Министерство образования Республики Беларусь

Учреждение образования

“БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ”

Факультет компьютерного проектирования

Дисциплина: Компьютерные сети

**ЛАБОРАТОРНАЯ РАБОТА № 3**

Android - квиз

Выполнил:

студент гр. 714301

Гельдымурадов С.

Проверил:

Минск 2020

Используется язык Java 1.8.

Подключены firebase-auth firebase-realtime-database admob, а так же открытая Rest Api для получения вопросов и варианты ответов. Для обращению к серверу используется retrofit.

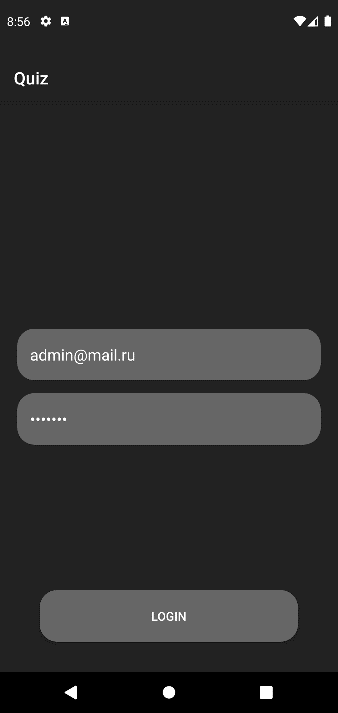


Рисунок 1.Главное окно

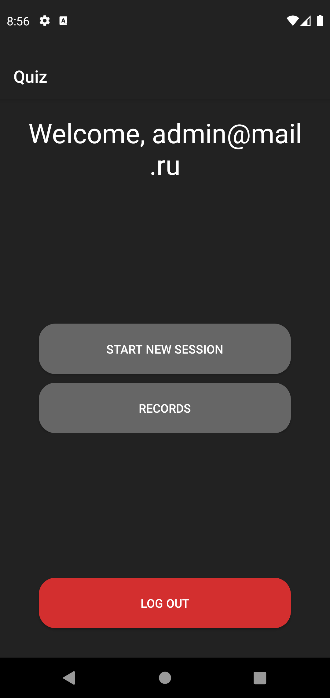


Рисунок 2. Окно после авторизации.

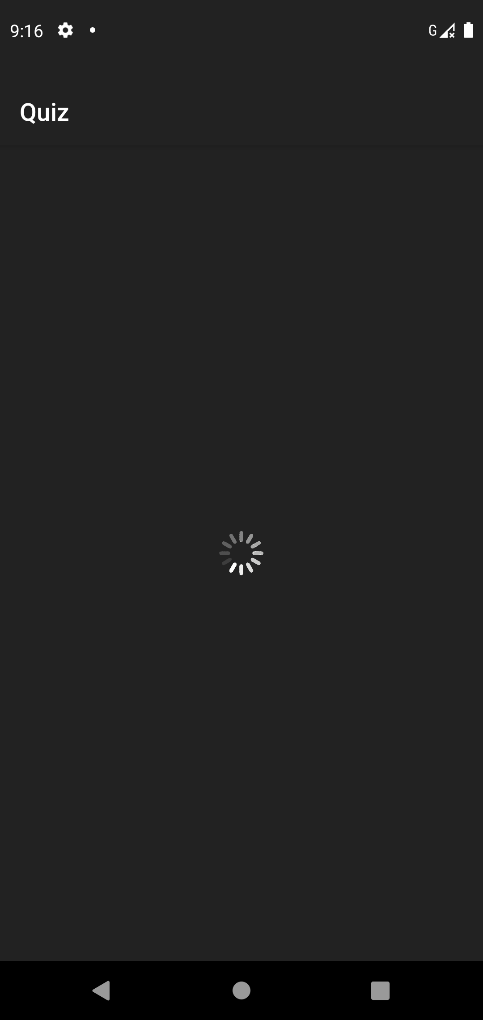


Рисунок 3. Анимированное окно загрузки.

