## Name: Lakshmana Kumar Mettu

## Class ID: 11

## Introduction:

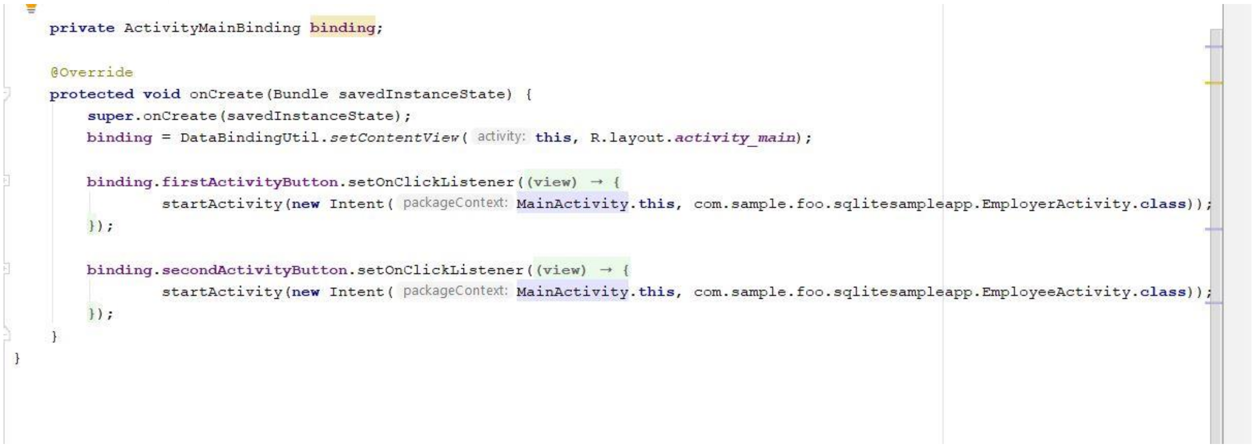
- This ICP is to know about fire-base connections and sample database connections and also adding features for the real time database in android applications.

## Programming elements:

- Android Studio
- Firebase Database
- SQLite Database

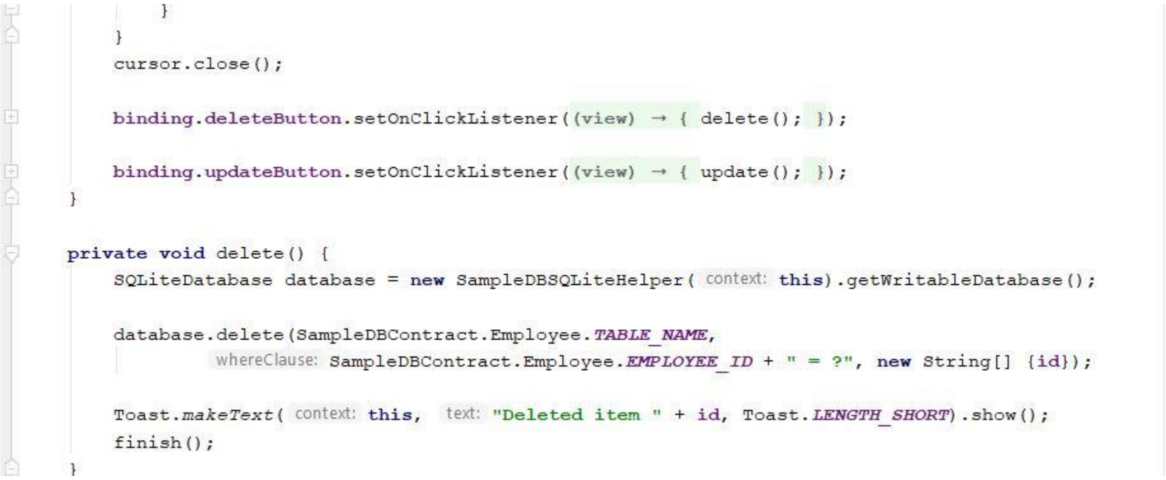
## Part-1 Objective:

- The main objective of part-1 is to implementation of update and delete functionalities for employee and employer given in source code.
- For this task I am using SQLite database to store the details of both employee and employer before updating and after updating.
- The corresponding code snippets and output screen shots for Part-1 is given below.
- Both has search and save buttons.
- The main activity which connects to database shown below.

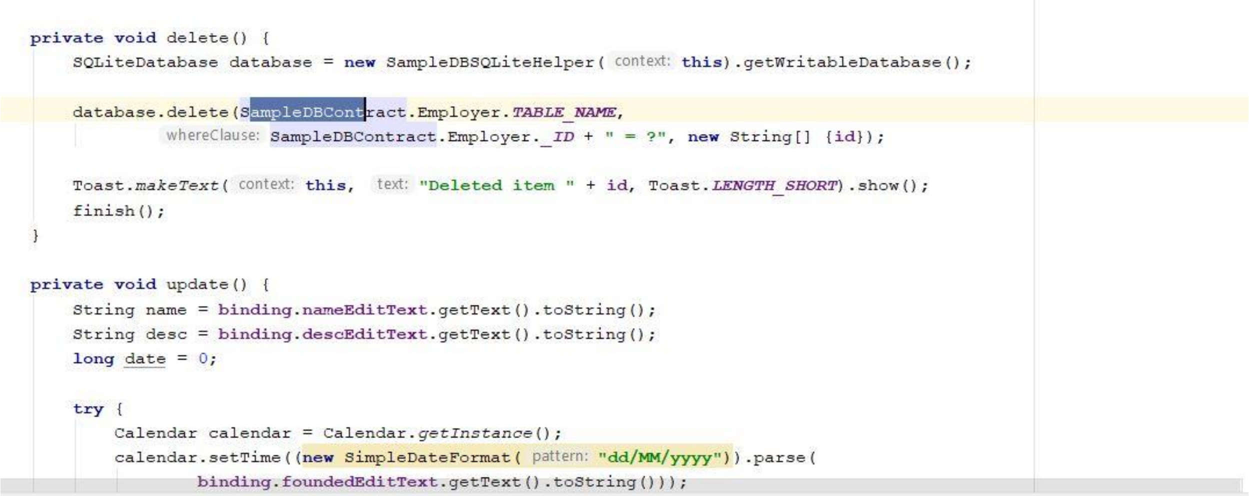


```
    private ActivityMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = DataBindingUtil.setContentView( activity: this, R.layout.activity_main);

        binding.firstActivityButton.setOnClickListener((view) → {
                startActivity(new Intent( packageContext: MainActivity.this, com.sample.foo.sqlitesampleapp.EmployerActivity.class));
        });

        binding.secondActivityButton.setOnClickListener((view) → {
                startActivity(new Intent( packageContext: MainActivity.this, com.sample.foo.sqlitesampleapp.EmployeeActivity.class));
        });
    }
}
```

- Implementation of delete and update functionalities in employee and employer are shown below.

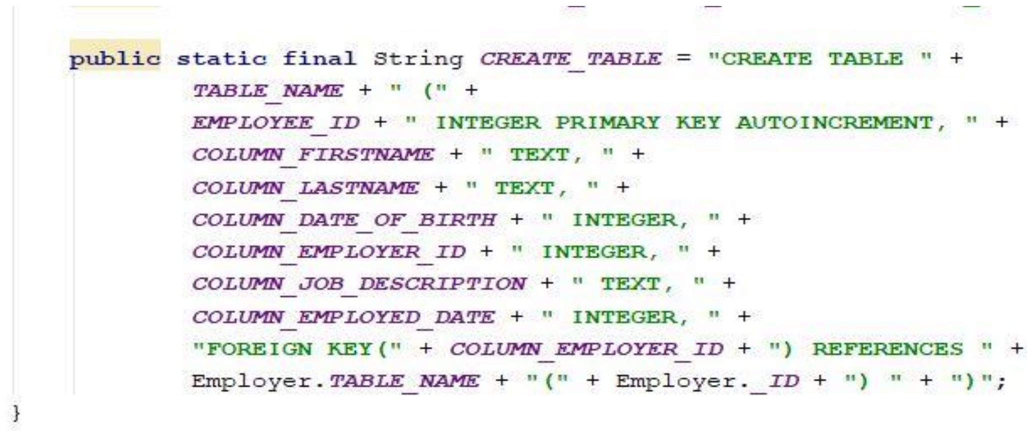


```
            }
        }
        cursor.close();

        binding.deleteButton.setOnClickListener((view) → { delete(); });

        binding.updateButton.setOnClickListener((view) → { update(); });
    }

    private void delete() {
        SQLiteDatabase database = new SampleDBSQLiteHelper( context: this).getWritableDatabase();

        database.delete(SampleDBContract.Employee.TABLE_NAME,
                whereClause: SampleDBContract.Employee.EMPLOYEE_ID + " = ?", new String[] {id});

        Toast.makeText( context: this,  text: "Deleted item " + id, Toast.LENGTH_SHORT).show();
        finish();
    }
```



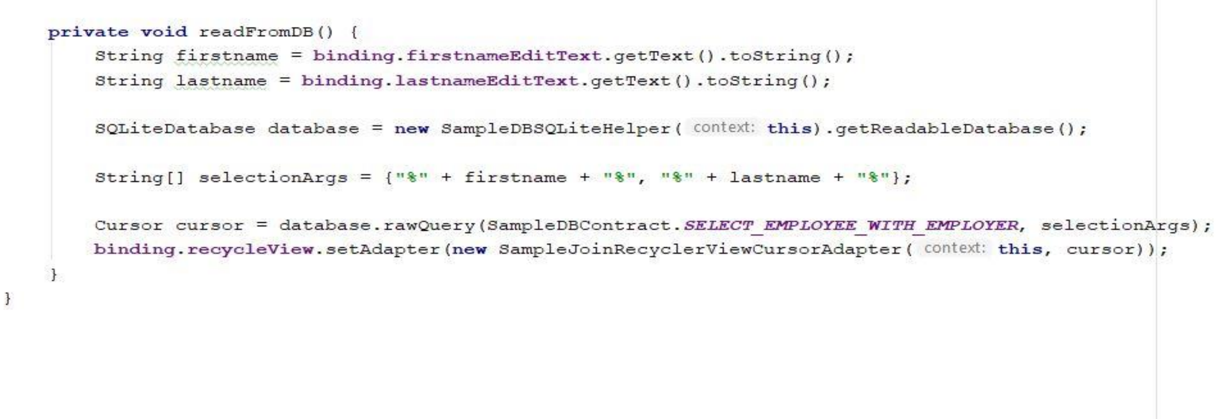
```
private void delete() {
    SQLiteDatabase database = new SampleDBSQLiteHelper( context: this).getWritableDatabase();

    database.delete(SampleDBContract.Employer.TABLE_NAME,
            whereClause: SampleDBContract.Employer._ID + " = ?", new String[] {id});

    Toast.makeText( context: this,  text: "Deleted item " + id, Toast.LENGTH_SHORT).show();
    finish();
}

private void update() {
    String name = binding.nameEditText.getText().toString();
    String desc = binding.descEditText.getText().toString();
    long date = 0;

    try {
        Calendar calendar = Calendar.getInstance();
        calendar.setTime((new SimpleDateFormat( pattern: "dd/MM/yyyy")).parse(
                binding.foundedEditText.getText().toString()));
```

- And Database storing details code snippet given below.



