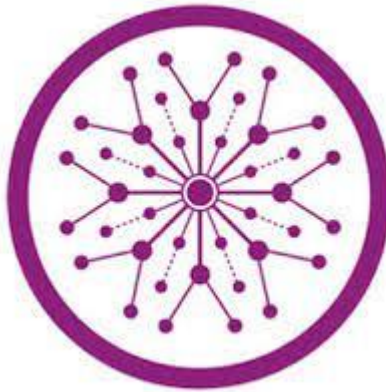


VIBE.LI (Messaging app)

Final Year Project

Session 2020-2022

A project submitted in partial fulfillment of the degree of
Master in Information Technology



Department of Information Technology
Faculty of Computer Science & Information Technology
The Superior University Lahore

Fall 2021

Development	[<input checked="" type="checkbox"/>] Development [<input type="checkbox"/>] Research [<input type="checkbox"/>] R&D			
Area of specialization	Android App			
FYP ID	MITM-FYP-S21-006			
Project Group Members				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	Mitm-s20-001	Hamza Abbas	Mitm-s20-001@superior.edu.pk	
(ii)	Mitm-s20-004	Haniya Fatima	Mitm-s20-004@superior.edu.pk	
(iii)	Mitm-s20-006	Qurat-ul-ain	Mitm-s20-006@superior.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I **Haniya Fatima** S/D of **Tariq Mehmood**, group leader of FYP under registration no **Mitm-s20-004** at Information Technology Department, The Superior University Lahore. I declare that my FYP report is checked by my supervisor.

Date: _____ Name of Group Leader: Haniya Fatima Signature: _____

Name of Supervisor: Rohail Shahzad

Designation: Lecturer

Signature: _____

HoD: Dr. Asad Ali Naqvi

Signature: _____

Project Report

VIBE. Li (Messaging app)

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature
Hamza Abbas Haniya Fatima Qurat Uzair	1.0	29/03/2021	Original Draft	
Hamza Abbas Haniya Fatima Qurat Uzair	2.0	1/04/2021	Add explanation	
Hamza Abbas Haniya Fatima Qurat Uzair	3.0	3/04/2021	Correct Gantt chart	
Hamza Abbas Haniya Fatima Qurat Uzair	4.0	21/04/2021	Use case	
Hamza Abbas Haniya Fatima Qurat Uzair	5.0	12/05/2021	Template discussion	
Hamza Abbas Haniya Fatima Qurat Uzair	6.0	29/05/21	Project presentation	
Hamza Abbas Haniya Fatima Qurat Uzair	7.0	22/09/21	Properties add check	
Hamza Abbas Haniya Fatima Qurat Uzair	8.0	10/01/22	Project discussion	
Hamza Abbas Haniya Fatima Qurat Uzair	9.0	13/01/22	Project review	
Hamza Abbas Haniya Fatima Qurat Uzair	10.0	14/01//22	Final Presentation	

APPROVAL

PROJECT SUPERVISOR

Comments: _____

Name: _____

Date: _____ Signature: _____

PROJECT MANAGER

Comments: _____

Date: _____ Signature: _____

HEAD OF THE DEPARTMENT

Comments: _____

Date: _____ Signature: _____

Dedication

*This work is dedicated to my **MOTHER.***

Acknowledgement

In the name of **ALLAH**, the most gracious and merciful.

All praises to HIM who is the most benevolent, most caring n selfless one who enabled me and gave courage to me to complete this project. All praises to ALLAH ALMIGHTY who gave me the strength, stamina, courage n ability to do so Indeed, HIS blessings are countless.

The outcome of this project required a lot of guidance and assistance from many people, and I am extremely fortunate to have got it all in completion of my project. Whatever I have done is due to such guidance and assistance.

So, I express my heartfelt gratitude to my supervisor **Sir Rohail Shehzad** for rendering help and support during this project.

Executive Summary

Vibe.li is a mobile-phone-based software program that allows users to send and receive information using their phone's internet connection. It can send or receive a much wider range of data types than Short Message Service (SMS) or Multimedia Messaging Service (MMS). User can send receive not only texts but also pictures, audios etc.

Its interface is user friendly so it will be easy to use for everyone who have access to mobile phones. They can send and receive messages and data with the access of internet.

Table of Contents

Dedication.....	v
Acknowledgement	vi
Executive Summary	vii
Table of Contents	viii
List of Figures	x
List of Tables	xi
Chapter 1	1
Introduction	1
.1. Motivations and Challenges	2
.2. Goals and Objectives	2
.3. Literature Review/Existing Solutions	3
.4. Gap Analysis.....	3
.5. Proposed Solution	3
.6. Project Plan.....	4
.6.1. Work Breakdown Structure	4
.6.2. Gantt Chart.....	4
.7. Report Outline	4
Chapter 2	7
Software Requirement Specifications	7
.1. Introduction	8
.1.1. Purpose	8
.1.2. Document Conventions	8
.1.3. Intended Audience and Reading Suggestions	8
.1.4. Product Scope.....	9
.1.5. References.....	9
.2. Overall Description	9
.2.1. Product Perspective	9
.2.2. Product Functions.....	9
.2.3. Design and Implementation Constraints.....	10
.2.4. User Documentation.....	10
.1.1. Assumptions and Dependencies.....	10
.2. External Interface Requirements	11
.2.1. User Interfaces	11
In user interface these GUI Interfaces that are used by end user are:	11
• Splash screen.....	11
• Sign up.....	11
• Login Page	11
• Chats layout.....	11
• Groups layout.....	11
• Profile	11
.2.2. Hardware Interfaces.....	11
.2.3. Software Interfaces	11
.3. System Features	11
• Functional Requirements	13

4. Other Nonfunctional Requirements	14
Chapter 3	15
Use Case Analysis.....	15
3.1. Use Case Model.....	16
Chapter 4	20
System Design	20
4.1. Architecture Diagram.....	21
4.2. Domain Model.....	22
4.3. Entity Relationship Diagram with data dictionary	23
4.4. Class Diagram.....	24
4.5. Sequence / Collaboration Diagram	25
4.6. Operation contracts	26
4.7. Activity Diagram	27
4.8. State Transition Diagram.....	28
4.9. Component Diagram	29
4.10. Data Flow diagram [<i>only if structured approach is used - Level 0 and 1</i>].....	30
Chapter 5	31
Implementation	31
5.1. Important Flow Control/Pseudo codes.....	32
5.2. Components, Libraries, Web Services, and stubs.....	32
5.3. Deployment Environment	33
5.4. Tools and Techniques	33
5.5. Best Practices / Coding Standards	33
5.6. Version Control	33
Chapter 6	34
Testing and Evaluation	34
6.1. Use Case Testing.....	35
Login:.....	35
6.2. Boundary value analysis.....	36
6.3. Data flow testing	37
6.4. Unit testing.....	38
6.5. Performance testing.....	40
Chapter 7	41
Summary, Conclusion and Future Enhancements	41
7.1. Project Summary	42
7.2. Achievements and Improvements.....	42
7.3. Critical Review	42
7.4. Lessons Learnt	43
7.5. Future Enhancements/Recommendations.....	43
Appendices	44
Appendix A:	45
Reference and Bibliography	53
Index.....	55

List of Figures

1.1	Caption of first figure of first chapter	6
1.2	Caption of second figure of first chapter	7
2.1	Caption of first figure of second chapter	14
2.2	Caption of second figure of second chapter	22
2.3	Caption of third figure of second chapter	26
5.1	Caption of first figure of fifth chapter	49
5.2	Caption of second figure of fifth chapter	49

List of Tables

1.1	label of first table of first chapter	6
1.2	label of second table of first chapter	7
2.1	label of first table of second chapter	14
2.2	label of second table of second chapter	22
2.3	label of third table of second chapter	26
5.1	label of first table of fifth chapter	49
5.2	label of second table of fifth chapter	49

Chapter 1

Introduction

Chapter 1: Introduction

Our project is about a messaging app named as **vibe.li**. It is a mobile-phone-based software program that allows users to send and receive information using their phone's internet connection. It can send or receive a much wider range of data types than Short Message Service (SMS) or Multimedia Messaging Service (MMS). User can send and receive data through vibe.li. User's must have access to internet so, they can use all the features of vibe.li. Except that they don't need to pay anything for it.

Its interface will be user friendly as our main target is our country and Pakistan has lower literacy rate. INSHALLAH everyone can use vibe.li for their ease n convenience.

.1. Motivations and Challenges

Motivation:

- Our motivation is to provide our people the best app for their ease and convenience.
- We are hoping to produce our very own App for our country.

Challenges:

- One-to-one messaging is still very popular but difficult to achieve.
- Still, it will be difficult to switch users from What's App or other apps to vibe.li.
- It is difficult to win the trust of people as well.

.2. Goals and Objectives

Our goal is to make our app more friendly and easy for people so it can become easier for them to communicate each other. Vibe.li is for two or more people to communicate through the Internet connection and through text messages. It has all the benefits of an enhanced chat system with the facilities to share all kinds of documents. Knowing this and thinking in a more professional field, we can talk about it as a pretty good tool for members of a team or different teams to communicate with each other to coordinate when working, to share work documents

or any file needed. In addition, it can also serve as a contact channel for consumers who need to speak with a brand by doubts or any other reason.

Objectives:

- Awareness
- Productivity
- Communication
- Loyalty

.3. Literature Review/Existing Solutions

There are many messaging or communication apps in market right now and they are also working very well. Millions of people rely on them for communication purpose because they find them very effective to use. But still there are many things we can overcome to develop a better app like:

- Make it easier to understand.
- Easier to use.
- Effective for communicating.
- Data will be secured.

.4. Gap Analysis

It has much simple features than WhatsApp. We'll be developing it as user-friendly and simple app for our people.

.5. Proposed Solution

We are providing our users with a user-friendly mobile phone-based app for better communication, and we'll work on making it easier to use. We will try to provide them with better and more features than many other apps. So, they don't have to waste their time as well as credit. They just must have the access to internet, and they can use all the features like send and receive one-to-one or group messages, pictures, audios etc. They can communicate with everyone who have vibe.li.

