##### A Project report on

##### Tracker And Trigger

###### A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

**Bachelor of Technology**

**in**

**Computer Science and Engineering**

Submitted by

##### S.Sree (19H51A0556)

##### S.Abhinav (19H51A0557)

##### B.Snigdha (19H51A0558)

Under the esteemed guidance of

Dr. L.Chandra Sekhar Reddy

Assistant Professor



**Department of Computer Science and Engineering**

**CMR COLLEGE OF ENGINEERING AND TECHNOLOGY**

(An Autonomous Institution under UGC & JNTUH , Approved by AICTE, Permanently Affiliated to JNTUH, Accredited by NBA.)

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

#### 2019- 2023

**CMR COLLEGE OF ENGINEERING & TECHNOLOGY**

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the dissertation entitled **“TRACKER AND TRIGGER”** is a bonafied work done by **S.Sree (19H51A0556), S.Abhinav (19H51A0557), M.Snigdha (19H51A0558),** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Information Technology, submitted to the Department of Information Technology, CMR College of Engineering & Technology, Hyderabad during the academic year 2019-2023. The Results embodies in this project report have not been submitted to any other university or institute for the award of any degree.

**Dr. L.Chandra Sekhar Reddy Dr. K.Vijaya kumar**

**ASSISTANT PROFESSOR Professor and HOD**

**Dept. of CSE Dept. of CSE**

Submitted for viva voice Examination held on \_\_\_\_\_\_\_\_\_\_\_

**External Examine**

**ABSTRACT**

‘**Tracker and Trigger’** is an android app to keep track of the regular activities and to help people in managing their daily chores. This app can be used to manage groceries, kitchen appliances, home maintenance details. Users of the app can keep track of their groceries with quantity, books, furniture, grooming materials, medical reports and others with images. It provides a customised message with the necessary list of documents to the mobile number as well as mail id one hour before for meetings and other to-do activities scheduled. The user can then share the image and other details of any material or appliances which is available in their to-do list with their family, friends, relatives and colleagues through Gmail or WhatsApp.

**TABLE OF CONTENTS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CHAPTERS** | |  | **DESCRIPTION** | **PAGE NUMBER** |
|  |  |  | Abstract | i |
|  |  |  | List of figures | iii |
| 1 |  |  | **Introduction** | 1 |
|  | 1.1 |  | Description | 1 |
|  | 1.2 |  | Scope | 1 |
|  | 1.3 |  | Objectives | 1 |
| 2 |  |  | **Background Work** | 2 |
|  | 2.1 |  | Existing solutions | 2 |
|  | 2.2 |  | Disadvantages | 2 |
| 3 |  |  | **Proposed statement** | 3-4 |
|  | 3.1 |  | Proposed solutions | 3 |
|  |  | 3.1.1 | Advantages of Proposed solutions | 3 |
|  | 3.2 |  | Requirements | 3 |
|  |  | 3.2.1 | Requirements for hardware | 3 |
|  |  | 3.2.2 | Requirements for software | 3 |
|  | 3.3 |  | Description | 4 |
|  | 3.4 |  | App Configuration | 4 |
| 4 |  |  | **Designing** | 5 -6 |
| 5 |  |  | **Results and Conclusion** | 7-23 |
|  | 5.1 |  | Results and conclusion | 7 |
|  | 5.2 |  | Source code | 8-23 |
|  | 5.3 |  | Performance Analysis | 23 |
| 6 | 6.1 |  | **Conclusion** | 24 |
|  | 6.2 |  | Future Work | 24 |
|  | 6.3 |  | References | 24 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NUMBER** | **DESCRIPTION** | **PAGE NUMBER** |
| 4 | UML DIAGRAM | 5 |
| 4 | ER DIAGRAM | 6 |

1) **INTRODUCTION**

**1.1 DESCRIPTION**

* In modern household, due to hectic work schedules, it is quite difficult to remember as well as manage the availability of mandatory commodities in an urban home.
* Not only working professionals, homemakers, job-seekers, and others fail to keep track of their inventories or scheduled meetings which in turn either delay or terminate their regular duties.
* Sometimes, most of the people forget that they already have sufficient stock of certain item, and end up re-buying the same stuff again (toothpaste, soap, face wash, etc.) as a result lose money and time.
* The inability of efficiently managing the home inventories will not only impacts work schedules, but also causes stress, anger, and impatience. An automated application is proposed and intended to develop in order to keep track of the regular activities and aid people in managing the daily home inventories.

**1.2 SCOPE**

* This application helps us to know what should be done at what time with a gentle reminder.
* We don’t need to remember everything as this application intimates us about what we should do.

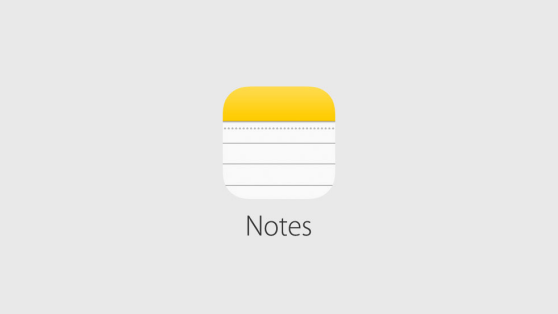
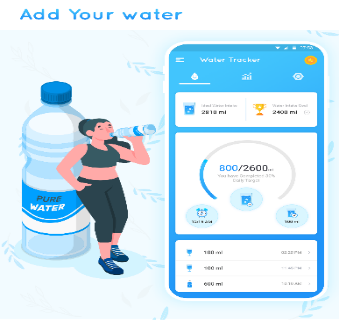
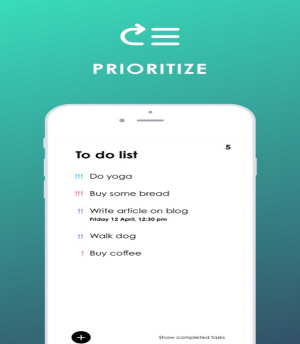
**1.3 OBJECTIVES**

* The main objective of this project is to remind the people what needs to be done now.
* To remove the workload that is kept on the brain and give reminders in scheduled time.
* To have a track of necessary groceries and general usage of items plus reminds when they are in shortage.

2) **BACKGROUND WORK**

**2.1 EXISTING SOLUTION**

* TO DO LIST
* Water Reminder App
* Maintenance App
* Notes

**  **

**2.2 DISADVANTAGES**

* Limited only for a particular use of field such as healthcare, maintenance and home appliances.
* Most of them are paid apps.
* Contains a lot of advertisements and not secure.

**3) PROPOSED SYSTEM**

## **3.1) PROPOSED SOLUTIONS**

Our idea for the existing problem is to develop an android app in order to keep track of the regular activities and help people in managing the daily home activities. App can be used to manage groceries, kitchen appliances, home maintenance details. Users of the app can keep track of their groceries with quantity, books, furniture, grooming materials, medical reports and others with image. It provides customized message with necessary list of documents to the mobile number as well as mail id one hour before for meetings and other to-do activities scheduled. User can share the image and other details of any material or appliances which is available in to-do list with their family, friends, relatives and colleagues through Gmail or WhatsApp.