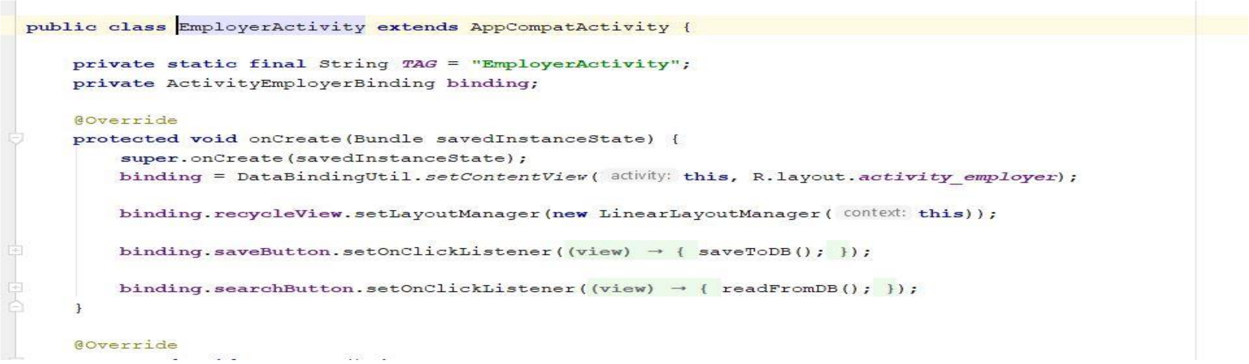
```java
    public static final String CREATE_TABLE = "CREATE TABLE " +
            TABLE_NAME + " (" +
            EMPLOYEE_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
            COLUMN_FIRSTNAME + " TEXT, " +
            COLUMN_LASTNAME + " TEXT, " +
            COLUMN_DATE_OF_BIRTH + " INTEGER, " +
            COLUMN_EMPLOYER_ID + " INTEGER, " +
            COLUMN_JOB_DESCRIPTION + " TEXT, " +
            COLUMN_EMPLOYED_DATE + " INTEGER, " +
            "FOREIGN KEY(" + COLUMN_EMPLOYER_ID + ") REFERENCES " +
            Employer.TABLE_NAME + "(" + Employer._ID + ") " + ")";
}
```

- Reading details from database has given below.



```java
    private void readFromDB() {
        String firstname = binding.firstnameEditText.getText().toString();
        String lastname = binding.lastnameEditText.getText().toString();

        SQLiteDatabase database = new SampleDBSQLiteHelper( context: this).getReadableDatabase();

        String[] selectionArgs = {"%" + firstname + "%", "%" + lastname + "%"};

        Cursor cursor = database.rawQuery(SampleDBContract.SELECT_EMPLOYEE_WITH_EMPLOYER, selectionArgs);
        binding.recycleView.setAdapter(new SampleJoinRecyclerViewCursorAdapter( context: this, cursor));
    }
}
```

- Both are having relative layout corresponding code snippet is given below.

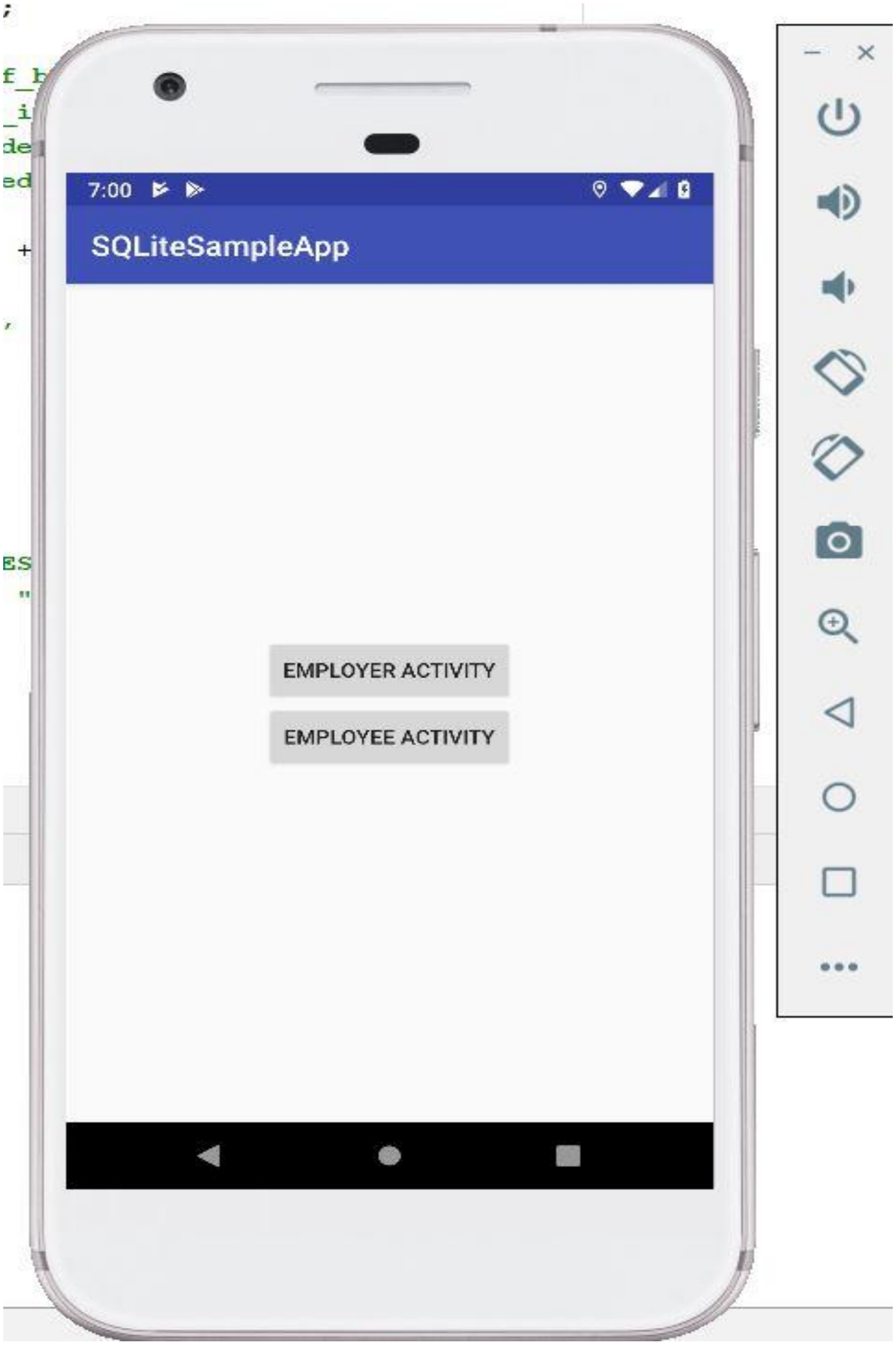


```java
public class EmployerActivity extends AppCompatActivity {

    private static final String TAG = "EmployerActivity";
    private ActivityEmployerBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = DataBindingUtil.setContentView( activity: this, R.layout.activity_employer);

        binding.recycleView.setLayoutManager(new LinearLayoutManager( context: this));

        binding.saveButton.setOnClickListener((view) → { saveToDB(); });

        binding.searchButton.setOnClickListener((view) → { readFromDB(); });
    }

    @Override
```

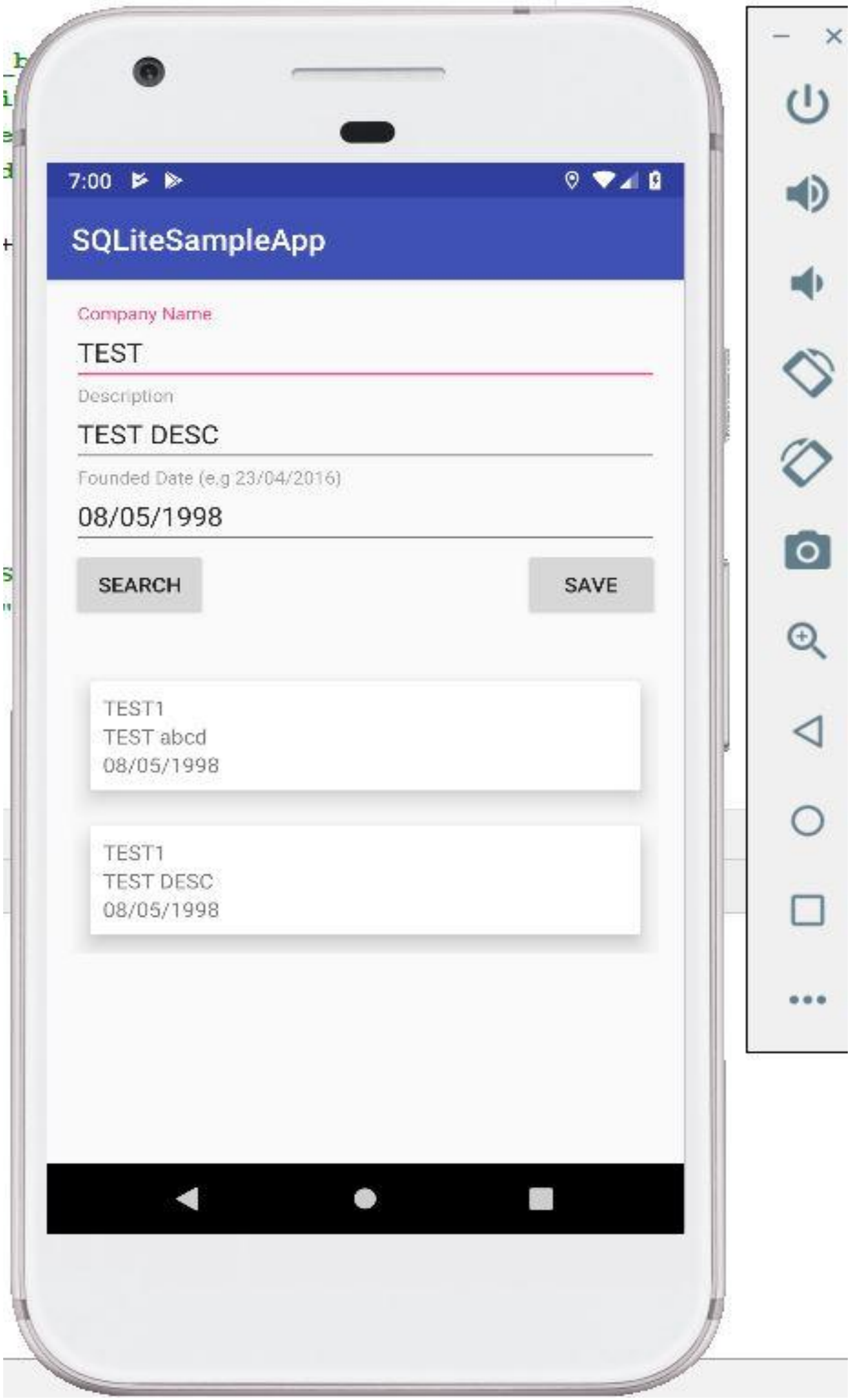
- Brief explanation about code and output given in video.

## Part-1 Output:

- Having both activity welcome page



- Employer Activity



7:00

SQLiteSampleApp

Company Name

TEST

Description

TEST DESC

Founded Date (e.g 23/04/2016)

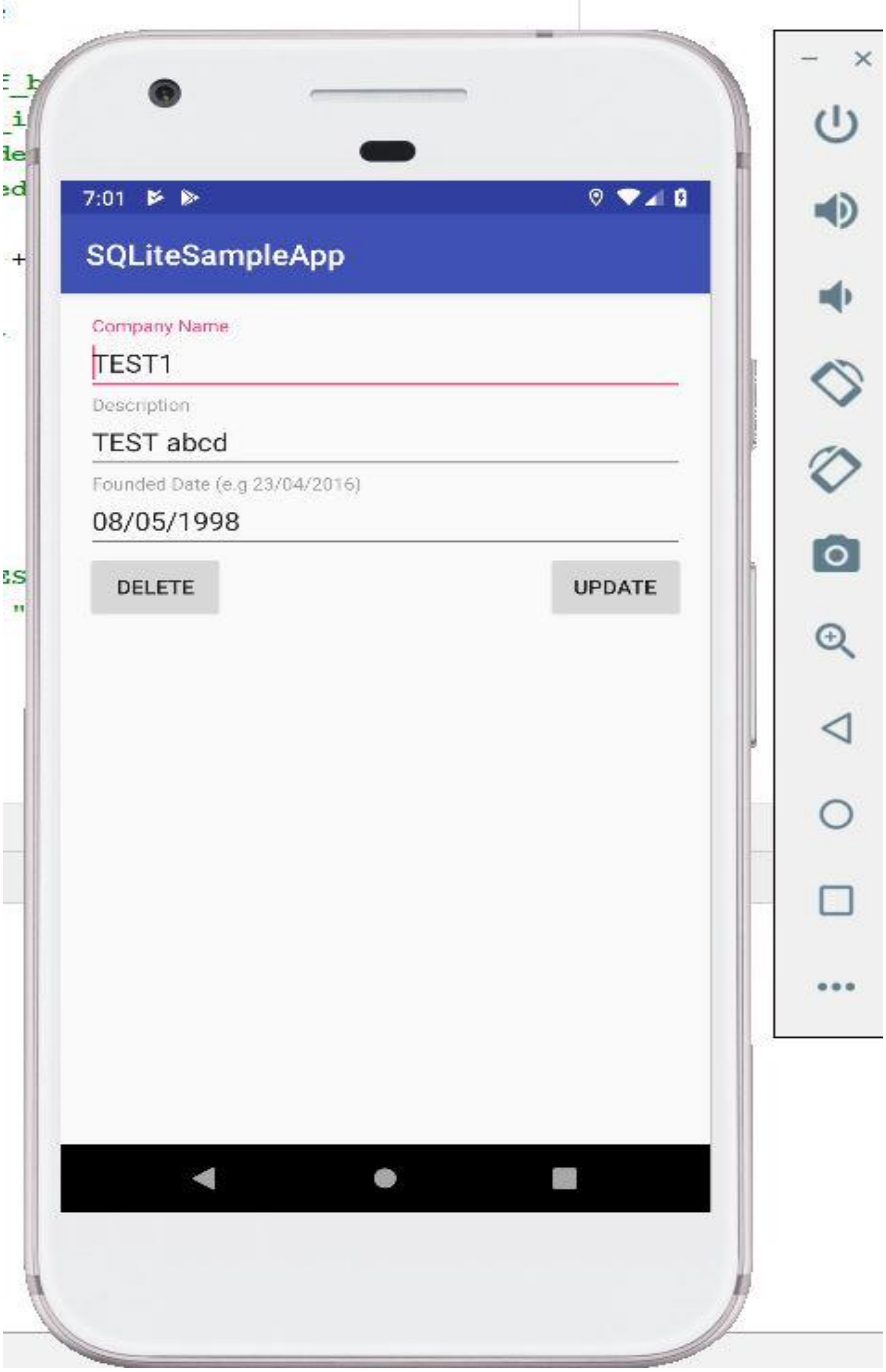
08/05/1998

SEARCH SAVE

TEST1
TEST abcd
08/05/1998

TEST1
TEST DESC
08/05/1998

- Employer activity having update and delete features for storing details.



