CSEE 5590-0002/COMP-SCI 490-0002: Web/Mobile Programming

Web Lab - Source Code – Lab1

Team ID – 3

Team Members – Roshna Toke (23)

Manaswini Vedula (24)

Yamini Saraswathi Bommineni(3)

Login Android application

Introduction

Developing a basic Android Login application for displaying user details using Android studio and SQLite Database.

Features

1. The Website allows Users to register through SignUp page and once the users register, they can go back to the Login page. They can log in with their credentials and can view the details.
2. The User also redirects to the Facebook Page by clicking Social Login Button.

Approaches

Creating a Database and a table name “user” and inserting values into it.

```
public DatabaseHelper(@Nullable Context context) { super(context, name: "Login.db",  factory: null ,  version: 1); }

@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL("Create table user(email text primary key, FullName text, University text, Phone text, address text, Password text, Gender text, B

}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("drop table if exists user");

}
public boolean insert(String email, String FullName, String University, String Phone, String address, String Password, String Gender, String Bran
{
    SQLiteDatabase db= this.getWritableDatabase();
    ContentValues contentValues= new ContentValues();
    contentValues.put("email",email);
    contentValues.put("FullName",FullName);
    contentValues.put("University",University);
    contentValues.put("Phone",Phone);
    contentValues.put("address",address);
    contentValues.put("Password",Password);
    contentValues.put("Gender",Gender);
    contentValues.put("Branch",Branch);

    long ins= db.insert( table: "user", nullColumnHack: null,contentValues);
    if(ins==-1) return false;
    else return true;
```

These User define methods to check if email already registered, whether the given credentials match and last method to fetch data related to the “primary key”

```
    }
    public Boolean checkemail(String email){
        SQLiteDatabase db= this.getReadableDatabase();
        Cursor cursor= db.rawQuery( sql: "Select * from user where email=?",new String[]{email});
        if(cursor.getCount()>0) return false;
        else return true;
    }
    public Boolean emailpassword(String email, String Password){
        SQLiteDatabase db = this.getReadableDatabase();
        Cursor cursor= db. rawQuery( sql: "select * from user where email=? and password=?", new String[]{email, Password});
        if(cursor.getCount()>0 ) return true;
        else return false;
    }
    public Cursor info(String email){
        SQLiteDatabase db =this.getReadableDatabase();
        Cursor cursor=db.rawQuery( sql: "select * from user where email=?",new String[]{email});
        return cursor;
    }

}
```

In the Main Login Page Checking the credentials to Login

```
        b1.setOnClickListener((v) -> {
                String email = ed1.getText().toString();
                String Password = ed2.getText().toString();
                Boolean checkemailpassword = db.emailpassword(email, Password);


                if (checkemailpassword == true) {
                    Toast.makeText(getApplicationContext(), text: "Sucessfully Login", Toast.LENGTH_SHORT).show();
                    reDirectToHomePage( );



                } else
                    Toast.makeText(getApplicationContext(), text: "Wrong Credentials", Toast.LENGTH_SHORT).show();

            }
        );
        b2.setOnClickListener((v) -> {
                Intent redirect = new Intent( packageContext: MainActivity.this, signup.class);
                startActivity(redirect);


        });
    }
    public void reDirectToHomePage() {
        String email=ed1.getText().toString();
        Intent redirect = new Intent( packageContext: MainActivity.this, view.class);
        redirect.putExtra( name: "email",email);
        startActivity(redirect);
    }
```

In the Signup page it checks whether all fields are filled or not. Secondly it checks passwords match or not and finally it checks whether the email already exists or not.

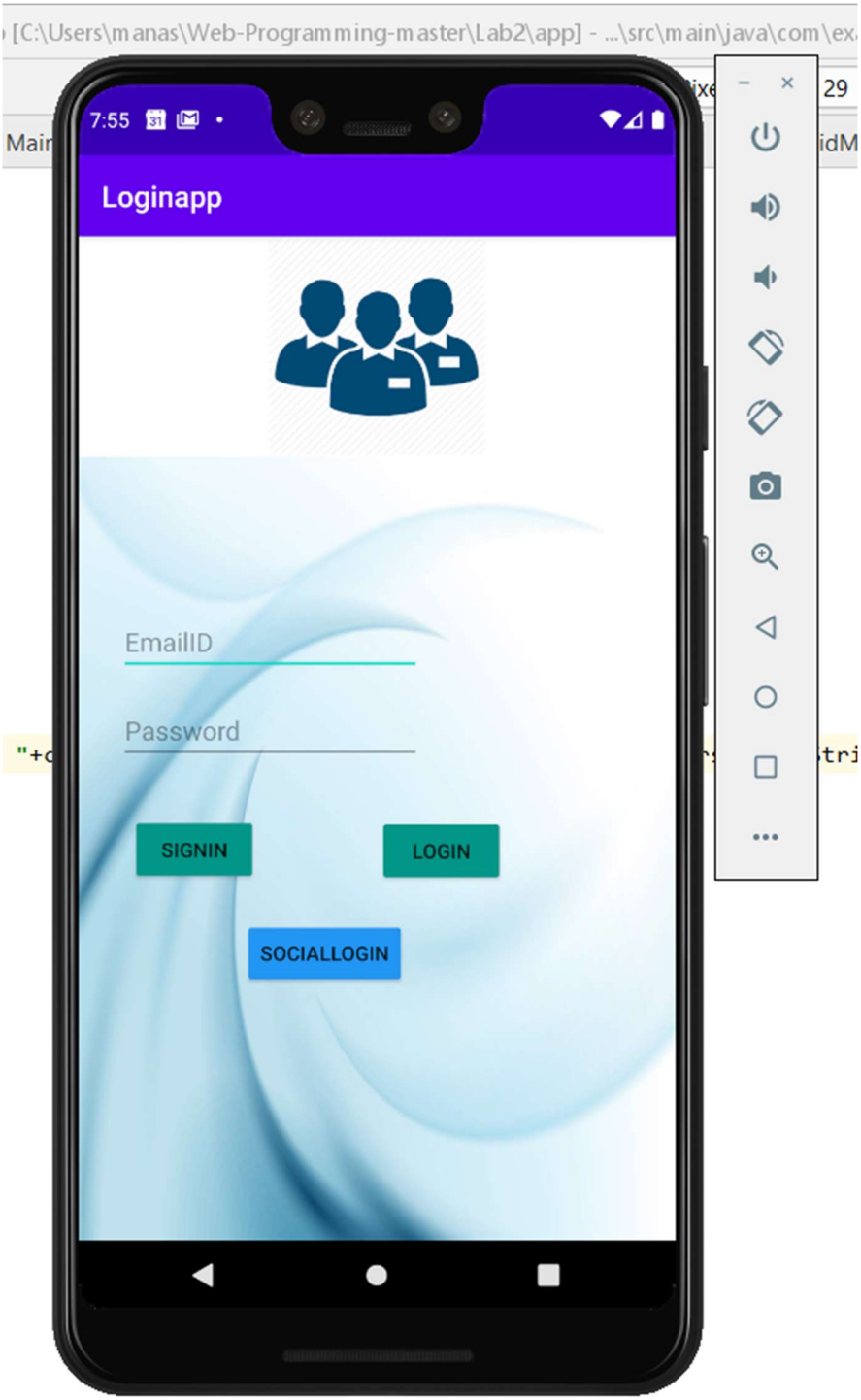
```
            String s2= ed2.getText().toString();
            String s3= ed3.getText().toString();
            String s4= ed4.getText().toString();
            String s5= ed5.getText().toString();
            String s6= ed6.getText().toString();
            String s7= ed7.getText().toString();
            String s8= ed8.getText().toString();
            String s9= ed9.getText().toString();
            if(s1.equals("")||s2.equals("")||s3.equals("")||s4.equals("")||s5.equals("")||s6.equals("")||s7.equals("")||s8.equals("")||s9.equals("")){
                Toast.makeText(getApplicationContext(), text: "Fields are empty", Toast.LENGTH_SHORT).show();
            }
            else{
                if(s6.equals(s7)){
                    Boolean checkemail = db.checkemail(s1);
                    if(checkemail==true){
                        Boolean insert = db.insert(s1,s2,s3,s4,s5,s6,s8,s9);
                        if(insert==true){
                            Toast.makeText(getApplicationContext(), text: "Registered  Sucessfully", Toast.LENGTH_SHORT).show();
                        }
                    }
                    else{
                        Toast.makeText(getApplicationContext(), text: "Email Already Exists", Toast.LENGTH_SHORT).show();
                    }
                }
                Toast.makeText(getApplicationContext(), text: "password do not match", Toast.LENGTH_SHORT).show();
            }

        });
```

