

**SARANATHAN COLLEGE OF ENGINEERING**

Panjappur, Tiruchirappalli – 620012

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**MINI PROJECT NOTE BOOK**

**CS8611 / MINI PROJECT**

**APRIL 2020**

**CHAT APPLICATION**

**A ANDROID APPLICATION FOR MESSAGING**

**A MINI PROJECT REPORT**

***Submitted by***

**VIJETHA J (813817104111)**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

**in**

**COMPUTER SCIENCE AND ENGINEERING**



**SARANATHAN COLLEGE OF ENGINEERING, TRICHY**



**ANNA UNIVERSITY : CHENNAI 600 025**

**APRIL 2020**

**ANNA UNIVERSITY : CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this Mini project (CS8611) report **“CHAT APPLICATION -A ANDROID APPLICATION FOR MESSAGING”** is the bonafide work of

**VIJETHA J (813817104111)**

who carried out the Mini-project work under my supervision.

SIGNATURE SIGNATURE

**Dr. S.A. Sahaaya Arul Mary, M.E., Ph.D Mr. P.B. Arun Prasad M.E.,** HEAD OF THE DEPARTMENT ASSISTANT PROFESSOR Computer Science and Engineering, Computer Science and Engineering, Saranathan College of Engineering, Saranathan College of Engineering, Panjapur, Panjapur,

Tiruchirapalli-620 012 Tiruchirapalli-620 012

**VIVA - VOCE EXAMINATION**

**CHAT APPLICATION - A ANDROID APPLICATION FOR MESSAGING**

*Submitted by*

**VIJETHA J (813817104111)**

The Viva - Voce Examination of this Mini-Project (CS8611) work done as a part of B.E. Computer Science and Engineering was held on .

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**SARANATHAN COLLEGE OF ENGINEERING**



**Venkateswara Nagar ,Panjappur**

**Tiruchirapalli-620012**

**VISION OF THE INSTITUTION**

Impart an inclusive engineering education that beyond being a facilitator for a career and rudimentary skills, equips the students to offer ethically & environmentally conscious solutions

to societal issues.

**MISSION OF THE INSTITUTION**

Develop the Institution into a Model Self Financing College for Engineering and Technology. Deliver Professional Training to our students with state-of-the art laboratories and converting them into Technocrats of international repute.

1. Create a nurturing, holistic environment of engineering education to facilitate every student realize their full potential.

2. Strive to make the students strong in basic concepts armed with appropriate skills to

enhance one’s ability to apply the knowledge to provide solutions to real time issues.

3. Maintain an ambience that facilitates the students to strengthen their ethical value systems.

4. Actively promote R&D and institute-industry interaction.

**VISION OF THE DEPARTMENT**

 To evolve as a centre of academic excellence and advanced research in Computer

Science and Engineering discipline.

**MISSION OF THE DEPARTMENT**

 To inculcate in students a profound understanding of fundamentals related to discipline.

 To inculcate skills, attitudes and their applications in solving real world problems with an inclination towards societal issues and research.

 To promote research in the emerging areas of computer science and technology

**PROGRAM EDUCATIONAL OBJECTIVES (PEOS)**

**PEO1** : Acquire strong foundation in the mathematical, scientific and engineering fundamentals necessary to formulate, solve and analyze engineering problems.

**PEO2** : Develop the ability to analyze the requirements of the software, understand the technical specifications, design and provide novel engineering solutions and efficient software/hardware designs.

**PEO3** : Have exposure to emerging cutting edge technologies, adequate training & opportunities to work as teams on multidisciplinary projects with effective communication skills and leadership qualities.

**PEO4** : Have awareness on the life-long learning and prepare them for research development and consultancy.

**PEO5:** Have a successful career and work with values & social concern bridging the digital

divide and meet the requirements of Indian and multinational companies.

**PROGRAM SPECIFIC OUTCOME (PSO)**

1. **PS01**: Foundation of mathematical concepts: To use mathematical methodologies to crack problem using suitable mathematical analysis, data structure and suitable algorithm.

2. **PSO2**: Foundation of Computer System: the ability to interpret the fundamental concepts and methodology of computer systems. Students can understand the functionality of hardware and software aspects of computer systems.

3. **PSO3**: Foundations of Software development: the ability to grasp the software development lifecycle and methodologies of software systems. Possess competent skills and knowledge of software design process. Familiarity and practical proficiency with a broad area of programming concepts and provide new ideas and innovations towards research.

**Program Outcomes: (POs)**

|  |  |
| --- | --- |
| **Sno** | **Program Outcomes** |
| **1** | **PO1 Engineering knowledge:**  Apply the knowledge of mathematics, science, engineering fundamentals, and engineering specialization to the solution of complex engineering problems. |
| **2** | **PO2 Problem Analysis:**  Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. |
| **3** | **PO3 Design/development of solutions:**  Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. |
| **4** | **PO4 : Conduct Investigations of Complex Problems** Use research-based knowledge and  research methods including design of exercises, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| **5** | **PO5 : Modern Tool Usage** : Create, select, and apply appropriate techniques, resources, and  modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations. |
| **6** | **PO6 : The Engineer and Society** Apply reasoning informed by the contextual knowledge to  assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice. |
| **7** | **PO7: Environment and sustainability:** Understand the impact of the professional  engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. |
| **8** | **PO8 : Ethics**  Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice. |
| **9** | **PO9: Individual and Team Work**  Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings. |
| **10** | **PO10 : Communication**  Communicate effectively on complex engineering activities with the engineering community |

|  |  |
| --- | --- |
|  | and with society at large, communication such as, being able to comprehend and write  effective reports and design documentation, make effective presentations, and give and receive clear instructions. |
| **11** | **PO11 : Project management and finance**  Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. |
| **12** | **PO12 :Life-long learning**  Recognize the need for, and have the preparation n and ability to engage in independent and life-long learning in the broadest context of technological change |