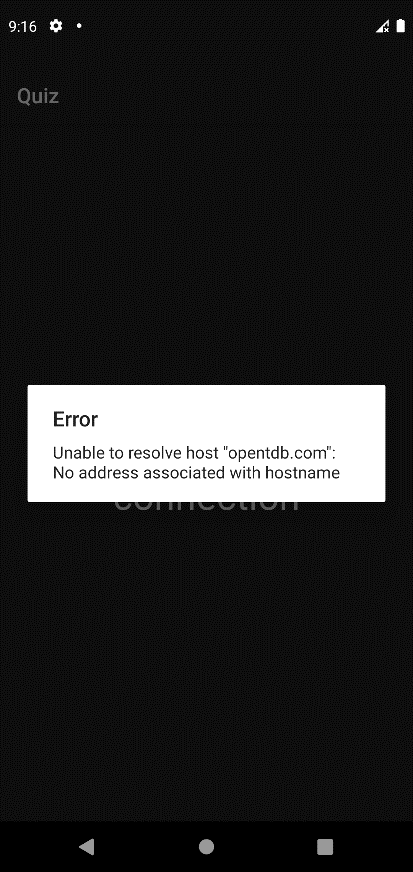


Рисунок 4. Окно – сообщение об проблеме подключения к серверу.

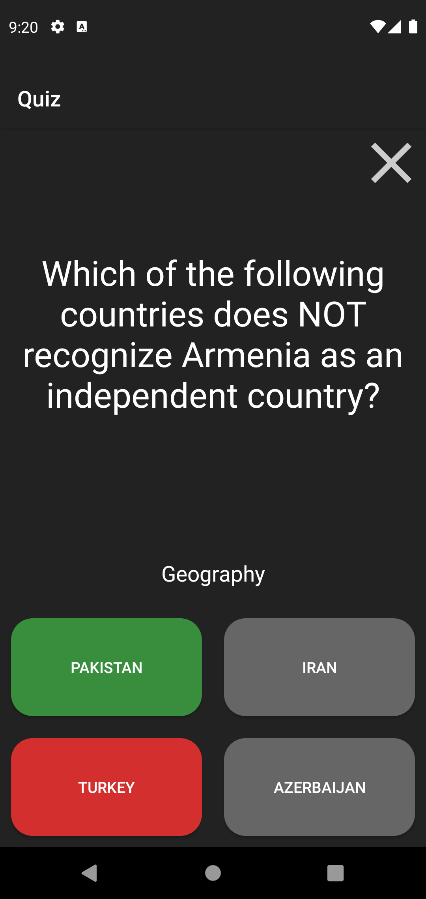


Рисунок 5. Вопросы.

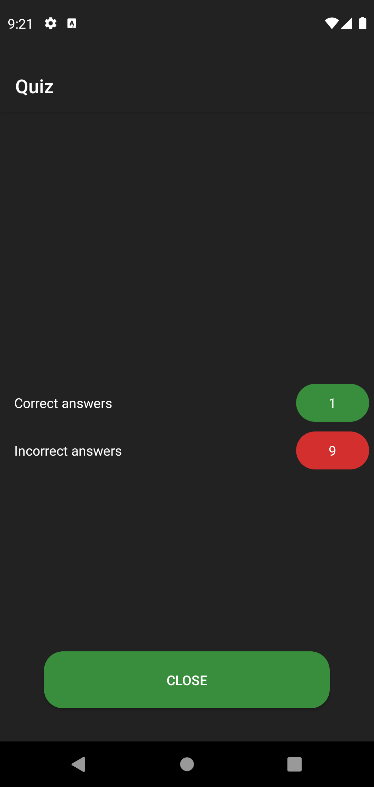


Рисунок 6. Результаты теста.

**Листинг кода.**

**https://github.com/bsuir-temp/ol3**

**public class MainActivity extends AppCompatActivity {**

**public final static String TAG = "MainActivity";**

**@Override**

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

**super.onCreate(savedInstanceState);**

**setContentView(R.layout.activity\_main);**

**FirebaseApp.initializeApp(getApplicationContext());**

**getSupportFragmentManager().beginTransaction().replace(R.id.content, new LoadingFragment()).commit();**

**/\***

**FirebaseDatabase database = FirebaseDatabase.getInstance();**

**DatabaseReference ref = database.getReference();**

**User user = new User();**

**user.email="lelelel";**

**user.loses=0;**

**user.wins=0;**

**addUser(user,ref);**

**ref.child(users).addListenerForSingleValueEvent(new ValueEventListener() {**

**@Override public void onDataChange(@NonNull DataSnapshot snapshot) { readTop((Map<String,Object>) snapshot.getValue()); }**

