## A

## Project Report

## On

# DDU FORUM

# BTech-sem VII

## Prepared By

## 

## Shubham V. Bhatt (IT-11)

## Umang V. Borad (IT-15)

**Hardik H. Adesara (IT-01)**

# 

## DEPARTMENT OF INFORMATION TECHNOLOGY

## FACULTY OF TECHNOLOGY,

## DHARMSINH DESAI UNIVERSITY

## COLLEGE ROAD, NADIAD- 387001

December, 2014

## A

## Project Report

## On

# DDU FORUM

# BTech-sem VII

## In partial fulfillment of requirements for

Bachelor of Technology

In

Information Technology

### Submitted By:

## Shubham V. Bhatt (IT-11)

## Umang V. Borad (IT-15)

**Hardik H. Adesara (IT-01)**

## Under the Guidance of

Prof. V. K. Dabhi



##### DEPARTMENT OF INFORMATION TECHNOLOGY

**FACULTY OF TECHNOLOGY, DHARMSINH DESAI UNIVERSITY**

**COLLEGE ROAD, NADIAD- 387001**

**CANDIDATE’S DECLARATION**

I/We declare that pre-final semester report entitled

“DDU FORUM ANDROID APPLICATION” is my /our own work conducted under the supervision of the guide Prof. V. K. Dabhi.

I/We further declare that to the best of my/our knowledge the report for B.Tech. VII semester does not contain part of the work which has been submitted either in this or any other university without proper citation.

Candidate’s Signature

Candidate’s Name: Bhatt Shubham V.

Student ID: 11ITUOS008

Candidate’s Signature

Candidate’s Name: Borad Umang V.

Student ID: 11ITUON011

Candidate’s Signature

Candidate’s Name: Adesara Hardik H.

Student ID: 11ITUOS001

Submitted To:

Prof. V. K. Dabhi,

Department of Information Technology,

Faculty of Technology,

Dharmsinh Desai University, Nadiad

Gujarat.

## DHARMSINH DESAI UNIVERSITY

## NADIAD-387001, GUJARAT



## CERTIFICATE

### This is to certify that the project carried out in the subject of Software Design Project, entitled “DDU FORUM ANDROID APPLICATION” and recorded in this report is a bonafide report of work of

### 1) Bhatt Shubham V. Roll No. IT-11 ID No: 11ITUOS008

**2) Borad Umang V. Roll No. IT-15 ID No: 11ITUON011**

**3) Adesara Hardik H. Roll No. IT-01 ID No. 11ITUOS001**

### Of Department of Information Technology, semester VII .They was involved in Project training during academic year 2014 -2015.

Prof. V. K. Dabhi

(Project Guide),

Department of Information Technology,

Faculty of Technology,

Dharmsinh Desai University, Nadiad

Date: 13/10/2014

Prof. R.S.Chhajed,

Head, Department of Information Technology,

Faculty of Technology,

Dharmsinh Desai University, Nadiad

Date: 13/10/2014

## ACKNOWLEDGEMENT

This project was successfully completed under the guidance and support of

Prof. V. K. Dabhi. We are thankful to sir for guiding us.

## Shubham V. Bhatt (IT-11)

## Umang V. Borad (IT-15)

**Hardik H. Adesara (IT-01)**

**Table of Contents**

**1. Introduction…………………………………………………………………………9**

1.1 Project details……………………………………………………………….9

1.2 Purpose……………………………………………………………………...9

1.3 Scope………………………………………………………………………..9

1.4 Objective……………………………………………………………………10

1.5 Technology and Literature Review…………………………………………10

**2. Project Management………………………………………………………………...11**

2.1 Feasibility Study…………………………………………………………….12

2.1.1 Operational Feasibility……………………………………………12

2.1.2 Technical Feasibility……………………………………………...12

2.1.3 Economic Feasibility……………………………………………...12

2.2 Product Perspective …………………………………………………………13

2.3Product Functions ……………………………………………………………13

2.4User Classes and Characteristics …………………………………………….13

2.5Design and Implementation Constraints …………………………………….14

2.6User Documentation …………………………………………………………14

2.7Assumptions and Dependencies ……………………………………………..15

**3. External Interface Requirements …………………………………………………..17**

3.1 User Interfaces ……………………………………………………………...18

3.2 Hardware Interfaces ………………………………………………………...18

3.3 Software Interfaces …………………………………………………………18

3.4 Communications Interfaces…………………………………………………19

**4. Features……………………………………………………………………………….20**

4.1 Requirements of new system ………………………………………………..21

4.1.1 Use case diagram ………………………………………………….21

4.1.2Managing your profile……………………………………………...21

4.2 Features of new system………………………………………………………22

4.3 Activity Diagram……………………………………………………………..25

4.4 Sequence Diagram……………………………………………………………26

4.5 Class Diagram………………………………………………………………...27

**5. System Design…………………………………………………………………………28**

5.1 System Architecture Design………………………………………………….29

5.1.1 Deployment Diagram……………………………………………….29

**6. Implementation Planning……………………………………………………………………30**

6.1 Implementation Environment…………………………………………………………31

6.2 Coding Standards……………………………………………………………………...31

.

**7. Testing………………………………………………………….……………………….32**

7.1 Testing Plan…….……………………………………………………...............33

7.2 Testing Strategy……………………………….……………………………….33

7.3 Testing Methods……………………………………...…………………..........34

7.4 Test Cases………………………………………………...…………………....35

7.4.1 Test Cases of System…………………………….………………......35

**8: User manual…….……………………………………………………………………..36**

**9: Limitation & Future Enhancement………………..…………………………………46**

9.1 Limitation……………………………………………………………………..47

9.2 Future Enhancement…………………………………………………………..47

**10: Conclusion & Discussion…………….………………………………………………48**

10.1 Conclusion & Future Enhancement…………………..……………………...49

10.2 Discussion…………………………………………………………………,,,.50

10.2.1 Self-Analysis of Project Viabilities………………………………………..50

10.2.2 Problem Encountered and Possible Solutions……………………………..50

10.2.3 Summary of Project Work……………………………………,,,….50

**11: References…………………………………………………………………………….51**

**CHAPTER 1**

INTRODUCTON

**1.1 Project Summary:**

Do you want to exchange text messages with your friends, family or colleagues quickly, easily and at no cost? “Wi-Fi Groups” is the best way to do it!

Wi-Fi Groups” enables all the devices connected to the same Wi-Fi network to create groups and freely communicate with one another.

You can either create your own Wi-Fi hotspot on your device and have others connected to it, or connect your device to an already existing network and let the fun begin!

Wi-Fi Groups makes use of the Samsung Chord technology.

The Samsung Chord is a brilliant and fun way to share content and Student events in real-time between devices. The Samsung Chord (Chord) SDK allows application developers to develop local information-sharing applications without a detailed knowledge of networking.

**1.2 Purpose:**

• Easy discovery and connection with nearby devices

• Real-time experiences between multiple devices

• Fast peer-to-peer connection, without the need of a server

• Decentralized networks where peers can come and go at any time

**1.3 Scope:**

The Application has to be built over the application framework. To develop an Android Application Package (APK) for instant messaging using the Wi-Fi direct.

The application features include:

• Simple text messaging

• Chat Service

• Multimedia content sharing i.e. image, audio, video and application

For example, image/jpeg, GIF, audio/mp3, video/mp4, and application/ MS-Word

**1.4 Objective:**

This SRS document describes the detailed requirements and specifications of the mobile application “DDU FORUM” that will be used for interactivity and communication in a Wi-Fi enabled zone in public places like malls, college-canteens, Airports etc. As the application is divided into two parts- one for the premium users including the shopkeepers, the brand owners, etc. and the other for the regular users consisting of the customers/visitors at the public places mentioned above, the requirements of both the entities have been captured in detail. This document covers the essential characteristics of the application like the User-Interface, the software interface, the hardware interface, the features provided by the application to the user, assumptions & constraints, along with the other non-functional requirements containing performance, security and quality attributes. Thus, this document covers the scope of the software product taking the whole application as a system.