**NBA Code for the Subject: C319 Subject Code: CS8611**

**Title: MINI PROJECT**

**COURSE OUTCOMES:** At the end of this course, Student will be able to

|  |  |
| --- | --- |
| **Course Code** | **Course outcome Description** |
| **C319.1** | **Gather and interpret technical literature to formulate a project proposal to solve challenging practical problems.** |
| **C319.2** | **Identify SDLC model and prepare software requirements specification.** |
| **C319.3** | **Design the software architecture.** |
| **C319.4** | **Apply modern tools for implementation using best coding practices and.testing at various levels of the project.** |
| **C319.5** | **Document the technical report on identified topic and present the ideas with effective communication skills** |
| **C319.6** | **Learn the concepts of project management and to work effectively as a member in team.** |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SUBJECT CODE : CS8611 NBA CODE : C319 TITLE : MINI PROJECT** | | | | | | | | | | | | | | | |
| **CoCode** | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** | **PSO1** | **PSO2** | **PSO3** |
| **C319.1** | **2** | **2** | **2** | **2** | **2** | **2** | **-** | **1** | **2** | **2** | **2** | **2** | **2** | **2** | **2** |
| **C319.2** | **2** | **2** | **2** | **2** | **2** | **2** | **-** | **1** | **2** | **2** | **2** | **2** | **2** | **2** | **2** |
| **C319.3** | **2** | **2** | **2** | **2** | **2** | **2** | **-** | **1** | **2** | **2** | **2** | **2** | **2** | **2** | **2** |
| **C319.4** | **2** | **2** | **2** | **2** | **2** | **2** | **-** | **1** | **2** | **2** | **2** | **2** | **2** | **2** | **2** |
| **C319.5** | **2** | **2** | **2** | **2** | **2** | **2** | **-** | **1** | **2** | **2** | **2** | **2** | **2** | **2** | **2** |
| **C319.6** | **2** | **2** | **2** | **2** | **2** | **2** | **-** | **1** | **2** | **2** | **2** | **2** | **2** | **2** | **2** |

**TABLE OF CONTENTS**

1.Abstract------------------------------------------------------------------------------ 1

2.Introduction-------------------------------------------------------------------------- 2

3.Proposed System-------------------------------------------------------------------- 4

4.System Architecture---------------------------------------------------------------- 5

5.Usecase diagram-------------------------------------------------------------------- 6

6.Requirements------------------------------------------------------------------------ 7

6.1 Functional requirements------------------------------------------------- 7

6.2 Software requirements--------------------------------------------------- 8

6.3 Non-Functional requirements------------------------------------------- 8

6.3.1 Security---------------------------------------------------------- 8

6.3.2 Reliability------------------------------------------------------- 8

6.3.3 Availability----------------------------------------------------- 8

6.3.4 Maintainability------------------------------------------------- 8

6.4 Interface requirements---------------------------------------------------- 8

6.4.1 GUI--------------------------------------------------------------- 8

6.4.2 Hardware interface--------------------------------------------- 8

6.4.23 Software interface--------------------------------------------- 9

6.5 Performance requirements----------------------------------------------- 9

6.6 Operational scenarios----------------------------------------------------- 9

7.Module explanation------------------------------------------------------------------ 10

7.1 Authentication(module 1)------------------------------------------------ 10

7.2 Read receipt(module 2)--------------------------------------------------- 10

7.3 Sending message(module 3)---------------------------------------------- 10

8.Testcases------------------------------------------------------------------------------- 11

9.Conclusion----------------------------------------------------------------------------- 12

10.Screenshots--------------------------------------------------------------------------- 13

11.Coding--------------------------------------------------------------------------------- 15

**ABSTRACT**

Teleconferencing or Chatting is a method of using technology to bring people and ideas together despite of geographical barriers. The communication brings people together, closer to each other. The communication is an important management function closely associated with all other managerial functions. It bridges the gap between individuals and groups through flow of information and understanding between them. Chatting on the Internet comes in many forms. You can have one-on-one chatting via instant messaging software. You can find it less expensive to chat over the Internet than to make a long distance call. This makes Internet chatting an attractive option if you live far away from family and friends. You can complete other tasks while chatting over the Internet. Chat messages are generally short in order to enable other participants to respond quickly.

The technology has been available for years but the acceptance it was quite recent. Our project is an example of a chat server .It is made up of 2 applications the client application, which runs on the Android device and server application which runs any Android Device on the network. To start chatting client should get connected to server where they can do private chat security measures were taken during the last one.

Keywords:

* Message
* Chatting
* Calling
* Personal Chat
* Group Chat

**2. INTRODUCTION**

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 s 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 on the Internet comes in many forms. You can have one-on-one chatting via instant messaging software.You can find it less expensive to chat over the Internet than to make a long distance call. Chatting is a method of using technology to bring people and ideas together despite of geographical barriers. The technology has been available for years but the acceptance it was quite recent. Our project is an example of chat server. It is made up of application the client application which runs on user mobile and server application which runs on any pc on the network. To start chatting our client should be connected to server where they can do group chatting and private chatting.

Our project is to create a chat application with a server and users to enable the users to chat with each other. To develop an instant messaging solution to enable users to seamlessly communicate with each other.