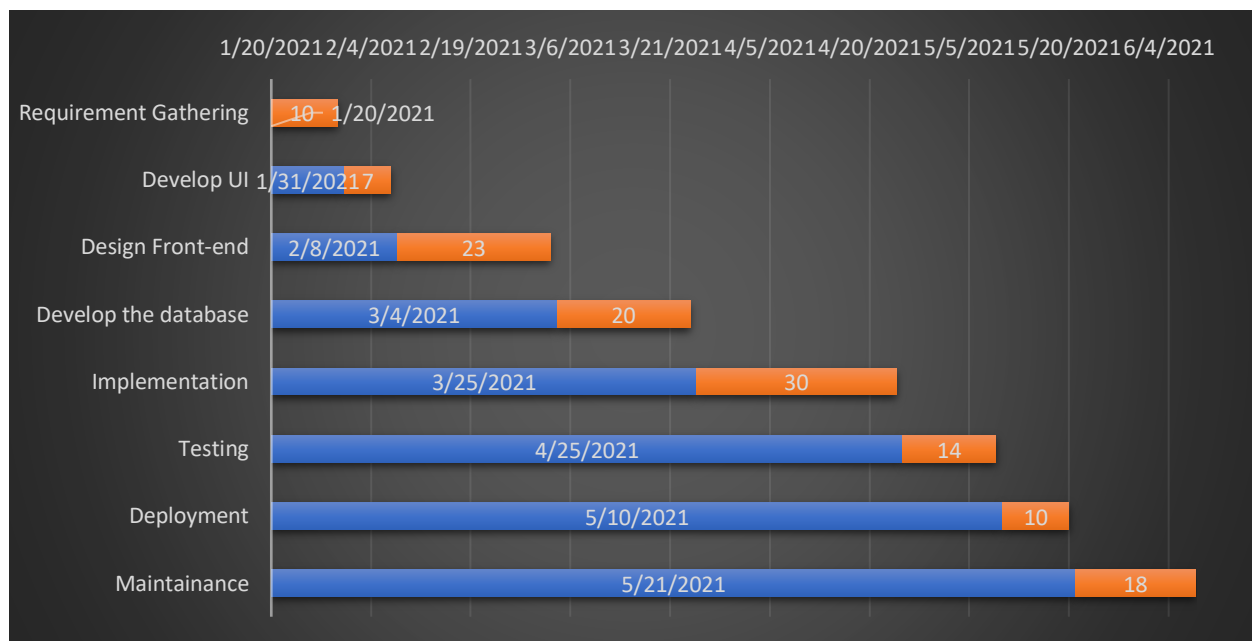
.6. Project Plan

- Draft the idea
- Research and list out what to do
- Graphic Design
- Promoting the idea
- Then final step build app

.6.1. Work Breakdown Structure

- First, we'll do documentation of our complete project.
- We'll pre-plan or draw our project with the help of diagrams like use case, class diagram DFD, entity relationship etc.
- After that we'll be working on development of our app.

.6.2. Gantt Chart



.7. Report Outline

- Introduction
- Background
- Motivations and challenges

- Goals and Objectives
- Literature Review/Existing Solution
- Gap Analysis
- Proposed Solution
- **Project Plan**
 - Planning
 - Requirement gathering
 - Finding functional and non-functional requirement
 - Choosing tools and techniques
 - Choosing architecture
- **People**
 1. Project sponsor
 2. Project manager
 3. Project team
 4. End user

Others

- **Designing**
 - Front end
 - Back end
- **Implementation**
 - Coding front end
 - Coding back end

- Testing
- Maintenance
- Work Breakdown Structure
- Roles & Responsibility Matrix
- Gantt chart

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

.1. Introduction

.1.1. Purpose

The main purpose of vibe.li is to send and receive messages, documents, photos, and videos. Vibe.li uses your phone's cellular or Wi-Fi connection to facilitate messaging and multimedia to nearly anyone on the planet, alone or in a group, and is especially nice for families and small collaborative workgroups.

.1.2. Document Conventions

This document follows Calibri font and font size 12 pt. Bold-faced text is used to emphasize section and sub-section headings. Highlighting is to point out words in the glossary.

.1.3. Intended Audience and Reading Suggestions

We are making this document by keeping in mind different types of readers. This document will be useful for different audience in various ways.

Audience	Use
Developers	They will use this document for guidance for designing and implementing phase.
Managers	They will have a look on the constraints all cover properly and time and cost are within limits or not.
Marketing Staff	They can use this document to make advertisements for this marketplace because by reading this document they will know what the app will do? How this app is different from others.

Clients	They can ensure their self by reading the SRS that they met their needs in marketplace or not.
Testers	They will test the implementation of the project according to the SRS.
Documentation Writer	They will use this document during the documentation of the project. It will be helpful for them.

.1.4. Product Scope

The scope of this app is to provide a platform for multiple multimedia sharing. Basically, it is an online instant messaging application that integrates user's contacts from their phonebook with the application and enables exchange of messages, sharing of multimedia free of cost over the internet. The aim of this app is to promote an efficient, user-friendly, time-consuming, safe way for clients for getting their work done without any difficulty.

.1.5. References

.2. Overall Description

.2.1. Product Perspective

To keep in view all reader we simply describe app which is used for multiple multimedia sharing.

.2.2. Product Functions

This section provides requirement overview of the system. Various functional modules

- **Registration:** Login with phone number.
- **Interface:** first splash screen appears and then login screen appears if and then signup (in case user id new).

- **Verification:** Needs to verify your phone number.
- **Chat:** one to one chat is available.
- **Group Chat:** This app also provides group chat for multiple people to chat at the same time.
- **Profile picture:** User can upload their profile picture.
- **Bio:** User can update their bio with whatever they want to.

.2.3. Design and Implementation Constraints

- Major thing in this project is time
- Default Language will be English
- App will follow all the copyright and cyber laws of PTA.

.2.4. User Documentation

- We will give the complete documentation of the project.

.1.1. Assumptions and Dependencies

- Users know the English language, as the interface will be provided in English.
- Languages used to develop it are XML, Kotlin etc.

.2. External Interface Requirements

.2.1. User Interfaces

In user interface these GUI Interfaces that are used by end user are:

- Splash screen
- Sign up
- Login Page
- Verification
- Chats layout
- Groups layout
- Search layout
- Profile

.2.2. Hardware Interfaces

The System must run over the Internet, all hardware shall require connection to internet will be hardware interface for system e.g., Modem

.2.3. Software Interfaces

- **Tools:**
Android Studio
- **Languages:**
Xml, Kotlin

.3. System Features

In system features, we will use the use-cases of different pages i.e., is sign-in, sign-up, login etc.

2.4 Use Case Table








Level 0	Level 1	Level 2	Actor
Chat Application	Authentication System	Registrar Login Logout	  User Admin
	Contacts Form	Friend List Find Friend Add Friend Remove Friend Block Friend	 User
	Chat Form	Send Message Group Chat Best Friend	 User
	Maintenance	User's Profile Database	 Admin
	Monitor	Check History Feedback	  Admin User

Figure 1: Use Case Table of Chat Application

- **Functional Requirements**

User Registration

User must be able to register for the application through his contact number. On installing the application, user must be prompted to register through phone number. If user skips this step, application should close. The user's id will be the unique identifier of his/her account on vibe.ly.

Adding New Contacts

The application should detect all contacts from the user's phone book. If any of the contacts have user accounts with vibe.ly, those contacts must automatically be added to the user's contact list on vibe.ly.

Send Message

User should be able to send instant message to any contact on Vibe.ly contact list. User should be notified when message is successfully delivered to the recipient by displaying a sent sign next to the message sent.

Send Attachments

User should be able to send audio (voice notes) and images as attachments.

Audio formats that the application should support: mp3

Image formats that the application should support jpg png

Broadcast Message

User should be able to create groups of contacts. User should be able to broadcast messages to these groups.

Message Status

User must be able to get information on whether the message sent to the intended recipient.

.4. Other Nonfunctional Requirements

Scalability

Vibe.ly should be able to provide instant messaging services to many users at any given time.

Performance

Application must be lightweight and must send messages instantly.

Security requirements:

The system will use database to store data.

Users can only send and receive images, messages, voice notes etc. But they cannot modify anything then their personal profile.

