

SYNOPSIS
ON

**Android Messenger Application
"CHITCHAT"**

Department of Computer Engineering & Applications
Institute of Engineering & Technology



GLA University

Mathura- 281406

Submitted by :

Krishna Kant Sharma(181500332)
Nikhil Mohan(181500427)
Akashi Gupta(181500061)
Shailja Tripathi(181500648)
Palash Goyal (181500443)

Submitted to :

Neeraj Khanna
(Mentor)

CERTIFICATE

This is to certify that the project work under titled as

CHITCHAT

Is the bonfire work of

Nikhil Mohan

KrishnaKant Sharma

Shailja tripathi

Akashi gupta

Palash goyal

during this project submission as fulfillment of the requirement for the System Design Project of Bachelor in Technology V Semester, of the **GLA University Mathura**,Project

Submitted by :

Krishna Kant Sharma(181500332)

Nikhil Mohan(181500427)

Akashi Gupta(181500061)

Shailja Tripathi(181500648)

Palash Goyal (181500443)

Submitted to :

Neeraj Khanna

(Mentor)

ACKNOWLEDGEMENT

If words are considered as a symbol of approval and token of appreciation then let the words play the heralding role expressing my gratitude. The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. We are grateful to our project guide Ms. Sangeeta Solanki for the guidance, inspiration and constructive suggestions that helped us in the preparation of this project. We also thank our colleagues who have helped in successful completion of the project.

Krishna Kant Sharma(181500332)

Nikhil Mohan(181500427)

Akashi Gupta(181500061)

Shailja Tripathi(181500648)

Palash Goyal (181500443)

Certificate of Completion

*This is to certify that **Nikhil** successfully
completed 37.5 total hours of **The Complete
Android Oreo Developer Course - Build 23 Apps!**
online course on June 6, 2020*

Rob Percival

Rob Percival, Instructor

Nick Walter

Nick Walter, Instructor

Codestars by Rob Percival

Codestars by Rob Percival, Instructor

&



Certificate no: UC-aa11e1aa-0e99-4d49-a880-cdd76f01afed
Certificate url: ude.my/UC-aa11e1aa-0e99-4d49-a880-cdd76f01afed

#BeAble

Certificate of Completion

***This is to certify that Shailja Tripathi successfully
completed 37.5 total hours of The Complete
Android Oreo Developer Course - Build 23 Apps!
online course on June 6, 2020***

Rob Percival

Rob Percival, Instructor

Nick Walter

Nick Walter, Instructor

Codestars by Rob Percival

Codestars by Rob Percival, Instructor

&



Certificate no: UC-aa11e1aa-0e99-4d49-a880-cdd76f01afed
Certificate url: [ude.my/UC-aa11e1aa-0e99-4d49-a880-cdd76f01afed](https://udemy.my/UC-aa11e1aa-0e99-4d49-a880-cdd76f01afed)
Version 2

#BeAble

Certificate of Completion

***This is to certify that **Palash Goyal** successfully
completed 37.5 total hours of **The Complete
Android Oreo Developer Course - Build 23 Apps!**
online course on June 6, 2020***

Rob Percival

Rob Percival, Instructor

Nick Walter

Nick Walter, Instructor

Codestars by Rob Percival

Codestars by Rob Percival, Instructor

&



Certificate no: UC-aa11e1aa-0e99-4d49-a880-cdd76f01afed
Certificate url: [ude.my/UC-aa11e1aa-0e99-4d49-a880-cdd76f01afed](https://udemy.com/certificate/UC-aa11e1aa-0e99-4d49-a880-cdd76f01afed)
Version 2

#BeAble

Certificate of Completion

***This is to certify that Akashi Gupta successfully
completed 37.5 total hours of The Complete
Android Oreo Developer Course - Build 23 Apps!
online course on June 6, 2020***

Rob Percival

Rob Percival, Instructor

Nick Walter

Nick Walter, Instructor

Codestars by Rob Percival

Codestars by Rob Percival, Instructor

&



Certificate no: UC-aa11e1aa-0e99-4d49-a880-cdd76f01afed
Certificate url: [ude.my/UC-aa11e1aa-0e99-4d49-a880-cdd76f01afed](https://udemy.com/UC-aa11e1aa-0e99-4d49-a880-cdd76f01afed)
Version 2

#BeAble

Certificate of Completion

This is to certify that *Krishna kant sharma* successfully completed 37.5 total hours of *The Complete Android Oreo Developer Course - Build 23 Apps!* online course on June 6, 2020

Rob Percival

Rob Percival, Instructor

Nick Walter

Nick Walter, Instructor

Codestars by Rob Percival

Codestars by Rob Percival, Instructor

&



Certificate no: UC-aa11e1aa-0e99-4d49-a880-cdd76f01afed
Certificate url: [ude.my/UC-aa11e1aa-0e99-4d49-a880-cdd76f01afed](https://www.udemy.com/certificate/UC-aa11e1aa-0e99-4d49-a880-cdd76f01afed)
Version 2

#BeAble

CONTENT

Abstract	3
1. Introduction	3
1.1 General Introduction to the topic	4
1.2 Area of Computer Science	5
1.3 Hardware and Software Requirements	6
2. Problem definition	7
3. Objectives	8
4. Implementation Details	11
5. Progress till Date & The Remaining work	11
6. Some Screenshots	12
7. References	13

Abstract

Teleconferencing or Chatting, is a method of using technology to bring people and ideas together despite geographical barriers. The technology has been available for years but the acceptance was quite present. Our project is an example of a chat application. It is made up of various technologies, which runs on the user's Android device with perfection . To start chatting clients should get connected to the server via the internet where they can do private and group chat security measures were taken during the last one. There are various examples of successful chat applications like whatsapp, messenger, instagram snapchat etc. Hence it is also a presentable application.