**1.5 TECHNOLOGY AND LITERATURE REVIEW**

Android is a Linux-based operating system designed primarily for touchscreen mobile

devices such as smartphones and tablet computers. Android is written primarily in a

Customized version of the Java programming language.

It has its own virtual machine for each and every process, called Dalvik. Android is just an abstraction of Java which is designed especially for mobiles and tablets. Hence all the Java files are converted into the Dex files also known as Dalvik Excutable files. Android applications run in a sandbox, an isolated area of the system that does not have access to the rest of the system's resources, unless access permissions are explicitly granted by the user when the application is installed.

Hence each and every process is extremely secure and runs in its own environment.

Also the most appealing feature of Android is its simple yet impressive user interface

due to which it is in ever increasing need of the market.

**CHAPTER 2**

PROJECT MANAGEMENT

**2.1 FEASIBILITY STUDY**

**2.1.1 Operational Feasibility**

This app needs to be operated in a Wi-Fi enabled area, but does not require the use of internet. It works throughout the day, provided the user is within a local area network. Choosing Wi-Fi as a medium and not any other is because today, most of the people use internet using Wi-Fi and it is available at most of the crowded public places and is the optimum place for communicating things/information. Also for communicating the advertisements, coming from the premium users, there needs to be a mechanism by which a new user entering the local network will be given access to all the pre-existing messages/ads from the premium users. This mechanism comprises of a process in which whenever a new user enters, then, out of the already connected users who have the access to all the ads, few (say 10% of the available users) will be selected randomly who will be considered for passing on the ads to the new user. Then, new user will be provided with the information from one of the already existing users, who are not busy (not passing on information to any other user) for passing on the ads/messages. Also the advertisements will be in text form, which will allow fast spread of information and prevent the exchange of information from being a time-consuming process. So, this method is feasible as regards to the operations within the system, making the app more efficient and resourceful.

The layout of the app with all its tab is simple enough for users to understand, provided the user knows how to operate the smart gadget and poses no problem regarding its usage.

**2.1.2Technical Feasibility**

Being an android/smart-phone application, there is nothing new for the users to learn about using it, as most of the people today use touch technology and are well-versed with it. The product is easy to deploy and can be obtained and installed through app stores on respective smart phones easily. Our main research would concern implementation of ways to enable broadcast over the Wi-Fi and for all the users on the Wi-Fi to be able to detect each other and reach out through a common space. We are also planning to port the app to other platforms like iOS for I-phone because this would increase the user base, as these two Operating systems for smart phones are most prominent these days.

**2.1.3Economic Feasibility**

Fortunately, most of the work has to be done by the software in this application, so only the expected operational costs matter. As discussed before, today the users are generally quite well versed about using the smart phones/I-phones, so training them about using this app is actually not required because the application is self-explanatory through its features. Here, we cut down on “training the user” cost. We make use of the prevalent technology and hence, avoid any cost in training the developers. The app is affordable by a normal user as the user is not charged of anything extra. The app doesn’t depend on any external hardware or human resource for maintenance and operation and is self-sufficient to run on its own. The app would cost a very small amount to the users who want to sell their products and become authentic and privileged users, but which is very much affordable for anyone since the people would be able to convey messages to their target audience so easily and proficiently. This would in turn, increase the scope of their own target audience.

**2.2 PRODUCT PERSPECTIVE**

The mobile application will be self-sustained and self-contained. It will be an autonomous system. The application will have two sections, one will be getting data from the users using the Wi-Fi device and the other section of the application will manage data. As the data will be exchanged among the users, who will be connected over a particular Wi-Fi (may not be having internet access) there is no scope of having an online server. The data that will be communicated will be stored on the databases stored on internal storage of the phone.

The mobile application will need to communicate with other users using Wi-Fi, the application will have to send some data and receive some. This all will be done over the Wi-Fi technology. The list of currently available users will be provided by the Wi-Fi application. The Wi-Fi application will carry out the tasks by using the Wi-Fi device hardware, see Figure 1.

The Database application will manage all the data that will be shared with the users, i.e. what data has to be displayed. The database application will fetch data from the database stored on the internal storage and from the Wi-Fi application. This can be seen in the Figure.

**2.3 PRODUCT FUNCTIONS**

The mobile application will let the users view the ads in an authentic advertisement tab and would allow the users to view all other messages in different tabs. It will allow the users to post a message in particular tab (except the authentic ad tab). The app will allow users to up vote/like or down vote a specific ad (message) in the authentic ad tab.

The user will be allowed to search a specific string and the resulting output will be of the form where the searched string will be highlighted.

The users will be allowed to chat privately with the other users if both the parties agree.

**2.4 USER CLASSES AND CHARACTERISTICS**

The DDU Forum project is meant to offer people at public places (example : a shopping mall, a library, etc.) , a platform which will provide them the ease of getting to know the events happening at that place better i.e. what is happening at that place at that instant of time.

Main user classes of this product are:

**2.4.1 Premium Users**:

They can use our android app to post their advertisements and offers (in case of malls, the shop owners) which can be seen by other non-premium users. This feature will be exclusively available to premium users. The normal user features like help, buy/sell, private messaging will also be available to them.

**2.4.2 Regular Users**:

|  |  |
| --- | --- |
|  |  |

The users who visit a Wi-Fi enabled public place come under this class. These users are not given the premium access in order to keep the spamming under control. These users can ask for help, provide help, post queries, browse through the deals and offers made available by the premium users and respond to personal ads by other regular users too.

**2.4.3 Users should know the following things:**

a) How to operate an android cellphone

b) How to download an application from Google Play Store.

c) How to install an application

**2.5 OPERATING ENVIRONMENT**

The main component of the Loc-Talk project is the software application, which will be limited to the Android operating system (specifically Android 2.2 and above). The application is not resource- or graphics-intensive, so there are no practical hardware constraints. The app will rely on several functionalities built into Android’s Application

Programming Interface (API), so ensuring appropriate usage of the API will be a major concern. Beyond that, the application is a self-contained unit and will not rely on any other hardware or server.

**2.6 DESIGN AND IMPLEMENTATION CONSTRAINTS**

**2.6.1 The following is the compilation of limitations and constraints on the system:**

1. The current API (Wi-Fi P2P) we are using is feasible only for android devices running on android API version- 4.0 and above.

2. Android’s Wi-Fi P2P framework compiles with the Wi-Fi Direct Certification program.

3. We are using a user’s mobile phone as the server/database and hence this imposes a computing constraint on user’s mobile phones as there can be phones of varying

RAM.The larger the RAM, the faster the computation.

4. Now, irrespective of the RAM, our application should run smoothly and quickly without occupying a lot of ‘App Data’ :

a. A large number of ‘Premium Users’ can incur a large amount of data as the

Authenticated Ads file can be of a very larger size.

b. Since, number of users can be very large, hence, there might be a lot of App Data in terms of chats and non-Authenticated posts.

|  |
| --- |
|  |

5. Since, there can be a large number of users connected to the same Wi-Fi and adhering to the limitation on the Bandwidth of the local host communication channel, we had to send and receive data (with all of the users doing the same simultaneously) as quickly as possible for which the file size has to be reasonably small.

