CHAT2TALK

A Major Project Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF COMPUTER APPLICATIONS

By

HARSHIKA SRIVASTAVA
(University Roll No. 1900290140015)
AMIT KUMAR
(University Roll No. 1900290140006)

Under the Supervision of

Dr. VIPIN KUMAR
Associate Professor
KIET Group of Institutions

Submitted to
DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
August 2021

DECLARATION

We hereby declare that the work presented in this report entitled **“Chat2Talk"**, was carried out by us. We have not submitted the matter embodied in this report for the award of any other degree or diploma of any other Universityor Institute.

We have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. We have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

We affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, We shall be fully responsible and answerable.

Harshika Srivastava (1900290140015)
Amit Kumar (1900290140006)

ACKNOWLEDGEMENT

At the outset, we would like to thank our guide and advisor**, Dr. VIPIN KUMAR Associate Professor,** for giving us an opportunity to work on this challenging topic and providing us ample and valuable guidance throughout the Project.

Without his encouragement and constant guidance, we would not have able to finish this project. He has been always a source of inspiration and motivator for innovative ideas during the entire span of this work.

We are grateful **Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications, KIET Group of Institutions, Ghaziabad** for providing all the necessary resources
to carry out this Project work.

We will be failing in our duty if we don't acknowledge the people behind this work to give us moral and psychological support. Our special thanks to our parents for their endless care and constant support.

Harshika Srivastava (1900290140015)
Amit Kumar **(**1900290140006)

CERTIFICATE

Certified that **Harshika Srivastava (1900290140015), Amit Kumar (1900290140006)** have carried out the project work having **"Chat2Talk"** for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date: **Harshika Srivastava (1900290140015)**

Amit Kumar (1900290140006)

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Date:

Dr. VIPIN KUMAR
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Ajay Kumar Shrivastava
Head, Department of Computer Application
KIET Group of Institutions, Ghaziabad

ABSTRACT

Our Project Chat2Talk is an Android based Messaging Application, which is designed and kept under the huge scope of the audience i.e. students and faculty of KIET Group of Institute.
In this era of social platforms having a medium of communication like Chat2Talk helps in sharing thoughts on status, quick chats wherein people can communicate initiating a chat to convert in Talks.

We have curated Chat2Talk to share messages, videos, stickers, that make Chat2Talk the most user-friendly and simple to use App.
We have used FIREBASE for Authentication of Phone Number and REAL TIME

The idea behind building this App was to understand how Android work and to learn FIREBASE and for this we thought it is best to build a chatting application
On daily basis we use social media apps for connecting with each other, so we were curious to see how this app works, that's why we started building this app.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PROJECT DESCRIPION

Communication is a mean for people to exchange messages. It has started since the beginning of human creation. Distant communication began as early as 1800 century with the introduction of television, telegraph and then telephony. Interestingly enough, telephone communication stands out as the fastest growing technology, from fixed line to mobile wireless, from voice call to data transfer. The emergence of computer network and telecommunication technologies bears the same objective that is to allow people to communicate. All this while, much efforts has been drawn towards consolidating the device into one and therefore indiscriminate the services. Chatting is a method of using technology to bring people and ideas together despite of the geographical barriers.

The technology has been available for years but the acceptance it was quit recent. Our project is an example of a chat server. It is made up of applications the client application which runs on the user's mobile on the network. To start chatting our client should get connected to server where they can do Group and private chatting.

Our Project Chat2Talk is an Android based Messaging Application, which is designed and kept under the huge scope of the audience i.e., students and faculty of KIET Group of Institute.
In this era of social platforms having a medium of communication like Chat2Talk helps in sharing thoughts on status, quick chats wherein people can communicate initiating a chat to convert in Talks. This project is to create a chat application with a server and users to enable the users to chat with each other's.

1.2 PROJECT SCOPE

The purpose of this project is to develop a java chat application. The objective of this process isas follows;

1. To develop an instant messaging solution to enable users to seamlessly communicate with each other.
2. The project should be very easy to use enabling even a novice person to use it.

Broadcasting Chat2Talk Application is going to be a text communication software, it will in communication between students for discussing their issues as well as ideas.

The limitation of Chat2Talk Application is that it does not support video calling feature. To overcome this limitation we are concurrently working making it possible.

Our college would like to have an android application wherein they can communicate instantly within their organization. The fact that the software uses an internal network setup within the organization makes it very secure from outside attacks.

The product has been successfully developed in terms of extendibility, portability, and maintainability and tested to meet all requirements that are

1. Authentication
2. Integrity
3. Confidentiality

Which are specified as the three basic concepts for the secure communication over a network.

1.3 FUTURE SCOPE

1. Extending this application by providing Authorization service.
2. Creating Database and maintaining users.
3. Increasing the effectiveness of the application by providing Voice Chat.
4. Extending it to Web Support.

1.4 IDENTIFICATION OF NEED

In the lockdown period we use many of the chatting application for connecting to the people outside world, even many of the company shifted its total work on online platform and use various platform to communicate or transfer the messages, but is that secure enough? No some apps/share the user data with others to earn profit and the user lose its privacy and sometime suffers loss due to misuse of their data.

So our team come up with a solution of an android based chatting application which will have a strict privacy policy so that the user's data can be safe. The messages will be completely end to end encrypted the user data AND MESSAGES will be kept secret with full protection.

- End -To-End Encryption of the message.
- Security of the data.

In our Chatting application the data of the user will be completely secure. The messages of the user will also be ended to end encrypted. As well as it will also provide the other feature like reaction on messages and status uploading etc.

1.5 PROBLEM STATEMENT

During the Lockdown the people cannot go from one place to another, so the only medium to communicate to the people is either call or through chat. But the problem related to chat is privacy of the people, so that their chat cannot be read by other people so our team come up with a solution to build a chatting application by using the android which is secure and respect the privacy of the people.

- End -To-End Encryption of the message.
- Security of the data.

In our Chatting application the data of the user will be completely secure. The messages of the user will also be end to end encrypted. As well as it will also provide the other feature like reaction on messages and status uploading etc.

1.6 SOFTWARE/TECHNOLOGY USED IN PROJECT

A. Android IDE

Figure 1.1: Android logo

Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug.

B. Firebase

Figure 1.2: Firebase logo

The Firebase Real-time Database is a cloud-hosted database. Data is stored as JSON and synchronized in Real-time to every connected client. When you build cross-platform apps with our Android and JavaScript SDKs, all of your clients share one Real-time Database instance and automatically receive updates with the newest data.

C. Java

Figure 1.3: JAVA logo

The mobile edition of Java is called Java ME. Java ME is based on Java SE and is supported by most smartphones and tablets. The Java Platform Micro Edition (Java ME) provides a flexible, secure environment for building and executing applications that are targeted at embedded and mobile devices

1.6.1 NON- FUNCTIONAL REQUIREMENTS

Performance Requirements:

To achieve good performance the following requirements must be satisfied

1. Scalability: The ease with which a system or component can be modified to fit the problem area.
2. Portability: The ease with which a system or component can be transferred from one hardware or software environment to another.
3. Security: It is the ideal state where all information can be communicated across the internet / company secure from unauthorized persons being able to read it and/or manipulate it.
4. Maintainability: The ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt toa changed environment.
5. Reliability: The ability of a system or component to perform its required functions under stated conditions for a specified period of time.
6. Reusability: The degree to which a software module or other work product can be used in more than one computing program or software system.

Safety Requirements:

In case scenarios where data integrity can be compromised, measures should betaken to ensure that all changes are made before system is shutdown. The user must have a registered account to use all facility of the web application.

1.6.2 FUNCTIONAL REQUIREMENTS

A. Firebase

Figure 1.4: Firebase logo

The Firebase Realtime Database is a cloud-hosted database in which data is stored as JSON. The data is synchronized in real-time to every connected client. All of our clients share one Realtime Database instances and automatically receive updates with the newest data, when we build cross-platform applications with our iOS, and JavaScript SDKs.

The Firebase Realtime Database is a NoSQL database from which we can store and sync the data between our users in real-time. It is a big JSON object which the developers can manage in real-time. By using a single API, the Firebase database provides the application with the current value of the data and updates to that data. Real-time syncing makes it easy for our users to access their data from any device, be it web or mobile.

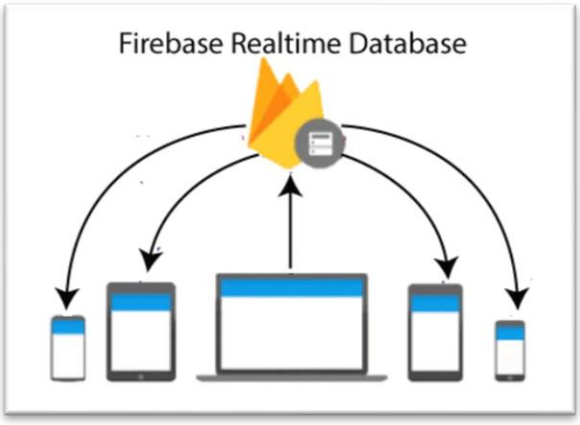


Figure 1.5: Realtime Firebase

B. Android Studio

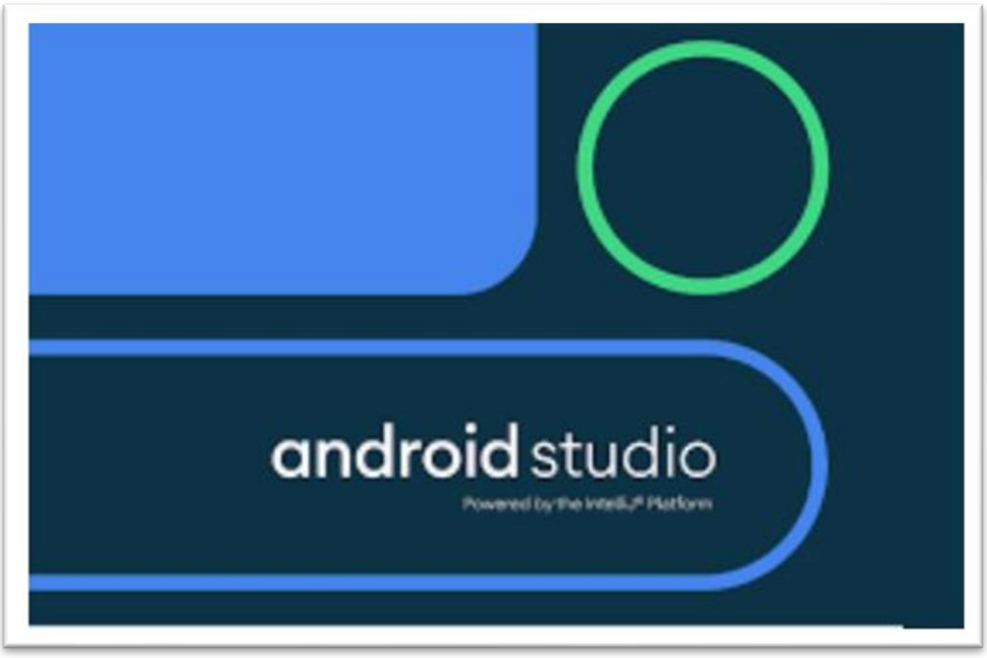


Figure 1.6: Android Studio Logo

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system

- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems

1.7 PROJECT SCHEDULE

The objective of software project planning is to provide a framework that enables the manager to make reasonable estimates of resources, costs and schedule. These estimates are made within a limited time frame at the beginning of a software project and should be updated regularly as the project progresses. In addition, estimates should attempts to define “best case” and “worst case” scenarios so that project outcomes can be bounded.

The first activity in software project planning is the determination of software scope. Function and performance allocated to software during system engineering should be assessed to establish a project scope that is ambiguous and understandable at management and technical levels. Software scope describes function, performance, constraints, interfacesand reliability.