**@Override public void onCancelled(@NonNull DatabaseError error) { }**

**});**

**\*/**

**}**

**}**

**public class LoadingView extends androidx.appcompat.widget.AppCompatImageView {**

**Runnable mUpdateViewRunnable = null;**

**float mRotateDegrees = 0;**

**boolean mNeedToUpdateView = false;**

**long mFrameTime = 0;**

**public LoadingView(Context context) {**

**super(context);**

**init();**

**}**

**public LoadingView(Context context, @Nullable AttributeSet attrs) {**

**super(context, attrs);**

**init();**

**}**

**public LoadingView(Context context, @Nullable AttributeSet attrs, int defStyleAttr) {**

**super(context, attrs, defStyleAttr);**

**init();**

**}**

**public void setAnimationSpeed(float scale) {**

**mFrameTime = (long)(1000 / 12 / scale);**

**}**

**@Override**

**protected void onDetachedFromWindow() {**

**super.onDetachedFromWindow();**

**mNeedToUpdateView = false;**

**}**

**@Override**

**protected void onAttachedToWindow() {**

**super.onAttachedToWindow();**

**mNeedToUpdateView = true;**

**post(mUpdateViewRunnable);**

**}**

**public void init(){**

**this.setImageResource(R.drawable.kprogresshud\_spinner\_light);**

**mFrameTime = 1000 / 12;**

**mUpdateViewRunnable = new Runnable() {**

**@Override**

**public void run() {**

**mRotateDegrees += 30f;**

**mRotateDegrees = mRotateDegrees < 360 ? mRotateDegrees : mRotateDegrees - 360;**

**invalidate();**

**if (mNeedToUpdateView) {**

**postDelayed(this, mFrameTime);**

**}**

**}**

**};**

**}**

**@Override**

**protected void onDraw(Canvas canvas) {**

**canvas.rotate(mRotateDegrees, getWidth() / 2f, getHeight() / 2f);**

**super.onDraw(canvas);**

**}**

**}**

**public interface RetroPoints {**

**@GET("/api.php?amount=10&type=multiple&encode=url3986")**

**Call<Wrapper> getQuestions();**

**}**

**public class RetrofitCore {**

**private static Retrofit retrofit;**

**private static final String ENDPOINT = "https://opentdb.com/";**

**public static Retrofit getRetrofitInstance() {**

**if (retrofit == null) {**

**retrofit = new retrofit2.Retrofit.Builder()**

**.baseUrl(ENDPOINT)**

**.addConverterFactory(GsonConverterFactory.create())**

**.build();**

**}**

**return retrofit;**

**}**

**}**

**public class Wrapper {**

**@SerializedName("results")**

**public List<Question> results;**

**}**

**public class User {**

**public String email;**

**public String password;**

**public long wins = 0;**

**public long loses = 0;**

**@Override**

**public String toString()**

**{**

**return "email ["+email+"] wins ["+wins+"] loses ["+loses+"]";**

**}**

**}**

**public class Question {**

**@SerializedName("category")**

**public String category;**

**@SerializedName("difficulty")**

**public String difficulty;**

**@SerializedName("question")**

**public String question;**

**@SerializedName("correct\_answer")**

**public String correct\_answer;**

**@SerializedName("incorrect\_answers")**

**public String[] incorrect\_answers;**

**}**

**public class ResultsFragment extends FragmentEvent {**

**public static final String TAG = "ResultsFragment";**

**@Override**

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

**super.onCreate(savedInstanceState);**

**}**

**@Nullable**

**@Override**

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

**View root = inflater.inflate(R.layout.fragment\_results, null);**

**((TextView)root.findViewById(R.id.correct)).setText(String.valueOf(getArguments().getInt("correct")));**

**((TextView)root.findViewById(R.id.incorrect)).setText(String.valueOf(getArguments().getInt("incorrect")));**

**final FirebaseAuthRepo repo = new FirebaseAuthRepo();**

**root.findViewById(R.id.close\_btn).setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**getFragmentManager().beginTransaction().replace(R.id.content,new MainMenuFragment()).commit();**