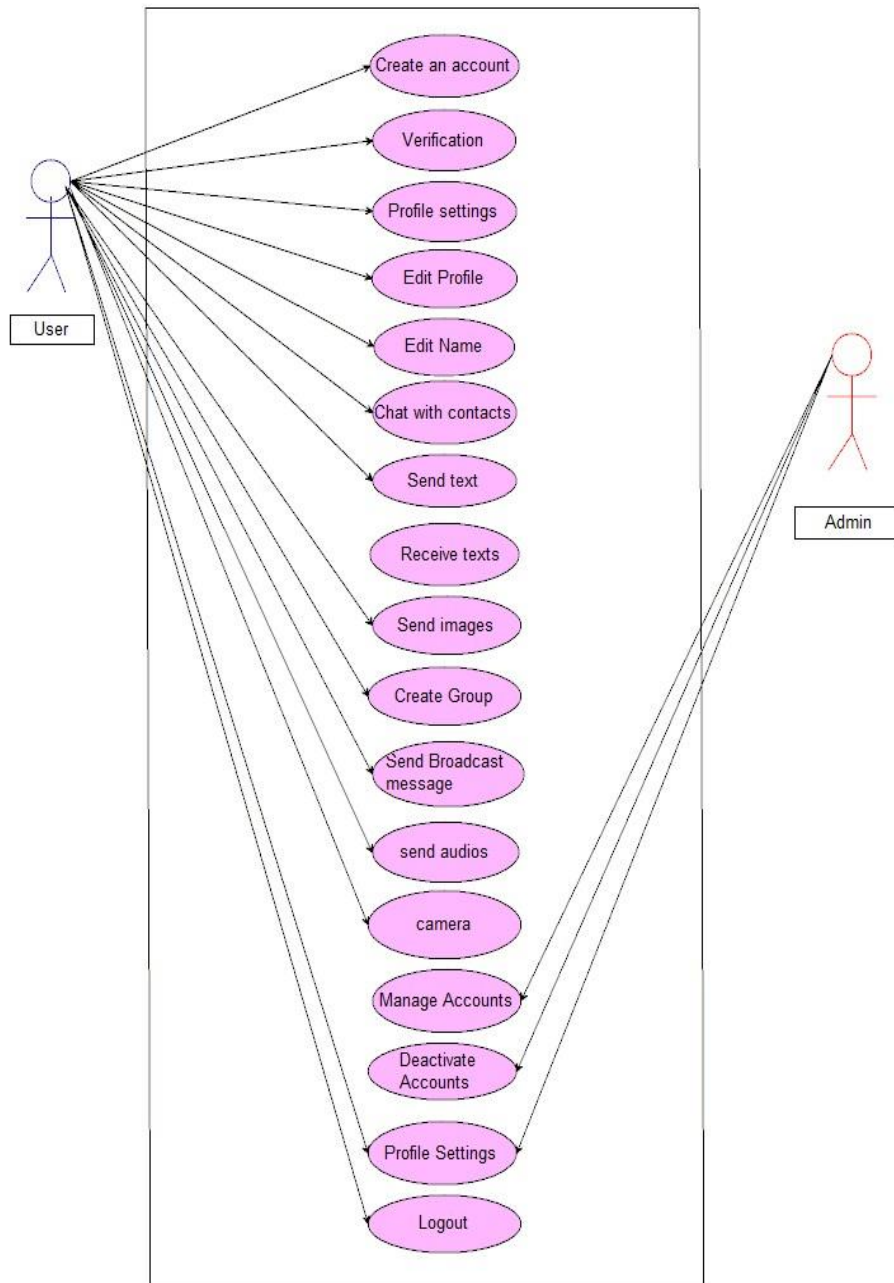
Chapter 3

Use Case Analysis

Chapter 3: System Analysis

In this chapter we will analyze the system with the help of use case

3.1. Use Case Model



1. Use Case: login

Primary Actors: User and Admin

Goal in context: To login a new account

Precondition:

User should have an account

Triggers: User and Admin needs to perform their actions

Scenario:

First splash screen appears and then login

Enter authentic username and password

Click on login.

Exception: Username or password can be wrong or forgotten

2. Use Case: Create a new account

Primary Actors: User and Admin

Goal in context: To create a new account

Precondition:

User should have an account

Triggers: User need to create an account

Scenario:

Enter authentic username and password

Click on login.

Exception: Username or password can be wrong or forgotten

3. Use Case: profile picture setting

Primary Actors: User

Goal in context: To update profile picture

Triggers: User need to create an account

4. Use Case: Chats with contacts

Primary Actors: User

Goal in context: To chat with existing contact

Precondition:

User should have an account

Triggers: User need to create an account

Scenario:

Enter authentic username and password

Click on login.

5. Use Case: Create group

Primary Actors: User

Goal in context: To create a group for group chat for multiple people to chat at a same time.

Precondition:

User should have an account

Scenario:

Create a group and add multiple people to chat at the same time

6. Use Case: profile settings

Primary Actors: User

Precondition:

User should have an account

Triggers: User need to change profile.

Scenario:

Any change in profile

7. Use Case: logout

Primary Actors: User

Goal in context: To create a new account

Precondition:

User should have an account

Triggers: User need to create a new account

Scenario:

When client to login with anew account

Then he/she logout first

Then login with new contact number

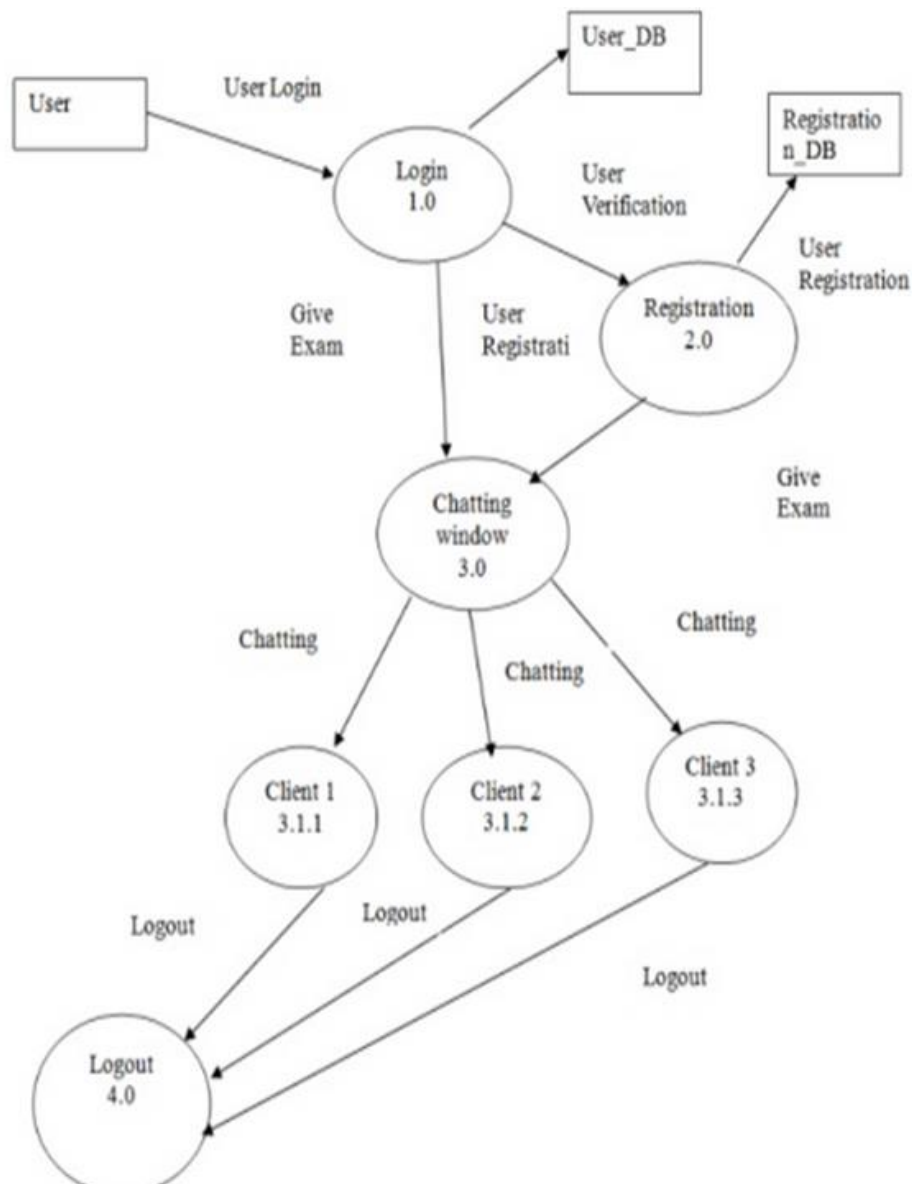
Chapter 4

System Design

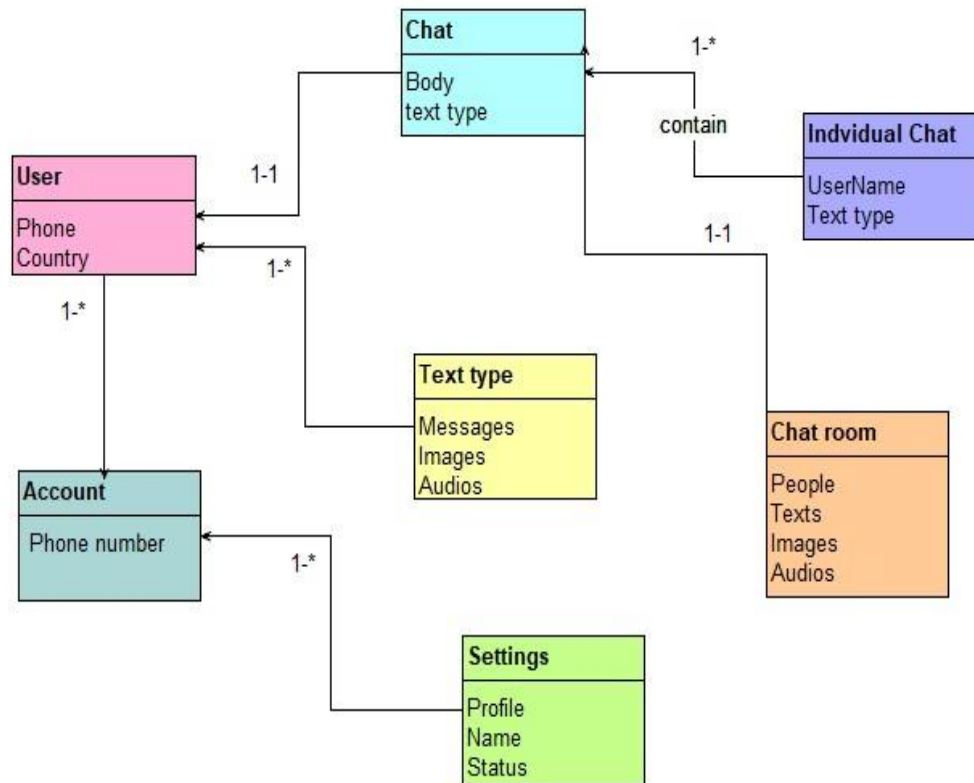
Chapter 4: System Design

This chapter is about the design of our app. It will include different type of diagrams. System diagram shows how the system interacts with its module to the user and a developer. Team can easily identify an error in the system before the deploy to the end user.