The project should be very easy to use enabling even a novice person to use it.

**2.1 THE APPLICATION WORKS AS FOLLOWS**

Once the user downloads the application, they can open and enter their respective phone number and they will be authenticated with the help of OTP. The users will be able to send messages to their recipient by saving their contacts details in their mobile phone. The users will be able to see whether the other contacts are in online. The service requires users to provide a standard [cellular](https://en.wikipedia.org/wiki/Cellular_network) [mobile number](https://en.wikipedia.org/wiki/Telephone_number) for registering with the service. In Chatapp, the ticks will appear neat to every message you send. The user first login using their respective username and password, which is authenticated using the firebase. The sender sends the message which is stacked in the firebase and is sent to correct receiver using their id. The firebase also maintains the chat history of each and every user. The read receipt is shown to the user.

A single grey tick means it's been sent successfully, a second grey tick means it's been delivered to the other person's phone. The double blue ticks inform you that the message has been read by the recipient.

'Online' and 'last seen' simply tell you if your contacts are online, or the last time they were using ChatApp. Online means that the person has ChatApp open in the foreground on their device and is connected to the internet. This doesn't necessarily mean they will have read any message you've sent them.

**3. PROPOSED SYSTEM**

The main purpose of this Messenger is to have a secured chat and media exchange which now a days has become a challenge in most of the wide used messenger apps. Here, the User has to register first to start a chat. Registration is authorized based on Google Fire Base Authentication so that a user can register only once and all necessity user details to start a chat are taken from Google account and verified before providing a login. After registration the user can login into the app and start the conversation. The User can set his/her desired profile picture. The User can start conversation with his/ her contacts that are already registered in the chat app. Message or a Media can be sent in a conversation to the other user. Once the other user viewed your message blue ticks will be shown to indicate message is delivered and viewed. Medias and Messages are highly end to end encrypted to ensure that messages are transferred securely. Media exchange includes images, videos, audios, documents and Contact cards. User can see when the message has been sent and can reply by writing message in Text area provided at the bottom of the chat. User can set or alter preferred settings under setting Icon provided at right corner every chat and in main chat list. User can set Online or Offline status can be viewed by the other users based on his/her interest. User can block an anonymous user if they find one. If the user is offline messages will be stored in cloud and once he/she comes online all messages will be delivered which are sent before.

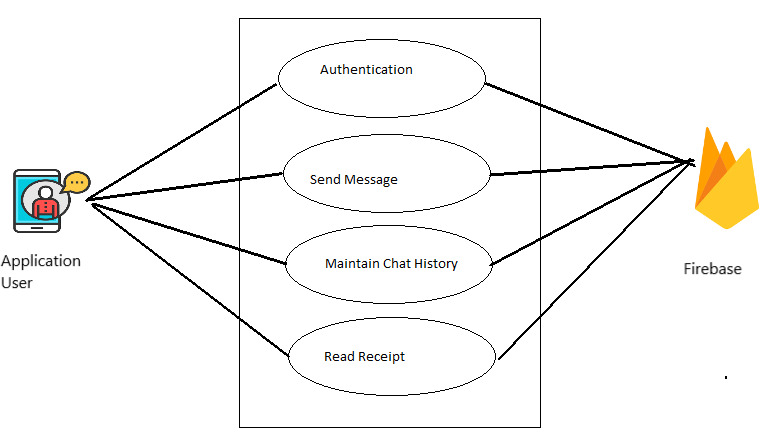
**4. SYSTEM ARCHITECTURE**



**Fig.4.1 ChatApp system architecture**

**4.1 FLOW**

As in Fig 4.1 user must enter the user login credentials or register in this app. The authentication system will authenticate the user details and password in the firebase. Once the use is logged in he/she can start a conversation with his/her friends. The user can even set up a display profile if required. The read receipt is also visible to the user.

**5. USE CASE DIAGRAM**

**Fig 5.1 Use case diagram of the application**

**5.1 FLOW**

1. User should download the Application and Register using the Google mail ID.
2. After Registration, Login into the application.
3. User can view the other users in USERS tab and start messaging.
4. Once started chatting that chat will appear in CHATS tab.
5. User can set profile picture in Profile tab.

**6. REQUIREMENTS**

**6.1 FUNCTIONAL REQUIREMENTS**

1. User Registration User must be able to register for the application through a valid phone number. On installing the application, user must be prompted to register their phone number. If user skips this step, application should close. The users phone number will be the unique identifier of his/her account on Chat Application.
2. Adding New Contacts The application should detect all contacts from the user’s phone book. If any of the contacts have user accounts with Chat Application, those contacts must automatically be added to the users contact list on Chat 8 Application. If any of the contacts have not yet registered on Chat Application, user should be provided with an invite option that sends those contacts a regular text message asking them to join Chat Application along with a link to the Chat Application on Google Playstore.
3. Send Message User should be able to send instant message to any contact on his/her Chat Application contact list. User should be notified when message is successfully delivered to the recipient by displaying a tick sign next to the message sent.
4. Broadcast Message User should be able to create groups of contacts. User should be able to broadcast messages to these groups.
5. Message Status User must be able to get information on whether the message sent has been read by the intended recipient. If recipient reads the message, 2 ticks must appear next to the message read.

**6.2 SOFTWARE REQUIREMENTS**

 Language: Java

 Tool: Android Studio

 Database: Firebase

 Operating System: Windows/Linux

**6.3 NON-FUNCTIONAL REQUIREMENTS**

1. Privacy Messages shared between users should be encrypted to maintain privacy.
2. Robustness In case users device crashes, a backup of their chat history must be stored on remote database servers to enable recoverability.
3. Performance Application must be lightweight and must send messages instantly.