In the View page it displays the user details with the help of String builder .

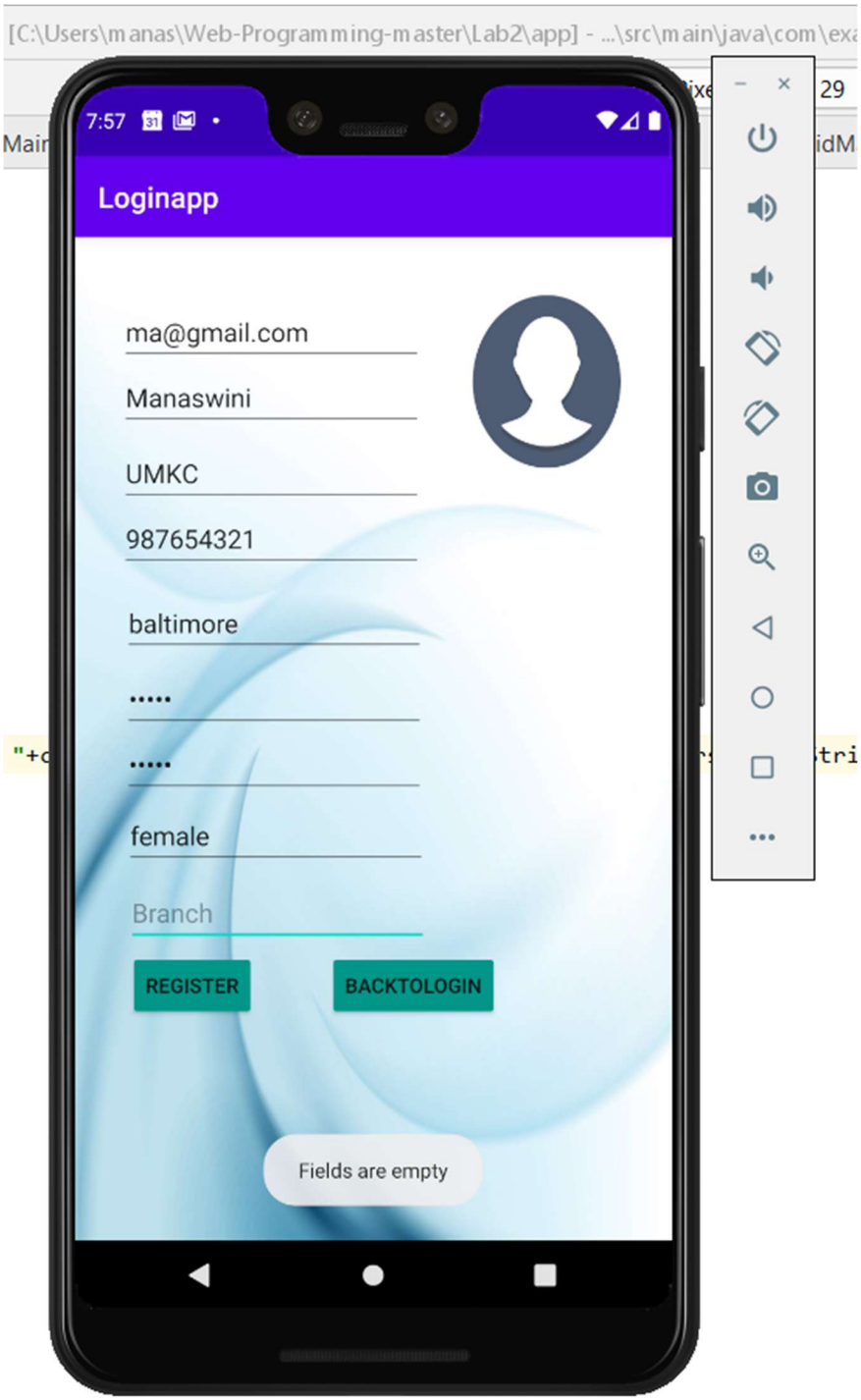
```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_view);
    b2 = (Button)findViewById(R.id.button5);
    t1=(TextView) findViewById(R.id.textView);
    DatabaseHelper db=new DatabaseHelper( context: this);
    String email= getIntent().getStringExtra( name: "email");
    Cursor cursor=db.info(email);
    StringBuilder sb= new StringBuilder();

    while(cursor.moveToNext()){
        sb.append("\nWelcome "+cursor.getString( columnIndex: 1)+"\n Your account details are follows\nEmail: "+ cursor.getString( columnIndex: 0)+"\n
    }

    t1.setText(sb);



    b2.setOnClickListener((v) -> {
            Intent redirect = new Intent( packageContext: view.this, MainActivity.class);
            startActivity(redirect);
    });
}
```