During early stages of project planning, a microscopic schedule is developed. This type of schedule identifies all major software engineering activities and the product functions to which they are applied. As the project gets under way, each entry on the macroscopic schedule is refined into detailed schedule. Here specific software tasks are identified and scheduled.

Scheduling has following principles:

1. Compartmentalization: the project must be compartmentalized into a number of manageable activities and tasks.

2. Interdependency: the interdependencies of each compartmentalized activity or tasks must be determined.

3. Time allocation: each task to be scheduled must be allocated some number of work units.

4. Effort validation: every project has a defined number of staff members.

5. Defined responsibilities: every task that is scheduled should be assigned to a specific team member.

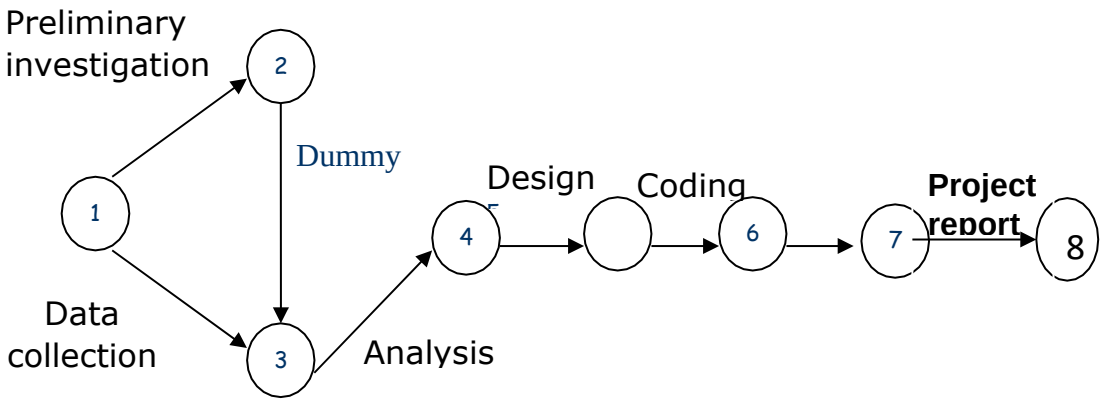
6. Defined outcomes: every task that is scheduled should have a defined outcome.

1.7.1 Pert chart

Program evaluation and review technique (pert) is a project scheduling method that is applied to software development.

Pert provide quantitative tool that allow the software planner to-Determine the critical path-the chain of tasks that determines the duration of the project; Establish “most likely” time estimates for individual tasks by applying statistical models; and calculate “boundary times” that defines a time “window” for a particular task.

Pert chart (program evolution review technique) for project-



1.7.2 Gantt Chart

When creating a project schedule, the planner begins with a set of tasks (the work breakdown structure). If automated tools are used, the work breakdown is input as a task network. Effort,duration and start dates are input are each task network. As a consequence of this input, a timeline chart also called a Gantt chart is generated. A timeline chart is developed for entire project.

Gantt chart for project:

Figure 1.7: Gantt Chart

CHAPTER 2

FEASBILITY STUDY

2.1 INTRODUCTION

Feasibility of the system in an important aspect, which is to be considered. The system needs to satisfy the law of economic, which states that the maximum output should be yielded in minimum available resources.

A feasibility analysis evaluates the project's potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions. There are five types of feasibility study separate areas that a feasibility study examines, described below.

1. Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building currently, this project is not technically feasible.

2. Economic Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

3. Legal Feasibility

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let's say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the

organization’s ideal location isn’t zoned for that type of business.

That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

4. Operational Feasibility

This assessment involves undertaking a study to analyze and determine whether and how well the organization's needs can be met by completing the project. Operational feasibility

studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

i. Scheduling Feasibility

This assessment is the most important for project success; after all, a project will fail if notcompleted on time. In scheduling feasibility, an organization estimates how much time theproject will take to complete.

When these areas have all been examined, the feasibility analysis helps identify anyconstraints the proposed project may face, including:

- Internal Project Constraints: Technical, Technology, Budget, Resource, etc.
- Internal Corporate Constraints: Financial, Marketing, Export, etc.
- External Constraints: Logistics, Environment, Laws, and Regulations, et

2.2 MAIN ASPECTS

There are three aspects of feasibility to be considered namely.

1. Technical
2. Operational
3. Economical

TECHNICAL:

In the technical aspects one may consider the hardware equipment for the installation of the software. The system being centralized will required very little hardware appliances. Hence this helps the system to work smoothly with limited amount of working capitals.

OPERATIONAL:

In the operational aspects may think of the benefits of the workload that many a personal may have to share. This is eased out and the required output may be retrieved in a very short time. Thus there is accuracy in the work on time is also saved there will be very little work that needs to be performed.

ECONOMICAL:

Economical system is definitely feasible because the software requirement is less and the operational working for the system requires less number of recruits. This help introduction over-staffing and wastage funds.

We studied on the position to evaluate solution. Most important factors in this study were tending to overlook the confusion inherent in Application Development the constraints and the assumed studies. It can be started that it the feasibility study is to serve as a decision document it must answer three key questions.

1. Is there a new and better way to do the job that will benefit the user?
2. What are the costs and savings of the alternatives?
3. What is recommended?

2.2.1 Technical feasibility:

This centers on the existing computer system (hardware, software etc.) and to what extent it can support the proposed additional equipment .in this stage of study, we have collected information about technical tools available by which I could decide my system design as the technical requirements.

2.2.2 Operational Feasibility:

In this stage of study we have checked the staff availability. I concentrate on knowledge of end users that are going to use the system. This is also called as behavioral feasibility in which I have studied on following aspects; people are inherently resistant to change, and computers have been known to facilitate change .An estimate has been made to how strong a reaction the user staff is having toward the development of a computerized system. It is common knowledge that computer installations have something to do with turnover. I had explained that there is need to educate and train the staff on new ways of conducting business.

2.2.3 Economical feasibility:

Economic analysis is the most frequently used method for evaluating the effectiveness of candidate system. More commonly known as cost\benefit analysis, the procedure is to determine the benefits and savings that benefits outweigh costs. The decision was to design and implement system because it is for having chanced to be approved. This is an on going effort that improves the accuracy at each phase of the system life cycle.

In developing cost estimates for a system I need to consider several cost elements. Among these is hardware personal facility. Operating and supply costs.

2.3 BENEFITS

Benefits of conducting a feasibility study:

- Improves project teams' focus
- Identifies new opportunities
- Provides valuable information for a "go/no-go" decision
- Narrows the business alternatives
- Identifies a valid reason to undertake the project
- Enhances the success rate by evaluating multiple parameters
- Aids decision-making on the project
- Identifies reasons not to proceed

2.4 SYSTEM REQUIREMENT SPECIFICATION

Any system can be designed after specifies the requirement of the user about that system. Forthis first of all gathered information from user by the preliminary investigation which is starting investigation about user requirement.

The data that the analysts collect during preliminary investigation are gathered through thevarious preliminary methods.

1) Documents Reviewing Organization

The analysts conducting the investigation first learn the organization involved in, or affectedby the project. Analysts can get some details by examining organization charts and studyingwritten operating procedures.

Collected data is usually of the current operating procedure:

- The information relating to clients, projects and students and the relationship between them was held manually.
- Managing of follow-ups was through manual forms.
- Complaints require another tedious work to maintain and solve.
- Payments details had to be maintained differently.

2) Gathering Information By Asking Questions

Interviewing is the most commonly used techniques in analysis. It is always necessary first toapproach someone and ask them what their problems are, and later to discuss with them the result of your analysis.

3) Questionnaires

Questionnaires provide an alternative to interviews for finding out information about a system. Questionnaires are made up of questions about information sought by analyst. The questionnaire is then sent to the user, and the analyst analyzes the replies.

4) Electronic Data Gathering

Electronic communication systems are increasingly being used to gather information. Thus itis possible to use electronic mail to broadcast a question to a number of users in an organization to obtain their viewpoint on a particular issue.

In my project, with the help of Marg software solutions, I have send questionnaire through electronic mail to twenty employees of the company and retrieved the information regardingthe problem faced by existing system.

5) Interviews

Interview allows the analysts to learn more about the nature of the project request and reasonof submitting it. Interviews should provide details that further explain the project and show whether assistance is merited economically, operationally or technically.

One of the most important points about interviewing is that what question youneed to ask.

It is often convenient to make a distinction between three kinds of question that is

- Open questions
- Closed question
- Probes

Open questions are general question that establish a persons view point on a particular subject.

Closed questions are specific and usually require a specific answer.

CHAPTER 3

DESIGN

3.1 INTRODUCTION

System is created to solve problems. One can think of the systems approach as an organizedway of dealing with a problem. In this dynamic world, the subject system analysis and design, mainly deals with the software development activities.

Since a new system is to be developed, the one most important phases of software development life cycle is system requirement gathering and analysis. Analysis is a detailed study of various operations performed by a system and their relationship within and outside the system. Using the following steps it becomes easy to draw the exact boundary of the new system under consideration.

All procedures, requirements must be analysed and documented in the form of detailedDFDs, logical data structure and miniature specifications.

System analyses also include sub-dividing of complex process involving the entire system,identification of data store and manual processes.

3.2 SYSTEM ANALYSIS

System is created to solve problems. One can think of the systems approach as an organizedway of dealing with a problem. In this dynamic world, the subject system analysis and design, mainly deals with the software development activities.

Since a new system is to be developed, the one most important phases of software development life cycle is system requirement gathering and analysis. Analysis involves detailed study of the current system, leading to specification of a new system. Analysis is a detailed study of various operations performed by a system and their relationship within and outside the system. Using the following steps it becomes easy to draw the exact boundary of the new system under consideration.

Keeping in view the problems and new requirements, workout the pros and cons includingnew area of the system.

All procedures, requirements must be analyzed and documented in the form of detailedDFDs, logical data structure and miniature specifications.

System analyses also include sub-dividing of complex process involving the entire system, identification of data store and manual processes.

System Analysis is conducted with the following steps

- Information gathering
- The tools of structured analysis
- Identification of Need
- System Planning and initial investigation
- Feasibility study

Information Gathering:

- Information about the firm
- Information about the workflow
- Various tools used are:
    - Review of literature
    - Procedure
    - Forms

Initial investigation:

- Problem definition and project initiation
- Determining the requirements
- Needs identification
- Dimension of planning
- Determination of feasibility

Feasibility Analysis:

- System Performance definition
- Identification of system objectives
- Description of outputs

Preliminary Investigation:

- Evaluation of project request is major purpose of preliminary investigation.
- It is the collecting information that helps committee members to evaluate merits of the project request and make judgment about the feasibility of the proposed projects.
- To answer the above questions, system analysts discuss with different category of person to collect facts about their business and their operations.
- When the request is made, the first activity the preliminary investigation begins.
- Preliminary investigation has three parts-

1. Request clarification
2. Feasibility study
3. Request approval

Request Clarification:

An information system is intended to meet needs of an organization. Thus the first step is inthis phase is to specify these needs and requirements.

- The next step is to determine the requirements met by the system. Many requests from employees and users in the organizations are not clearly defined. Therefore, it become necessary that project request must examine and clarified properly before considering system investigation.

- Information related to different needs of the System can be obtained by different users of the system. This can be done by reviewing different organization's documents such
- as current method of storing sales data, complaint data etc. By observing the onsite activities the analyst can get close information related to real system.

3.3 SDLC

Software Development Life Cycle (SDLC) is a framework that defines the steps involved in the development of software at each phase. It covers the detailed plan for building, deployingand maintaining the software.

SDLC defines the complete cycle of development i.e. all the tasks involved in planning,creating, testing, and deploying a Software Product.