7:01

SQLiteSampleApp

Company Name

TEST1

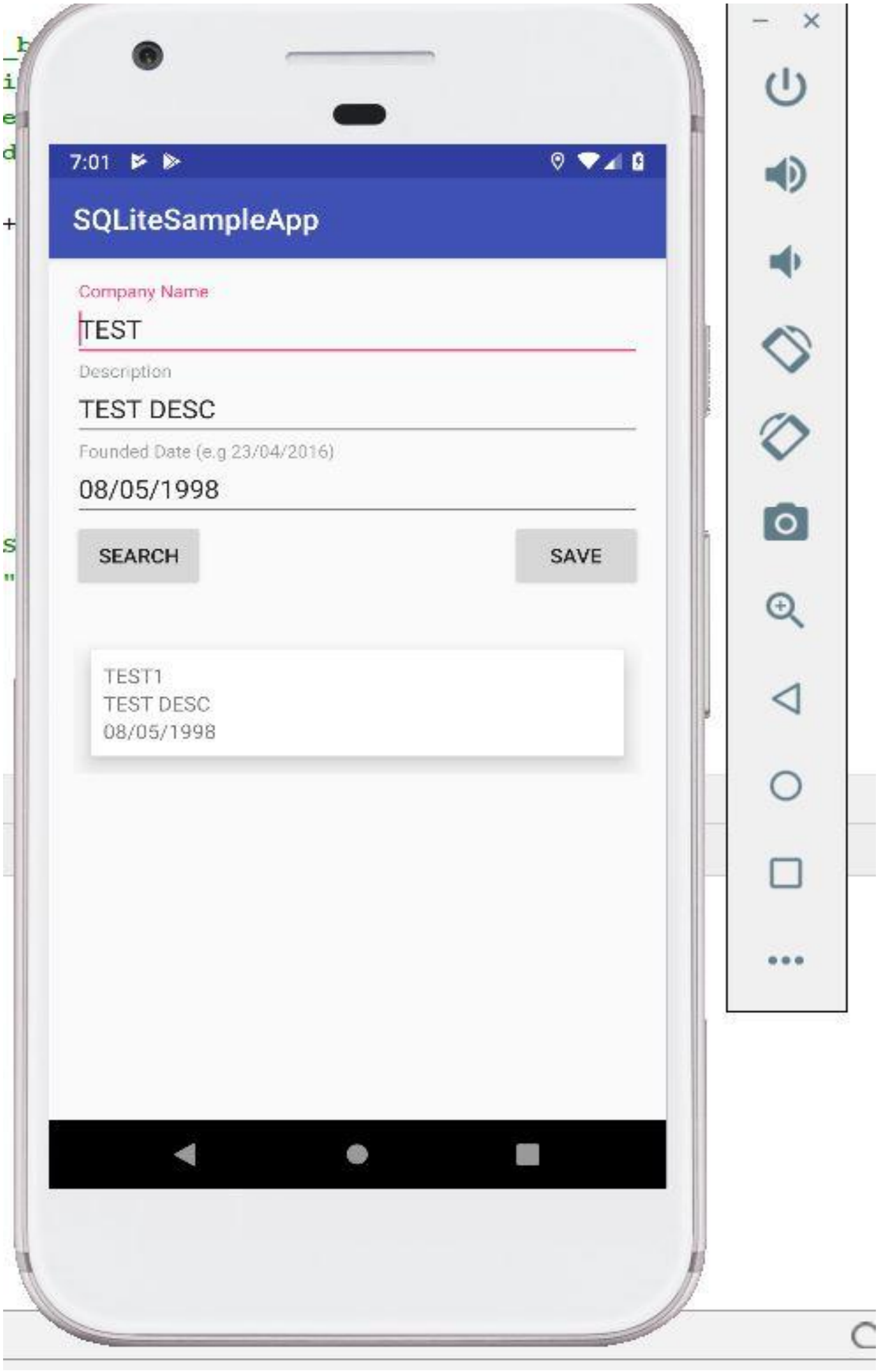
Description

TEST abcd

Founded Date (e.g 23/04/2016)

08/05/1998

- Updating feature example



7:01

SQLiteSampleApp

Company Name

TEST

Description

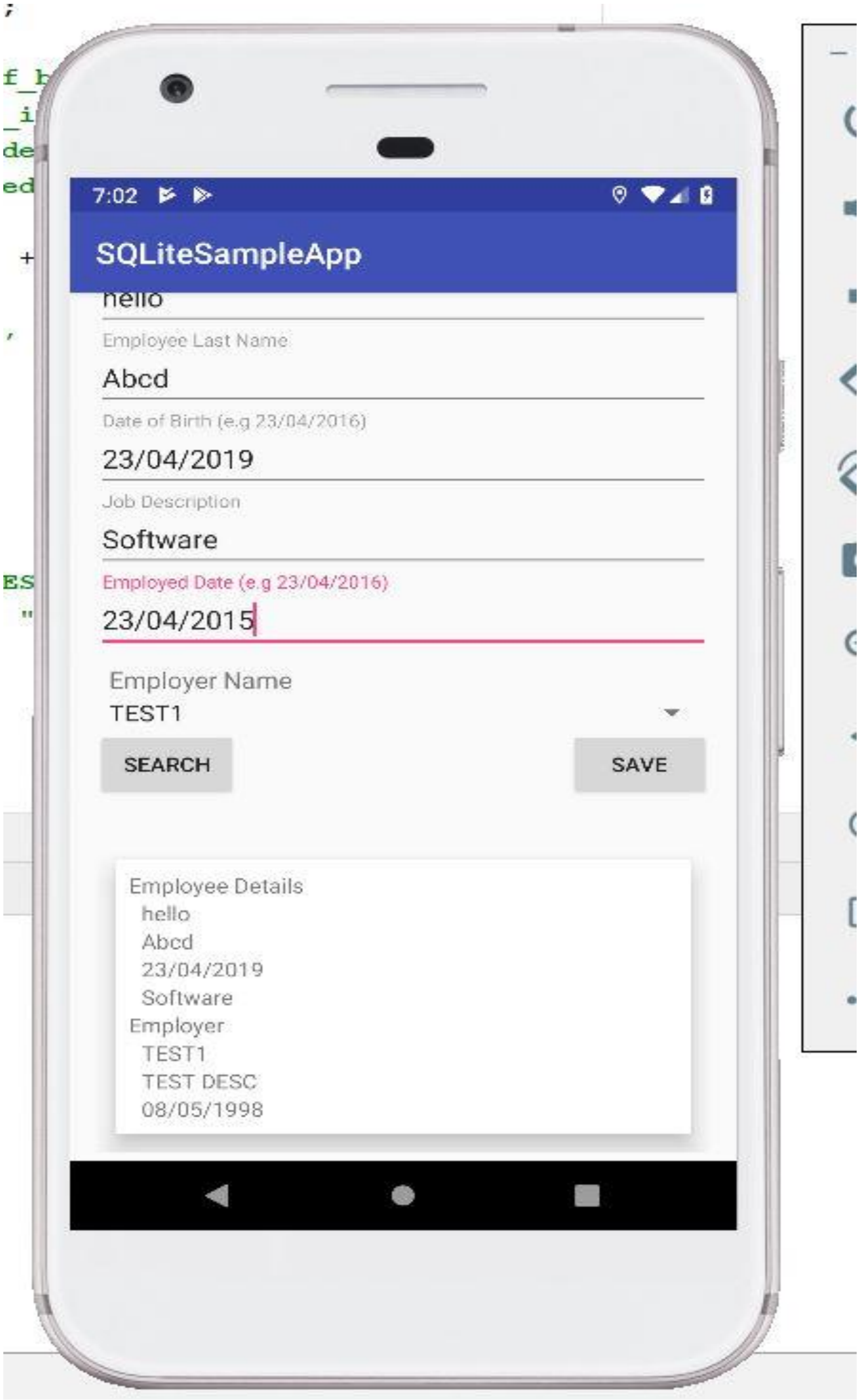
TEST DESC

Founded Date (e.g 23/04/2016)

08/05/1998

TEST1
TEST DESC
08/05/1998

- Employee activity storing details along with save and search options.



7:02

SQLiteSampleApp

hello

Employee Last Name
Abcd

Date of Birth (e.g 23/04/2016)
23/04/2019

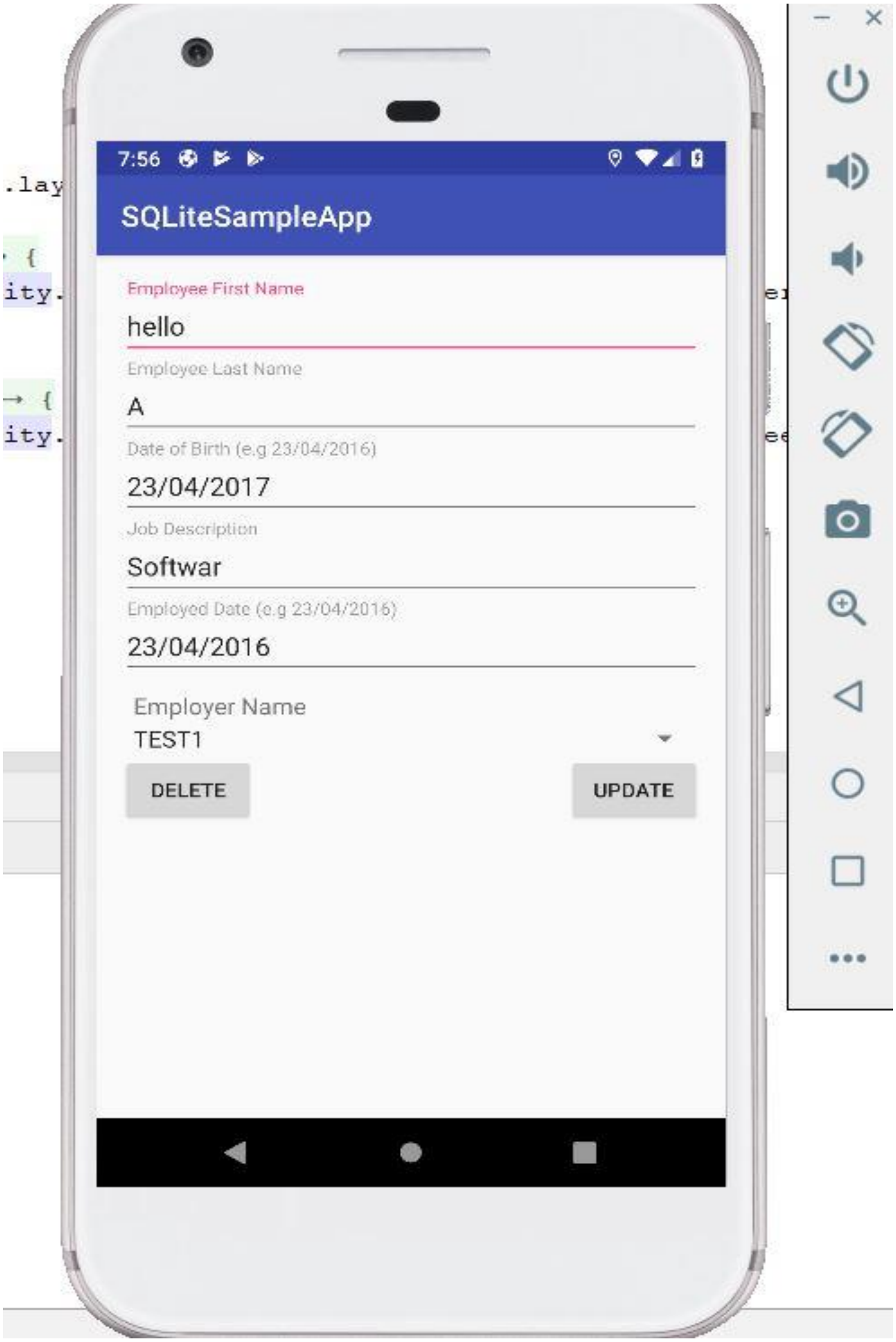
Job Description
Software

Employed Date (e.g 23/04/2016)
23/04/2015

Employer Name
TEST1

Employee Details
hello
Abcd
23/04/2019
Software
Employer
TEST1
TEST DESC
08/05/1998

- Employee also having delete and update features.