**3.1.1) Advantages of proposed solutions**

* Easy to use and free.
* Covers all the fields like healthcare, maintenance and home appliances.
* No Advertisements and harmful APIs.

**3.2) REQUIREMENTS**

Our project includes two types of system requirements:

* 1. Hardware
  2. Software

* + 1. **Requirements for hardware**
* 64-bit Microsoft® Windows® 8/10.
* x86\_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor.
* 8 GB RAM or more.
* 8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)
* 1280 x 800 minimum screen resolution.
  + 1. **Requirements for Software**
* Android Studio
* Java
* Firebase database

**3.3) DESCRIPTION**

**Android Studio** is the official Integrated Development Environment (IDE) for android application development. Android Studio provides more features that enhance our productivity while building Android apps.It has a flexible gradle-based build system. It has a fast and feature-rich emulator for app testing. Android Studio has a consolidated environment where we can develop for all Android devices. Apply changes to the resource code of our running app without restarting the app. Android Studio provides extensive testing tools and frameworks .It supports C++ and NDK. It provides build-in supports for Google Cloud Platform. It makes it easy to integrate Google Cloud Messaging and App Engine.

The Firebase real time Database is a cloud-hosted database in which data is stored as JSON. The data is synchronized in real-time to every connected client. All of our clients share one real time Database instances and automatically receive updates with the newest data, when we build cross-platform applications with our iOS, and JavaScript SDKs. The Firebase real time Database is a NoSQL database from which we can store and sync the data between our users in real-time. It is a big JSON object which the developers can manage in real-time. By using a single API, the Firebase database provides the application with the current value of the data and updates to that data. Real-time syncing makes it easy for our users to access their data from any device, be it web or mobile. The real time database helps our users collaborate with one another. It ships with mobile and web SDKs, which allow us to build our app without the need for servers. When our users go offline, the Real-time Database SDKs use local cache on the device for serving and storing changes. The local data is automatically synchronized, when the device comes online.

**3.4) APP CONFIGURATION**

SDK version :- 30 Android 11®

Build tools version :- 30.0.2

Android gradle version :- 4.1.1

Size :- 9.1 MB

**4) DESIGNING**

**4.1 UML DIAGRAM :-**

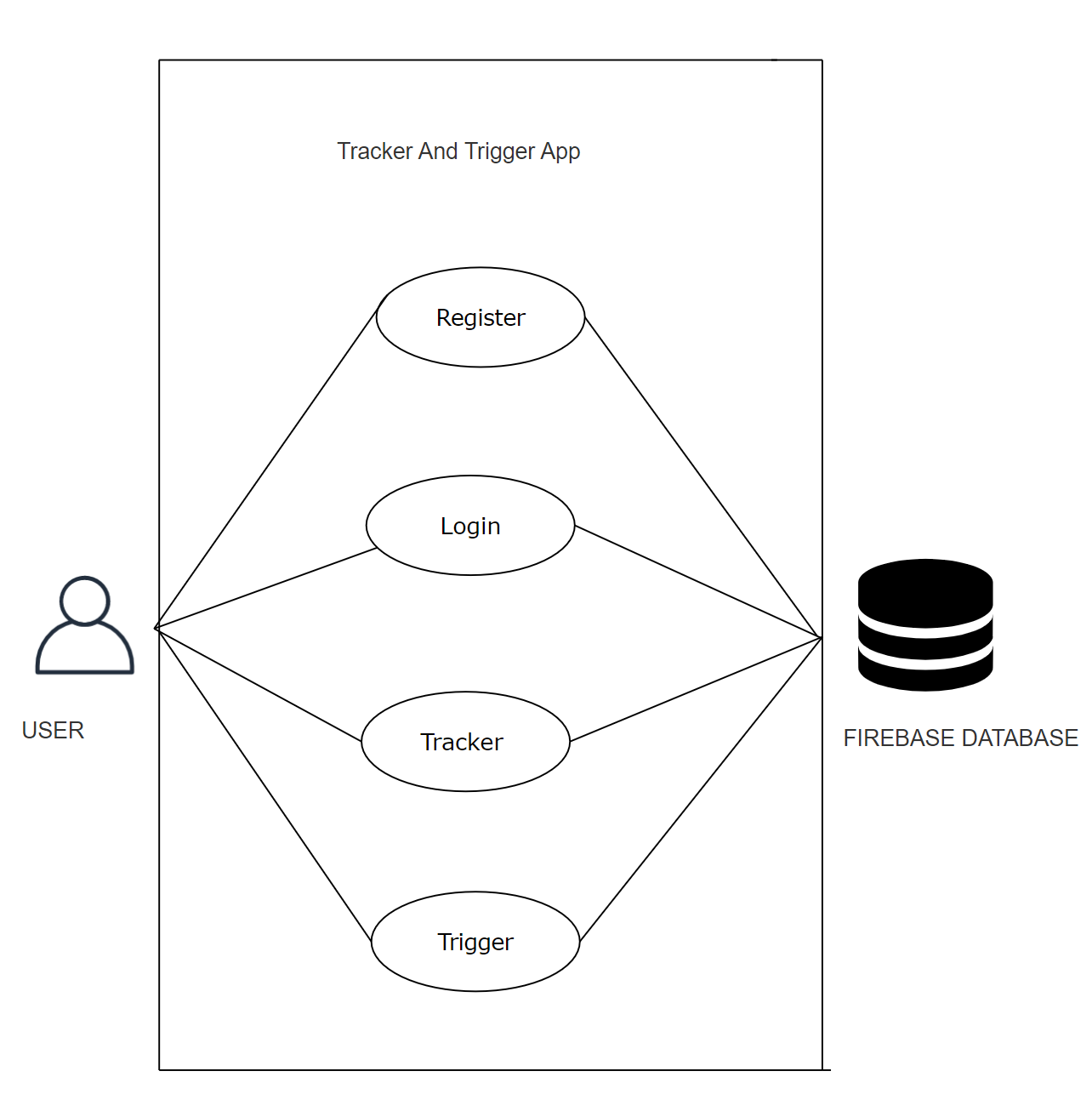


Figure 1: use case diagram for user and database interface.