Workflow:

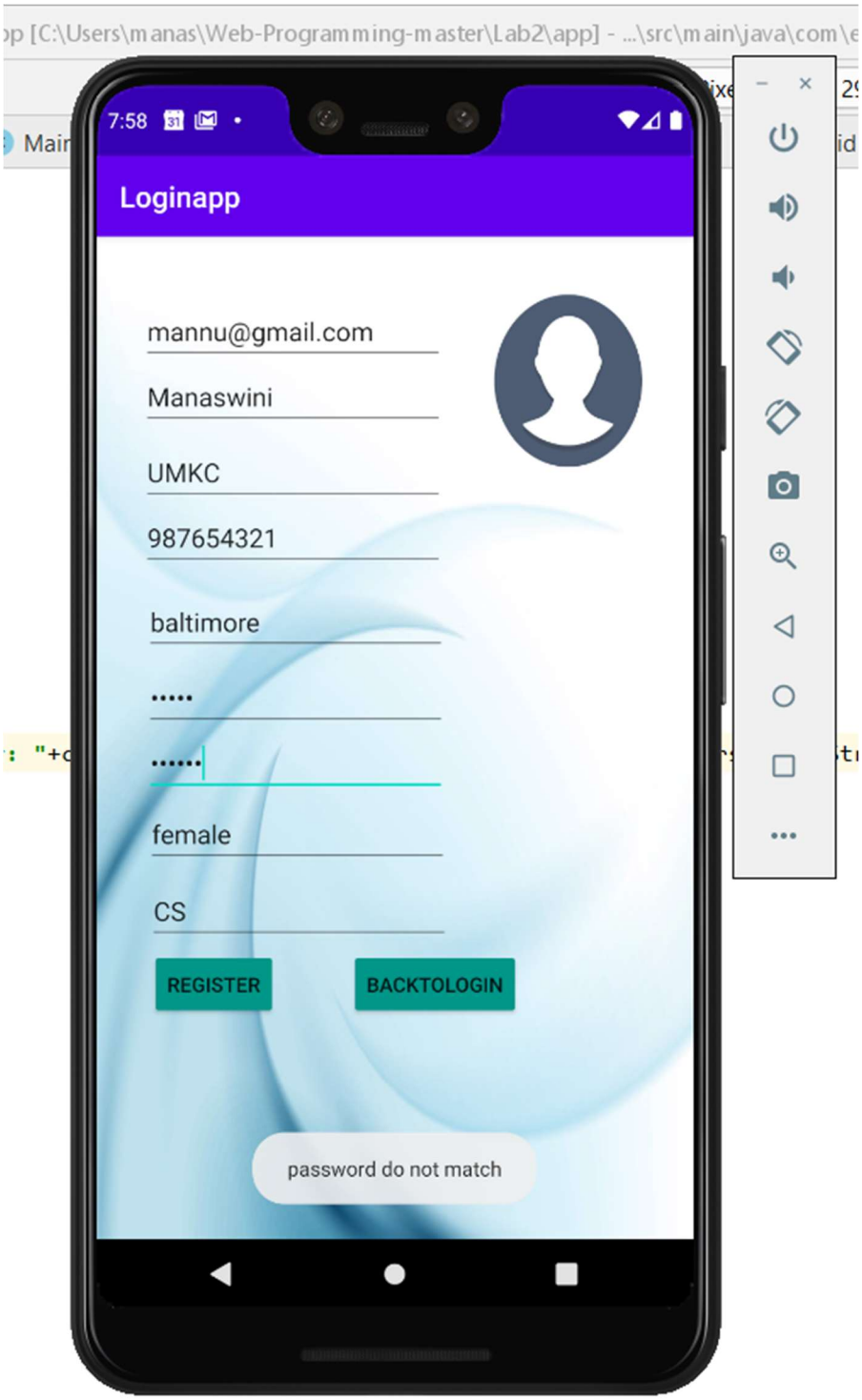
Initial Login Page



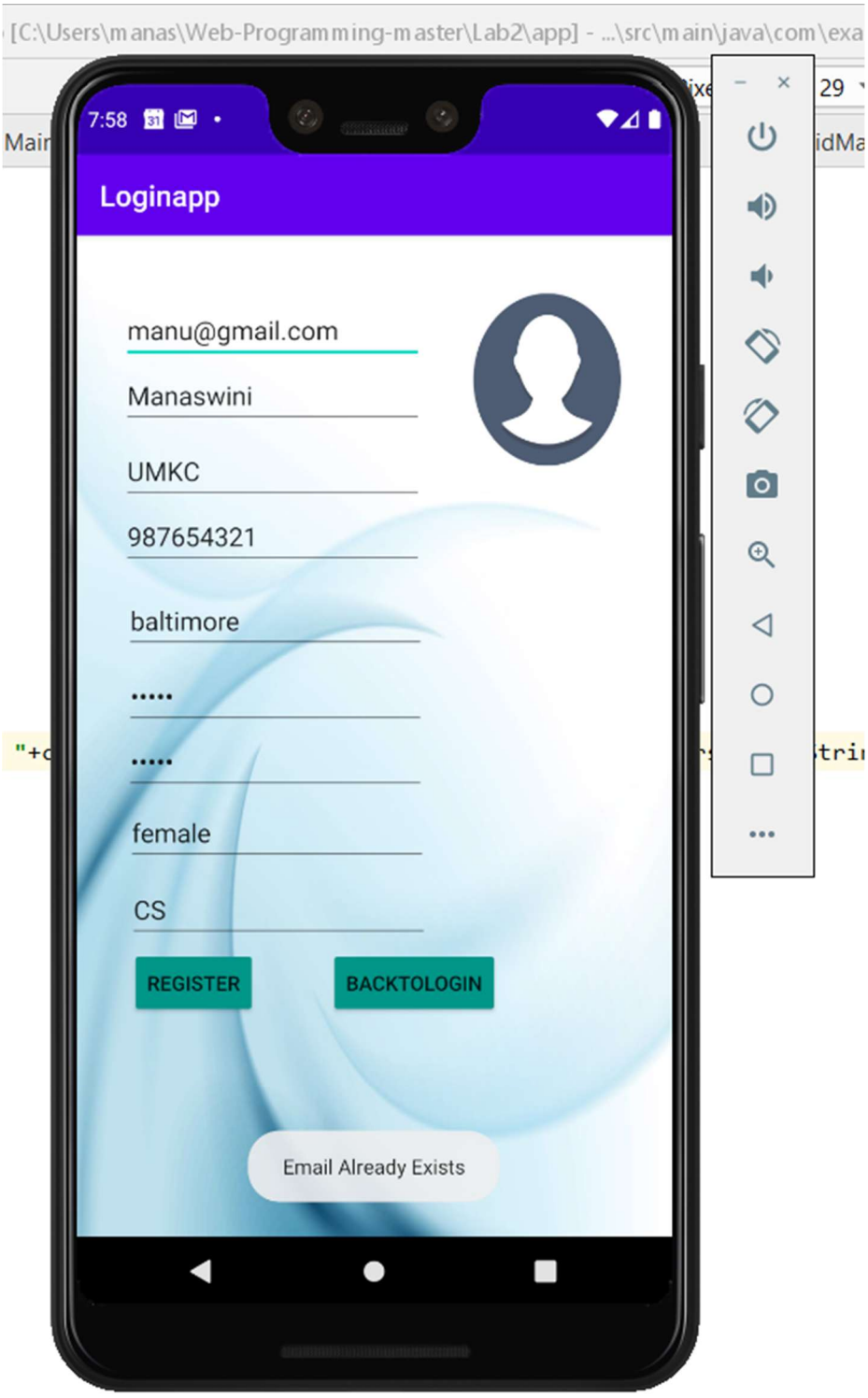
After clicking on Sign In we need to enter all the details else it will display a message saying Fields are empty.