Employee First Name
hello

Employee Last Name
A

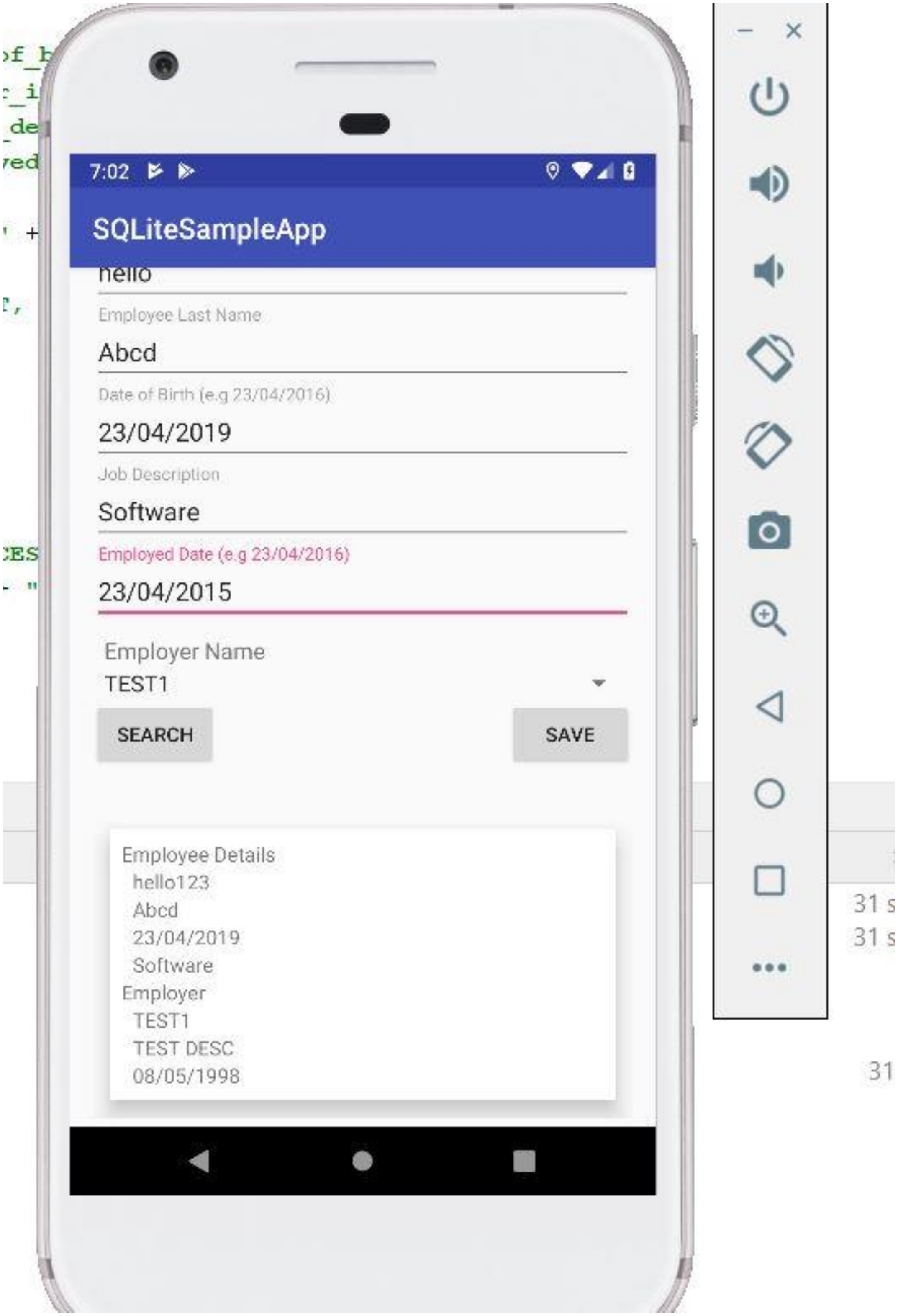
Date of Birth (e.g 23/04/2016)
23/04/2017

Job Description
Softwar

Employed Date (e.g 23/04/2016)
23/04/2016

Employer Name
TEST1

- Updated from hello to hello123 in first name and stored in database.



7:02

SQLiteSampleApp

hello

Employee Last Name

Abcd

Date of Birth (e.g 23/04/2016)

23/04/2019

Job Description

Software

Employed Date (e.g 23/04/2016)

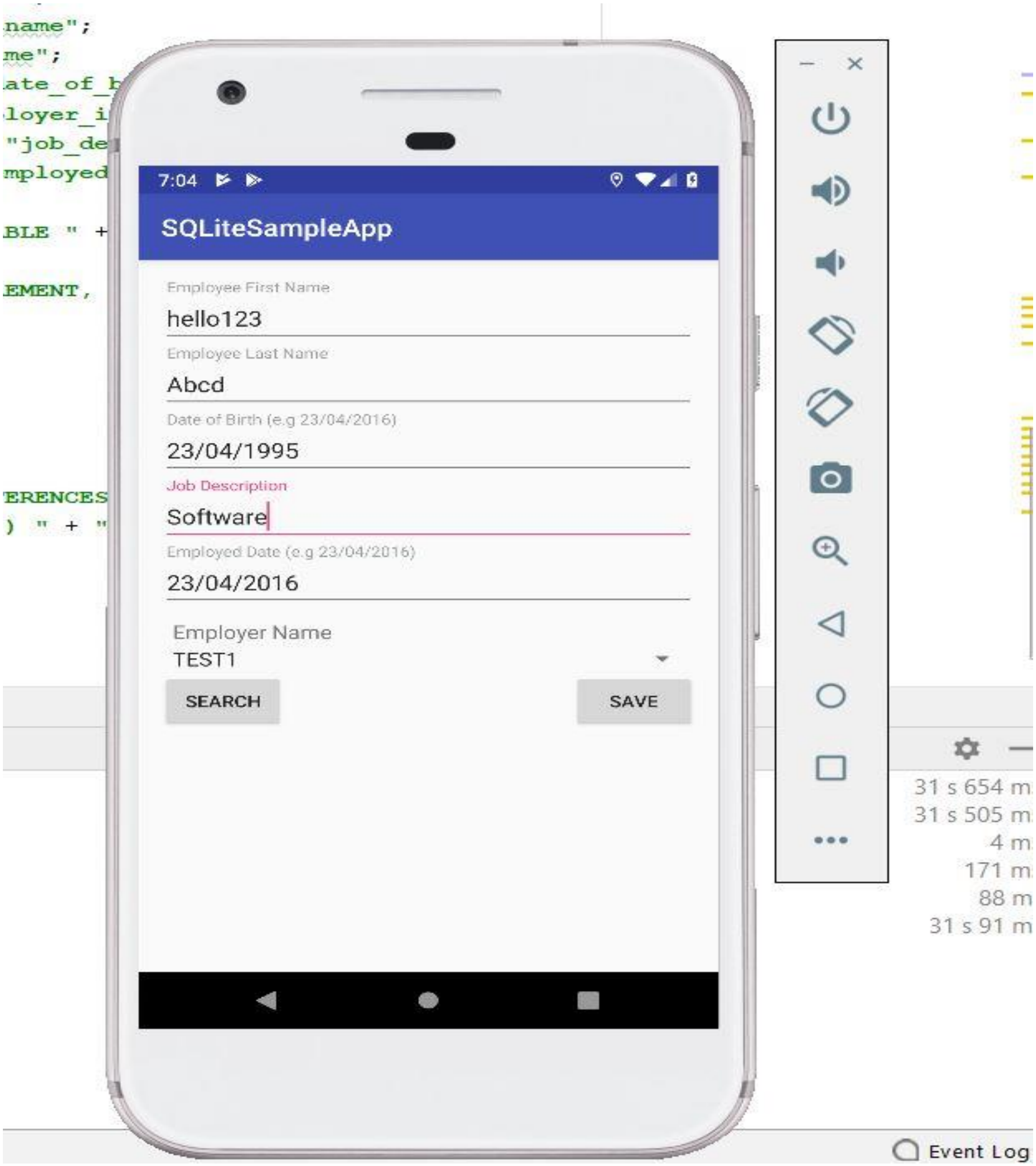
23/04/2015

Employer Name

TEST1

Employee Details
hello123
Abcd
23/04/2019
Software
Employer
TEST1
TEST DESC
08/05/1998

- Deleting the stored details from database.



```
name";
me";
ate_of_b
loyer_i
"job_de
mployed

BLE " +

EMENT,

ERENCES
) " + "
```

7:04

SQLiteSampleApp

Employee First Name
hello123

Employee Last Name
Abcd

Date of Birth (e.g 23/04/2016)
23/04/1995

Job Description
Software

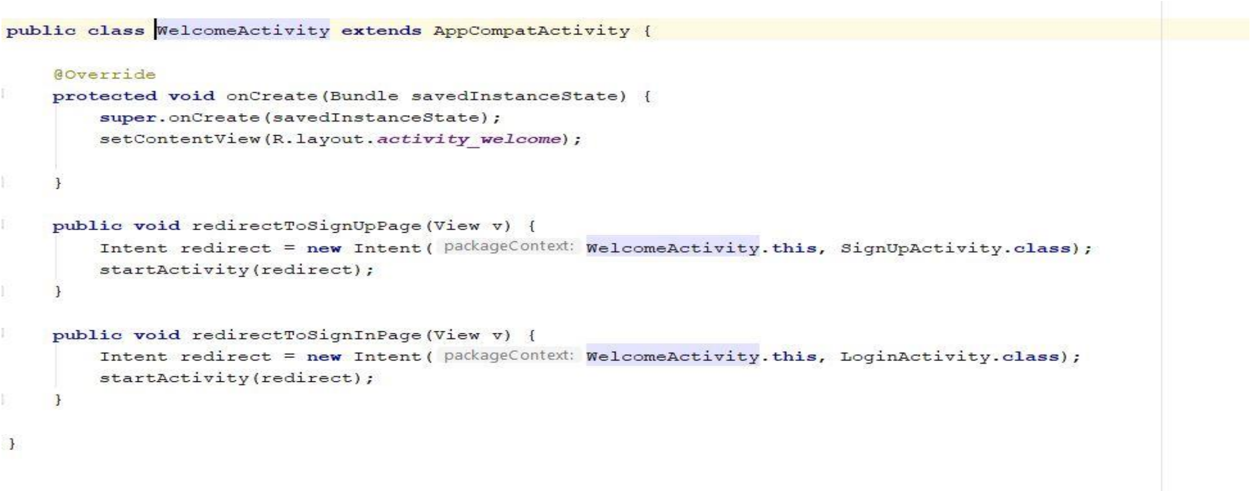
Employed Date (e.g 23/04/2016)
23/04/2016

Employer Name
TEST1

```
31 s 654 m
31 s 505 m
      4 m
    171 m
     88 m
 31 s 91 m
```

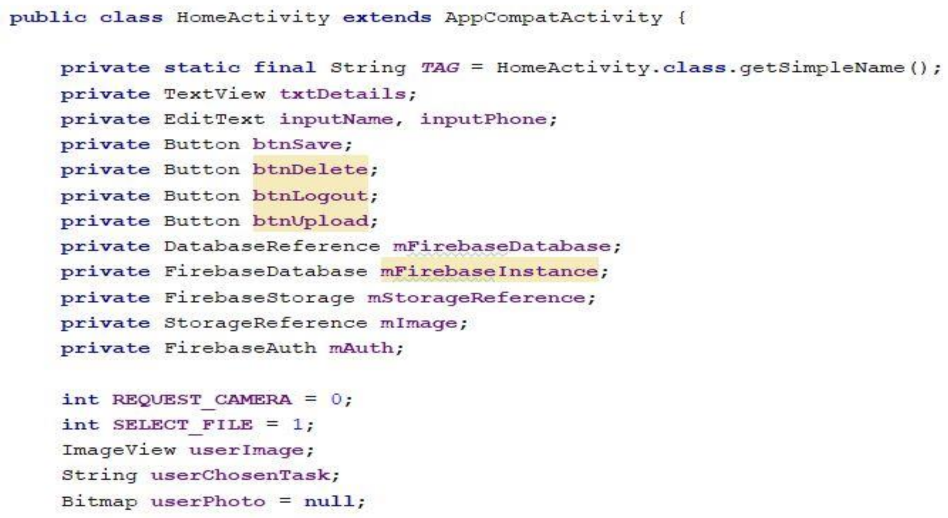
## Part-2&3 Objective:

- To create firebase database authentication.
- Adding images into the database.
- Deleting the images from application
- Creating logout button
- Corresponding code snippets for this one is shown below.
- Welcome activity having both signin and signup features shown below.