OTP verification :-

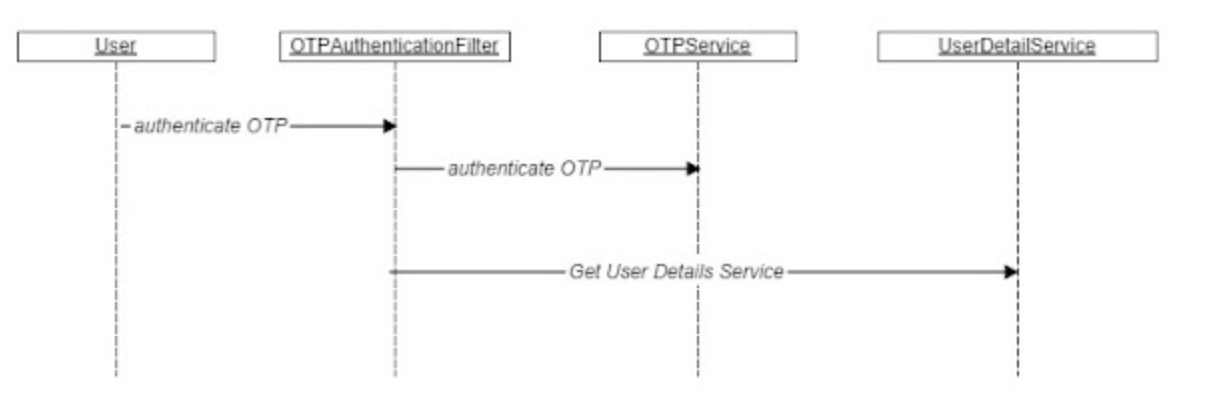


Figure 2:sequence diagram for OTP verification

**4.2 ER DIAGRAM**

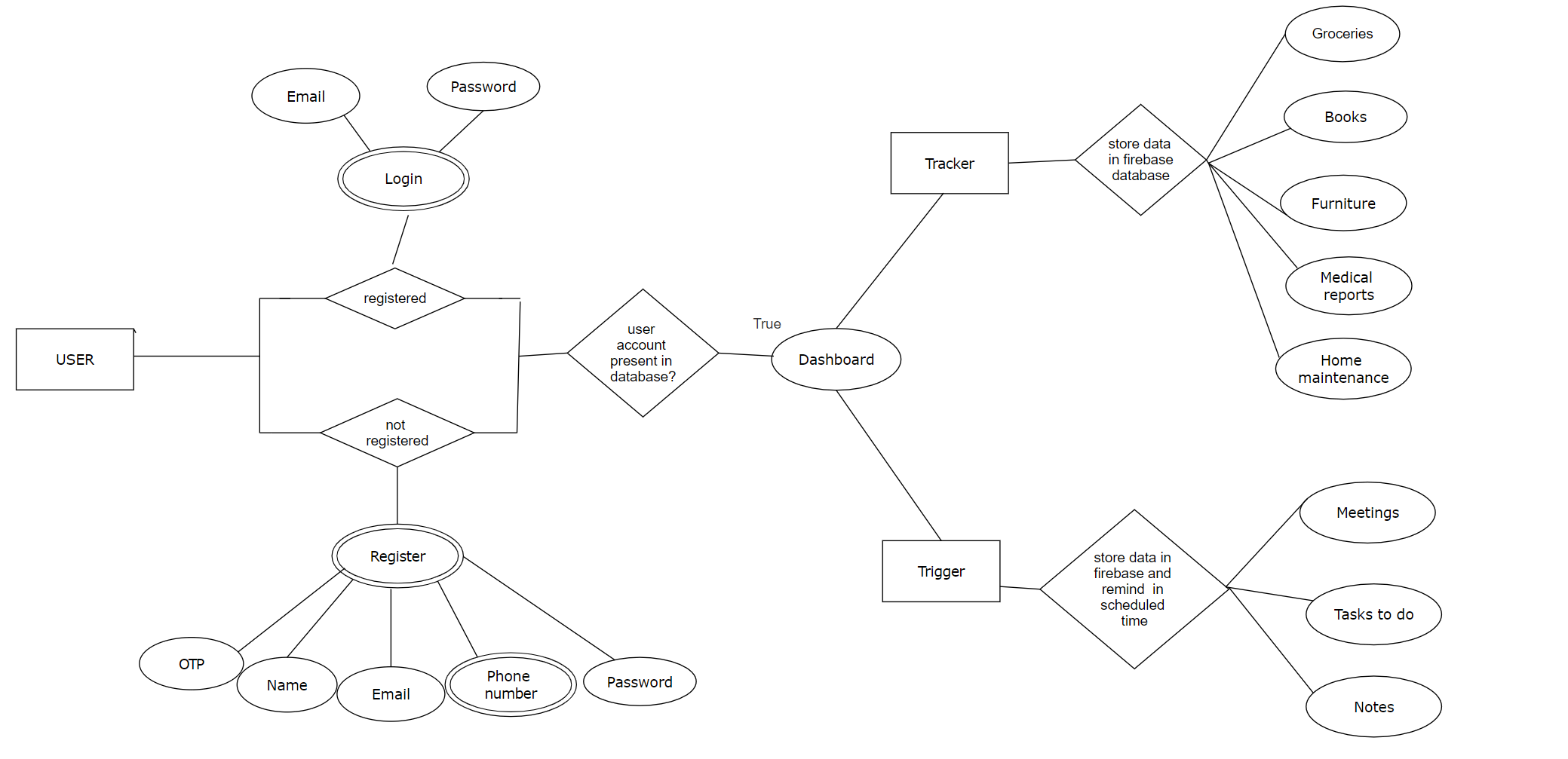
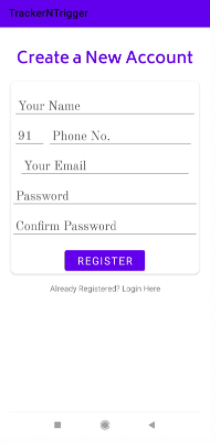
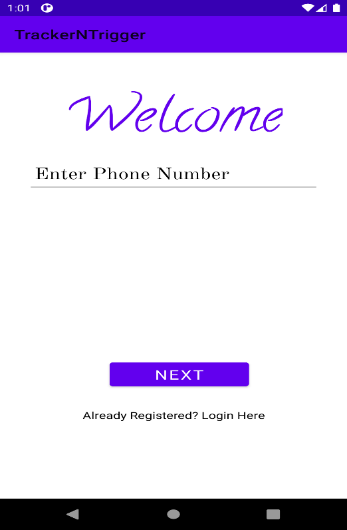
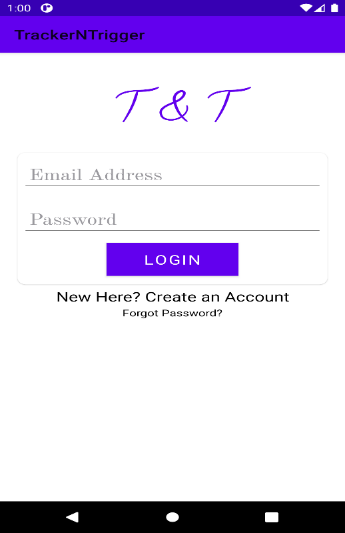
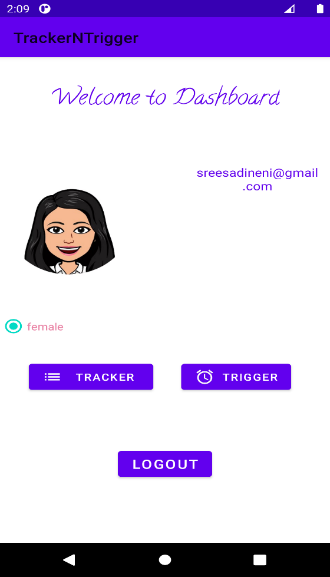
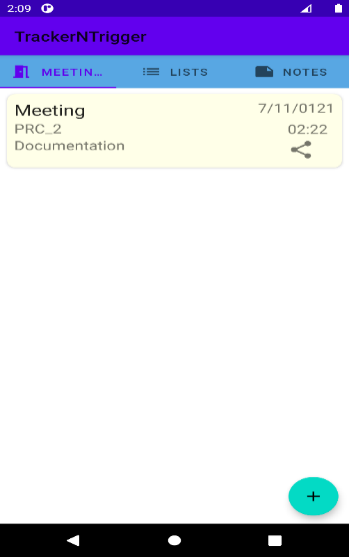