**6.4INTERFACE REQUIREMENTS**

**6.4.1 GUI**

1. The starting page should be get the login or register credential.
2. The messages have to be sent correctly to the recipient.
3. The online status has to be shown if the client is connected to the network.

**6.4.2 HARDWARE INTERFACE**

1. Android Phone
2. 128 MB minimum RAM Required
3. Internet or LAN Connections
4. Processor with Speed of 500 MHz

**6.4.3 SOFTWARE INTERFACE**

Software Interface: The database should be ready to process queries.

**6.5 PERFORMANCE REQUIREMENTS**

* The App should perform message sending actions accurately.
* Double ticks have to be shown if the message is delivered.
* Blue ticks have to be shown if the message is read.

**6.6 OPERATIONAL SCENARIOS**

The Real-time-database will be connected to each user on successful connection to the internet.

**7. MODULE EXPLANATION:**

7.1AUTHENTICATION

The user will login using his/her login credentials. The user’s username and password will be authenticated in the firebase. If the user’s password is valid the app will lead the user to its home page where the user can start a conversation with his/her peers. But if the password is incorrect, an alert message will be popped as incorrect password or username. So this authentication system enables the user a strong security aspect. In case the user forgot his/her password, then the user can opt for clicking the “forgot password” button. This will send a reset password e-mail to the requested user. Using that mail the user can set a new password and login into the app and start conversation.

7.2 READ RECEIPT

In this chat app the user will be able to see the read receipt of their messages that has been sent. Once the receiver receives the message, grey color double tick will be displayed neatly near the message. If the receiver is not connected to the internet connection, a single grey tick will be displayed near the message that has been sent. If the receiver reads the message that is been send by the user, then the double grey tick will be changed to blue color which indicates that the receiver has read the message. In this way the sender will be able to know whether the message has been read by the receiver.

7.3 SENDING MESSAGE

The messages which are set by the sender will form a stack in the firebase.

When the sender sends his/her first message this stack will be created in the firebase. When the sender sends a message to a particular receiver, the sender’s id is converted to the receiver in the stack which is created. When the user sends the message the receiver will get a notification.

**8. TEST CASE**

Table No: 8.1 Test case

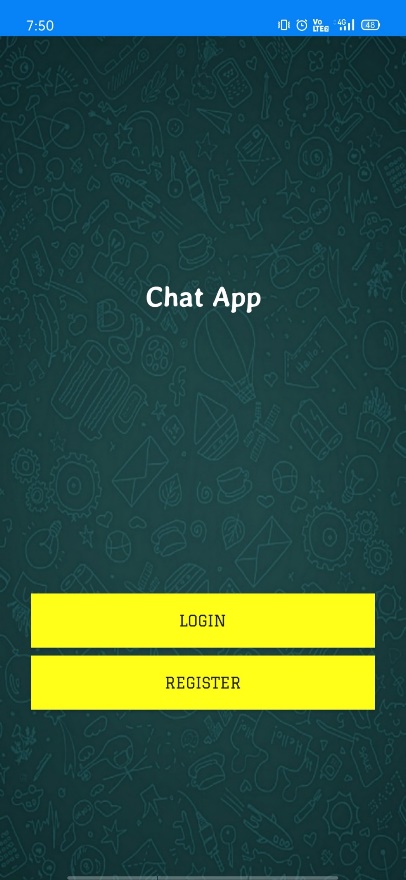
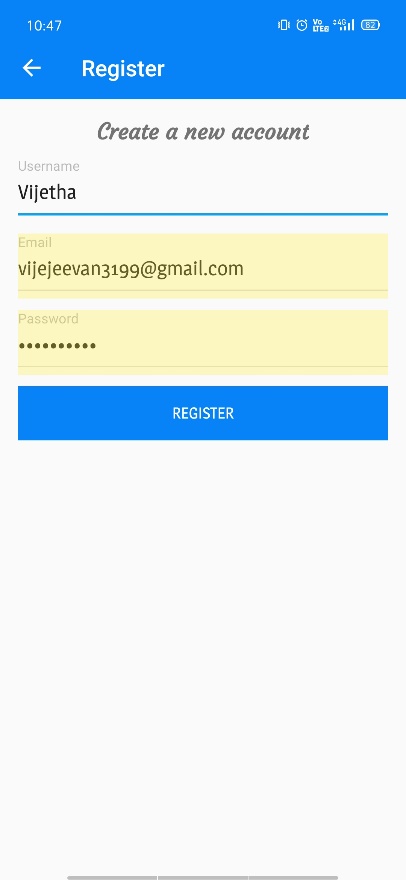
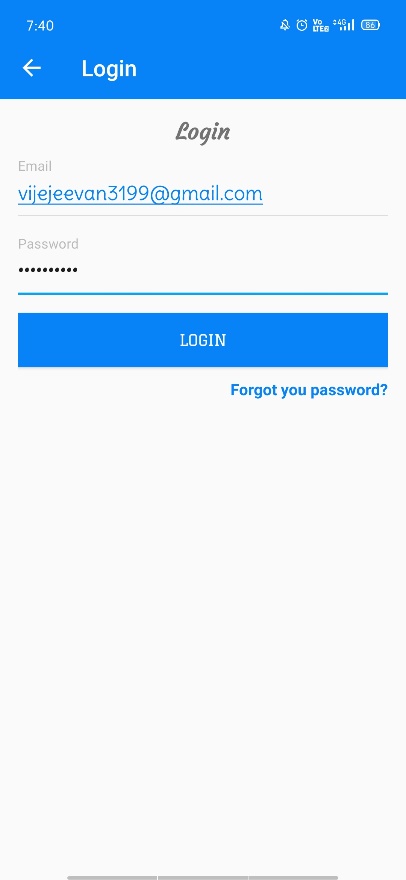
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **TEST**  **CASE**  **ID** | **Test scenario** | **Test steps** | **Test data** | **Expected Result** | **Actual Result** | **Pass**  **/Fail** | **Sign** |
| TUO1 | User registration | 1. Enter valid google mail ID.  2. Enter 6 digit password.  3. Enter valid username. | Email id = “[vijejeevan31 99@gmail.com](mailto:vijejeevan31%2099@gmail.com)”  Password = “Abc123”  Username=”Vijetha” | Registered Successfully. | Registered Successfully. | Pass |  |
| TUO2 | Login user | 1.Enter registered email ID.  2.Enter the Password. | Email id = “[vijejeevan31 99@gmail.com](mailto:vijejeevan31%2099@gmail.com)”  Password = “Abc123” | Login Successfully. | Login Successfully. | Pass |  |
| TUO3 | Updated Profile Picture | 1.Navigate to Profile Activity.  2.Click on Profile image logo.  3.select the image. | Image src = “img123.jpg” | Profile Picture Updated Successfully. | Profile Picture Updated Successfully. | Pass |  |
| TUO4 | Send Message | 1.Navigate to users Activity.  2.Tap on the user you want send message.  3.Enter the message in textarea and click send button. | Message = “Hello, This is Test Message”. | Message Sent | Message Sent | Pass |  |
| TUO5 | Receive Message | 1.Check for the Notification.  2.Click the Notification to view the Message.  3.else, In the Chats Activity the message will be received. | Message = “Hello, This is Test Message”. | Message  Received. | Message  Received. | Pass |  |
| TUO6 | Read Receipt  (Delivered) | 1.Send the Message to any User. | Message = “Hello, This is Test Message”.  (User should not open the message) | Delivered | Delivered | Pass |  |
| TUO7 | Read Receipt  (Seen) | 1.Send the Message to any User. | Message = “Hello, This is Test Message”.  (User should open the message) | Seen | Seen | Pass |  |