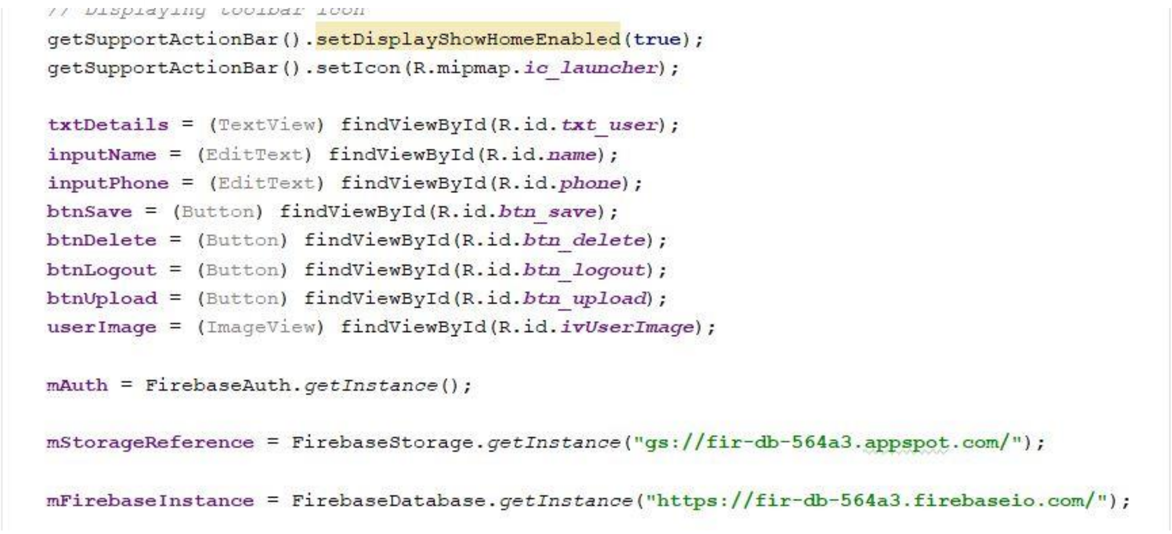


```java
public class WelcomeActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_welcome);

    }

    public void redirectToSignUpPage(View v) {
        Intent redirect = new Intent( packageContext: WelcomeActivity.this, SignUpActivity.class);
        startActivity(redirect);
    }

    public void redirectToSignInPage(View v) {
        Intent redirect = new Intent( packageContext: WelcomeActivity.this, LoginActivity.class);
        startActivity(redirect);
    }

}
```

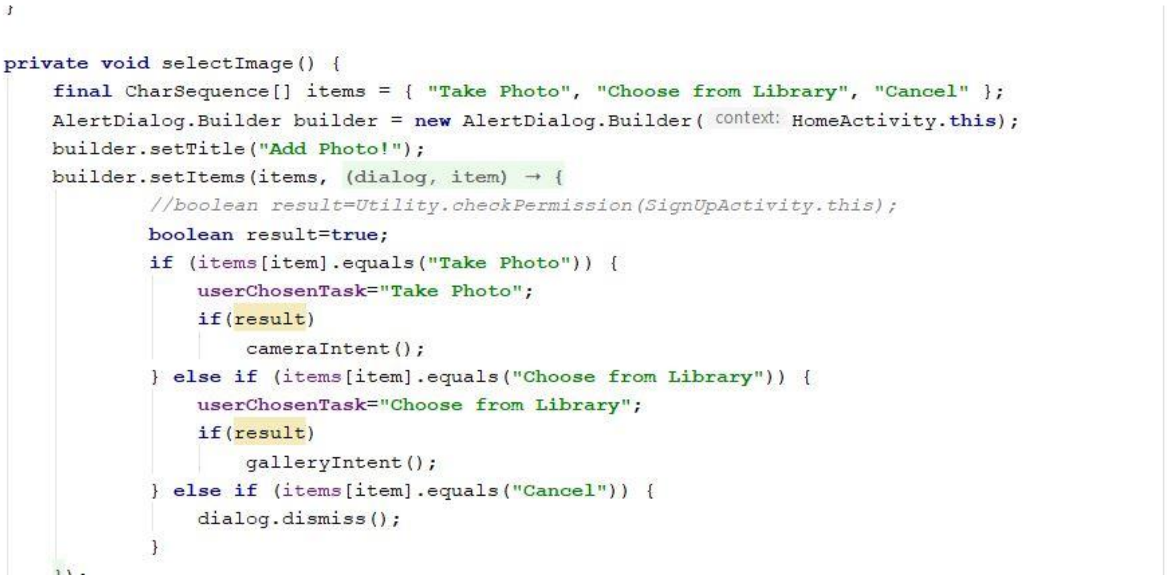
- Home activity having firebase instance and also camera accessing along with connecting to remaining activities are shown below.



```java
public class HomeActivity extends AppCompatActivity {

    private static final String TAG = HomeActivity.class.getSimpleName();
    private TextView txtDetails;
    private EditText inputName, inputPhone;
    private Button btnSave;
    private Button btnDelete;
    private Button btnLogout;
    private Button btnUpload;
    private DatabaseReference mFirebaseDatabase;
    private FirebaseDatabase mFirebaseInstance;
    private FirebaseStorage mStorageReference;
    private StorageReference mImage;
    private FirebaseAuth mAuth;

    int REQUEST_CAMERA = 0;
    int SELECT_FILE = 1;
    ImageView userImage;
    String userChosenTask;
    Bitmap userPhoto = null;
```

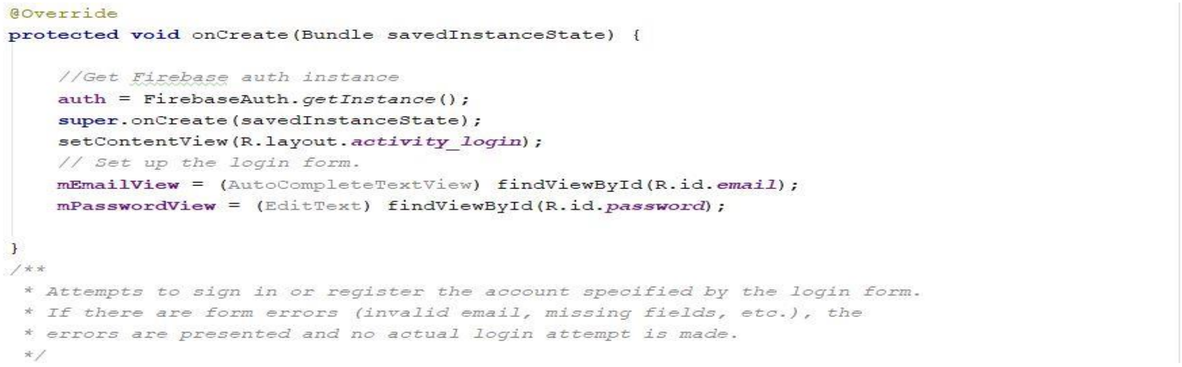


```
    // Displaying toolbar icon
    getSupportActionBar().setDisplayShowHomeEnabled(true);
    getSupportActionBar().setIcon(R.mipmap.ic_launcher);

    txtDetails = (TextView) findViewById(R.id.txt_user);
    inputName = (EditText) findViewById(R.id.name);
    inputPhone = (EditText) findViewById(R.id.phone);
    btnSave = (Button) findViewById(R.id.btn_save);
    btnDelete = (Button) findViewById(R.id.btn_delete);
    btnLogout = (Button) findViewById(R.id.btn_logout);
    btnUpload = (Button) findViewById(R.id.btn_upload);
    userImage = (ImageView) findViewById(R.id.ivUserImage);

    mAuth = FirebaseAuth.getInstance();

    mStorageReference = FirebaseStorage.getInstance("gs://fir-db-564a3.appspot.com/");

    mFirebaseInstance = FirebaseDatabase.getInstance("https://fir-db-564a3.firebaseio.com/");
```

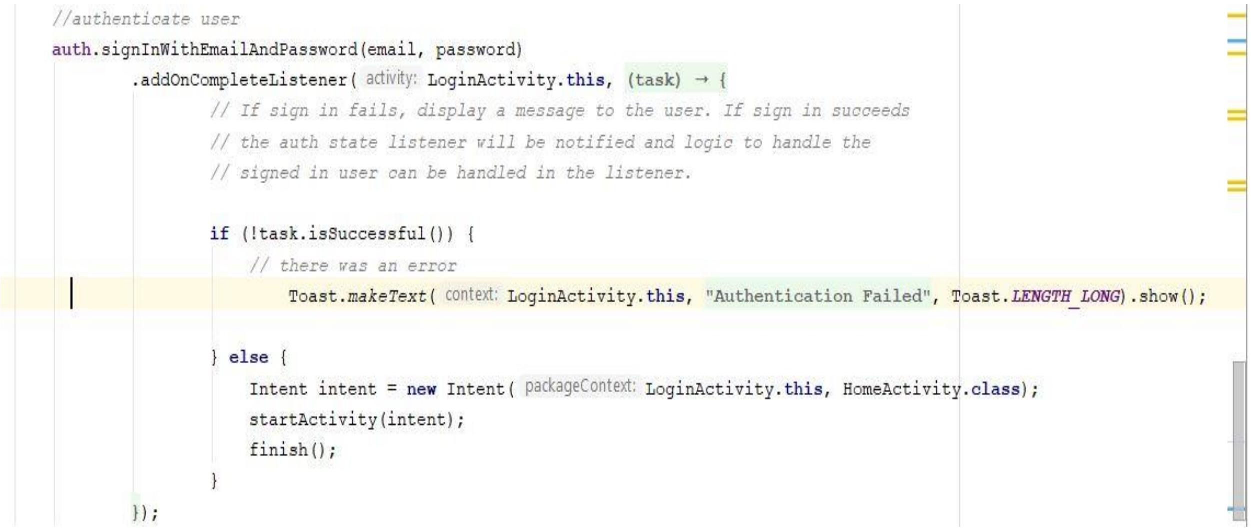


```
}

private void selectImage() {
    final CharSequence[] items = { "Take Photo", "Choose from Library", "Cancel" };
    AlertDialog.Builder builder = new AlertDialog.Builder( context: HomeActivity.this);
    builder.setTitle("Add Photo!");
    builder.setItems(items, (dialog, item) → {
            //boolean result=Utility.checkPermission(SignUpActivity.this);
            boolean result=true;
            if (items[item].equals("Take Photo")) {
                userChosenTask="Take Photo";
                if(result)
                    cameraIntent();
            } else if (items[item].equals("Choose from Library")) {
                userChosenTask="Choose from Library";
                if(result)
                    galleryIntent();
            } else if (items[item].equals("Cancel")) {
                dialog.dismiss();
            }
    });
```

- Login Activity having mail and password options shown below.