6. Our application must adhere to the Android terms of service for their mobile Applications.

**2.6.2 The following is the compilation of implementation aspects of the system:**

1. Java is used for the in-App coding and designing Algorithms for managing data and other stuffs of the Application.

2. Wi-Fi peer to peer APIs have been used to connect, register, discover and share data with other peers on the network.

3. For managing App Data space on the user’s phone, we have limited the Authenticated Ads file and non-Authenticated posts file to be of ‘Text-Only’ form.

Hence, files can even be sent and received quickly over the same Wi-Fi with all theusers using the network at simultaneously.

4. For enabling users to view Ads only of some certain shops, we have implemented a search/find feature.

5. To organize the non-Authenticated posts as per their context, we have created different ‘Tabs’/’Columns’ dedicated to some specific context.

**2.7 USER DOCUMENTATION**

We are providing online documentation for users which we can call it as online help or on-screen help. Online documentation is so designed that it is suitable for viewing in screen. Basically we are using Web Help which has a similar structure and functionality to compiled HTML Help. Web Help is cross-platform help that runs on Windows, UNIX, Mac OS, and other operating systems. To view Web Help, a user needs only a web browser. Web Help of the software has following features:

1. Necessary background information and theory about the subject.

2. Full information about the product.

3. An explanation of each dialog box, screen, field, tab, and button.

4. An explanation of all the options that users have.

**2.8 ASSUMPTIONS AND DEPENDENCIES**

Since, usage of various features wasn’t clear from the survey and interviews, hence, the following dependencies are still not clear:

1. Number of people purchasing the Application to become the ‘Premium users’ in order to get their Ads featured in the Exclusive Authenticated Tab is not fixed.

2. Bandwidth and range of the local host communication channel (Wi-Fi) is not fixed as it will vary from location to location.

3. Presence and working of Wi-Fi in every targeted place is not ensured.

4. Consent of people in viewing the Ads on their phone before moving on to various shops and purchasing is not yet guaranteed.

5. Android has to accept the Application and publish it to their App store.

**CHAPTER 3**

**EXTERNAL INTERFACE REQUIREMENTS**

**3.1 USER INTERFACES**

1. On connecting the Wi-Fi and starting the app (or resuming) the authentic ads tab will be viewed.

2. The messages (ads, public messages and private messages) will be viewed in thread view format.

3. The tabs (e.g. authentic ads tab, private chat tab, help tab, etc.) could be accessed by sliding from left section to right section. These tabs will be viewed in list view format.

4. The Premium users will be allowed to post in the authentic ads tab through a text field, while Regular users will not be shown this text field.

5. The Regular users will be allowed to input the text in other tabs using the text field.

6. User Profile section could be accessed through “menu” button and that will be in a custom list view format.

**3.2 HARDWARE INTERFACES**

Since neither the database application nor the Wi-Fi application have any designated hardware, it does not have any direct hardware interfaces. The Wi-Fi is managed by the Wi-Fi application in the mobile phone and the hardware connection to the database server is managed by the underlying operating system on the mobile phone.

**3.3 SOFTWARE INTERFACES**

1. We will use Java for rendering the User Interface, for which, Android 4.0 and above with JVM and SDK is a must.

2. Eclipse 3.8.2 will be used for all programming practices.

3. SQLite3 database for storing and accessing the text formatted advertisements and other text formatted information.

4. Photoshop and InkScape will be used for designing the logos and icons.

5. Hierarchy Viewer will be used for debugging and optimizing the application's user interface.

6. Layout opt will be used for quickly analyzing our application's layouts. Android Emulator for designing, debugging, and testing our application in an actual Android run-time environment.

7. We plan to use the following Java libraries

a) android.net.wifi

b) java.net

c) java.io

d) java.lang

e) org.json

8. Data Items:

a) User’s Profile Data : Username: String (at the time of application setup, editable)

b) User Joining Message: Ping broadcasted by the user on connecting to the

c) Network’s service to the available peers.

d) User’s Input Data: Text formatted input for posting in a tab.

e) JSON object: Input from the user is converted into a JSON object and

f) Broadcasted to the available peers.

g) System data : All the parsed JSON objects from all the users (at a point of time)

h) and the peer list are stored in a database on every peer’s device, which will be

i) Retrieved and manipulated through SQLite.

9. Data sharing mechanism is constrained to only text input (no other input format is allowed) to lower the storage demands and maintain efficient and quick distribution of data in the network.

**3.4 COMMUNICATIONS INTERFACES**

1. We will be using Wi-Fi Peer-to-Peer API (level 14 API, Android 4.0) to enable the device to register for the application’s service, connecting to the peers, retrieving the list of available peers.

2. Messages between devices will be exchanged through transferring of JSON objects which will be created on the sender’s side and parsed on the receiver’s side. Sockets will be used to send and receive these messages.

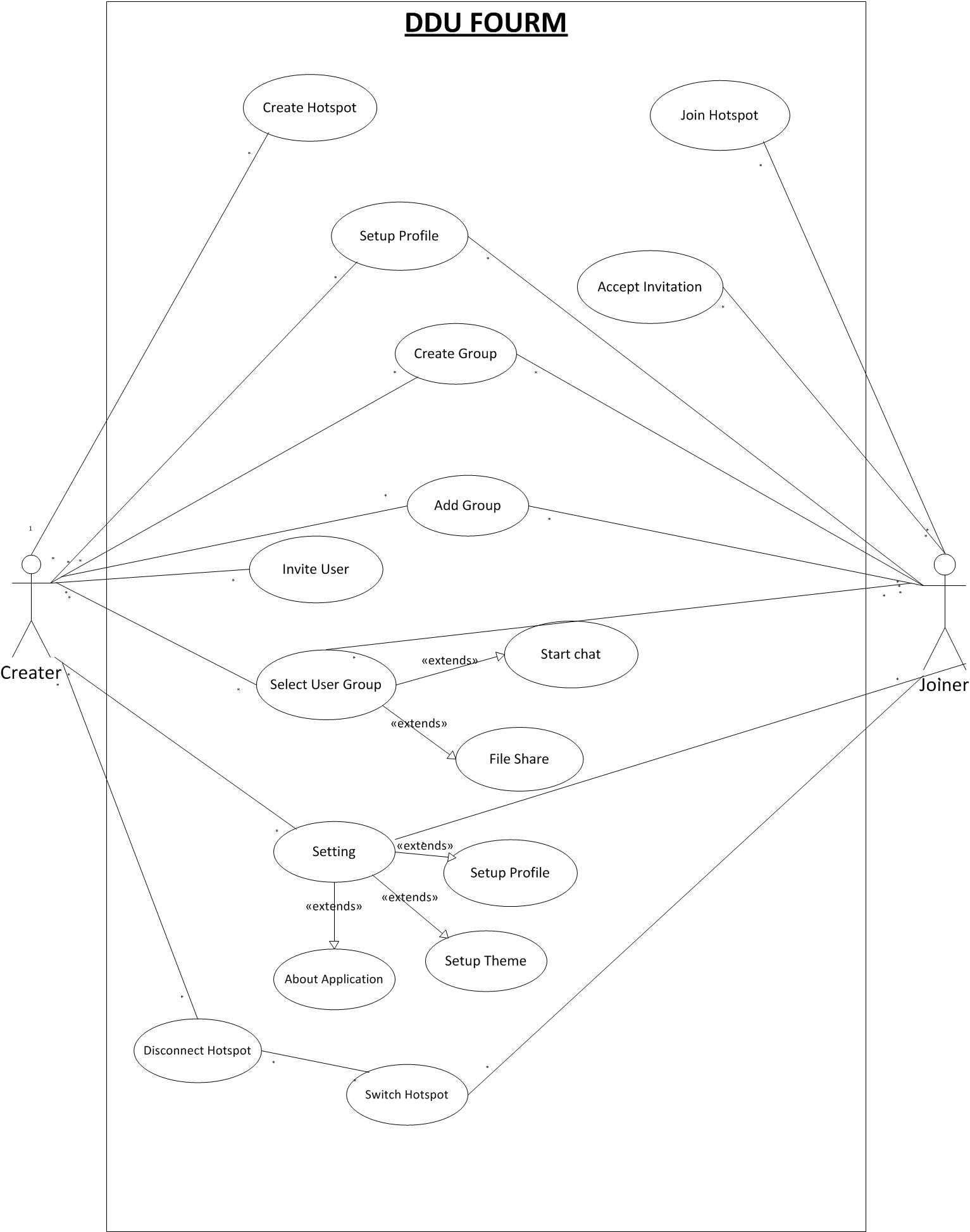
Synchronization mechanism is implicitly available through Wi-Fi P2P API that we are using. The current list of peers can be obtained through the functions already available in the API and the data is continuously being synced through exchanges of JSON objects.