1. Introduction

Android is a Linux based operating system designed primarily for touch screens mobile devices such as smartphones and tablet computers. The operating system has developed a lot in the last 15 years starting from black and white phones to recent smartphones or mini computers. One of the most widely used mobile OS these days is android. The android is software that was founded in Palo Alto of California in 2003.

Communication is a means 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, for 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 effort has been drawn towards consolidating the device into one and therefore indiscriminate the services.

1.1 General Introduction

Our project is an example of chat application which is basically based on public chatting (message is broadcast to all connected users) and private person to person chat

It is made of two applications

Client application which runs on user's Android device

Server application which runs Google firebase servers

To start chatting client should get authenticated to servers

Q.What is a Chat application?

Online chat is a less stringent definition may be primarily and direct text-based or video-based (webcams), one-on-one or one-to-many group chat (formerly also known as synchronous conferencing), using tools such as instant messengers ,Internet relay Chat (IRC) ,talker and possibly MUDs.

Q.What is Firebase in android?

Real time Database: Firebase provides a real time database and backend as a service .The service provides applications developers an API that allows application data to be synchronized across clients and stored on Firebase cloud.

Firebase Analytics is a cost-free app measurement solution that provides insight into app usage and user engagement.

Resources

Gradle build tool : -

Gradle is a build automation tool for multi-language software development. It controls the development process in the tasks of compilation and packaging to testing, deployment, and publishing. Supported languages include Java (Kotlin, Groovy, Scala) , C/C++, JavaScript.

Gradle builds on the concepts of Apache Ant and Apache Maven, and introduces a Groovy- & Kotlin-based domain-specific language contrasted with the XML-based project configuration used by Maven.^[3] Gradle uses a directed acyclic graph to determine the order in which tasks can be run, through providing dependency management.

Gradle was designed for multi-project builds, which can grow to be large. It operates based on a series of build tasks that can run serially or in parallel. Incremental builds are supported by determining the parts of the build tree that are already up to date; any task dependent only on those parts does not need to be re-executed. It also supports caching of build components, potentially across a shared network using the Gradle Build Cache.

It produces web-based build visualization called Gradle Build Scans. The software is extensible for new features and programming languages with a plugin subsystem.

Maven : -

Maven is a build automation tool used primarily for Java projects. Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project.

Maven addresses two aspects of building software: how software is built, and its dependencies. Unlike earlier tools like Apache Ant, it uses conventions for the build procedure, and only exceptions need to be written down. An XML file describes the software project being built, its dependencies on other external modules and components, the build order, directories, and required plug-ins. It comes with pre-defined targets for performing certain well-defined tasks such as compilation of code and its packaging. Maven dynamically downloads Java libraries and Maven plug-ins from one or more repositories such as the Maven 2 Central Repository, and stores them in a local cache.[2] This local cache of downloaded artifacts can also be updated with artifacts created by local projects. Public repositories can also be updated.

Maven is built using a plugin-based architecture that allows it to make use of any application controllable through standard input. A plugin for the .NET framework exists and is maintained and a C/C++ native plugin is maintained for Maven

Alternative technologies like Gradle and sbt as build tools do not rely on XML, but keep the key concepts Maven introduced. With Apache Ivy, a dedicated dependency manager was developed as well that also supports Maven repositories.

Xml and android layouts : -

Each layout file must contain one (and only one!) root element. Linear Layouts, Relative Layouts, and Frame Layouts (see Root Views section below) may all be root elements. Other layouts may not be. All other XML elements will reside within this root object.

A View is simply an object from Android's built-in View class. It represents a rectangular area of the screen, and is responsible for displaying information or content, and event handling. Text, images, and buttons are all Views in Android.

A ViewGroup is a subclass of View, and is essentially an 'invisible container' that holds multiple Views or ViewGroups together, and defines their layout properties.

Root Views

The following three layout types may be the root element in an Android XML Layout:

- A Linear Layout aligns its contents into a single direction, whether vertical or horizontal.
- A Relative Layout displays its child content in positions relative to the parent. (ie: lining an element up to the top edge of a parent, centering it within a parent, etc.)
- A Frame Layout is a placeholder on a screen that can display only a single view. (For this reason, Frame Layouts should be used sparingly; usually as a placeholder for Fragments, which we will cover later.)

Common ViewGroups

Other commonly used ViewGroups are as follows. These may not be the root element of a layout, but they may reside within the root element:

- A List View displays a list of scrollable items.
- A Grid View displays items in a two-dimensional, scrollable grid.
- A Table Layout groups views into rows and columns.