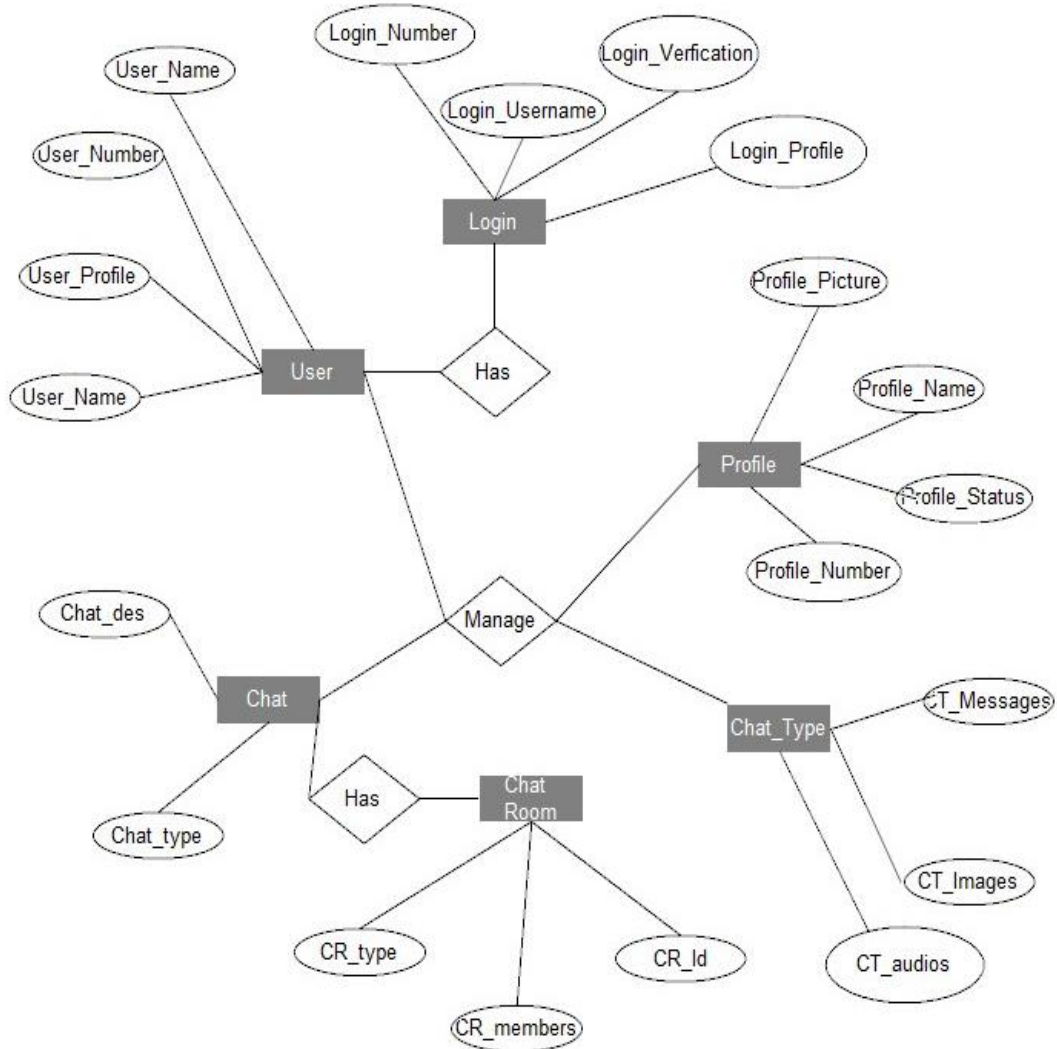
4.1. Architecture Diagram



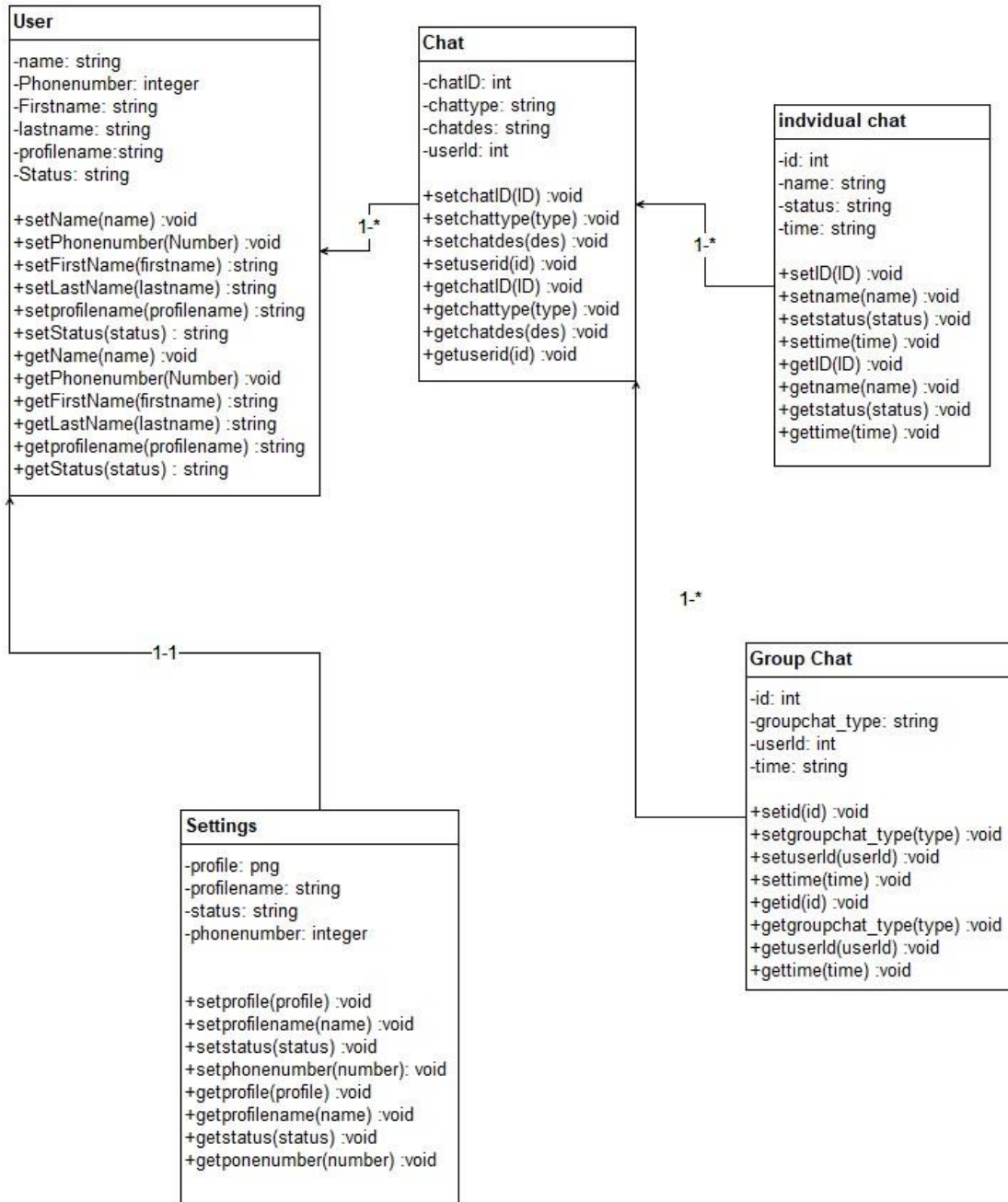
4.2. Domain Model



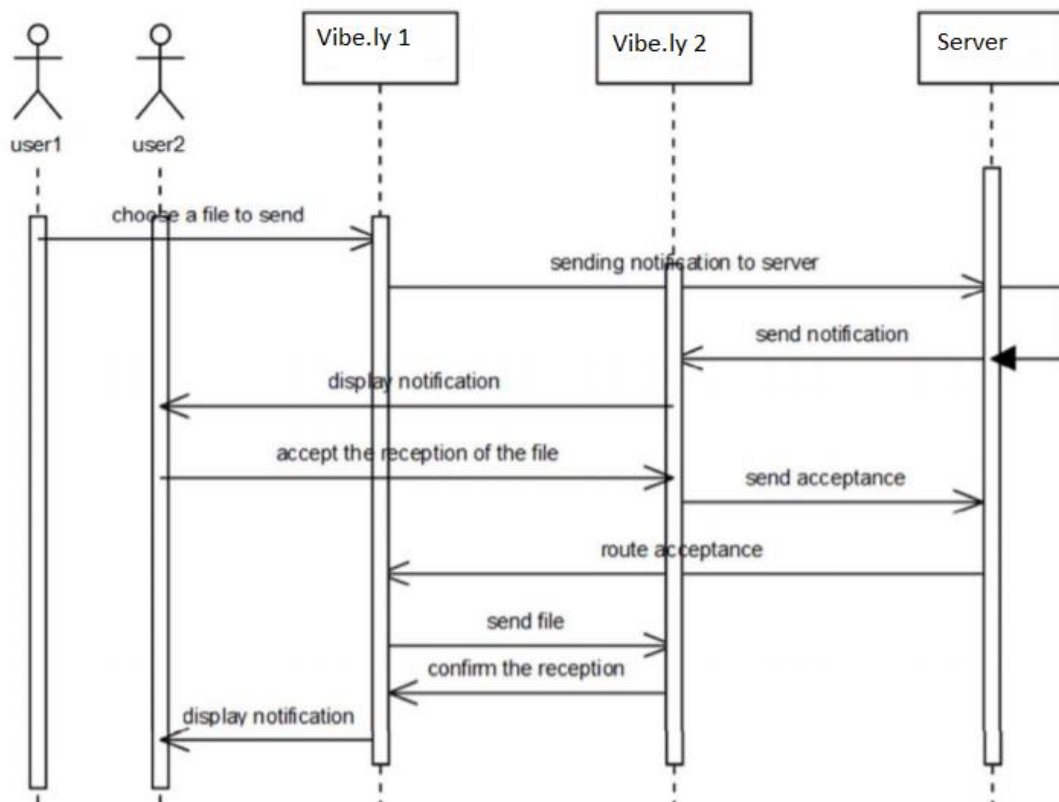
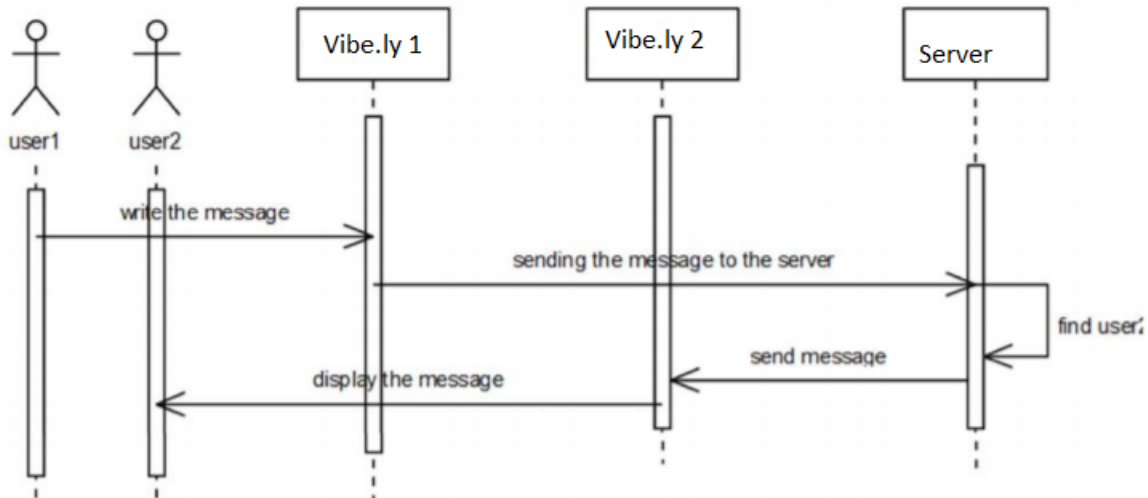
4.3. Entity Relationship Diagram with data dictionary