**9. CONCLUSION**

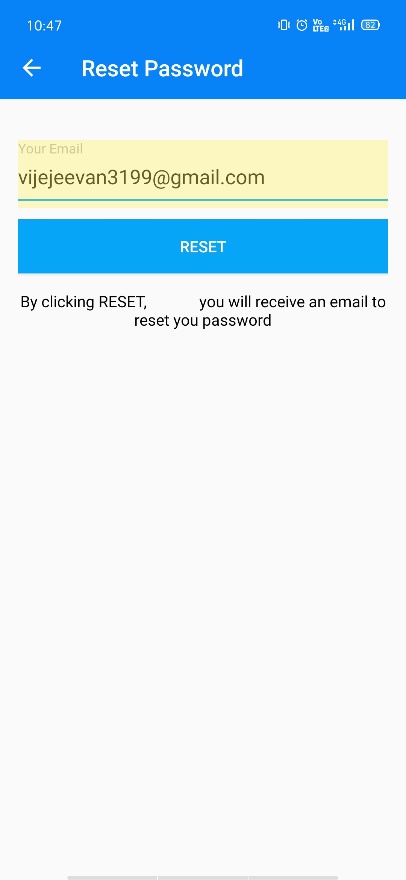
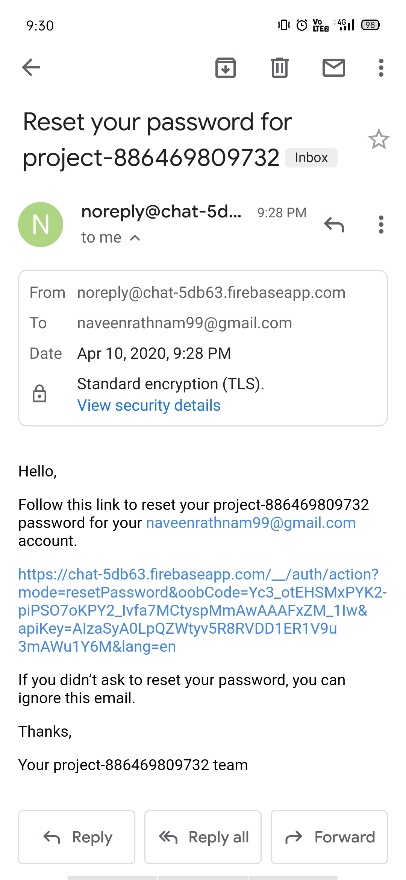
The communication brings people together, closer to each other. The communication is an important management function closely associated with all other managerial functions. It bridges the gap between individuals and groups through flow of information and understanding between them. With the emergence of internet, chatting has become a easy and comfortable mode of communication. So we have developed a app called ChatApp which is exclusive for chatting. In this app a user can login using his login credencials and chat with his/her peers. There is always a room for improvements in any apps. Right now we are just dealing with text communication. There are several android apps which serve similar purpose as this project, but these apps were rather difficult to use and provide confusing interfaces. Our app has simple functionalities and is easy to use. A positive first impression is essential in human relationship as well as in human computer interaction. This project hopes to develop a chat service Android app with high quality user interface. In future we may be extended to include features such as: File Transfer, Voice Message. Video Message, Audio Call, Video Call, Group Call.

