structure) is the way to break down team’s work collectively into small distinct areas for easy and effective manageability. By assigning a time assumption for task completion a certain task is carried out individually and the project is carried out wholly in this system also referred as Work Breakdown System. Although we are carrying out agile model as development model and since the project completion time is precise but furthermore thoughts which may evolve can be added to this system eventually so it helps to maintain project basic task that needs to be completed in due date and time. The activities that shall be undertaken individually during this project completion is diagrammatically shown below in the following hierarchical chart given below:

Scoping

Planning

Presentation

User Preference

Interface Design

Interface Design Model

Architecture

Documentation Process

Analysis Process

Design Process

Implementation Process

Testing Process

Final Report

User Manual

Integration Testing

Unit Testing

Code Implementation

Database Design

Structural Model

Behavioral Model

Use Cases

Requirement Collections

Monitoring & Controlling

Project Management Process

UMeChat System

Fig: Work Breakdown Diagram

The reason to adopt Work Breakdown Structure are as follows:

* Accurate time estimation and effort required for project.
* Easier and faster project planning.
* Since it is assigned to smaller groups task-wise easy work handling.
* Reduces Risk to the projects.

|  |  |  |
| --- | --- | --- |
| WBS | Task Name | Days |
|  | **Project: UMeChat System** | **109** |
| **1** | **Project Management Process** | **15** |
| 1.1 | Scoping | 5 |
| 1.2 | Planning | 5 |
| 1.3 | Monitoring & Controlling | 5 |
| **2** | **Analysis Process** | **29** |
| 2.1 | Requirement Collections | 9 |
| 2.2 | Use Cases | 10 |
| 2.3 | Architecture | 10 |
| **3** | **Design Process** | **26** |
| 3.1 | Structural Model | 6 |
| 3.2 | Behavior Model | 8 |
| 3.3 | Interface Design Model | 12 |
| **4** | **Implementation Process** | **21** |
| 4.1 | Database Design | 3 |
| 4.2 | Code Implementation | 14 |
| 4.3 | Interface Design | 4 |
| **5** | **Testing Process** | **7** |
| 5.1 | Unit Testing | 2 |
| 5.2 | Integration Testing | 2 |
| 5.3 | User Preference | 3 |
| **6** | **Documentation Process** | **11** |
| 6.1 | User Manual | 4 |
| 6.2 | Final Report | 4 |
| 6.3 | Presentation | 3 |
| **7** | **Total** | **109** |

Milestone

Milestone remarks a project phase from one to another starting from project’s begin to project’s end date. The start and end time of all individual task carried down by Work Breakdown System is calculated with a milestone with task completion step wise. It is used to measure the progress of project towards its goal within project lifecycle. By using a milestone the task completion date can be pre-estimated and project due date can be fixed through its lifecycle.

**Advantage of using Milestone**

-Easy Accountable of work Progress including Deadline.

-Work Phase Time Consumption pre-estimated.

- Provides ease Data handling compatible medium

-Identify Critical path in working phases.

The following milestone of my project is tabulated and displayed below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N. | Milestones | Start Date | End Date | No of Days |
| 1 | Project Proposal Phase   1. Scoping 2. Planning 3. Monitoring & Controlling | 26th March 2019  26th March 2019  31st March 2019  5th April 2019 | 9th April 2019  30th March 2019  4th April 2019  9th April 2019 | 15  5  5  5 |
| 2 | Analysis Phase   1. Requirements Collections 2. Use Cases 3. Architecture | 10th April 2019  10th April 2019  19th April 2019  29th April 2019 | 8th May 2019  18th April 2019  28th April 2019  8th May 2019 | 29  9  10  10 |
| 3 | Design Phase   1. Structural Model 2. Behavior Model 3. Interface Design Model | 9th May 2019  9th May 2019  15th May 2019  23nd May 2019 | 3rd June 2019  14th May 2019  22nd May 2019  3rd June 2019 | 26  6  8  12 |
| 4 | Implementation Phase   1. Database Design 2. Code Implementation 3. Interface Design | 4th June 2019  4th June 2019  7th June 2019  21st June 2019 | 24th June 2019  6th June 2019  20th June 2019  24th June 2019 | 21  3  14  4 |
| 5 | Testing Phase   1. Unit Testing 2. Integration Testing 3. User Preference | 25th June 2019  25th June 2019  27th June 2019  28th June 2019 | 1st July 2019  26th June 2019  28th June 2019  1st July 2019 | 7  2  2  3 |
| 6 | Documentation Phase   1. User manual 2. Final Report 3. Presentation | 2nd July 2019  2nd July 2019  6th July 2019  10th July 2019 | 12th July 2019  5th July 2019  9th July 2019  12th July 2019 | 11  4  4  3 |

-Project Proposal Phase: I have given 15 days prior to proposal phase as it requires less effort and can be carried out easily. Also in this phase Scoping, Planning & Monitoring/Controlling co-exists. Also I have equally associated time to each sub-phases in this phase. I.e. 5 days.

-Analysis Phase: I have given 29 days prior to analysis phase as it is the phase we gather information and the most time consuming phase. In this phase Requirement collections, Use Cases & Architecture co- exists. I have associated most days to Use cases & Architecture as it takes certain criteria to follow up. I.e. 10 days.

-Design Phase: I have given 26 days prior to designing phase as it consumes much effort and is relatively associated with designing of components of my project. In this phase Structural Model, Behavioral Model & Interface Design Model co-exists. In this phase I have given more prior to Interface Design Model as it must have an easy and effective interface. I.e. 12 days.

-Implementation Phase: I have given 21days prior to implementation where I mainly focused on coding parts. In this Project Database Design, Code Implementation & Interface design co-exists. In this phase I have given most time to Code Implementation as because it is the toughest part where code are imported in a working manner. I.e. 14 days.

-Testing Phase: I have given 7 days prior to this phase as it is a small process just to check if the system works out as expected. In this phase Unit testing, Integration testing & User Preference co-exists. I have given most time consumption to User Preference as because gathering information on one individual is not adequate so taking many user preference it requires more time. I.e. 3 days.

-Documentation Phase: I have given 11 days prior to this phase as it mainly focuses on elaborating system in a visualized way. In this phase User manual, final report & Presentation co-exists. I have given the most time to User manual & final report as because presentation does not require much effort. I.e. 4 days.

Scheduling

Gantt chart is one of the commonly used project management time scheduled Chart use to distinguish task phase carried out throughout the project. The Schedule of my respected project is shown below with the presentation of Gantt chart as well.

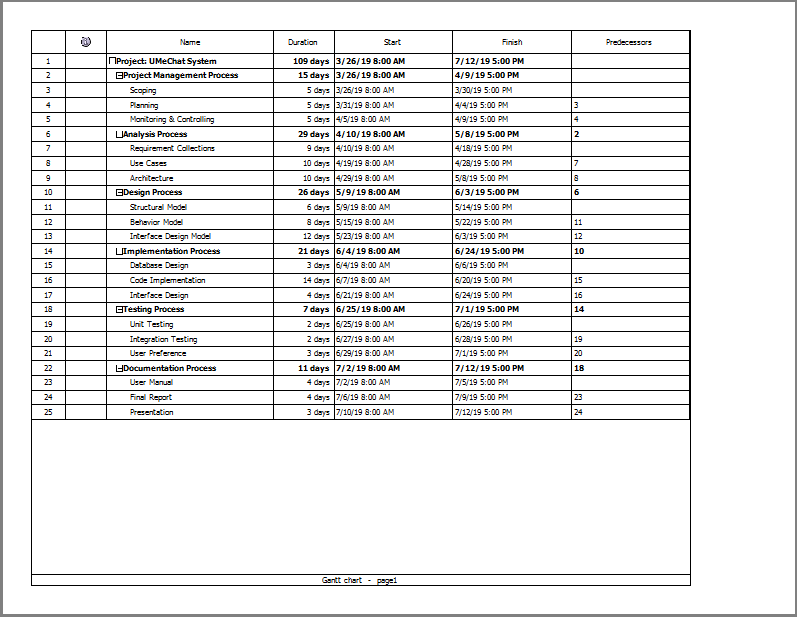
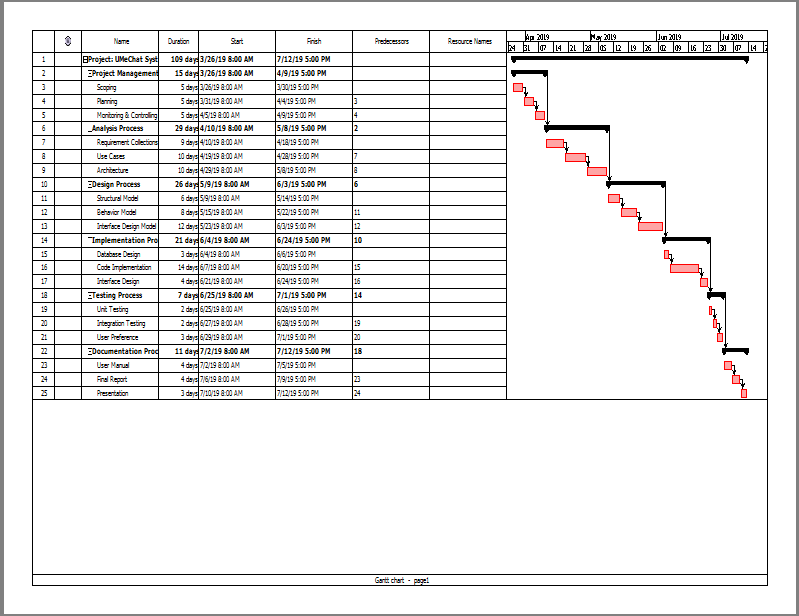
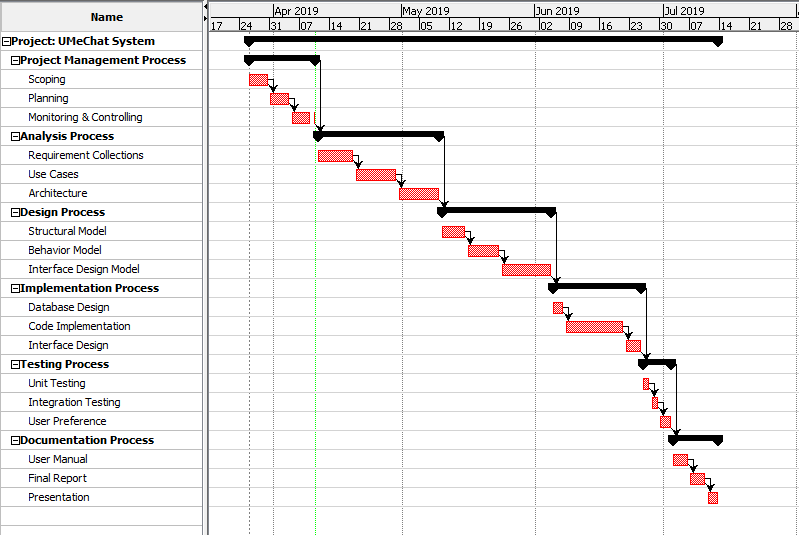


Fig: Scheduling

Gantt chart Fig: Gantt chart page: 1Fig: Gantt chart page: 2

# Analysis

1. **Introduction to Analysis:**

The conversion method in which difficult task is split out into many more phases to simplify the task and make it understandable is known as Analysis. It can be regarded as one of the most important phase of software development. It is done to evaluate user demands and project goals. Firstly information is gathered, a feasibility study is carried out the user demands and project goals is maintained thoroughly. Also features of the system are estimated in this phase for both hardware and software component. In agile development methodology Analysis is done prior second to Planning project. Wholly speaking of Analysis there are still many processes to be carried out for the completion of analyzing phase in software development.

1. **Analysis methodology:**

Software development methodology can be described as the frame work which is used to plan, structure and control the process of developing system. My UMeChat project will follow the SDLC model throughout process of development. Software Development Life Cycle (SDLC) is a process used by the software organization in order to design, develop and test high quality soft-wares. Also I would be considering Soft System Approach. In this process engineering approach can be inappropriate with fuzzy requirements.

Advantage of Soft system Approaches:

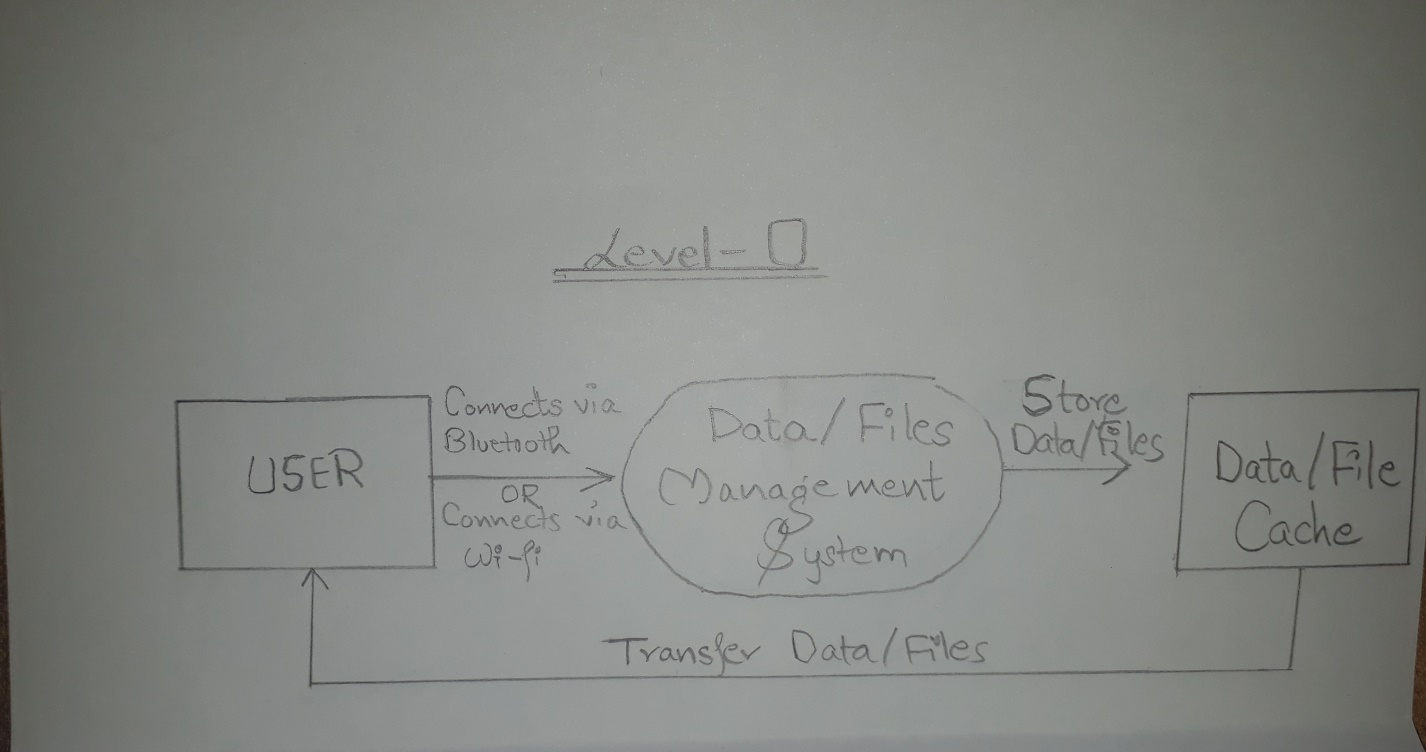
1. Human factors important so given more priority.
2. It is a creative, intuitive approach to problem-solving.
3. Outcomes are learning, better understanding, rather than a ‘solution’.

Disadvantage of Soft system Approaches:

1. The organizational problems are ‘messy’ and poorly defined.
2. There are many problems to be solved.
3. It can have unpredictable, nondeterministic, non-definable complexity.

* **Data Flow Diagram:**

Data flow diagram is the graphical representation of the major phases of project process.



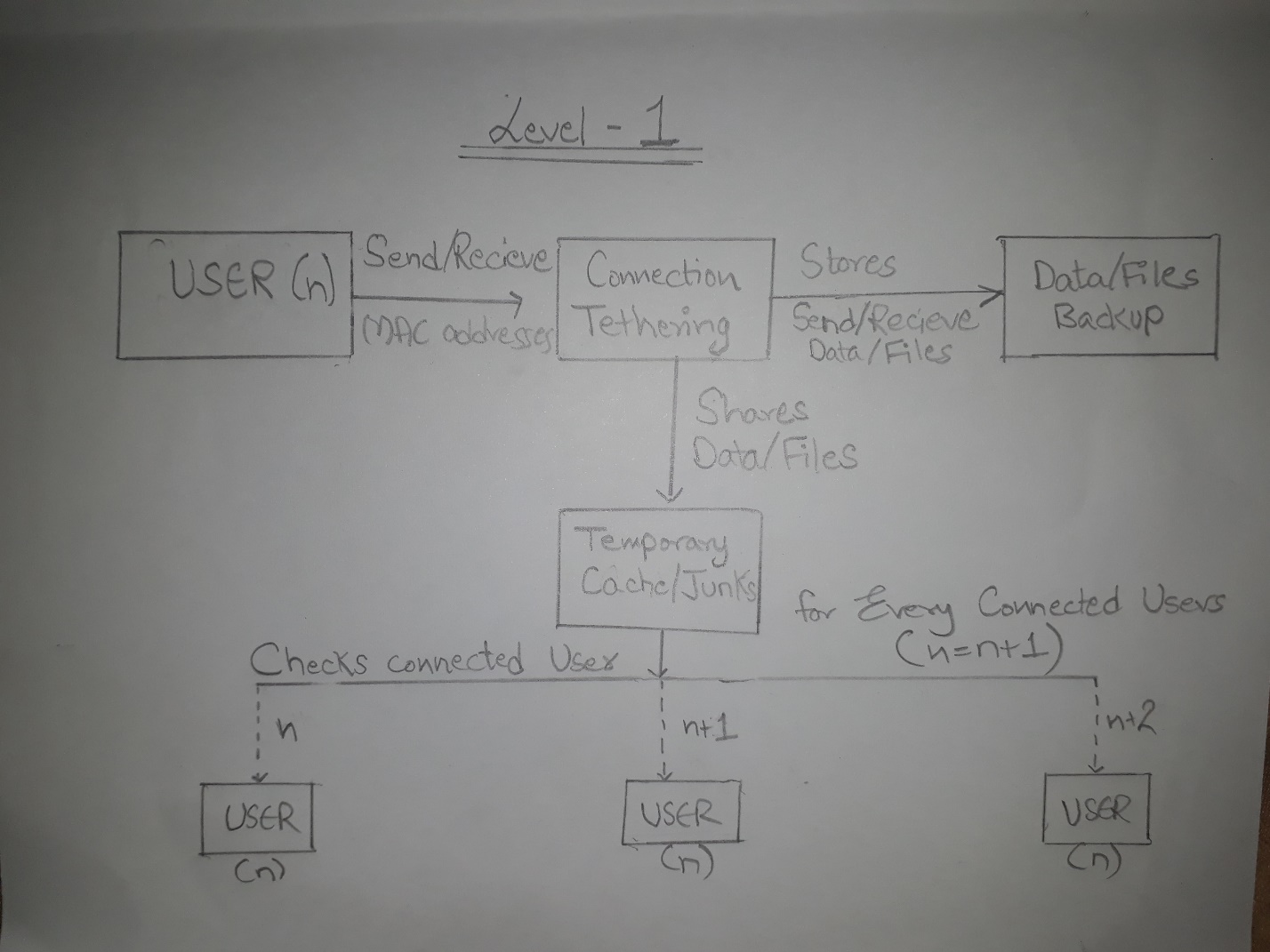


Fig: DataFlow Diagram

1. **Feasibility study:**

Feasibility study is used in determining the viability of idea, such as ensuring a project is legally and technically feasible as well as economically justifiable.