Layout Attributes

Every type of layout has attributes that define the way its elements appear. There are both common attributes that *all* layouts share, and attributes specific to some of the layout types listed above. The following are attributes that apply to all layouts:

- `android:id`: A unique ID that corresponds to the view.
- `android:layout_width`: The width of the layout. (required for every view)
- `android:layout_height`: The height of the layout. (required for every view)
- `android:layout_marginTop`: Extra space on the top of the layout.
- `android:layout_marginBottom`: Extra space on the bottom of the layout.
- `android:layout_marginLeft`: Extra space to the left of the layout.
- `android:layout_marginRight`: Extra space to the right of the layout.
- `android:layout_weight`: Specifies how much of the extra space in the layout should be allocated to the view.
- `android:paddingLeft`: Padding to the left of the view.
- `android:paddingRight`: Padding to the right of the view.
- `android:paddingTop`: Padding at the top of the view.
- `android:paddingBottom`: Padding at the bottom of the view.

Relative Sizing

Height and width properties can be set to specific measurements, but the following are much more common:

- `android:layout_width=wrap_content`: Sets the width of the view to whatever size is required by its contents. This may also be used with height.
- `android:layout_width=match_parent`: Sets the width of the view to the width of its parent. This may also be used with height.

Setting Properties and IDs

Much like in HTML, we can also add id attributes to XML elements. You'll often need to do this in order to later locate and interact with a specific view. However, the syntax for assigning an ID differs from that in HTML.

Assigning an id attribute looks something like this:

```
android:id="@+id/search_button"
```

Here, the `@+id/` portion indicates that the name following is a string containing this element's ID, and that it is a new resource that should be created and added to our resources.

For instance, the following XML will create a Button with the id search_button:

```
<Button android:id="@+id/search_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/my_button_text"/>
```

Targeting Views by ID

Similar to the way we used jQuery to target individual HTML elements and alter them in some fashion, we can also target individual XML elements and interact with them. The following code can be used to target the example XML button from above:

```
Button myButton = (Button) findViewById(R.id.search_button);
```

The findViewById() method can locate both Views and root views. Don't worry too much about this specific syntax *quite* yet; we'll explore this more in future lessons. Just notice that we're re-using the same id attribute we assigned in the XML to later locate this element in our Java logic.

In the next lesson, we'll walk through creating an Android XML layout from scratch in our MyRestaurants application.

Note on Constraint Layouts

Newer versions of Android Studio may create default layouts called ConstraintLayouts instead of LinearLayouts or RelativeLayouts. ConstraintLayouts are a newer layout format that is more powerful, but also more complex for beginners to work with.

In the next lesson we'll walk through how to turn a default ConstraintLayout *back* into a Linear or Relative layout, so you may gain practice with these standard layouts in your own application. Then later this week, when we become more and more comfortable with layouts in general, we'll learn how to create and use Constraint layouts too!

Keep in mind that just because a newer option, feature, or version of a tool or technology exists, doesn't mean all companies and products immediately migrate to the newer

option! Especially if older versions or features are still successfully serving their needs. This goes both for Android layouts, and *any other* web development technology. You'll undoubtedly still see these standard layouts out in the field. So, to best prepare for your internship and eventual employment, it's important to learn both the original *and* new layout options.

1.2 Area of Computer Science

This application is developed for android users by using various android open source and closed source libraries including RESTful Firebase Api The application server runs on Firebase server which provides functionality for authentication,uploadation, downloads Java is used in backend and for front end application is created by xml default editor in android studio.

1.3.1 Hardware Requirements

Ram : 8Gb,4Gb(min)
Processor : i3 5th generation or higher
Android phone for debugging
Data Cable

1.3.2 Software Requirements

- Android Studio
- Android emulator
- Firebase Sdk
- Android Sdk
- Picasso Library
- Firebase Ui Library
- Default android library
- Browser
- Android version 9.0 or higher
- Internet connection

2.Problem Definition

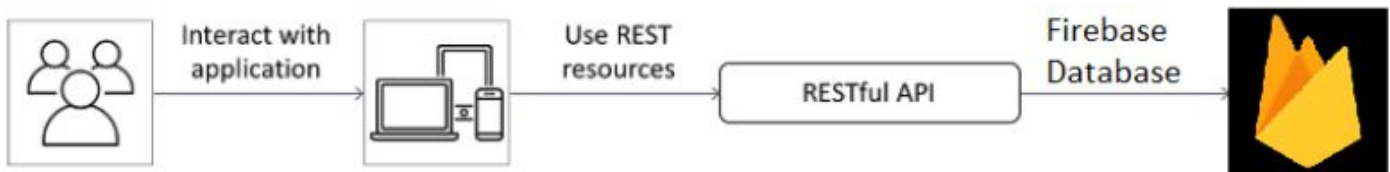
This project is to create a chat application with a Firebase server, and to enable the users to chat and To develop an instant messaging solution to enable users to seamlessly communicate with each other. The project should be very easy and handy to use.