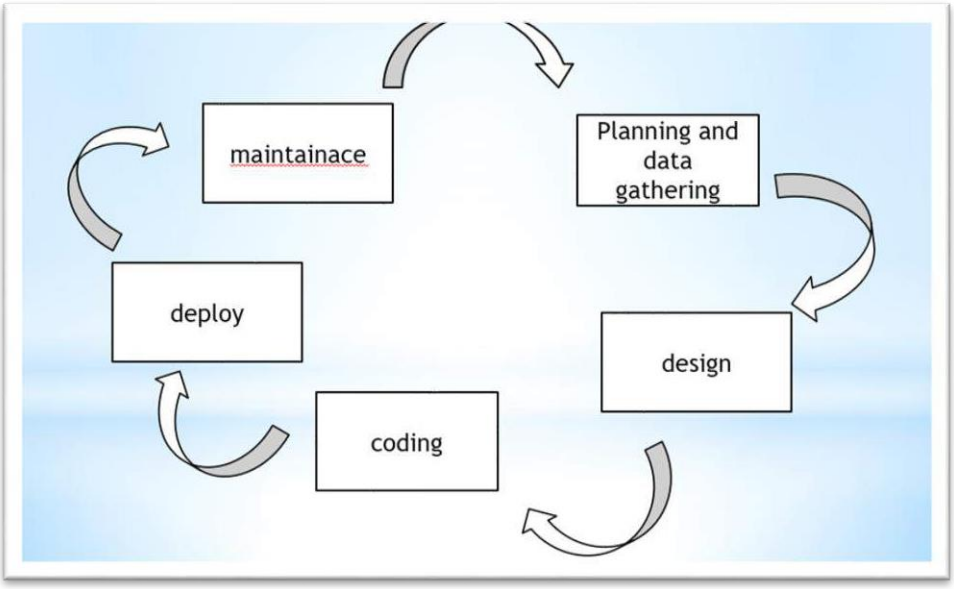


Figure 3.1: Above image depicting the planning step

SDLC Phases

Given below are the various phases:

- Requirement gathering and analysis
- Design
- Implementation or coding
- Testing

- Deployment
- Maintenance

1) Requirement Gathering and Analysis

During this phase, all the relevant information is collected from the customer to develop aproduct as per their expectation. Any ambiguities must be resolved in this phase only.

Business analyst and Project Manager set up a meeting with the customer to gather all the information like what the customer wants to build, who will be the end-user, what is the purpose of the product. Before building a product a core understanding or knowledge of the product is very important.

For Example, A customer wants to have an application which involves money transactions. In this case, the requirement has to be clear like what kind of transactions will be done, how it will be done, in which currency it will be done, etc.

Once the requirement gathering is done, an analysis is done to check the feasibility of thedevelopment of a product. In case of any ambiguity, a call is set up for further discussion.

Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers andalso should be reviewed by the customer for future reference.

2) Design

In this phase, the requirement gathered in the SRS document is used as an input and softwarearchitecture that is used for implementing system development is derived.

3) Implementation or Coding

Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented inthis phase.

4) Testing

Testing starts once the coding is complete and the modules are released for testing. In thisphase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.

Retesting, regression testing is done until the point at which the software is as per the customer's expectation. Testers refer SRS document to make sure that the software is as per the customer's standard.

5) Deployment

Once the product is tested, it is deployed in the production environment or first UAT (UserAcceptance testing) is done depending on the customer expectation.

In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing. If the customer finds the application as expected, then sign off is provided by the customer to go live.

6) Maintenance

After the deployment of a product on the production environment, maintenance of the product i.e., if any issue comes up and needs to be fixed or any enhancement is to be done istaken care by the developers.

3.4 SOFTWARE ENGG. PARADIGM APPLIED

Software engineering is a layered technology. The foundation for software engineering is theprocess layer. Software engineering processes the glue that holds the technology layers together and enables ratios and timely development of computer software. Process defines a framework for a set of key process areas that must be established for effective delivery of software engineering technology.

Software engineering methods provide the technical how-top's for building software. Methodsencompass a broad array of tasks that include requirements analysis, design, program

construction, testing and support. Software engineering tools provide automated or semi-automated support for the process and the methods. When tools are integrated so that information created by one tool can be used by another tool, a system for the support of software development, called computer-aided software engineering is established.

The following paradigms are available:

1. The Waterfall Model
2. The Prototyping Model
3. The Spiral model

Etc.

3.5 ARCHIETECTURE OF THE SYSTEM

An Architecture of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks.
Its for higher level, less detailed descriptions that are intended to clarify overall concepts without concern for the details of implementation.,

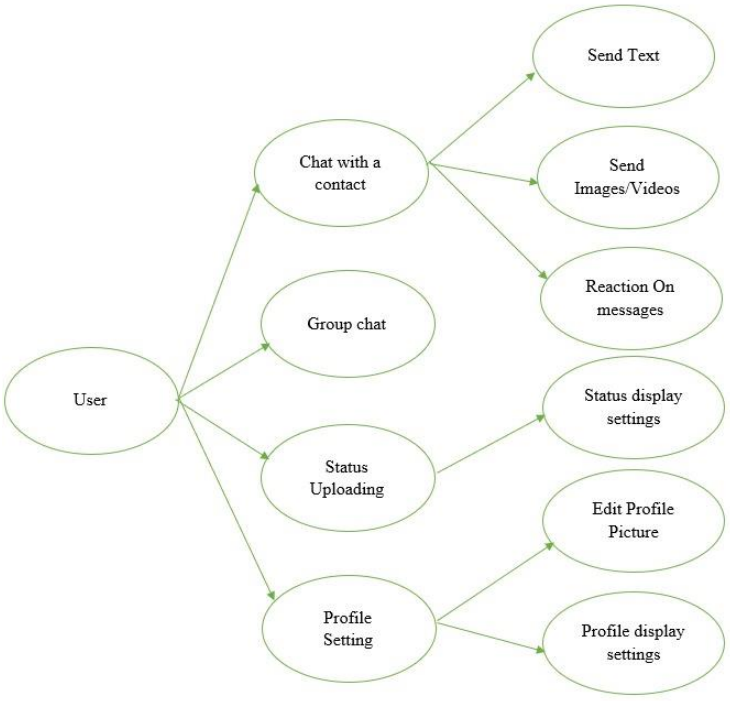


Figure 3.2: Architecture of the system

3.6 CONTROL FLOW GRAPH

A Control Flow Graph (CFG) is the graphical representation of control flow or computationduring the execution of programs or applications. Control flow graphs are mostly used in static analysis as well as compiler applications, as they can accurately represent the flow inside of a program unit.

Characteristics of Control Flow Graph:

- Control flow graph is process oriented & directed graph.
- Control flow graph shows all the paths that can be traversed during a program execution.

Edges in CFG portray control flow paths and the nodes in CFG portray basic block

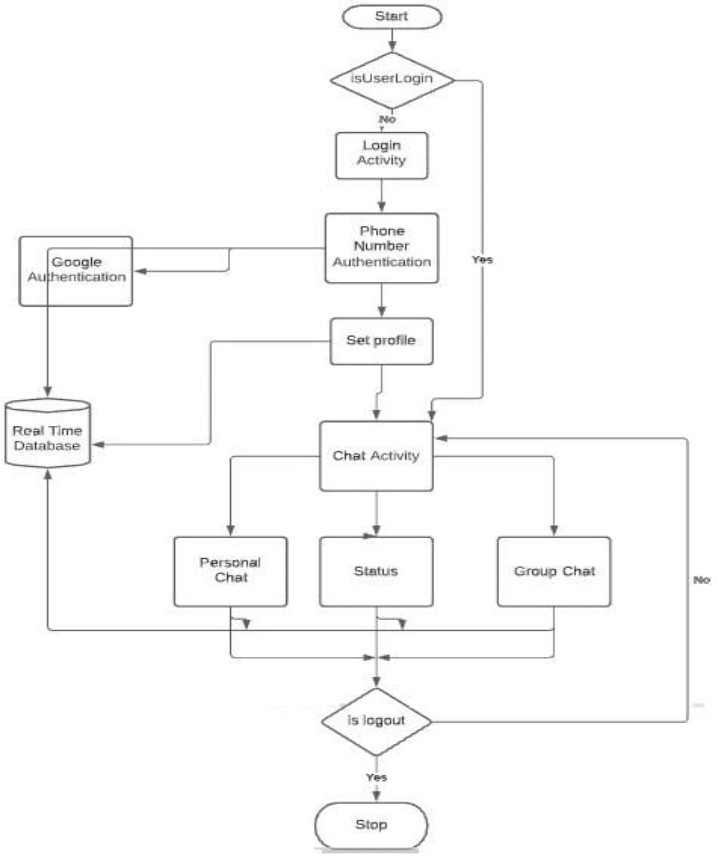


Figure 3.3: Control Flow Diagram

CHAPTER 4

REPORT

4.1 GIST

We have designed an Android Based Chatting Application for connecting the people through chat. In this Chatting Application the privacy of the user has been Prioritize. The Data of the user will be kept safely. For the user authentication we have used the Google firebase authentication (Phone number authentication).

And for storing the users profile information we have used the google database. For real time chatting we have used the Google Realtime database, which will allow user to send and receive messages from one user to another user. the user can also send the files such as text files or images or videos etc. All the data will be end-to-end encrypted.

The user will also be allowed to upload the profile as well as status on their application. As well as can see the status uploaded by other people. The user will also be able to send reactions on sender's messages.

4.2 SOME SNIPPETS

4.2.1 APP

First opening page of our Application Chat2Talk where user has to verify its Phone Number.