1. Technical Feasibility: Rapid growth in progress of project can be expected. It would be reasonable and pretty much comparative to upcoming technologies.
2. Economic feasibility: It will reduce the cost within effective communication. It is also to bring benefits to the owner. It would be affordable and reliable and would be relevant to project scope.
3. Legal feasibility: Any change in legalities issued by government the product marked price may alter.
4. Operational Feasibility: It evaluates problem solving methods and takes advantages of opportunities provided with the represented course of project.
5. Scheduling Feasibility: Simply put it is the time management process of work completion where work are further sub divided and is regarded as one of the vital process of study.
6. **Software Requirement Specification(SRS):**

In software development SRS is also most important task. It can be defined as the process of gathering information from the user to solve problems. If the collected information is incorrect then whole project leads to failure. For this specification of such projects must be done carefully and effectively. There are two types of the requirements they are:

1. Functional Requirement
2. Non-functional Requirement
3. **Functional Requirement:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Title | Description | Rational | Dependencies |
| FR01 | Bluetooth Activation | User can activate Bluetooth. | Connectivity. | N/A |
| FR02 | Wi-Fi Activation | User can activate Wi-Fi. | Connectivity. | N/A |
| FR03 | Bluetooth Discoverable | User can enable Discoverability. | Discoverability Enabling. | FR01 |
| FR04 | Bluetooth Pairing | User can connect to other devices within its coverage area through Bluetooth, | Pairing. | FR01,FR02 |
| FR05 | Wi-Fi Pairing | User can connect to other devices within its coverage area through Wi-Fi hotspots, | Pairing. | FR02 |
| FR06 | Adding suitable nicknames and avatars | User can give connected devices suitable nicknames and avatars. | Connected Devices evaluation. | FR04,FR05 |
| FR07 | Creating a group | User can create a group. | User can name the group and choose a suitable image for the group image. | N/A |
| FR08 | Adding to a group | User can add people to a group. | User can add connected devices in a group. | FR04,FR05 |
| FR09 | Removing from a group | User can remove people from a group. | User can remove connected devices in a group. | FR08 |
| FR010 | Editing a group | User can edit a group. | User can rename the group and re-choose a suitable image for the group image. | FR07 |
| FR011 | Deleting a group | User can delete a group. | User can delete a created group. | FR07 |
| FR012 | Messaging | User can message to the connected devices within or excluded from a group. | User can send/receive text messages to the connected devices. | FR04,FR05 |
| FR013 | File Transfer | User can send or receive files with connected devices. | User can send/receive files to the connected devices. | FR04,FR05 |
| FR014 | Interactive UI | User can easily operate on the program with ease with better user interface. | User can use the program  Without facing any difficulties. | N/A |
| FR015 | Add themes to Chats | User can add themes on the chat background. | User can define his/her own chatting background. | N/A |
| FR016 | Send/Receive stickers to Chats | User can send/receive stickers. | User can make the chat interesting with cool new updated stickers. | N/A |
| FR017 | Listing Available Devices | User can be the available devices on the network. | User can see the device availability. | N/A |

1. Non-functional Requirement:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Title | Description | Rational | Dependencies |
| NFR01 | Security | User can maintain security through connections. | Effective Security | N/A |
| NFR02 | Availability | Program should be easily available to everyone. | Easily Available | N/A |
| NFR03 | Reliability | Program system should be reliable in terms of efficiency. | Most Reliable | N/A |
| NFR04 | Maintainability | System should be easier to handle. | Maintainable | N/A |
| NFR05 | Scalability | The system should be adapting changes per demands. | Expansion of system | N/A |
| NFR06 | Testability | System should be tested. | Well Tested | N/A |
| NFR07 | Interface | System should have better user interface (UI). | Well Interface | N/A |
| NFR08 | Performance | System should work without any obligation. | Best Performance | N/A |
| NFR09 | Manageability | System should have easy manageability. | Well Manageable | N/A |
| NFR010 | Recoverability | System should recover lost data or files. | Data Recovery Option | N/A |
| NFR011 | Data integration | System should provide security to data. | High Data Security | N/A |
| NFR012 | User friendly | System should be easy to use and well equipped. | Best equipped facilities | N/A |

1. **Moscow Prioritization:**

It helps to define the requirements prioritization. It also helps to assigning time and helps to development process of the system. Moscow also used to prioritize the requirement of system.

* Must have: important requirements of system that is very essential.
* Should have: it is not much important to system but it expand its scope.
* Could have: it is not essential to the system.
* Won’t have: the requirements that is not necessary to the system.

The prioritization of requirement for my project given below:

For Users

|  |  |  |
| --- | --- | --- |
| S.N | Functional requirement | Moscow |
| FR01 | Bluetooth Activation | Must have |
| FR02 | Wi-Fi Activation | Must have |
| FR03 | Bluetooth Discoverable | Must have |
| FR04 | Bluetooth Pairing | Must have |
| FR05 | Wi-Fi Pairing | Must have |
| FR06 | Adding suitable nicknames and avatars | Could have |
| FR07 | Creating a group | Could have |
| FR08 | Adding to a group | Should have |
| FR09 | Removing from a group | Could have |
| FR010 | Editing a group | Should have |
| FR011 | Deleting a group | Should have |
| FR012 | Messaging | Must have |
| FR013 | File Transfer | Must have |
| FR014 | Interactive UI | Must have |
| FR015 | Add themes to Chats | Could have |
| FR016 | Send/Receive stickers to Chats | Could have |
| FR017 | Listing Available Devices | Should have |

Non-functional requirements

|  |  |  |
| --- | --- | --- |
| S.N | Non-functional requirements | Moscow |
| NFR01 | Security | Must have |
| NFR02 | Availability | Should have |
| NFR03 | Reliability | Must have |
| NFR04 | Maintainability | Should have |
| NFR05 | Scalability | Must have |
| NFR06 | Testability | Should have |
| NFR07 | Interface | Must have |
| NFR08 | Performance | Should have |
| NFR09 | Manageability | Should have |
| NFR010 | Recoverability | Should have |
| NFR011 | Data integration | Won’t have |
| NFR012 | User friendly | Must have |

1. **Hardware/Software specification:**

Hardware Specification

 Processor: Intel(R) Core (TM) i5-5200U CPU @ 2.20GHz 2.20

 Ram: 4GB or higher

 Hard Disk: 100 GB or higher

 Display Type: Standard VGA or SVGA card

 Peripherals: Keyboard, Mouse

Software Specification

 Operating system: Windows 7 or higher, Linux

 Front-end: Android Studio

 Back-end: None.

1. **Use Case Diagram:**

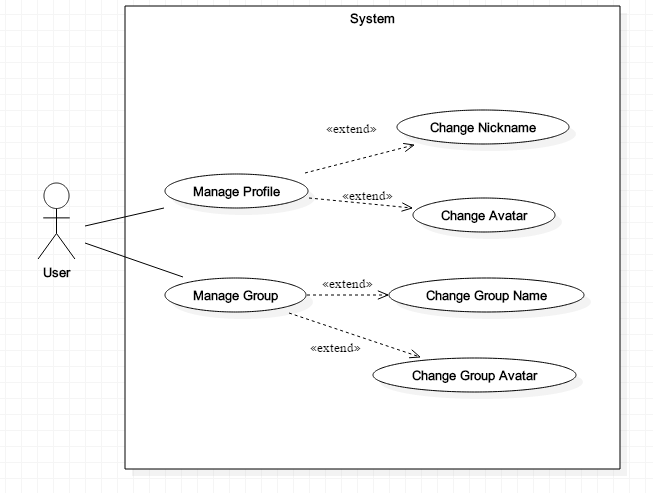
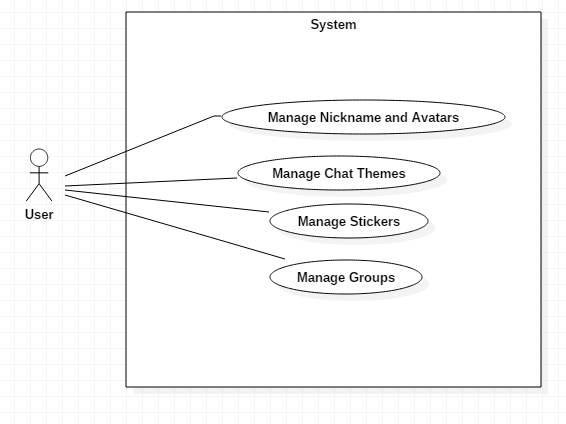
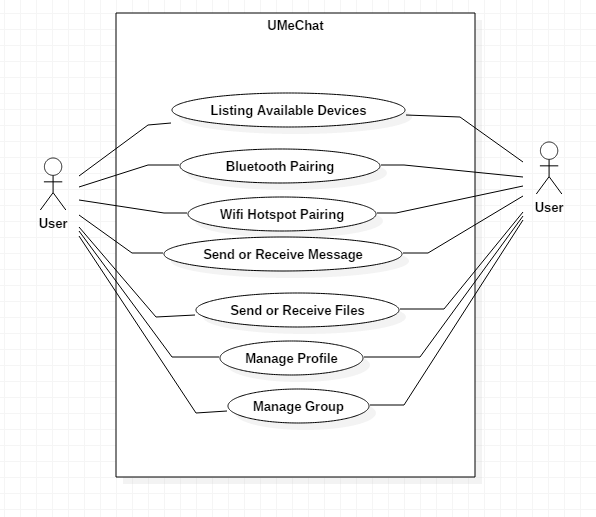
Use case diagram is a diagrammatic interaction between actor and system drawn in UML. It represents user interaction in diagram and shows relation between system and user. It is very effective in developmental process. 

Fig: Use case diagram

Title: “UMeChat”:

|  |  |
| --- | --- |
| Id | U1 |
| Primary actor | Users |
| Supporting actor | N/A |
| Primary flow | 1. User can list out available users. 2. User can pair with each another. 3. User can send or receive messages. 4. User can send or receive files. 5. User can manage each profile with nickname and avatar. 6. User can manage group name and avatars. |
| justification | Shows the system working mechanics. |
| Alternative flow | N/A |

Title: “System-1”:

|  |  |
| --- | --- |
| Id | U2 |
| Primary actor | Users |
| Supporting actor | N/A |
| Primary flow | 1. User manage nickname and avatars for connected devices. 2. User manage chat themes available. 3. User manage chat stickers. 4. User manage group name and avatars. |
| justification | Shows basic necessary system data to be included. |
| Alternative flow | N/A |

Title: “System-2”:

|  |  |
| --- | --- |
| Id | U3 |
| Primary actor | Users |
| Supporting actor | N/A |
| Primary flow | 1. User can change the nickname and avatars. 2. User can change the group name and avatars. |
| Justification | Shows how users can manage connected devices within or excluded from a group. |
| Alternative flow | N/A |

**NLA (Natural Language Analysis)**

Natural Language Analysis can be defined as the use of ability of systems to process sentences in a natural language rather than in artificial computer language such as C++ or C# and separating out nouns and adjectives.

|  |  |  |
| --- | --- | --- |
| Nouns: classes | Verbs: functions | Adjective : attributes |
| Users | Connect | Detail |
| Application | Send | Color |
| Chat | Receive | Theme |
| Bluetooth | Change | User\_Id |
| Wi-Fi | Manage | Files |
| Stickers | Edit |  |
| Avatar | Recover |  |

**Initial class diagram:**

Initial class diagram is the diagrammatic representation of static view of application. It is further classified into attributes and operations breaking them into a singular tabular format for easy configuration and construction of codes. Thus it provides a static representation of application. Initial class diagram consists of three types of relationship which are associations, compositions and dependency. Initial diagram of UMeChat is illustrated below:

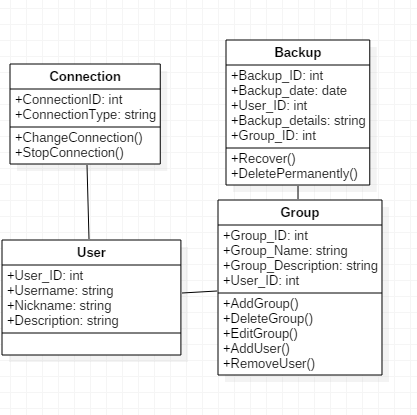


Fig: Initial Class Diagram

# Design

**Introduction to Design**

The basic process of realization of concept into model, configuration, pattern, plan or specification which helps to achieve certain designated objectives is also known as the term design.

**Structural Model**

It can be defined as a set of interconnected components which compositely represents a structure in accordance to defined data.

**Final Class Diagram**

In object oriented modelling Class Diagram are the main building block or component which are used to represent different objects in a system.

**Behavioral Model**

It can be defined as a set of sequential assignment statements which compositely represents a behavior in accordance to defined data.

**Flowchart (Data Flow Diagram)**

Flowchart can defined as a visual representation of sequence of steps and decisions required to perform the processes.

**Flowchart Notations**

Start Box: Represents Start/ End Point

Process Box: Represents Process Point

Decision Box: Represents a Decisive Point

Data Input/output Box: Represents input or output Point

Arrow: Represents Relationship between representative shapes

**System Flowchart**

Start

Dashboard

Connection

Hotspot Pairing

Bluetooth Pairing

Paired Successfully?

YES

NO

Add to Chat

Add Device Details

End

Start

End

Add to Group

Removed Successful

YES

NO

Confirmation?

Remove People to Group

YES

NO

Delete Group

Delete Successful

Confirmation?

Add People to Group

Edit Group Details

Add Group Details

Edit Group

Create Group

Dashboard

**Activity Diagram**

Activity diagram is the behavioral diagram in UML representation which describes dynamic aspects of the system.

**Justification:**

It is regarded as essential tool in an analyst’s repertoire.

It is easily comprehensible for both analysts and stakeholders.

Easily understandable to end users.

**Activity Notations**

Action: Represents activity or processes.

Initial: Represents starting point.

Final: Represents ending point.

Fork node: Represents multiple concurrent flows.

Join node: Represents singular flows of multiple activity.

Decision: Represents decision activity.

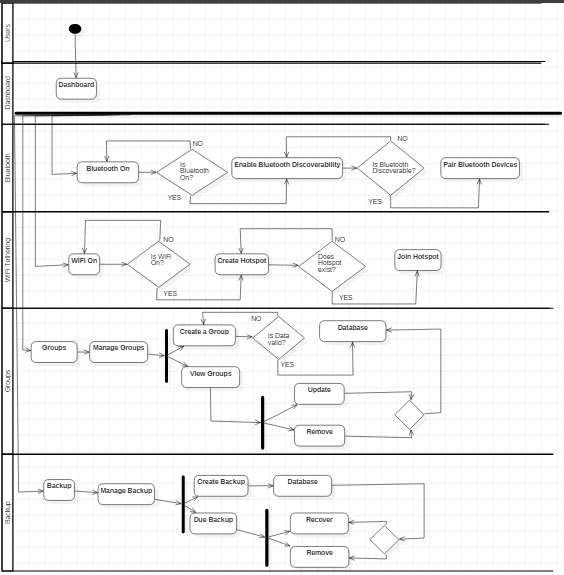
Action Flow: Represents flow of activity.

Activity Partition

Swim lane: Represents group related activities.

It can be both vertical and horizontal.

Actual Activity



This activity diagram is a simple illustration of my UMeChat Application System. Firstly User are to be upfront to dashboard where they can either make a connection through Bluetooth or WiFi tethering through which the whole files and data transferring takes place. Also User can create a group and add or exclude people from it as well as maintain backup regularly so that the files/data are stored for sustainable period.

Since it is for no respective organization and used for personal it does not require Admin and no login system.

**Sequence Diagram**

Sequence Diagram is the behavioral diagram in UML representation which defines the functionality of the system.

**Justification**

It describes how objects altogether work.

It is used by IT professionals to understand the requirement for a new system.

It is also known as event diagram.

It describes the logic of sophisticated function.

**Sequence Notations:**

Actor: It represents entity.

Lifeline: It represents time event of activities.

Object: It represents objects.

Synchronous Message: It represent flow of message at

A same instance.

Asynchronous Message: It represents flow of message at

A different instance.

Self-Message: It represents flow of message within itself.

ALT

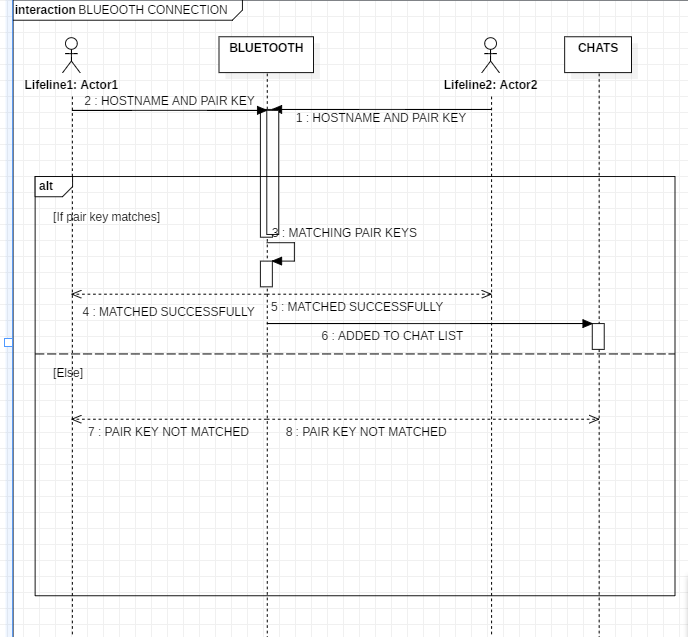
Alternative: It represents alternate activity.