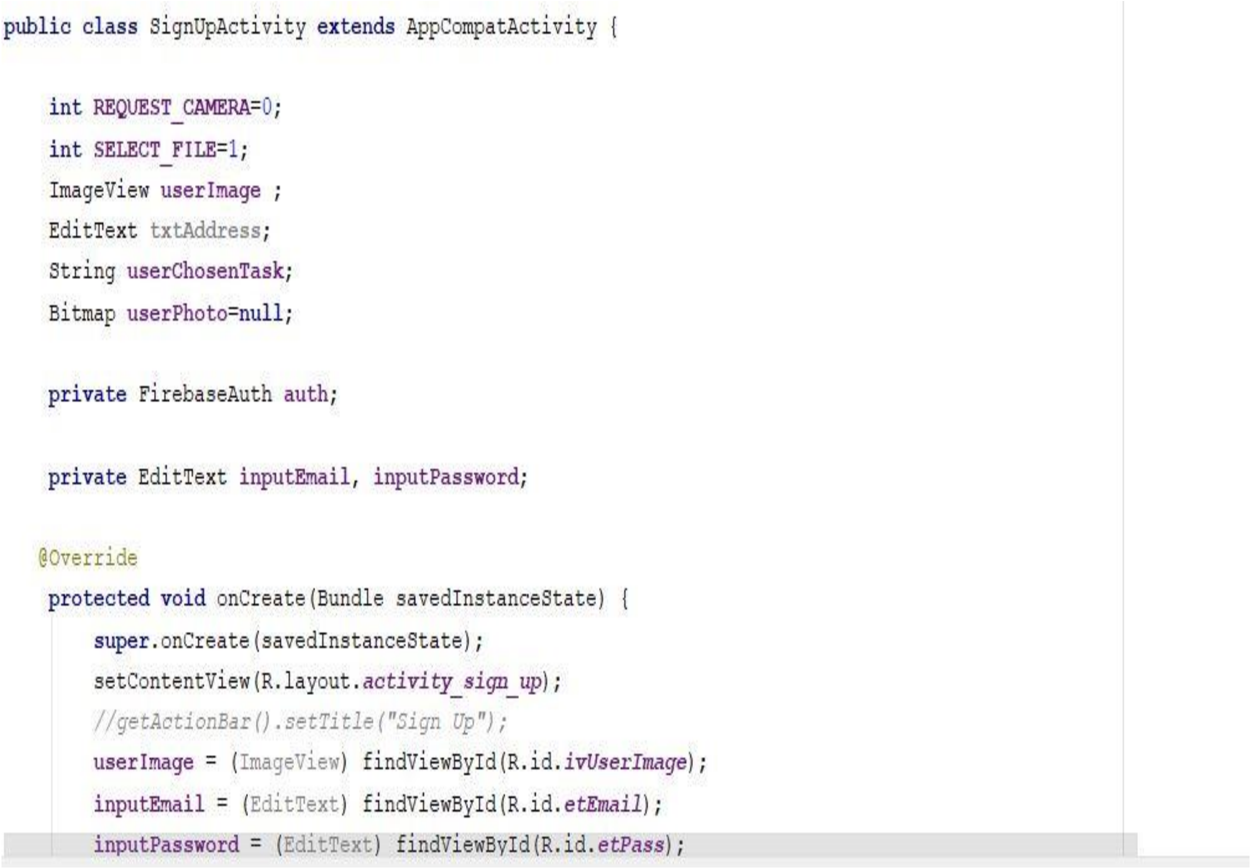


```
@Override
protected void onCreate(Bundle savedInstanceState) {

    //Get Firebase auth instance
    auth = FirebaseAuth.getInstance();
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);
    // Set up the login form.
    mEmailView = (AutoCompleteTextView) findViewById(R.id.email);
    mPasswordView = (EditText) findViewById(R.id.password);

}
/**
 * Attempts to sign in or register the account specified by the login form.
 * If there are form errors (invalid email, missing fields, etc.), the
 * errors are presented and no actual login attempt is made.
 */
```



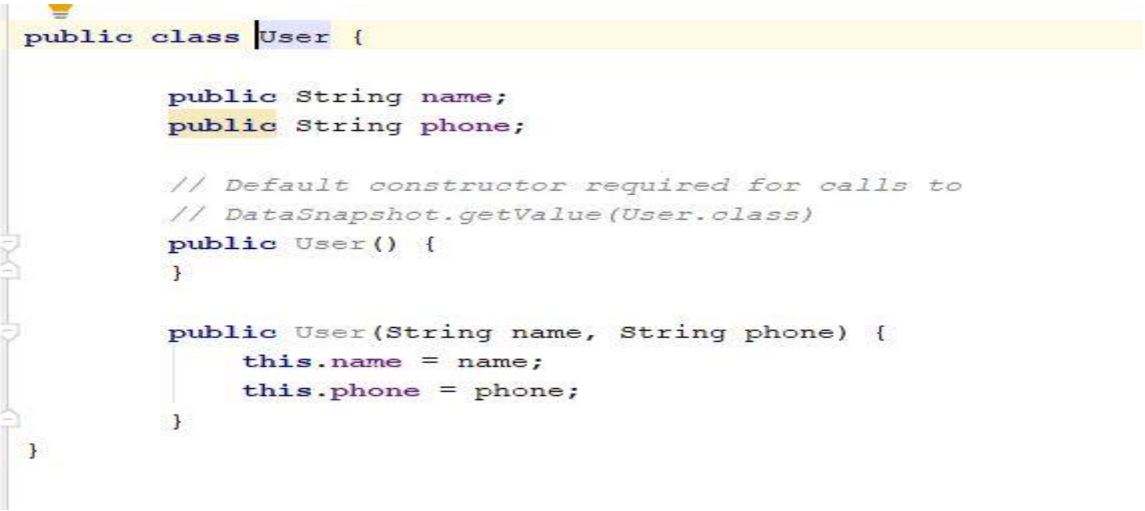
```
//authenticate user
auth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener( activity: LoginActivity.this, (task) → {
                // If sign in fails, display a message to the user. If sign in succeeds
                // the auth state listener will be notified and logic to handle the
                // signed in user can be handled in the listener.

                if (!task.isSuccessful()) {
                    // there was an error
                        Toast.makeText( context: LoginActivity.this, "Authentication Failed", Toast.LENGTH_LONG).show();

                } else {
                    Intent intent = new Intent( packageContext: LoginActivity.this, HomeActivity.class);
                    startActivity(intent);
                    finish();
                }
        });
```

- Signupactivity.java has given below.



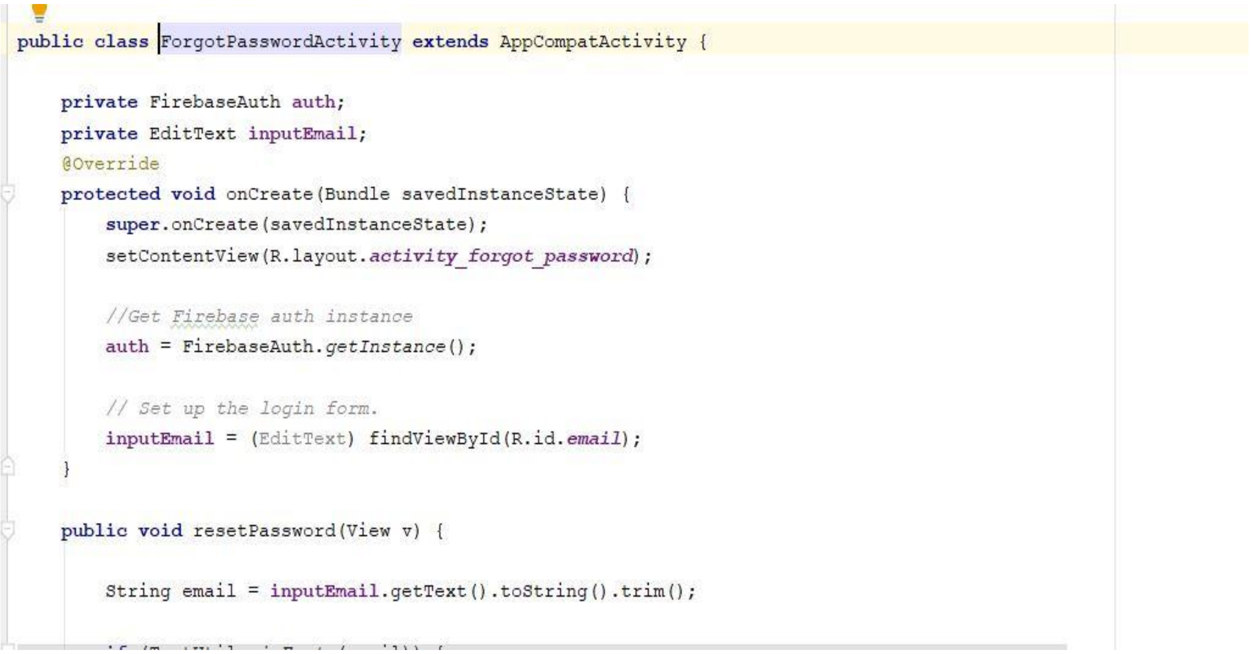
```
public class SignUpActivity extends AppCompatActivity {

    int REQUEST_CAMERA=0;
    int SELECT_FILE=1;
    ImageView userImage ;
    EditText txtAddress;
    String userChosenTask;
    Bitmap userPhoto=null;

    private FirebaseAuth auth;

    private EditText inputEmail, inputPassword;

   @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign_up);
        //getActionBar().setTitle("Sign Up");
        userImage = (ImageView) findViewById(R.id.ivUserImage);
        inputEmail = (EditText) findViewById(R.id.etEmail);
        inputPassword = (EditText) findViewById(R.id.etPass);
```

- User has string name and password in home page.



```
public class User {

        public String name;
        public String phone;

        // Default constructor required for calls to
        // DataSnapshot.getValue(User.class)
        public User() {
        }

        public User(String name, String phone) {
            this.name = name;
            this.phone = phone;
        }
}
```

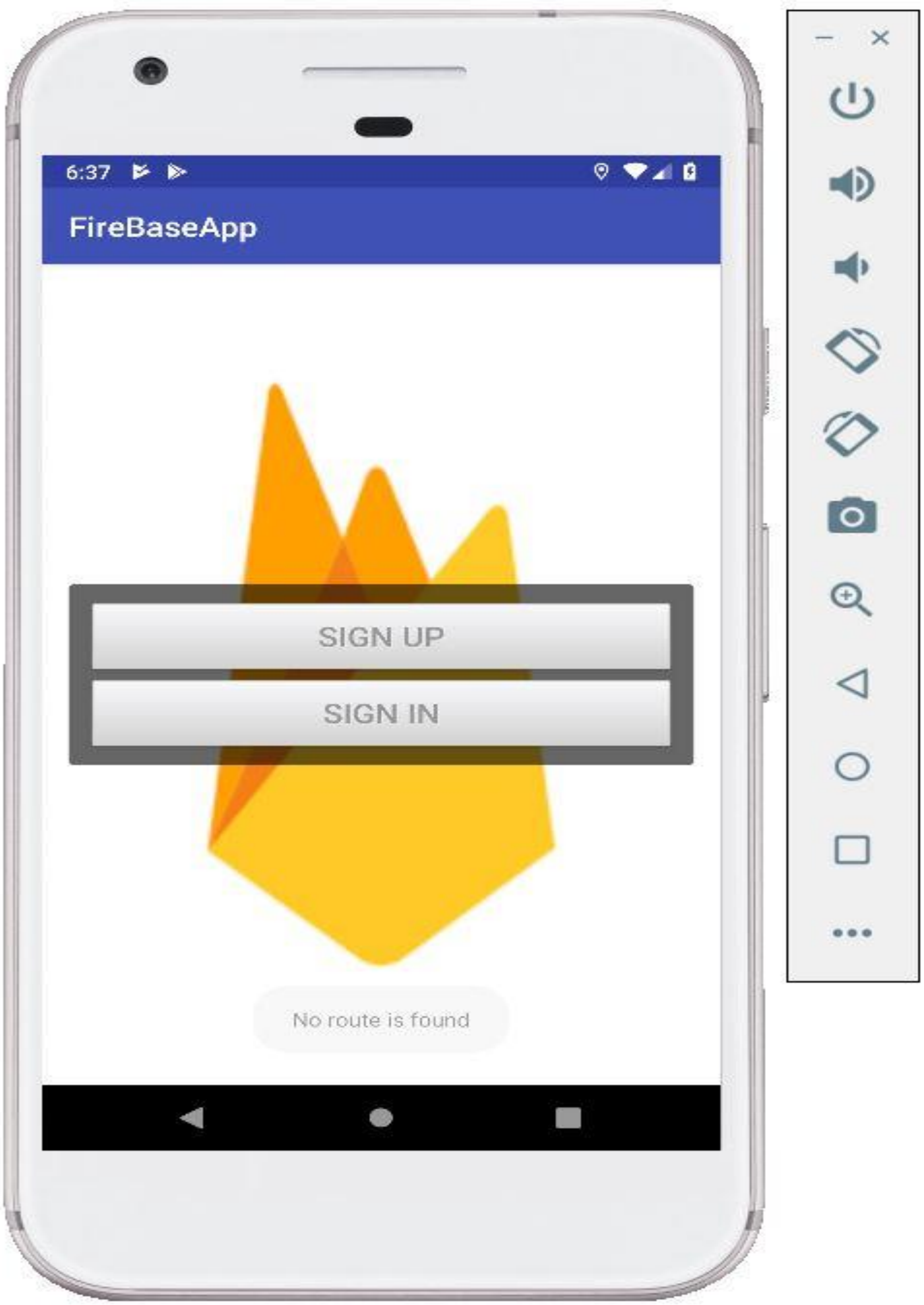
- User can recreate the password if they forget their password shown in forget password activity.