**10. SCREEN SHOTS**

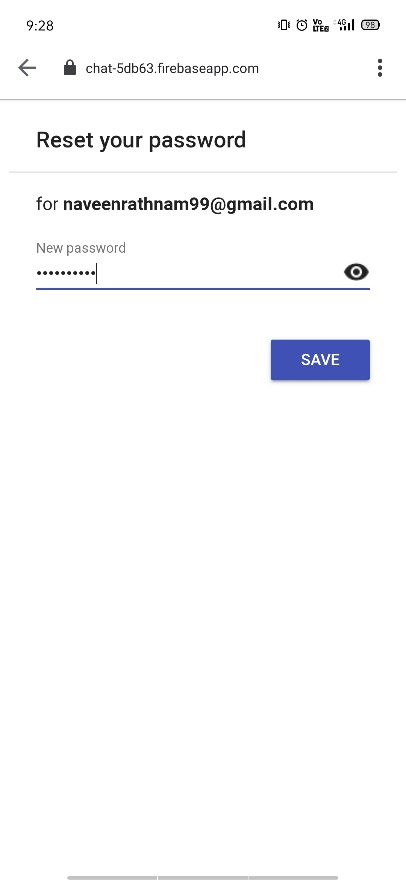
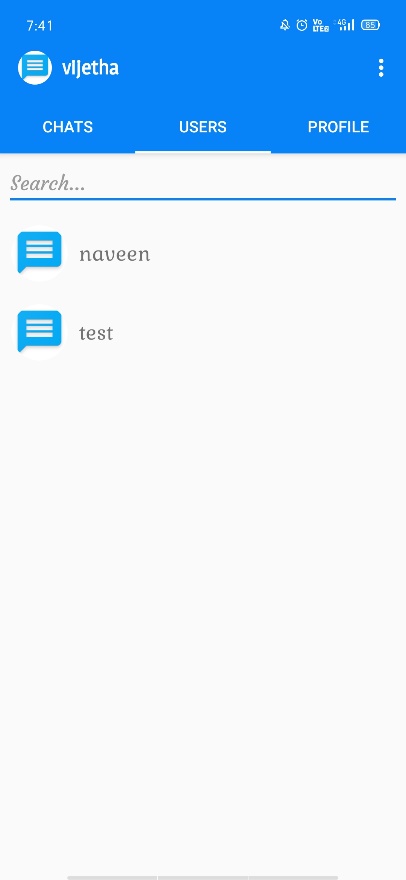
Main Activity Register Activity Login Activity

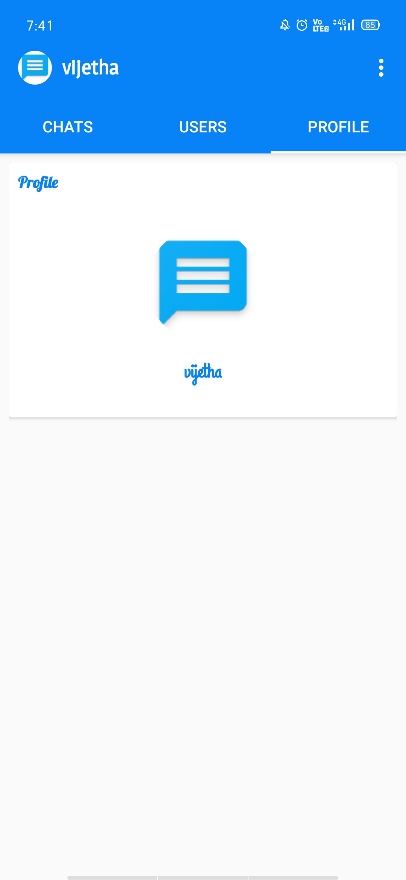
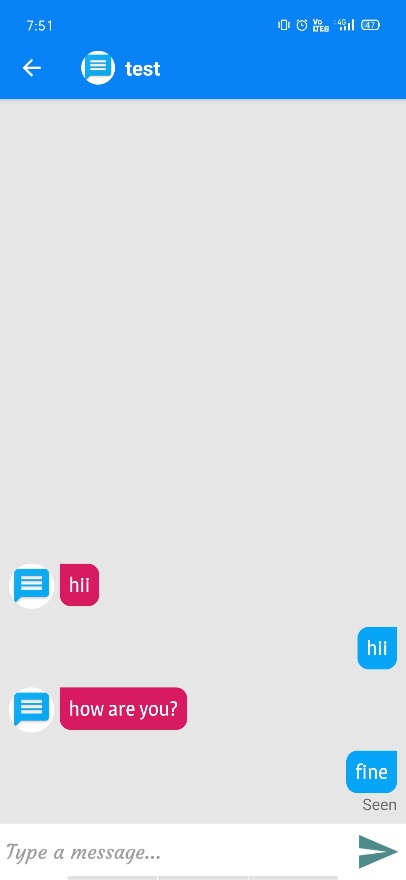
Forget Password Reset Link

Set New Password Chats Activity Users Activity

Profile Activity Chat Activity

**11. CODING**

//MainActivity.java

public class MainActivity extends AppCompatActivity {

CircleImageView profile\_image;

TextView username;

FirebaseUser firebaseUser;

DatabaseReference reference;

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Toolbar toolbar = findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

getSupportActionBar().setTitle("");

profile\_image = findViewById(R.id.profile\_image);

username = findViewById(R.id.username);

firebaseUser = FirebaseAuth.getInstance().getCurrentUser();

reference = FirebaseDatabase.getInstance().getReference("Users").child(firebaseUser.getUid());

reference.addValueEventListener(new ValueEventListener() {

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

User user = dataSnapshot.getValue(User.class);

username.setText(user.getUsername());

if (user.getImageURL().equals("default")){

profile\_image.setImageResource(R.mipmap.ic\_launcher);

