EASYCHAT (EC)

A
Project Report

Submitted in partial fulfilment of the
Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

DIVYA VARSHINI MOTURI(1602-17-737-073)

KRANTHI KUMAR(1602-17-737-078)

VARUN(1602-17-737-113)

Under the guidance of

MR. KRISHNA KISHORE

Assistant Professor

Department of Information Technology
Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2019-20

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology

DECLARATION BY THE CANDIDATE

We, **Divya Varshini Moturi, Kranthi Kumar, Varun,** bearing hall ticket numbers,**1602-17-737-073, 1602-17-737-078, 1602-17-737-113,** hereby declare that the project report entitled **EasyChat** under the guidance of **Mr. Krishna Kishore,** Assistant Professor, Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfilment of the requirement of MINI PROJECT of VI Semester of **Bachelor of Engineering** in **Information Technology.**

This is a record of bonafide work carried out by me and the results embodied in this project.

Divya Varshini Moturi 1602-17-737-073

Kranthi Kumar 1602-17-737-078

Varun 1602-17-737-113

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology

BONAFIDE CERTIFICATE

This is to certify that the project entitled **EasyChat** being submitted by **Moturi Divya Varshini, U. Kranthi Kumar, Varun** bearing **1602-17-737-073, 1602-17-737-078, 1602-17-737-113** in partial fulfilment of the requirements for the completion of MINI PROJECT of Bachelor of Engineering, VI Semester, in Information Technology is a record of bonafide work carried out by him/her under my guidance.

Mr. Krishna Kishore Rao
Assistant Professor

Dr. K. Ram Mohan
HOD , IT

Internal Mentor : Mandala Ravali (1602-16-737-032)

ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of the project would not have been possible without the kind support and help of many individuals. We would like to extend my sincere thanks to all of them. We would like to take the opportunity to express our humble gratitude to **Mr. Krishna Kishore Sir** under whom we executed this project. We would also use this opportunity to thank our senior **Mandala Ravali (1602-16-737-032)**. We are grateful to his guidance, and constructive suggestions that helped us in the preparation of this project. His constant guidance and willingness to share his vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks. We would like to thank all faculty members and staff of the **Department of Information Technology** for their generous help in various ways for the completion of this project.

Finally, yet importantly, We would like to express our heartfelt thanks to our **HOD Dr. K. Ram Mohan Rao Sir** and classmates for their help and wishes for the successful completion of this project.

ABSTRACT

The purpose of the project is to make the communication easy between individuals. People from rural areas who have difficulty in understanding English or even Hindi and users who keep travelling most of the time, users who communicate with people from different regions on a daily basis face difficulty when communicating in the languages they are not confident in. EasyChat or EC is an application which provides a way to make their communication efficient. The users can select a language and use it as a normal chatting app without any difficulties. The main goals of the project are:

- Make people from rural areas and foreigners comfortable when communicating.
- Reduce the wastage of time trying to use an interpreter.
- Omitting the struggle of learning a new language for a few messages.

TABLE OF CONTENTS

INTRODUCTION

1. Application Description

Mobile phone users in India crossed 581 million users in 2014 and have been on a steady rise over the last decade. According to a survey by eMarketer in 2015, India is estimated to have over 800 million mobile phone users in 2019.

In a country like India with 23 official languages and 150 recognized languages, communication is definitely a problem. EasyChat is an Android Application that helps people who want to communicate with people of other regions without any language barriers. They can communicate without any translator or interpreter in between. Thus, by this it increases in the retention of privacy between the respective individuals.

"Language is no longer a barrier."

2. Scope

This application is designed for making the communication easy while keeping rural people and foreigners into consideration. This application can help them to communicate even when a language is not known without hindering their privacy.

RELATED WORK

Some of the chat applications with inbuilt translators are Viber, Skype and TransFire. Here, we are going to consider chat applications namely Viber and Skype and compare it with our application EasyChat.

Viber has a new translation feature where the messages are automatically translated into the language the user selects to. The received messages of the user must be long pressed for several options to be displayed and in those several options, translate is to be selected for the message to be translated. Here, both the translated and untranslated messages are displayed.

In Viber, the people who cannot use smart phones easily face problems when they try to translate the messages as the translate icon is not readily available to the user's eye.

EasyChat also provides the same facilities as Viber but, the translate option here, appears beside every message received and by clicking on that option, the message is translated into the desired language of the user.

This feature in EasyChat helps the users in efficient chatting without any confusion and also, it reduces the wastage of time of the user.

Skype on the other side is both a web and an android application. When a user uses Skype to communicate with others, the message typed in the user's language is directly translated and sent to the receiver. That is, the receiver most of the time doesn't know that the messages sent can be of different languages and are being translated before even they are being received by them.

By this, people wanting to learn new languages by communicating with others may not be able to achieve their goals.

This situation is handled in EasyChat. As mentioned above, in EasyChat, the user receives the messages in untranslated fashion. The messages that the receiver wants to translate can be translated with an EC icon without any complications.

The areas the we can improve in EasyChat are

- Allowing users to request and accept friend requests.
- Making the chat even more easy by introducing speech to text plus translation in the application.

