Introduction

Our project is developed using Android and MySQL Tool and technology.

Our project code is totally dynamic.

Our this project is used for Chating with Friends.

Our project is Mobile based and User can easily access this Application.

Acknowledgement

It is our great pleasure to present our project report on “ChatMe” which we conceived the 6th Semester of BCA affiliated to BKNMU , Junagadh.

We are very thankful to all, who have helped us in preparing this project. We are very happy to present this “ PROJECT REPORT ”. Before you, expecting that you will acknowledge. It is the matter of great pleasure for us. As a part of the academic study as the students of 6th sem of BCA.

We are required to undergo training for software project to the organization in order to obtain the practical knowledge and broaden our view regarding the same.

we would like to Mr. Dhaval Trivedi Sir. Who has contributed the required information with special interest and guidance throghout out project work and treated us as their younger brothers. We are indebted to all the people of our collage, S.S.S.D.I.I.T Computer Collage, concerned, who have actively involved in marketing of the project.

|  |  |  |
| --- | --- | --- |
| Index | Title | Page No |
| 1 | Project Profile | 4 |
| 2 | Requirement | 5 |
| 3 | Feasibility Study | 7 |
| 4 | Hardware And Software Requirement | 9 |
| 5 | Data Dictionary | 10 |
| 6 | Normalization | 13 |
| 7 | Data-Flow Diagram | 16 |
| 8 | Pert Chart | 21 |
| 9 | Gannt Chart | 22 |
| 10 | Use Case Diagram | 23 |
| 11 | E.R. Diagram | 24 |
| 12 | Testing | 25 |
| 13 | Implementation And Maintenance | 28 |
| 14 | Limitations | 29 |
| 15 | Future Enhancement | 29 |
| 16 | Screen Layout of Project and Coding | 30 |

Project Profile

**Project Title** :- ChatMe

**Developed By** :- 1) Gondaliya Dharmesh

**2)** Maheta Parth

**Class** :- BCA Sem – 6th

**Front End** :- Android

**Back End** :- MySQL

**Collage :-** SSSDIIT Collage Junagadh

**Project Submit** :- Bhakta Kavi Narsinh Mehta University

**Project Evalutor** :- Mr. Dhaval Trivedi sir

**Academic Year:- 2019-2020**

Requirement

**Questionary**

Which Type of application you need?

Mobile Application

What is the title of your software?

ChatMe

What is purpose of your software?

The main purpose of application is communication with people without money.

Where to store your application data?

Online In Website

Do you have any Application in use?

No

Do you have any website?

No

Do you have any domain name and hosting?

No

How much time perioude you need for completion of your

application?

Within 3 months

Your Application will cost 25000 rupees for

development?

Ok, Done !

Feasibility Study

Economical Feasibility

Considering the client’s need for a ChatMe project, the

cost of Designing and Development cost around Rs. 25000.

It also covers the cost of internet access.

The costs incurred for a ChatMe project are in keeping

with the general market, and are acceptable to the client.

Technical Feasibility

User cost for using ChatMe Application are as follows:

* Android Operting System
* Android Version 5.0 or higher
* Internet Connectivity
* 1 GB or more Ram

The client can afford the cost to use the software.

Operational Feasibility

The user of the Application is must know English,

Know how to use internet, know how to install and uninstall

application, know how to update application.

* The client is Technically, Economically and

Operationally Feasible. So, this project is acceptable.

Hardware Requirement

Android phone

Ram

Phone Memory (Minimum 512 mb)

Software Requirement

Android Operating System

Android Version Lolipop or higher

Data Dictionary

Database Name :- chat\_me

Tables:-

1. User\_detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Key | Default | Extra |
| ID | Smallint(5) | Primary | None | Auto\_Increment |
| NAME | Varchar(30) | Primary | None | --- |
| EMAIL | Varchar(30) | --- | None | --- |
| PASSWORD | Text | --- | None | --- |
| PICTURE | Text | --- | ‘demo.jpg’ | --- |
| STATUS | Text | --- | ‘Hey… IM Using ChatMe’ | --- |
| DATE\_TIME | Timestamp | --- | Current\_timestamp | --- |
| LAST\_SEEN | Timestamp | --- | Current\_timestamp | --- |

1. statuses

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Key |
| UID | Smallint(5) | Foreign key |
| FILEPATH | Varchar(100) | --- |
| DATE\_TIME | Timestamp | --- |

1. friend

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Key |
| UID | Smallint(5) | Foreign key |
| FID | Smallint(5) | Foreign key |

1. request

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Key |
| UID | Smallint(5) | Foreign key |
| FID | Smallint(5) | Foreign key |

1. chat

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Key | Extra |
| ID | lint(10) | Primary | Auto\_Increment |
| \_FROM | Smallint(5) | Foreign key | --- |
| ­\_TO | Smallint(5) | Foreign key | --- |
| MESSAGE | Text | --- | --- |
| IS\_FILE | Tinyint(1) | --- | --- |
| FILE\_PATH | Text | --- | --- |
| STATUS | Smallint(1) | --- | --- |

Normalization

Database Name :- chat\_me

Tables:-

1. User\_detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Key | Default | Extra |
| ID | Smallint(5) | Primary | None | Auto\_Increment |
| NAME | Varchar(30) | Primary | None | --- |
| EMAIL | Varchar(30) | --- | None | --- |
| PASSWORD | Text | --- | None | --- |
| PICTURE | Text | --- | ‘demo.jpg’ | --- |
| STATUS | Text | --- | ‘Hey… IM Using ChatMe’ | --- |
| DATE\_TIME | Timestamp | --- | Current\_timestamp | --- |
| LAST\_SEEN | Timestamp | --- | Current\_timestamp | --- |

1. statuses

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Key |
| UID | Smallint(5) | Foreign key |
| FILEPATH | Varchar(100) | --- |
| DATE\_TIME | Timestamp | --- |

1. friend

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Key |
| UID | Smallint(5) | Foreign key |
| FID | Smallint(5) | Foreign key |

1. request

|  |  |  |
| --- | --- | --- |
| Field Name | Data Type | Key |
| UID | Smallint(5) | Foreign key |
| FID | Smallint(5) | Foreign key |