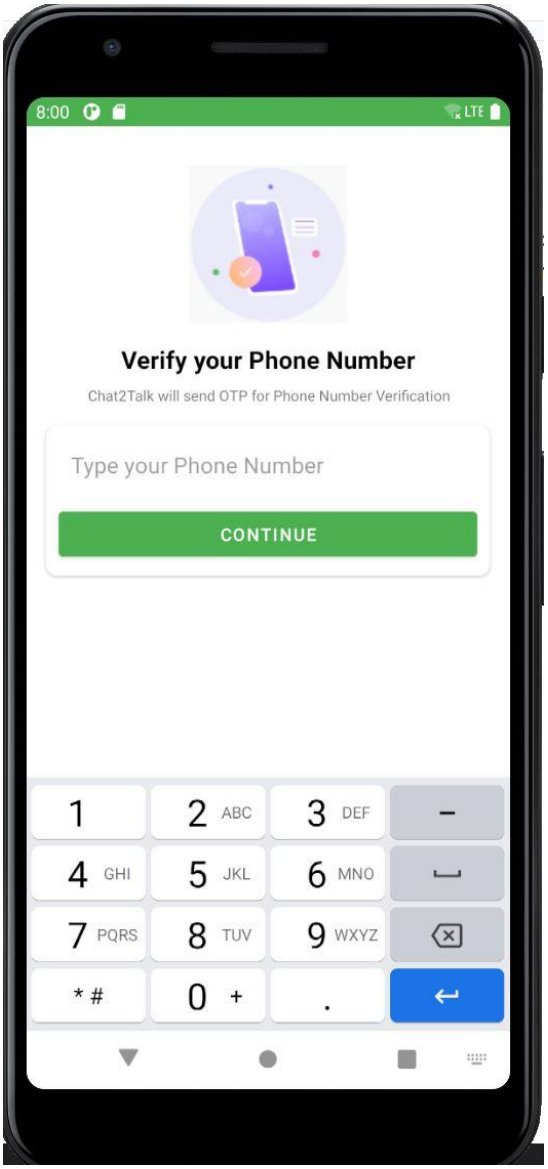


Figure 4.1: User Verification Page

In the given Screenshot we can see that user will receive OTP for verification

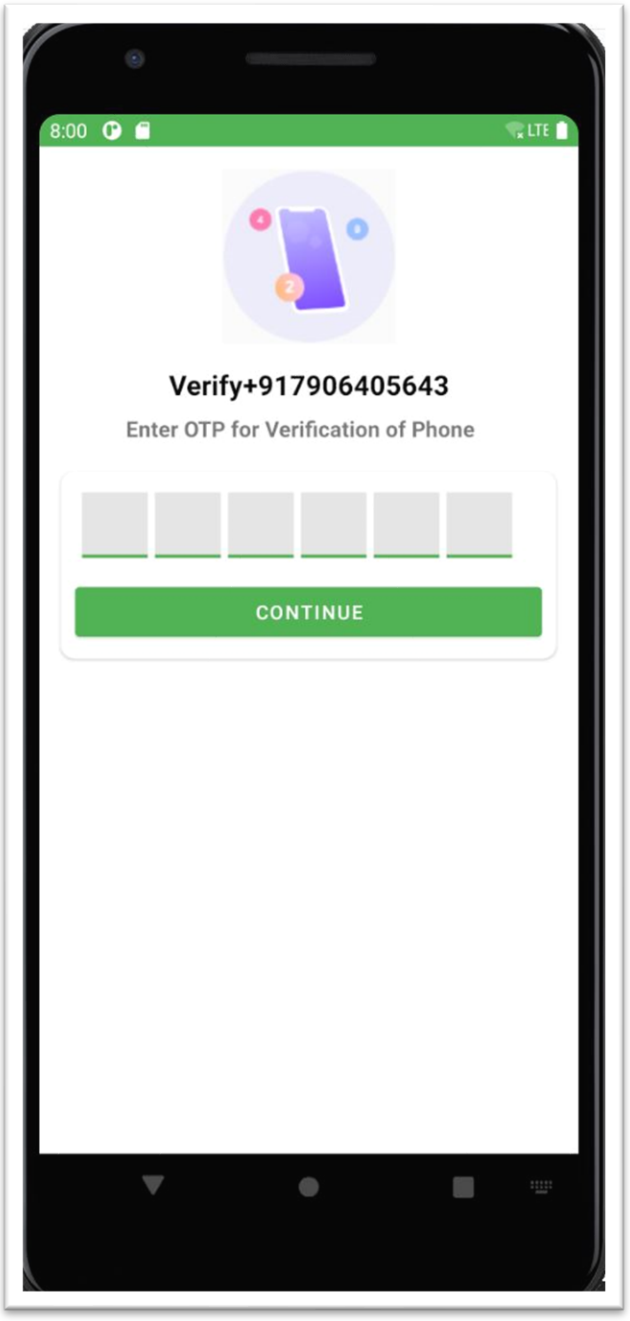


Figure 4.2: OTP Verification

In the given below screenshot user will setup profile image and its username

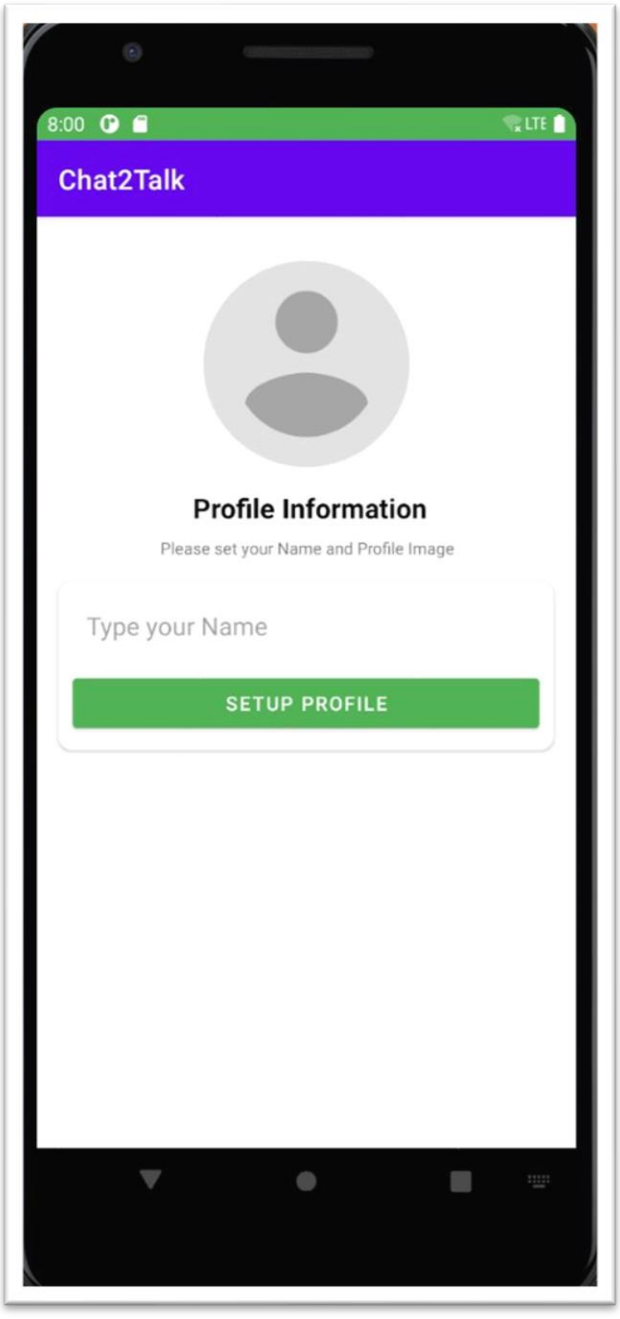


Figure 4.3: Setting profile

Given below screenshot is showing all the registered user of Chat2Talk App which have done there Phone Number Verification and have Set-up their profile.

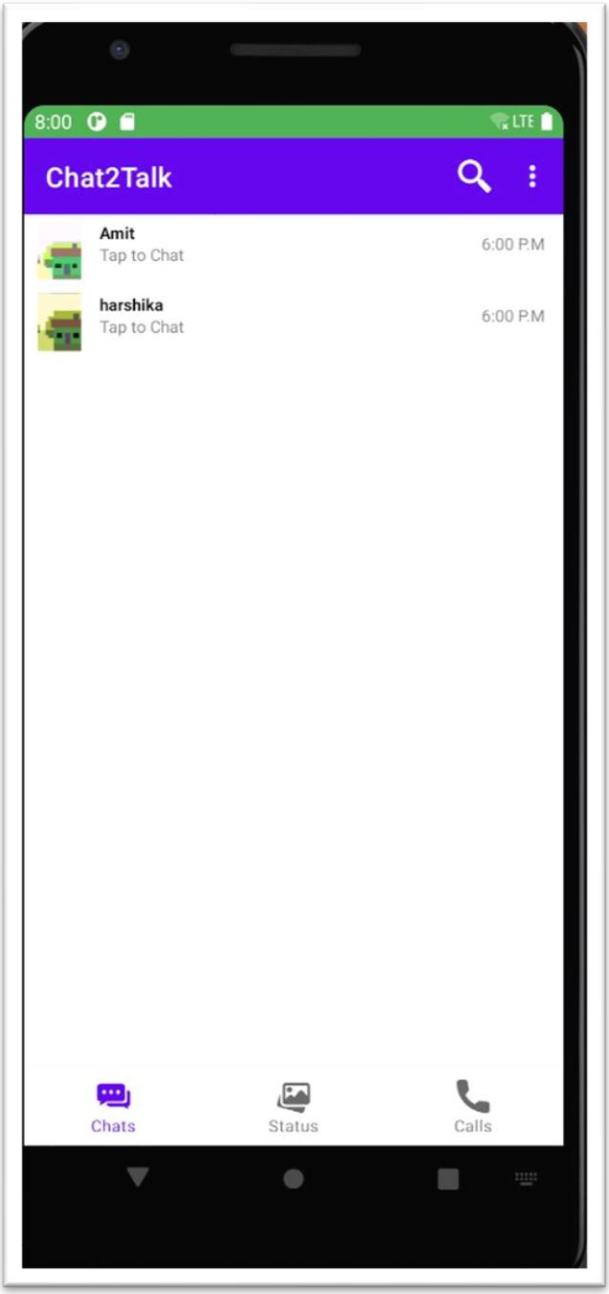


Figure 4.4: Registered User

4.2.2 FIREBASE

All Firebase Realtime Database data is stored as JSON objects. You can think of the databaseas a cloud hosted JSON tree. Unlike a SQL database, there are no tables or records. When you add data to the JSON tree, it becomes a node in the existing JSON structure with an associated key. You can provide your own keys, such as user IDs or semantic names, or they can be provided for you using push().