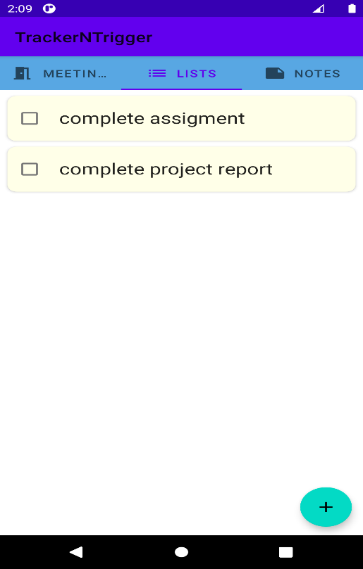
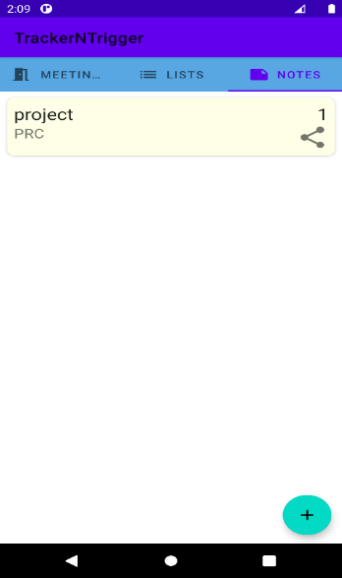
Figure 3 : Entity relation diagram of the app

**5) RESULTS AND DISCUSSIONS**

**5.1) RESULTS AND DISCUSSIONS:-**

  ****

** **

**5.2) SOURCE CODE :-**

package com.example.myapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnFailureListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.android.material.button.MaterialButton;  
import com.google.android.material.textview.MaterialTextView;  
import com.google.firebase.FirebaseException;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.PhoneAuthCredential;  
import com.google.firebase.auth.PhoneAuthProvider;  
import com.google.firebase.firestore.CollectionReference;  
import com.google.firebase.firestore.DocumentReference;  
import com.google.firebase.firestore.DocumentSnapshot;  
import com.google.firebase.firestore.FirebaseFirestore;  
import com.google.firebase.firestore.Query;  
import com.google.firebase.firestore.QueryDocumentSnapshot;  
import com.google.firebase.firestore.QuerySnapshot;  
  
import java.util.HashMap;  
import java.util.Map;  
import java.util.concurrent.TimeUnit;  
  
public class Phone extends AppCompatActivity {  
  
 EditText mPhone, mCC, mOTP;  
 String otpCode = "123456";  
 FirebaseAuth fAuth;  
 FirebaseFirestore fStore;  
 ProgressBar;  
 String verificationID;  
 Boolean verificationOnProgress = false, check = false;  
 PhoneAuthProvider.ForceResendingToken token;  
 PhoneAuthCredential credential;  
 String PhoneNum, FullName, Email, userID;  
 int count=0;  
 MaterialTextView state,resend, mloginpageBtn;  
 MaterialButton mNextBtn;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_phone*);  
 mPhone = findViewById(R.id.*Phone*);  
 mCC = findViewById(R.id.*CC*);  
 mOTP = findViewById(R.id.*OTP*);  
 mNextBtn = findViewById(R.id.*submitBtn*);  
 progressBar = findViewById(R.id.*progressBar*);  
 fAuth = FirebaseAuth.*getInstance*();  
 state = findViewById(R.id.*state*);  
 fStore = FirebaseFirestore.*getInstance*();  
 mloginpageBtn = findViewById(R.id.*textView3*);  
 Intent = getIntent();  
 Email = intent.getStringExtra("email\_key");  
 Intent intent1 = getIntent();  
 FullName = intent1.getStringExtra("name\_key");  
 mloginpageBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startActivity(new Intent(getApplicationContext(),Login.class));  
 finish();  
 }  
 });  
 mNextBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 if(!(mPhone.getText().toString().isEmpty()) && mPhone.getText().toString().length() == 10) {  
 PhoneNum = "+91" + mPhone.getText().toString();  
 if (!verificationOnProgress) {  
 mNextBtn.setEnabled(false);  
 progressBar.setVisibility(View.*VISIBLE*);  
 state.setVisibility(View.*VISIBLE*);  
 Toast.*makeText*(Phone.this, "Sending OTP...", Toast.*LENGTH\_SHORT*).show();  
 Log.*d*("phone", "Phone No.: " + PhoneNum);  
 requestPhoneAuth(PhoneNum);  
 } else {  
 mNextBtn.setEnabled(false);  
 mOTP.setVisibility(View.*GONE*);  
 progressBar.setVisibility(View.*VISIBLE*);  
 state.setText("Logging in");  
 state.setVisibility(View.*VISIBLE*);  
 otpCode = mOTP.getText().toString();  
 if (otpCode.isEmpty()) {  
 mOTP.setError("Required");  
 return;  
 }  
 credential = PhoneAuthProvider.*getCredential*(verificationID, otpCode);  
 verifyAuth(credential);  
 }  
 }  
 else{  
 Toast.*makeText*(Phone.this, "Phone number is not valid", Toast.*LENGTH\_SHORT*).show();  
 }  
  
 }  
 });  
 }  
 private void requestPhoneAuth(String phoneNum) {  
 PhoneAuthProvider.*getInstance*().verifyPhoneNumber(PhoneNum,60L, TimeUnit.*SECONDS*,this,  
 new PhoneAuthProvider.OnVerificationStateChangedCallbacks(){  
  
 @Override  
 public void onCodeAutoRetrievalTimeOut(String s) {  
 super.onCodeAutoRetrievalTimeOut(s);  
 Toast.*makeText*(Phone.this, "OTP Timeout, Please Re-generate the OTP Again.", Toast.*LENGTH\_SHORT*).show();  
 startActivity(new Intent(getApplicationContext(), Phone.class));  
 }  
  