PROPOSED WORK

Motivation of the Proposal

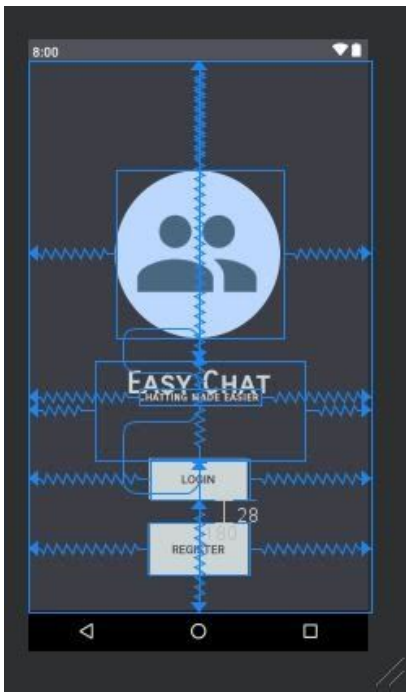
EasyChat helps in translating the messages the rural people receive from the authorities without any third persona in between. This helps in retaining their privacy. It can also help the travellers who can't speak in the regional languages. They can omit learning a new language to travel. This can make the communication between the population easy and efficient while retaining their privacy.

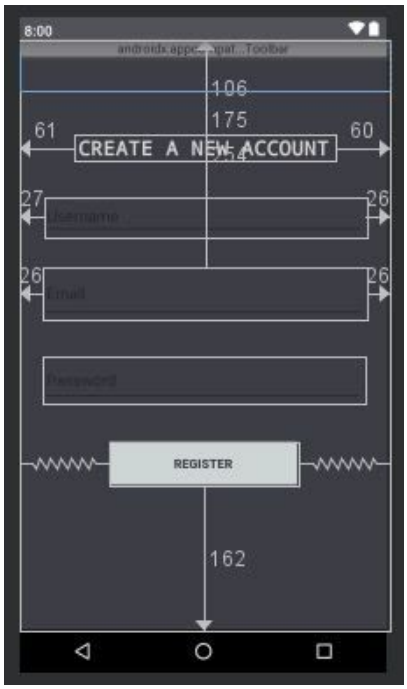
1. Use Cases

Objective : Developing a concrete platform for the people to use for translation as well as chat without any difficulties.

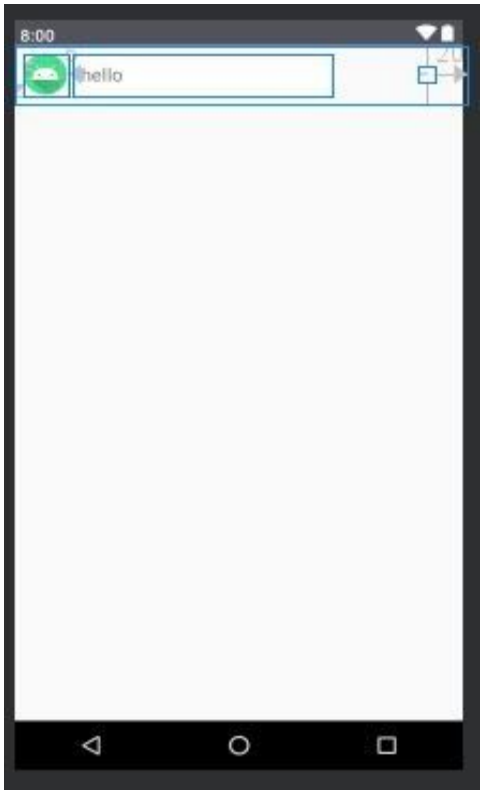
- Translation of the messages received to the user's language
- Sending the messages from the user
- Saving the recent language translations
- Language Flexibility : Allowing the user to select language of the message to be sent.

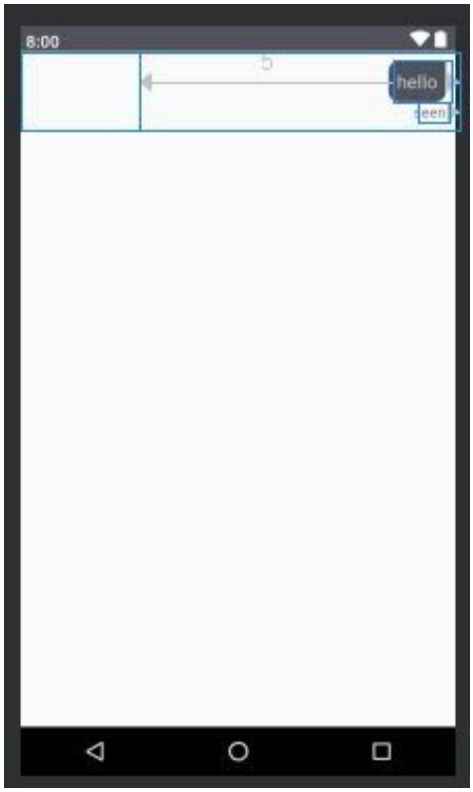
2. UI Prototypes

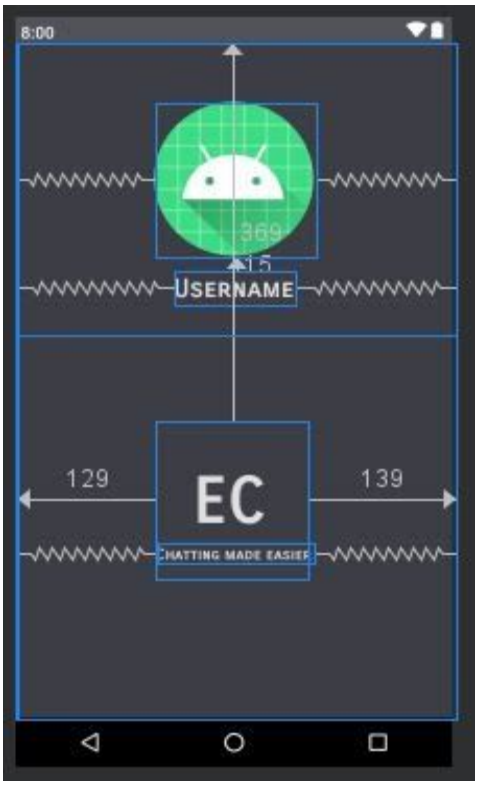





8:00
Username
Item 0
Item 1
Item 2
Item 3
Item 4
Item 5
Item 6
Item 7
Item 8
Item 9
Type a Message