1. chat

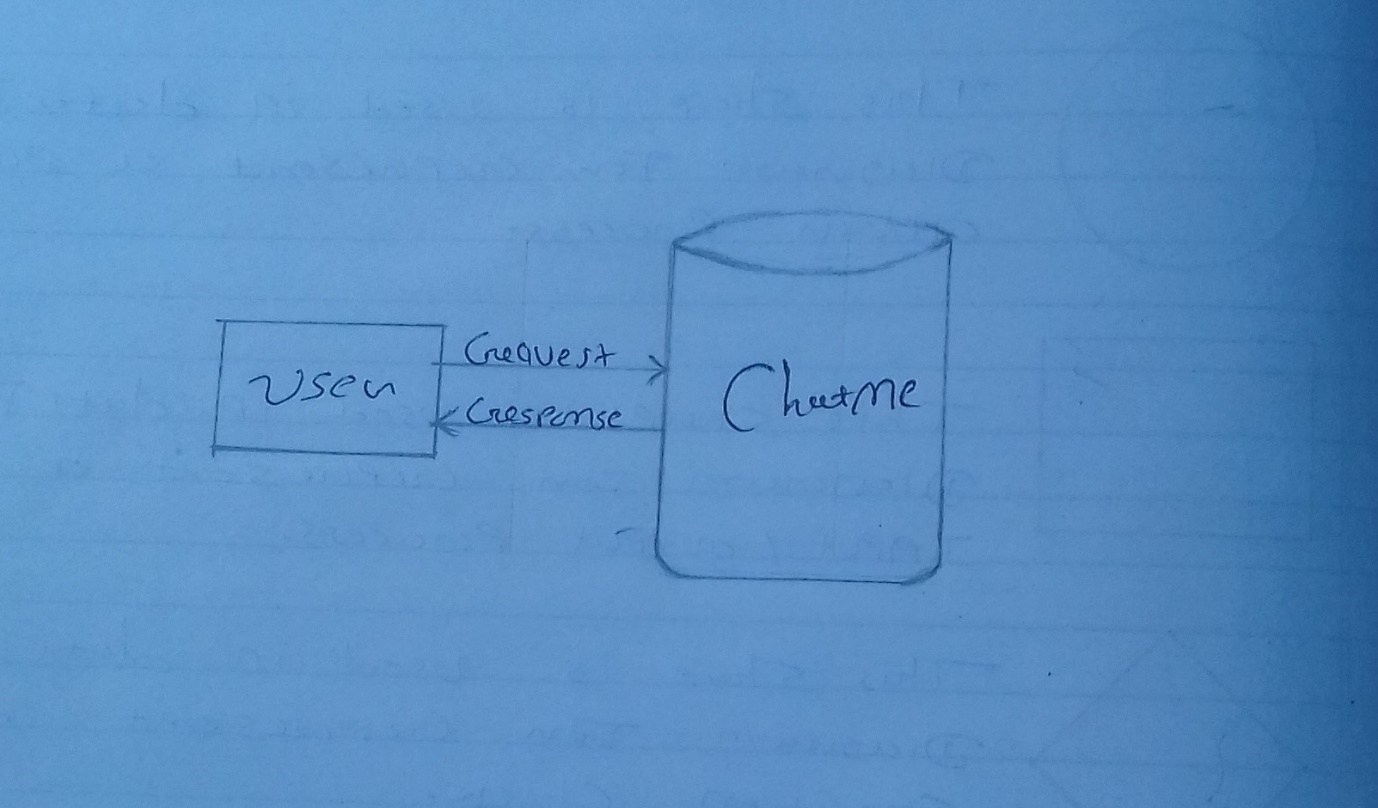
|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Key | Extra |
| ID | lint(10) | Primary | Auto\_Increment |
| \_FROM | Smallint(5) | Foreign key | --- |
| ­\_TO | Smallint(5) | Foreign key | --- |
| MESSAGE | Text | --- | --- |
| IS\_FILE | Tinyint(1) | --- | --- |
| FILE\_PATH | Text | --- | --- |
| STATUS | Smallint(1) | --- | --- |

Data – Flow Diagram

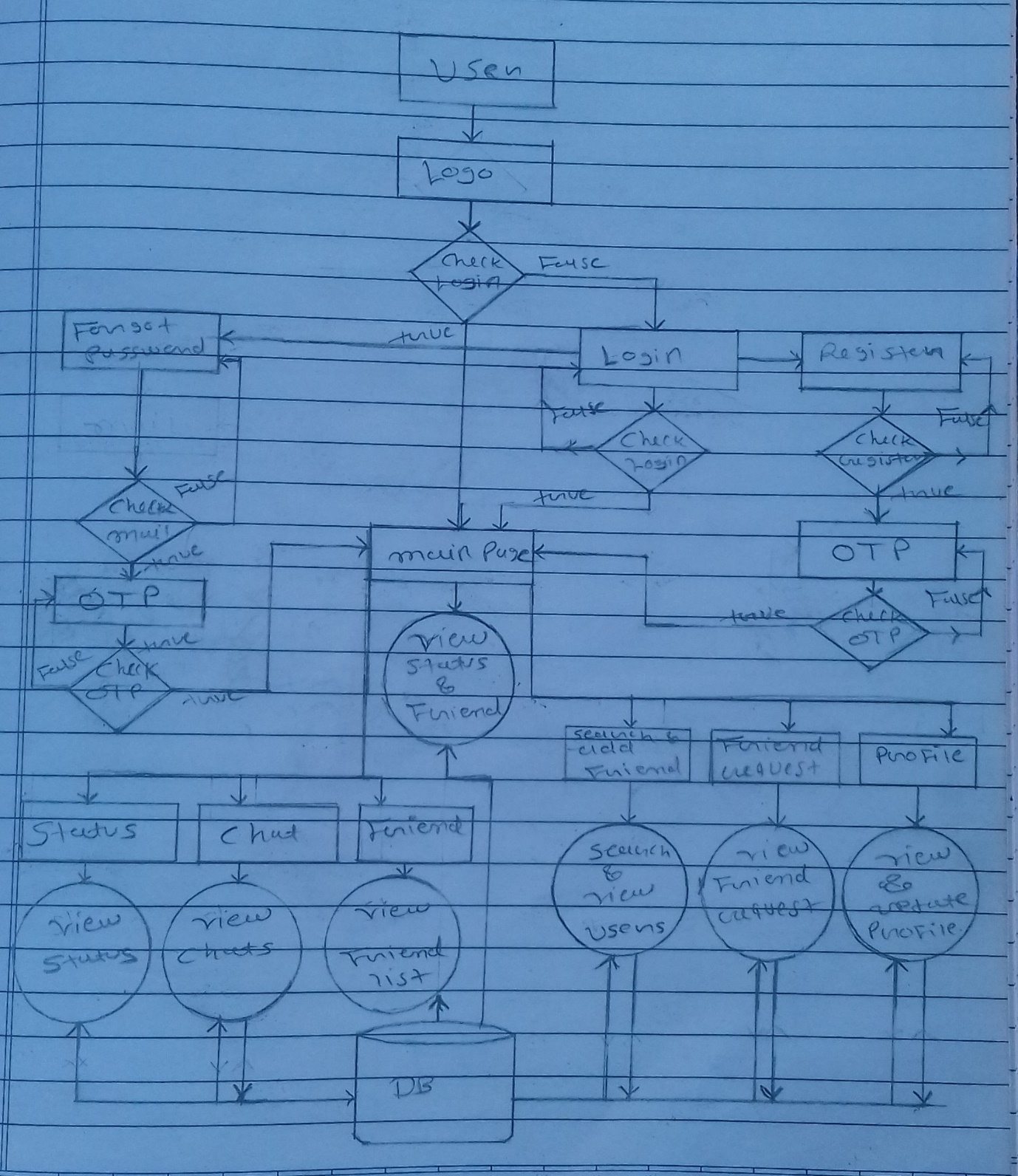
Symbols

|  |  |
| --- | --- |
| Symbol | Description |
|  | This Shape is used in data-flow diagram for represent a particular process. |
|  | This Shape is used in data-flow diagram for represent a input / output process. |
|  | This Shape is used in data-flow diagram for represent a condition check. |
|  | This Shape is used in data-flow diagram for represent a condition check. |
|  | This Shape is used in data-flow diagram for represent a flow of data |
|  | This Shape is used in data-flow diagram for represent a particular database |