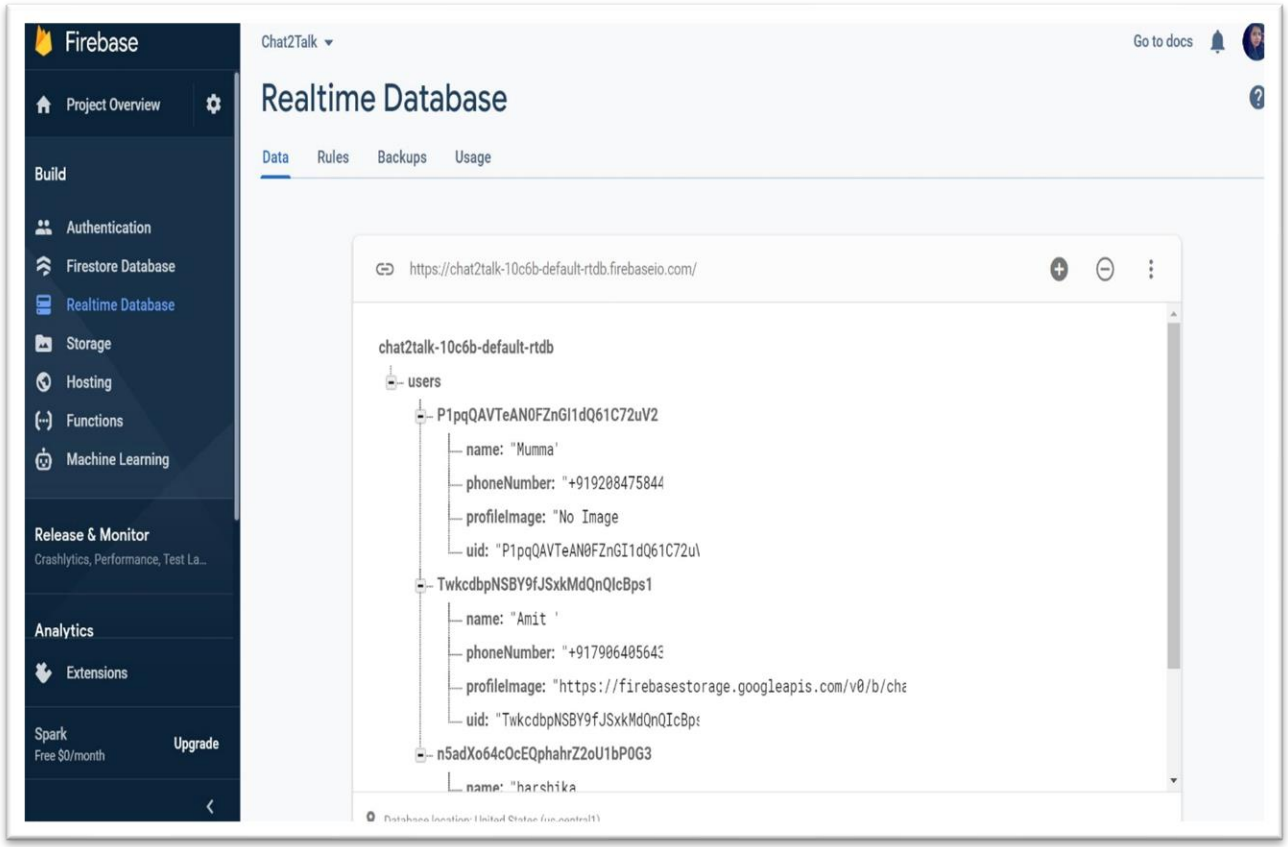


Figure 4.5: Firebase database

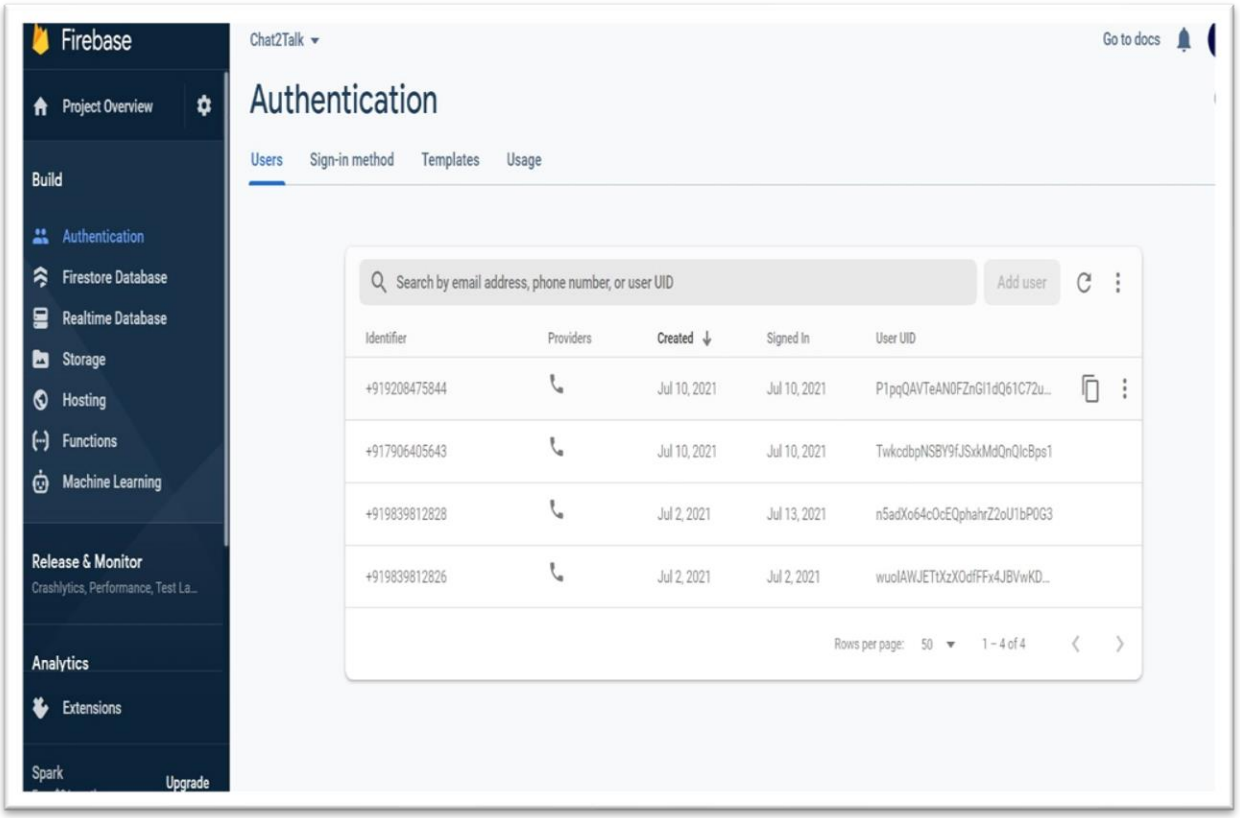


Figure 4.6: Firebase Database Authentication

4.2.3 ANDROID STUDIO

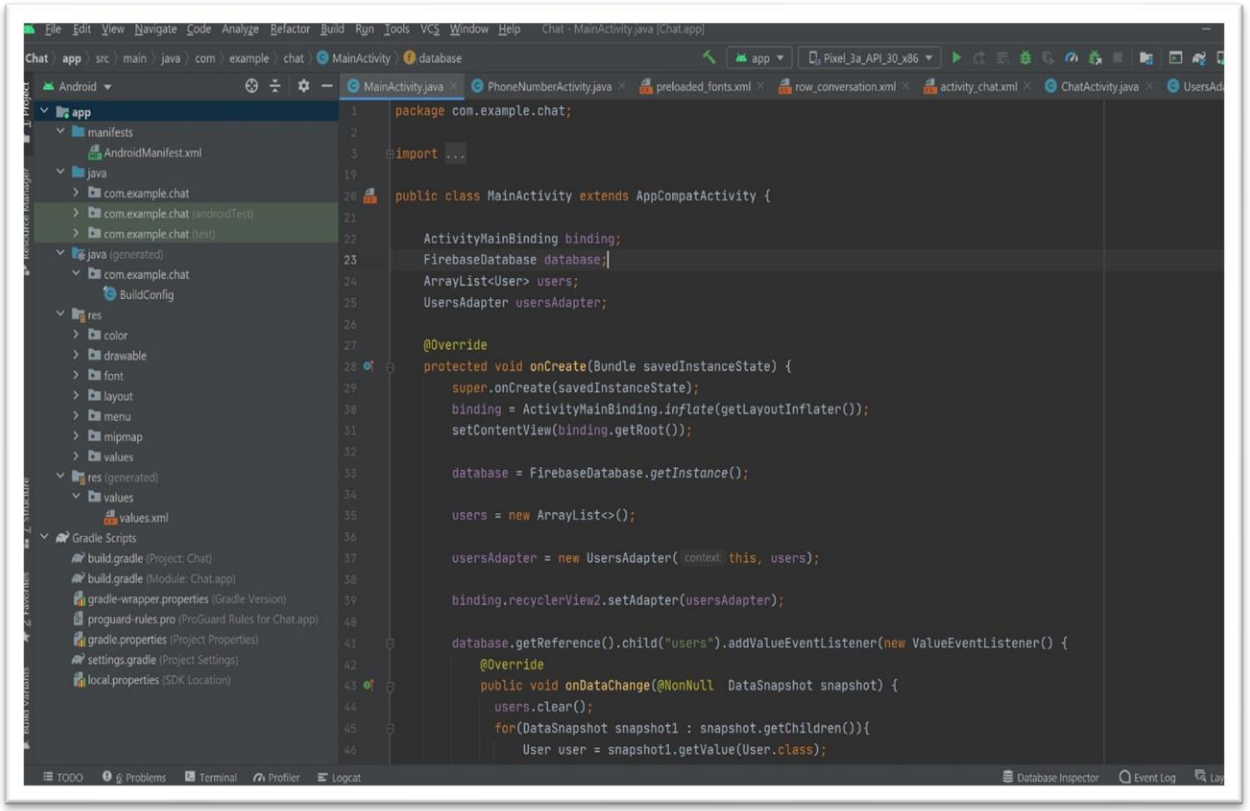


Figure 4.7: Android Studio Main Window

1. The **toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.
2. The **navigation bar** helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the **Project** window.
3. The **editor window** is where you create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.
4. The **tool window bar** runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.

CHAPTER 5

CODING

This chapter contains some codes of the project. The goal of the coding is to translate the design of the system into code in a given programming language. For a given design, the aim of this phase is to implement the design in the best possible manner. The coding phase affects both testing and maintenance profoundly.

Some Codes are as Written below:

MainActivity.java

The main activity code is a Java file MainActivity.java. This is the actual application file which ultimately gets converted to a Dalvik executable and runs your application. Following is the code generated of our application Chat2Talk.

```
package com.example.chat2talk;


import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;

import com.example.chat2talk.databinding.ActivityMainBinding;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    ActivityMainBinding binding;
    FirebaseDatabase database;
    ArrayList<User> users;
    UserAdapter userAdapter;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        database = FirebaseDatabase.getInstance();
        users = new ArrayList<>();

        userAdapter = new UserAdapter(this,users);
        binding.recyclerView.setAdapter(userAdapter);

        database.getReference().child("users")
              .addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(  DataSnapshot snapshot) {
                users.clear();
                for (DataSnapshot snapshot1: snapshot.getChildren()){
                    User user = snapshot1.getValue(User.class);
                    users.add(user);
                }
                userAdapter.notifyDataSetChanged();
            }

            @Override
            public void onCancelled(  DatabaseError error) {

            }
        });


    }

    @Override
    public boolean onOptionsItemSelected( MenuItem item) {
        switch(item.getItemId()){
            case R.id.search:
                Toast.makeText(this, "search",
Toast.LENGTH_SHORT).show();
            case R.id.groups:
                Toast.makeText( this,"Groups Clicked",
Toast.LENGTH_SHORT).show();
            case R.id.settings:
                Toast.makeText( this,"Settings Clicked",
Toast.LENGTH_SHORT).show();

        }
        return super.onOptionsItemSelected(item);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
```

```
        getMenuInflater().inflate(R.menu.topmenu,menu);
        return super.onCreateOptionsMenu(menu);
    }
}
```

PhoneNumberActivity.java

```
package com.example.chat2talk;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import com.example.chat2talk.databinding.ActivityPhoneNumberBinding;
import com.google.firebase.auth.FirebaseAuth;

public class PhoneNumberActivity extends AppCompatActivity {

    ActivityPhoneNumberBinding binding;
    FirebaseAuth auth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding =
ActivityPhoneNumberBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        auth = FirebaseAuth.getInstance();

        if(auth.getCurrentUser() != null) {
            Intent intent = new Intent(PhoneNumberActivity.this,
MainActivity.class);
            startActivity(intent);
            finish();
        }

        getSupportActionBar().hide();

        binding.phoneBox.requestFocus();
        binding.continueBtn.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent( PhoneNumberActivity.this,
OTPActivity.class);
                intent.putExtra("phoneNumber",
binding.phoneBox.getText().toString());

                startActivity(intent);
            }
```

```
        });
    }
}
```

OTPActivity.java

```
package com.example.chat2talk;


import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.inputmethod.InputMethodManager;
import android.widget.Toast;

import com.example.chat2talk.databinding.ActivityOtpactivityBinding;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.FirebaseException;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.PhoneAuthCredential;
import com.google.firebase.auth.PhoneAuthOptions;
import com.google.firebase.auth.PhoneAuthProvider;
import com.mukesh.OnOtpCompletionListener;

import java.util.concurrent.TimeUnit;

public class OTPActivity extends AppCompatActivity {

   ActivityOtpactivityBinding binding;
   FirebaseAuth auth;
   String verificationId;

   ProgressDialog dialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding
=ActivityOtpactivityBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

       dialog = new ProgressDialog(this);
        dialog.setMessage("Sending OTP...");
        dialog.setCancelable(false);
        dialog.show();

        auth = FirebaseAuth.getInstance();
```

```
        String phoneNumber = getIntent().getStringExtra("phoneNumber");
        binding.phoneLbl.setText("Verify " + phoneNumber );

        PhoneAuthOptions options = PhoneAuthOptions.newBuilder(auth)
                .setPhoneNumber(phoneNumber)
                .setTimeout(60L, TimeUnit.SECONDS)
                .setActivity(OTPActivity.this)
                .setCallbacks(new
PhoneAuthProvider.OnVerificationStateChangedCallbacks() {
                    @Override
                    public void onVerificationCompleted(
PhoneAuthCredential phoneAuthCredential) {

                    }

                    @Override
                    public void onVerificationFailed(FirebaseException
e) {

                    }

                    @Override
                    public void onCodeSent(String verifyId,
PhoneAuthProvider.ForceResendingToken forceResendingToken) {
                        super.onCodeSent(verifyId,
forceResendingToken);
                        // for waiting for the OTP
                        dialog.dismiss();

                        verificationId = verifyId;
                        // for showing keyboard in otp activity


                        InputMethodManager imm =(InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);

imm.toggleSoftInput(InputMethodManager.SHOW_FORCED,0);
                       binding.otpView.requestFocus();
                    }
                }).build();
        PhoneAuthProvider.verifyPhoneNumber(options);
        binding.otpView.setOtpCompletionListener(new
OnOtpCompletionListener() {
            @Override
            public void onOtpCompleted(String otp) {
                PhoneAuthCredential credential =
PhoneAuthProvider.getCredential(verificationId,otp);


auth.signInWithCredential(credential).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
                    @Override
```

```
                    public void onComplete(Task<AuthResult> task) {
                        if(task.isSuccessful()){



                            // for showing logged in
                           // Toast.makeText(OTPActivity.this,"Logged
In",Toast.LENGTH_SHORT).show();


                            Intent intent = new Intent(
OTPActivity.this,SetupProfileActivity.class);

                            startActivity(intent);
                            finishAffinity();
                        }
                        else
                        {
                          //  Intent intent = new Intent(
OTPActivity.this,SetupProfileActivity.class);

                            Toast.makeText(OTPActivity.this, "Failed",
Toast.LENGTH_SHORT).show();
                            //Intent intent = new Intent(
OTPActivity.this,SetupProfileActivity.class);
                           // startActivity(intent);
                        }
                    }
                });
            }
        });
    }
}
```