**CHAPTER 4**

SYSTEM ANALYSIS

**4.1 REQUIREMENTS OF NEW SYSTEM**

**4.1.1 USE CASE DIAGRAM**

****

**4.1.2 SYSTEM REQUIREMENTS (SRS)**

**Functional Requirements**

This section of the document illustrates different functions provided by the

application:

**R1: Sign up/Registration:**

**Input:** Login/Registration screen will display text view for inputting Student ID and password**.**

**Output:** If Student id and password are correct then successfully profile is created.

**Pre-condition:** DDU Forum should be installed on the system and signup screen should be available on application.

**Post-condition:** Sign up must be successfully done.

**Flow:** After entering user id & password it will check the database and return confirmation message if data valid.

**R2: Log in**

**Input:** Student id and password is required for successful completion of login phase.

**Output**: After clicking on login button, control goes to Student profile and Student can see as well as access all available functionalities.

**Pre-condition:** Student that login into the system must have registered account and login option should be available on the screen.

**Post-Condition**: Login should be successfully done to enter into main screen.

**Flow:** After completion of registration, control is redirected to login screen, if a user has logged in previously on a device; it will save the information and will not ask for login information again unless the user clears the cache of the system.

And after inserting Student id and password and clicked on log in, main screen of the Student should be displayed.

**R3: Main Screen**

**R3.1: Chatting**

**Input**: Create a Wi-Fi hotspot and join it.

**Output**: Display connected Student or Group.

**Pre-condition**: Student has to select group (multi chat) or single Student (personal chat).

**Post-condition**: Student must be able to do chat either with group or single Student.

**Flow**: Authentic Student can join group or single Student for chatting.

**R3.2: Sharing Data**

**Input**: Select file which Student want to share.

**Output**: Intended Student’s screen will get prompt message to accept the file.

**Flow**: Student who wants to share file can select it from his directory and intended Student will receive file using Wi-Fi technology.

**R4: Multi-Chat & Sharing**

**R4.1: Using Group**

**Input**: Student can be able to select for chatting & sharing.

**Output**: Student can now successfully join the selected group.

**Pre-condition**: There must be at least one group to join.

**Flow**: List of all group are displayed then any one group can be selected at a time and with that group chatting can be done.

**R4.2: With selected Students**

**Input:** Select the Student whom we want to chat or share.

**Output:** The Student is now connected with the selection he made.

**Pre-condition:** There must be at least two Student connected with hotspot.

**Post-condition**: All connected Student can be able to receive messages or files.

**Flow:** List of all hotspot connected Students are displayed and from them Student are selected for chatting or sharing.

**R5: Settings**

**R5.1: Setup Profile**

**Input:** Student can change their profile pic or nickname.

**Output:** All changes are performed.

**R5.2: Theme**

**Input:** Student can change theme of application.

**Output:** Accordingly themes are changed.

**R5.3: About DDU Forum**

**Input:** Select “About” option.

**Output:** It will show detail information for how to use this application and information regarding developers.

**Non - Functional requirements:**

**R1: Advertisement:**

Under this facility, Advertisement is displayed on the screen.

**R2: Help:**

Help option provides various FAQS or general query solution or counseling related general help.

**R3: Contact:**

Student can contact to toll free number of this DDU Forum if he has any queries and he wants its solution. This option is available on the home page.

**R4: Database:**

There will be a database system for all the information about the Student, which is helpful for contact with Student and advisor.

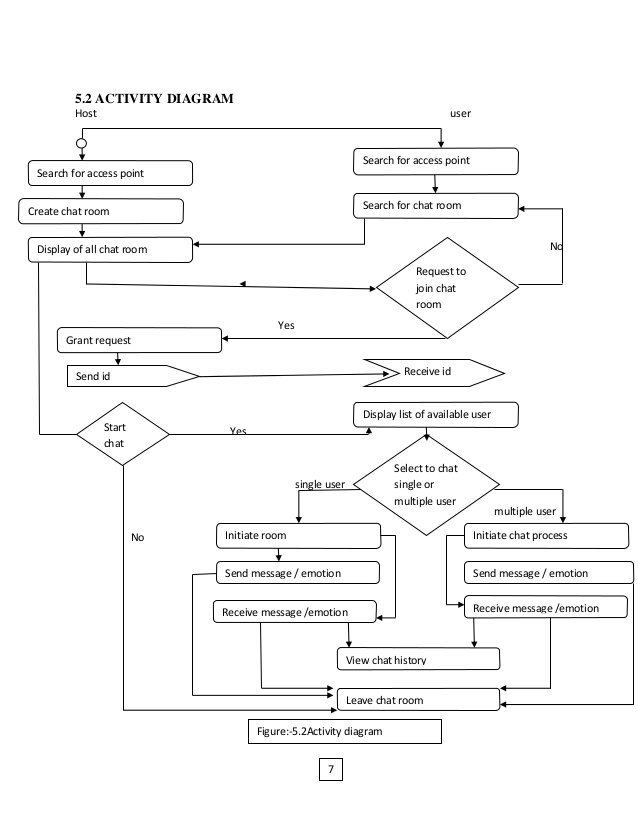
**R5: Scalability:**

When there is large number of Students try to login into DDU Forum then system should handle it in efficient manner.