4.4. Class Diagram



4.5. Sequence / Collaboration Diagram



4.6. Operation contracts

Project-based agreements can vary depending on the parameters of the work and how much of it has already been planned. Common types of agreements include:

- **Fixed fee contract:** When the company and contractor agree on a fee for the entire project.
- **Cost-plus contract:** When the contract includes payments for labor and materials, as well as a percentage for overheads.
- **Incentive-based contract:** When payments are tied to the project achieving specified goals.

We have applied incentive-based contract for developing vibe.li.

OC1: Login

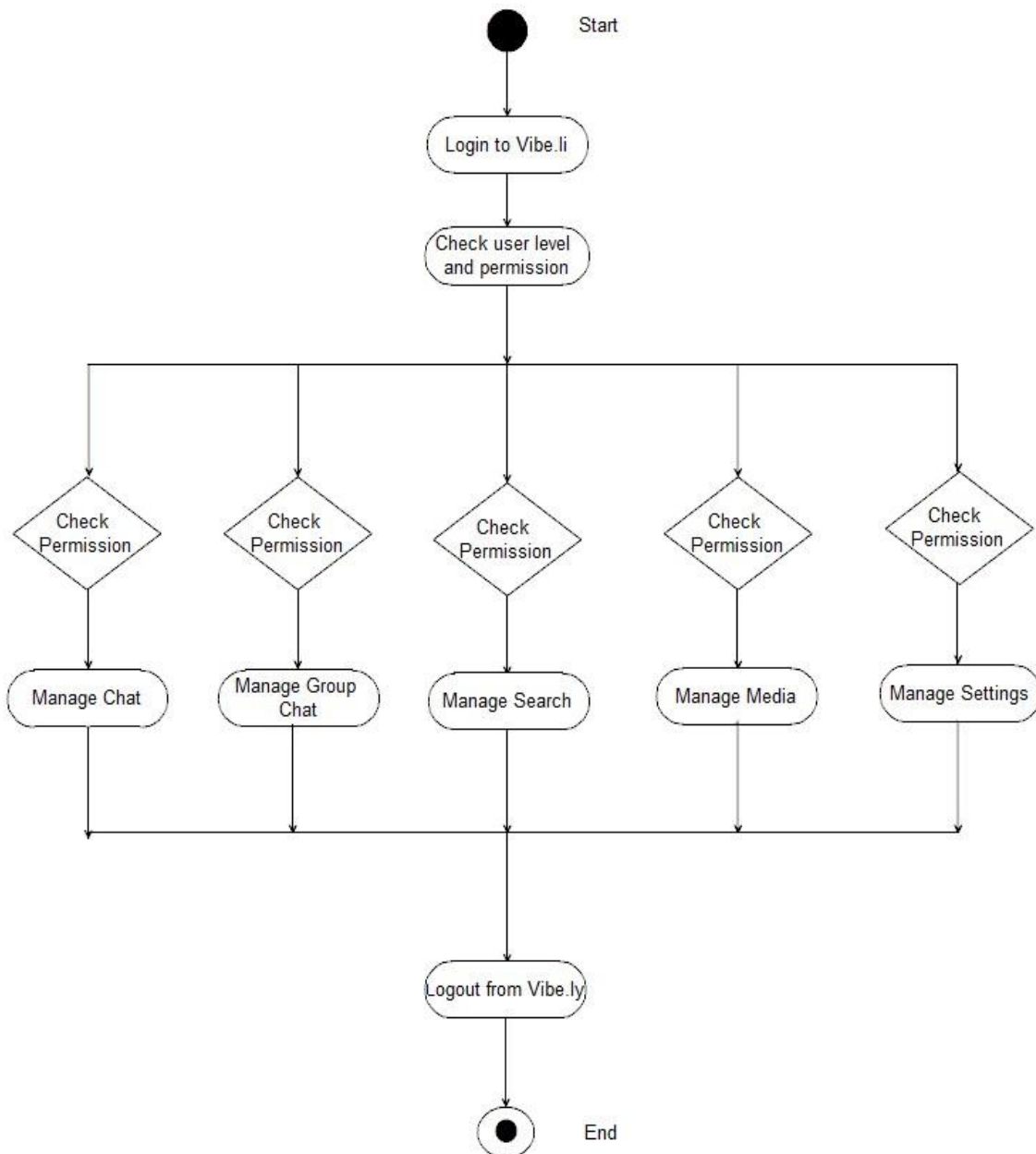
Operation: login (phone number, Verification)