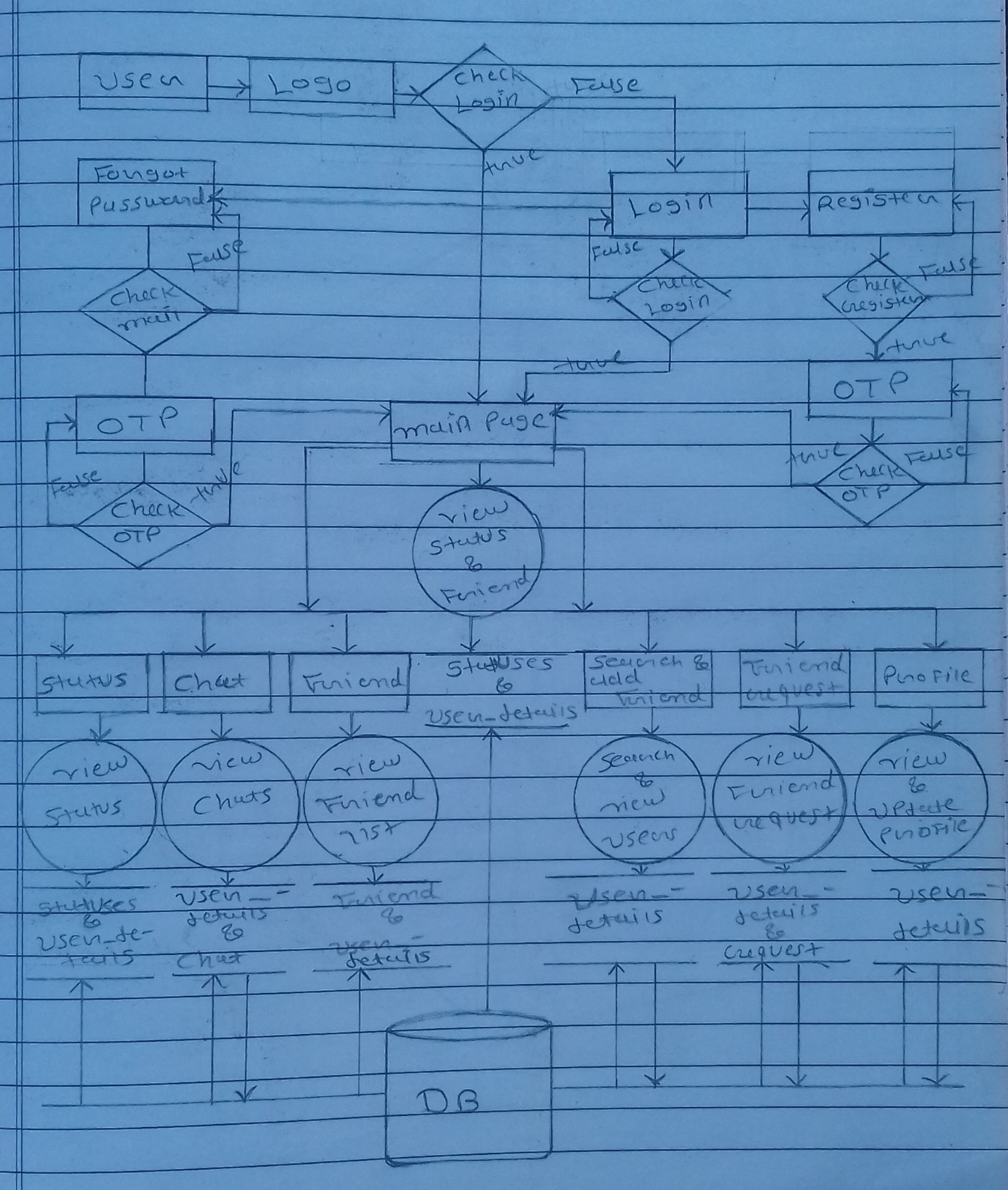
0 Level Data – Flow Diagram



1 Level Data – Flow Diagram



2 Level Data – Flow Diagram



Pert Chart

2-2-20

10-12-19

20-12-19

10-2-20

15-12-19

5-12-19

**Data Dictionary**

**Testing**

**Coding**

10-2-20

25-11-19

**Requirements**

**Implementation**

**Documentation**

**Feasibility Study**

**Designing**

**Maintenance**

Gannt Chart

10-2-2020

10-2-2020

3-2-2019

20-12-19

2-2-2020

15-12-19

10-12-19

5-12-2019

25-11-19

**Requirements**

**Feasibility Study**

**Data Dictionary**

**Designing**

**Coding**

**Testing**

**Documentation**

**Maintenance**

**Implementation**

**Use Case Diagram**

**Logo**

**Login**

**Register**

**Forgot Password**

USER

**OTP**

**Main Page**

**Status**

**Chat**

**Friend**

**Search & Add Friend**

**Friend request**

**Profile**

E.R. Diagram

|  |
| --- |
| Chat |
| ID |
| ­\_FROM |
| \_TO |
| MESSAGE |
| IS\_FILE |
| FILE\_PATH |
| STATUS |

|  |
| --- |
| User\_detail |
| ID |
| NAME |
| EMAIL |
| PASSWORD |
| PICTURE |
| STATUS |
| DATE\_TIME |
| LAST\_SEEN |

|  |
| --- |
| request |
| UID |
| FID |

|  |
| --- |
| Status |
| UID |
| FILE\_PATH |
| DATE\_TIME |

|  |
| --- |
| friend |
| UID |
| FID |

Testing

Testing is that step to software engineering process that could be viewed destructive rather than constructive.

In a software development project, errors can be injected at any stage during the development. Also some requirements errors and design errors can remain undetected. Ultimately, these errors will be reflected in the code.

Since the code is frequently the only product that can be executed and whose actual behavior can be observed, testing is the phase where the errors remaining form earlier phases also must be detected. Hence, testing performs a very critical role for quality assurance and for ensuring the reliability of the software.

Testing Methods

* **White Box Testing**

In this system the testing is to search the undiscovered errors. White Box Testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software developer can derive test cases.

* Guarantee all paths will redirects the another

page as the Individual working place.

* Log in page is directed as the User name and

Password both are same than go in next page

otherwise display the Error. And Put again user

name and password.

* **Black Box Testing**

Black box testing methods focus on the functional requirements of the software. That is, black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

* Incorrect Information fills in the Project entry,

Module entry and Functionality entry Form so

generate errors Massage

* In this system when the database structure is

different from insert value than give the Error.

**Implementation And Maintenance**

A software implementation method is a systematically structured approach to effectively integrate a software based service or component into workflow of an organizational structure or an individual end user.

we used parallel running to implement my software.

There is an error executing this Application. In this application connection error are execute, and it is not connect to the online database.

In this error we solved this problem and implemented it.

Limitations

* In this Mobile Application project can not upload

posts.

* In this Mobile Application project can not upload

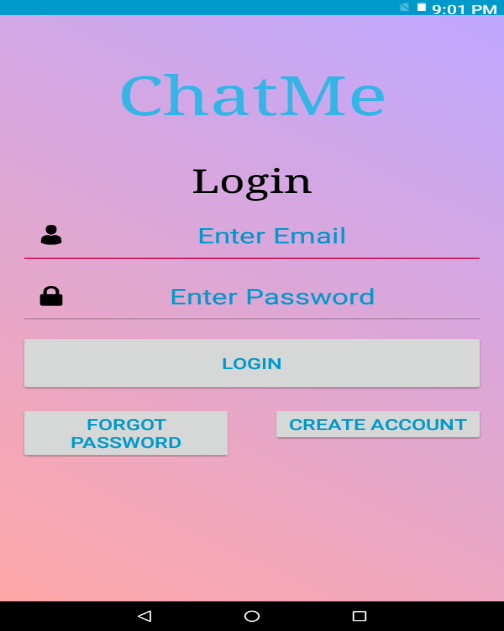
video status.

Future Enhancement

* Chat Backup
* Share Files Between Friends
* Upload Posts
* Upload Video Status

Screen Layout And Coding

**Login Activity**



public class LoginPage extends AppCompatActivity {

    Animation anim;

    EditText user,pass;

    String UserId,Password;

    String url = "http://192.168.43.191/chatme/login.php";

    Database\_Handler db\_handler;

    @Override

    protected void onCreate(@Nullable Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_login\_page);

StringRequest stringRequest = new StringRequest(Request.Method.POST, url,

new Response.Listener<String>() {

@Override

public void onResponse(String response) {

                                    try {

                                        JSONArray jsonArray = new JSONArray(response);

JSONObject jsonObject = jsonArray.getJSONObject(0);

 if(jsonObject.getString("login").equals("Login Success")) {

     user.setText("");

     pass.setText("");

   try{

   db\_handler = new Database\_Handler(getApplicationContext());

     addToMy\_Detail(jsonObject);

     db\_handler.deleteuser();

     db\_handler.addusr(My\_Detail.My\_ID,My\_Detail.My\_NAME,My\_Detail.My\_EMAIL,My\_Detail.My\_PASSWORD,My\_Detail.My\_PICTURE, My\_Detail.My\_STATUS);

  }

  catch (Exception ex){

         ex.printStackTrace();

                }

Intent in = new Intent(getApplicationContext(), Main.class);

                    startActivity(in);

                                       }

                                        else if(jsonObject.getString("login").toString().equals("Login Fail")){

                                            FancyToast.makeText(getApplicationContext(),"Login Fail\nEnter Currect UserId And Password",Toast.LENGTH\_LONG,FancyToast.ERROR,false).show();

                                            return;

                                        }

                                    } catch (JSONException e) {

                                        e.printStackTrace();