8:00
hello


8:00
hello

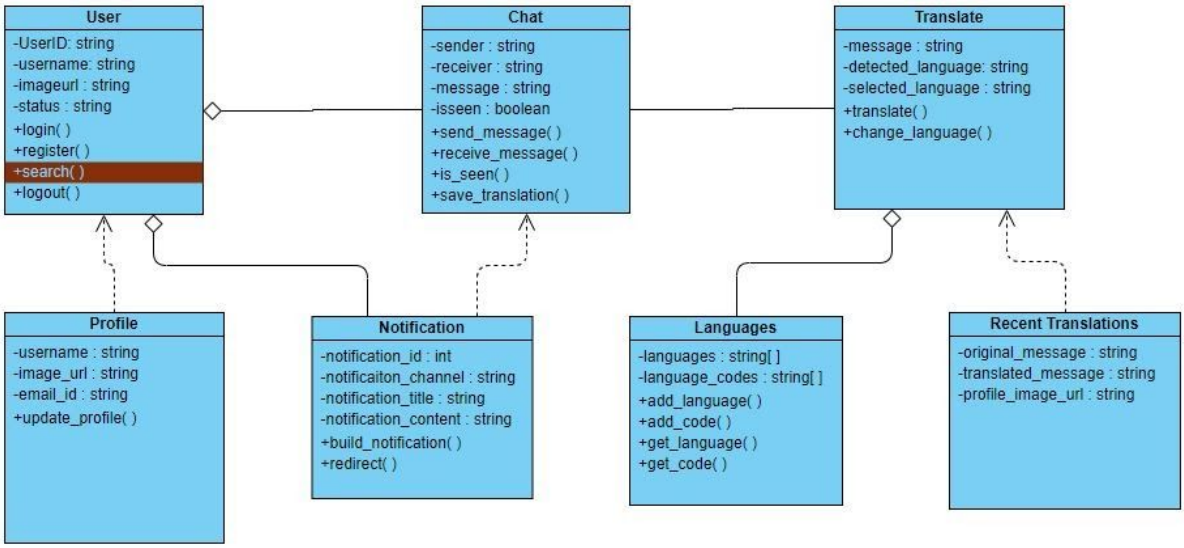

8:00
USERNAME
129
EC
139
CHATTING MADE EASIER

3. Architecture and Technology Used

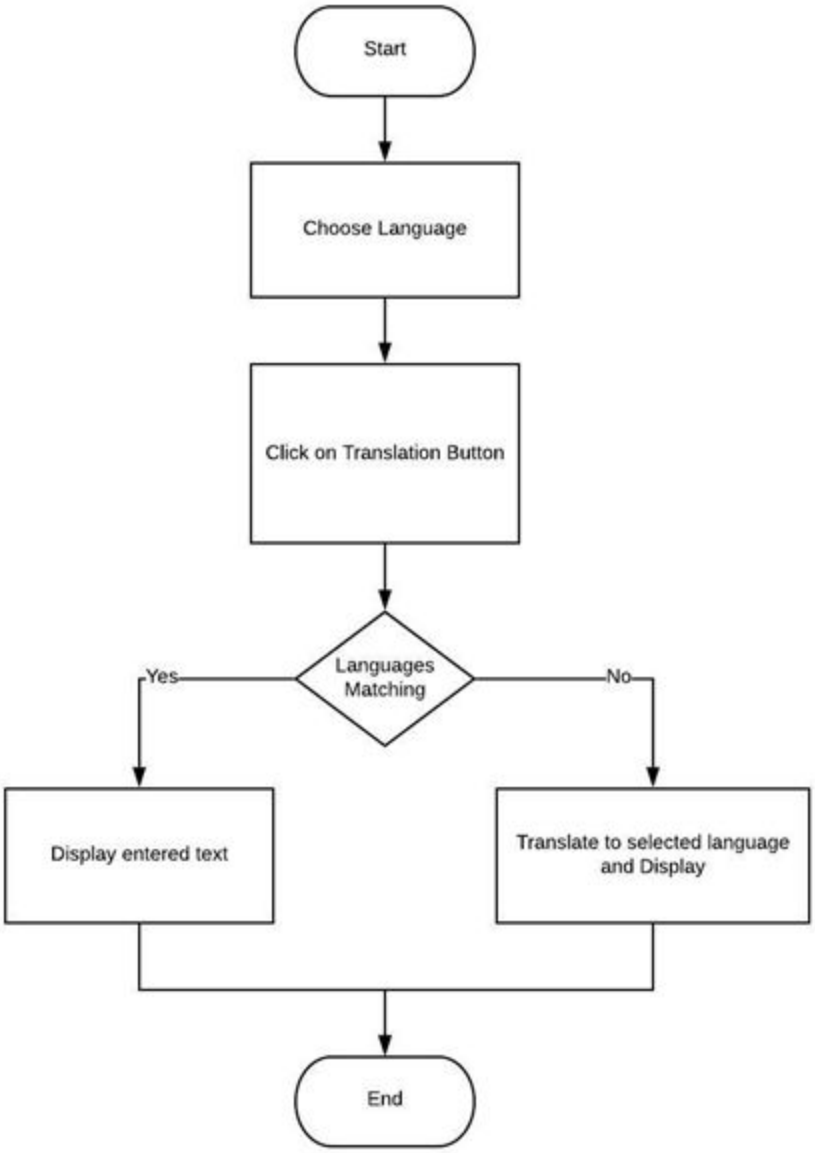
- Android Studio
- ADB Emulator
- Google Translation API
- ML Toolkits
- Firebase (Real Time Database & Authentication)
- Languages used : Java and Kotlin