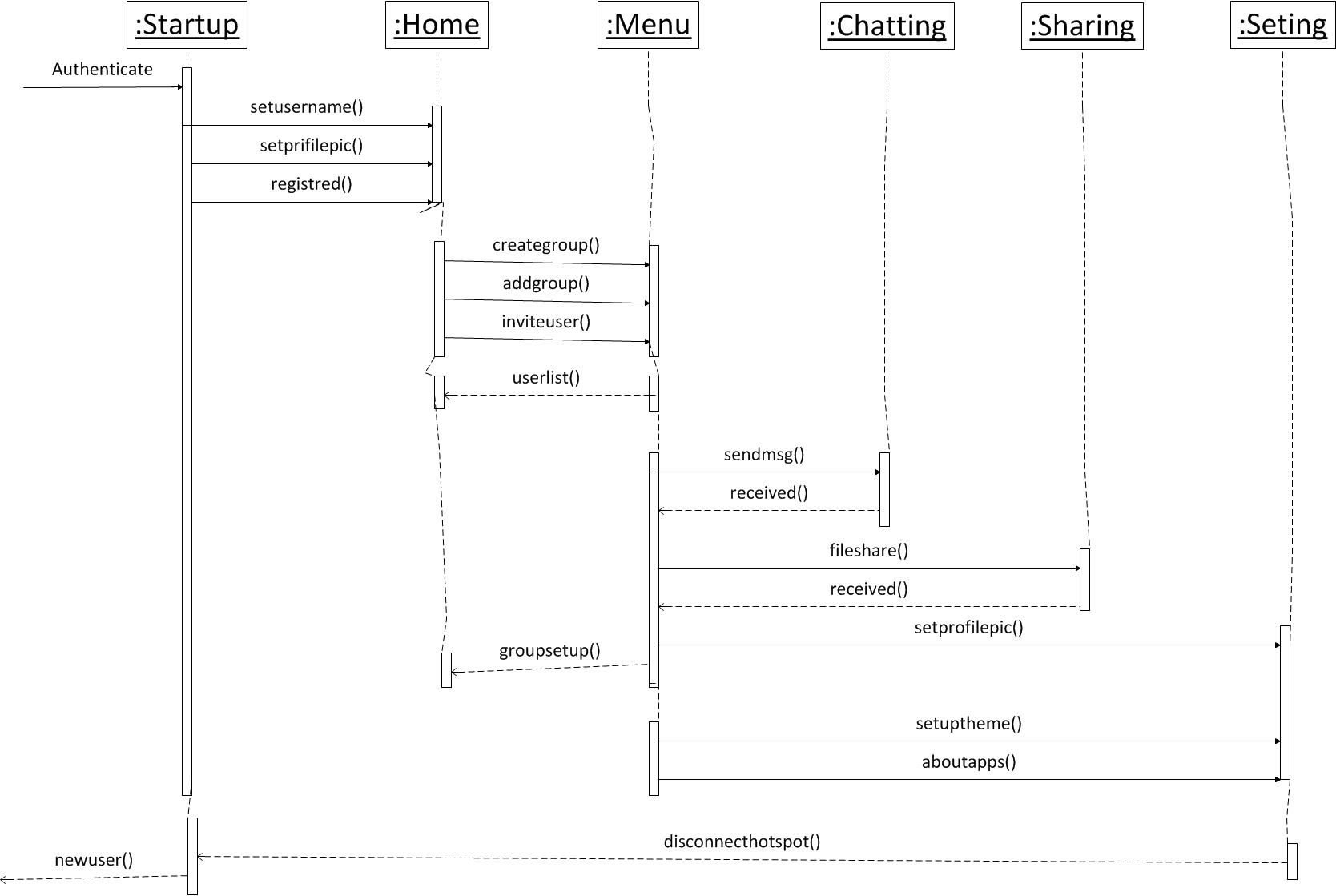
.

**4.2 FEATURES OF NEW SYSTEM**

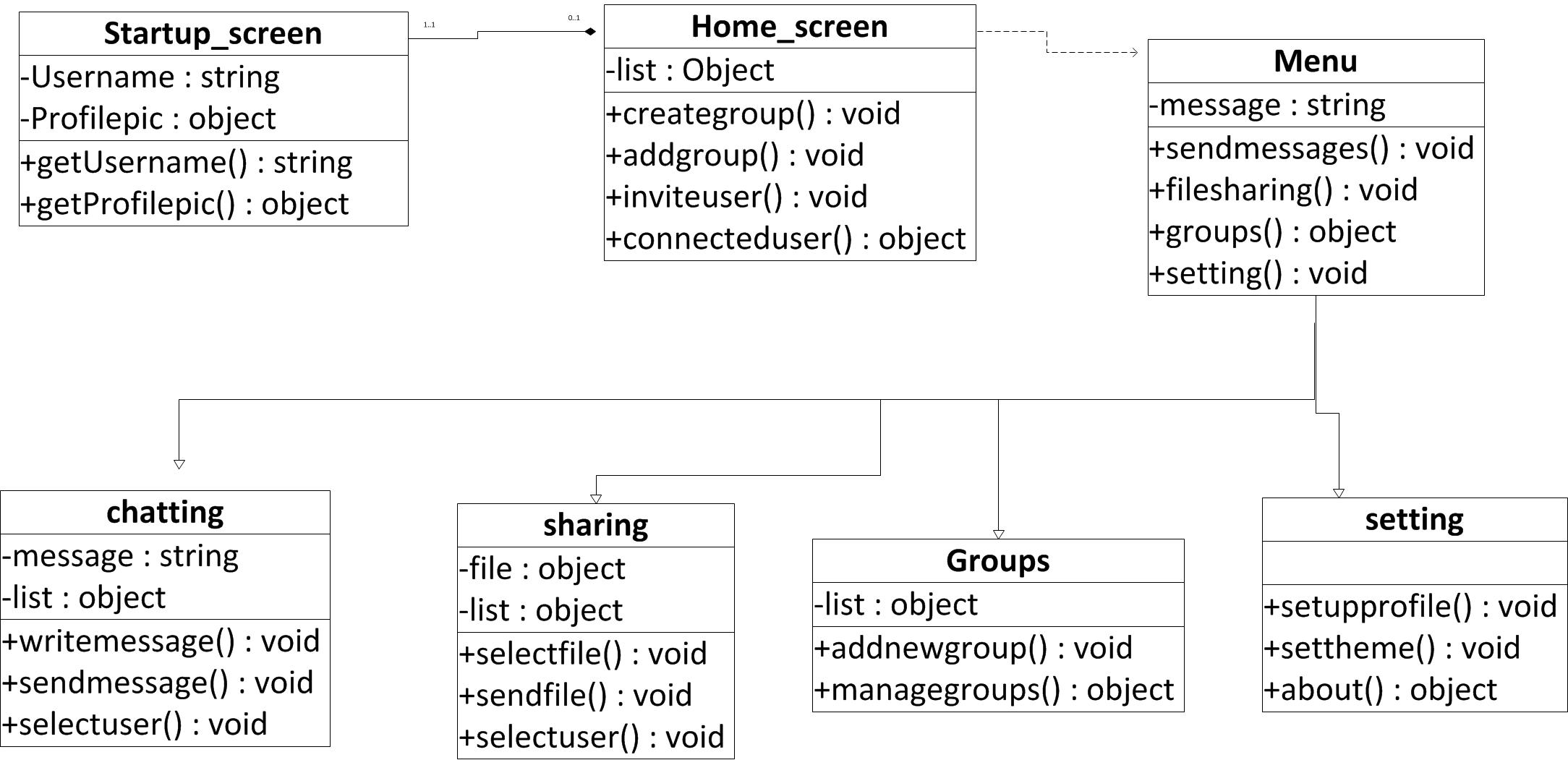
**4.3 ACTIVITY DIAGRAM**

****

**4.4 SEQUENCE DIAGRAM**



**4.5 CLASS DIAGRAM**



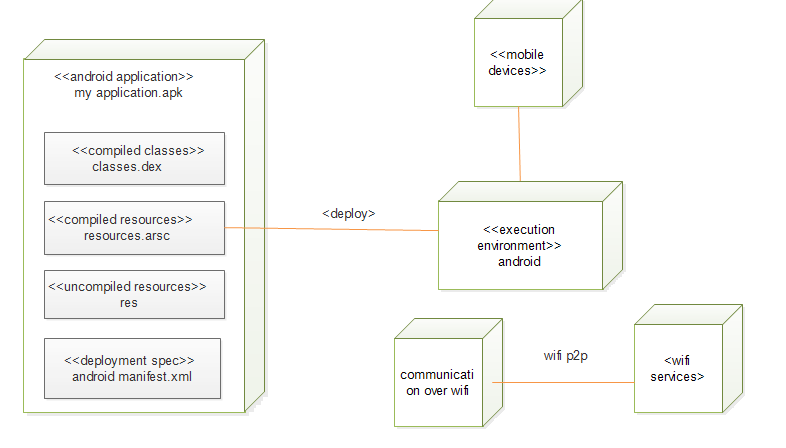
**CHAPTER 5**

SYSTEM DESIGN

**5. SYSTEM DESIGN**

**5.1 SYSTEM ARCHITECTURE DESIGN:**

**5.1.1 Deployment Diagram:**

****

**CHAPTER 6**

IMPLEMENTATION PLANNING

**6.1 IMPLEMENTATION ENVIRONMENT**

The system is an Android Based application.