3.Objective

The objective is to build an android application fully in working condition by considering the following details :

- 1.Private chat
- 2.Group chat
- 3.Profile and status Uploading and viewing
4. Accepting and sending request
5. Saving contact if request Accepted
- 6.Uploading and changing profile image
- 7.Finding Friends
- 8.Last seen and online status

4.Implementation Details



Part 1: To allow User to authenticate with Firebase

- Authentication with phone number

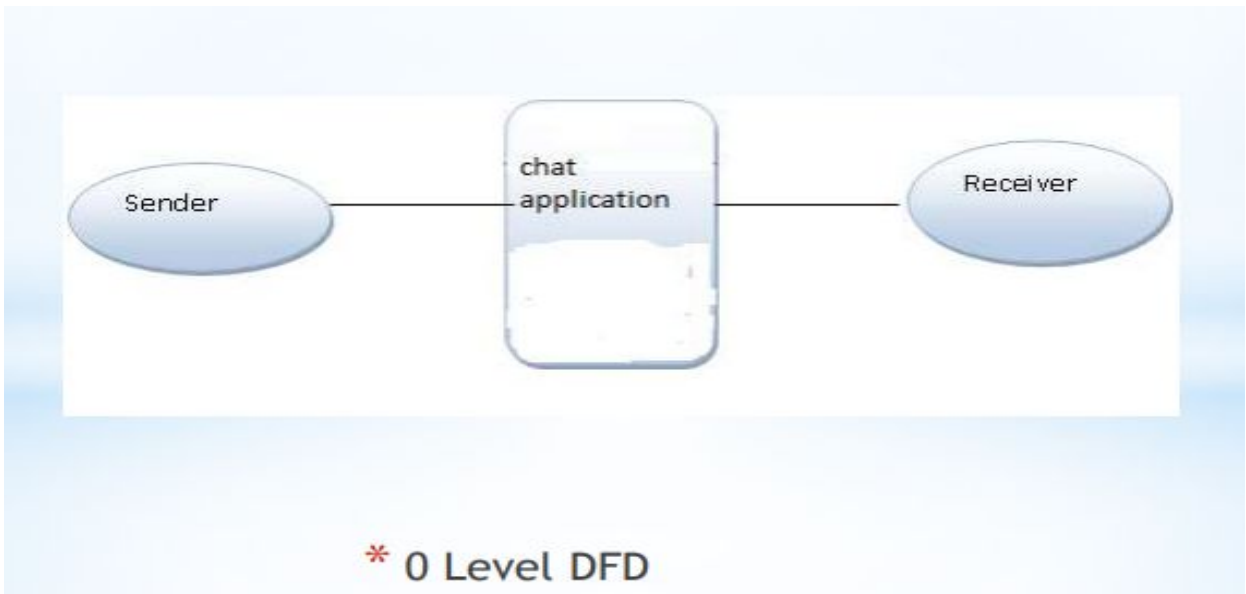
- Authentication with email

- Regular expression for checking

- Authentication with phone pre otp receiving

- Authentication with phone post otp receiving

- Timer implementation for otp time out(120 sec)



Part 2: Design Setting profile

Using **“de.hdodenhof.circleimageview.CircleImageView”** for circular imageView

Firestore Realtime database implementation for user info tracking

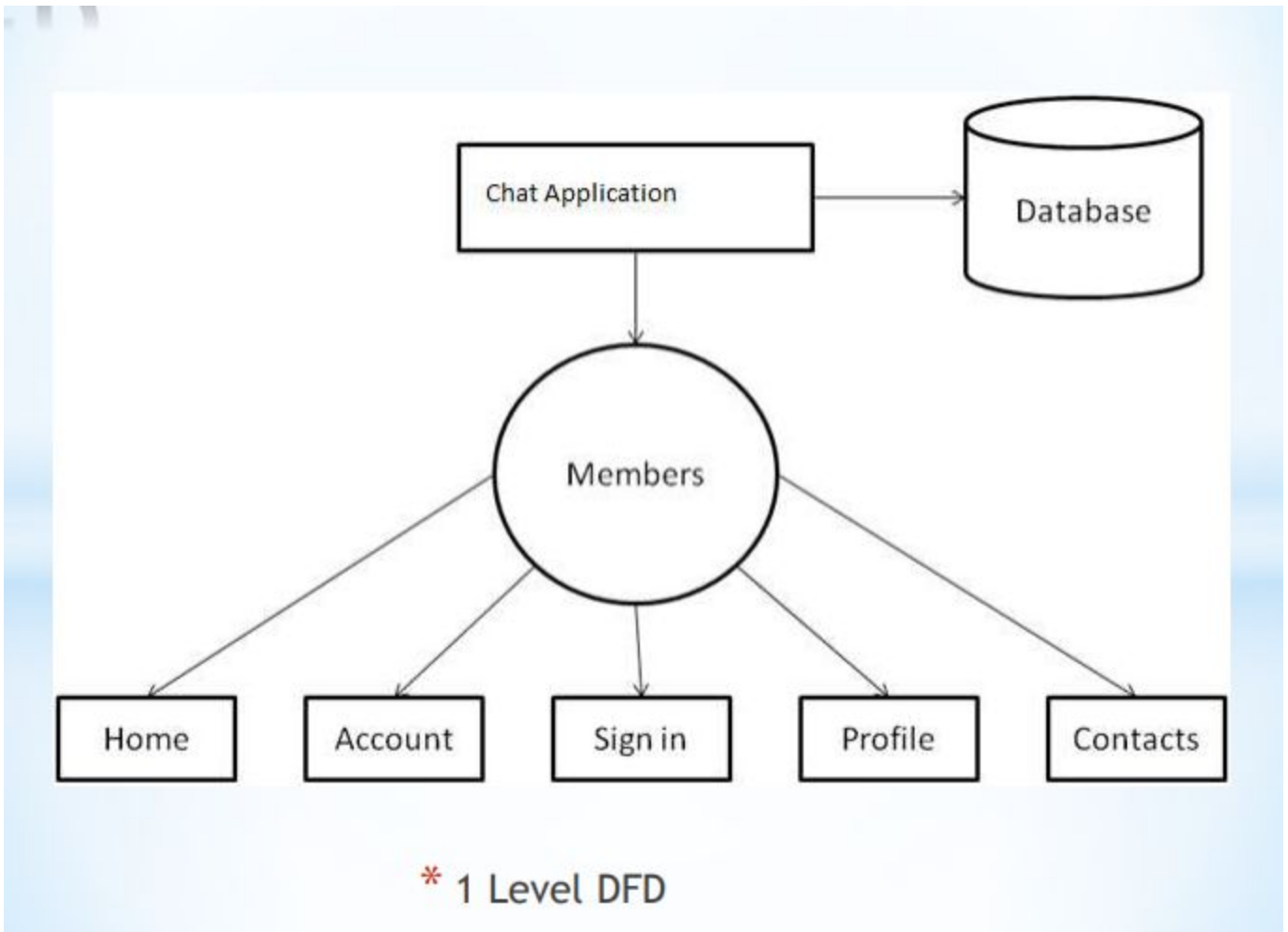
Firestore Database Reference implementation for user profile image upload