4. Design

4.1. UML Static diagram (Class Diagram) :



4.2. UML Run time diagram (Flow Chart) :



5. Implementation

5.1. Description of Main Classes :

User Class :

```
public class User {
  private String id;
  private String username;
  private String imageURL;
  private  String status;
  private String search;
  public User() {
  }

  public User(String id, String username, String URL,String status,String search){
      this.id=id;
      this.username=username;
      this.status=status;
      this.search=search;
      imageURL=URL;

  }

  public String getId() {
      return id;
  }

  public void setId(String id) {
      this.id = id;
  }

  public String getUsername() {
      return username;
  }

  public void setUsername(String username) {
      this.username = username;
  }

  public String getImageURL() {
      return imageURL;
  }

  public void setImageURL(String imageURL) {
      this.imageURL = imageURL;
  }
```

```
    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

    public String getSearch() {
        return search;
    }

    public void setSearch(String search) {
        this.search = search;
    }
}
```

User class contains user ID, username, image URL, status, search attributes. The image URL attribute holds the url of the profile image of the user. Status attribute holds the status of the user which can be either ‘online’ or ‘offline’. Search attribute holds the search value of the user for making the search function to work efficiently.All the attributes are accessed and modified using getter and setter functions.

Chat class :

```
public class Chat {

    private String sender;
    private String receiver;
    private String msg;
    private boolean isseen;

    public Chat() {
    }

    public Chat(String sender, String receiver, String msg,boolean isseen) {
        this.sender = sender;
        this.receiver = receiver;
        this.msg = msg;
        this.isseen=isseen;
    }

    public String getSender() {
```

```
        return sender;
    }

    public void setSender(String sender) {
        this.sender = sender;
    }

    public String getReceiver() {
        return receiver;
    }

    public void setReceiver(String receiver) {
        this.receiver = receiver;
    }

    public String getMsg() {
        return msg;
    }

    public void setMsg(String msg) {
        this.msg = msg;
    }


    public boolean isIsseen() {
        return isseen;
    }

    public void setIsseen(boolean isseen) {
        this.isseen = isseen;
    }
}
```

The chat class contains the attributes sender, receiver, msg and isseen. The sender attribute holds the ID of the sender of the message. The receiver attribute holds the ID of the receiver of the message. Msg attribute holds the message in String format. The boolean isseen attribute can take two values, either true or false indicating whether the message has been seen or not. All the attributes are accessed and modified using getter and setter functions.

Languages class :

```
package com.example.easychatec.Model;

public class Languages {

    private String[] languageCollection;
    private String[] languageCode;

    public Languages() {
        String languages[] =
{"Telugu","Hindi","English","Tamil","Marathi","Bengali","Kannada","Malayalam","Gujarati",
"Korean","Chinese","Japanese","French","Spanish","German","Italian","Latin","Arabic","Tur
kish","Vietnamese","Zulu"};
        String codes[] =
{"te","hi","en","ta","mr","bn","kn","ml","gu","ko","zh-TW","ja","fr","es","de","it","la","ar
","tr","vi","zu"};
        languageCollection=languages;
        languageCode=codes;
    }

    public String[] getLanguageCollection() {
        return languageCollection;
    }

    public String[] getLanguageCode() {
        return languageCode;
    }

    public void add(String language, String code){
        String[] array =  new String[languageCollection.length+1];
        for(int i=0;i<languageCollection.length;i++){
            array[i]=languageCollection[i];
        }
        array[languageCollection.length]=language;
        String[] array1 =  new String[languageCode.length+1];
        for(int i=0;i<languageCode.length;i++){
            array[i]=languageCode[i];
        }
        array[languageCode.length]=code;

    }


}
```

Languages class contains a language collection which holds the list of all the languages that are presently available in the app and codes collection which holds the list of the codes of all the corresponding languages. Languages can be added to the collection using add function.

User Adapter :

```
public class UserAdapter extends RecyclerView.Adapter<UserAdapter.viewHolder> {
    private Context mContext;
    private List<User> mUsers;
    private boolean isOn;
    private String Tid;
    private String ID;
    public UserAdapter() {
    }

    public UserAdapter(Context mContext, List<User> mUsers,boolean isOn) {
        this.mContext = mContext;
        this.mUsers = mUsers;
        this.isOn=isOn;
    }

    @NonNull
    @Override
    public viewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view= LayoutInflater.from(mContext).inflate(R.layout.users_item,parent,false);

        return new viewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull final viewHolder holder, int position) {

        final User user=mUsers.get(position);
        Tid=user.getId();
        if(Main3Activity.RED.contains(Tid)){
            holder.ms.setVisibility(View.VISIBLE);
        }
        holder.username.setText(user.getUsername());
        if(user.getImageURL().equals("default")){
            holder.profile_image.setImageResource(R.mipmap.ic_launcher);
        }else{
            Glide.with(mContext).load(user.getImageURL()).into(holder.profile_image);
        }
```

```
    if(isOn){
        if(user.getStatus().equals("online")){
            holder.img_on.setVisibility(View.VISIBLE);
            holder.img_off.setVisibility(View.GONE);
        }else{
            holder.img_off.setVisibility(View.VISIBLE);
            holder.img_on.setVisibility(View.GONE);
        }
    }else{
        holder.img_off.setVisibility(View.GONE);
        holder.img_on.setVisibility(View.GONE);
    }
    DatabaseReference ref = FirebaseDatabase.getInstance().getReference("chats");


    holder.itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {

            Main3Activity.RED.clear();
            Intent intent = new Intent(mContext, MessageActivity.class);
            intent.putExtra("userid", user.getId());
            mContext.startActivity(intent);
        }
    });

}

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

public static class viewHolder extends RecyclerView.ViewHolder{
    public TextView username;
    public ImageView profile_image;
    public ImageView img_on;
    public ImageView img_off;
    public TextView ms;
    public viewHolder(View itemView){
        super(itemView);
        ms=itemView.findViewById(R.id.ms);
        username=itemView.findViewById(R.id.username);
        profile_image=itemView.findViewById(R.id.profile_image);
        img_on=itemView.findViewById(R.id.img_on);
        img_off=itemView.findViewById(R.id.img_off);
```

```
    }
}
}
```