                                    }

                                }

                            },

                            new Response.ErrorListener() {

                                @Override

                                public void onErrorResponse(VolleyError error) {

                                    FancyToast.makeText(getApplicationContext(),"Error :- "+error.getMessage().toString(),Toast.LENGTH\_LONG,FancyToast.ERROR,false).show();

                                }

                            }){

                        @Override

                        protected Map<String, String> getParams() throws AuthFailureError {

   Map<String,String> params = new HashMap<String,String>();

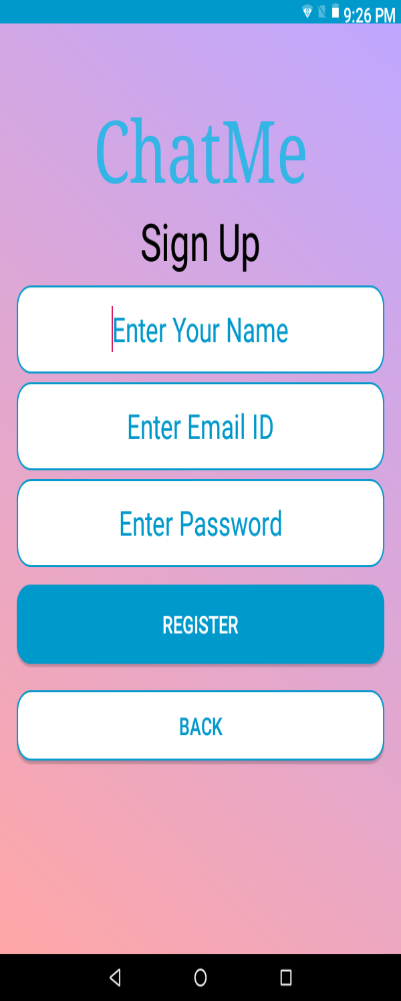
                            params.put("l\_user",UserId);

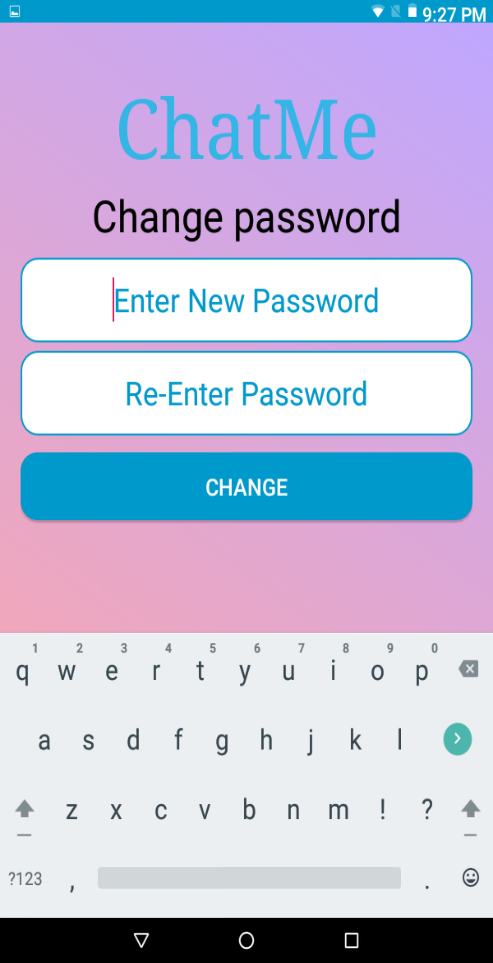
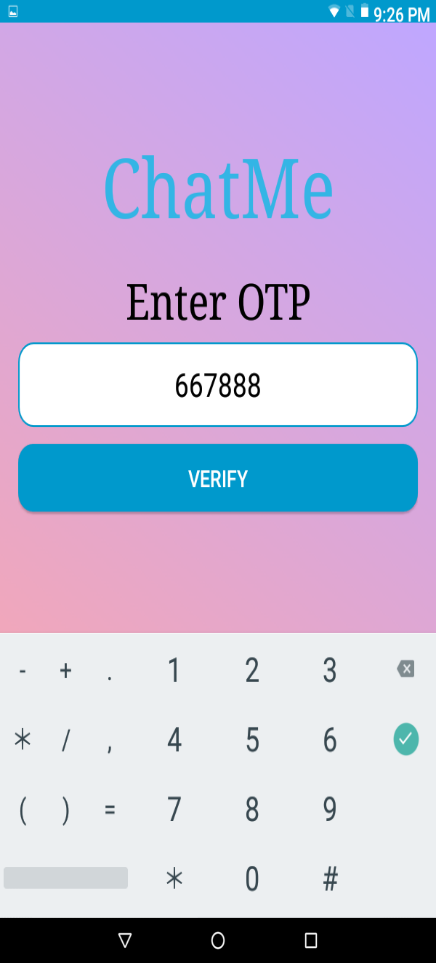
                            params.put("l\_pass",Password);

                            return params; }

                    };

  MySingleton.getInstance(LoginPage.this).addRequestQueue(stringRequest);

**Register And VeriFication Activity**

****

public class RegisterPage extends AppCompatActivity {

    Animation anim;

    EditText name,user,pass;

    String Name,UserId,Password;

    LinearLayout ll;

    String url = "http://192.168.43.191/chatme/check.php";

    @Override

    protected void onCreate(@Nullable Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_register\_page);

        name = (EditText) findViewById(R.id.nameet);

        user = (EditText) findViewById(R.id.emailet);

        pass = (EditText)findViewById(R.id.passwordet);

        ll = (LinearLayout) findViewById(R.id.register\_linear);

public class OtpPage extends AppCompatActivity {

    EditText otpverify;

    String url = "http://192.168.43.191/chatme/register.php"

    Database\_Handler db\_handler;

    @Override

    protected void onCreate(@Nullable Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_otp\_page);

public class ChangePassword extends AppCompatActivity {

    EditText pass1,pass2;

    String url = "http://192.168.43.191/chatme/changepass.php";    String pass;

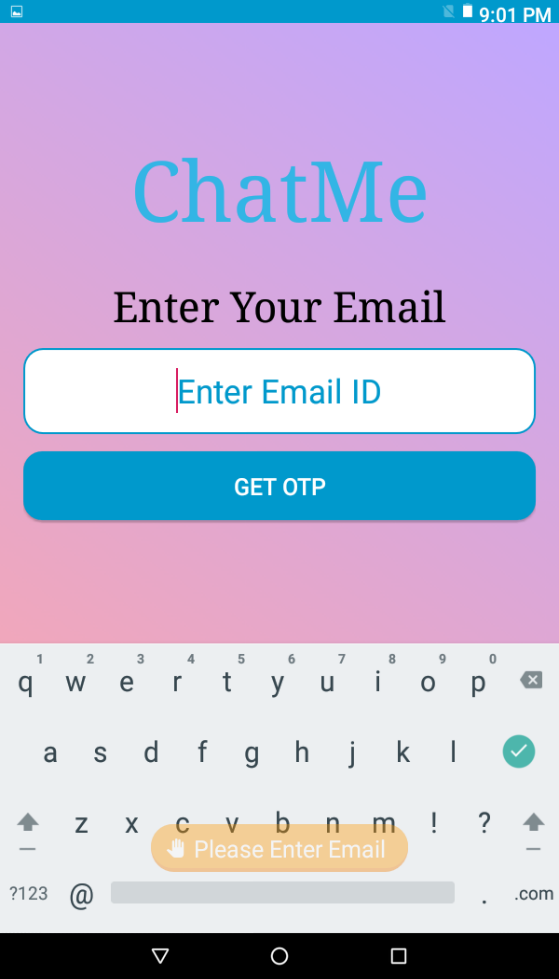
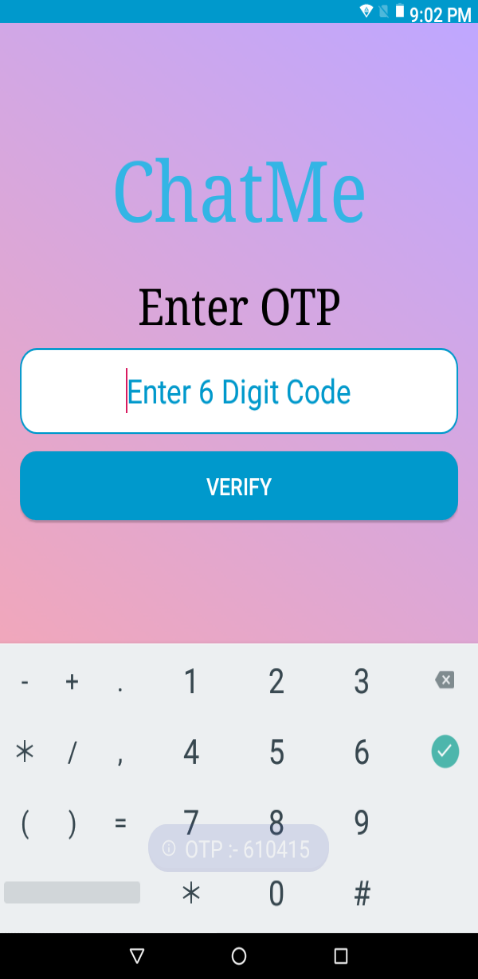
    @Override