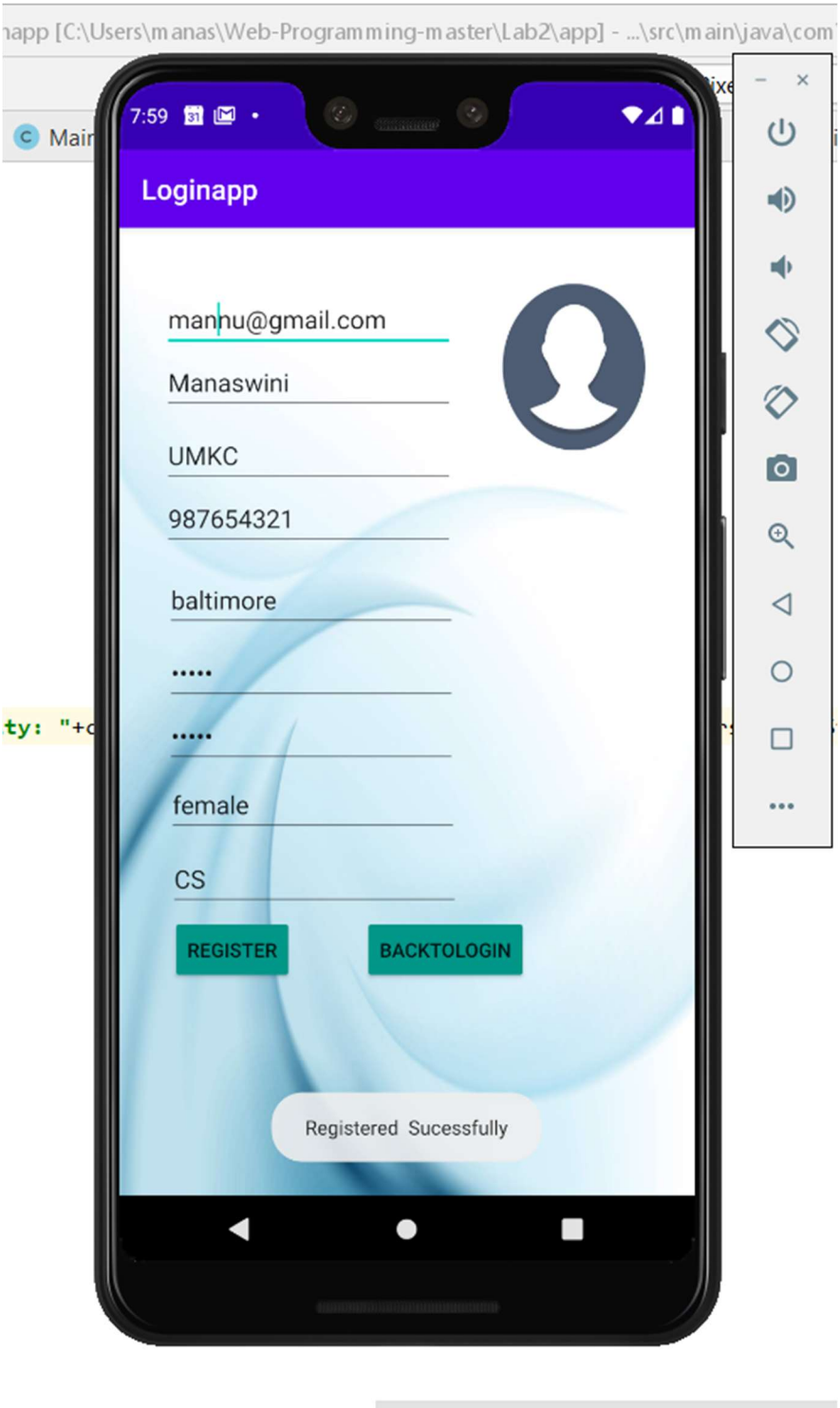
When both passwords do not match it will display a message saying that passwords do not match