SetupProfileActivity.java

```
package com.example.chat2talk;


import androidx.appcompat.app.AppCompatActivity;

import android.app.ProgressDialog;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.text.method.CharacterPickerDialog;
import android.view.View;
import android.widget.Toast;

import com.example.chat2talk.databinding.ActivitySetupProfileBinding;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
```

```
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import java.util.Date;

import java.util.HashMap;

public class SetupProfileActivity extends AppCompatActivity {



  ActivitySetupProfileBinding binding;
    FirebaseAuth auth;
    FirebaseDatabase database;
    FirebaseStorage storage;
    Uri selectedImage;

    ProgressDialog dialog;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding =
ActivitySetupProfileBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        dialog = new ProgressDialog(this);
        dialog.setMessage("Updating profile....");
        dialog.setCancelable(false);


        database = FirebaseDatabase.getInstance();
        storage = FirebaseStorage.getInstance();
        auth = FirebaseAuth.getInstance();

        binding.imageView.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View v) {
                Intent intent = new Intent();
                intent.setAction(Intent.ACTION_GET_CONTENT);
                intent.setType("image/*");
                startActivityForResult(intent,45);


            }
        });
        binding.continueBtn.setOnClickListener(new
View.OnClickListener() {
            @Override
```

```
            public void onClick(View v) {
                String name =binding.naamBox.getText().toString();
                if(name.isEmpty()){
                    binding.naamBox.setError("Please Enter Your Name");
                    return;
                }
                dialog.show();
                if(selectedImage!=null){
                    StorageReference reference =
storage.getReference().child("Profile").child(auth.getUid());

reference.putFile(selectedImage).addOnCompleteListener(new
OnCompleteListener<UploadTask.TaskSnapshot>() {
                        @Override
                        public void onComplete(
Task<UploadTask.TaskSnapshot> task) {
                            if(task.isSuccessful()){

reference.getDownloadUrl().addOnSuccessListener(new
OnSuccessListener<Uri>() {
                                    @Override
                                    public void onSuccess(Uri uri) {
                                        String imageUrl =
uri.toString();
                                        String uid = auth.getUid();
                                        String phone =
auth.getCurrentUser().getPhoneNumber();
                                        String name =
binding.naamBox.getText().toString();

                                        User user = new
User(uid,name,phone,imageUrl);

                                           database.getReference()
                                                   .child("users")
                                                   .child(uid)
                                                   .setValue(user)

.addOnSuccessListener(new OnSuccessListener<Void>() {
                                                      @Override
                                                      public void
onSuccess(Void unused) {

dialog.dismiss();
                                                          Intent
intent = new Intent(SetupProfileActivity.this,MainActivity.class);

startActivity(intent);
                                                          finish();
                                                      }
                                                  });
                                    }
                                });
```

```
                            }
                        }
                    });
                }else{
                    String uid = auth.getUid();
                    String phone =
auth.getCurrentUser().getPhoneNumber();

                    User user = new User(uid,name,phone,"No Image");

                    database.getReference()
                            .child("users")
                            .child(uid)
                            .setValue(user)
                            .addOnSuccessListener(new
OnSuccessListener<Void>() {
                                @Override
                                public void onSuccess(Void unused) {
                                    dialog.dismiss();
                                    Intent intent = new
Intent(SetupProfileActivity.this,MainActivity.class);
                                    startActivity(intent);
                                    finish();
                                }
                            });
                }
            }
        });


    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(data!=null){
            if(data.getData() != null)
            {
                Uri uri = data.getData(); // filepath
                FirebaseStorage storage =
FirebaseStorage.getInstance();
                long time = new Date().getTime();
                StorageReference reference =
storage.getReference().child("Profiles").child(time+"");
                reference.putFile(uri).addOnCompleteListener(new
OnCompleteListener<UploadTask.TaskSnapshot>() {
                    @Override
                    public void onComplete(
Task<UploadTask.TaskSnapshot> task) {
                        if(task.isSuccessful()) {

reference.getDownloadUrl().addOnSuccessListener(new
```

```
OnSuccessListener<Uri>() {
                                    @Override
                                    public void onSuccess(Uri uri) {
                                        String filePath = uri.toString();
                                        HashMap<String, Object> obj = new
HashMap<>();
                                        obj.put("image", filePath);

database.getReference().child("users")

.child(FirebaseAuth.getInstance().getUid())

.updateChildren(obj).addOnSuccessListener(new OnSuccessListener<Void>()
{
                                            @Override
                                            public void onSuccess(Void
aVoid) {

                                            }
                                        });
                                    }
                                });
                            }
                        }
                    });


                    binding.imageView.setImageURI(data.getData());
                    selectedImage = data.getData();
                }
            }
        }
}


UserAdapter.java

package com.example.chat2talk;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

import com.bumptech.glide.Glide;
import com.example.chat2talk.databinding.RowConversationBinding;

import java.util.ArrayList;


import androidx.recyclerview.widget.RecyclerView;

public class UserAdapter extends
```

```
RecyclerView.Adapter<UserAdapter.UsersViewHolder> {

    Context context;
    ArrayList<User> users;
   public UserAdapter(Context context,ArrayList<User> users){
       this.context=context;
       this.users=users;
   }
    @Override
    public UsersViewHolder onCreateViewHolder(  ViewGroup parent, int
viewType) {

       View view =
LayoutInflater.from(context).inflate(R.layout.row_conversation,parent,f
alse);
        return new UsersViewHolder(view);
    }

    @Override
    public void onBindViewHolder(  UserAdapter.UsersViewHolder holder,
int position) {

       User user = users.get(position);
       holder.binding.userName.setText(user.getName());

        Glide.with(context).load(user.getProfileImage())
                .placeholder(R.drawable.avatar)
                .into(holder.binding.profile);
    }

    @Override
    public int getItemCount() {
        return users.size();
    }

    public class UsersViewHolder extends RecyclerView.ViewHolder{

        RowConversationBinding binding;

        public UsersViewHolder( View itemView){
            super(itemView);
            binding = RowConversationBinding.bind(itemView);
        }
    }
}
```

User.java

```
package com.example.chat2talk;

public class User {
    private String uid,name,phoneNumber,profileImage;
```

```
    public User(){

    }

    public User(String uid, String name, String phoneNumber, String
profileImage) {
        this.uid = uid;
        this.name = name;
        this.phoneNumber = phoneNumber;
        this.profileImage = profileImage;
    }

    public String getUid() {
        return uid;
    }

    public void setUid(String uid) {
        this.uid = uid;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public String getProfileImage() {
        return profileImage;
    }

    public void setProfileImage(String profileImage) {
        this.profileImage = profileImage;
    }
}
```

ChatActivity.java

```
package com.example.chat2talk;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
```

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;

import android.app.ProgressDialog;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.View;
import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.UploadTask;

import com.example.chat2talk.databinding.ActivityChatBinding;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;

public class ChatActivity extends AppCompatActivity {

    ActivityChatBinding binding;

    MessagesAdapter adapter;
    ArrayList<Message> messages;

    String senderRoom, receiverRoom;

    FirebaseDatabase database;
    FirebaseStorage storage;

    ProgressDialog dialog;
    String senderUid;
    String receiverUid;

    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        binding = ActivityChatBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        setSupportActionBar(binding.toolbar);

        database = FirebaseDatabase.getInstance();
        storage = FirebaseStorage.getInstance();

        dialog = new ProgressDialog(this);
        dialog.setMessage("Uploading image...");
        dialog.setCancelable(false);

        messages = new ArrayList<>();


        String name = getIntent().getStringExtra("name");
        String profile = getIntent().getStringExtra("image");

        binding.name.setText(name);
        Glide.with(ChatActivity.this).load(profile)
                .placeholder(R.drawable.avatar)
                .into(binding.profile);

        binding.imageView2.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });

        receiverUid = getIntent().getStringExtra("uid");
        senderUid = FirebaseAuth.getInstance().getUid();


database.getReference().child("presence").child(receiverUid).addValueEv
entListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if(snapshot.exists()) {
                    String status = snapshot.getValue(String.class);
                    if(!status.isEmpty()) {
                        if(status.equals("Offline")) {
                            binding.status.setVisibility(View.GONE);
                        } else {
                            binding.status.setText(status);
                            binding.status.setVisibility(View.VISIBLE);
                        }
                    }
                }
```

```
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });

        senderRoom = senderUid + receiverUid;
        receiverRoom = receiverUid + senderUid;

        adapter = new MessagesAdapter(this, messages, senderRoom,
receiverRoom);
        binding.recyclerView.setLayoutManager(new
LinearLayoutManager(this));
        binding.recyclerView.setAdapter(adapter);

        database.getReference().child("chats")
                .child(senderRoom)
                .child("messages")
                .addValueEventListener(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot
snapshot) {
                        messages.clear();
                        for(DataSnapshot snapshot1 :
snapshot.getChildren()) {
                            Message message =
snapshot1.getValue(Message.class);
                            message.setMessageId(snapshot1.getKey());
                            messages.add(message);
                        }

                        adapter.notifyDataSetChanged();
                    }

                    @Override
                    public void onCancelled(@NonNull DatabaseError
error) {

                    }
                });

        binding.sendBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String messageTxt =
binding.messageBox.getText().toString();

                Date date = new Date();
                Message message = new Message(messageTxt, senderUid,
date.getTime());
                binding.messageBox.setText("");
```

```
                String randomKey =
database.getReference().push().getKey();

                HashMap<String, Object> lastMsgObj = new HashMap<>();
                lastMsgObj.put("lastMsg", message.getMessage());
                lastMsgObj.put("lastMsgTime", date.getTime());


database.getReference().child("chats").child(senderRoom).updateChildren
(lastMsgObj);

database.getReference().child("chats").child(receiverRoom).updateChildr
en(lastMsgObj);

                database.getReference().child("chats")
                        .child(senderRoom)
.child("messages")
.child(randomKey)
.setValue(message).addOnSuccessListener(new OnSuccessListener<Void>() {
@Override
public void onSuccess(Void aVoid) {
database.getReference().child("chats")
.child(receiverRoom)
.child("messages")
.child(randomKey)
.setValue(message).addOnSuccessListener(new OnSuccessListener<Void>() {
@Override
public void onSuccess(Void aVoid) {

}
});
}
});

}
});

binding.attachment.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Intent intent = new Intent();
intent.setAction(Intent.ACTION_GET_CONTENT);
intent.setType("image/*");
startActivityForResult(intent, 25);
}
});

final Handler handler = new Handler();
binding.messageBox.addTextChangedListener(new TextWatcher() {
@Override
public void beforeTextChanged(CharSequence s, int start, int count, int
after) {
```

```
}

@Override
public void onTextChanged(CharSequence s, int start, int before, int
count) {