**Multi-user vs. Single-user:**

Single user applications are the application where it is useful to only one user

at a time.

While in Multi user given application is used by many user at the same time and

thus web application are used by many users at the same time. Our system is a

multi-user system as we have more than one user who can use the system at a

same time.

**GUI vs. Non-GUI:**

Non GUI application uses command Prompt for input and output while GUI

application has graphics form to interface and other graphics property for various

I/O operation and are easy to use

Our System is a GUI based and thus easy and effective to use therefore user can

easily give input and take Input.

**6.2 CODING STANDARDS**

The front-end of the project is based on Eclipse. The project Library Management

System on Android had fixed standard for designing the GUI so that all modules are

found consistent as far as GUI is concerned. The GUI standards includes following

specifications.

The sized of components like buttons, textboxes, combo-boxes, list etc.

All the authority is given to only administrator.

The coding standard provides the guideline for coding a module during

development. When these standards are strictly followed the code becomes

more readable and understandable making the process of debugging easy.

Specifications for coding loops and other control structures.

Specifications for defining the variables etc. and defying the functions.

**CHAPTER 7**

TESTING

**7.1 TESTING PLAN**

**What is ‘Software Testing’?**

Testing involves operation of a system or application under controlled conditions

and evaluating the results. The controlled conditions should include both normal and

abnormal conditions. Testing should intentionally attempt to make things go wrong

to determine if things happen when they don’t happen when they should. It is

oriented to ‘detection’.

**The need for Testing:**

No matter how good a programmer is, no application will ever be one hundred

percent correct. Testing was important to us in order to ensure that the application

works as efficient as possible and conforms to the needs of the system.

Testing was carried out throughout the development of the application, not just

the application has been developed, as at this stage it took a great deal of effort

to fix any bugs or design problems that were occurred.

**7.2 TESTING STRATEGY**

When our application was configured and customized in the system, the test was

observed that this configuration or customization does not cause any improper

processing or violation. The following care was taken when the application was

developed at the local machine.

The interface may have something not proper, which can be tested by this checklist:

Number of input parameter equal to number of argument?

Parameter and argument attributes match?

Number of arguments transmitted to called forms equal to number of

parameters?

Attributes of arguments transmitted to called forms to attributes of

parameters?

Number attributes and order of arguments to built-in functions correct?

The local data structures for a form are common source of errors. The

following types of errors should be searched for,

Improper or inconsistent typing

Erroneous initialization or default values

Incorrect (misspelled or truncated) variables names

Inconsistent data types

Underflow, overflow and addressing exception

As far as unit testing is concerned we did it at the time of coding in an

informal but extensive way, so as to reduce number of problems arising out

of incorrect syntax, incorrect variable, function names etc.

Close the database connection when not required.

Care was taken to check for any infinite loop that exists in code before

executing the code.

**7.3 TESTING METHODS**

**While Box Testing**

Also known as glass box, structural, clear box and open box testing. A software

testing technique whereby explicit knowledge of the internal workings of the item

being tested are used to select the test data. Unlike black box testing, white box testing

uses specific knowledge of programming code to examine outputs. The test is accurate

only if the tester knows what the program is supposed to do, it means that he must be

completely aware that for particular input a particular output must be obtained.

The main benefit of this type of testing is Tester can see if the program diverges from

its intended goal.

This test concentrates on the examination of the code rather than the specification. We

have included three different forms of white box testing.

**Statement Coverage Criterion:**

This is the simplest coverage criterion. We are checking in it that each statement of the

program was executed “at least once”.

**Branch Coverage Criterion:**

An improvement over statement is **Branch Coverage**. In that we are running a series

of test to ensure that all branches are tested at least once.

**Path Coverage Criterion:**

There are many errors which were not detected by statement or branch testing. The

reason is that some errors are related to some combination of branches and it may be

not check in other test. We are checking in this test is all path of programs are executed

or not.

**Black Box Testing**

Black-box and white-box are test design methods. Black-box test design treats the

system as a "black-box", so it doesn't explicitly use knowledge of the internal structure.

Black-box test design is usually described as focusing on testing functional

requirements. Also known as behavioral, functional, opaque-box, and closed-box.

Black Box Testing was helpful us to find error such as:

Interface error

Incorrect or missing functions.

Testing

DDU(Faculty Of Tech., Dept. Of IT)

Errors in data structures or external database access.

Performance Errors.

**Unit Testing**

Unit testing is a method of testing the correctness of a particular module of source

code. The idea is to write test cases for every non-trivial function or method in the

module so that each test case is separate from the others if possible. The developers

mostly do this type of testing.

In this method of testing we test all individual components to ensure that they operate

correctly. Each component are tested independently without other system components.

**Integration Testing**

It is the phase of software testing in which individual software modules we are

combined and tested as a group. It follows unit testing and precedes system testing.

The purpose of Integration testing is to verify functional, performance and reliability

requirements placed on major design items.

It takes as its input modules that have been checked out by unit testing, groups them in

larger aggregates, applies tests defined in an Integration test plan to those aggregates,

and delivers as its output the integrated system ready for system testing.

**7.4 Test Cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Description** | **Prerequisite** | **Expected Output** | **Result Output** | **Pass/Fail** |
| 1.1 | Connect to local Wi-Fi | The Wi-Fi should be enabled in the device | Successful establishment of connection | Successful establishment of connection | Pass |
| 1.2 | Destination address | Server should be running properly | Valid IP address should be provided | Valid IP address is provided | Pass |
| 1.3 | Setting user  name | The application should be installed | The user should be able to set his desired user name | The user is able to set his user name | Pass |

**Test Object**

**CHAPTER 8**

USER MAUAL

**8. USER MANUAL**

A user guide or user's guide, also commonly known as a manual, is a technical

communication document intended to give assistance to people using a particular

system. It is usually written by a technical writer, although user guides are written by

programmers, product or project managers, or other technical staff, particularly in

smaller companies.

User guides are most commonly associated with electronic goods, computer hardware

and software.

Our user guides contain both a written guide and the associated images. In the case of

our application, it is usual to include screenshots of how the program should look. The

language used is matched to the intended audience.

We have prepared our user manual according to various user roles and module wise, so

a novice user can understand system very quick. There are basically 3 roles tied with

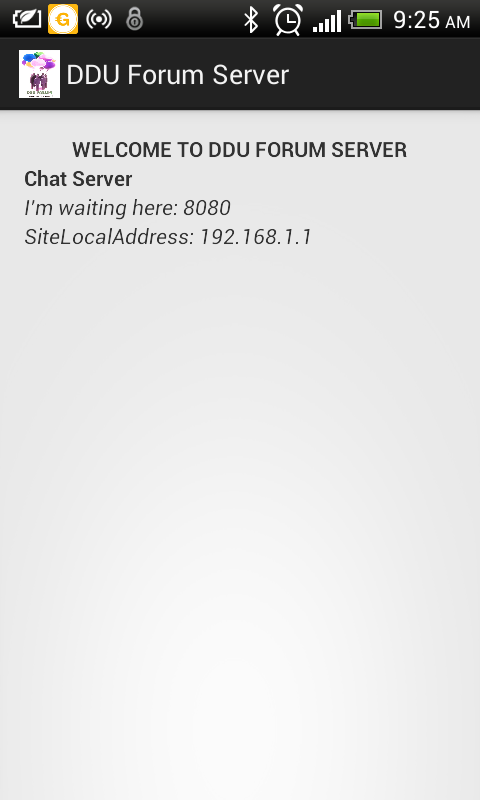
system. But only Student role is applicable in Android Application.