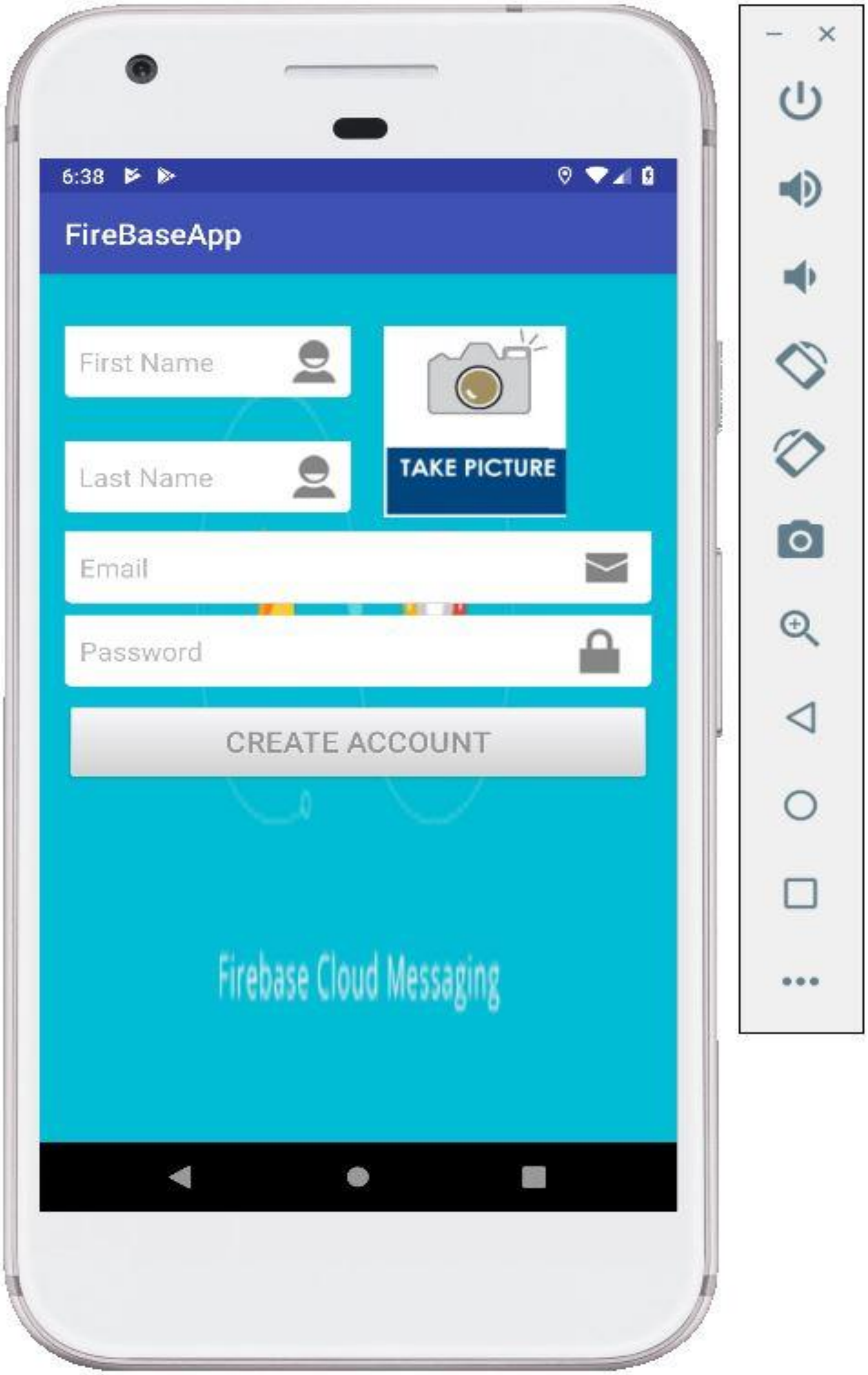
```
public class ForgotPasswordActivity extends AppCompatActivity {

    private FirebaseAuth auth;
    private EditText inputEmail;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_forgot_password);

        //Get Firebase auth instance
        auth = FirebaseAuth.getInstance();

        // Set up the login form.
        inputEmail = (EditText) findViewById(R.id.email);
    }

    public void resetPassword(View v) {

        String email = inputEmail.getText().toString().trim();

        if (TextUtils.isEmpty(email)) {
```