User Adapter is the item which is added to the recycler view in User fragment. It holds the profile image and name of the user. Several listeners are attached to the adapter for listening to the data changes so that whenever there is a change in the username or profiles the adapter is updated. Also a onClickListener is added to direct the user to messaging activity so that a messaging environment is created between the intended users. It also contains a small circular image element to show whether the user is online or offline. This circular image turns green when the corresponding user is online. Each user is listed with a user adapter in the users fragment.

MessageAdapter :

```
public class MessageAdapter extends RecyclerView.Adapter<MessageAdapter.viewHolder>{

    public static Translations translations = new Translations();
    public static final int MSG_TYPE_LEFT=0;
    public static final int MSG_TYPE_RIGHT=1;
    private Context mContext;
    private List<Chat> mChats;
    private String imageURL;
    private int viewtype;
    FirebaseUser fuser;

    public String translatedText;
    public Translate translate;
    private boolean connected;



    public MessageAdapter() {
    }

    public MessageAdapter(Context mContext, List<Chat> mChats,String imageURL) {
        this.mContext = mContext;
        this.mChats = mChats;
        this.imageURL=imageURL;
    }
```

```
    @NonNull
    @Override
    public MessageAdapter.viewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        if(viewType==MSG_TYPE_RIGHT) {
            View view = LayoutInflater.from(mContext).inflate(R.layout.chat_item_right, parent, false);
            return new viewHolder(view);
        }else{
            View view = LayoutInflater.from(mContext).inflate(R.layout.chat_item_left, parent, false);
            return new viewHolder(view);
        }

    }

    @Override
    public void onBindViewHolder(@NonNull final MessageAdapter.viewHolder holder, int position) {

        Chat chat = mChats.get(position);
        String message = chat.getMsg();
        holder.show_message.setText(message);
        final String originalText=message;
        if(imageURL.equals("default")){
            holder.profile_image.setImageResource(R.mipmap.ic_launcher);
        }else{
            Glide.with(mContext).load(imageURL).into(holder.profile_image);
        }

        if(viewtype==MSG_TYPE_RIGHT){
            if(chat.isIsseen()){
                holder.msg_seen.setText("Seen");
            }else{
                holder.msg_seen.setText("Delivered");
            }
        }else{
            holder.msg_seen.setVisibility(View.GONE);
        }
        if(viewtype==MSG_TYPE_LEFT){
            holder.trans_btn.setOnClickListener(new View.OnClickListener() {

                public void onClick(View v) {
                    if(checkInternetConnection()) {

                        //If there is internet connection, get translate service and start translation:
                        getTranslateService();
                        translate(holder,originalText);


                    } else {
```

```
                    //If not, display "no connection" warning:
                    holder.trans_message.setVisibility(View.VISIBLE);

holder.trans_message.setText(mContext.getResources().getString(R.string.no_connection));
                }

            }
        });




    }


    public boolean checkInternetConnection() {

        //Check internet connection:

        ConnectivityManager connectivityManager = (ConnectivityManager)
mContext.getSystemService(Context.CONNECTIVITY_SERVICE);

        //Means that we are connected to a network (mobile or wi-fi)
        connected =
connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_MOBILE).getState() ==
NetworkInfo.State.CONNECTED ||
                connectivityManager.getNetworkInfo(ConnectivityManager.TYPE_WIFI).getState() ==
NetworkInfo.State.CONNECTED;

        return connected;
    }

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

    public static class viewHolder extends RecyclerView.ViewHolder{
        public TextView show_message;
        public ImageView profile_image;
        public ImageButton trans_btn;
        public View rootView;
        public TextView trans_message;
        public TextView msg_seen;

        public viewHolder(View itemView){
```

```
        super(itemView);
        rootView = itemView;
        show_message=itemView.findViewById(R.id.show_message);
        trans_message=itemView.findViewById(R.id.translated_message);
        profile_image=itemView.findViewById(R.id.profile_image);
        trans_btn=itemView.findViewById(R.id.trans_btn);
        msg_seen= itemView.findViewById(R.id.msg_seen);

    }
}

@Override
public int getItemViewType(int position) {
    fuser = FirebaseAuth.getInstance().getCurrentUser();
    if(mChats.get(position).getSender().equals(fuser.getUid())){
        viewtype= MSG_TYPE_RIGHT;
        return MSG_TYPE_RIGHT;
    }else{
        viewtype=MSG_TYPE_LEFT;
        return MSG_TYPE_LEFT;
    }
}
}
```