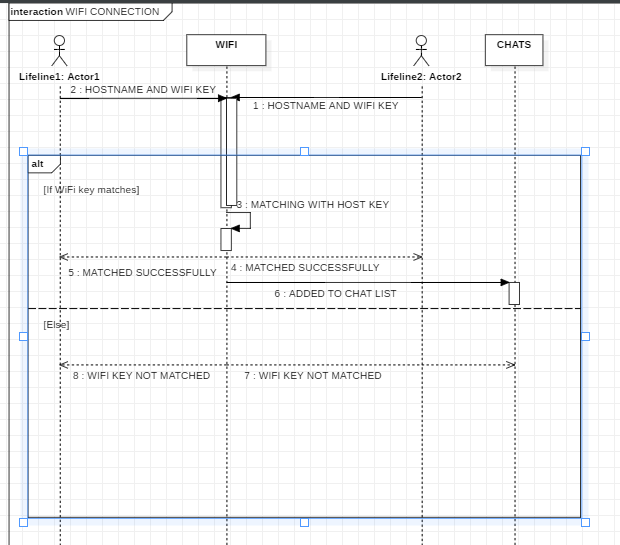
OPT

Optional: It represents optional activity.

LOOP

Loop: It represents looping activity.

Actual Sequence DiagramThis is the sequence diagram where a user can connect to the other user via Bluetooth interaction and this simply is the illustration of the sequence of Bluetooth Connection.



Whereas this is the sequential diagram for connection through WIFI which is similar to Bluetooth. In these both scenario a user can only access permission if both the defined user have paired up or match the password to host set up password through which chat system would be accessible to both of them.

**ER DIAGRAM**

Entity relationship diagram are the data structure model that represents entities and their relationship which can be one to one, one to many, many to many.

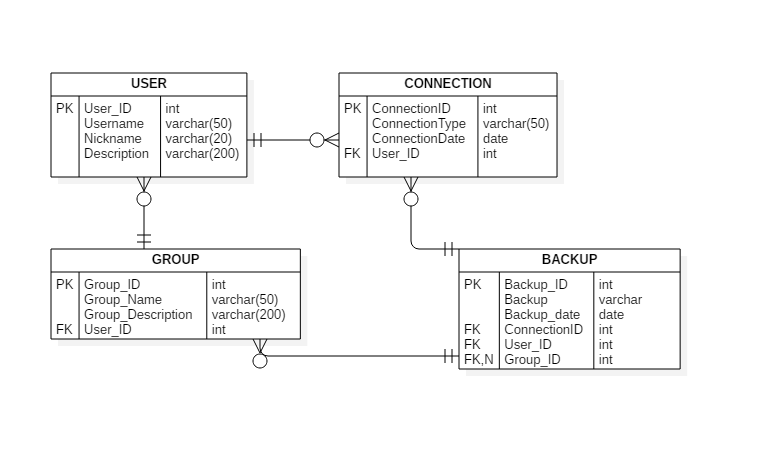


Fig: ENTITY RELATIONSHIP DIAGRAM

**DATABASE MODEL**

It is the data model which represents logical structure of database and defines how a data is to be stored.

**Data Dictionary**

It can be defined as a set of information within information I.e. it defines the contents, formats and structure of database.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TABLE NAME | ATTRIBUTES | DATATYPE | LENGTH | PRIMARY KEY | FOREIGN KEY | NULLABLE |
| USER | User\_ID | int |  | YES | NO | NO |
|  | Username | varchar | 50 | NO | NO | NO |
|  | Nickname | varchar | 20 | NO | NO | NO |
|  | Description | varchar | 200 | NO | NO | NO |
| GROUP | Group\_ID | int |  | YES | NO | NO |
|  | Group\_Name | varchar | 50 | NO | NO | NO |
|  | Group\_Description | varchar | 200 | NO | NO | NO |
|  | User\_ID | int |  | NO | YES | NO |
| BACKUP | Backup\_ID | int |  | YES | NO | NO |
|  | Backup | varchar |  | NO | NO | NO |
|  | Backup\_date | date |  | NO | NO | NO |
|  | ConnectionID | int |  | NO | YES | NO |
|  | User\_ID | int |  | NO | YES | NO |
|  | Group\_ID | int |  | NO | YES | YES |
| CONNECTION | ConnectionID | int |  | YES | NO | NO |
|  | ConnectionType | varchar | 50 | NO | NO | NO |
|  | ConnectionDate | date |  | NO | NO | NO |

**ARCHITECTURAL MODEL**

Architectural model can be defined as the scaling model specified to meet design of the whole system. My architectural model is followed by 3 tier architecture and they are as follows:

1. Client Layer

It is also defined as a presentation layer as because it contains User interface part of our application. It is the main designing phase.

1. Business Layer

It is also defined as a logical layer as because it contains all the business logic like data validation, calculation, etc.

It acts as an interface between Client and Data Layers.

1. Data Layer

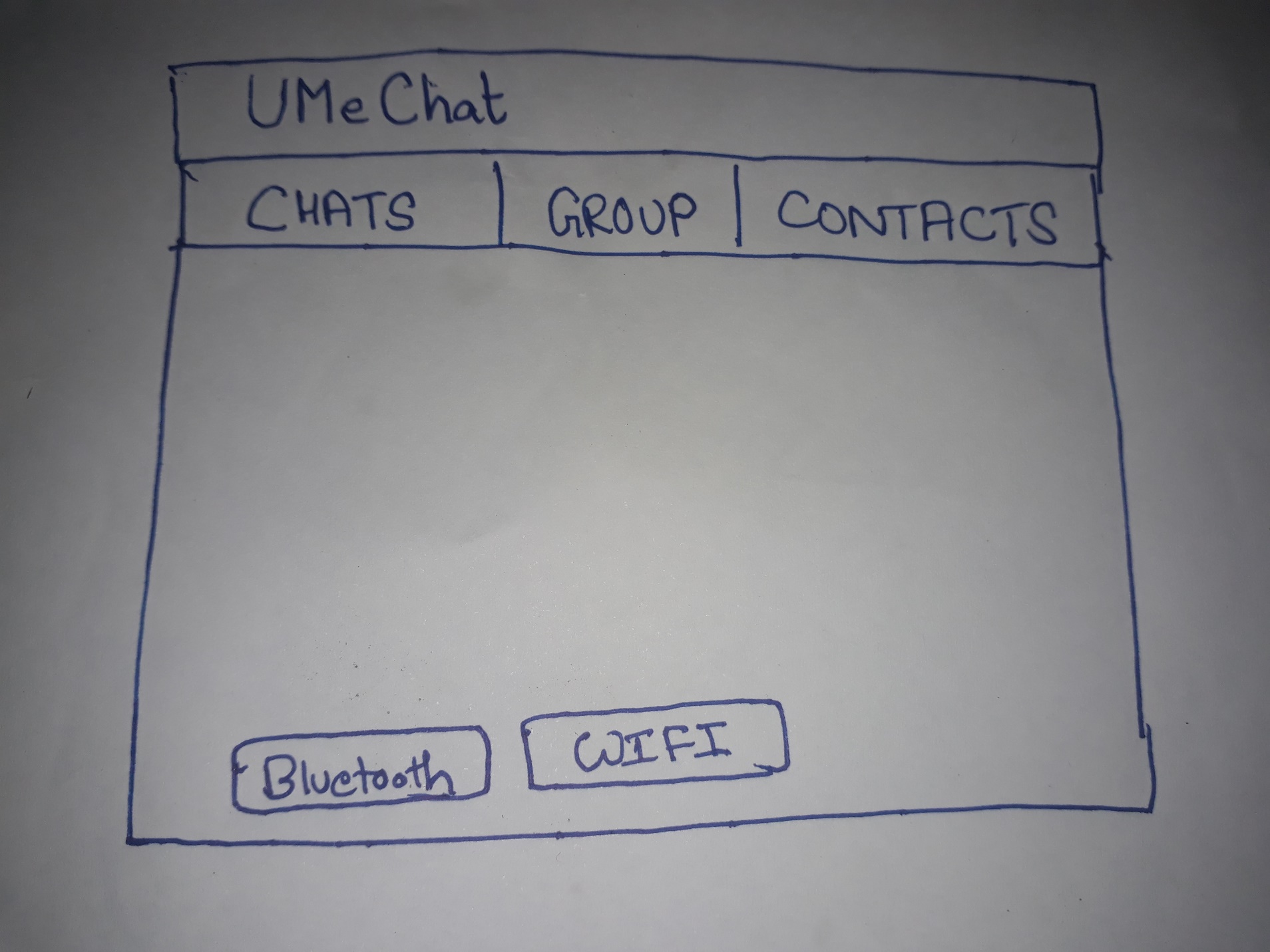
It is also defined as a data layer as because it contains all the data manipulation methods from our database. Certain facilities like data insertion, deletion, and update are carried out in this layer.

**UI MODELLING**

User Interface is the modelling technique used by programmers for easy and effective ongoing system.

**PROTOTYPING**

It can be simply defined as sample model of the system which is made out just in case of early detection of any errors and capability of our system. I have used paper prototyping as my default prototype foundation as it is cheap and easily amendable. Also since I am working on Agile Methodology furthermore updates are possible throughout the process.

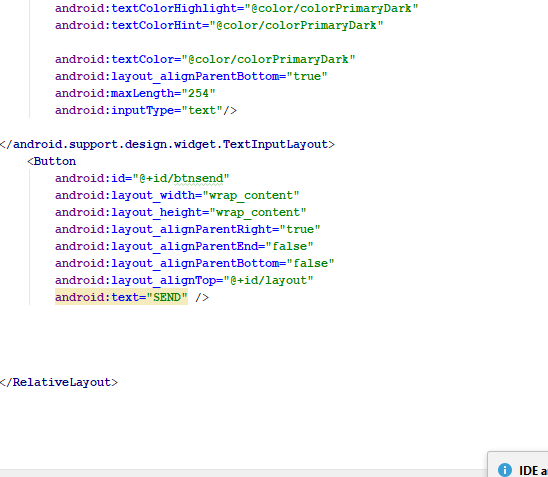
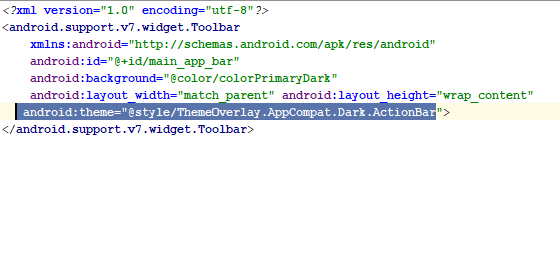


# Implementation

XML FILES of UMeChat android

Activity\_main.xml Activity\_main2.xml

Activity\_main3.xml

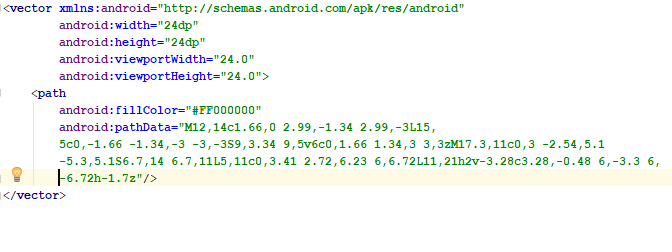
Activity\_test.xmlApp\_bar\_layout.xml

Device\_adapter\_view.xmlFragment\_chats.xml

Fragment\_contacts.xmlFragment\_groups.xml

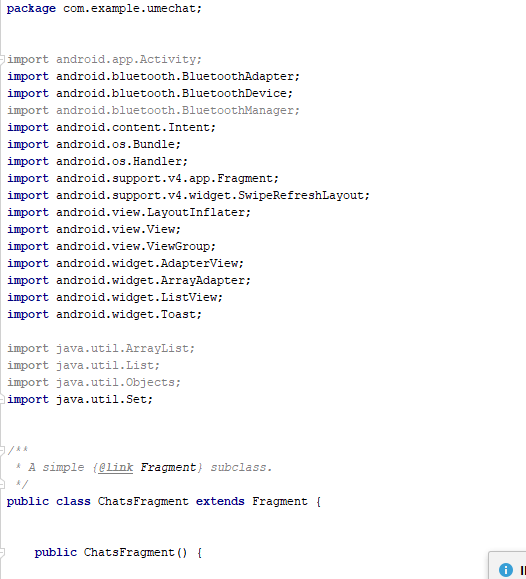
Drawabes Extra:

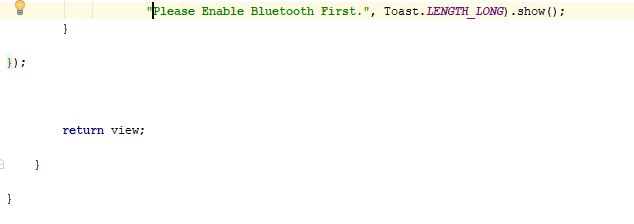
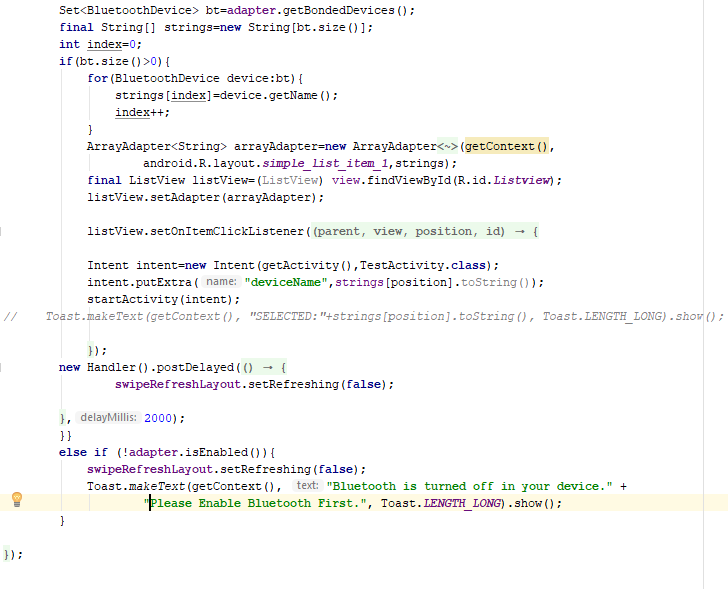
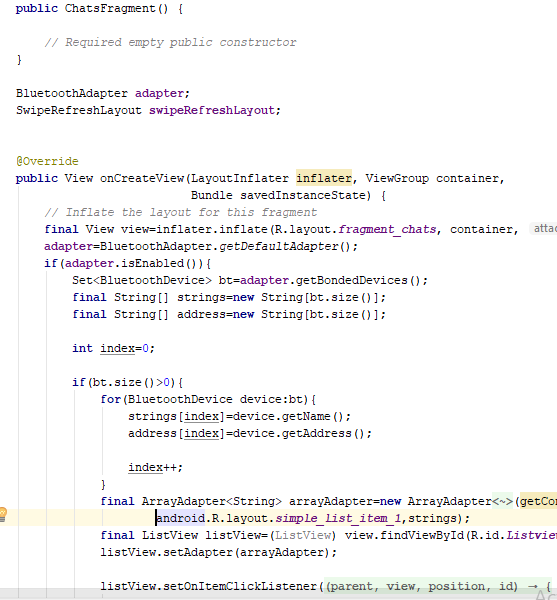
Ic\_launcher\_bckground.xmlic\_mic\_black\_24dp.xml

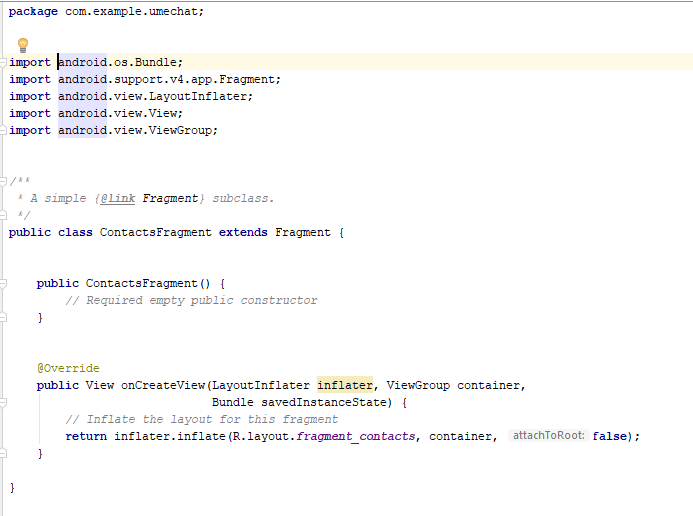


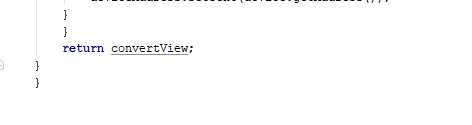
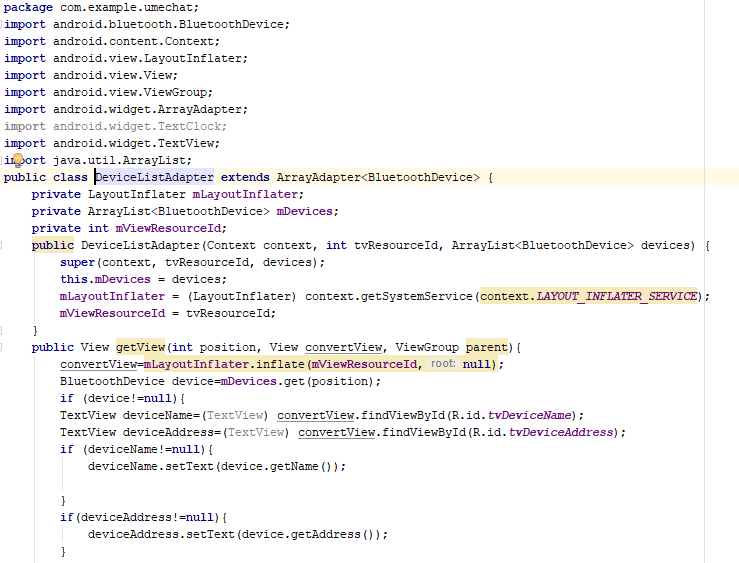
Class Coding:

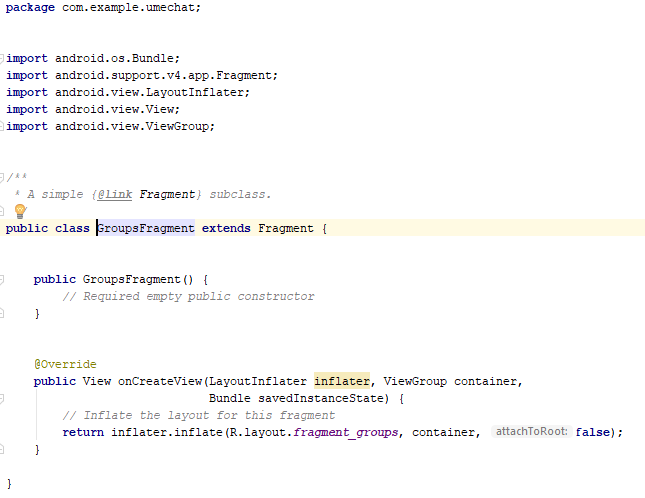
ChatsFragment.java

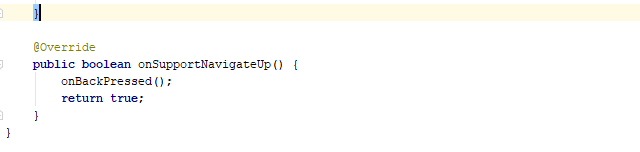
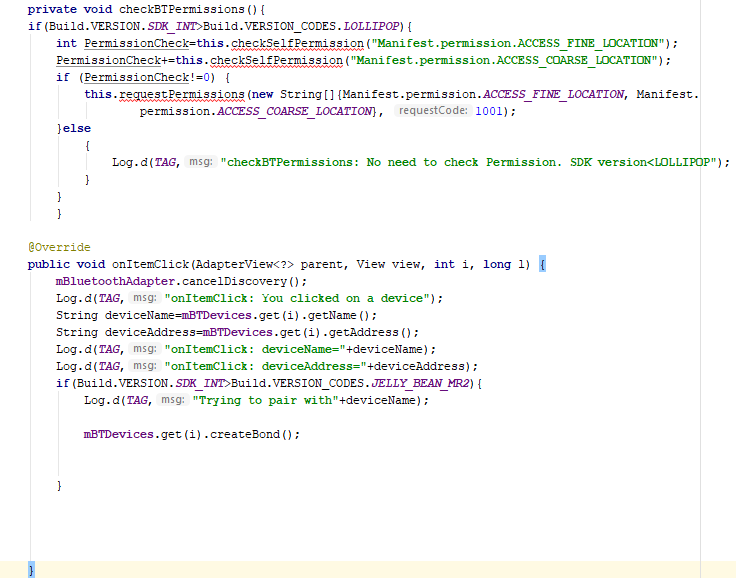
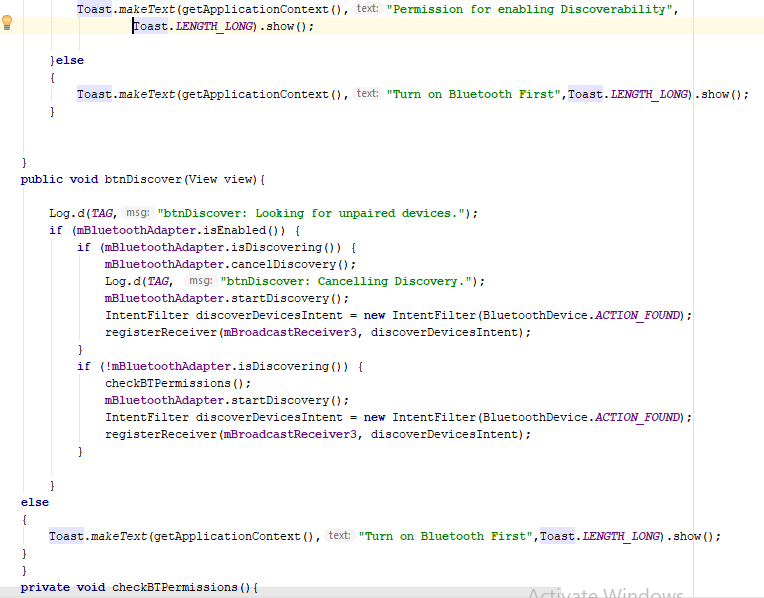
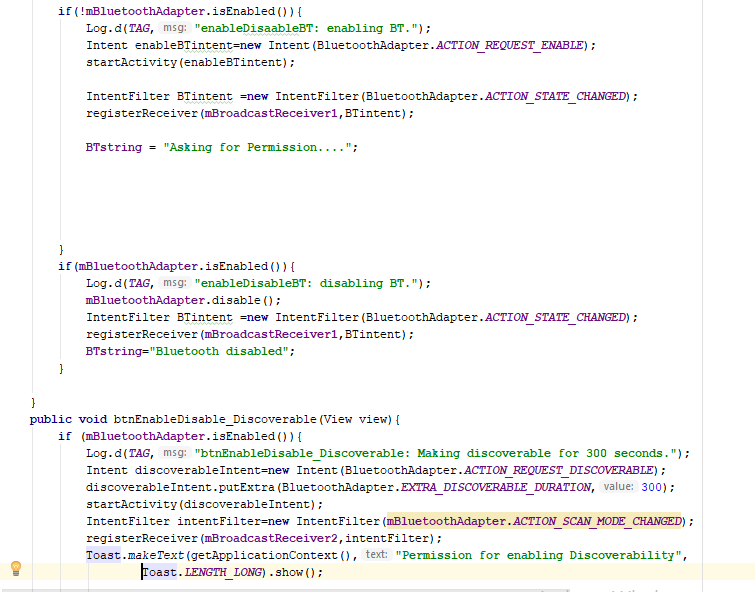
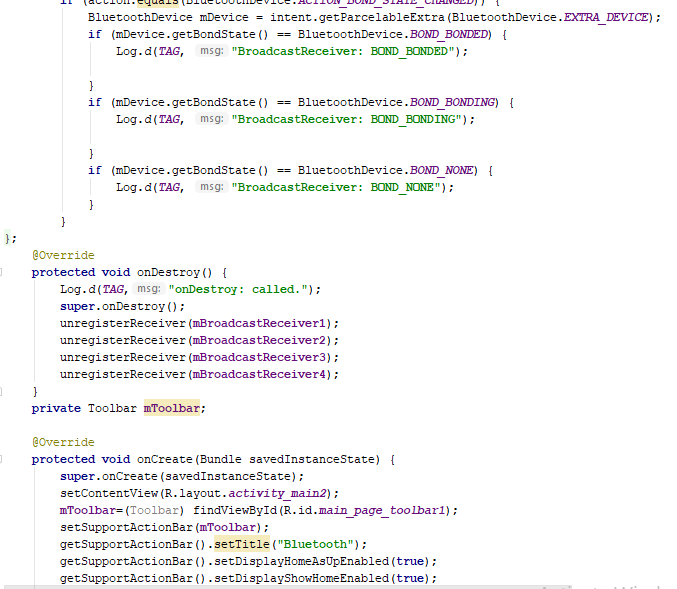
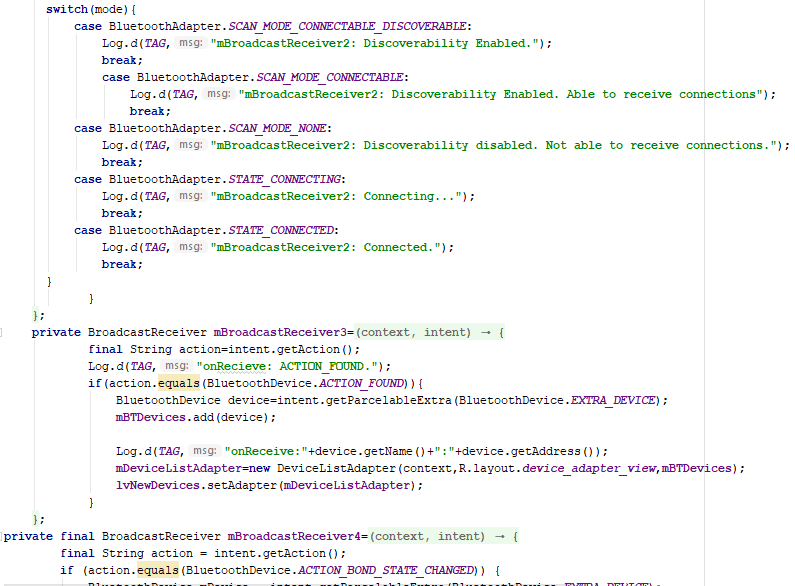
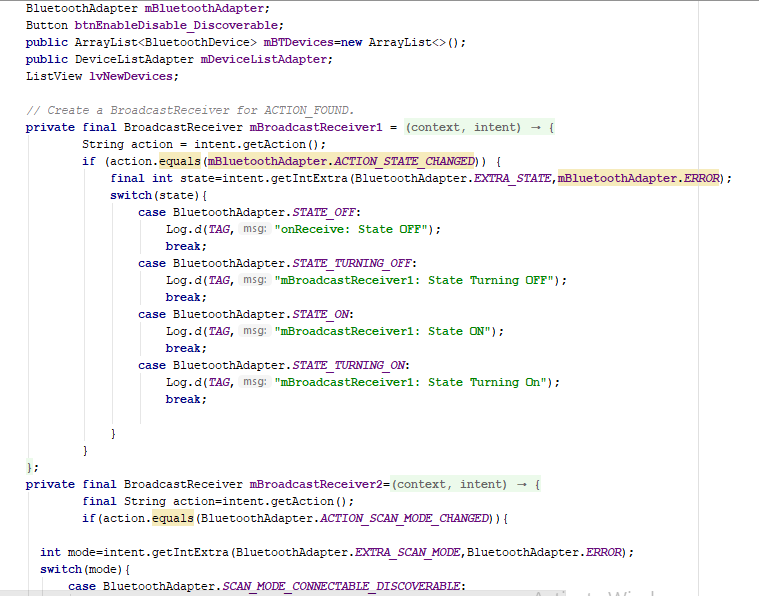
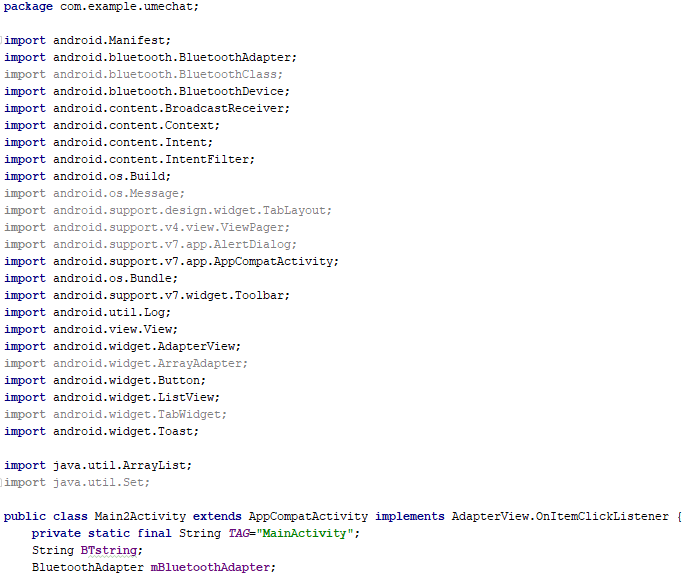


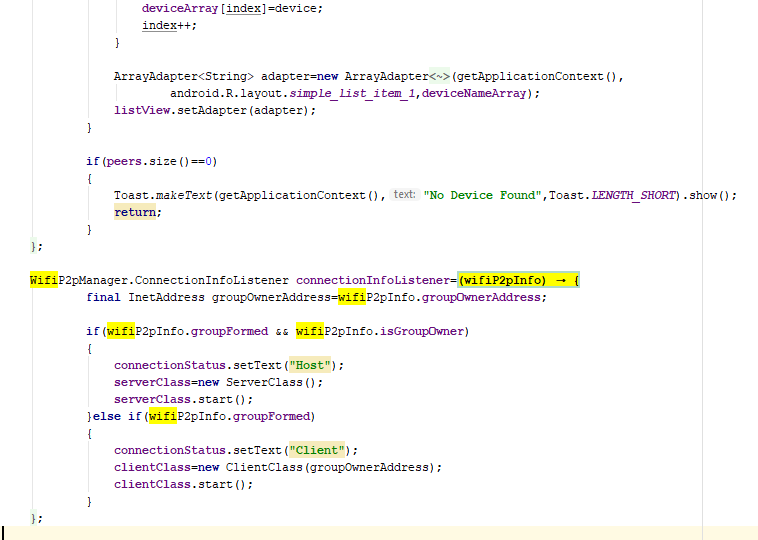
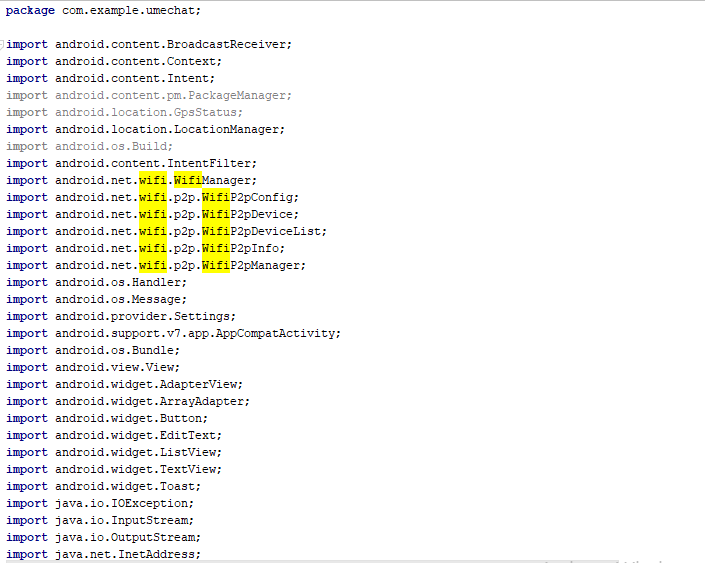


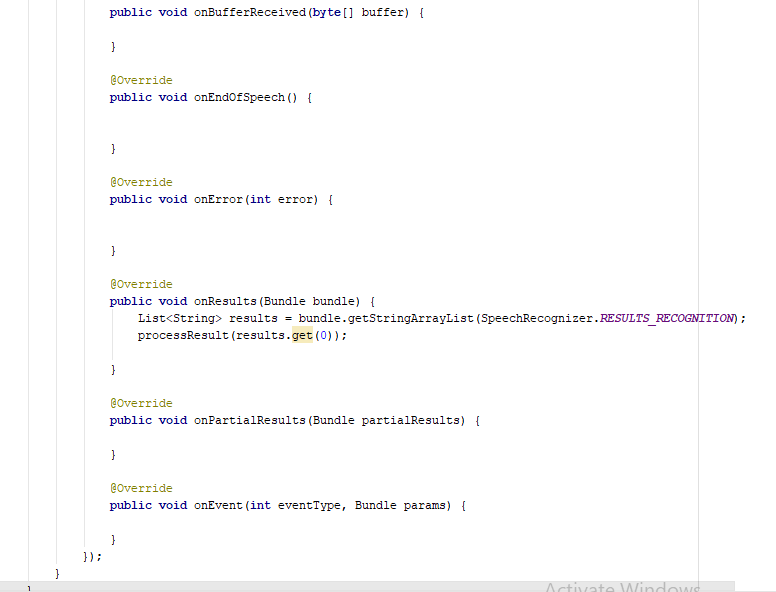
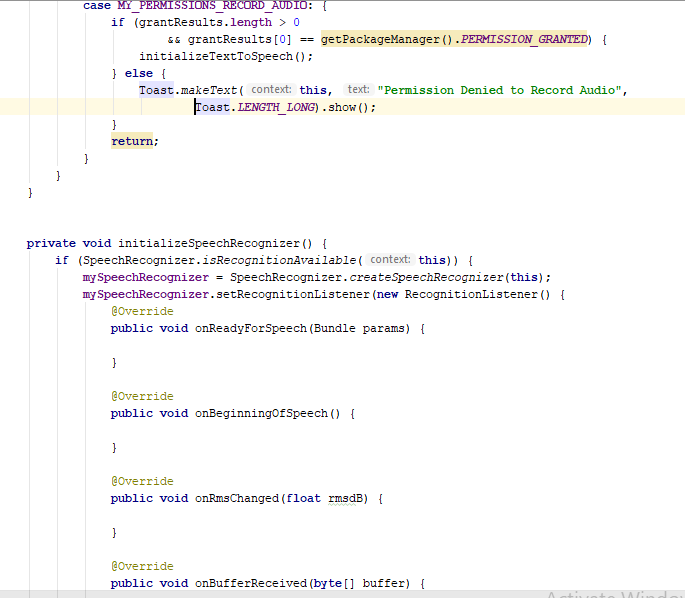
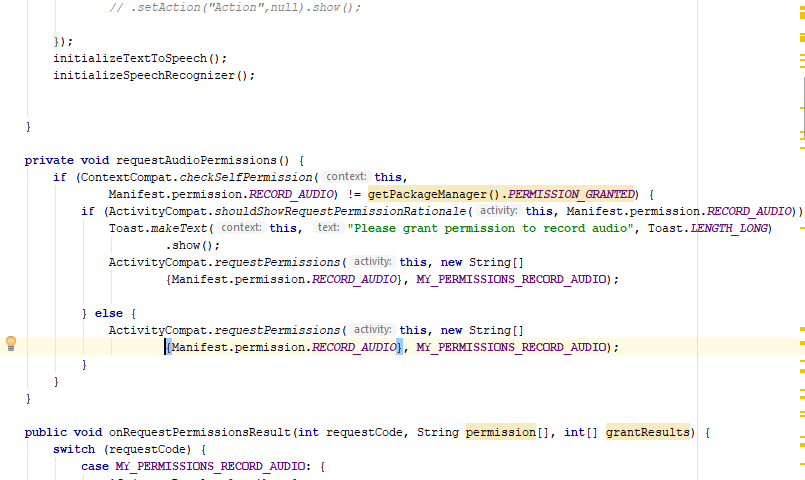
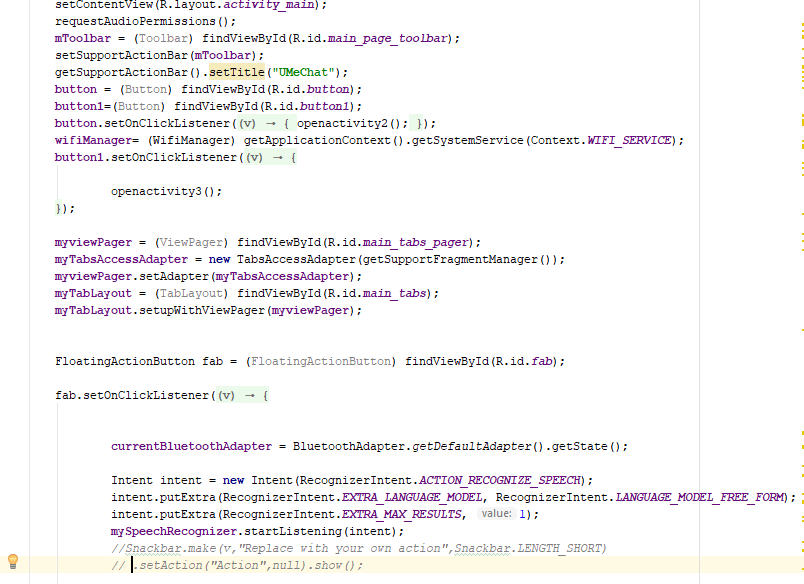
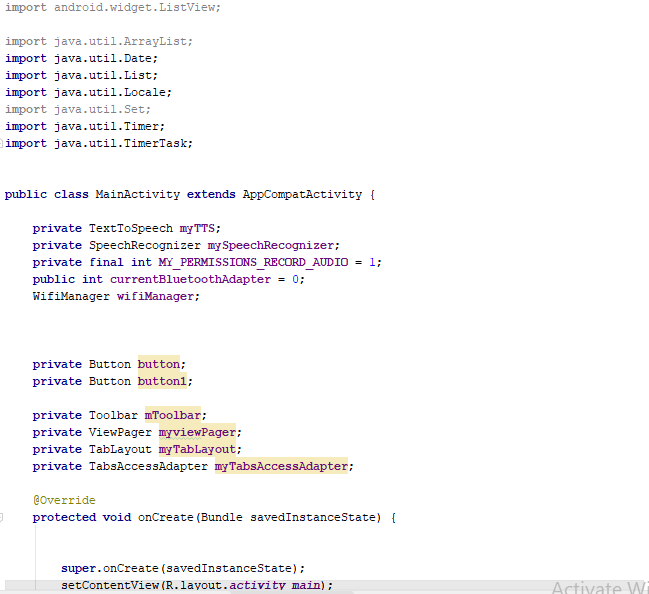
ContactsFragment.java