**}**

**});**

**return root;**

**}**

**}**

**public class QuestionFragment extends FragmentEvent {**

**public static final String TAG = "QuestionFragment";**

**private int session\_index = 0;**

**public List<Question> questions = new ArrayList();**

**private View[] items = new View[9];**

**private Random random = new Random();**

**private int correctAnswer = 0;**

**private int correctAnswers = 0;**

**private int incorrectAnswers = 0;**

**private boolean lock = false;**

**private View.OnClickListener closeQuize = new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**Fragment fragment = new MainMenuFragment();**

**getFragmentManager().beginTransaction().replace(R.id.content,fragment).commit();**

**}**

**};**

**private View.OnClickListener submitAnswer = new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**if(lock)return;**

**items[2+correctAnswer].setBackgroundResource(R.drawable.button\_background\_success);**

**switch (v.getId())**

**{**

**case R.id.answer\_1:**

**if(0!=correctAnswer) {**

**v.setBackgroundResource(R.drawable.button\_background\_danger);**

**incorrectAnswers++;**

**}else correctAnswers++;**

**break;**

**case R.id.answer\_2:**

**if(1!=correctAnswer)**

**{**

**v.setBackgroundResource(R.drawable.button\_background\_danger);**

**incorrectAnswers++;**

**}else correctAnswers++;**

**break;**

**case R.id.answer\_3:**

**if(2!=correctAnswer)**

**{**

**v.setBackgroundResource(R.drawable.button\_background\_danger);**

**incorrectAnswers++;**

**}else correctAnswers++;**

**break;**

**case R.id.answer\_4:**

**if(3!=correctAnswer)**

**{**

**v.setBackgroundResource(R.drawable.button\_background\_danger);**

**incorrectAnswers++;**

**}else correctAnswers++;**

**break;**

**}**

**new Thread(() -> {**

**lock = true;**

**try { Thread.sleep(800); } catch (InterruptedException unimportant) { }**

**getActivity().runOnUiThread(() -> {**

**onNext();**

**lock = false;**

**});**

**}).start();**

**}**

**};**

**@Override**

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

**super.onCreate(savedInstanceState);**

**RetroPoints service = RetrofitCore.getRetrofitInstance().create(RetroPoints.class);**

**Call<Wrapper> call = service.getQuestions();**

**call.enqueue(new Callback<Wrapper>() {**

**@Override**

**public void onResponse(Call<Wrapper> call, Response<Wrapper> response) {**

**questions = response.body().results;**

**items[6].setVisibility(View.GONE);**

**items[7].setVisibility(View.VISIBLE);**

**onNext();**

**}**

**@Override**

**public void onFailure(Call<Wrapper> call, Throwable t) {**

**DisconnectedFragment fragment = new DisconnectedFragment();**

**getFragmentManager().beginTransaction().replace(R.id.content,fragment,TAG).commit();**

**onError(t.getLocalizedMessage());**

**}**

**});**

**}**

**@Nullable**

**@Override**

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

**View view = inflater.inflate(R.layout.fragment\_question, null);**

**items[0] = view.findViewById(R.id.question);**

**items[1] = view.findViewById(R.id.category);**

**items[2] = view.findViewById(R.id.answer\_1);**

**items[3] = view.findViewById(R.id.answer\_2);**

**items[4] = view.findViewById(R.id.answer\_3);**

**items[5] = view.findViewById(R.id.answer\_4);**

**for(int i=2;i<6;i++)items[i].setOnClickListener(submitAnswer);**

**items[6] = view.findViewById(R.id.loading\_frame);**

**items[7] = view.findViewById(R.id.question\_frame);**

**items[8] = view.findViewById(R.id.close\_btn);**

**items[8].setOnClickListener(closeQuize);**

**return view;**

**}**

**public void onNext() {**

**try {**

**if(session\_index==10)**

**{**

**Fragment fragment = new ResultsFragment();**

**Bundle bundle = new Bundle();**

**bundle.putInt("correct",correctAnswers);**

**bundle.putInt("incorrect",incorrectAnswers);**

**fragment.setArguments(bundle);**

**getFragmentManager().beginTransaction().replace(R.id.content,fragment).commit();**

**return;**

**}**

**Question question = questions.get(session\_index++);**

**correctAnswer = random.nextInt(4);**