Picasso Library for retrieving and placing user profile image by url

Implementing “com.theartofdev.edmodo.cropper” for crop image activity

Create Public Group feature for broadcasting messages to all user

Send and cancel chat request to any other user with username and user status retrieval



Part 3: Request Fragment and contact Fragment

Using FirebaseUi Library

Sended request will appear here with status request sent

Received request will appear here with options accept and decline

Request deleted from either side will delete it from both side

Accepted request will be saved to contacts

Direct chat option when clicked on saved contacts

Added contacts request will be removed



Part 4: Chat and Group Fragment

Group Fragment will contain all created group

Clicking on group name will send you to group chat activity for particular group

Send and receive messages will appear in format as

“Date : time

Message

Sender name”

Chat fragment will contain all contacts either you have messaged or not

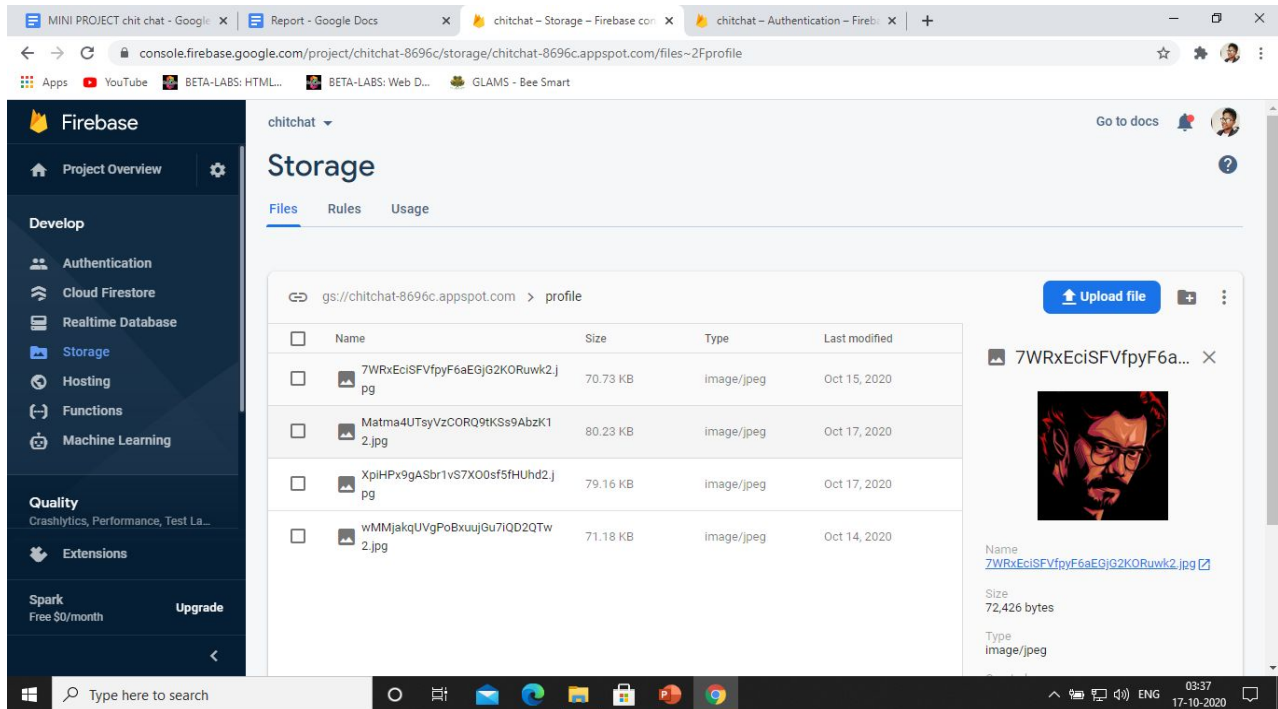
Chat fragment will show users last seen **“Last seen date : time”**