}

@Override
public void afterTextChanged(Editable s) {
database.getReference().child("presence").child(senderUid).setValue("ty
ping...");
handler.removeCallbacksAndMessages(null);
                handler.postDelayed(userStoppedTyping,1000);
            }

            Runnable userStoppedTyping = new Runnable() {
                @Override
                public void run() {

database.getReference().child("presence").child(senderUid).setValue("On
line");
                }
            };
        });


        getSupportActionBar().setDisplayShowTitleEnabled(false);

//        getSupportActionBar().setTitle(name);
//
//        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode,
@Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if(requestCode == 25) {
            if(data != null) {
                if(data.getData() != null) {
                    Uri selectedImage = data.getData();
                    Calendar calendar = Calendar.getInstance();
                    StorageReference reference = storage.
getReference().child("chats").child(calendar.getTimeInMillis() + "");
                    dialog.show();

reference.putFile(selectedImage).addOnCompleteListener(new
OnCompleteListener<UploadTask.TaskSnapshot>() {
                        @Override
```

```
                        public void onComplete(@NonNull
Task<UploadTask.TaskSnapshot> task) {
                            dialog.dismiss();
                            if(task.isSuccessful()) {

reference.getDownloadUrl().addOnSuccessListener(new
OnSuccessListener<Uri>() {
                                    @Override
                                    public void onSuccess(Uri uri) {
                                        String filePath =
uri.toString();

                                        String messageTxt =
binding.messageBox.getText().toString();

                                        Date date = new Date();
                                        Message message = new
Message(messageTxt, senderUid, date.getTime());
                                        message.setMessage("photo");
                                        message.setImageUrl(filePath);
                                        binding.messageBox.setText("");

                                        String randomKey =
database.getReference().push().getKey();

                                        HashMap<String, Object>
lastMsgObj = new HashMap<>();
                                        lastMsgObj.put("lastMsg",
message.getMessage());
                                        lastMsgObj.put("lastMsgTime",
date.getTime());


database.getReference().child("chats").child(senderRoom).updateChildren
(lastMsgObj);

database.getReference().child("chats").child(receiverRoom).updateChildr
en(lastMsgObj);


database.getReference().child("chats")
                                                .child(senderRoom)
                                                .child("messages")
                                                .child(randomKey)

.setValue(message).addOnSuccessListener(new OnSuccessListener<Void>() {
                                                @Override
                                                public void onSuccess(Void
aVoid) {

database.getReference().child("chats")

.child(receiverRoom)
```

```
.child("messages")

.child(randomKey)

.setValue(message).addOnSuccessListener(new OnSuccessListener<Void>() {
                                                        @Override
                                                        public void
onSuccess(Void aVoid) {

                                                        }
                                                    });
                                                }
                                            });


//Toast.makeText(ChatActivity.this, filePath,
Toast.LENGTH_SHORT).show();
                                        }
                                    });
                                }
                            }
                        });
                    }
                }
            }
        }

        @Override
        protected void onResume() {
            super.onResume();
            String currentId = FirebaseAuth.getInstance().getUid();

database.getReference().child("presence").child(currentId).setValue("On
line");
        }

        @Override
        protected void onPause() {
            super.onPause();
            String currentId = FirebaseAuth.getInstance().getUid();

database.getReference().child("presence").child(currentId).setValue("Of
fline");
        }

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
            getMenuInflater().inflate(R.menu.chat_menu, menu);
            return super.onCreateOptionsMenu(menu);
        }

        @Override
```

```
    public boolean onSupportNavigateUp() {
        finish();
        return super.onSupportNavigateUp();
    }
```

Layout

A layout defines the structure for a user interface in your app, such as in an activity. All elements in the layout are built using a hierarchy of View and View Group objects. A View usually draws something the user can see and interact with

```
Activity_main.xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="401dp"
        android:layout_height="671dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        tools:listitem="@layout/row_conversation"

app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
        >

    </androidx.recyclerview.widget.RecyclerView>

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:elevation="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:menu="@menu/menu"
        tools:layout_editor_absoluteX="16dp">


</com.google.android.material.bottomnavigation.BottomNavigationView>
</androidx.constraintlayout.widget.ConstraintLayout>

Activity_phone_number.xml
```

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="10dp"
    android:background="#D9E2E3"
    tools:context=".PhoneNumberActivity">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="227dp"
        android:layout_height="165dp"
        android:layout_marginTop="16dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/ph" />

    <TextView
        android:id="@+id/phoneLbl"
        android:layout_width="292dp"
        android:layout_height="63dp"
        android:layout_marginTop="16dp"
        android:gravity="center"
        android:text="Verify Your phone Number"
        android:textColor="#100F0F"
        android:textSize="20sp"
        app:layout_constraintEnd_toEndOf="@+id/imageView"
        app:layout_constraintStart_toStartOf="@+id/imageView"
        app:layout_constraintTop_toBottomOf="@+id/imageView" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="388dp"
        android:layout_height="33dp"
        android:layout_marginTop="16dp"
        android:gravity="center"
        android:shadowColor="#0B0A0A"
        android:text="chat2talk will send an sms to verify your phone
number."
        android:textColor="@android:color/black"
        android:textStyle="bold"
        app:layout_constraintEnd_toEndOf="@+id/phoneLbl"
        app:layout_constraintStart_toStartOf="@+id/phoneLbl"
        app:layout_constraintTop_toBottomOf="@+id/phoneLbl" />

    <androidx.cardview.widget.CardView
        android:layout_width="400dp"
        android:layout_height="213dp"
        android:layout_marginStart="16dp"
```

```
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        app:cardBackgroundColor="#FAF3F3"
        app:cardCornerRadius="10dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView2">

        <LinearLayout
            android:layout_width="380dp"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:orientation="vertical">

            <EditText
                android:id="@+id/phoneBox"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_margin="10dp"
                android:background="@drawable/textbox_outline"
                android:ems="10"
                android:hint="Type your phone number"
                android:inputType="phone|number"
                android:padding="15dp"
                tools:visibility="visible" />

            <Button
                android:id="@+id/continueBtn"
                android:layout_width="321dp"
                android:layout_height="50dp"
                android:layout_gravity="center"

                android:layout_marginTop="15dp"
                android:height="100dp"
                android:minHeight="72dp"
                android:text="continue"
                app:backgroundTint="#5C3B85" />
        </LinearLayout>

    </androidx.cardview.widget.CardView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_oypactivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_marginTop="10dp"
android:background="#D9E2E3"
tools:context=".OTPActivity">

<ImageView
    android:id="@+id/imageView"
    android:layout_width="227dp"
    android:layout_height="165dp"
    android:layout_marginTop="16dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/ph" />

<TextView
    android:id="@+id/phoneLbl"
    android:layout_width="393dp"
    android:layout_height="58dp"
    android:layout_marginTop="16dp"
    android:gravity="center"
    android:text="Verify +917906405643"
    android:textColor="#100F0F"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="@+id/imageView"
    app:layout_constraintStart_toStartOf="@+id/imageView"
    app:layout_constraintTop_toBottomOf="@+id/imageView" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="388dp"
    android:layout_height="33dp"
    android:layout_marginTop="16dp"
    android:gravity="center"
    android:shadowColor="#0B0A0A"
    android:text="Enter Your OTP to verify Phone Number"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="@+id/phoneLbl"
    app:layout_constraintStart_toStartOf="@+id/phoneLbl"
    app:layout_constraintTop_toBottomOf="@+id/phoneLbl" />

<androidx.cardview.widget.CardView
    android:layout_width="389dp"
    android:layout_height="147dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="16dp"
    app:cardCornerRadius="10dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.294"
    app:layout_constraintStart_toStartOf="parent"
```

```
        app:layout_constraintTop_toBottomOf="@+id/textView2">

        <LinearLayout
            android:layout_width="396dp"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:orientation="vertical">

            <com.mukesh.OtpView
                android:id="@+id/otp_view"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:layout_marginTop="10dp"
                android:inputType="number"
                android:itemBackground="@color/gray"
                android:textColor="@android:color/black"
                app:OtpItemCount="6"
                app:OtpLineColor="@color/pur"
                app:OtpViewType="line"
                tools:visibility="visible" />

            <Button
                android:id="@+id/continueBtn"
                android:layout_width="321dp"
                android:layout_height="50dp"
                android:layout_gravity="center"

                android:layout_marginTop="15dp"
                android:height="100dp"
                android:minHeight="72dp"
                android:text="continue"
                app:backgroundTint="#5C3B85" />
        </LinearLayout>

    </androidx.cardview.widget.CardView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_setup_profile.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/nameBox"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="10dp"
    android:background="#D9E2E3"
```

```
tools:context=".SetupProfileActivity">

<de.hdodenhof.circleimageview.CircleImageView
    android:id="@+id/imageView"
    android:layout_width="206dp"
    android:layout_height="176dp"
    android:layout_marginTop="32dp"
    android:src="@drawable/avatar"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/phoneLbl"
    android:layout_width="292dp"
    android:layout_height="63dp"
    android:layout_marginTop="24dp"
    android:gravity="center"
    android:text="Profile Info"
    android:textColor="#100F0F"
    android:textSize="20sp"
    app:layout_constraintEnd_toEndOf="@+id/imageView"
    app:layout_constraintStart_toStartOf="@+id/imageView"
    app:layout_constraintTop_toBottomOf="@+id/imageView" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="388dp"
    android:layout_height="33dp"
    android:layout_marginTop="24dp"
    android:gravity="center"
    android:shadowColor="#0B0A0A"
    android:text="Please Set Your Name And Optional Profile Image"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="@+id/phoneLbl"
    app:layout_constraintStart_toStartOf="@+id/phoneLbl"
    app:layout_constraintTop_toBottomOf="@+id/phoneLbl" />

<androidx.cardview.widget.CardView
    android:layout_width="400dp"
    android:layout_height="213dp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="24dp"
    android:layout_marginEnd="16dp"
    app:cardCornerRadius="10dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2">

    <LinearLayout
        android:layout_width="380dp"
        android:layout_height="wrap_content"
```

```
            android:layout_margin="10dp"
            android:orientation="vertical">

            <EditText
                android:id="@+id/naamBox"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_margin="10dp"
                android:background="@drawable/textbox_outline"
                android:ems="10"
                android:hint="Type Your Name"
                android:inputType="text"
                android:padding="15dp"
                tools:visibility="visible" />

            <Button
                android:id="@+id/continueBtn"
                android:layout_width="321dp"
                android:layout_height="50dp"
                android:layout_gravity="center"

                android:layout_marginTop="15dp"
                android:height="100dp"
                android:minHeight="72dp"
                android:text="continue"
                app:backgroundTint="#5C3B85" />
        </LinearLayout>

    </androidx.cardview.widget.CardView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_chat.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/back"
    tools:context=".chatActivity">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="50dp"
        app:cardCornerRadius="30dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent">
```

```
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal">

            <EditText
                android:id="@+id/msgBox"
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:layout_weight="1"
                android:background="@android:color/transparent"
                android:ems="10"
                android:hint="type a message"
                android:inputType="textPersonName"
                android:padding="8dp"
                android:textSize="20dp" />

            <ImageView
                android:id="@+id/imageView2"
                android:layout_width="40dp"
                android:layout_height="wrap_content"
                android:layout_marginRight="10dp"
                app:srcCompat="@drawable/attachment" />

            <ImageView
                android:id="@+id/imageView3"
                android:layout_width="40dp"
                android:layout_height="wrap_content"
                android:layout_marginRight="10dp"
                app:srcCompat="@drawable/camera" />
        </LinearLayout>
    </androidx.cardview.widget.CardView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Row_conversation.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="10dp">

    <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/profile"
        android:layout_width="50dp"
        android:layout_height="50dp"
```

```
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:src="@drawable/avatar" />

    <TextView
        android:id="@+id/userName"
        android:layout_width="156dp"
        android:layout_height="24dp"
        android:layout_marginStart="8dp"
        android:text="Sample Name"
        android:textStyle="bold"
        app:layout_constraintStart_toEndOf="@+id/profile"
        app:layout_constraintTop_toTopOf="@+id/profile" />

    <TextView
        android:id="@+id/lastMsg"
        android:layout_width="wrap_content"
        android:layout_height="26dp"
        android:text="Tap to chat"
        app:layout_constraintStart_toStartOf="@+id/userName"
        app:layout_constraintTop_toBottomOf="@+id/userName" />

    <TextView
        android:id="@+id/msgTime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="8dp"
        android:text="06.00PM"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <View
        android:id="@+id/view"
        android:layout_width="wrap_content"
        android:layout_height="5dp"
        android:layout_marginTop="8dp"
        android:background="#8469FA"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/profile" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Menu

top_menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:id="@+id/search"
```

```
        android:icon="@drawable/loupe"
        android:title="search"
        app:showAsAction="always" />
    <item
        android:id="@+id/groups"
        android:title="groups" />
    <item
        android:id="@+id/settings"
        android:title="settings" />
</menu>
```

menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:icon="@drawable/chat"
        android:title="chats" />
    <item
        android:icon="@drawable/mode"
        android:title="status" />
</menu>
```

Values

Values.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="default_web_client_id" translatable="false">
855241948233-
3t3210dj8vi31rn1d7esscl66nvn8phj.apps.googleusercontent.com
</string>
    <string name="gcm_defaultSenderId" translatable="false">
855241948233
</string>
    <string name="google_api_key" translatable="false">
AIzaSyDFMiEUPigTWWKtXJzlSba17HaqrIqE0ls
</string>
    <string name="google_app_id" translatable="false">
1:855241948233:android:138b92456878f6498c1398
</string>
    <string name="google_crash_reporting_api_key" translatable="false">
AIzaSyDFMiEUPigTWWKtXJzlSba17HaqrIqE0ls
</string>
    <string name="google_storage_bucket"
translatable="false">chat2talk-2883f.appspot.com
</string>
    <string name="project_id" translatable="false">
chat2talk-2883f
</string>
</resources>
```

build.graddle(Project: chat2talk)

```
// Top-level build file where you can add configuration options common
to all sub-projects/modules.
buildscript
{
    repositories
{
        google()
        mavenCentral()
    }
    dependencies
{
        classpath "com.android.tools.build:gradle:4.2.1"
        classpath 'com.google.gms:google-services:4.3.8'

        // NOTE: Do not place your application dependencies here; they
belong
        // in the individual module build.gradle files
    }
}

allprojects
{
    repositories
 {
        google()
        mavenCentral()
      //  jcenter() // Warning: this repository is going to shut down
soon
        maven
{
 url "https://jitpack.io" }
    }
}

task clean(type: Delete)
{
    delete rootProject.buildDir
}
```

build.graddle(Module: chat2talk app)

```
plugins
 {
    id 'com.android.application'
}
apply plugin: 'com.google.gms.google-services'

android
{
    compileSdkVersion 30
    buildToolsVersion "30.0.3"

    buildFeatures
```

```
{

viewBinding true
    }

    defaultConfig
{
        applicationId "com.example.chat2talk"
        minSdkVersion 17
        targetSdkVersion 30
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner
"androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes
 {
        release
 {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-
optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions
{
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies
{

    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'com.google.android.material:material:1.2.1'
    implementation 'androidx.constraintlayout:constraintlayout:2.0.1'
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-
core:3.3.0'

    implementation 'com.github.mukeshsolanki:android-otpview-
pinview:2.1.2'

    implementation platform('com.google.firebase:firebase-bom:28.1.0')
    implementation 'com.google.firebase:firebase-analytics'
    implementation 'com.google.firebase:firebase-auth'
    implementation 'com.google.firebase:firebase-database'
    implementation 'com.google.firebase:firebase-storage'
```

```
implementation 'de.hdodenhof:circleimageview:3.1.0'
// for converting image into string from int
    implementation
'com.github.bumptech.glide:glide:4.12.0'annotationProcessor
'com.github.bumptech.glide:compiler:4.12.0'

}
```

CHAPTER 6

TESTING

6.1 INTRODUCTION

6.1.1 Testing Objectives

The following are the testing objectives:

-Testing is a process of executing a program with the intent of finding an error.

-A good test case is one that has a high probability of finding an as-yet-undiscovered error

-successful test is one that uncovers an as yet undiscovered error.

6.1. 2 Testing Principles

The basic principles that guide software testing are as follows:

- -All tests should be traceable to customer requirements.
- -Tests should be planned long before testing begins.
- -The parate principle applies to software testing.

Pareto principle states that 80 percent of all errors uncovered during testing will likely betraceable to 20 percent of all program components.

Testing should begin “in the small “and progress toward testing “in the large.”

Exhaustive testing is not possible.

6.2 LEVEL OF TESTING

There are different levels of testing

->Unit Testing

->Integration Testing

->System Testing

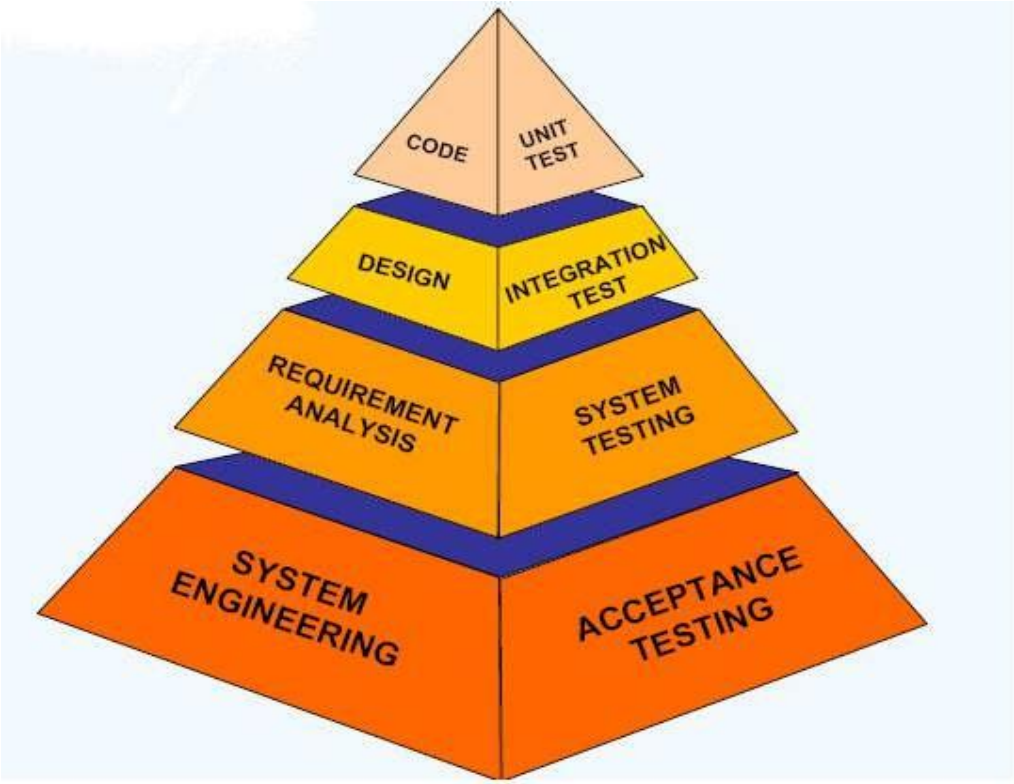


Figure 5.1: Testing pyramid

6.2.1 Unit testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The important control parts are tested to uncover with in theboundary of the module. The module interface is tested to ensure that the information properly flows into and out of the program unit and boundary conditions are tested to ensure that the modules operate properly at boundaries established to limit or restrict processing. Test date is provided through testing screens.

6.2.2 Integration testing

Integrating testing is a systematic technique for constructing Program structure while conducting tests to uncover error associates with interfacing. The objective isto take unit modules and built a program structure that has been directed by design.

• Integration Testing will test whether the modules work well together.

• This will check whether the design is correct.

6.2.3 System testing

System testing is the process of testing the completed software as a part of the environment itwas created for. It is done to ensure that all the requirements specified by the customer are met. System testing involves functional testing and performance testing.

• System Testing will contain the following testing:

- Functional Testing.
- Performance Testing.

• Function Testing will test the implementation of the business needs.

• Performance Testing will test the non-functional requirements of the system like the speed, load etc.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The main objective of the project is to develop a Secure Chat Application. We had taken a wide range of literature review to achieve all the tasks, where we came to know about some of the products that are existing in the market.

We made a detailed research in that path to cover the loopholes that existing systems are facing and to eradicate them in our application. In the process of research, I came to know about the latest technologies and different algorithms.

At last, we decided to make an Android Application using Realtime Firebase.

As a result, the product has been successfully developed in terms of extendibility, portability, and maintainability and tested in order to meet all requirements that are

- Authentication
- Integrity
- Confidentiality

With the knowledge I have gained by developing this application, I am confident that in the future I can make the application more effectively by adding these services.

- Extending this application by providing Authorization service.
- Creating Database and maintaining users.
- Increasing the effectiveness of the application by providing Voice Chat.
- Extending it to Web Support.

7.2 FUTURE SCOPE

The Future scope is to make the system more user friendly and enhanced.

- We will Add Status Uploading features in this.
- We have planned for Sending Reactions on the Messages.
- We will add the audio call feature in this.
- We are also planning to add single/group video call features in this application.
- Extending Web Support
- We will make our application more secure.
- We are planning to implement a strict user privacy policy

BIBLIOGRAPHY

1. Li Ma et al, Research and Development of Mobile Application for Android Platform, International Journal of Multimedia and Ubiquitous Engineering 9(4):187-198 • April 2014

2. Nikhil M. Dongre, Nikhil M. Dongre, Journal of Computer Engineering (IOSR-JCE), Volume 19, Issue 2, Ver. I (Mar.-Apr. 2017), PP 65-77

3. Javed Ahmad Shaheen et al, Android OS with its Architecture and Android Application with Dalvik Virtual Machine Review, International Journal of Multimedia and Ubiquitous Engineering Vol. 12, No. 7 (2017), pp. 19-30`

4. Sajid Nabi Khan, Ikhlaq Ul Firdous, Review on Android App Security, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 7, Issue 4, April 2017

5. April 2017 8. Lazarela Lazareska, Kire Jakimoski et al, Analysis of the Advantages and Disadvantages of Android and iOS Systems and Converting Applications from Android to iOS Platform and Vice Versa, American Journal of Software Engineering and Applications 2017; 6(5): 116-120

6. Bin Peng et al, The Android Application Development College Challenge, 2012 IEEE 14th International Conference on High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems, 18 October 2012
7. Shao Guo-Hong, Application Development Research Based on Android Platform,2014 7th International Conference on Intelligent Computation Technology and Automation, 08 January 2015

8. S Karthick, Android security issues and solutions, 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 13 July 2017

9. Pravin Auti, Sangam Mahale, Vikram Zanjad, Madhuri Dangat, n.d. An Android Based Global Chat Application. 4(1), pp. 1-2

10. Pravin Auti, Sangam Mahale, Vikram Zanjad, Madhuri Dangat, n.d. An Android Based Global Chat Application. 4(1)

11. S, A. K., n.d. Mastering Firebase for Android Development: Build real-time, scalable, and cloud-enabled Android apps with Firebase. s.l.: s.n

12. The books, which are referred, and which really helped me in building this system in time, are as follows: -

- Professional Android Application Development by Reto Meier.

- The Busy Coder's Guide to Advanced Android Development

There are some web sites also helped me in building this software. These are: -
https://developer.android.com/
https://www.javatpoint.com/android-tutorial
https://www.tutorialspoint.com/android/index.htm

GITHUB PROJECT LINK

https://github.com/HarshikaSrivastava/Chat2Talk