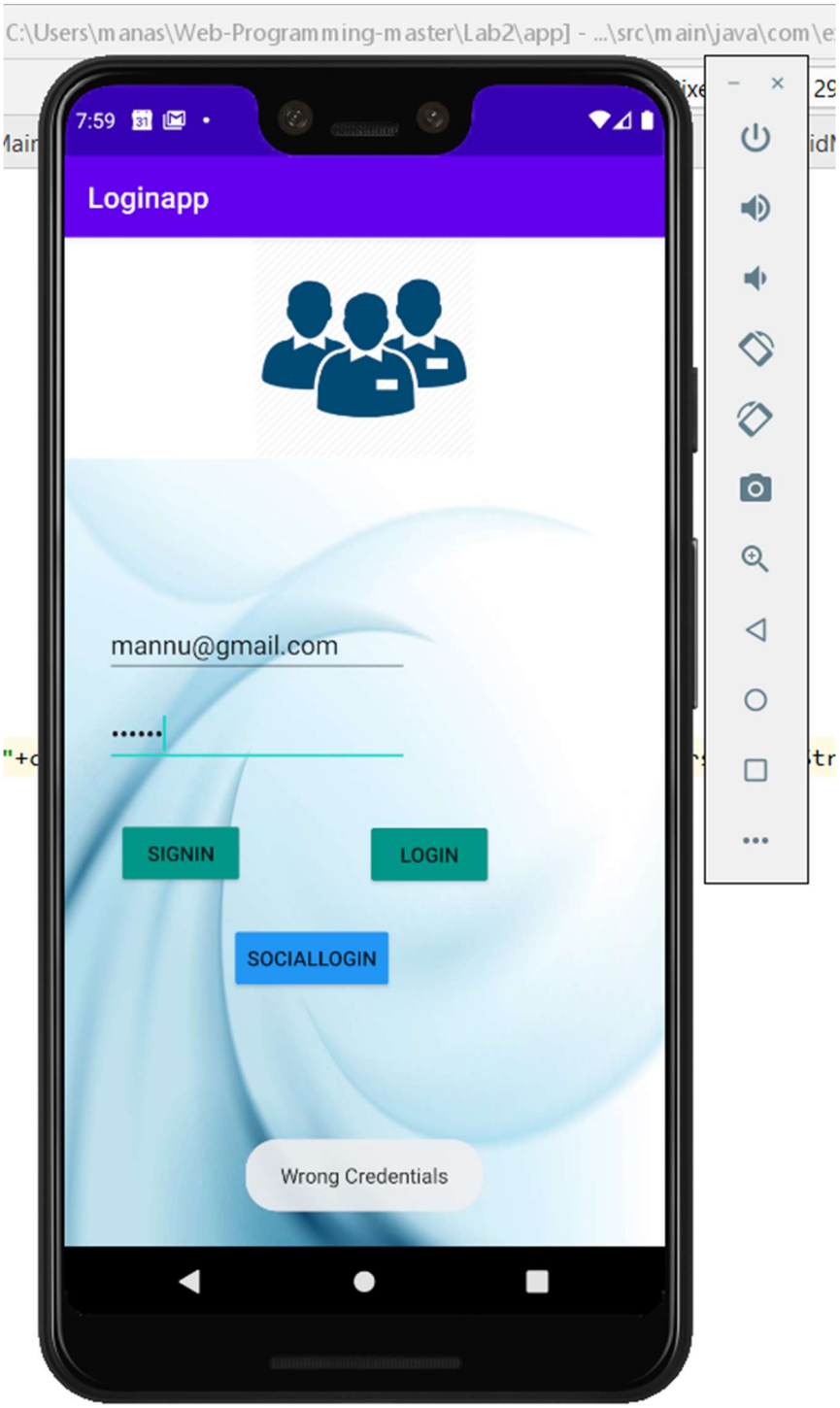
When the email already pre-exists it will display message saying that email already exists.



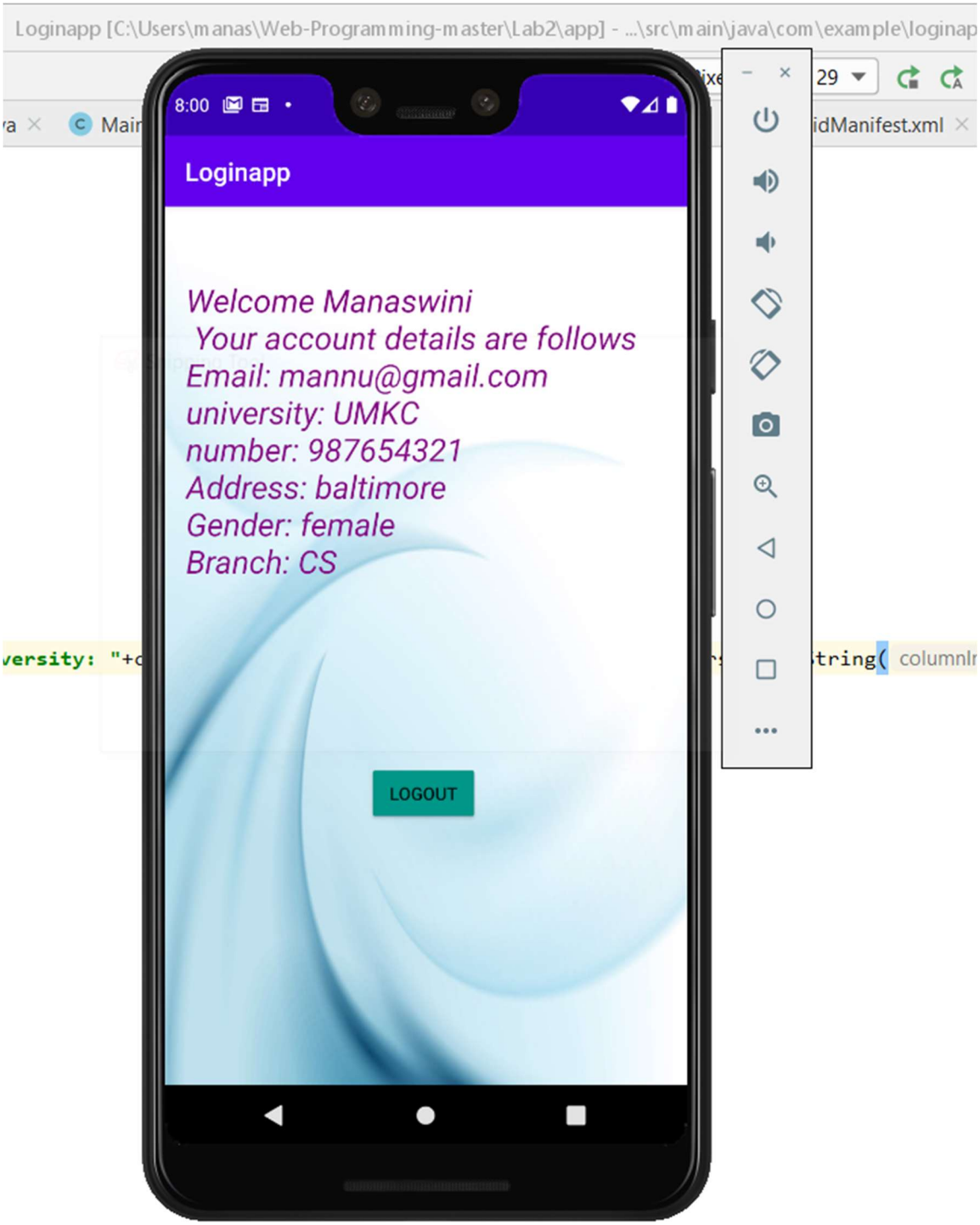
When we enter all details correctly then it will say Register Successfully.