    protected void onCreate(@Nullable Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_change\_password);

**ForgotPassword And Verification**

public class ForgotPassword extends AppCompatActivity {

    EditText email;

    String url = "http://192.168.43.191/chatme/check.php";

    @Override

    protected void onCreate(@Nullable Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_forgot\_password);

public class OtpPage extends AppCompatActivity {

    EditText otpverify;

    String url = "http://192.168.43.191/chatme/register.php";

    Database\_Handler db\_handler;

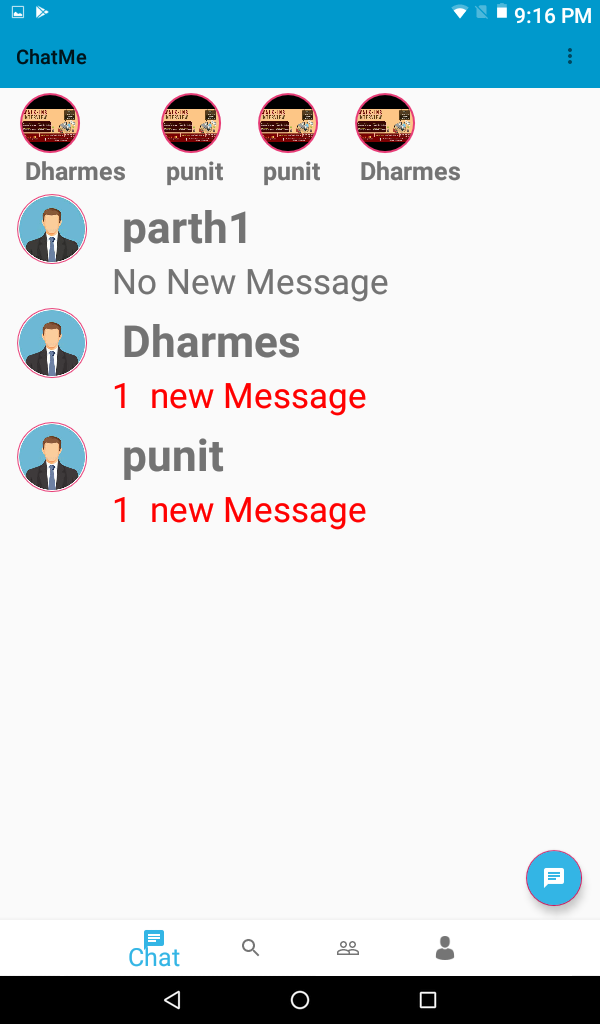
    @Override

    protected void onCreate(@Nullable Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_otp\_page);

**MainActivity**



public class Main extends AppCompatActivity {

    FloatingActionButton addbtn;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        Toolbar toolbar=findViewById(R.id.top\_toolbar);

        setSupportActionBar(toolbar);

private BottomNavigationView.OnNavigationItemSelectedListener navListener = new BottomNavigationView.OnNavigationItemSelectedListener() {

        @Override

        public boolean onNavigationItemSelected(@NonNull MenuItem menuItem) {

            Fragment selectFragment = null;

            switch (menuItem.getItemId()) {

                case R.id.nav\_home:

                    selectFragment = new Fragment\_Home();

                    break;

                case R.id.nav\_search:

                    selectFragment = new Fragment\_Search();

                    break;

                case R.id.nav\_profile:

                    selectFragment = new Fragment\_Profile();

                    break;

                case R.id.nav\_request:

                    selectFragment = new Fragment\_Request();

                    break;

            }

            getSupportFragmentManager().beginTransaction().replace(R.id.frame, selectFragment).commit();

            return true;

        }

    };

    @Override

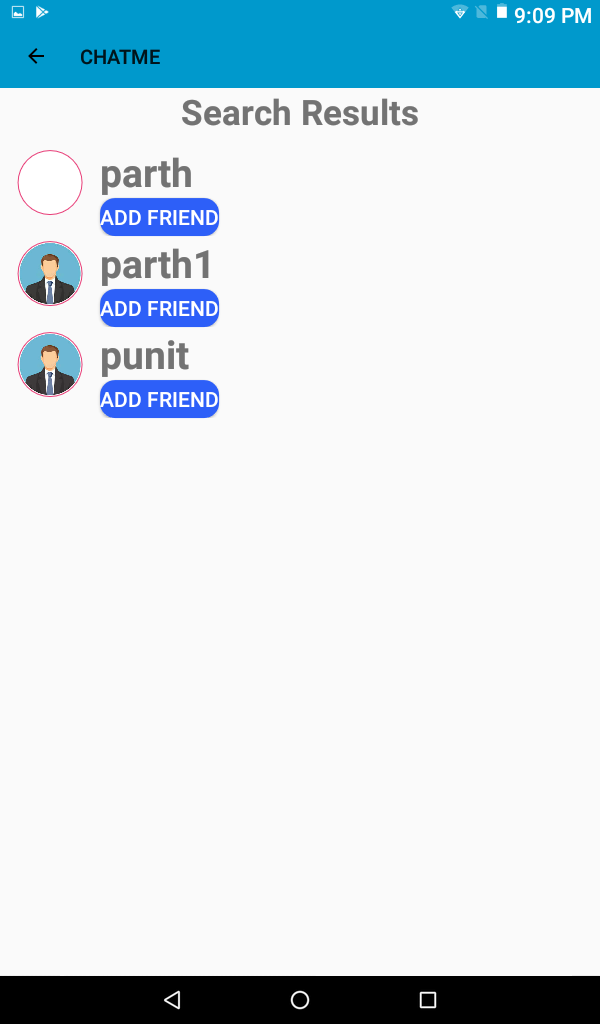
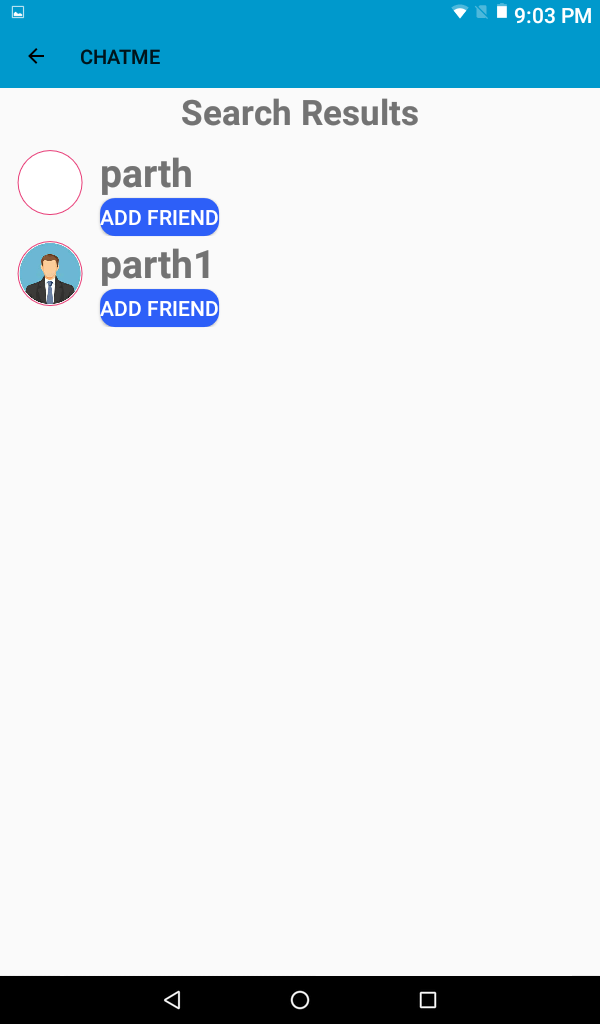
    public boolean onCreateOptionsMenu(Menu menu) {

        getMenuInflater().inflate(R.menu.top\_nav,menu);

        return true;