Cross References: Use Cases: System

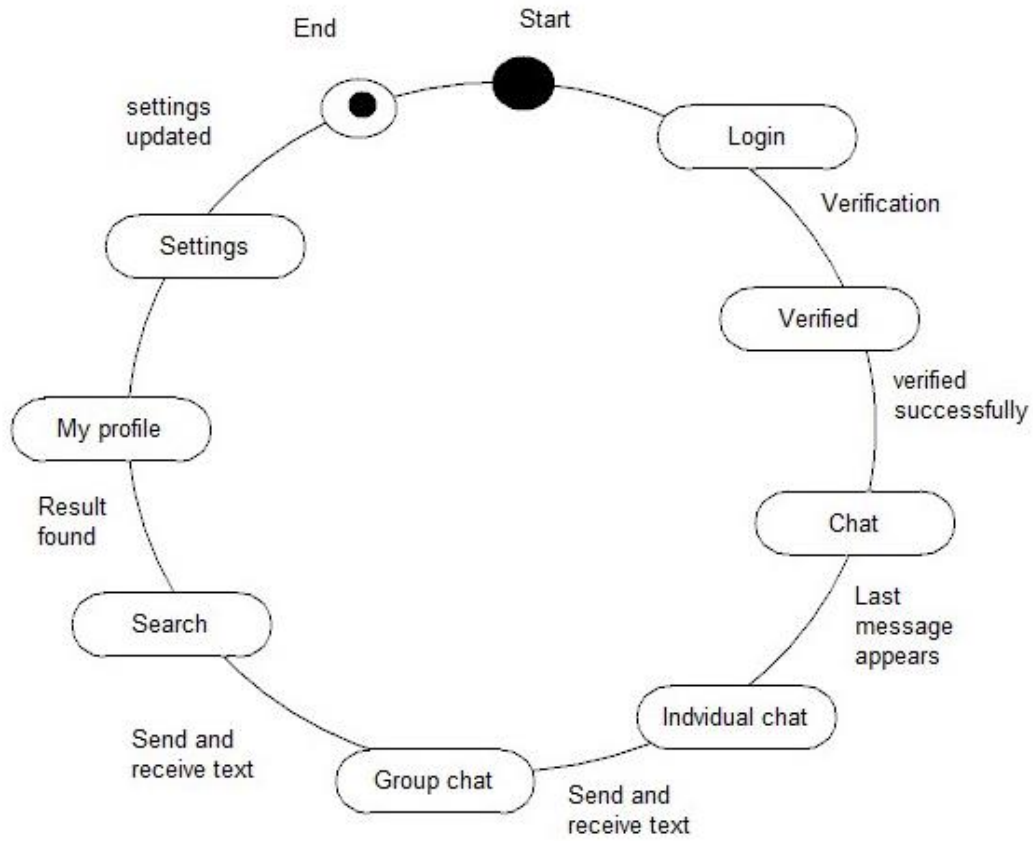
Preconditions: User is not logged in

Post Conditions: A user logged in and got more access

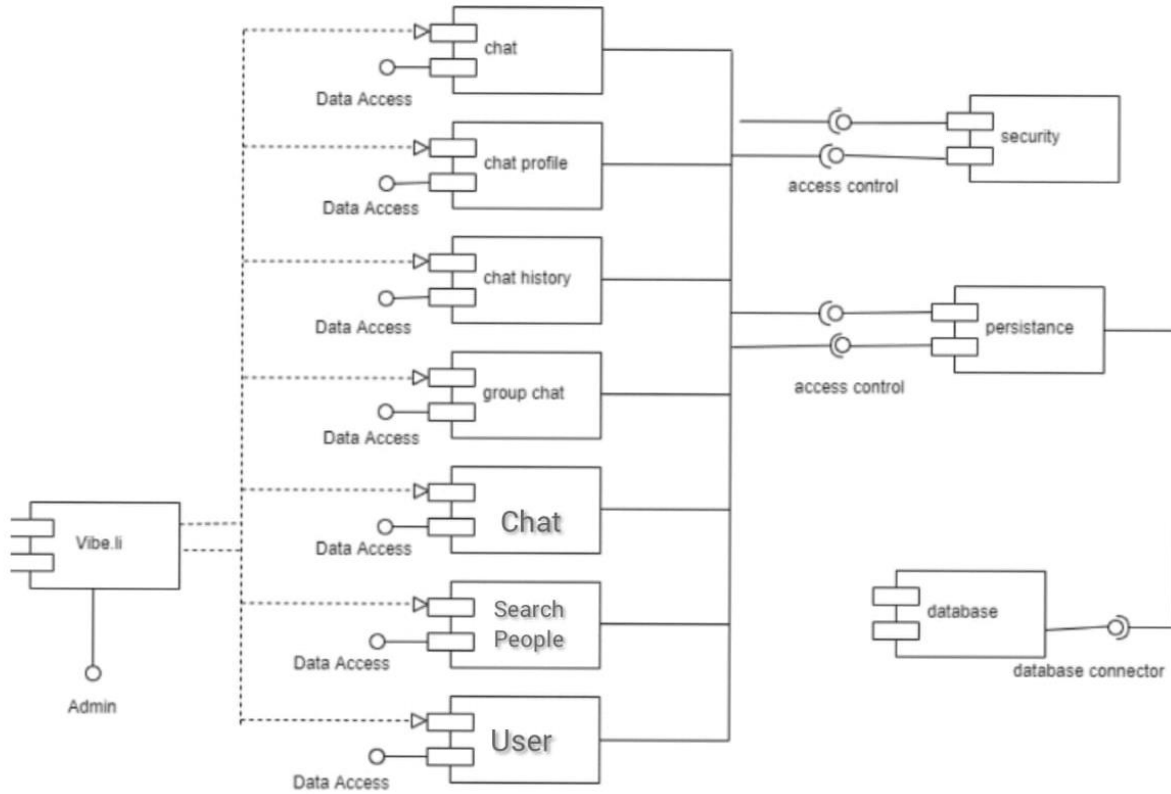
4.7. Activity Diagram



4.8. State Transition Diagram

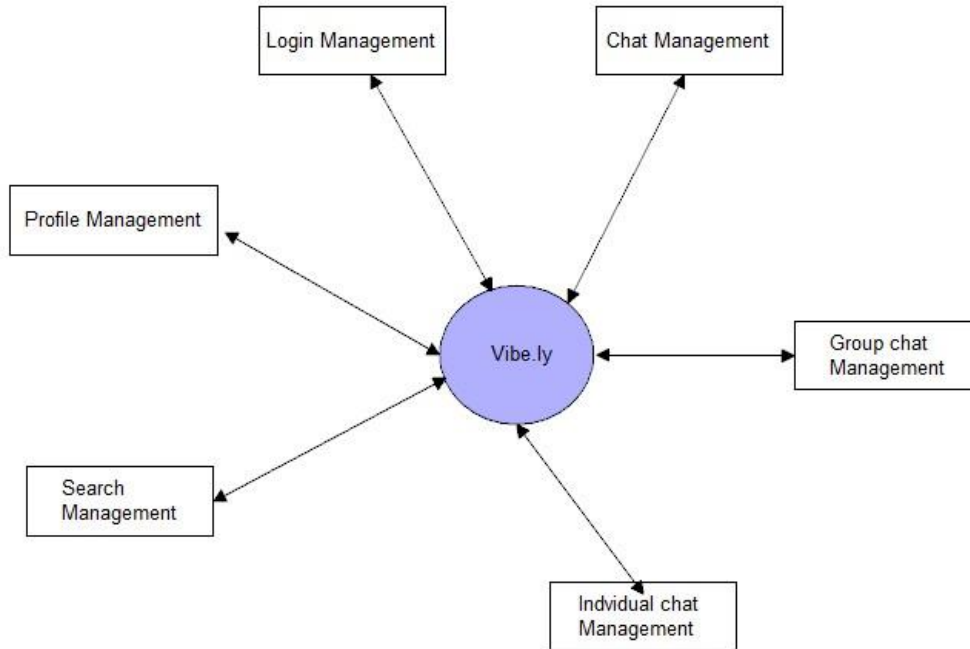


4.9. Component Diagram

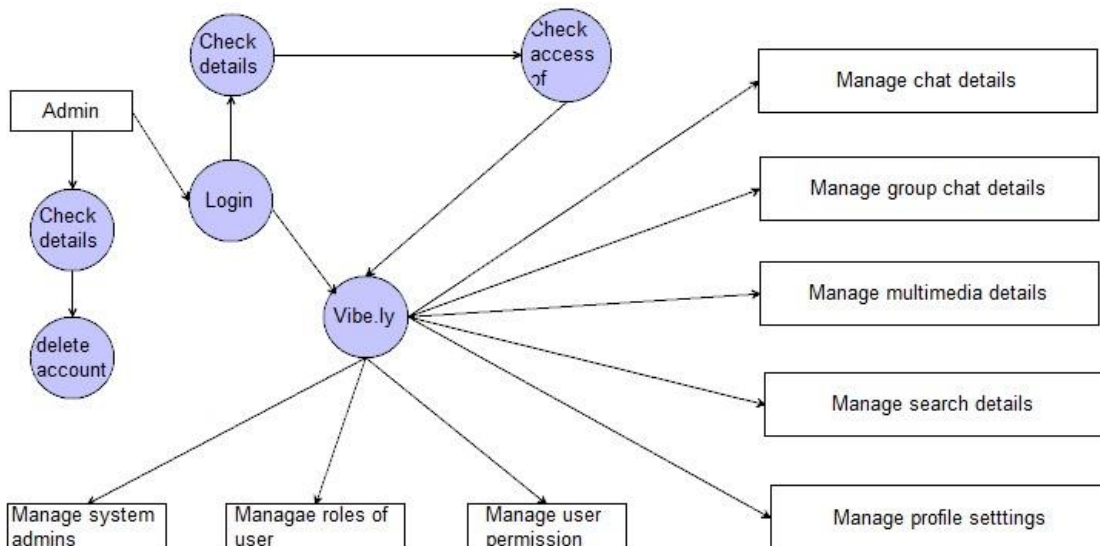


4.10. Data Flow diagram [only if structured approach is used - Level 0 and 1]

Level 0:



Level 1:



Chapter 5

Implementation

Chapter 5: Implementation

This chapter is all about implementation. Programming languages for layout design xml for coding Kotlin and for backend firebase. We will make sure to implement each and everything.

Will be implementing:

- SRS
- Requirements
- Tools
- Techniques
- Testing's
- Class Diagrams
- Activity Diagrams
- Sequence
- Architecture Design

5.1. Important Flow Control/Pseudo codes

All data store in the database.

User/Customer:

User/Customer can access the functionality of the system, like access to sending and receiving messages, voice notes, edit profile etc.

Administration:

Administration is being able to handle information about users and data of users. They check the accounts of users.

5.2. Components, Libraries, Web Services, and stubs

We have used android studio and for front-end we have used xml and for backend kotlin. So, we will need all the libraries needed for these frameworks.

Services:

- 24-hour access to users

- Efficient
- Fast
- Less error in sending and receiving texts
- Consistency

5.3. Deployment Environment

We can deploy our app in any android phone like Samsung, Huawei, infinix, vivo, oppo etc.

It can be deployed in any model of any android phone.

5.4. Tools and Techniques

Our technical team master a comprehensive set of tools and technologies during project development like:

- Android studio
- Firebase
- Kotlin

5.5. Best Practices / Coding Standards

- We have use android studio for coding standards.
- Use full English descriptors that accurately describe the variable.

5.6. Version Control

For version Control of Online using Git/GitHub, Android will be best. During development, a source control management system was utilized to keep track on changes made to the source code over time up until completion. We individually complete our tasks in our separate machines and the finally joined all this and build Vibe.ly in a one single machine in which our complete project is up and running.

Chapter 6

Testing and Evaluation

Chapter 6: Testing and Evaluation

In this chapter we will briefly describe all the phases of system testing and in the end, we will evaluate the developed system according to the result obtained from the testing that will determine how much the system is useable and what are the faults and bugs that needs to be fixing before the deployment of the system.

6.1. Use Case Testing

Login:

<u>Headings</u>	<u>Sub-Headings</u>	
Operations		Login()
Cross-Reference	Use-Case	Login, View Profile
	Scenario	Login
Pre-Conditions		N/A
Post-Conditions	Instantiation	An instance "a" of Account was created. An instance "P" of Profile was created.
	Attribute Modification	A.Id becomes ID.
	Association	A was associated with Account. A was associated with the Profile.

Search:

<u>Headings</u>	<u>Sub-Headings</u>	
Operations		Search ()
Cross-Reference	Use-Case	View Friend
	Scenario	View Location
Pre-Conditions		The person should be logged into the system A person must select a name to be able to search for a friend
Post-Conditions	Instantiation	An instance "Search" of Search was created. An instance "S" of was created.
	Attribute Modification	N/A
	Association	A search was associated with the Friend.

6.2. Boundary value analysis

Boundary value analysis (BVA) is based on testing the boundary values of valid and invalid partitions. The Behavior at the edge of each equivalence partition is more likely to be incorrect than the behavior within the partition, so boundaries are an area where testing is likely to yield defects. Every partition has its maximum and minimum values, and these maximum and minimum values are the boundary values of a partition. Boundary value for a valid partition is a valid boundary value. Similarly, a boundary value for an invalid partition is an invalid boundary value. Tests can be designed to cover both valid and invalid boundary values. When designing test cases, a test for each boundary value is chosen.