} else { Glide.with(getApplicationContext()).load(user.getImageURL()).into(profile\_image);

}

}

final TabLayout tabLayout = findViewById(R.id.tab\_layout);

final ViewPager viewPager = findViewById(R.id.view\_pager);

reference = FirebaseDatabase.getInstance().getReference("Chats");

reference.addValueEventListener(new ValueEventListener() {

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

ViewPagerAdapter viewPagerAdapter = new ViewPagerAdapter(getSupportFragmentManager());

int unread = 0;

for (DataSnapshot snapshot : dataSnapshot.getChildren()){

Chat chat = snapshot.getValue(Chat.class);

if (chat.getReceiver().equals(firebaseUser.getUid()) && !chat.isIsseen()){

unread++;

}

}

if (unread == 0){

viewPagerAdapter.addFragment(new ChatsFragment(), "Chats");

} else {

viewPagerAdapter.addFragment(new ChatsFragment(), "("+unread+") Chats");

}

viewPagerAdapter.addFragment(new UsersFragment(), "Users");

viewPagerAdapter.addFragment(new ProfileFragment(), "Profile");

viewPager.setAdapter(viewPagerAdapter);

tabLayout.setupWithViewPager(viewPager);

}

}

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.menu, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()){

case R.id.logout:

FirebaseAuth.getInstance().signOut();

startActivity(new Intent(MainActivity.this, StartActivity.class).setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP));

return true;

}

return false;

}

class ViewPagerAdapter extends FragmentPagerAdapter {

private ArrayList<Fragment> fragments;

private ArrayList<String> titles;

ViewPagerAdapter(FragmentManager fm){

super(fm);

this.fragments = new ArrayList<>();

this.titles = new ArrayList<>();

}

}

private void status(String status){

reference = FirebaseDatabase.getInstance().getReference("Users").child(firebaseUser.getUid());

HashMap<String, Object> hashMap = new HashMap<>();

hashMap.put("status", status);

reference.updateChildren(hashMap);

}

}

//LoginActivity.java

public class LoginActivity extends AppCompatActivity {

MaterialEditText email, password;

Button btn\_login;

FirebaseAuth auth;

TextView forgot\_password;

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

Toolbar toolbar = findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

getSupportActionBar().setTitle("Login");

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

auth = FirebaseAuth.getInstance();

email = findViewById(R.id.email);

password = findViewById(R.id.password);

btn\_login = findViewById(R.id.btn\_login);

forgot\_password = findViewById(R.id.forgot\_password);

forgot\_password.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

startActivity(new Intent(LoginActivity.this, ResetPasswordActivity.class));

}

});

btn\_login.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String txt\_email = email.getText().toString();

String txt\_password = password.getText().toString();

if (TextUtils.isEmpty(txt\_email) || TextUtils.isEmpty(txt\_password)){

Toast.makeText(LoginActivity.this, "All fileds are required", Toast.LENGTH\_SHORT).show();

} else {

auth.signInWithEmailAndPassword(txt\_email, txt\_password)

.addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()){

Intent intent = new Intent(LoginActivity.this, MainActivity.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

finish();

} else {

Toast.makeText(LoginActivity.this, "Authentication failed!", Toast.LENGTH\_SHORT).show();

} } }); } } });

}

}

//MessageActivity.java

public class MessageActivity extends AppCompatActivity {

CircleImageView profile\_image;

TextView username;

FirebaseUser fuser;

DatabaseReference reference;

ImageButton btn\_send;

EditText text\_send;

MessageAdapter messageAdapter;

List<Chat> mchat;

RecyclerView recyclerView;

Intent intent;

ValueEventListener seenListener;

String userid;

APIService apiService;

boolean notify = false;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_message);

Toolbar toolbar = findViewById(R.id.toolbar);

setSupportActionBar(toolbar);

getSupportActionBar().setTitle("");

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

toolbar.setNavigationOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

// and this

startActivity(new Intent(MessageActivity.this, MainActivity.class).setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP));

}

});

apiService = Client.getClient("https://fcm.googleapis.com/").create(APIService.class);

recyclerView = findViewById(R.id.recycler\_view);

recyclerView.setHasFixedSize(true);

LinearLayoutManager linearLayoutManager = new LinearLayoutManager(getApplicationContext());

linearLayoutManager.setStackFromEnd(true);

recyclerView.setLayoutManager(linearLayoutManager);

profile\_image = findViewById(R.id.profile\_image);

username = findViewById(R.id.username);

btn\_send = findViewById(R.id.btn\_send);

text\_send = findViewById(R.id.text\_send);

intent = getIntent();

userid = intent.getStringExtra("userid");

fuser = FirebaseAuth.getInstance().getCurrentUser();

btn\_send.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

notify = true;

String msg = text\_send.getText().toString();

if (!msg.equals("")){

sendMessage(fuser.getUid(), userid, msg);

} else {

Toast.makeText(MessageActivity.this, "You can't send empty message", Toast.LENGTH\_SHORT).show(); }

text\_send.setText(""); }

});

reference = FirebaseDatabase.getInstance().getReference("Users").child(userid);

reference.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

User user = dataSnapshot.getValue(User.class);

username.setText(user.getUsername());

if (user.getImageURL().equals("default")){

profile\_image.setImageResource(R.mipmap.ic\_launcher);