    }

**Search Friend**

** **

public class search\_result extends AppCompatActivity {

    ArrayList<String> f\_NAME = new ArrayList<String>();

    ArrayList<String> f\_PICTURE = new ArrayList<String>();

    ArrayList<String> f\_ID = new ArrayList<String>();

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.search\_result);

        Toolbar titleBar=(Toolbar)findViewById(R.id.search\_result\_toolbar);

        setSupportActionBar(titleBar);

        titleBar.setNavigationIcon(R.drawable.ic\_arrow\_back\_black\_24dp);

        titleBar.setNavigationOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                finish();

            }

        });

        //getting intent from Fragment\_search

        Intent i=getIntent();

        String response=i.getStringExtra("response");

        load\_friend(response);

    }

    private void load\_friend(String response){

        try{

            JSONArray jarray = new JSONArray(response);

            for(int n=0;n<jarray.length();n++)

            {

                JSONObject jobject = jarray.getJSONObject(n);

                f\_ID.add(jobject.getString("ID"));

                f\_NAME.add(jobject.getString("NAME"));

                f\_PICTURE.add(jobject.getString("PICTURE"));

            }

        }

        catch(JSONException e){}

        showlist();

    }

    private void showlist(){

        if(f\_ID.size()!=0) {

            RecyclerView recycle\_Friend = findViewById(R.id.search\_result\_recycleview);

            recycle\_Friend.setLayoutManager(new LinearLayoutManager(getApplicationContext(), LinearLayoutManager.VERTICAL, false));

            recycle\_Friend.setAdapter(new FriendAdapter(getApplicationContext(),f\_ID, f\_NAME, f\_PICTURE));

        }

        else{

            TextView tv=(TextView)findViewById(R.id.searchresult);

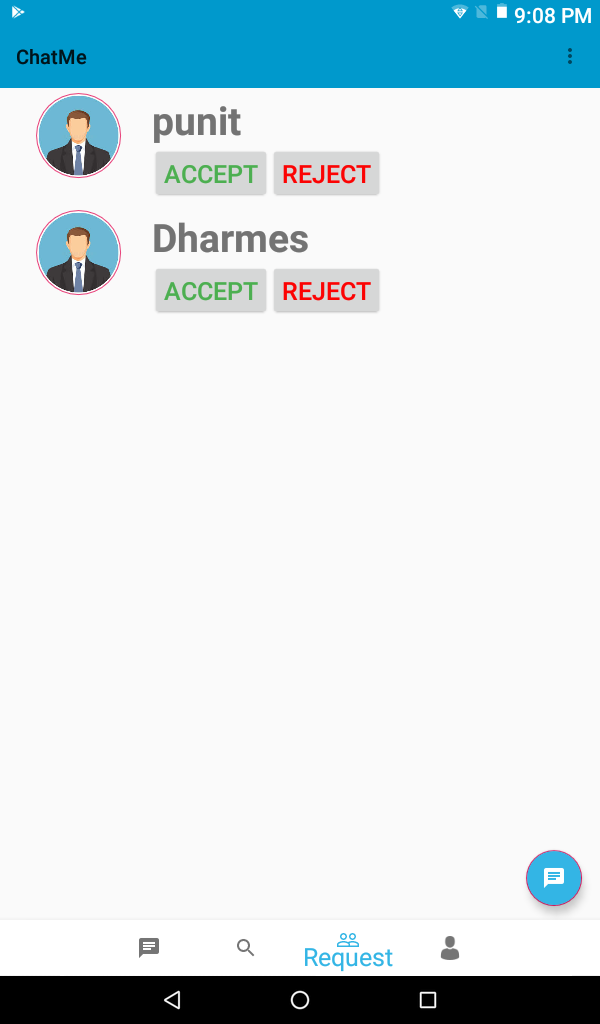
            tv.setText("No Result Found");

        }

    }

}

**Request**

****

public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {

        view = inflater.inflate(R.layout.fragment\_request, container, false);

        load\_request();

        return view;

    }

    private void load\_request(){

        String url = "http://192.168.43.191/chatme/friend\_requests.php";

        StringRequest MyStringRequest = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {

            @Override

            public void onResponse(String response) {

                fill\_request(response);

            }

        }, new Response.ErrorListener() { //Create an error listener to handle errors appropriately.

            @Override

            public void onErrorResponse(VolleyError error) {

                Toast.makeText(getContext(),error.toString(),4).show();

            }

        }) {

            protected Map<String, String> getParams() {

                Map<String, String> MyData = new HashMap<String, String>();

                MyData.put("ID",My\_Detail.My\_ID);

                return MyData;

            }

        };

        MySingleton.getInstance(getContext()).addRequestQueue(MyStringRequest);

    }

public class RequestAdapter extends RecyclerView.Adapter<RequestAdapter.RequestHolder> {

    private ArrayList<String> Id;

    private ArrayList<String> Name;

    private ArrayList<String> Picture;

    private ArrayList<String> Status;

    private ArrayList<String> Last\_Seen;

    private Fragment\_Request  fragment;

    public RequestAdapter(Fragment\_Request  frag,ArrayList<String> id, ArrayList<String> name, ArrayList<String> picture, ArrayList<String> status, ArrayList<String> last\_Seen) {

        Id = id;

        Name = name;

        Picture = picture;

        Status = status;

        Last\_Seen = last\_Seen;

        fragment=frag;

    }

    @NonNull

    @Override

    public RequestHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

        LayoutInflater LV=LayoutInflater.from(parent.getContext());

        View view=LV.inflate(R.layout.request\_list,parent,false);

        return  new RequestHolder(view);

    }

    @Override

    public void onBindViewHolder(@NonNull RequestHolder holder, int position) {

        Picasso.get().load("http://192.168.43.191/chatme/img/"+Picture.get(position)).into(holder.RequestImg);

        holder.RequestTxt.setText(Name.get(position));

    }

    @Override

    public int getItemCount() {

        return Name.size();

    }

    public void removeItem(int position) {

        Name.remove(position);

        notifyItemRemoved(position);

    }

    public class RequestHolder extends RecyclerView.ViewHolder{

        public RequestHolder(@NonNull View itemView) {

            super(itemView);

            RequestImg = itemView.findViewById(R.id.request\_img);

            RequestTxt = itemView.findViewById(R.id.request\_name);

            Request\_accept = itemView.findViewById(R.id.request\_accept\_btn);

            Request\_reject = itemView.findViewById(R.id.request\_reject\_btn);

**MyProfile**

****

public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup container, Bundle savedInstanceState) {

        view=inflater.inflate(R.layout.fragment\_profile, container, false);

        profileImg=(ImageView)view.findViewById(R.id.my\_profile\_img);

        name=(TextView)view.findViewById(R.id.profile\_name);

        status=(TextView)view.findViewById(R.id.profile\_status);

        Picasso.get().load("http://192.168.43.191/chatme/img/"+My\_Detail.My\_PICTURE).into(profileImg);

        name.setText(My\_Detail.My\_NAME);

        status.setText(My\_Detail.My\_STATUS);

private void updateProfile(final String img){

        String url = "http://192.168.43.191/chatme/updateprofile.php";

        StringRequest MyStringRequest = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {

            @Override

            public void onResponse(String response) {

                  updateMy\_Detail(response);

                }

        }, new Response.ErrorListener() { //Create an error listener to handle errors appropriately.