**Logo of DDU FORUM:**

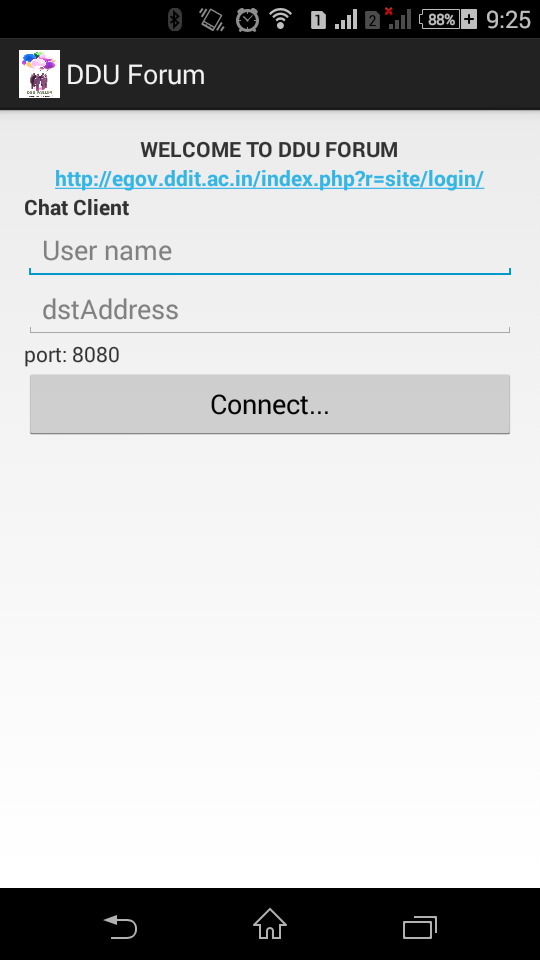
****

**Execution of Application**

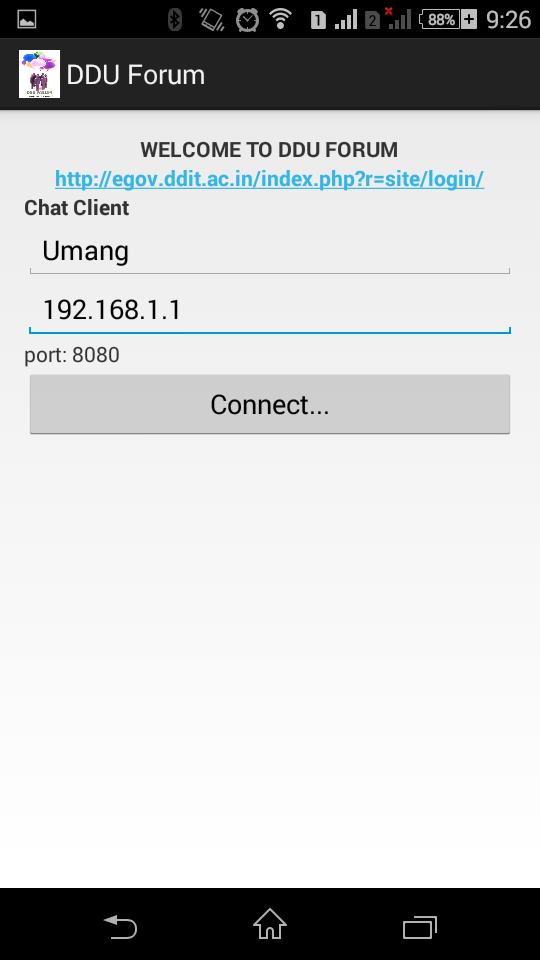
**Step 1: Server is ready to handle & monitor clients.**

****

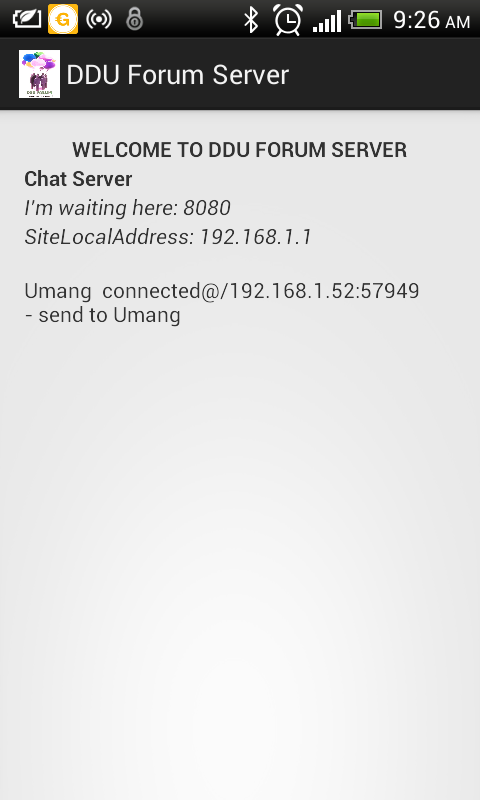
**Step 2: Client is ready to connect with server.**

****

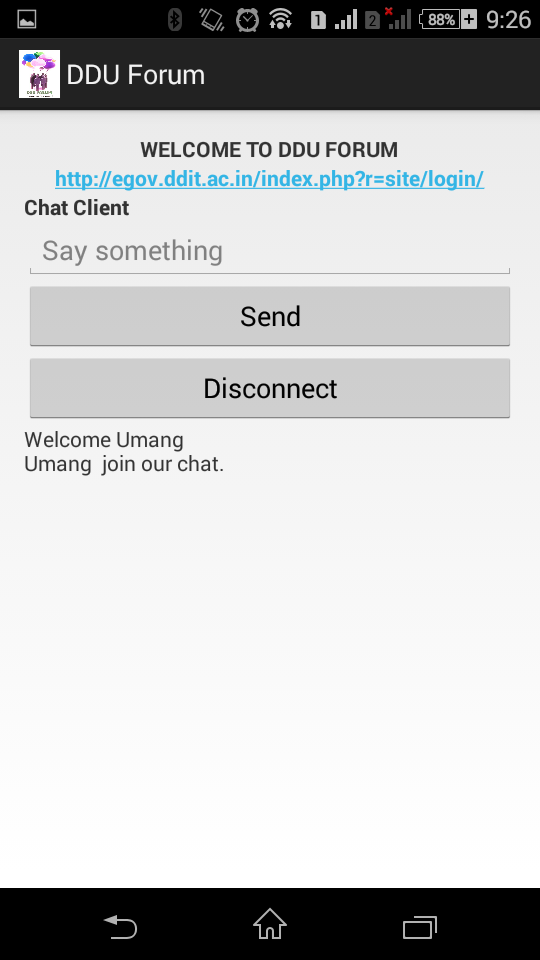
**Step 3: Client request to server for connection.**