 @Override  
 public void onCodeSent(String s, PhoneAuthProvider.ForceResendingToken forceResendingToken) {  
 super.onCodeSent(s, forceResendingToken);  
 verificationID = s;  
 token = forceResendingToken;  
 verificationOnProgress = true;  
 progressBar.setVisibility(View.*GONE*);  
 state.setVisibility(View.*GONE*);  
 mNextBtn.setText("Verify");  
 mNextBtn.setEnabled(true);  
 mOTP.setVisibility(View.*VISIBLE*);  
 }  
  
 @Override  
 public void onVerificationCompleted(PhoneAuthCredential phoneAuthCredential) {  
  
 *// called if otp is automatically detected by the app* verifyAuth(phoneAuthCredential);  
  
 }  
  
 @Override  
 public void onVerificationFailed(FirebaseException e) {  
 Toast.*makeText*(Phone.this, e.getMessage(), Toast.*LENGTH\_SHORT*).show();  
  
 }  
 });  
 }  
  
 private void verifyAuth(PhoneAuthCredential credential) {  
 fAuth.signInWithCredential(credential).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
 checkUserProfile();  
  
 }  
 }  
 });  
 }  
 private void checkUserProfile() {  
 DocumentReference docRef = fStore.collection("phone").document(fAuth.getCurrentUser().getUid());  
 docRef.get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {  
 @Override  
 public void onSuccess(DocumentSnapshot documentSnapshot) {  
 if(documentSnapshot.exists()){  
 Toast.*makeText*(Phone.this, "Profile Exists", Toast.*LENGTH\_SHORT*).show();  
 mPhone.setError("number already exists");  
 startActivity(new Intent(getApplicationContext(),Phone.class));  
 finish();  
 }else {  
 Toast.*makeText*(Phone.this, "Profile Do not Exists.", Toast.*LENGTH\_SHORT*).show();  
  
 check = true;  
 DocumentReference docRef = fStore.collection("phone").document(fAuth.getCurrentUser().getUid());  
 Map<String,Object> user = new HashMap<>();  
 user.put("PhoneNum",PhoneNum);  
  
 *//add user to database* docRef.set(user).addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 Toast.*makeText*(Phone.this, "Profile Created", Toast.*LENGTH\_SHORT*).show();  
 Log.*d*("zxcvbnm", "onSuccess: User Profile Created." + fAuth.getCurrentUser().getUid());  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Log.*d*("zxcvbnm", "onFailure: Failed to Create User " + e.toString());  
 }  
 });  
 Intent = new Intent(Phone.this, Register.class);  
 intent.putExtra("Phone\_key", PhoneNum);  
 startActivity(intent);  
 startActivity(new Intent(getApplicationContext(),Register.class));  
 finish();  
 }  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Toast.*makeText*(Phone.this, "Profile failure"+ e.getMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
}

package com.example.myapp;  
import androidx.annotation.NonNull;  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import com.facebook.CallbackManager;  
import com.facebook.login.widget.LoginButton;  
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;  
import com.google.android.gms.tasks.OnFailureListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.firestore.DocumentReference;  
import com.google.firebase.firestore.FirebaseFirestore;  
  
import java.util.Arrays;  
import java.util.HashMap;  
import java.util.Map;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class Register extends AppCompatActivity {  
 private final static int *RC\_SIGN\_IN* = 123;  
 private static final String *TAG* = "fb";  
 EditText mFullName, mEmail, mPassword, mconfPassword, mPhone;  
 Button mRegisterBtn, mSignUpBtn;  
 TextView mloginpageBtn;  
 ProgressBar;  
 FirebaseAuth fAuth;  
 FirebaseFirestore fStore;  
 String userID, PhoneNum, email, password, confPass, fullName;  
 GoogleSignInOptions gso;CallbackManager mCallbackManager;  
 LoginButton;  
 Boolean isDataValid = false;  
 int count = 0;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_register*);  
 mFullName = findViewById(R.id.*PersonName*);  
 mPhone = findViewById(R.id.*Phone*);  
 mEmail = findViewById(R.id.*Email*);  
 mPassword = findViewById(R.id.*Password*);  
 mconfPassword = findViewById(R.id.*confPassword*);  
 mRegisterBtn = findViewById(R.id.*registerBtn*);  
 mloginpageBtn = findViewById(R.id.*textView3*);  
 progressBar = findViewById(R.id.*progressBar*);  
 fAuth = FirebaseAuth.*getInstance*();  
 fStore = FirebaseFirestore.*getInstance*();  
 *//mSignUpBtn = findViewById(R.id.SignUpBtn);* Bundle = getIntent().getExtras();  
 PhoneNum = bundle.getString("Phone\_key");  
 *//Creating the User* mRegisterBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 email = mEmail.getText().toString().trim();  
 password = mPassword.getText().toString().trim();  
 fullName = mFullName.getText().toString();  
 confPass = mconfPassword.getText().toString().trim();  
  
 *//Validaing Input Data* do {  
 count = 0;  
 validateData(mFullName);  
 validateData(mEmail);  
 validateData(mPassword);  
 validateData(mconfPassword);  
 isValidPassword(password);  
  
 if (!(password.equals(confPass))) {  
 isDataValid = false;  
 mconfPassword.setError("Password does not Match");  
 } else {  
 count++;  
 }  
  
 if (mFullName.length() > 15) {  
 isDataValid = false;  
 mFullName.setError("UserName cannot be greater than 15 characters.");  
 } else {  
 count++;  
 }  
  
 if (count == 7) {  
 isDataValid = true;  
 }  
  
 } while (isDataValid = false);  
  
  
 if (TextUtils.*isEmpty*(email)) {  
 mEmail.setError("Email is Required");  
 return;  
 }  
  
 *// Registering the user in Firebase* if (count == 7) {  
 progressBar.setVisibility(View.*VISIBLE*);  
  
 fAuth.createUserWithEmailAndPassword(email, password).addOnSuccessListener(new OnSuccessListener<AuthResult>() {  
 @Override  
 public void onSuccess(AuthResult authResult) {  
  
 *//Verification Link* FirebaseUser fUser = fAuth.getCurrentUser();  
 fUser.sendEmailVerification().addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 Toast.*makeText*(Register.this, "Verification Email has been Sent", Toast.*LENGTH\_SHORT*).show();  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Toast.*makeText*(Register.this, "Error" + e.getMessage(), Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 Toast.*makeText*(Register.this, "User Account Created", Toast.*LENGTH\_SHORT*).show();  
  
 *//Storing data in firestore Database* userID = fAuth.getCurrentUser().getUid();  
 DocumentReference = fStore.collection("users").document(userID);  
 Map<String, Object> user = new HashMap<>();  
 user.put("FullName", fullName);  
 user.put("Email", email);  
 user.put("Phone", PhoneNum);  
 documentReference.set(user).addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 Log.*d*("TAG", "onSuccess: User Profile is Created" + userID);  
  