            @Override

            public void onErrorResponse(VolleyError error) {

                Toast.makeText(getContext(),error.toString(),1).show();

            }

        }) {

            protected Map<String, String> getParams() {

                Map<String, String> MyData = new HashMap<String, String>();

                MyData.put("file",img);

                MyData.put("ID",My\_Detail.My\_ID);

                MyData.put("NAME",name.getText().toString());

                MyData.put("STATUS",status.getText().toString());

                return MyData;

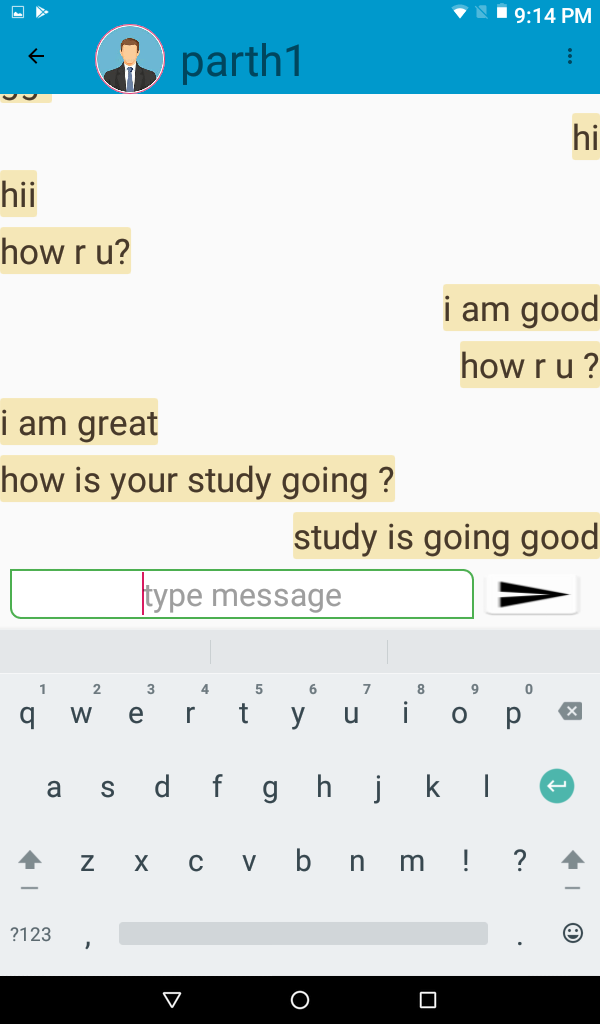
            }

        };

        MySingleton.getInstance(getContext()).addRequestQueue(MyStringRequest);

    }

**CHATSCREEN**

**** protected void onCreate(@Nullable Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.chat\_screen);

        Toolbar titleBar = (Toolbar) findViewById(R.id.chat\_title\_toolbar);

        setSupportActionBar(titleBar);

        getSupportActionBar().setDisplayShowTitleEnabled(false);

        load\_chat();

private void load\_chat(){

        String url = "http://192.168.43.191/chatme/showchat.php";

        StringRequest MyStringRequest = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {

            @Override

            public void onResponse(String response) {

                fill\_chat(response);

            }

        }, new Response.ErrorListener() { //Create an error listener to handle errors appropriately.

            @Override

            public void onErrorResponse(VolleyError error) {

                Toast.makeText(getApplicationContext(),error.toString(),1).show();

            }

        }) {

            protected Map<String, String> getParams() {

                Map<String, String> MyData = new HashMap<String, String>();

                MyData.put("ID",My\_Detail.My\_ID);

                MyData.put("FID",in.getStringExtra("id"));

                return MyData;

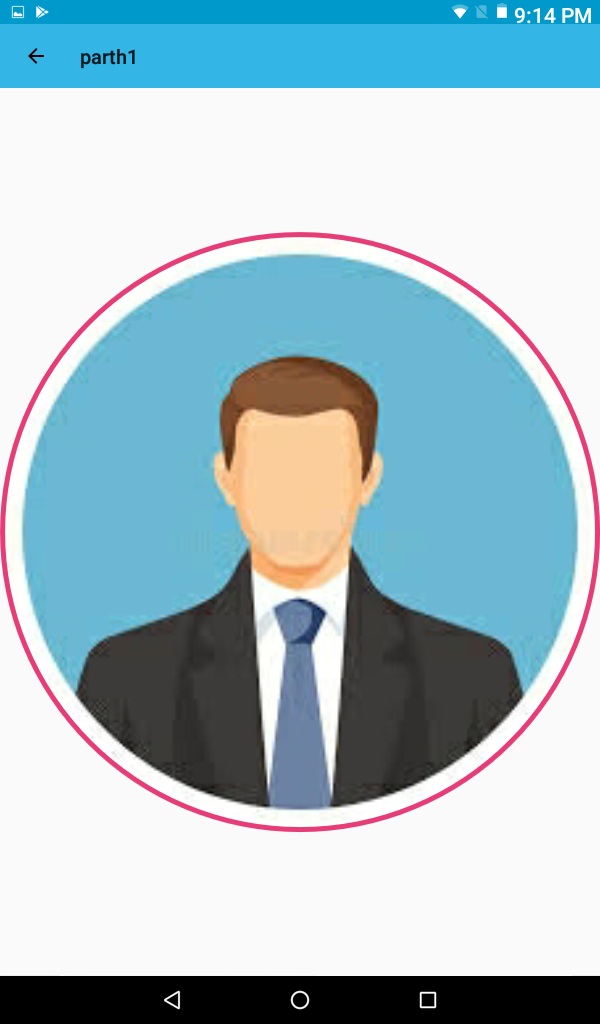
            }

        };

        MySingleton.getInstance(getApplicationContext()).addRequestQueue(MyStringRequest);

    }

**ViewProfile**



protected void onCreate(@Nullable Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.view\_img);

        Toolbar titleBar=(Toolbar)findViewById(R.id.view\_img\_toolbar);

        ImageView img=(ImageView)findViewById(R.id.profile\_detail\_img);

        setSupportActionBar(titleBar);

        Intent in=getIntent();

        Picasso.get().load("http://192.168.43.191/chatme/img/"+in.getStringExtra("path")).into(img);

        titleBar.setTitle(in.getStringExtra("name"));

        titleBar.setNavigationIcon(R.drawable.ic\_arrow\_back\_black\_24dp);

        titleBar.setNavigationOnClickListener(new View.OnClickListener() {

            @Override

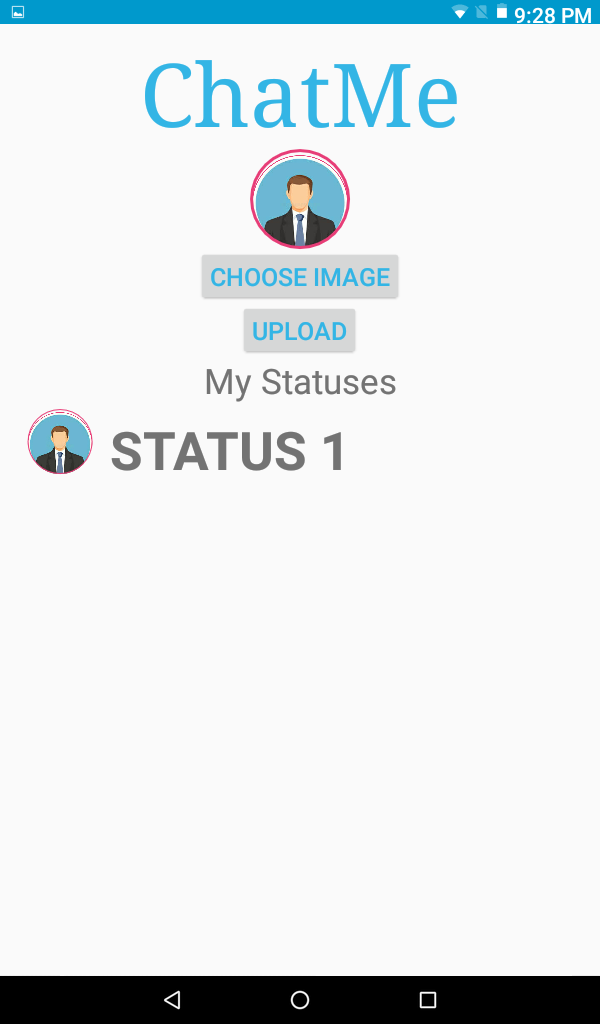
            public void onClick(View v) {

                finish();