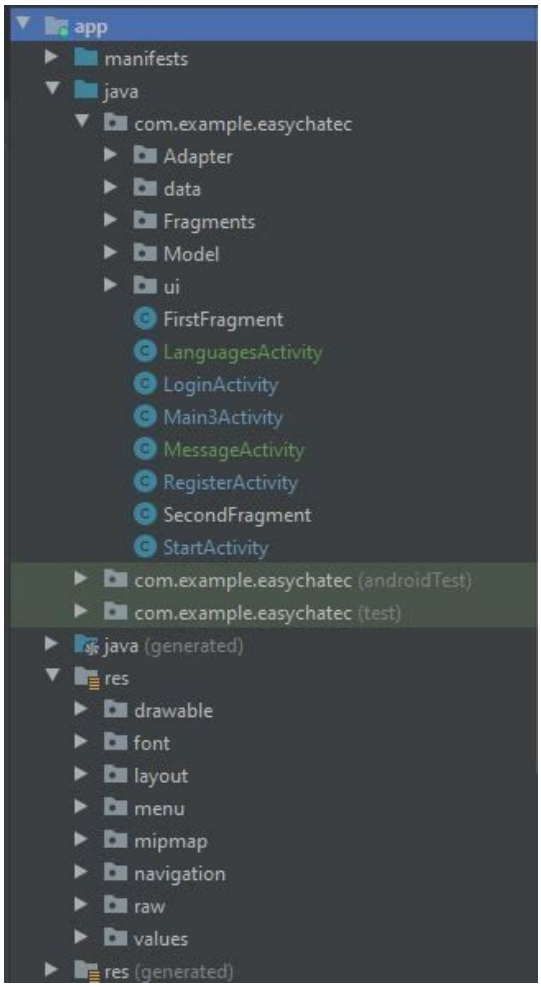
The message adapter can be of two layouts which explains two message types, left is for received message and right is for sent messages. For received messages an EC button is inflated for availing translation service. On pressing the EC button a new message element with translate message is concatenated below the original message element. List of messages are displayed with an appropriate number of message adapters.

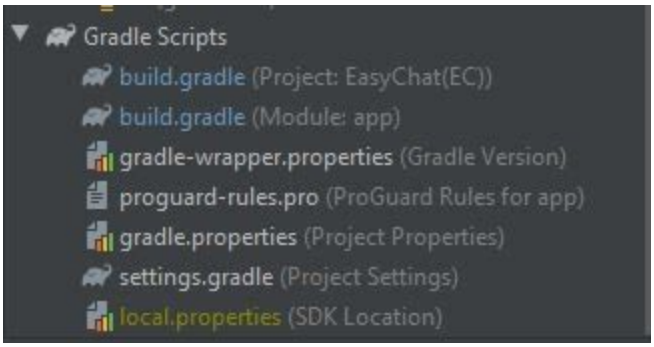
5.2. Github links and folder structure :

Github Link:

https://github.com/Kranthi-Kumar-Naidu/EasyChatApp

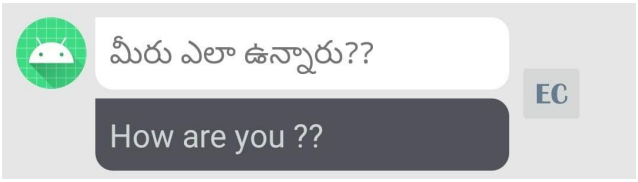
Folder Structure :

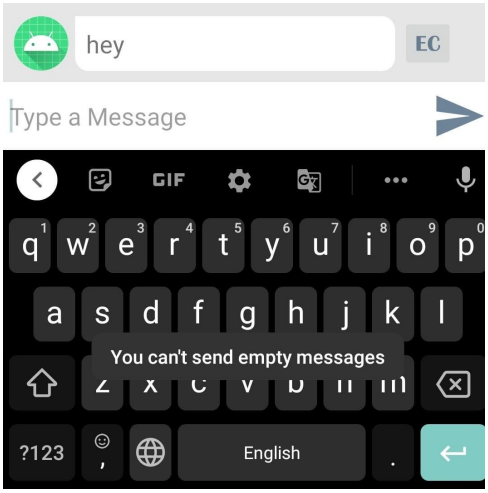




6. Testing

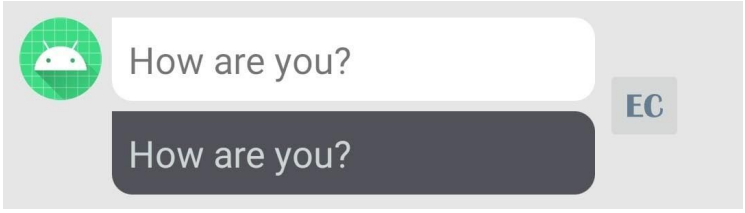
6.1. Translating Received message into selected language