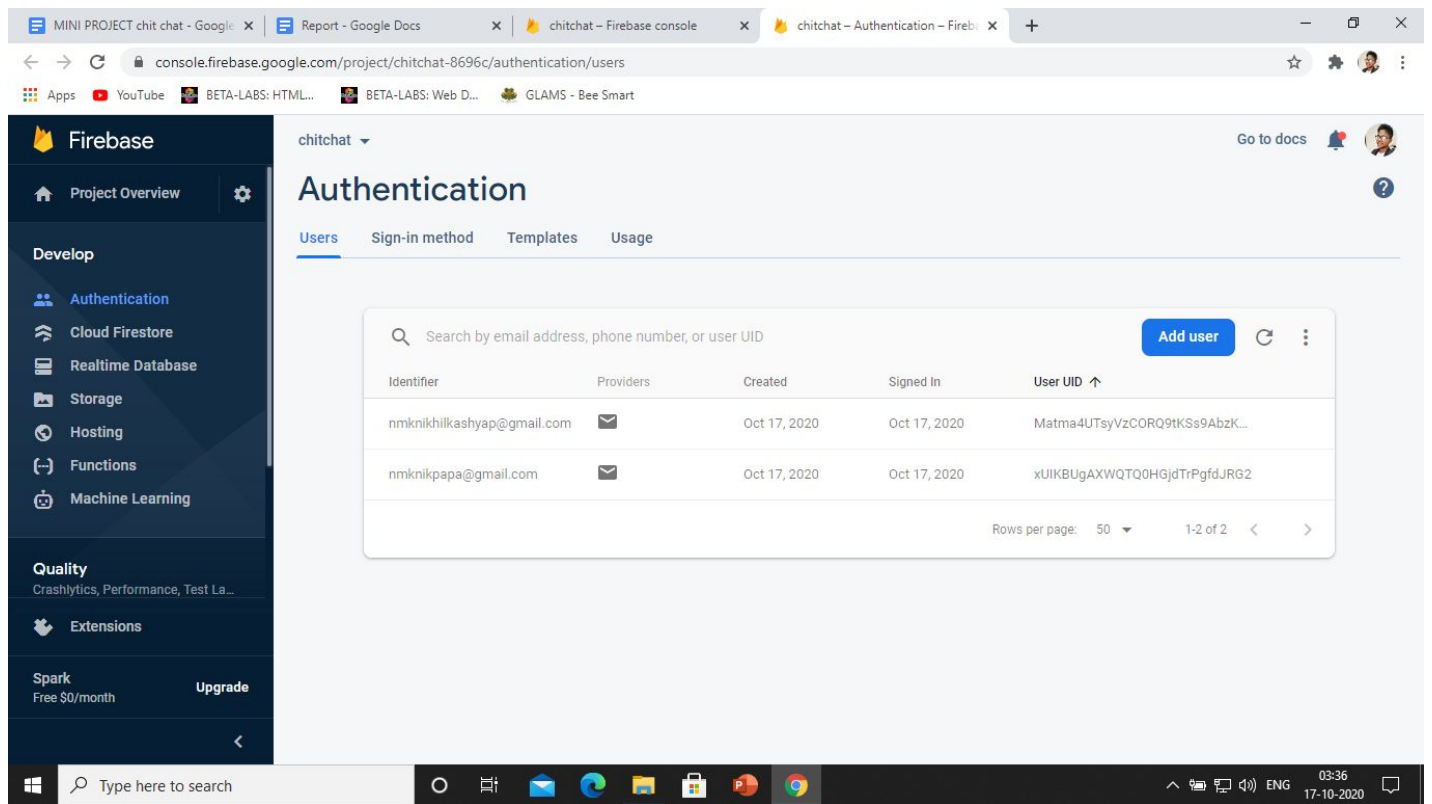
Tracking online Status

5.Progress

Work	Date
Part-1	Completed 20 Aug 2020
Part-2	Completed 20 sept 2020
Part-3	Completed(20-oct-2020)
Part-4 and final submission	Completed(20-Nov-2020)