When we enter the wrong credentials for Login then it will pop up a message saying wrong credentials.

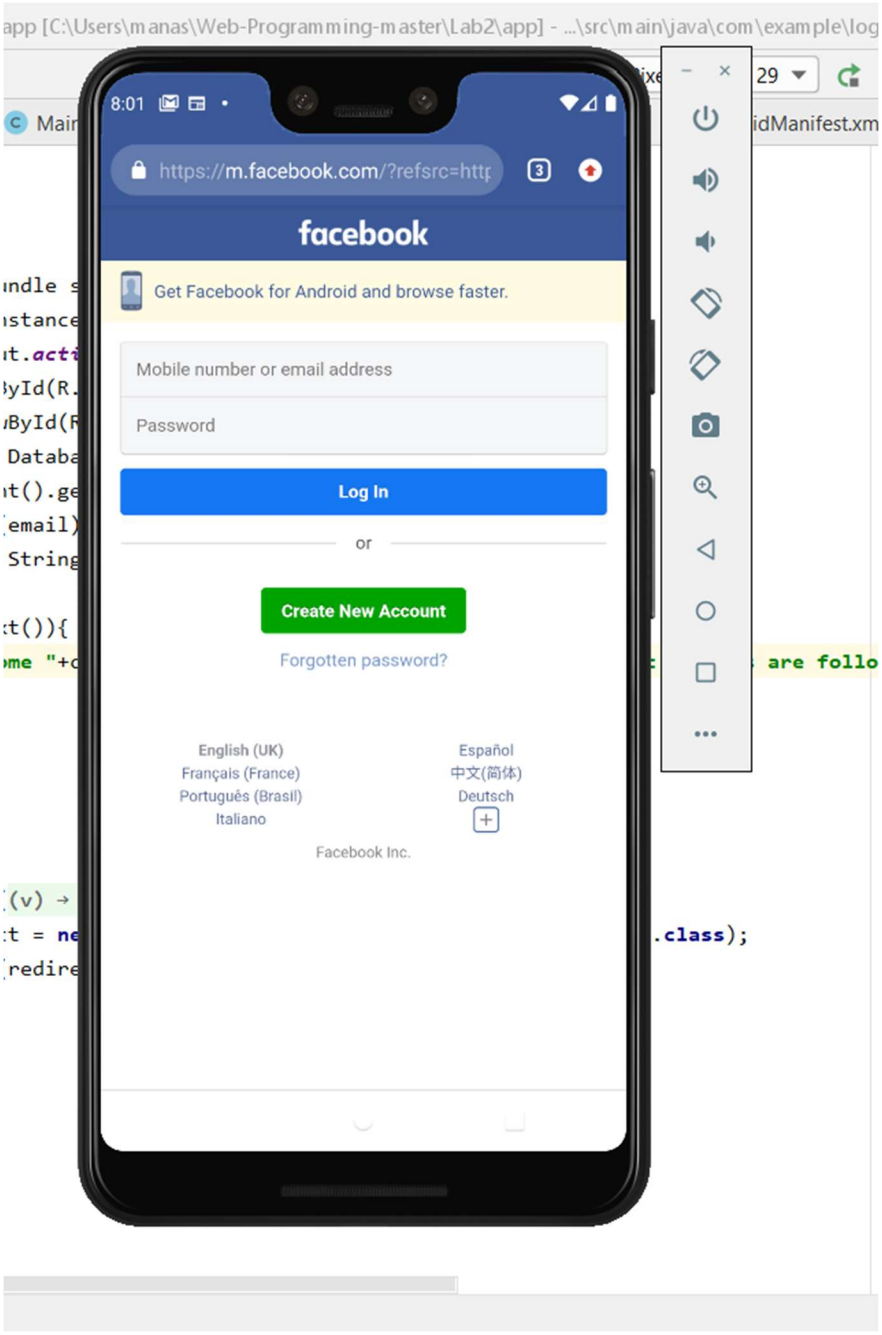


After Successful Login it will display the user details.



When the user clicks on Social Login button it redirects to the Facebook home page.

.



Conclusion:

By doing this task we got familiarity with the Android Studio modules, switching between activities and SQLite Database CRUD operations

TASK 2 – ANDROID CHAT APPLICATION

Introduction

This Application is a **Real time Android Chat Application** that uses **Firebas**e as its storage**.** Very few applications work on peer to peer basis. Peer to Peer basis application creates a connection between your device and each of your contact's devices. Other applications use servers to connect to the devices. WhatsApp uses server's connection to send texts and media.

Objectives

To build a user friendly application with calling, texting, sharing media features and to make it compatible with all android versions.

Approaches

We were inspired from Chat application's features which were developed from simple text messaging to audio calling, media and Location sharing, Video calling.