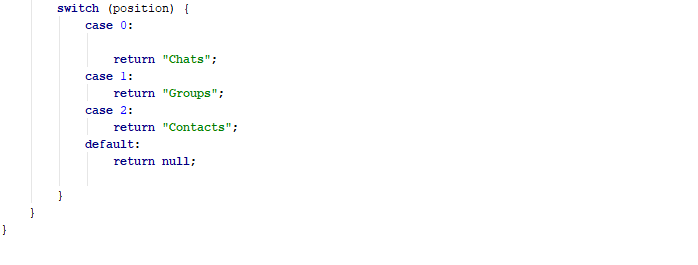
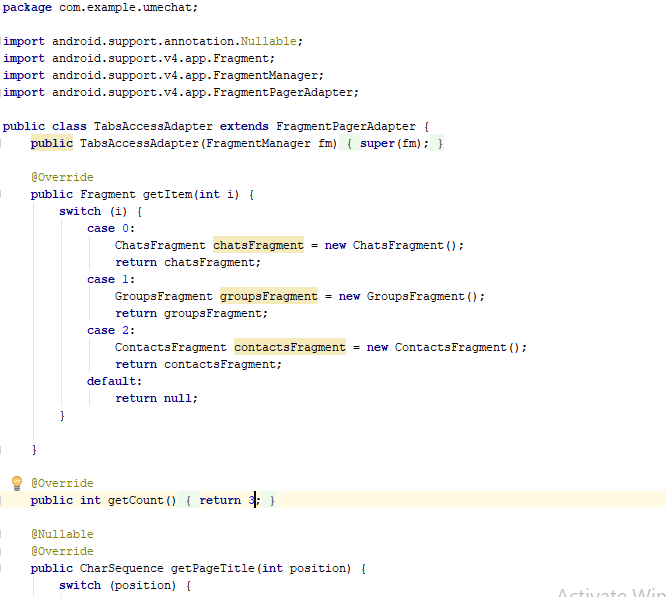
DeviceListAdapter.java

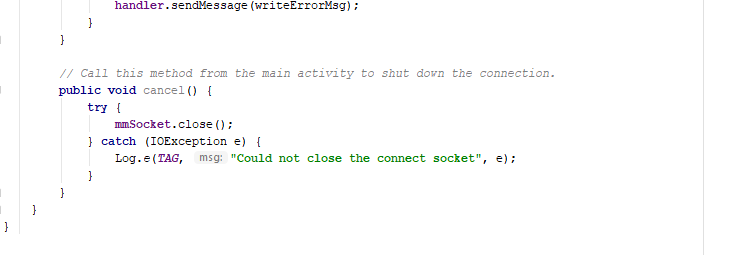
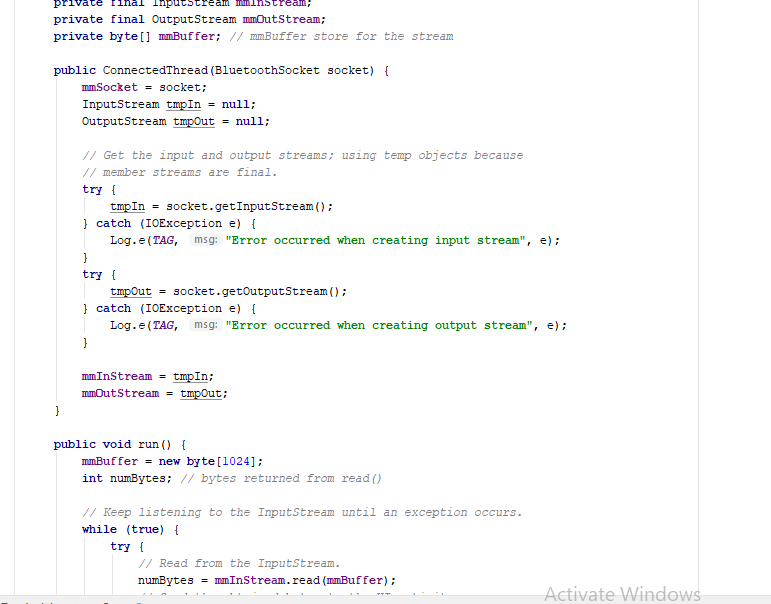
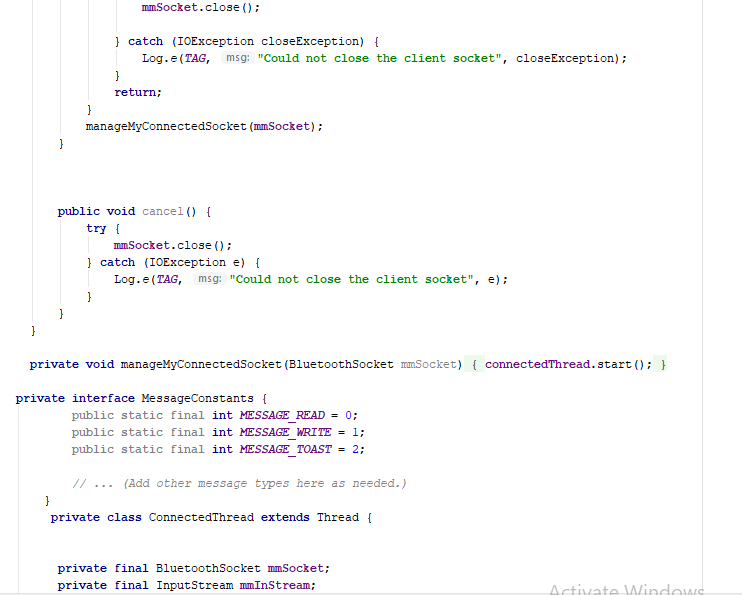
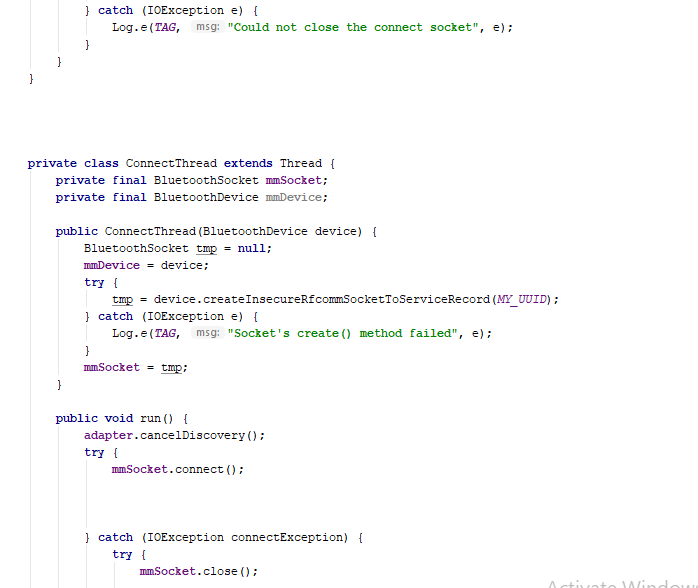
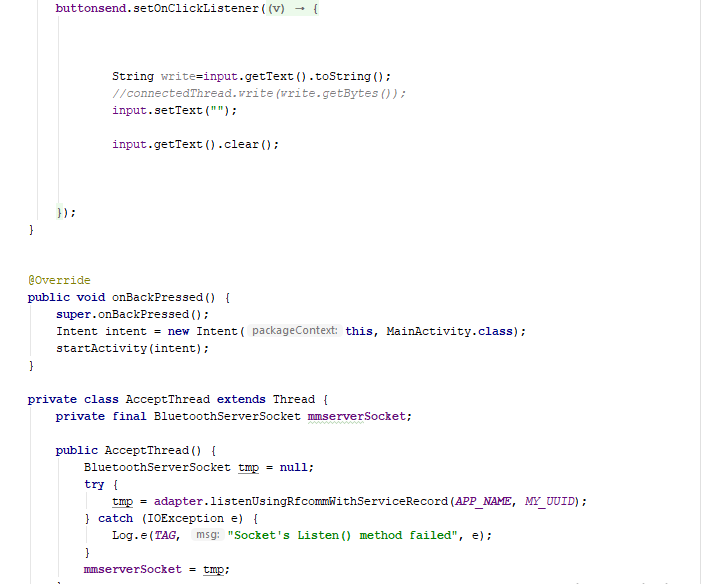
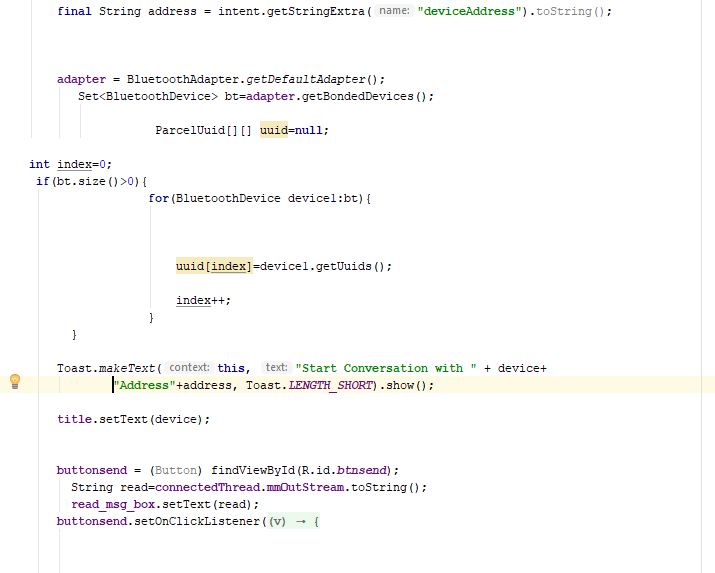
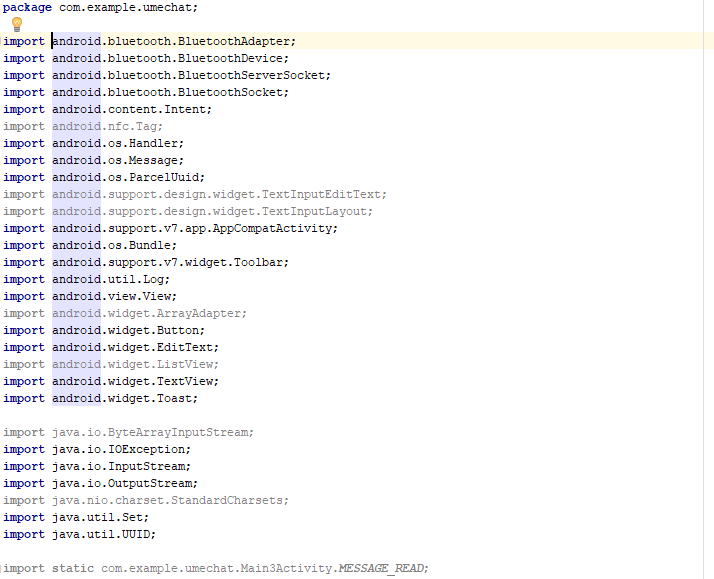
GroupsFragment.java

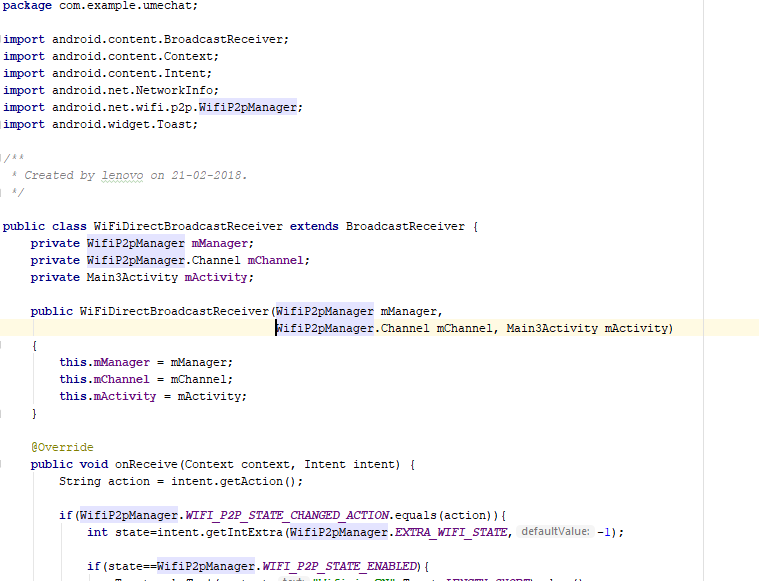
Main2Activity.java

Main3Activity.java 

MainActivity.java 

TabsAccessAdapter.java

TestActivity.java

WiFiDirectBroadcastReceiver.java

# Testing

It is the simply process of conduction where the codes of system is validated and debugged. It is simply categorized into two types of testing and they are:

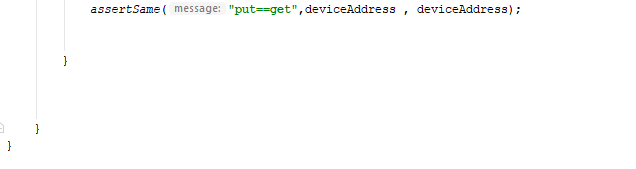
-White-box Testing

-Black-box Testing

White-box Testing simply includes the types of testing where the codes of test process conduct are visible i.e. Testing process is detectable and transparent.

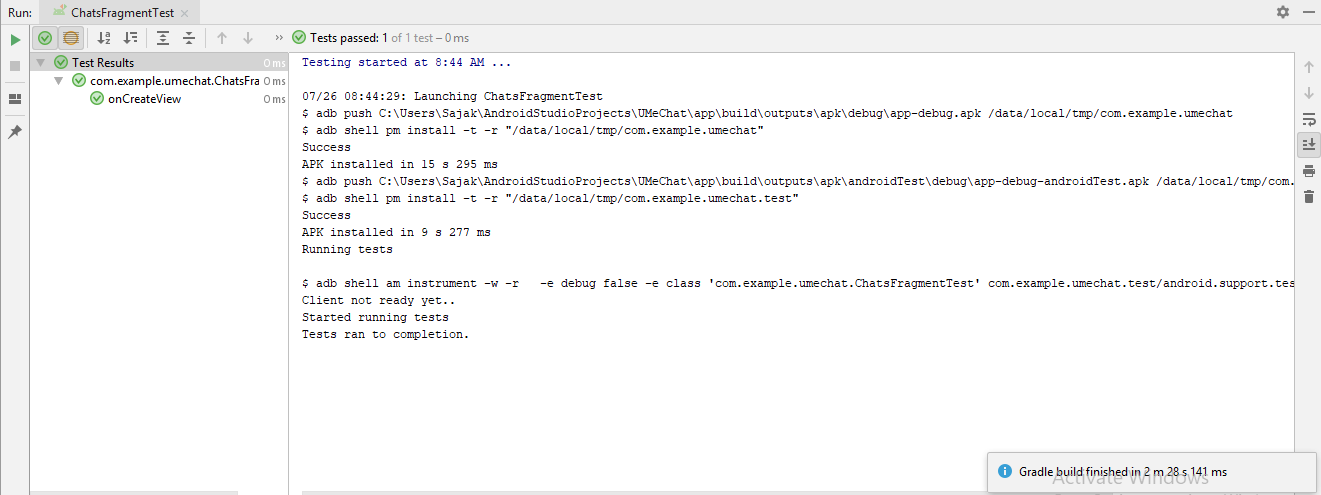
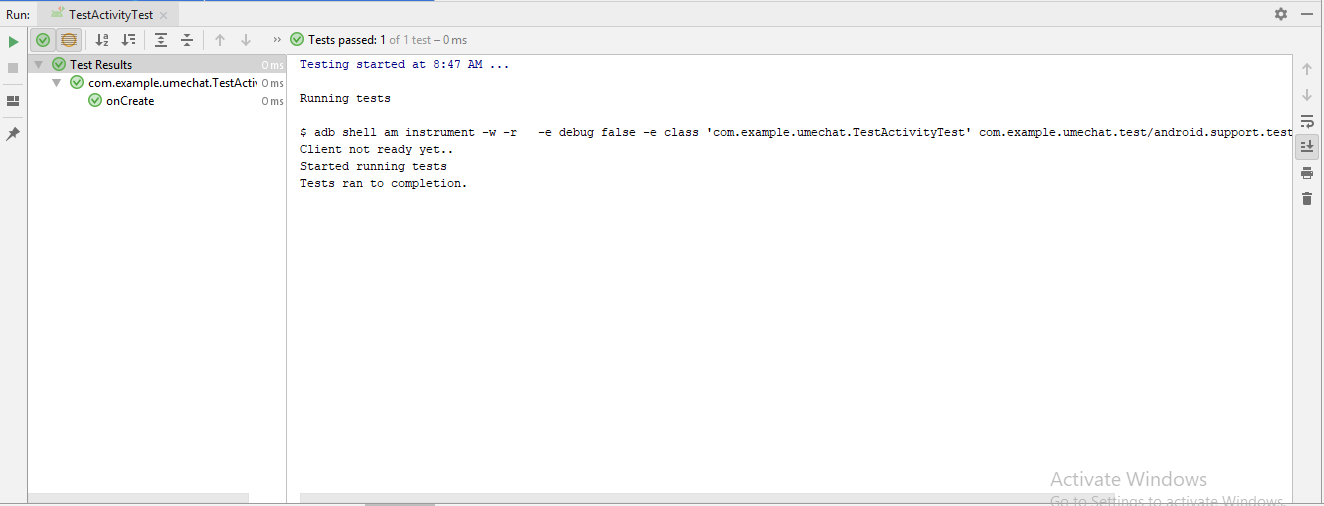
Black-box on the other hand includes those types of testing where the codes of test process conduct are not visible i.e. Testing process is un-detectable and opaque.