            }

        });

**Upload Status**

 protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.upload\_status);

private void addStatus(final String img){

        String url = "http://192.168.43.191/chatme/addstatus.php";

        StringRequest MyStringRequest = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {

            @Override

            public void onResponse(String response) {

                Toast.makeText(getApplicationContext(),"Added",1).show();

                Intent i=new Intent(getApplicationContext(),upload\_status.class);

                startActivity(i);

                finish();

            }

        }, new Response.ErrorListener() {

            @Override

            public void onErrorResponse(VolleyError error) {

                Toast.makeText(getApplicationContext(),error.toString(),1).show();

            }

        }) {

            protected Map<String, String> getParams() {

                Map<String, String> MyData = new HashMap<String, String>();

                MyData.put("file",img);

                MyData.put("ID",My\_Detail.My\_ID);

                return MyData;

            }

        };

        MySingleton.getInstance(getApplicationContext()).addRequestQueue(MyStringRequest);

    }

    private void load\_status(){

        String url = "http://192.168.43.191/chatme/mystatus.php";        StringRequest MyStringRequest = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {            @Override

            public void onResponse(String response) {

                fill\_status(response);            }

        }, new Response.ErrorListener() {

            @Override

            public void onErrorResponse(VolleyError error) {

                Toast.makeText(getApplicationContext(),error.toString(),1).show();

            }

        }) {

            protected Map<String, String> getParams() {

                Map<String, String> MyData = new HashMap<String, String>();

                MyData.put("ID",My\_Detail.My\_ID);

                return MyData;

            }

        };

        MySingleton.getInstance(getApplicationContext()).addRequestQueue(MyStringRequest);

    }

**DatabaseHandler**

public class Database\_Handler extends SQLiteOpenHelper {

    private static final int DATABASE\_VERSION = 1;

    private static final String DATABASE\_NAME = "chat\_me";

    private static final String TABLE = "user\_details";

    private static final String USER\_ID="ID";

    private static final String USER\_NAME = "NAME";

    private static final String USER\_EMAIL = "EMAIL";

    private static final String USER\_PASS = "PASSWORD";

    private static final String USER\_PICTURE = "PICTURE";

    private static final String USER\_STATUS = "STATUS";

    SQLiteDatabase sldb;

    Context cn;

    public Database\_Handler(Context ctx){

        super(ctx, DATABASE\_NAME, null, DATABASE\_VERSION);

        cn = ctx;

    }

    @Override

    public void onCreate(SQLiteDatabase db) {

        String CREATE\_TABLE = "CREATE TABLE " + TABLE+ "("

                + USER\_ID + " TEXT ," + USER\_NAME + " TEXT,"

                + USER\_EMAIL + " TEXT,"+ USER\_PASS + " TEXT,"+ USER\_PICTURE

                + " TEXT,"+ USER\_STATUS + " TEXT)";

        db.execSQL(CREATE\_TABLE);

    }

    @Override

    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

        db.execSQL("DROP TABLE IF EXISTS " + TABLE);

        // Create tables again

        onCreate(db);

    }

    public void addUser(String id,String u\_name,String u\_email,String u\_pass){

        sldb = this.getWritableDatabase();

        ContentValues values = new ContentValues();

        values.put(USER\_ID,id);

        values.put(USER\_NAME,u\_name);

        values.put(USER\_EMAIL,u\_email);

        values.put(USER\_PASS,u\_pass);

        values.put(USER\_PICTURE,"demo.jpeg");

        values.put(USER\_STATUS,"Hey..I M Using ChatME");

        // Inserting user details

        sldb.insert(TABLE, null, values);

        sldb.close(); // Closing database connection

    }    public void deleteuser(){

        sldb = this.getWritableDatabase();

        sldb.delete(TABLE,null,null);

        sldb.close();

    }

    public void addusr(String id,String name, String email, String password, String picture, String status) {

        sldb = this.getWritableDatabase();

        ContentValues value = new ContentValues();

        value.put(USER\_ID,id);

        value.put(USER\_NAME,name);

        value.put(USER\_EMAIL,email);

        value.put(USER\_PASS,password);

        value.put(USER\_PICTURE,picture);

        value.put(USER\_STATUS,status);

        sldb.insert(TABLE,null,value);

        sldb.close();

    }

    public boolean isUser(){

        sldb=this.getReadableDatabase();

        String[] columns=new String[]{USER\_ID,USER\_NAME,USER\_EMAIL,USER\_PASS,USER\_PICTURE,USER\_STATUS};

        Cursor c=sldb.query(TABLE,columns,null,null,null,null,null);

        if(c.moveToNext()){

            return true;

        }

        return false;

    }

    public String[] getMailPass(){

        sldb = this.getReadableDatabase();

        String[] columns = new String[]{USER\_ID,USER\_NAME,USER\_EMAIL,USER\_PASS,USER\_PICTURE,USER\_STATUS};

        Cursor c = sldb.query(TABLE,columns,null,null,null,null,null);

        c.moveToFirst();

        String[] res = new String[]{c.getString(c.getColumnIndex(USER\_EMAIL)),c.getString(c.getColumnIndex(USER\_PASS))};

        return res;

    }

}

**RecyclerAdapter**

public RequestHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

        LayoutInflater LV=LayoutInflater.from(parent.getContext());

        View view=LV.inflate(R.layout.request\_list,parent,false);

        return  new RequestHolder(view);

    }

    @Override

    public void onBindViewHolder(@NonNull RequestHolder holder, int position) {

        Picasso.get().load("http://192.168.43.191/chatme/img/"+Picture.get(position)).into(holder.RequestImg);

        holder.RequestTxt.setText(Name.get(position));

    }

    @Override

    public int getItemCount() {

        return Name.size();

    }

    public void removeItem(int position) {

        Name.remove(position);

        notifyItemRemoved(position);

    }

    public class RequestHolder extends RecyclerView.ViewHolder{

        ImageView RequestImg;

        TextView  RequestTxt;

        Button Request\_accept;

        Button Request\_reject;

        public RequestHolder(@NonNull View itemView) {

            super(itemView);

            RequestImg = itemView.findViewById(R.id.request\_img);

            RequestTxt = itemView.findViewById(R.id.request\_name);

            Request\_accept = itemView.findViewById(R.id.request\_accept\_btn);

            Request\_reject = itemView.findViewById(R.id.request\_reject\_btn);

            itemView.setOnClickListener(new View.OnClickListener() {

                @Override

                public void onClick(View v) {

                    Intent in=new Intent(v.getContext(),Profile\_Detail.class);

                    in.putExtra("name",Name.get(getAdapterPosition()));

                    in.putExtra("path",Picture.get(getAdapterPosition()));

                    in.putExtra("status",Status.get(getAdapterPosition()));

                    in.putExtra("last\_seen",Last\_Seen.get(getAdapterPosition()));

                    v.getContext().startActivity(in);

                }

            });

            Request\_accept.setOnClickListener(new View.OnClickListener() {

                @Override

                public void onClick(View v) {

                    fragment.request\_answer(Id.get(getAdapterPosition()),"y");

                    removeItem(getAdapterPosition());

                    Toast.makeText(v.getContext(), "Accepted", 1).show();                }

            });

            Request\_reject.setOnClickListener(new View.OnClickListener() {

                @Override

                public void onClick(View v) {

                    fragment.request\_answer(Id.get(getAdapterPosition()),"n");

                    removeItem(getAdapterPosition());

                    Toast.makeText(v.getContext(),"Rejected",1).show();

                }

            });

        }

    }

}  
 **Thank You☺**