6.ScreenShots





3:43 P



nik1

code

REMOVE FRIEND

3:44 P

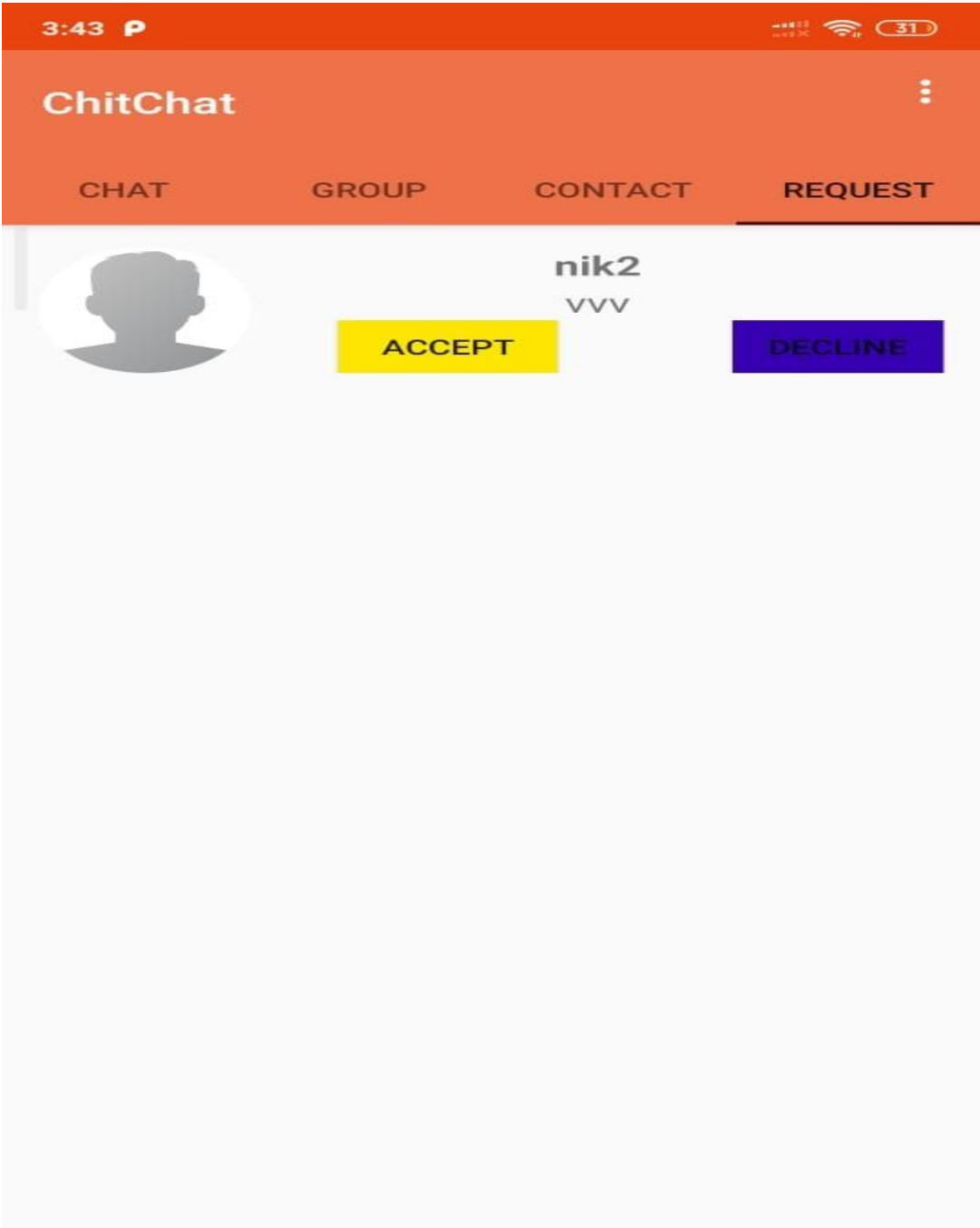


nik2

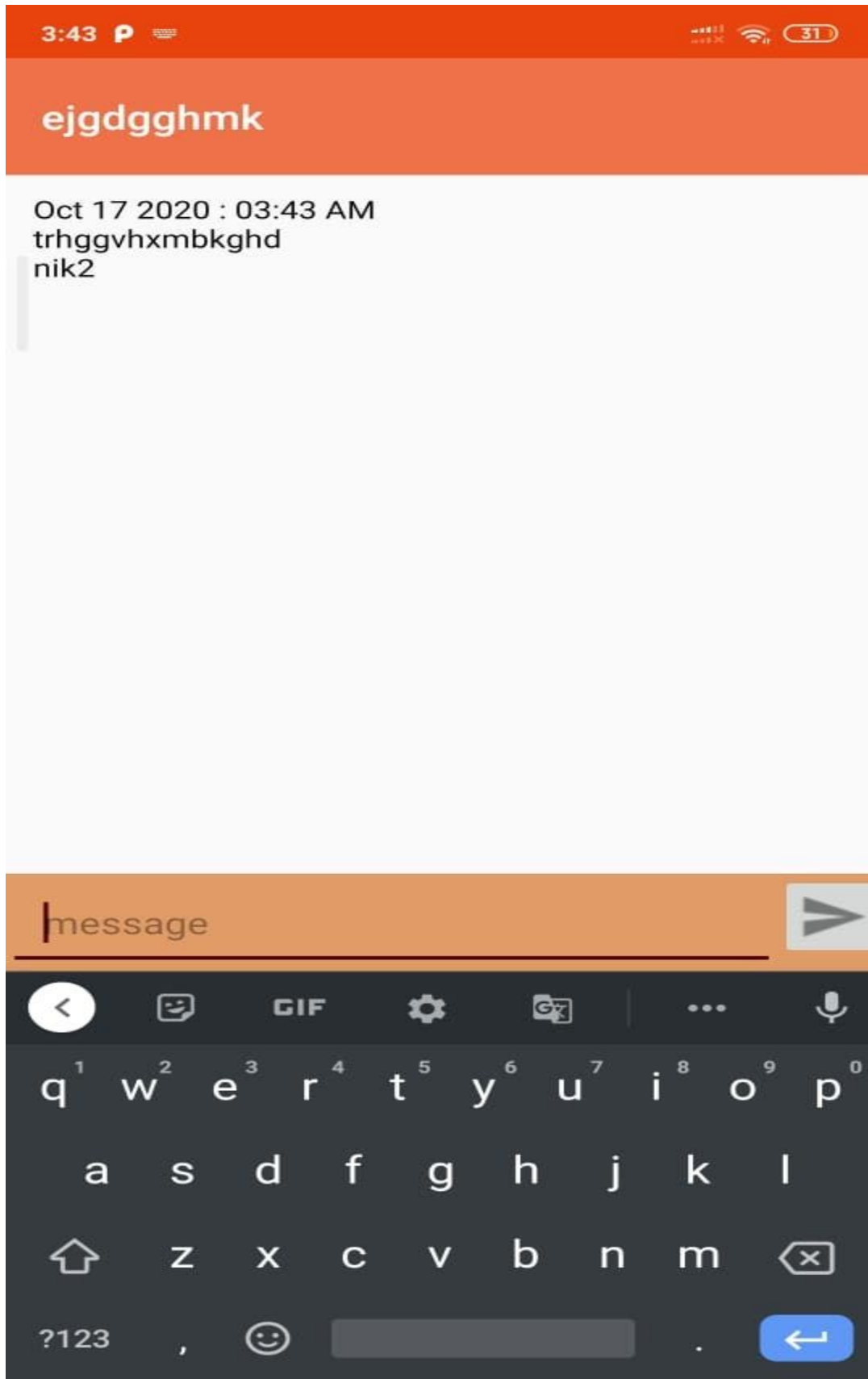
VVV

ACCEPT CHAT REQUEST

DECLINE CHAT REQUEST







Find Friends



nik1
code



nik2
vvv

ChitChat



CHAT

GROUP

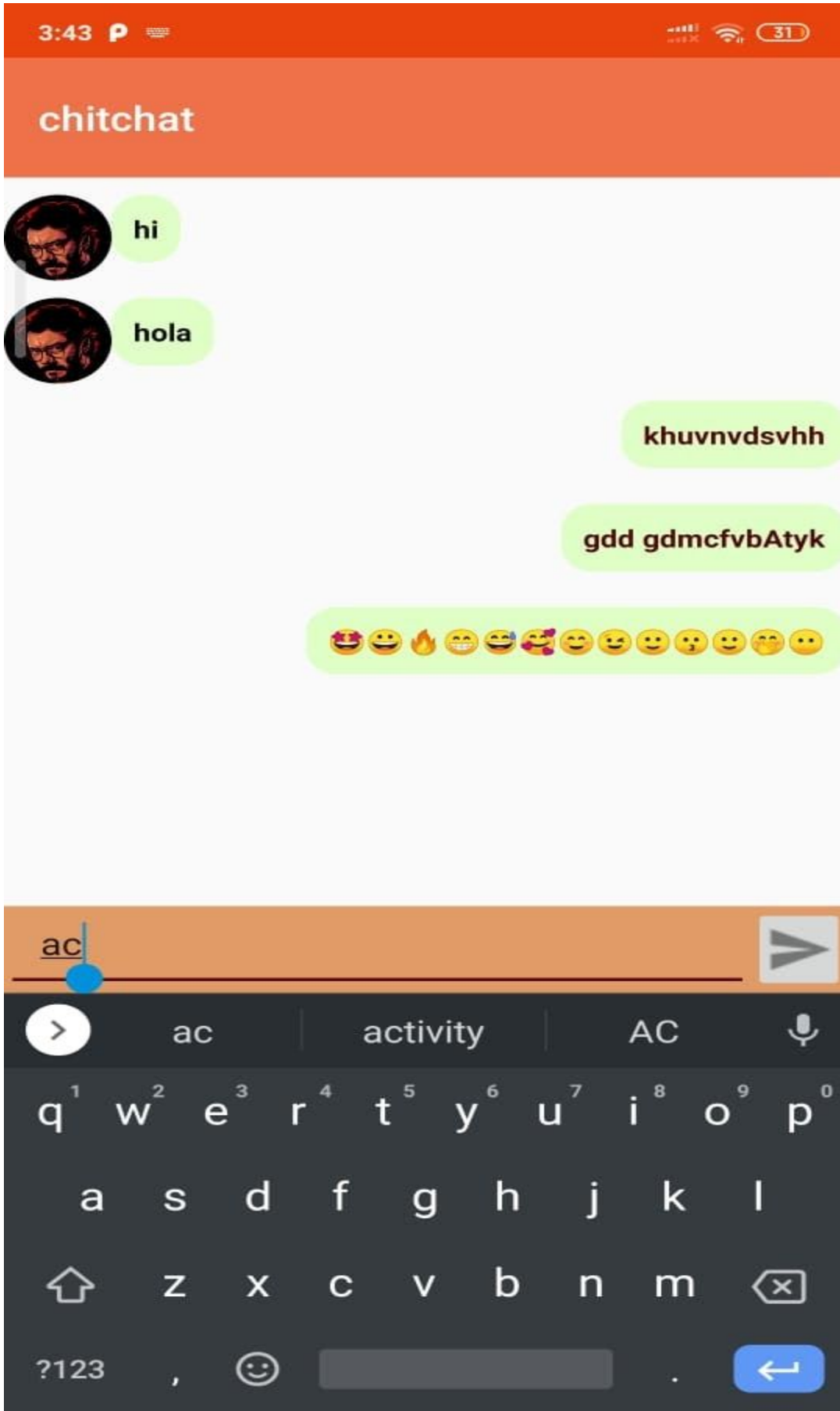
CONTACT

REQUEST



nik1

Last seen: Oct 17 2020
03:42 AM





code

UPDATE

7.References

- [Android developers](#)
- [code with uncle bob \(blog\)](#)
- [Stack Overflow](#)
- [view chit-chat on github \(repository link\)](#)
- [Firebase Ui](#)
- [picasso library](#)
- [Circle imageView](#)
- [Firebase](#)
- [Chitchat Apk click here to download](#)
- **Book reference Android Programming for beginners**