 }  
 });  
 startActivity(new Intent(Register.this, MainActivity.class));  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Toast.*makeText*(Register.this, "Error ! " + e.getMessage(), Toast.*LENGTH\_SHORT*).show();  
 progressBar.setVisibility(View.*GONE*);  
 }  
 });  
 }  
 }  
 });  
  
 }  
 public void validateData(EditText field) {  
 if (field.getText().toString().isEmpty()) {  
 isDataValid = false;  
 field.setError("Required Field.");  
 } else {  
 count++;  
 }  
 }  
  
 public void isValidPassword(String Password) {  
  
 *// Regex to check valid password.* String regex = "^(?=.\*[0-9])"  
 + "(?=.\*[a-z])(?=.\*[A-Z])"  
 + "(?=.\*[@#$%^&+=])"  
 + "(?=\\S+$).{8,20}$";  
  
 *// Compile the ReGex* Pattern p = Pattern.*compile*(regex);  
  
 Matcher m = p.matcher(Password);  
 if (m.matches() == true) {  
 count++;  
 } else {  
 mPassword.setError("Password should contain at least 8 characters and at most 20 characters.\n" +  
 "It should at least one digit.\n" +  
 "It should contain at least one upper case alphabet.\n" +  
 "It should contain at least one lower case alphabet.\n" +  
 "It should contain at least one special character which includes !@#$%&\*()-+=^.\n" +  
 "It should not contain any white space.");  
 }  
  
 }  
}

public void logout(View view){  
 FirebaseAuth.*getInstance*().signOut();  
 GoogleSignIn.*getClient*(this,new GoogleSignInOptions.Builder(GoogleSignInOptions.*DEFAULT\_GAMES\_SIGN\_IN*).build()).signOut().addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void aVoid) {  
 startActivity(new Intent(getApplicationContext(),Login.class));  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Toast.*makeText*(MainActivity.this, "Signout Failed", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 }  
  
 private void updateUI(FirebaseUser user){  
 if(user != null){  
 mTextUser.setText(user.getEmail());  
 if(user.getPhotoUrl() != null){  
 String photoUrl = user.getPhotoUrl().toString();  
 photoUrl = photoUrl + "?type=large";  
 Picasso.*get*().load(photoUrl).into(mPic);  
 }  
 }  
 else{  
 mTextUser.setText(" ");  
 }  
 }  
}

package com.example.myapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuInflater;  
import android.view.MenuItem;  
import android.widget.EditText;  
import android.widget.Spinner;  
import android.widget.Toast;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.firestore.CollectionReference;  
import com.google.firebase.firestore.FirebaseFirestore;  
  
public class MaintenanceNew extends AppCompatActivity {  
 private EditText editTextActivity;  
 private EditText editTextPhoneNo;  
 private EditText editTextStatus;  
 FirebaseAuth fAuth=FirebaseAuth.*getInstance*();  
 FirebaseUser user;  
 String userId;  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_maintenance\_new*);  
 user = fAuth.getCurrentUser();  
 userId=user.getUid();  
 getSupportActionBar().setHomeAsUpIndicator(R.drawable.*ic\_close*);  
 setTitle("Add Home Maintenance Activity");  
 setContentView(R.layout.*activity\_maintenance\_new*);  
 editTextActivity=findViewById(R.id.*newActivity*);  
 editTextPhoneNo=findViewById(R.id.*newPhoneNo*);  
 editTextStatus=findViewById(R.id.*newStatus*);  
 }  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 MenuInflater menuInflater=getMenuInflater();  
 menuInflater.inflate(R.menu.*new\_grocery\_menu*,menu);  
 return super.onCreateOptionsMenu(menu);  
 }  
 @Override  
 public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
 switch(item.getItemId()){  
 case R.id.*save\_grocery*:  
 saveActivity();  
 return true;  
 default: return super.onOptionsItemSelected(item);  
 }  
  
 }  
  
 private void saveActivity() {  
 String activity = editTextActivity.getText().toString();  
 String phone =editTextPhoneNo.getText().toString();  
 String status=editTextStatus.getText().toString();  
 if (activity.trim().isEmpty()) {  
 Toast.*makeText*(this, "Please insert a Home Maintenance Activity.", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 if (status.trim().isEmpty()){  
 Toast.*makeText*(this, "Please insert a valid status", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 CollectionReference maintenanceRef = FirebaseFirestore.*getInstance*().collection("users").document(userId)  
 .collection("Home Maintenance");  
 maintenanceRef.add(new HomeMaintenance(activity,phone,status));  
 Toast.*makeText*(this, "Home Maintenance Activity added", Toast.*LENGTH\_SHORT*).show();  
 finish();  
 }  
}

package com.example.myapp;  
  
import androidx.annotation.Nullable;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.recyclerview.widget.RecyclerView;  
  
import android.app.ProgressDialog;  
import android.content.Intent;

private ImageButton addItemBtn;  
 private EditText searchitemEt;  
 private RecyclerView itemsRv;  
 FirebaseUser user;  
 String userid;  
 private FirebaseAuth fAuth;  
 private ProgressDialog;  
 private ArrayList<modelitem> medicalreportslist;  
 private AdapterMedicalreport;  
 private String dashboard\_category;  
 private static final String *TAG* = "DocSnippets";  
 @Override  
 protected void onStart() {  
 super.onStart();  
 loadallitems();  
 }  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_medicalreport*);  
  
 addItemBtn = findViewById(R.id.*addItemBtn*);  
 searchitemEt = findViewById(R.id.*searchitemEt*);  
 itemsRv = findViewById(R.id.*itemsRv*);  
 fAuth = FirebaseAuth.*getInstance*();  
 user = fAuth.getCurrentUser();  
 userid=user.getUid();  
 searchitemEt.addTextChangedListener(new TextWatcher() {  
 @Override  
 public void beforeTextChanged(CharSequence s, int start, int count, int after) {  
  
  
 }  
  
 @Override  
 public void onTextChanged(CharSequence s, int start, int before, int count) {  
  
 try {  
 adapterMedicalreport.getFilter().filter(s);  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
  
 @Override  
 public void afterTextChanged(Editable s) {  
  
  
 }  
 });  
 addItemBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent = new Intent(MedicalreportActivity.this, addMedicalReportActivity.class);  
 startActivity(intent);  
 }  
 });  
 }  
 private void loadallitems() {  
 medicalreportslist=new ArrayList<>();  
 FirebaseFirestore db = FirebaseFirestore.*getInstance*();  
 db.collection("users").document(userid)  
 .collection("Medical report")  
  