6.3. Data flow testing

Modules no:	Module's description	Expected result	Actual result	Status
01	User Sign in	Validate data from server	Data validate	Pass
02	OTP verified	Through SMS	Data validate	Pass
03	Single Chat	Data fetch from the server	Data Fetched	Pass
04	Send Chat	Data must be stored in database	Data Stored	Pass
05	Group Chat	Data must be stored in data base	Data stored	Pass
06	Create Group	Data fetch from the server	Data Fetched	Pass
07	Send message in group	Data must be stored in data base	Data stored	Pass
08	Search Friend	Data fetch from the server	Data Fetched	Pass
09	Edit Profile	Data must be edited in data base	Data updated	Pass

6.4. Unit testing

App:

Test Case	Step Details	Test Data	Expected Result	Actual Result	Pass / Fail
Sign In	<ul style="list-style-type: none"> Open application Go to Sign Up Page Add details Verification Page will open 	Phone Number: 03234375938 Password: OTP	User will be registered successfully	User registered	Pass
Verification	<ul style="list-style-type: none"> Open application Go to login page. Enter phone number 	Phone Number: 03214981065 Password: OTP	User will be logged in successfully	User logged in	Pass
User can Search Friend	<ul style="list-style-type: none"> Open Application. Login Add details about what you want to search Click search. Results will appear if data exists 	Search Qurat = Qurat	Search query will be generated successfully.	Search results generated.	Pass
User can Send message	<ul style="list-style-type: none"> Open Application. Login Select Fsinglechat and send HI Appointment then confirm Appointment 	message Displayed = "HI"	Messages will be displayed.	Messages Displayed	Pass

--	--	--	--	--	--

User can View his Profile	<ul style="list-style-type: none"> • Open Application. • Login. • Click on Profile option 	Profile Displayed = "profile settings"	Profile will be displayed.	Profile Displayed	Pass
----------------------------------	--	--	----------------------------	-------------------	------

User can update his Profile	<ul style="list-style-type: none"> • Open application Login. • Click on profile option. 	<ul style="list-style-type: none"> • Click on Edit button • Select profile 	Profile Updated = "for user"	Profile will be updated successfully	Profile updated.	Pass
	<ul style="list-style-type: none"> • Table will open • Click on Edit Profile will be updated 					

6.5. Performance testing

- Maximum Response time is less 10 sec.
- Average response time 5 sec.

Constraints	Response Time
Application Startup	Less than 5 sec (Sometimes depend on internet Speed)
Data fetching	Less than 2 sec (Sometimes depend on internet Speed)
Application state update	Less than 1 sec
Transition in application	0.5 sec
Redirecting between inner pages of app	Less than 3 sec

Chapter 7

Summary, Conclusion and Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

Pakistan does not have any kind of chatting and messaging app of its own. People mostly use WhatsApp for communication purpose. Our country must not left behind in any mean so with vibe.li our people can communicate with anyone who have vibe.li. They can send and receive messages, images, voice notes and they'll also have access to camera means they can click pictures and send them.

7.2. Achievements and Improvements

Vibe.li is best communication app for people seeking a good, simple, and user-friendly app. Its interface is user-friendly, and it is good for people who are not highly educated. Users can join vibe.ly by creating account and just doing one step verification. After that they don't have to pay anything for using app. They only need an active internet connection, and they can facilitate themselves with all the features given. They can send and receive messages (one-to-one, group chat), pictures, voice notes. Users can create many groups with their family, friends, co-worker's etc.

7.3. Critical Review

Vibe.li is developed keeping Pakistan's people in mind as literacy rate is very low in Pakistan so, we made its interface user-friendly so, that our app can be used by more and more people. It will be easy for everyone to use all the features.

7.4. Lessons Learnt

The project has been a very good learning opportunity for all of us. We have gained experience on how to work on real time projects under the supervision of industry.

We have learned the complexity of a developing a messaging app. We have completed All basic functionalities for a messaging app. The overall learning experience was good as we collaborated with each other.

7.5. Future Enhancements/Recommendations

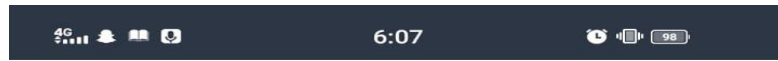
In future, more advanced and modified modules would be added to this application. For future we have a new module in mind that we want to implement like sending and receiving videos, files, calling. Which will be done on these after completing the modules mentioned in the project's scope?

Appendices

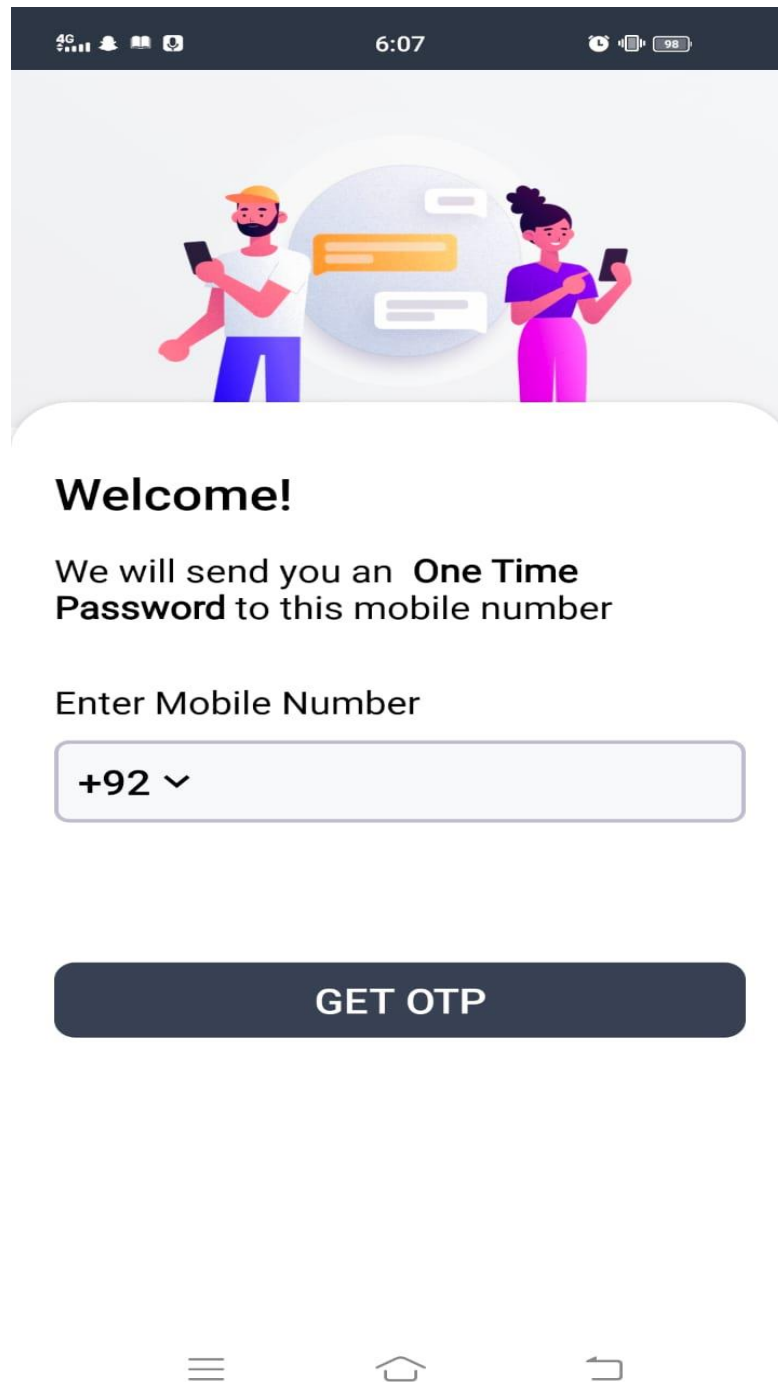
Appendix A:

User Manual

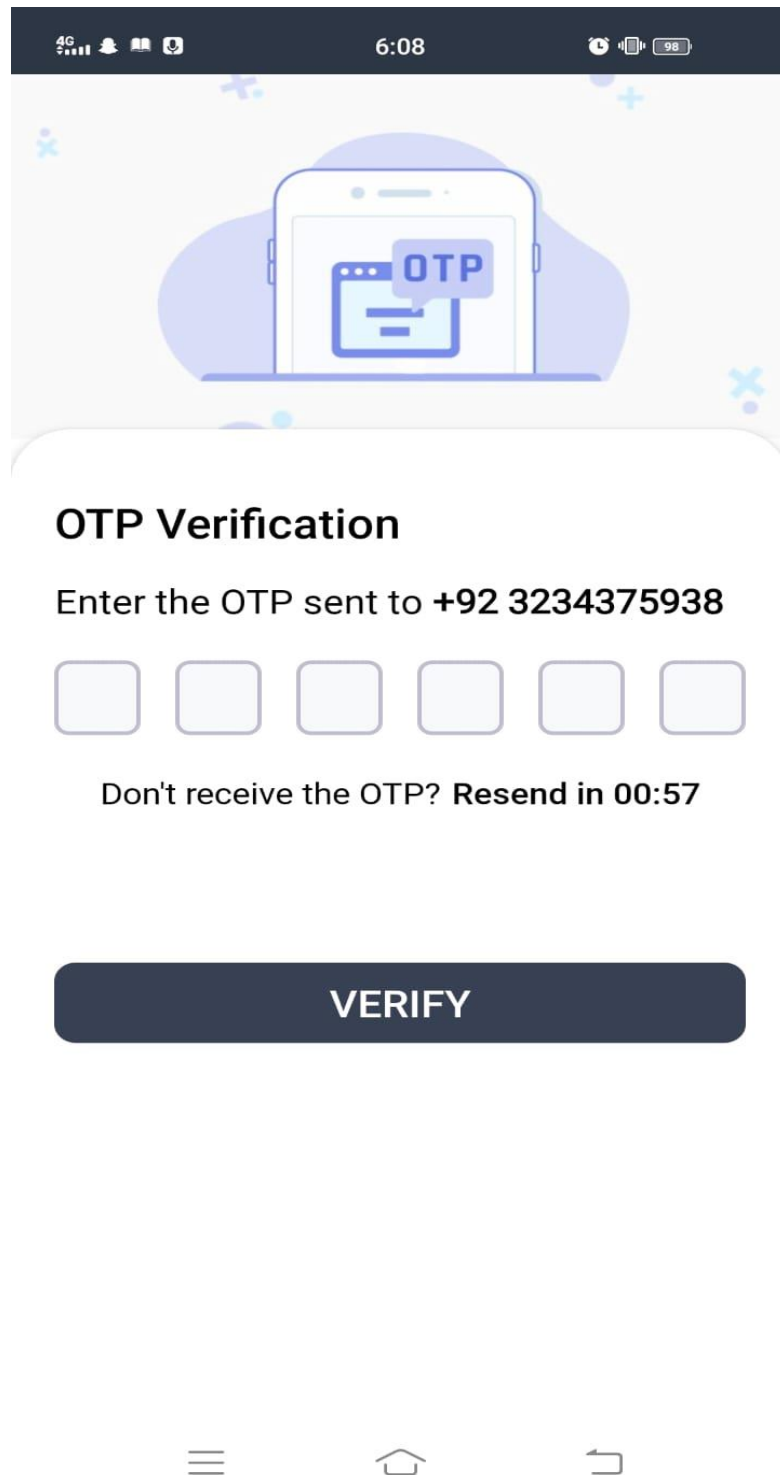
A.1. Splash



A.2. Login



A.3. Verification



A.4. Set name and Profile



Haniya|Fatima

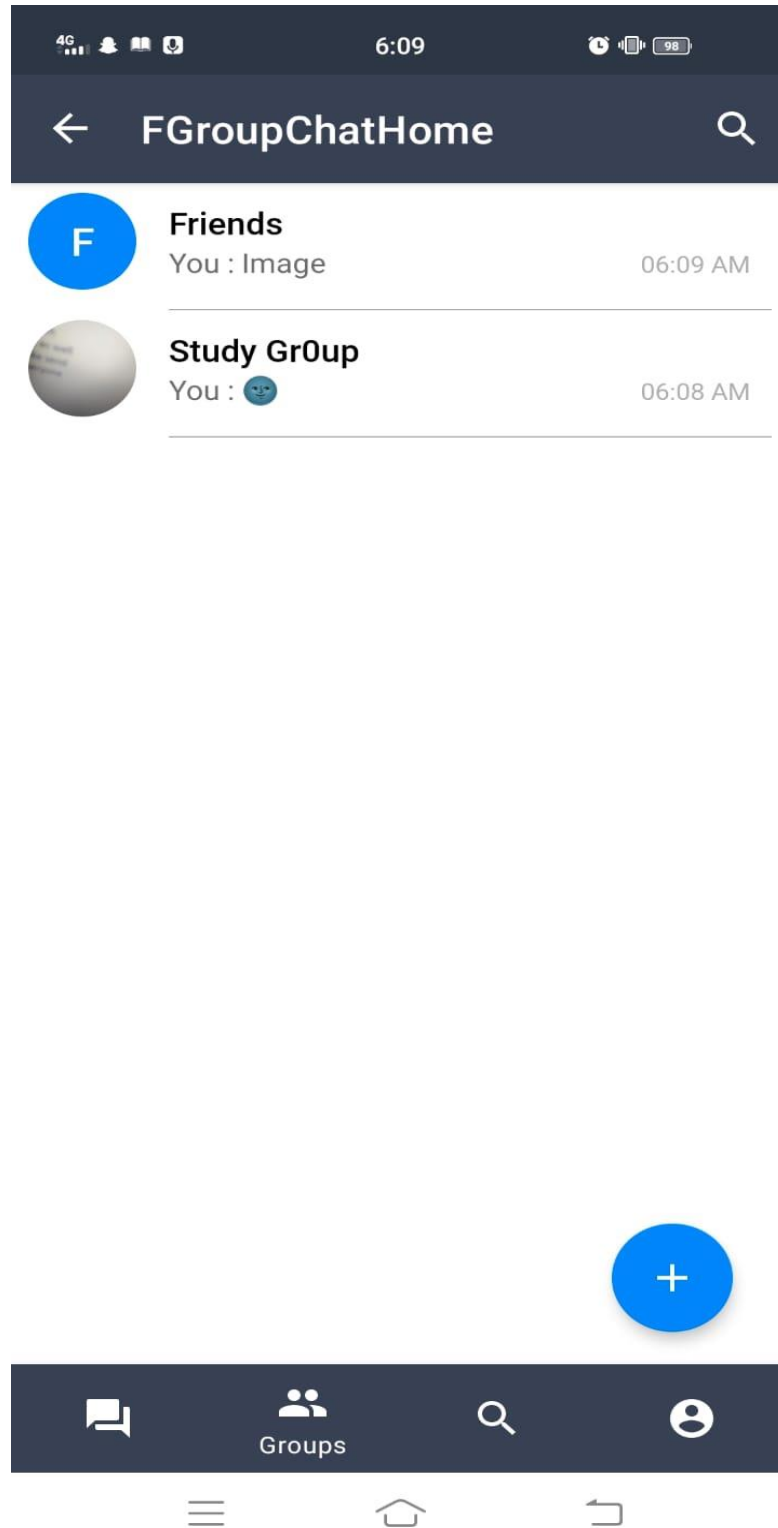
This user name and picture will be visible to your Letschat users. Other users can find you by searching.



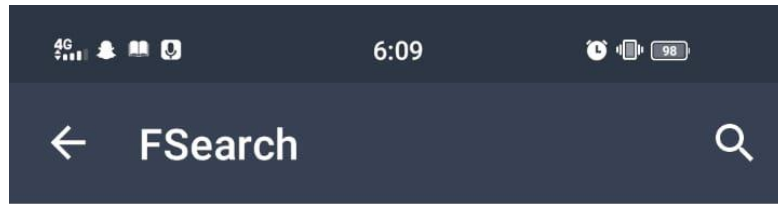
A.5. FsingleChat or one-to-one chat



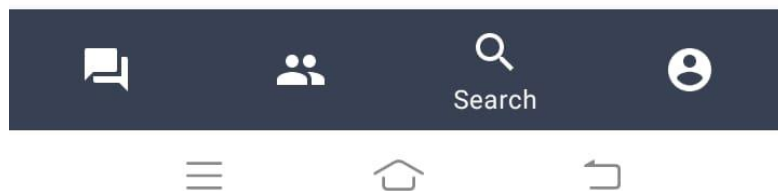
A.6. FgroupChat or group chat



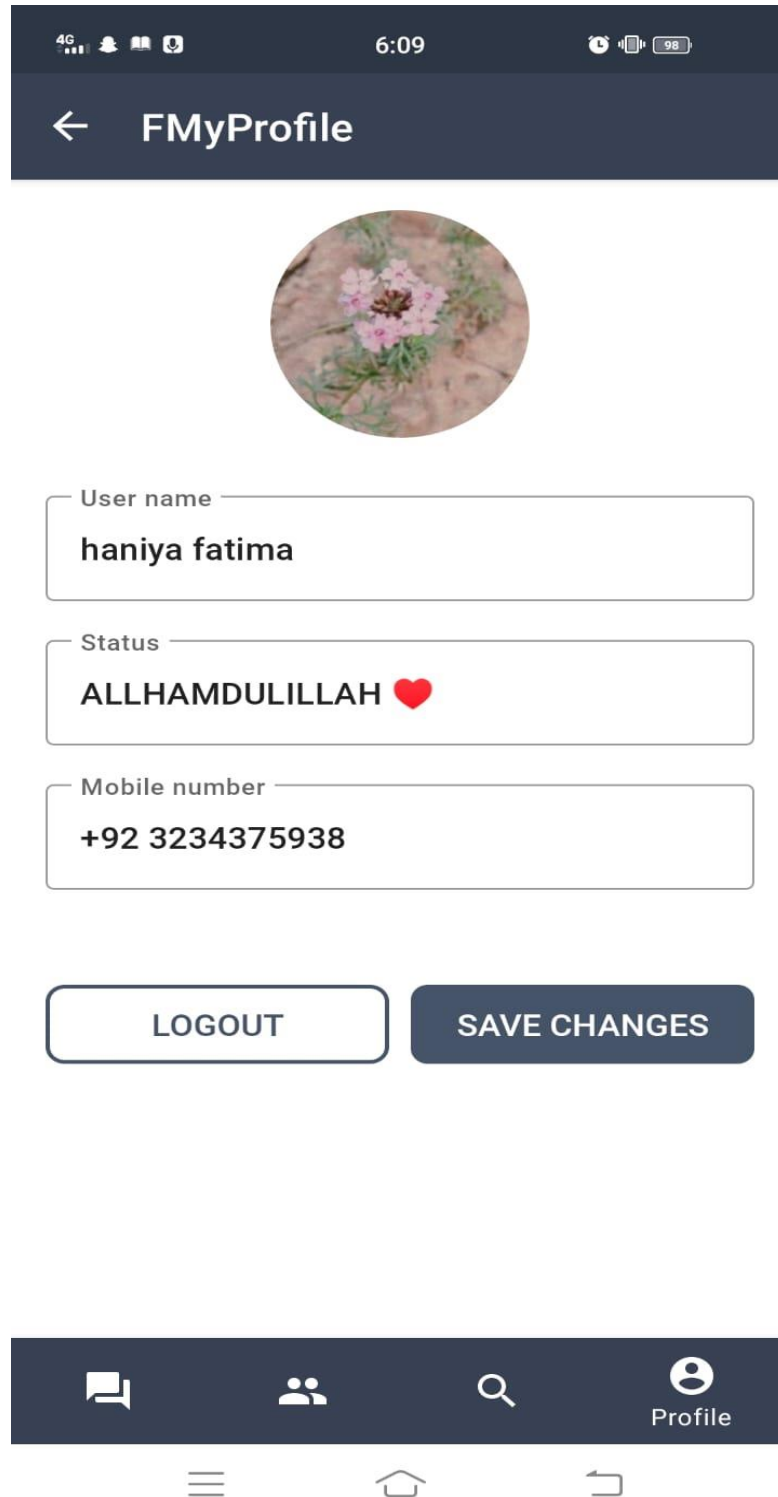
A.7. Fsearch



Search someone..



A.8. FmyProfile



Reference and Bibliography

Reference and Bibliography

- [1] <https://www.whatsapp.com/>
- [2] <https://hangouts.google.com/>
- [3] <https://medium.com/flutter-community/things-to-keep-in-mind-while-developing-a-messaging-app-51a4437bf1ee>
- [4] <https://stackoverflow.com/questions/70704716/how-to-recreate-fragments-after-configuration-change>

Index

Index

[A]

[B]

[C]