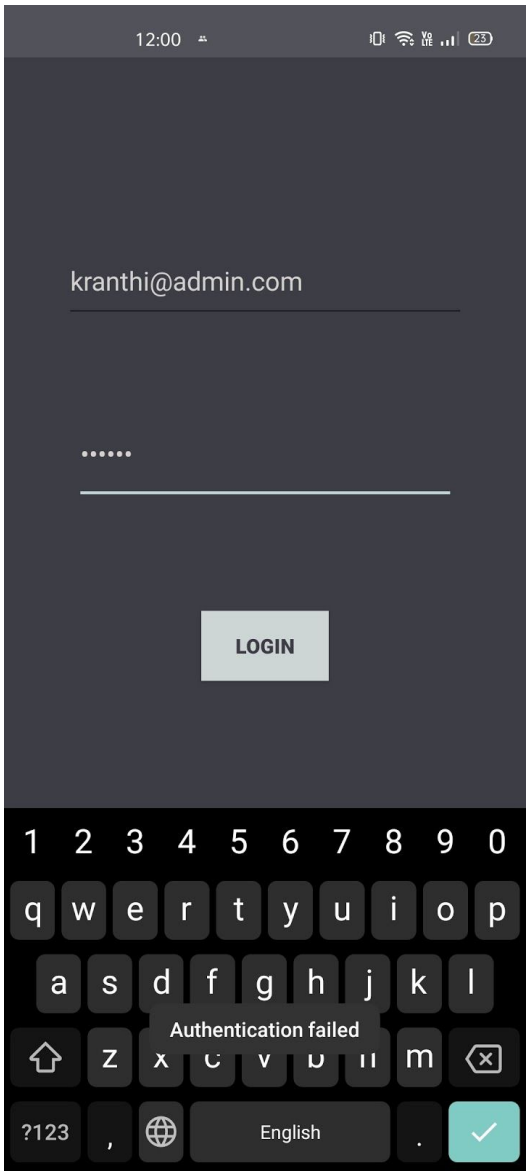
If the received message is in a language other than selected language, the language in which the message is received is automatically detected and is translated into selected language once the EC button is pressed. When the user tries to send an empty page a message is displayed on the screen.

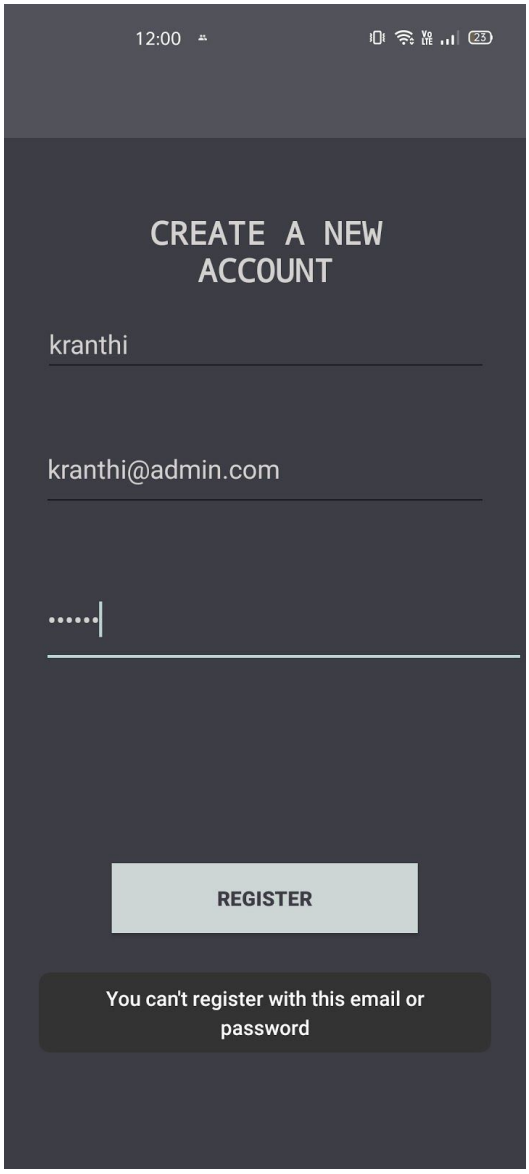
6.2. Displaying same text if message is received in same language



If the received message is in the selected language, then the same message is displayed.

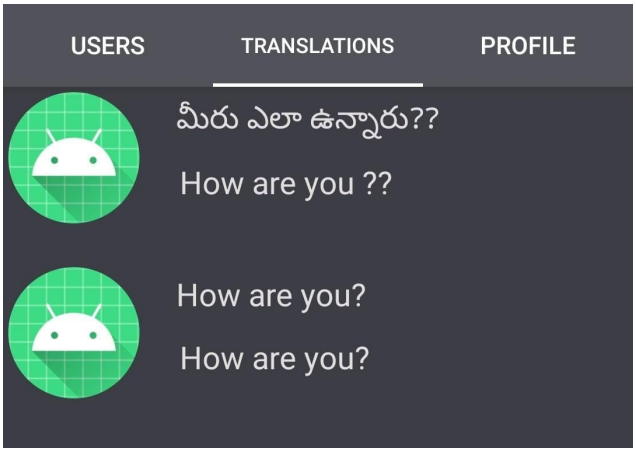
6.3. Registration & Login process





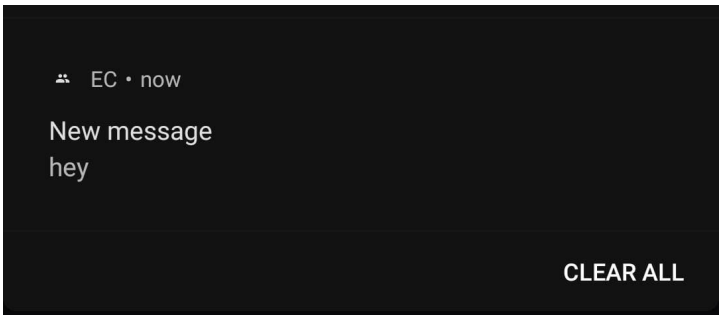
Authentication is set up using firebase and ensures registration and login processes are carried out efficiently. When password is weak during registration or email already exists or password is wrong during login or for any other registration or login issues an appropriate message is displayed on screen.

6.4. Saving recent translations



Recent translations are stored and displayed in the translations tab which include both original and translated messages.