Parameters

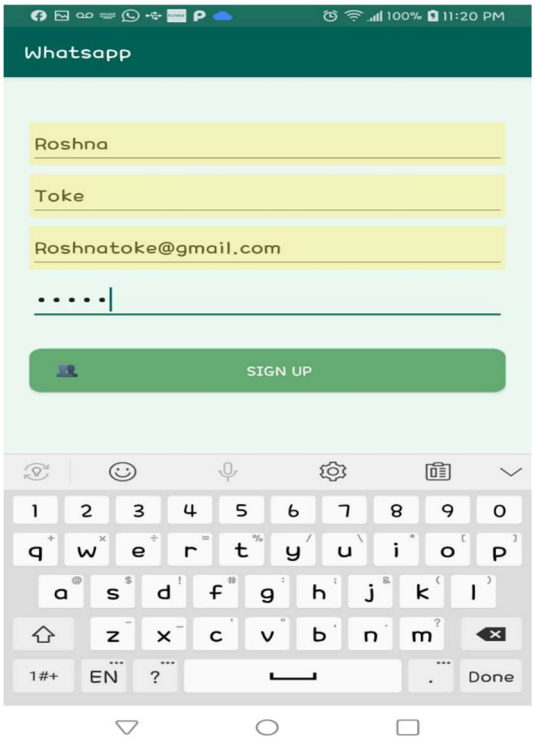
Android studio

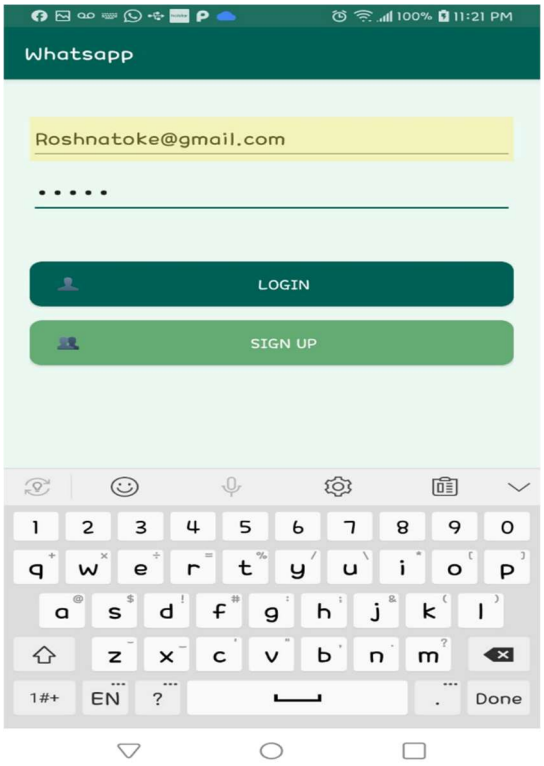
Firebase

SQLite

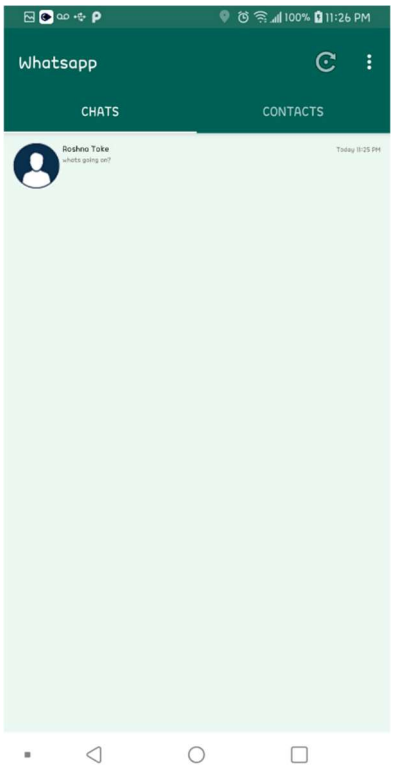
Workflow

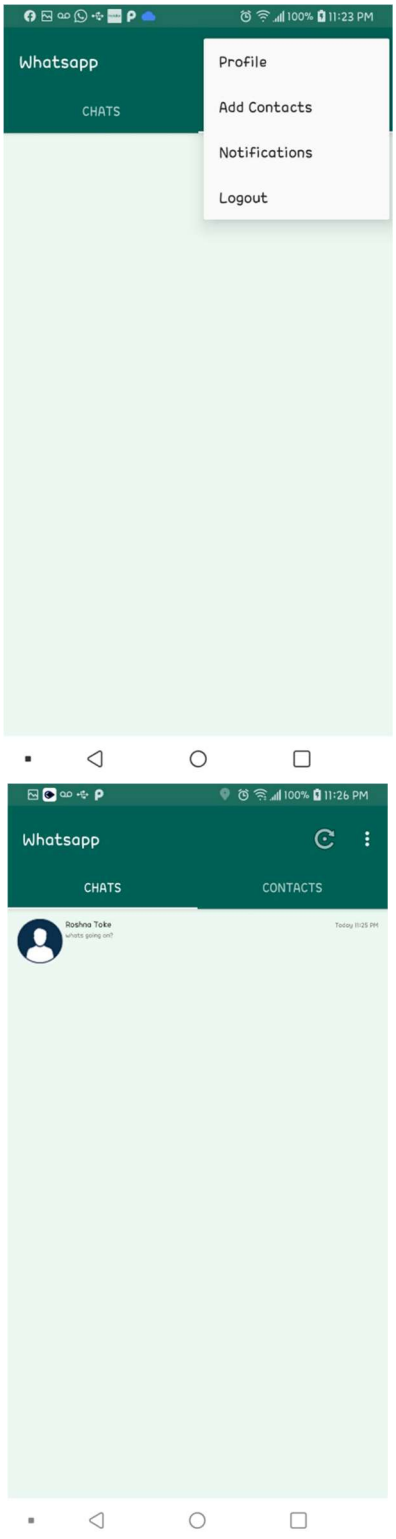
1. When the Application opens, it asks for Email and Password as shown below in the second picture.
   If you are a new user click Sign up and get registered by filling the Sign up form. Then you can Log in with your credentials.
   If you are already a registered user you can directly enter your credentials and Log in.
   Receive messages and Contact requests are displayed on the notification bar.



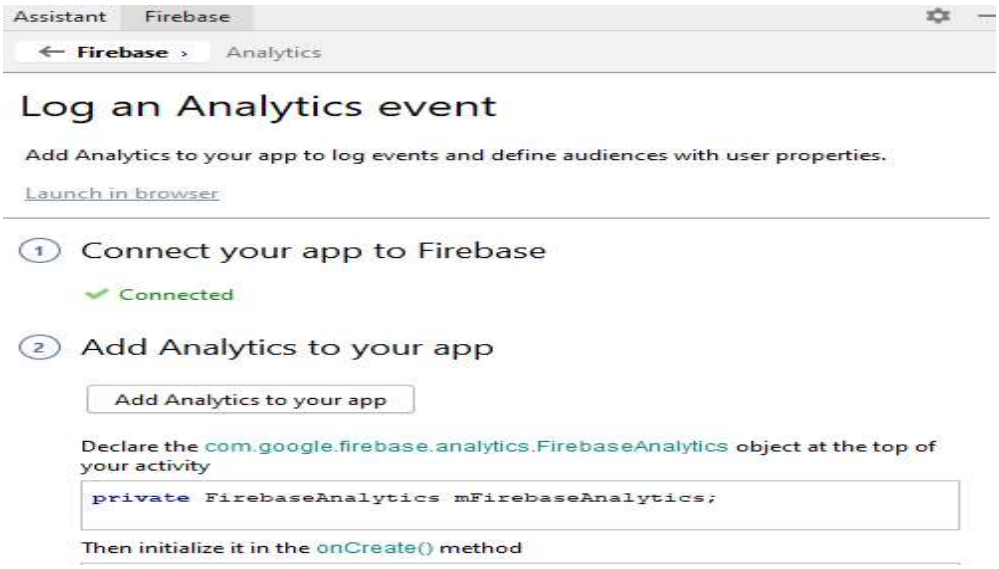


2. The Main page appears like the below image which contains Chat windows and Contact List. The menu bar on the top right has options to check our Profile, Contacts, Notifications and to Log out.