} else { Glide.with(getApplicationContext()).load(user.getImageURL()).into(profile\_image); }

readMesagges(fuser.getUid(), userid, user.getImageURL());

}

seenMessage(userid);

}

private void seenMessage(final String userid){

reference = FirebaseDatabase.getInstance().getReference("Chats");

seenListener = reference.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

for (DataSnapshot snapshot : dataSnapshot.getChildren()){

Chat chat = snapshot.getValue(Chat.class);

if (chat.getReceiver().equals(fuser.getUid()) && chat.getSender().equals(userid)){

HashMap<String, Object> hashMap = new HashMap<>();

hashMap.put("isseen", true);

snapshot.getRef().updateChildren(hashMap);

} } } }

private void sendMessage(String sender, final String receiver, String message){

DatabaseReference reference = FirebaseDatabase.getInstance().getReference();

HashMap<String, Object> hashMap = new HashMap<>();

hashMap.put("sender", sender);

hashMap.put("receiver", receiver);

hashMap.put("message", message);

hashMap.put("isseen", false);

reference.child("Chats").push().setValue(hashMap);

final DatabaseReference chatRef = FirebaseDatabase.getInstance().getReference("Chatlist")

.child(fuser.getUid())

.child(userid);

chatRef.addListenerForSingleValueEvent(new ValueEventListener() {

}

final DatabaseReference chatRefReceiver = FirebaseDatabase.getInstance().getReference("Chatlist")

.child(userid)

.child(fuser.getUid());

chatRefReceiver.child("id").setValue(fuser.getUid());

final String msg = message;

reference = FirebaseDatabase.getInstance().getReference("Users").child(fuser.getUid());

reference.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

User user = dataSnapshot.getValue(User.class);

if (notify) {

sendNotifiaction(receiver, user.getUsername(), msg);

}

notify = false;

}

}

private void readMesagges(final String myid, final String userid, final String imageurl){

mchat = new ArrayList<>();

reference = FirebaseDatabase.getInstance().getReference("Chats");

reference.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot dataSnapshot) {

mchat.clear();

for (DataSnapshot snapshot : dataSnapshot.getChildren()){

Chat chat = snapshot.getValue(Chat.class);

if (chat.getReceiver().equals(myid) && chat.getSender().equals(userid) ||

chat.getReceiver().equals(userid) && chat.getSender().equals(myid)){

mchat.add(chat);

}

messageAdapter = new MessageAdapter(MessageActivity.this, mchat, imageurl);

recyclerView.setAdapter(messageAdapter);

}

}

}

}

XML CODE :

//activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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:orientation="vertical"

tools:context=".MainActivity">

<android.support.design.widget.AppBarLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<android.support.v7.widget.Toolbar

android:id="@+id/toolbar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/colorPrimaryDark"

android:theme="@style/Base.ThemeOverlay.AppCompat.Dark.ActionBar"

app:popupTheme="@style/MenuStyle">

<de.hdodenhof.circleimageview.CircleImageView

android:layout\_width="30dp"

android:layout\_height="30dp"

android:id="@+id/profile\_image"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/username"

android:textSize="18sp"

android:fontFamily="@font/magra"

android:layout\_marginLeft="25dp"

android:textColor="#fff"

android:textStyle="bold"

android:layout\_marginStart="25dp" />

</android.support.v7.widget.Toolbar>

<android.support.design.widget.TabLayout

android:id="@+id/tab\_layout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/colorPrimaryDark"

app:tabSelectedTextColor="#fff"

app:tabIndicatorColor="#fff"

app:tabTextColor="#fff"/>

</android.support.design.widget.AppBarLayout>

<android.support.v4.view.ViewPager

android:id="@+id/view\_pager"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"/>

</LinearLayout>

//activity\_message.xml

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout 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="#e6e6e6"

tools:context=".MessageActivity">

<android.support.design.widget.AppBarLayout

android:id="@+id/bar\_layout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content">

<android.support.v7.widget.Toolbar

android:id="@+id/toolbar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/colorPrimaryDark"

android:theme="@style/Base.ThemeOverlay.AppCompat.Dark.ActionBar"

app:popupTheme="@style/MenuStyle">

<de.hdodenhof.circleimageview.CircleImageView

android:layout\_width="30dp"

android:layout\_height="30dp"

android:id="@+id/profile\_image"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/username"

android:textSize="18sp"

android:layout\_marginLeft="25dp"

android:textColor="#fff"

android:textStyle="bold"

android:layout\_marginStart="25dp" />

</android.support.v7.widget.Toolbar>

</android.support.design.widget.AppBarLayout>

<android.support.v7.widget.RecyclerView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:id="@+id/recycler\_view"

android:layout\_below="@id/bar\_layout"

android:layout\_above="@id/bottom"/>

<RelativeLayout

android:layout\_width="match\_parent"

android:padding="5dp"

android:id="@+id/bottom"

android:background="#fff"

android:layout\_alignParentBottom="true"

android:layout\_height="wrap\_content">

<EditText

android:id="@+id/text\_send"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_centerVertical="true"

android:layout\_toLeftOf="@id/btn\_send"

android:background="@android:color/transparent"

android:fontFamily="@font/courgette"

android:hint="Type a message..." />

<ImageButton

android:layout\_width="40dp"

android:layout\_height="40dp"

android:background="@drawable/ic\_action\_name"

android:id="@+id/btn\_send"

android:layout\_alignParentEnd="true"

android:layout\_alignParentRight="true" />

</RelativeLayout>

</RelativeLayout>