 .addSnapshotListener(new EventListener<QuerySnapshot>() {  
 @Override  
 public void onEvent(@Nullable QuerySnapshot value, @Nullable FirebaseFirestoreException error) {  
 if(error!=null){  
 Log.*w*(*TAG*,"listen failed",error);  
 }  
 medicalreportslist.clear();;  
  
 for (QueryDocumentSnapshot document : value) {  
 Log.*d*(*TAG*, document.getId() + " => " + document.getData());  
 modelitem modelitemTemp = document.toObject(modelitem.class);  
 medicalreportslist.add(modelitemTemp);  
 }  
 *//setup adapter* adapterMedicalreport = new AdapterMedicalreport(MedicalreportActivity.this,medicalreportslist);  
 *//set adapter* itemsRv.setAdapter(adapterMedicalreport);  
 }  
 });  
 }  
}

package com.example.myapp;  
public class Grocery {  
 private String grocery\_Item;  
 private String grocery\_desc;  
 private int present\_Quantity;  
 private int buy\_Quantity;  
 private String itemIgnoreCase;  
  
 public Grocery() {  
  
 }  
 public Grocery(String grocery\_Item, String grocery\_desc, int present\_Quantity, int buy\_Quantity) {  
 this.grocery\_Item = grocery\_Item;  
 this.grocery\_desc = grocery\_desc;  
 this.present\_Quantity = present\_Quantity;  
 this.buy\_Quantity = buy\_Quantity;  
 this.itemIgnoreCase=grocery\_Item.toLowerCase();  
 }  
 public String getGrocery\_Item() {  
 return grocery\_Item;  
 }  
  
 public String getGrocery\_desc() {  
 return grocery\_desc;  
 }  
  
 public int getPresent\_Quantity() {  
 return present\_Quantity;  
 }  
  
 public int getBuy\_Quantity() {  
 return buy\_Quantity;  
 }  
   
 public String getItemIgnoreCase() {  
 return itemIgnoreCase;  
 }  
}

package com.example.myapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuInflater;  
import android.view.MenuItem;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.ArrayAdapter;  
import android.widget.EditText;  
import android.widget.Spinner;  
import android.widget.Toast;  
  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
import com.google.firebase.firestore.CollectionReference;  
import com.google.firebase.firestore.FirebaseFirestore;  
  
public class NewBookActivity extends AppCompatActivity implements AdapterView.OnItemSelectedListener {  
 private EditText editTextBookName;  
 private EditText editTextAuthorName;  
 private EditText editTextBookDesc;  
 private Spinner spinnerBookStatus;  
 FirebaseAuth fAuth=FirebaseAuth.*getInstance*();  
 FirebaseUser user;  
 String userId;  
 private String text;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_new\_book*);  
 user = fAuth.getCurrentUser();  
 userId=user.getUid();  
 getSupportActionBar().setHomeAsUpIndicator(R.drawable.*ic\_close*);  
 setTitle("Add Book");  
 setContentView(R.layout.*activity\_new\_book*);  
 editTextBookName=findViewById(R.id.*newBookName*);  
 editTextAuthorName=findViewById(R.id.*newAuthorName*);  
 editTextBookDesc=findViewById(R.id.*newBookDesc*);  
 spinnerBookStatus=findViewById(R.id.*newBookStatus*);  
 ArrayAdapter<CharSequence> arrayAdapter = ArrayAdapter.*createFromResource*(this,  
 R.array.*BookStatus*, android.R.layout.*simple\_spinner\_item*);  
 arrayAdapter.setDropDownViewResource(android.R.layout.*simple\_spinner\_dropdown\_item*);  
 spinnerBookStatus.setAdapter(arrayAdapter);  
 spinnerBookStatus.setOnItemSelectedListener(NewBookActivity.this);  
 }  
  
 @Override  
 public void onItemSelected(AdapterView<?> parent, View, int position, long id) {  
 text = parent.getItemAtPosition(position).toString();  
 }  
  
 @Override  
 public void onNothingSelected(AdapterView<?> parent) {  
  
 }  
 @Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 MenuInflater menuInflater=getMenuInflater();  
 menuInflater.inflate(R.menu.*new\_grocery\_menu*,menu);  
 return super.onCreateOptionsMenu(menu);  
 }  
 @Override  
 public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
 switch(item.getItemId()){  
 case R.id.*save\_grocery*:  
 saveBook();  
 return true;  
 default: return super.onOptionsItemSelected(item);  
 }  
  
 }  
 private void saveBook() {  
 String bookName = editTextBookName.getText().toString();  
 String authorName =editTextAuthorName.getText().toString();  
 String bookDesc=editTextBookDesc.getText().toString();  
 if (bookName.trim().isEmpty()) {  
 Toast.*makeText*(this, "Please insert a Book Name.", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 if (authorName.trim().isEmpty()){  
 Toast.*makeText*(this, "Please insert an Author Name.", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 CollectionReference bookRef = FirebaseFirestore.*getInstance*().collection("users").document(userId)  
 .collection("Books");  
 bookRef.add(new Book(bookName, authorName, bookDesc,text));  
 Toast.*makeText*(this, "Book added", Toast.*LENGTH\_SHORT*).show();  
 finish();  
 }  
}

package com.example.myapp;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
import android.app.AlarmManager;  
import android.app.DatePickerDialog;  
import android.app.PendingIntent;  
import android.app.TimePickerDialog;  
  
import com.allyants.notifyme.NotifyMe;  
import com.google.android.gms.tasks.OnFailureListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.firebase.Timestamp;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.firestore.CollectionReference;  
import com.google.firebase.firestore.DocumentReference;  
import com.google.firebase.firestore.FirebaseFirestore;  
  
import java.util.Calendar;  
import java.util.Date;  
  
public class NewMeetActivity extends AppCompatActivity {  
 EditText titleEditText, docsEditText, agendaEditText;  
 Button setMeetDate, setMeetTime, saveMeet;  
 TextView dateTextView, timeTextView;  
 DatePickerDialog;  
  
 String dateStr = "dd/mm/yy", timeStr = "hh:mm";  
 Calendar dateTime;  
 int yearSave, monthSave, daySave, hourSave, minuteSave;  
   
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_new\_meet*);  
  
 titleEditText = (EditText) findViewById(R.id.*edit\_text\_meet\_title*);  
  
 docsEditText = (EditText) findViewById(R.id.*edit\_text\_meet\_docs*);  
 agendaEditText = (EditText) findViewById(R.id.*edit\_text\_meet\_agenda*);  
  
 dateTextView = (TextView) findViewById(R.id.*text\_view\_meet\_date*);  
 timeTextView = (TextView) findViewById(R.id.*text\_view\_meet\_time*);  
  
 setMeetDate = (Button) findViewById(R.id.*button\_set\_meet\_date*);  
 setMeetDate.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 handleDateButton();  
 }  
 });  
 setMeetTime = (Button) findViewById(R.id.*button\_set\_meet\_time*);  
 setMeetTime.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 handleTimeButton();  
 }  
 });  
  
 saveMeet = (Button) findViewById(R.id.*save\_meet*);  
 saveMeet.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 saveMeet();  
 }  
 });  
  