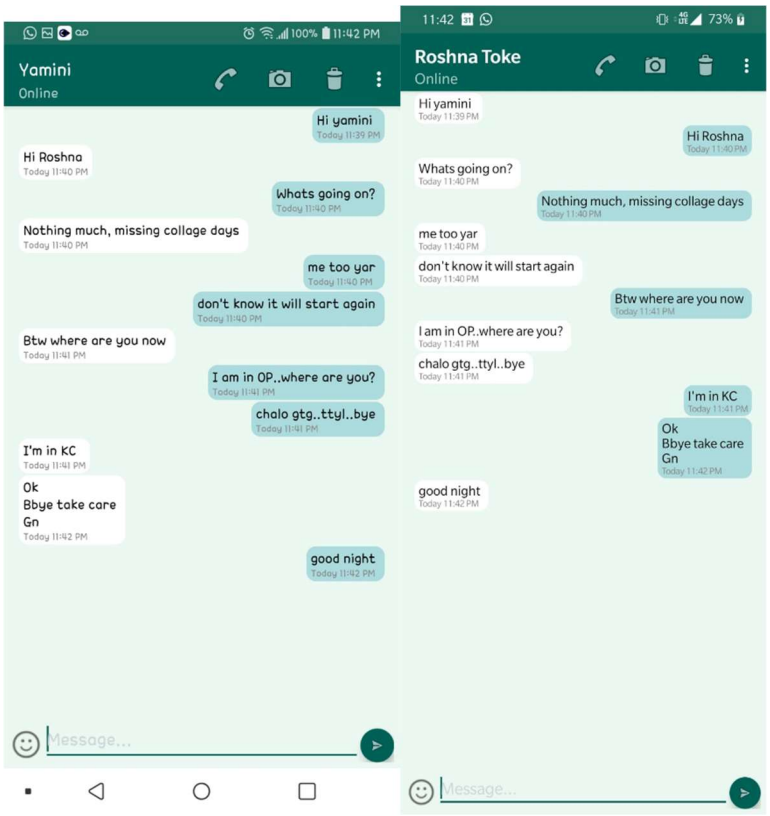




At the very first login, we fetch data from firebase and store in local DB using SQLite.

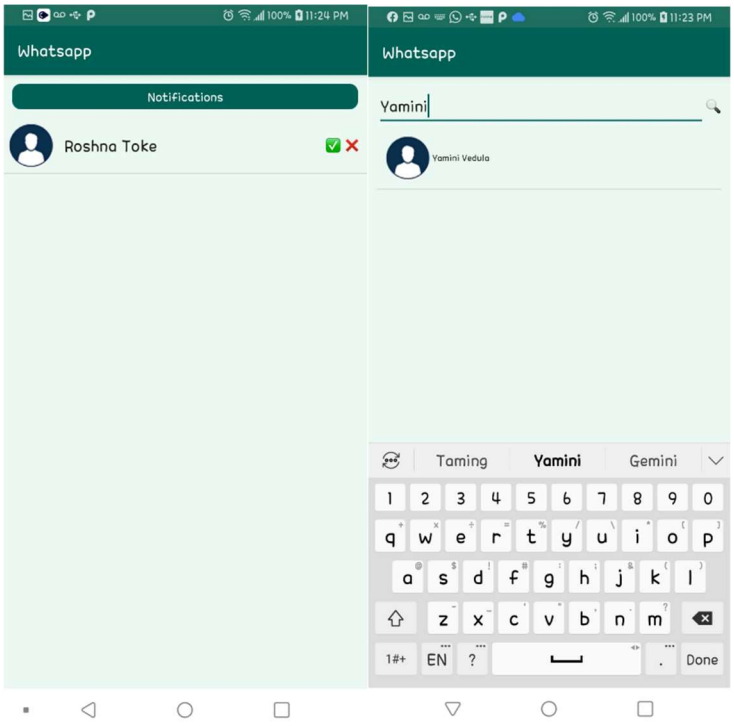


3. User can see his/her past conversations in the Chat windows and can make audio call, share images, Videos and Emojis. Last seen and all message time stamps are displayed. User can send voice notes using the microphone.

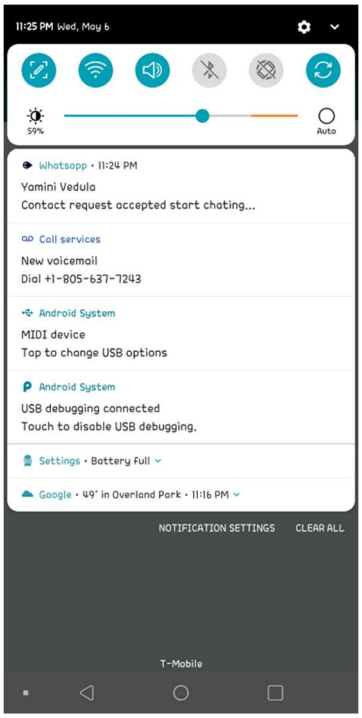


We can delete the conversation and can Update contact details of our friends. Delete Contact option will unfriend the contact

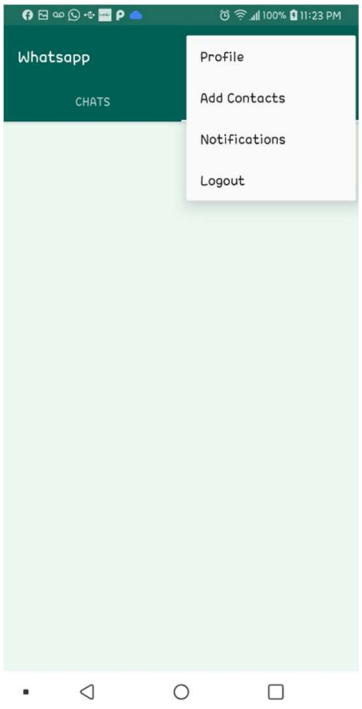
4. To add friends who are not in our contact list we can add them through Add Contact options where he must enter his friend's registered email address. This sends a request to your friend. Request can be Accepted or Rejected. Only after he accepts your request you will be able to chat with him. To see the newly added contact in your contact list, refresh the list and search for the contact.



5. Notifications will be displayed as shown in the below image.



6.User has an option to Log out. If he clicks Log out, a warning message pops up asking you whether you are sure to log out. You will stop receiving notifications if you Log out.



Conclusion:

This application is developed for all android versions. It doesn't work on IOS and Windows. It requires less storage and keeps the user's details secured making it a user friendly application.

References

https://www.androidtutorialpoint.com/firebase/real-time-android-chat-application-using-firebase-tutorial/

https://www.youtube.com/watch?v=gPqJcPtN18I&list=PLxefhmF0pcPmtdoud8f64EpgapkclCllj