****

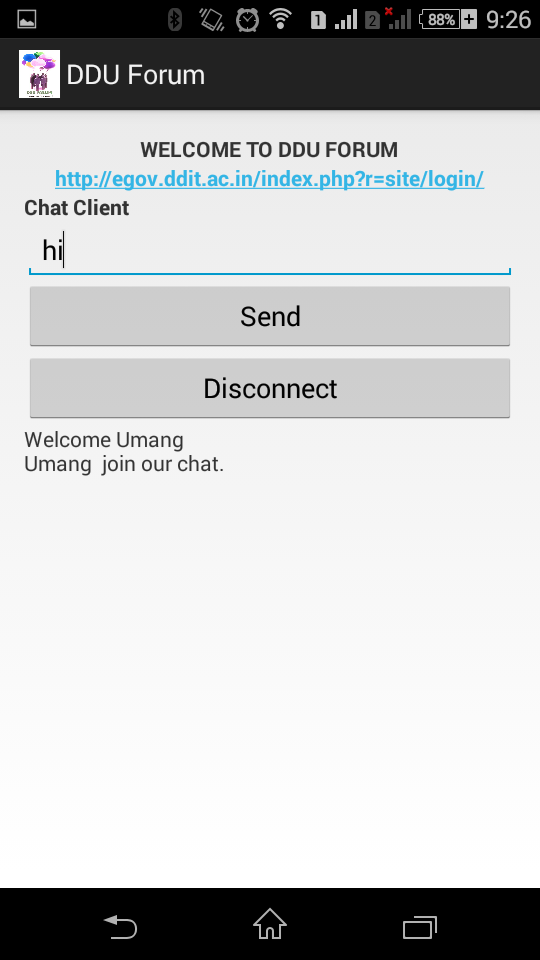
**Step 4: Server accept the connection and client is ready for chatting.**

****

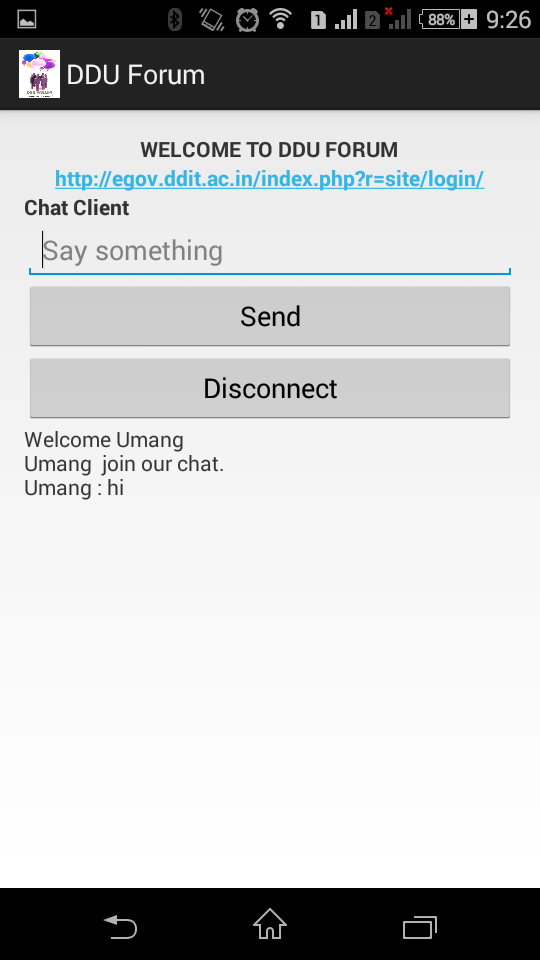
**Step 5: Client can send the message and disconnect the connection and also can go to student panel site of ddu.**

****

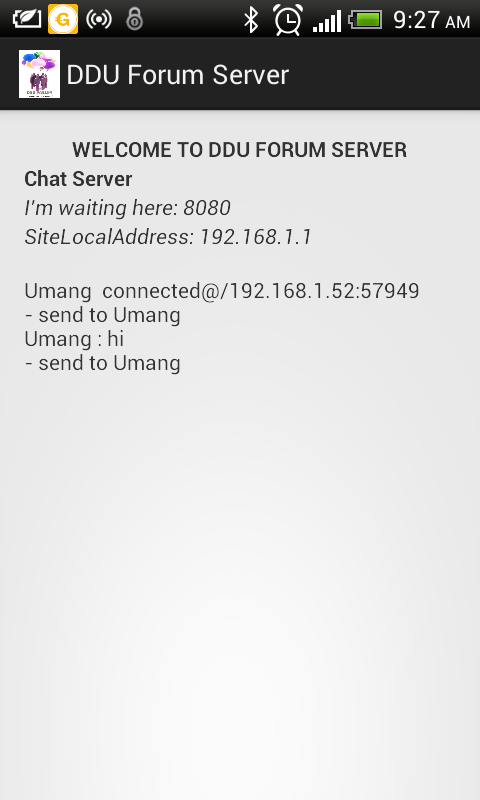
**Step 6: Client write the message and send it to another clients via server.**

****

**Step 7: client sent message to another clients and it received.**

****

**Step 8: server can monitor all messages and make log of it.**

****

**CHAPTER 9**

LIMITATION & FUTURE

ENHANCEMENT

**9.1 LIMITATION**

1. Peer to peer complete and secure connection need to establish for personal chat feature.

2. Every group member can share files and download share files in group chat.

3. Smart phone maintain separate folder for my application data.

4. Shared data quality must be remaining as it is like original data.

5. On the client screen instead of displaying link it use hyperlink tag to redirect.

**9.2 FUTURE ENHANCEMENT**

1. A feature for personal chat is also welcomed but with safety constraints of its own.

2. File sharing must be included in future.

3. GUI must be improved in future.

4. We give more security to messages.

5. Authentication of clients can make more secure.

6. Every client can set profile pic.

7. My application is efficient in RAM usage.

8. My application’s startup and loading must be as fast as possible.

9. Clients feedback is very important for us so we do changes on base of that in future.

**CHAPTER 10**

CONCLUSION AND DISCUSSION

**10.1 CONCLUSION AND FUTURE ENHANCEMENT**

In the end we concluded with these requirement cards.

1. Central System needed to showcase offers.

2. Malls do have Wi-Fi but cannot commit any help from mall owners.

3. Need an application to show offers to thousands of people without temporal loss of data.

4. Need of decentralized database for message storage.

a) This was particularly required because since our application will not be requiring an internet connection.

5. Need of a premium user who can permanently post advertisements.

a) There will need to be some mechanism by which some advertisements should be shown permanently since these would be valid for the whole day.

b) These advertisements should not get lost in the large data exchange in various tabs.

c) Hence a new tab is needed where advertisement will be posted and will be shown highlighted during the whole part of the day.

d) To avoid misuse of this feature, we will need premium users who will have to pay to use this feature.

e) It is evident from the interviews, that they are willing to pay some small amount to become premium users.

6. Need for anonymity of users.

7. Maximum users use Android 4.0 or above. Hence priority would be to make an application compatible with android 4.0 or above.8. The feature of offers can decide upon what visitors buy in a mall since a majority of them do not have a predicated wish-list.

9. A feature for personal chat is also welcomed but with safety constraints of its own

**10.2 DISCUSSION**

**10.2.1 Self-Analysis of Project Viabilities**

According to me, this projected is completed with the primary functionalities as specified earlier but then again there is lot more than this which can be done. The project is well capable to handle the given job for some particular task but not all of them. So then it is a challenge to further develop it in to well flagged software as it was challenge to develop up to this very stage.

**10.2.2 Problem Encountered and Possible Solutions**

There were many problems encountered during the design and the development phase of the project.

The problem in identifying MAC address of each client and send message to particular client.

The problem to maintain a MAC address list in server.

**10.2.3 Summary of Project work**

I have completed my project work using software engineering and system

Analysis and design approach. I have done work with preplanned scheduling related with time constraints and result oriented progress in project development.

**REFERENCES**

1. www.androidhive.com

2. www.vogella.com

3. www.stackoverflow.com

4. developer.android.com

5. www.wikipedia.com

6. www.youtube.com

7. www.lynda.com

8. www.thenewboston.com