For White-Box Testing I have used Junit java testing in android studio because it is simple and easy to conduct. These are some of the testing processes that are into this category.

White-Box Testing

Test-Result Profiler

Here we run the test by simply right-clicking on the test and execute the test and profiler dialogue of test result is shown as shown in the below diagrams.

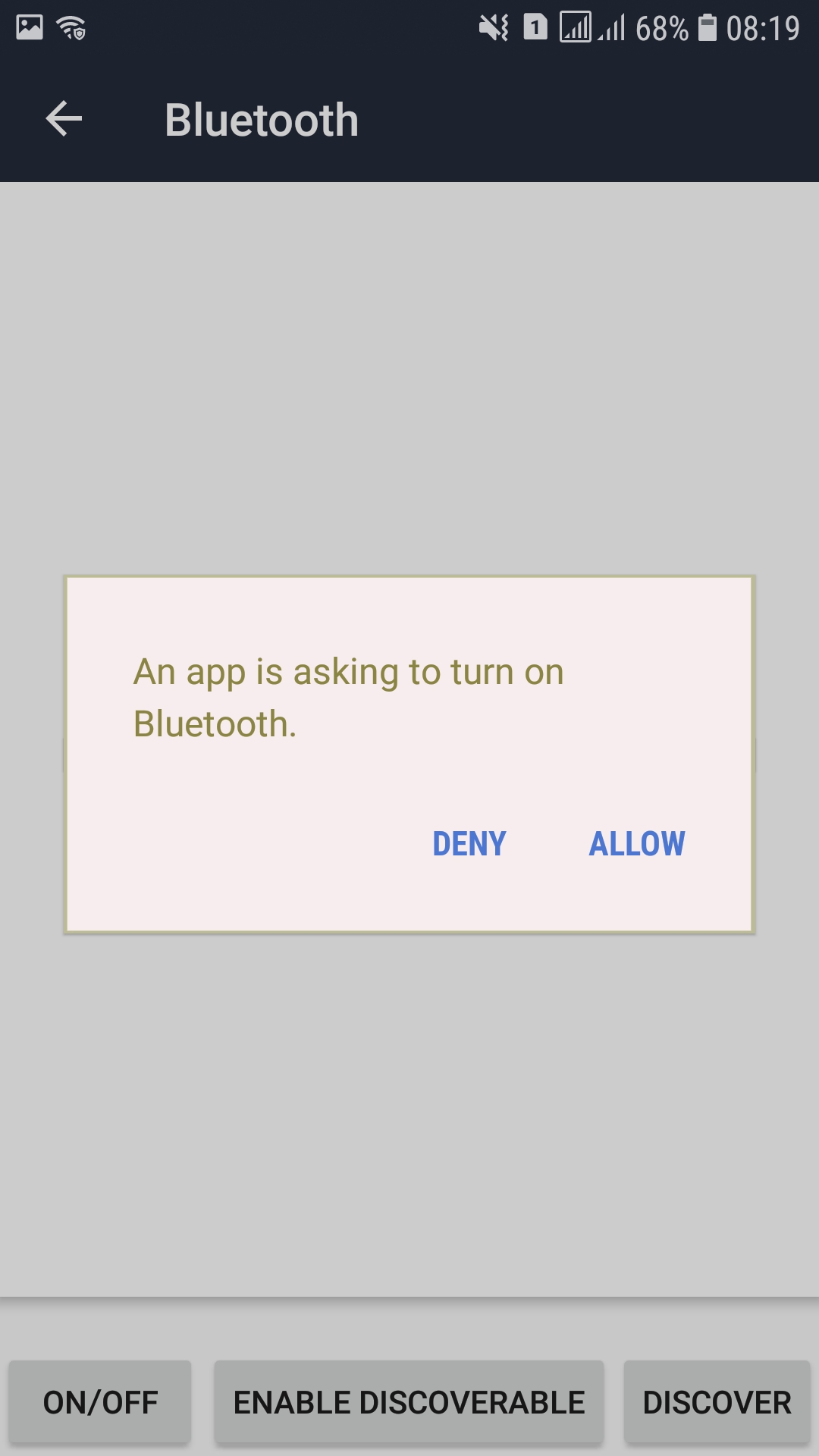
ChatsFragmentTestMain2ActivityTestTestActivityTest

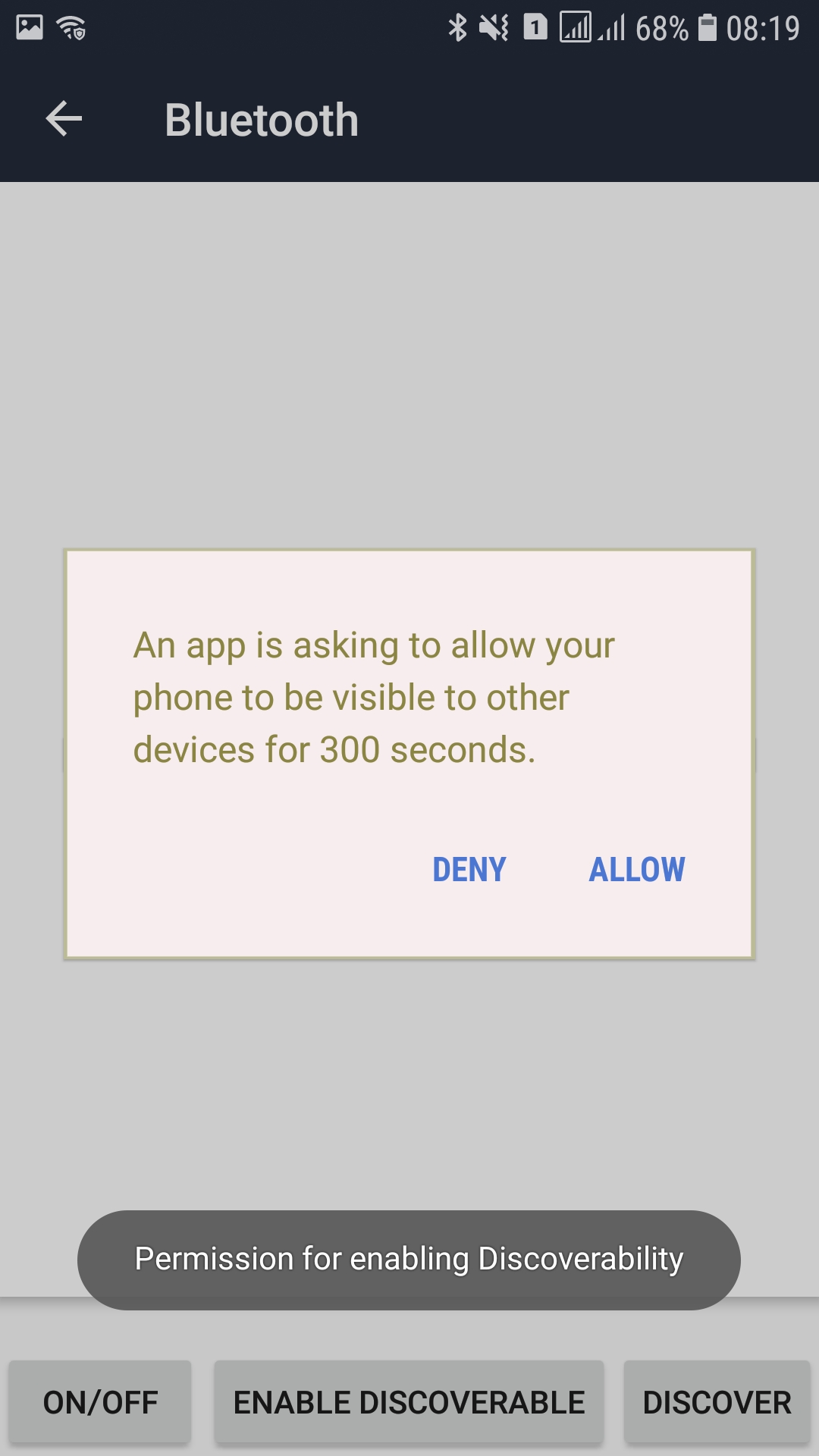
For Black-Box testing I have simply run the application in my device and screenshot the performance and these are some test process in this category.

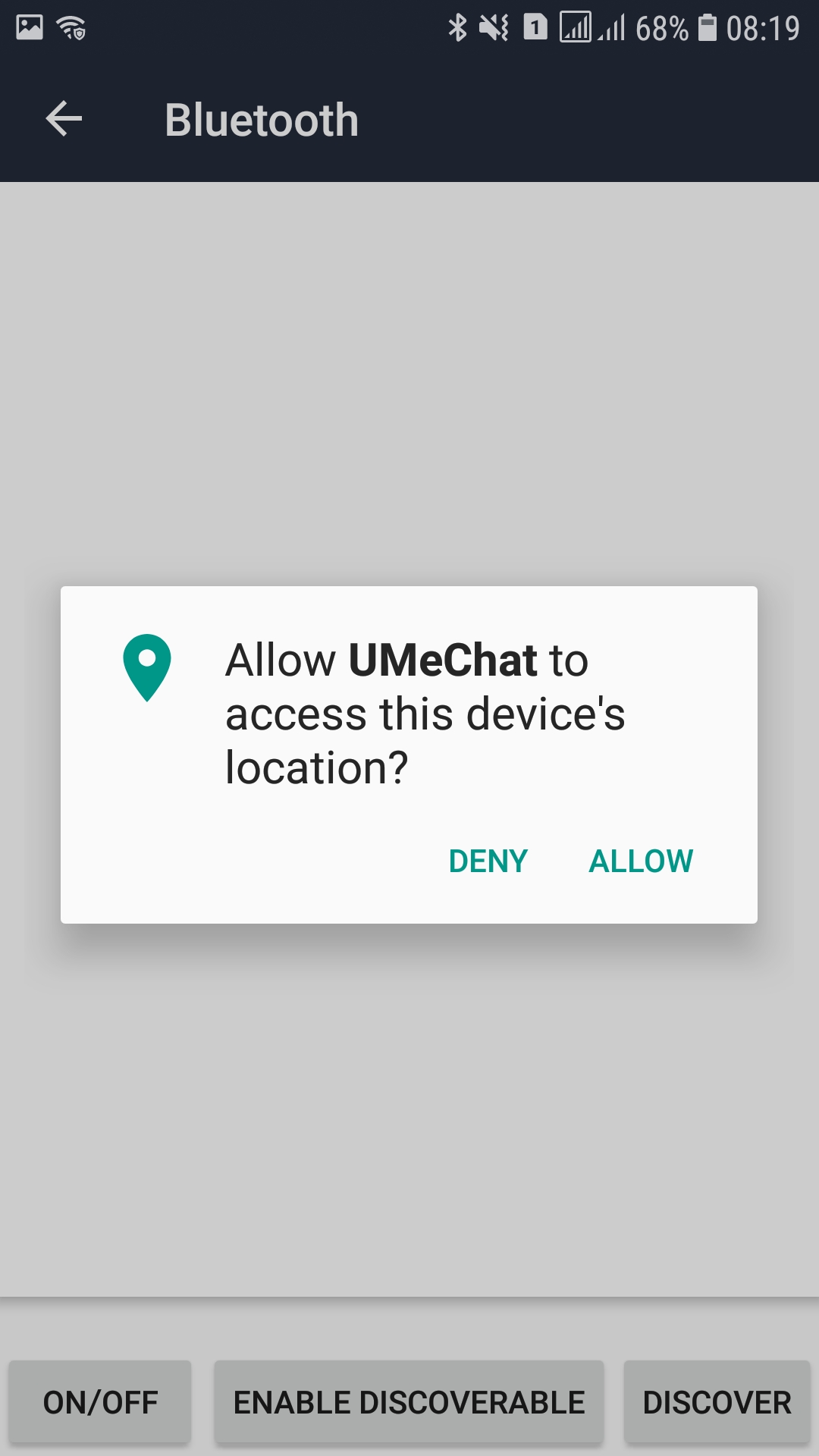
Black-Box Testing:

Application Start-Phase

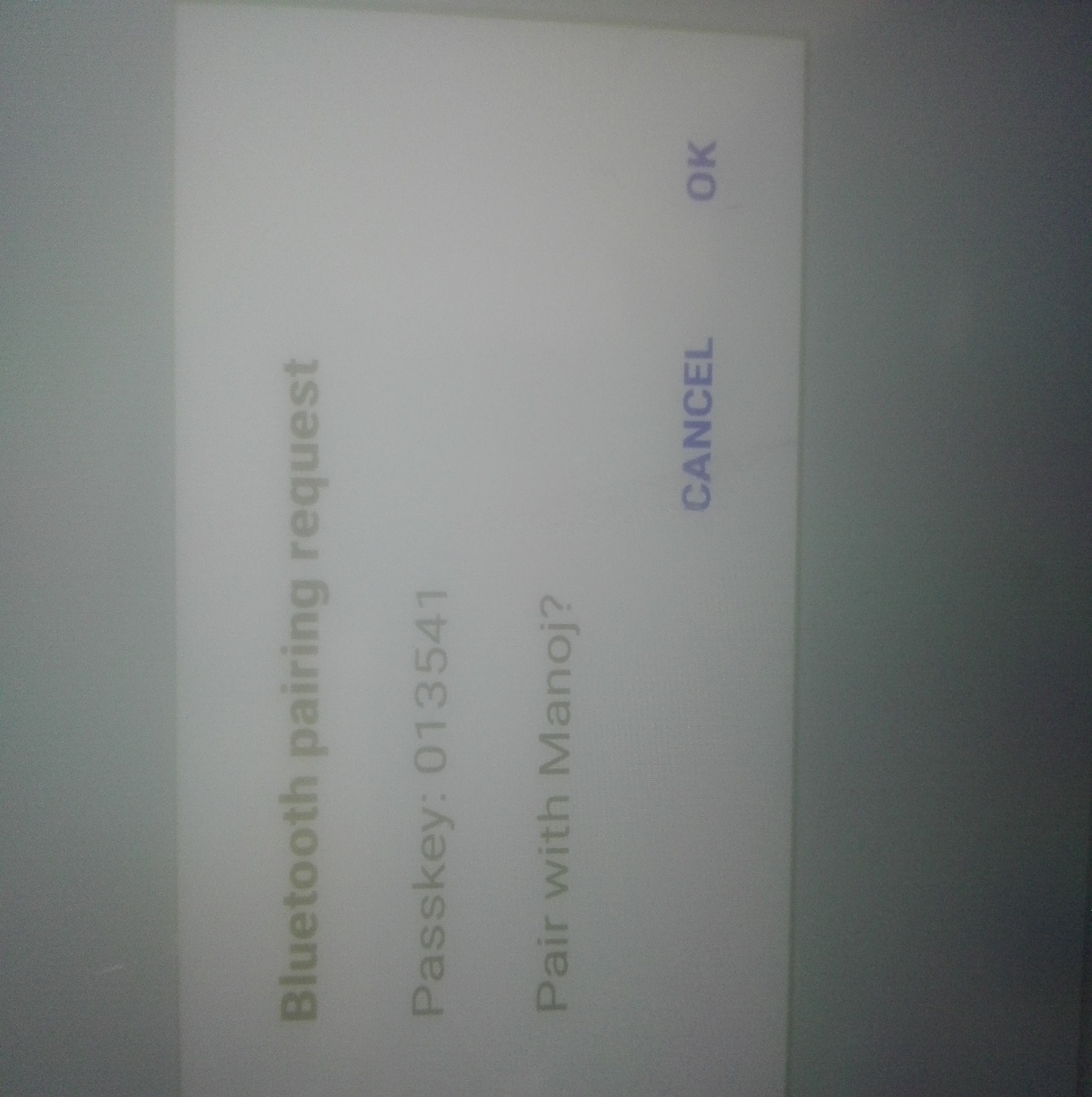


Permission for Bluetooth

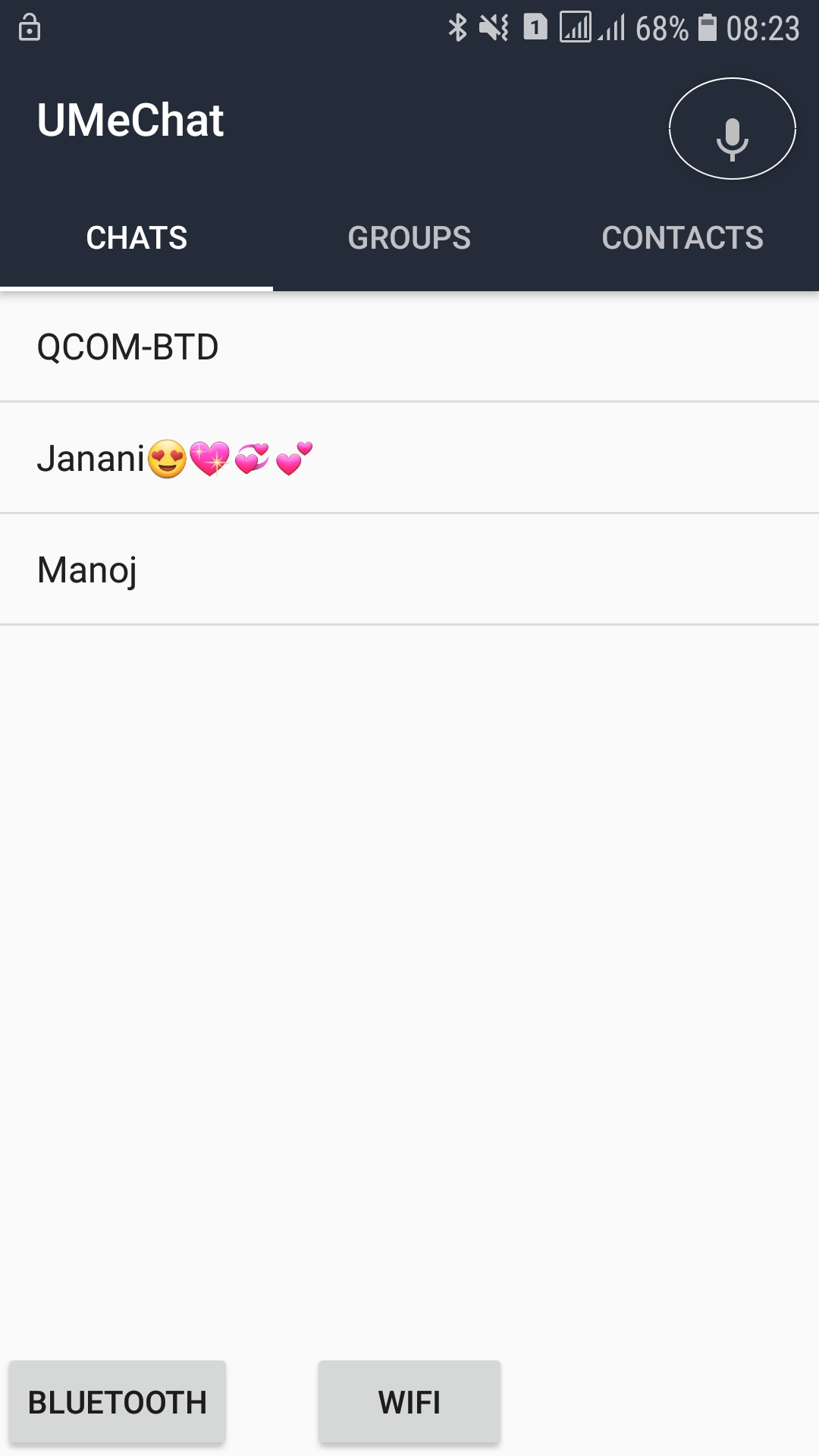
Permission for Bluetooth Discovery

Permission for device location

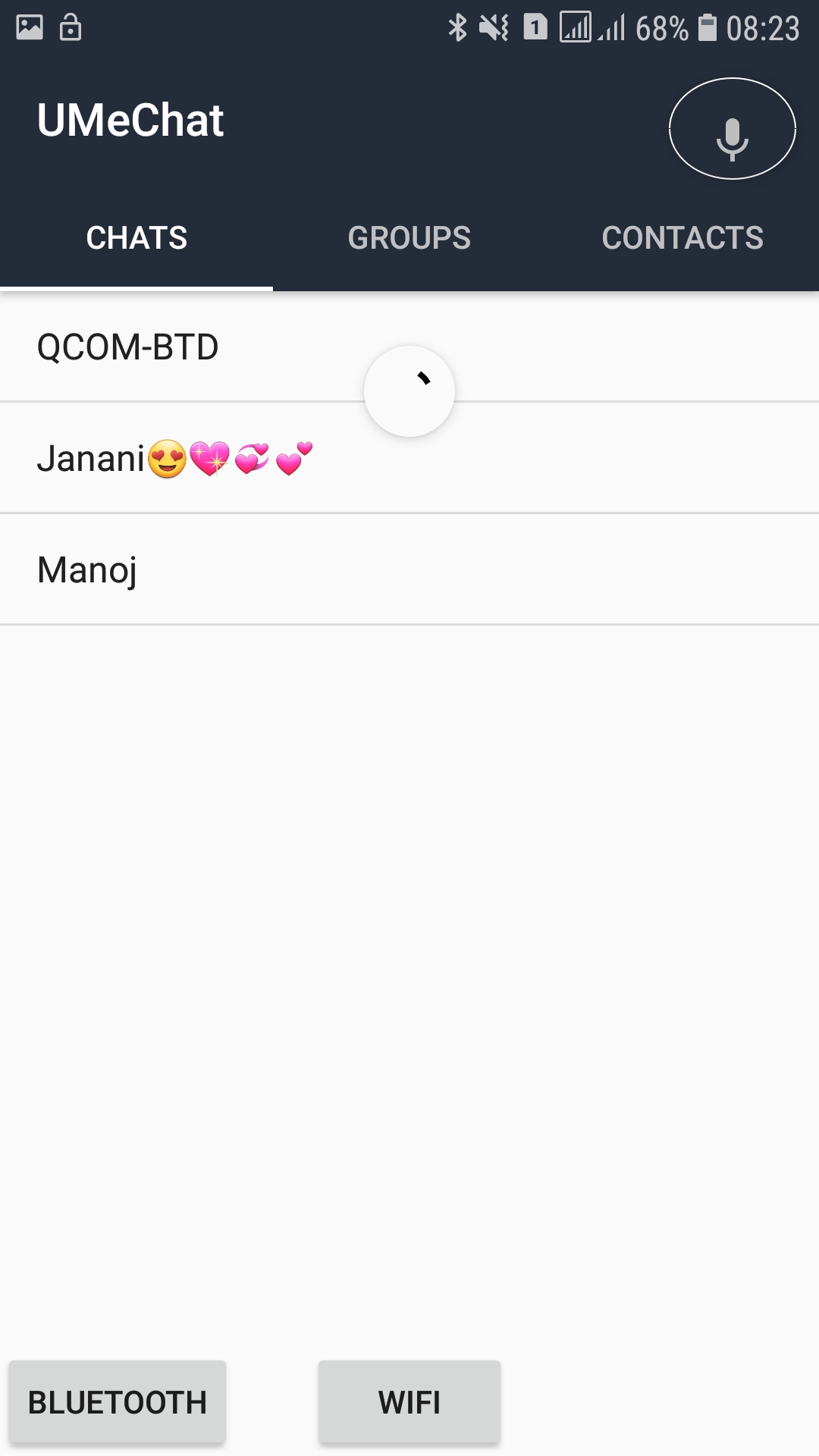
List of Available Devices

Pairing with Available Device

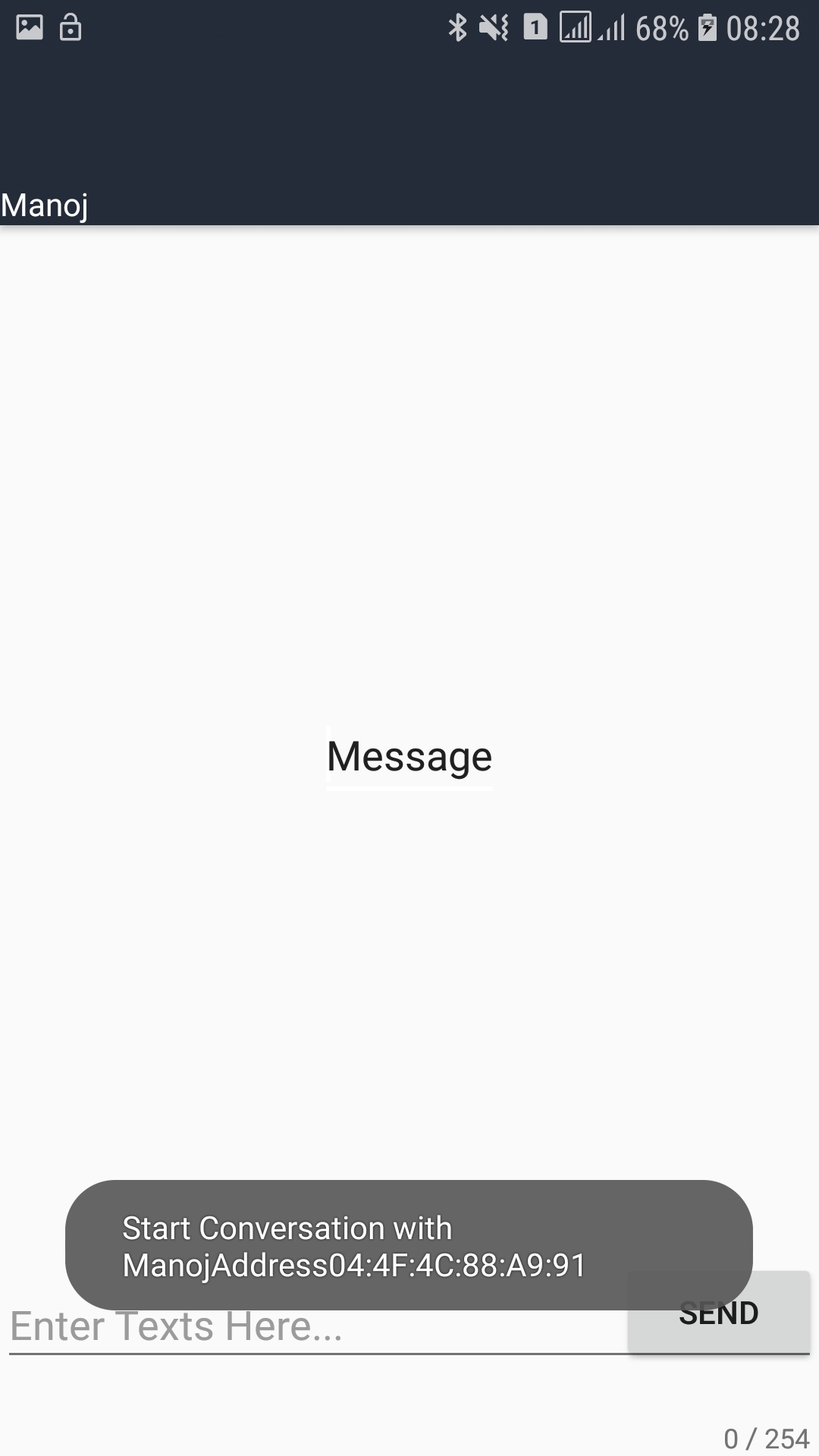
List of Paired Devices



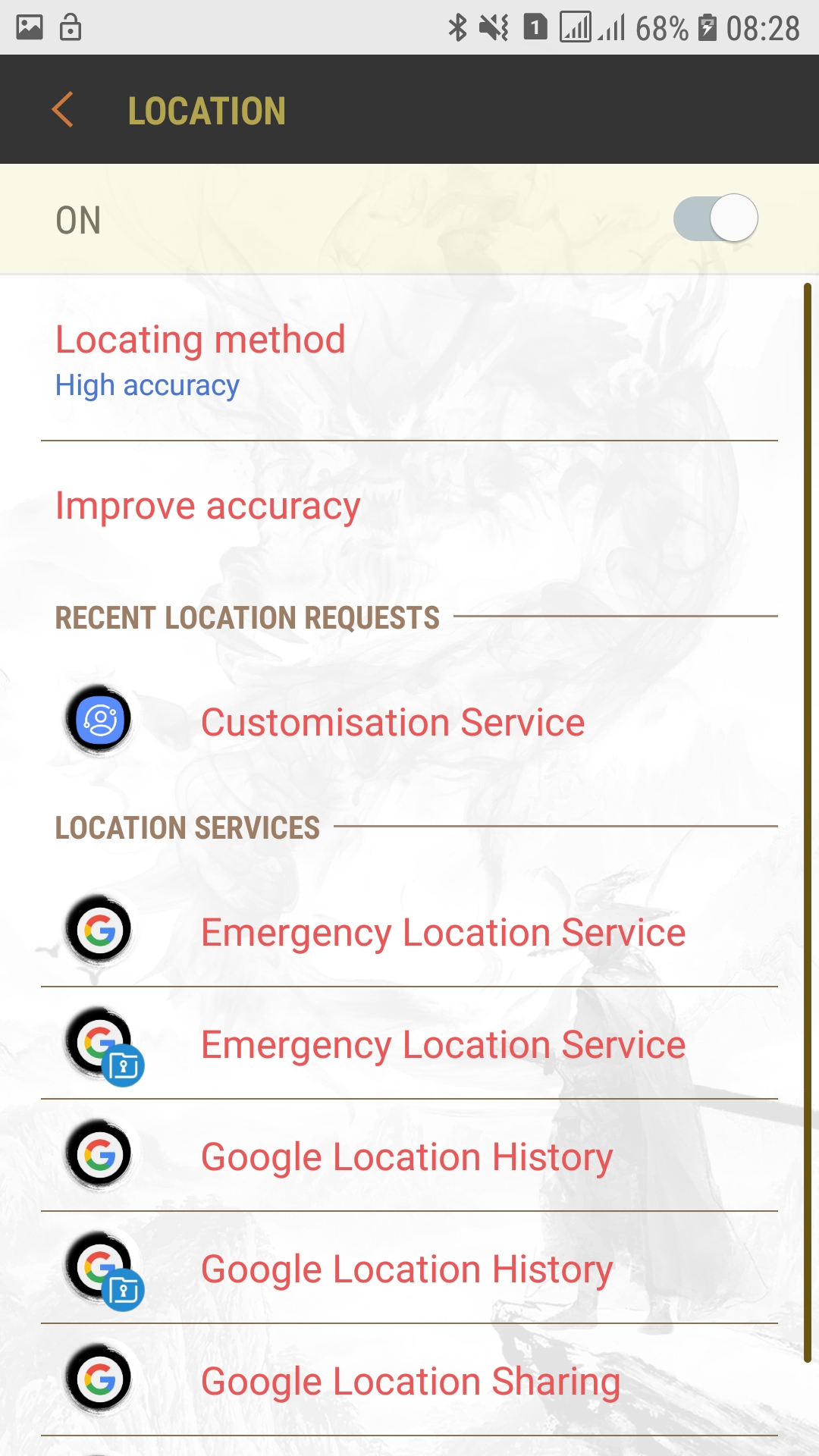
Refreshing List

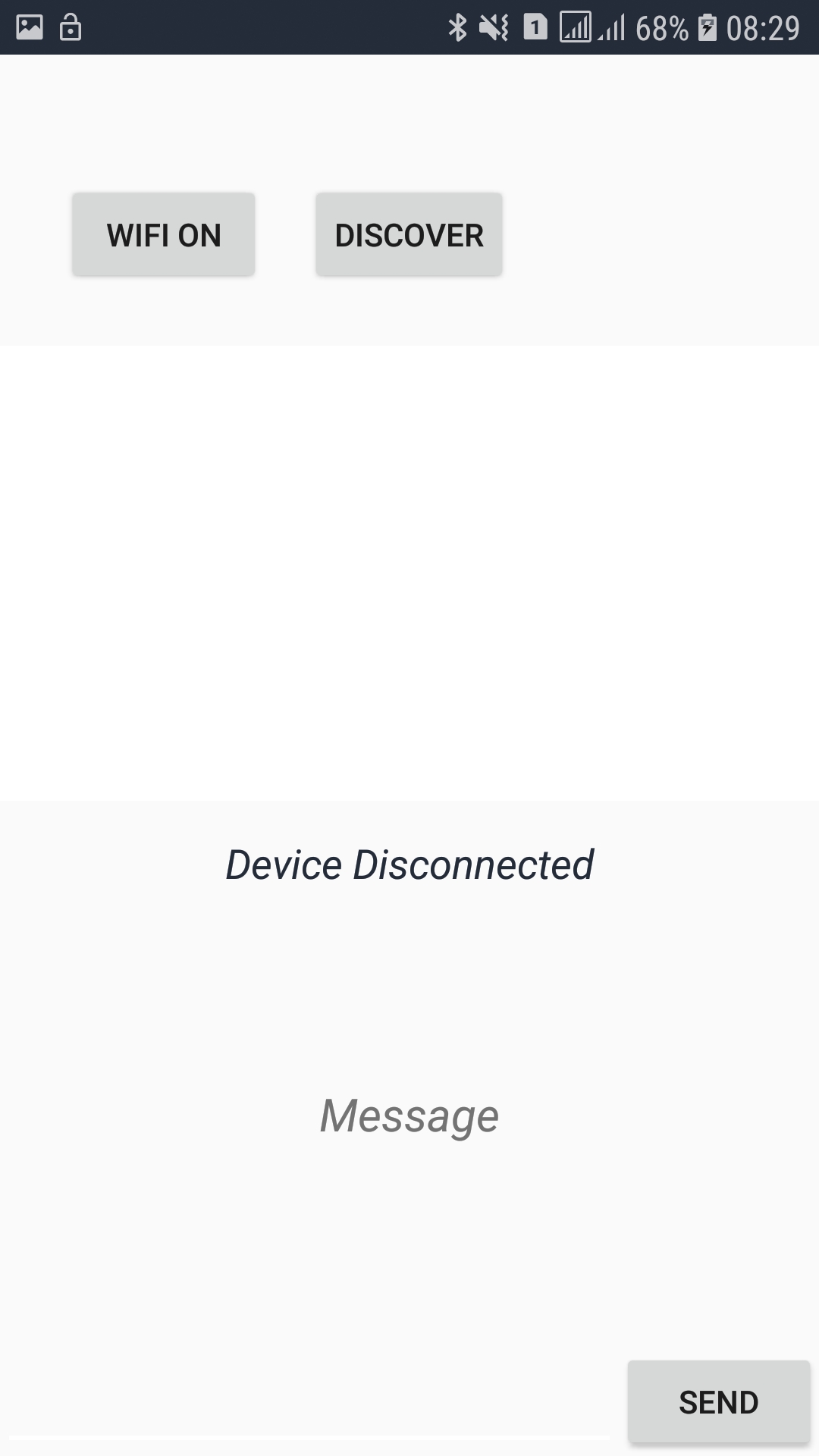


New Chat Activity

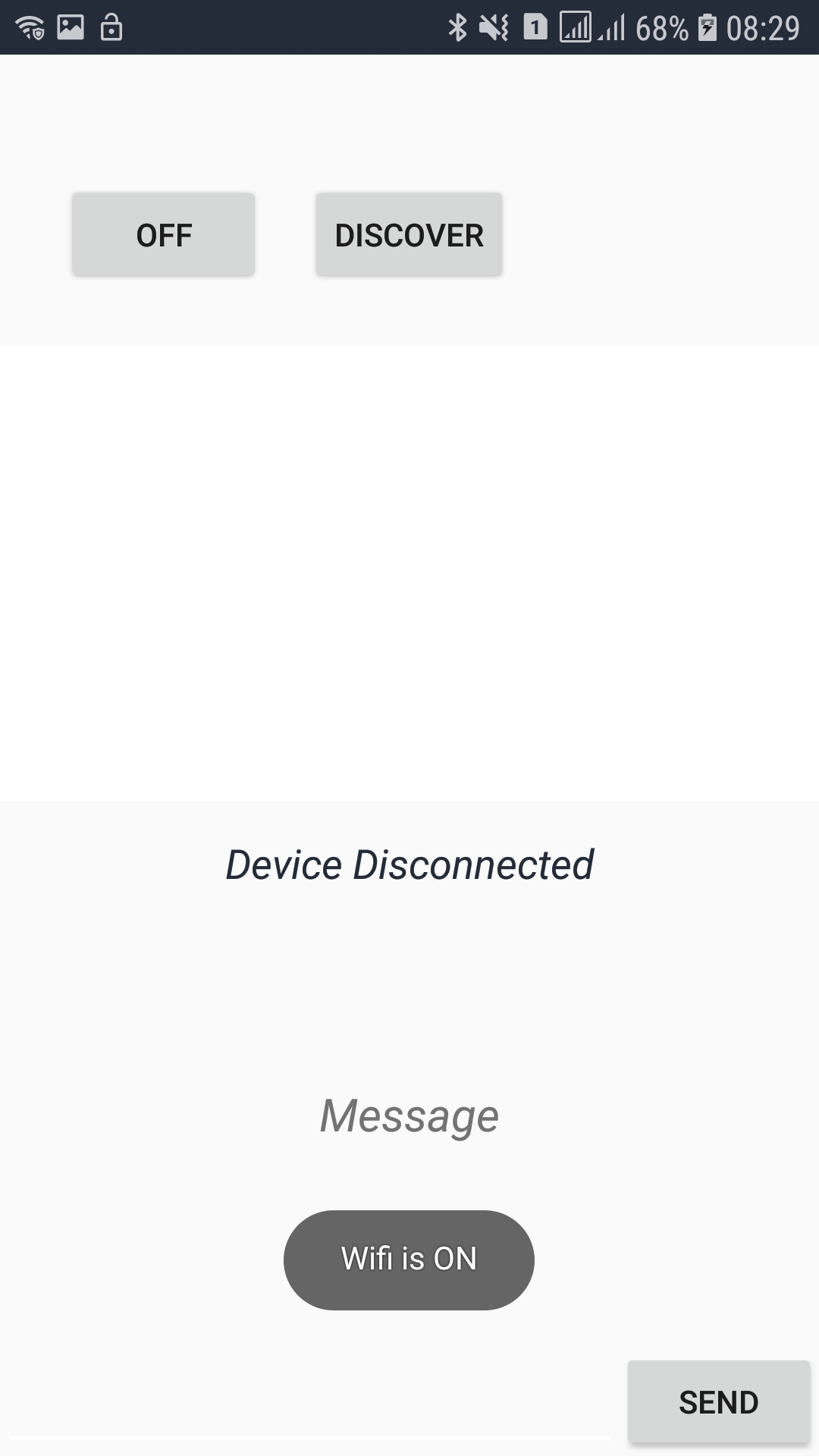


Location ON

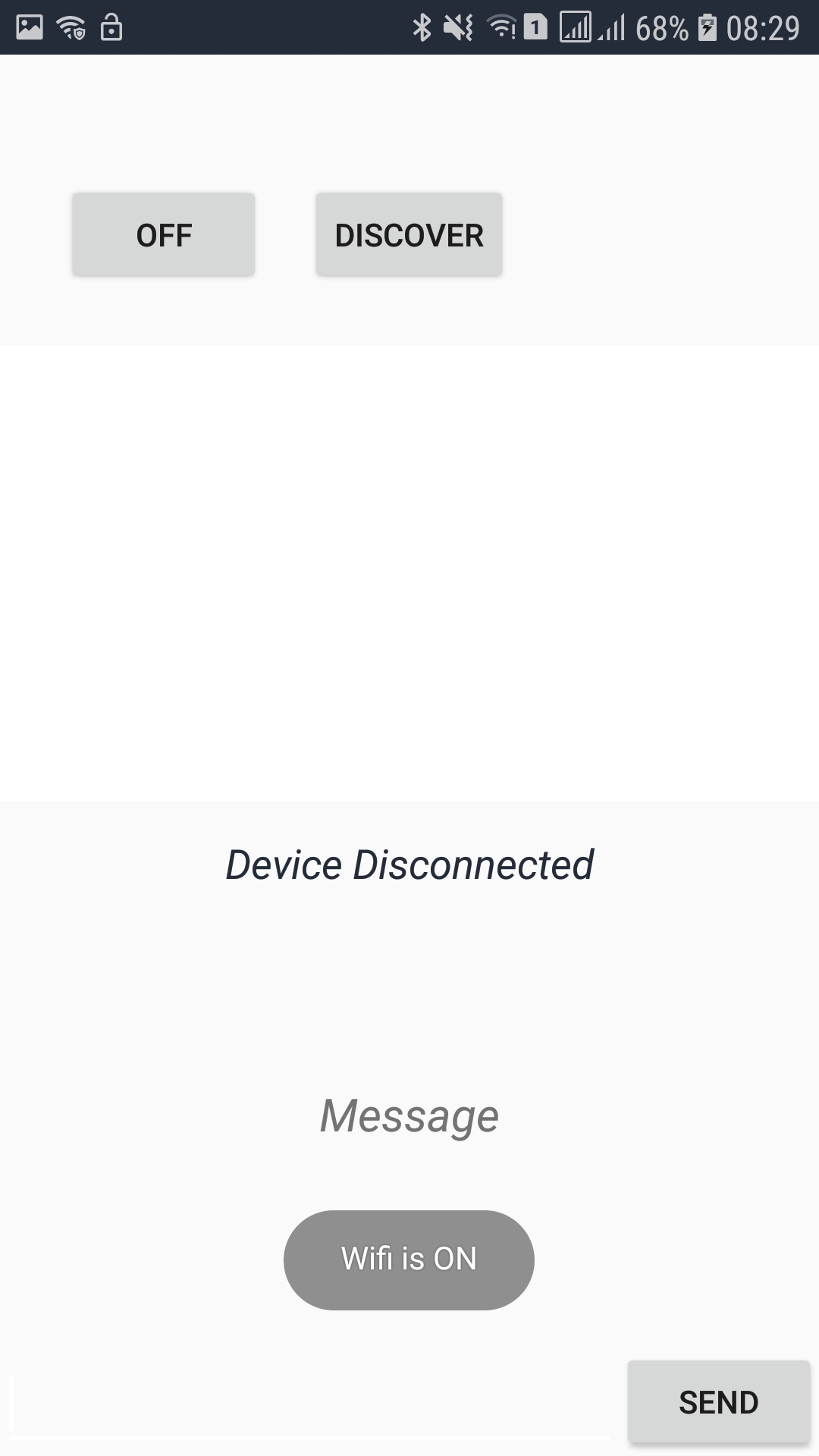


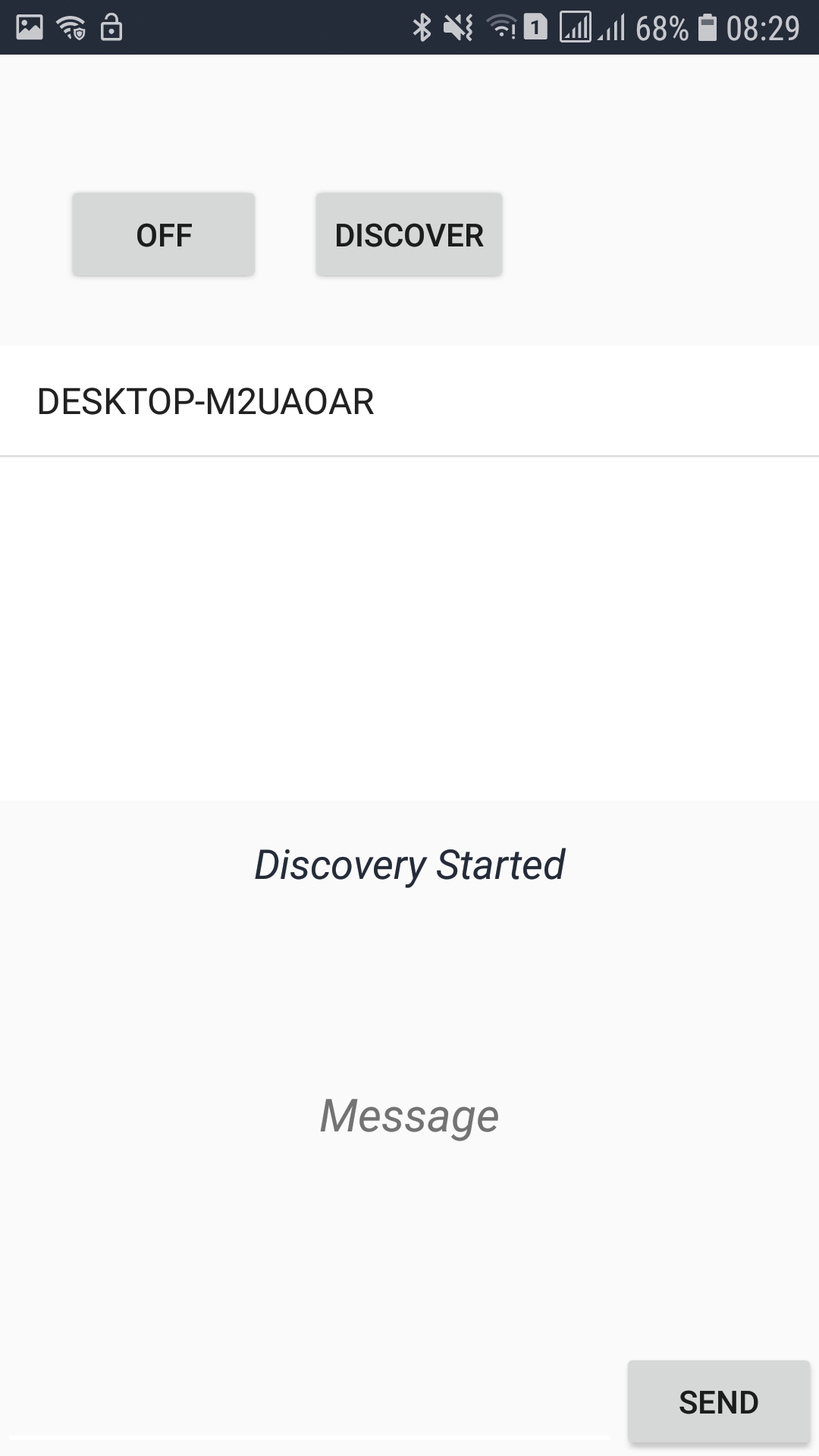
Wireless Chat Start Phase 

Enable WiFi



Enable WiFi



Discover Available Device

Connect Devices





# Other Project issues

Limitation of my Project

The Limitation of my project are illustrated below:

* This system does not covers a wide range area making long ranged communication not possible without prior equipment’s.
* This system is not on practice making it vulnerable on demands.
* This system might bring conflict in ones day to day life.
* This system demotivates people from social collaboration verbally.
* This system requires a large amount of time and also it has not met many more features to be included because of poor knowledge of Java.

Future Work

The current application is still incomplete as it does not meet all the features that I thought would be in this application. Weak foundation seems to be the main reason for this project to not meet the required criteria. Thus the future work is simply to recreate what is left and build this application accordingly to make it a perfect deal application. Also this project had some bugs and errors which are just neglected as syntax which was checked repeatedly but the system was just to contrast for me being new to Java I was devastated while it couldn’t meet all the condition required in this project but I have included most of the features I could possibly add up. So my future plans are also to clear out these error and make it efficient and effective application.

Risk Management

Risk Management is the way of optimizing risk factors by recognizing, solving and preventing such risks that may occur in System lifecycle during project implementation. By calculating the impact of the risk which is the product of likelihood and consequence that might occur we can determine the risk factor rate in a system and take cautious risk solving methodologies to prevent furthermore difficulties in system development.

I.e. Impact=Likelihood X Consequences

The scale of rating is ranged from 1-3 and rated upon approaches.

|  |  |
| --- | --- |
| **Likelihood** | **Value** |
| Low | 1 |
| Medium | 2 |
| High | 3 |

Fig: Risk Like hood values

The scale of consequence is ranged from 1-5 and rated upon probability of consequence occurrences.

|  |  |
| --- | --- |
| **Consequence** | **Value** |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very high | 5 |

Fig: Risk consequence values

Possible Risks

1. Device malfunction
2. Natural Disaster
3. Malware Detection/Spyware
4. Error Detection
5. Irresponsive UI
6. Data Theft
7. Poor estimation

Now by illustrating the possible risk factors we have tabulated risk Management Table below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S.N. | Risks | Likelihood | Consequence | Impact | Action |