 }  
  
 public void handleDateButton() {  
 Calendar = Calendar.*getInstance*();  
 int YEAR = calendar.get(Calendar.*YEAR*);  
 int MONTH = calendar.get(Calendar.*MONTH*);  
 int DATE = calendar.get(Calendar.*DATE*);  
  
 datePickerDialog = new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() {  
 @Override  
 public void onDateSet(DatePicker view, int YEAR, int MONTH, int DATE) {  
 dateStr = DATE + "/" + (MONTH+1) + "/" + YEAR;  
 dateTextView.setText(dateStr);  
 calendar.set(YEAR, MONTH, DATE);  
 Intent = new Intent(NewMeetActivity.this, MeetingBroadcastReceiver.class);  
  
 intent.putExtra("time",titleEditText.getText().toString());  
 yearSave = YEAR;  
 monthSave = MONTH;  
 daySave = DATE;  
 }  
 }, YEAR, MONTH, DATE);  
 datePickerDialog.show();  
  
 }  
  
 public void handleTimeButton() {  
 Calendar = Calendar.*getInstance*();  
 int HOUR = calendar.get(Calendar.*HOUR*);  
 int MINUTE = calendar.get(Calendar.*MINUTE*);  
  
 TimePickerDialog = new TimePickerDialog(this, new TimePickerDialog.OnTimeSetListener() {  
 @Override  
 public void onTimeSet(TimePicker view, int HOUR, int MINUTE) {  
 timeStr = String.*format*("%02d", HOUR) + ":" + String.*format*("%02d", MINUTE);  
 timeTextView.setText(timeStr);  
 hourSave = HOUR;  
 minuteSave =MINUTE;  
 }  
 }, HOUR, MINUTE, false);  
 timePickerDialog.show();  
 }  
  
 public void saveMeet() {  
 String title = titleEditText.getText().toString();  
 String agenda = agendaEditText.getText().toString();  
 String docs = docsEditText.getText().toString();  
 Calendar finalDateTime = Calendar.*getInstance*();  
  
 finalDateTime.set(Calendar.*DATE*,daySave);  
 finalDateTime .set(Calendar.*HOUR\_OF\_DAY*,hourSave);  
 finalDateTime.set(Calendar.*MINUTE*,minuteSave);  
 finalDateTime.set(Calendar.*SECOND*,0);  
 long alarmStartTime=finalDateTime.getTimeInMillis();  
 Timestamp meetDateTime = new Timestamp((long)finalDateTime.getTimeInMillis()/1000, 0);  
 long ms = meetDateTime.getSeconds();  
  
 if (title.trim().isEmpty() || agenda.trim().isEmpty()) {  
 Toast.*makeText*(this, "Please insert a title and agenda", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 FirebaseAuth fAuth = FirebaseAuth.*getInstance*();  
  
 CollectionReference meetingsRef = FirebaseFirestore.*getInstance*()  
 .collection("users").document(fAuth.getCurrentUser().getUid()).collection("Meetings");  
 meetingsRef.add(new Meeting(title, agenda, docs, meetDateTime, ms */\* dateTime, dateStr, timeStr\*/*))  
 .addOnSuccessListener(new OnSuccessListener<DocumentReference>() {  
 @Override  
 public void onSuccess(DocumentReference documentReference) {  
 Toast.*makeText*(NewMeetActivity.this, "firestore success", Toast.*LENGTH\_SHORT*).show();  
 }  
 }).addOnFailureListener(new OnFailureListener() {  
 @Override  
 public void onFailure(@NonNull Exception e) {  
 Toast.*makeText*(NewMeetActivity.this, "Error", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
 Toast.*makeText*(this, "Meeting added", Toast.*LENGTH\_SHORT*).show();  
  
 Calendar notifCalendar = Calendar.*getInstance*();  
 notifCalendar.set(yearSave, monthSave, daySave, hourSave , minuteSave);  
  
  
 Intent = new Intent(NewMeetActivity.this, MeetingBroadcastReceiver.class);  
 intent.putExtra("time",titleEditText.getText().toString());  
  
 PendingIntent = PendingIntent.*getBroadcast*(  
 NewMeetActivity.this, 0, intent, 0);  
  
  
 AlarmManager = (AlarmManager) getSystemService(Context.*ALARM\_SERVICE*);  
 alarmManager.set(AlarmManager.*RTC\_WAKEUP*, notifCalendar.getTimeInMillis(), pendingIntent);  
 }  
  
}

package com.example.myapp;  
import com.google.firebase.Timestamp;  
public class Meeting {  
 private String title;  
 private String agenda;  
 private String docs;  
 Timestamp meetTimeStamp;  
 long seconds = 0L;  
  
 public Meeting() {  
  
 }  
 public Meeting(String title, String agenda, String docs, Timestamp meetTimeStamp, long seconds*/\*String date, String time\*//\*Date meetDateTime\*/*) {  
 this.title = title;  
 this.agenda = agenda;  
 this.docs = docs;  
 this.meetTimeStamp = meetTimeStamp;  
 this.seconds = seconds;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
 public String getAgenda() {  
 return agenda;  
 }  
  
 public String getDocs() {  
 return docs;  
 }  
  
 String TimeToString() {  
 String timeStr = String.*format*("%02d", meetTimeStamp.toDate().getHours()) + ":" + String.*format*("%02d", meetTimeStamp.toDate().getMinutes());  
 return timeStr;  
 }  
  
 String DateToString() {  
 String dateStr = meetTimeStamp.toDate().getDate() + "/" + meetTimeStamp.toDate().getMonth() + "/" + String.*format*("%04d", meetTimeStamp.toDate().getYear());  
 return dateStr;  
 }  
  
 public Timestamp getMeetTimeStamp() {  
 return meetTimeStamp;  
 }  
  
}

**5.3) PERFORMANCE ANALYSIS :-**

* Data entered in the app is stored in the firebase database without any redundancy and latency.
* Notifications and remainders are successfully sent in the scheduled time.
* User can access the app from any mobile.
* Secure authentication and password reset option is available to the user.
* Password reset email is sent to user’s registered email address without delay when the password is forgotten.

**6) CONCLUSION**

**6.1) CONCLUSION :-**

By using this app there is no need of remembering all the things that to be done in daily life as our app reminds you often**. Automated reminders can help you build habits** but it can also help you remember things that are too importantto be trusted even to habit. This helps people to decrease the work load so that they can concentrate on other required things.

**6.2) FUTURE WORK :-**

* Predicting grocery lists of user using machine learning and recommending the healthy options for groceries.
* Scanning the medical prescription and intimating the user about the what medicine should be taken at what time.

**6.3) REFERENCES :-**

* [**https://firebase.google.com/**](https://firebase.google.com/)
* [**https://developer.android.com/**](https://developer.android.com/)
* [**https://cloud.smartdraw.com/**](https://cloud.smartdraw.com/)
* [**https://developers.google.com/android-publisher**](https://developers.google.com/android-publisher)
* [**https://en.wikipedia.org/wiki/Notification\_service**](https://en.wikipedia.org/wiki/Notification_service)
* **https://console.firebase.google.com/u/0/project/myapp-bd594/overview**