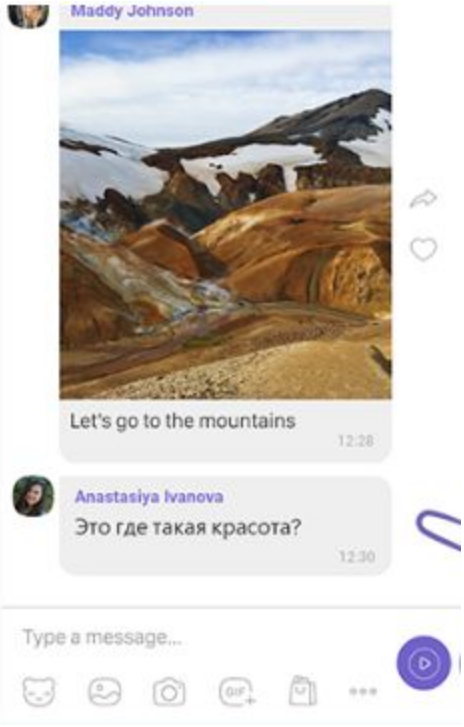
6.5. Notification

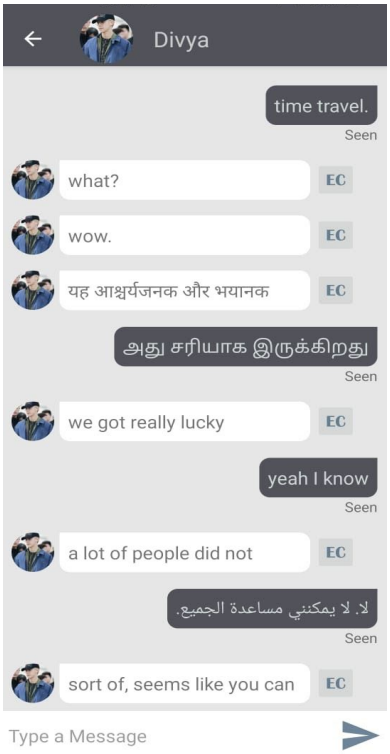


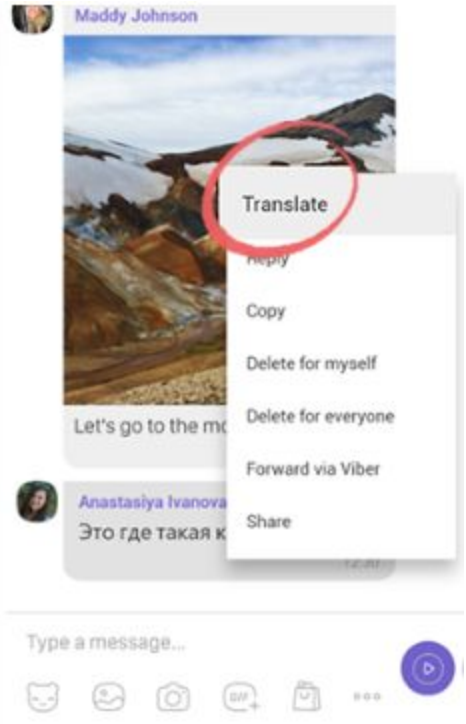
When a user receives a new message a notification is generated and displayed. Which includes the text message received.

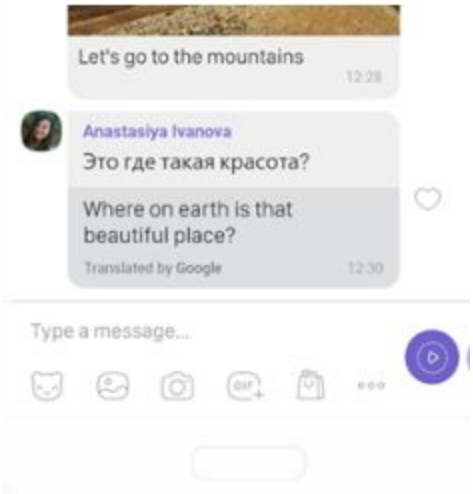
RESULTS

- Some of the existing chat applications with inbuilt translation are skype, viber, TransFire.
- Here, we are trying to compare the efficiency of translation usage in Viber and EasyChat.









It is pretty clear from the above snapshots that easychat is much easier to use and saves a lot of time. Users need not explicitly select language for every message they receive. The tagline ‘Chatting made easier’ is pretty much justified.

DISCUSSION AND FUTURE WORK

- User Interface enhancements
- Optimization of storage and database for better performance.
- Implementation of OCR and Speech to Text conversion.

REFERENCES

Codelabs
https://codelabs.developers.google.com/codelabs/firebase-android/

Firebase Authentication
https://firebase.google.com/docs/auth/android/firebaseui

Firebase Real time Database
https://firebase.google.com/docs/database/android/start

Firebase Storage
https://firebase.google.com/docs/storage/android/start

UI elements
https://developer.android.com/guide/topics/ui/layout/recyclerview
https://github.com/hdodenhof/CircleImageView

Notification
https://developer.android.com/training/notify-user/build-notification

Google translation API

https://stackoverflow.com/questions/8147284/how-to-use-google-translate-api-in-my-java-application

https://medium.com/@yeksancansu/how-to-use-google-translate-api-in-android-studio-projects-7f09cae320c7

https://cloud.google.com/translate/docs

ML toolkits

https://firebase.google.com/docs/ml-kit/

Error queries

https://stackoverflow.com/questions

https://www.codementor.io/collections/learn-android-development-online-bwba0mlle