| 1 | Device malfunction | 2 | 4 | 8 | Device Maintenance necessary. |
| 2 | Natural Disaster | 1 | 4 | 4 | Backup of data or files and proper handling. |
| 3 | Malware Detection/Spyware | 2 | 4 | 8 | Antimalware /Antivirus software’s to be used. |
| 4 | Error Detection | 2 | 5 | 10 | Approach of developing must be carried out effectively. |
| 5 | Irresponsive UI | 2 | 4 | 8 | Pre- testing of system in different devices. |
| 6 | Data Theft | 3 | 4 | 12 | Data Security must be made high and pairing password must be difficult. |
| 7 | Poor estimation | 3 | 4 | 12 | Estimate proper time and work effort with proper milestone necessarily. |

Configuration Management

It is the management process which helps to review all system and make certain that any changes in any one system doesn’t adversely affect others. It helps to re-install pre-version if necessary. This structural representation keeps all the records of changes that occurs throughout the system. All data and files are arranged in each directory accordingly for easy maintenance and data configuration.

The figure of directory structure is given below:



Fig: Directory Structure of my Project

I have used Git-Lab for this Project. My Git-Lab id is:sinoxcrest@gmail.com/sinoxcrest

Repository Link: https://github.com/sinoxcrest/UMeChat

Fig: Sign Up to Git Lab

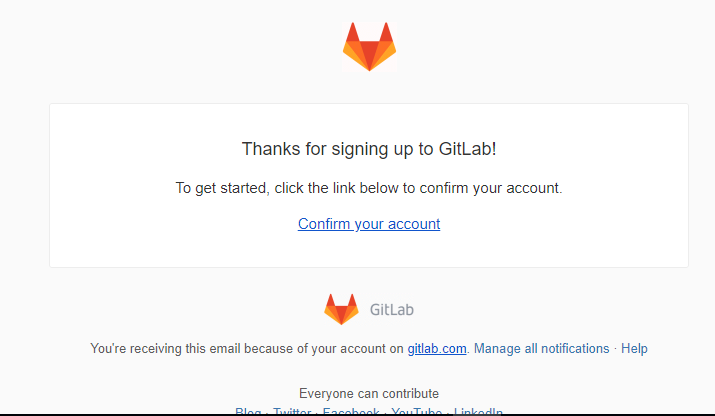


Fig: Conforming account on GitLab

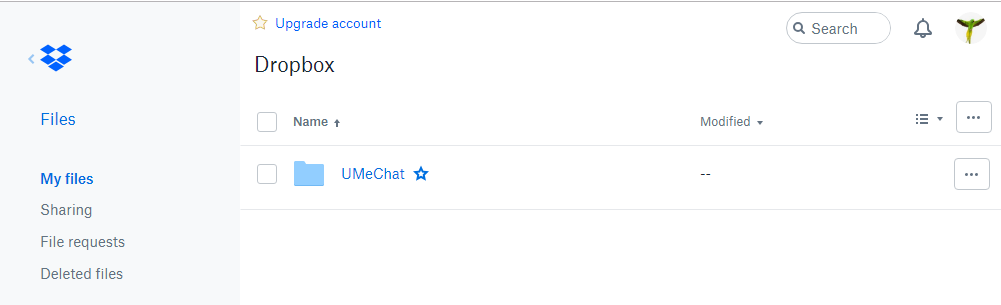


Fig: Dropbox Backup

# Conclusion

From this project I concluded many more things about Importance of Planning, Scoping and much more which are very essential in system development. Also we learn a lot about risk management and Design pattern as well. This very project has helped me in many aspects so I would like to thank my teacher for assigning me with this task. Also I came to know the necessity of Proposal based project Management, Importance of work breakdown structure, Design patterns to be followed, System architecture to be followed and much more. The Gantt chart made to pre-estimate time consumption was also stated in this proposal aspects which would help to further divide task work accordingly in my project. Thus this proposal project was a success and was quiet beneficial for my project development methodology.

# Reference and Bibliography

1. Itinfo.am. (2019). *Software Development Methodologies*. [online] Available at: http://www.itinfo.am/eng/software-development-methodologies/ [Accessed 8 Apr. 2019].
2. Java2s.com. (2019). *Java Design Patterns Tutorial - Java Design Pattern - MVC Pattern*. [online] Available at: http://www.java2s.com/Tutorials/Java/Java\_Design\_Patterns/0300\_\_Java\_MVC\_Pattern.htm [Accessed 8 Apr. 2019].
3. Pmbypm.com. (2019). *7 reasons to create WBS*. [online] Available at: https://www.pmbypm.com/7-reasons-to-create-wbs/#.XKuhdZgzbIV [Accessed 8 Apr. 2019].
4. WhatIs.com. (2019). *What is milestone? - Definition from WhatIs.com*. [online] Available at: https://whatis.techtarget.com/definition/milestone [Accessed 8 Apr. 2019].
5. Interserver Tips. (2019). *What is MVC? Advantages and Disadvantages of MVC - Interserver Tips*. [online] Available at: https://www.interserver.net/tips/kb/mvc-advantages-disadvantages-mvc/ [Accessed 10 Apr. 2019].

# Appendix