## Part 2&3 Output:

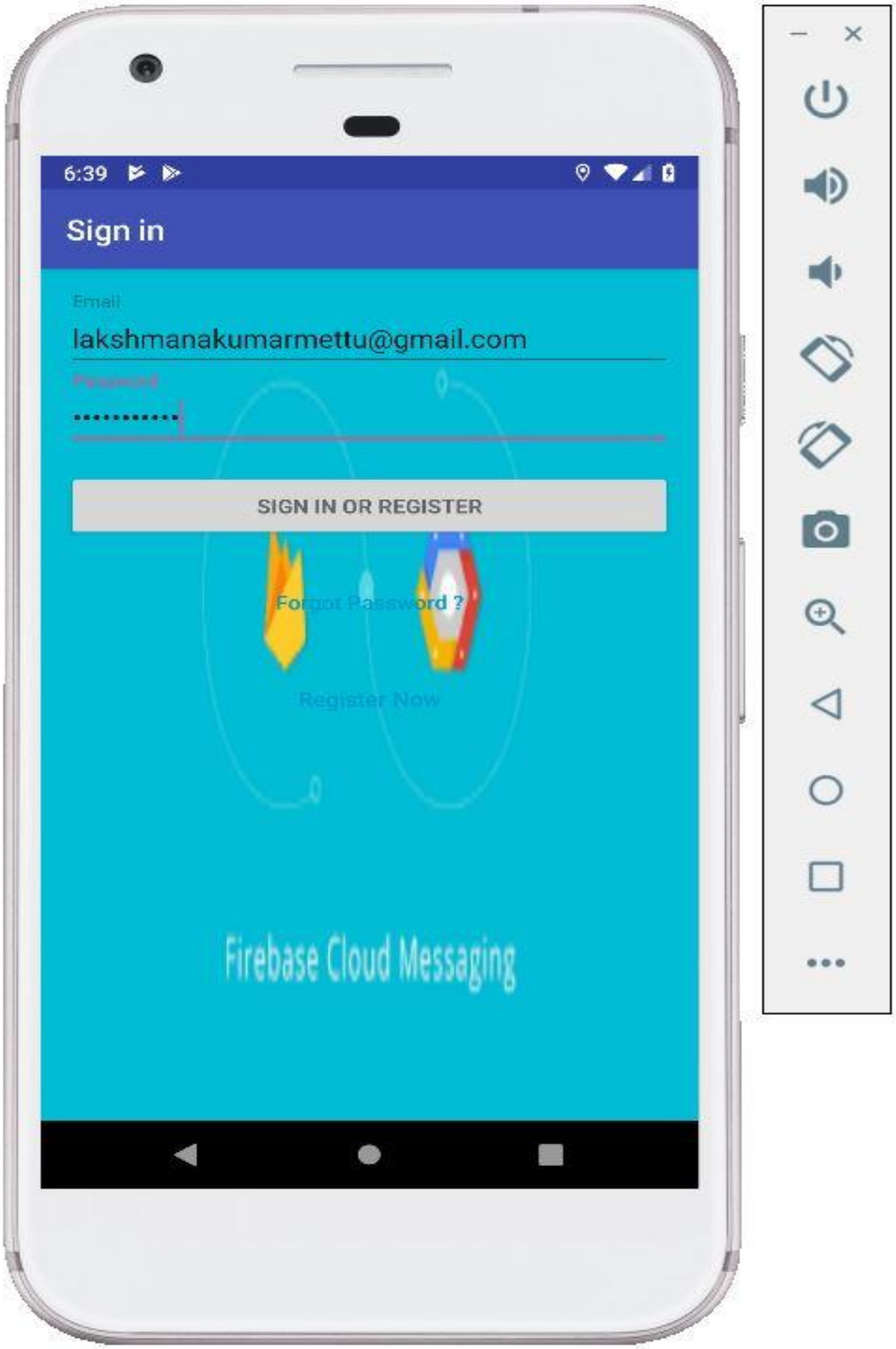
- Welcome page



- Signup activity



- Signin or Register



Sign in

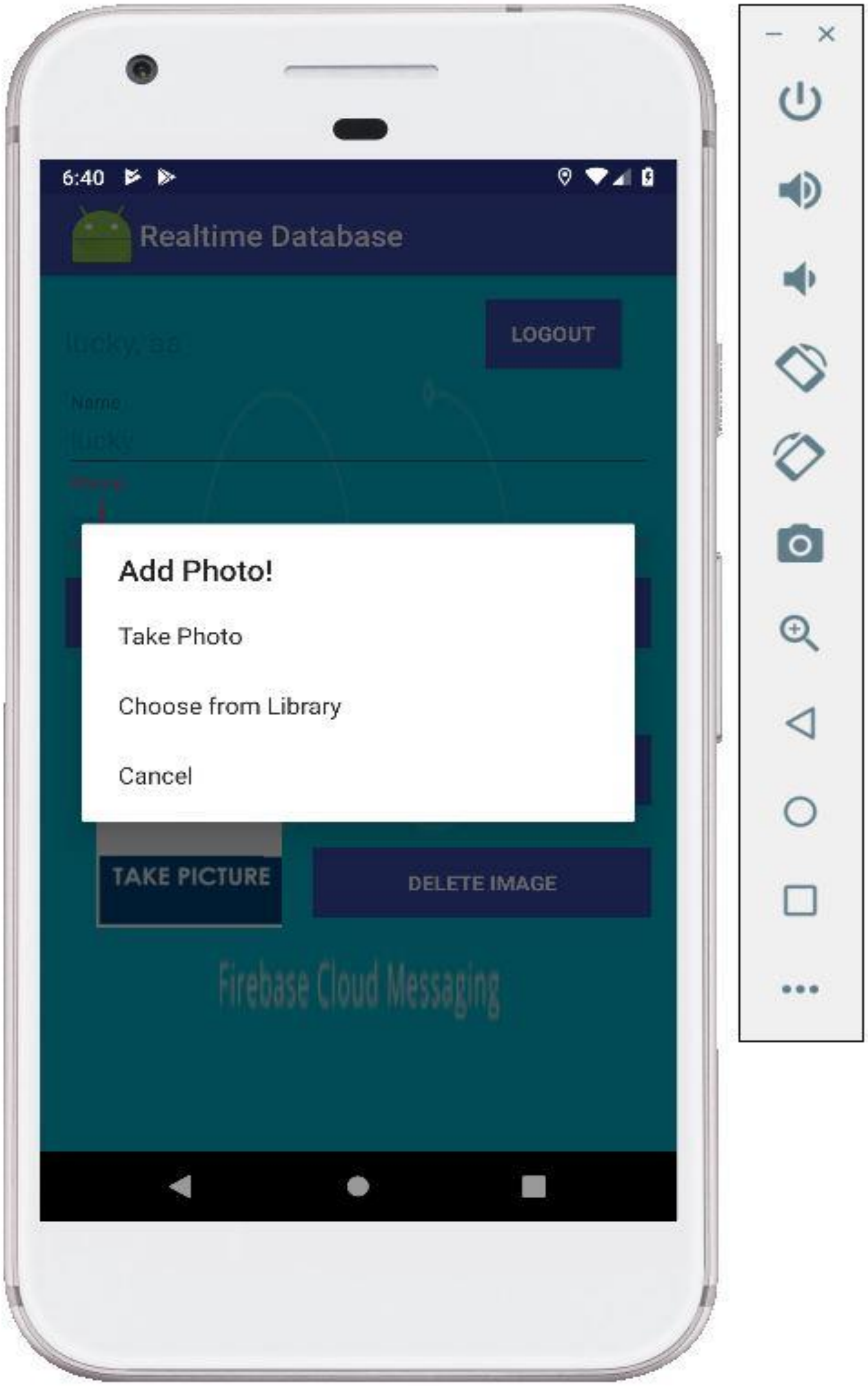
Email

lakshmanakumarmettu@gmail.com

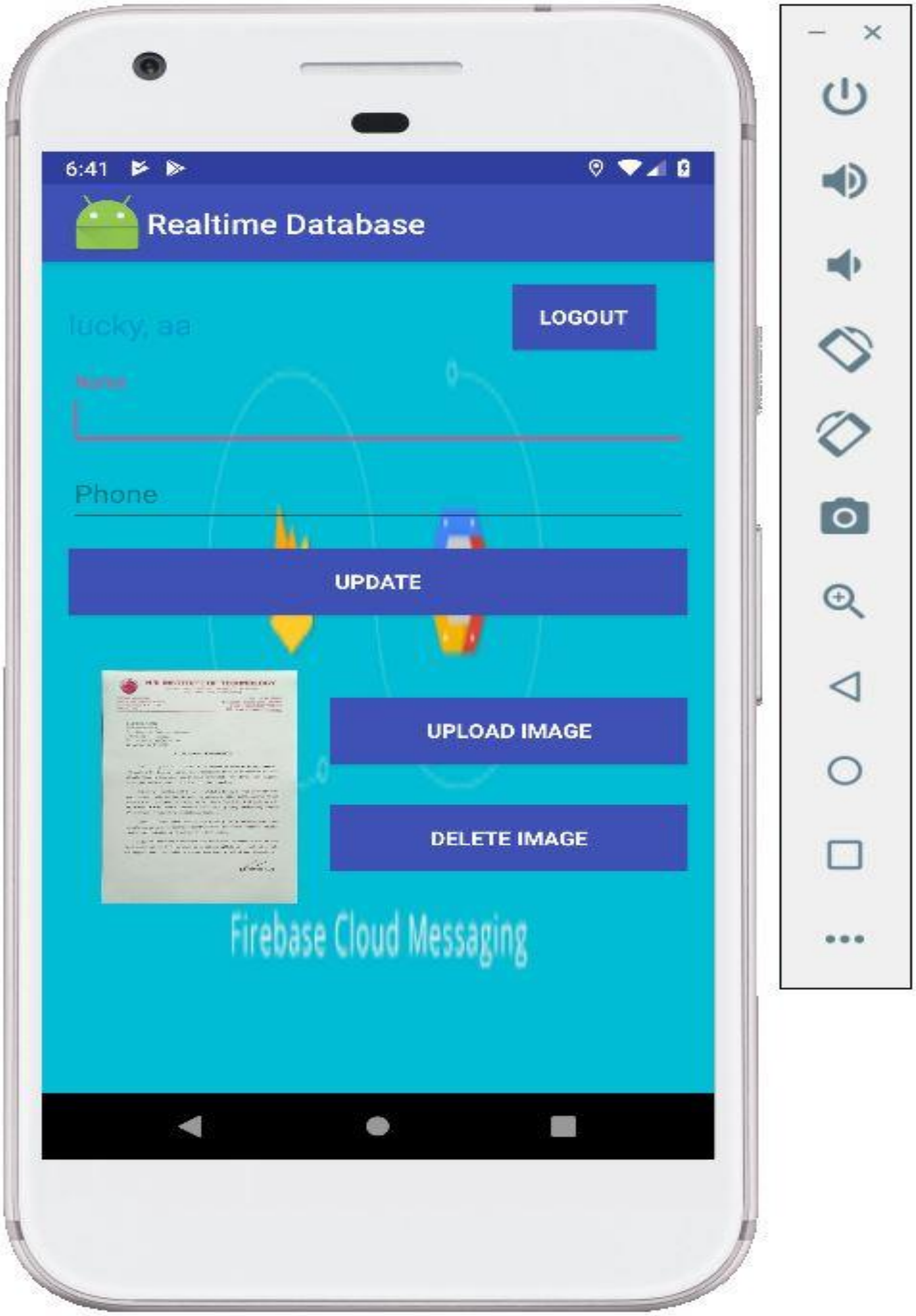
Password

••••••••••••

- Home activity having photo uploading feature.

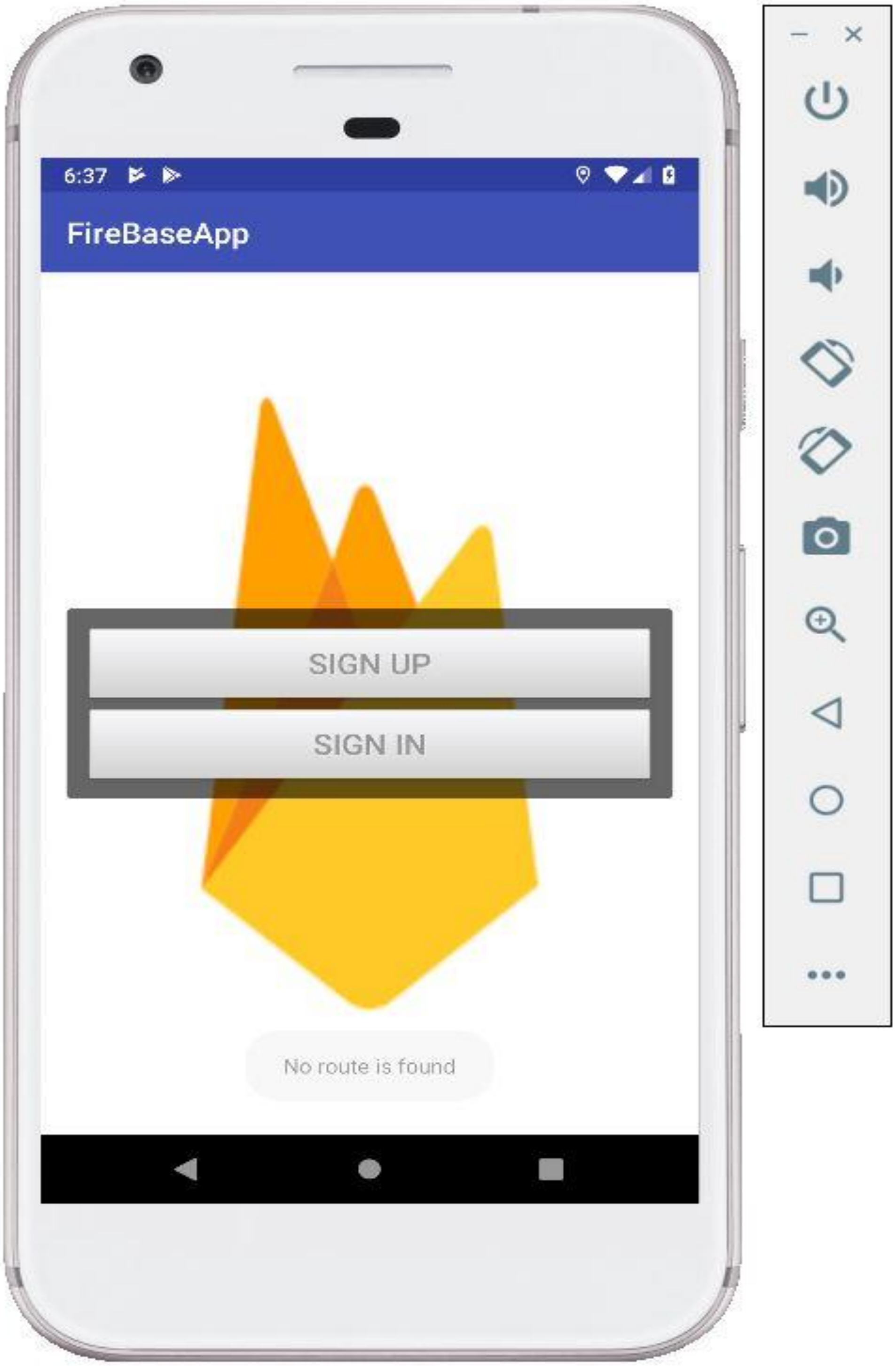


- Uploaded image and also having delete image feature.

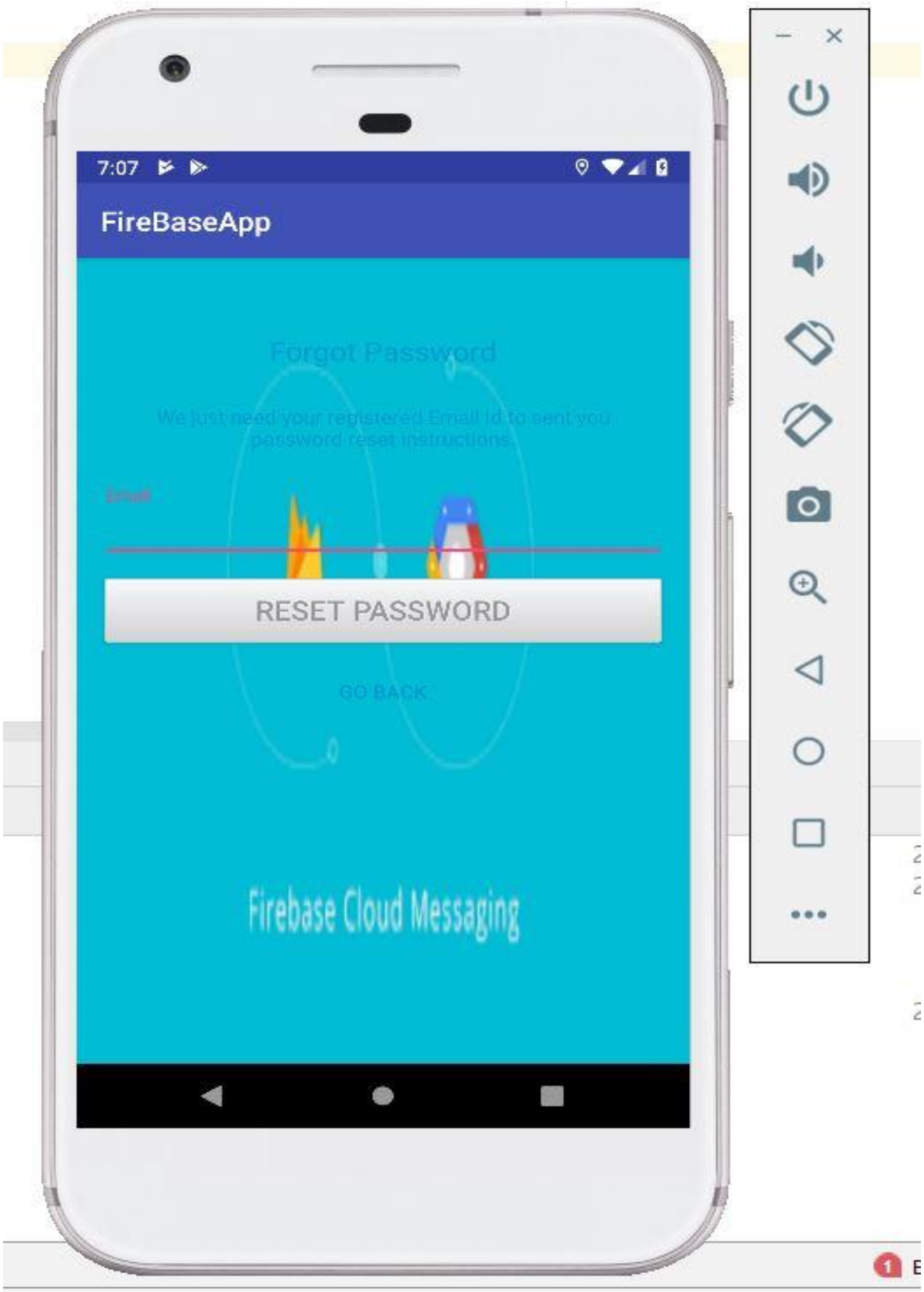


lucky, aa

- When you click on logout button it redirects to welcome activity.



- Forget password option



Forgot Password

We just need your registered Email Id to sent you password reset instructions.

## Conclusion:

- Hence Using this ICP, learnt about real database applications in android and usage of android elements like camera and modification of details in real time database.