**((TextView)items[0]).setText(java.net.URLDecoder.decode(question.question, StandardCharsets.UTF\_8.name()));**

**((TextView)items[1]).setText(java.net.URLDecoder.decode(question.category, StandardCharsets.UTF\_8.name()));**

**for(int i=2;i<6;i++)items[i].setBackgroundResource(R.drawable.button\_background\_default);**

**int i = 2;**

**for (String answer : question.incorrect\_answers) {**

**if (correctAnswer + 2 == i)**

**((TextView)items[i++]).setText(java.net.URLDecoder.decode(question.correct\_answer,StandardCharsets.UTF\_8.name()));**

**((TextView)items[i++]).setText(java.net.URLDecoder.decode(answer, StandardCharsets.UTF\_8.name()));**

**}**

**if (correctAnswer == 3)**

**((TextView)items[5]).setText(java.net.URLDecoder.decode(question.correct\_answer,StandardCharsets.UTF\_8.name()));**

**}catch (Exception ex){ex.printStackTrace();}**

**}**

**}**

**public class MainMenuFragment extends FragmentEvent {**

**public static final String TAG = "MainMenuFragment";**

**FirebaseAuthRepo authRepo = new FirebaseAuthRepo();**

**private View items[] = new View[4];**

**private View.OnClickListener startQ = new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**Fragment fragment = new QuestionFragment();**

**getFragmentManager().beginTransaction().replace(R.id.content,fragment).commit();**

**}**

**};**

**private View.OnClickListener logout = new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**authRepo.logout();**

**Fragment fragment = new LoginFragment();**

**getFragmentManager().beginTransaction().replace(R.id.content,fragment).commit();**

**}**

**};**

**@Override**

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

**super.onCreate(savedInstanceState);**

**}**

**@Nullable**

**@Override**

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

**View root = inflater.inflate(R.layout.fragment\_main, null);**

**items[0] = root.findViewById(R.id.welcome);**

**((TextView)items[0]).setText("Welcome, "+authRepo.mAuth.getCurrentUser().getEmail());**

**items[1] = root.findViewById(R.id.btn\_start);**

**items[1].setOnClickListener(startQ);**

**items[3] = root.findViewById(R.id.logout\_button);**

**items[3].setOnClickListener(logout);**

**return root;**

**}**

**}**

**public class LoginFragment extends FragmentEvent {**

**public static final String TAG = "LoginFragment";**

**private FirebaseAuthRepo auth;**

**private View items[] = new View[6];**

**private View.OnClickListener login = new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**User user = new User();**

**user.email = ((EditText)items[0]).getText().toString();**

**user.password = ((EditText)items[1]).getText().toString();**

**items[4].setVisibility(View.GONE);**

**items[5].setVisibility(View.VISIBLE);**

**try {**

**auth.signIn(getActivity(), user, task -> {**

**if (task.isSuccessful()) {**

**Fragment fragment = new MainMenuFragment();**

**getFragmentManager().beginTransaction().replace(R.id.content, fragment).commit();**

**} else**

**{**

**items[5].setVisibility(View.GONE);**

**items[4].setVisibility(View.VISIBLE);**

**onError("Failed login");**

**}**

**});**

**}catch (Throwable error){**

**items[5].setVisibility(View.GONE);**

**items[4].setVisibility(View.VISIBLE);**

**onError(error.getLocalizedMessage());**

**}**

**}**

**};**

**@Override**

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

**super.onCreate(savedInstanceState);**

**auth = new FirebaseAuthRepo();**

**}**

**@Nullable**

**@Override**

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

**View root = inflater.inflate(R.layout.fragment\_login, null);**

**items[0] = root.findViewById(R.id.field\_email);**

**items[1] = root.findViewById(R.id.field\_password);**

**items[2] = root.findViewById(R.id.login\_button);**

**items[2].setOnClickListener(login);**

**items[4] = root.findViewById(R.id.login\_frame);**

**items[5] = root.findViewById(R.id.loading\_frame);**

**return root;**

**}**

**}**

**public class LoadingFragment extends FragmentEvent {**

**public static final String TAG = "LoadingFragment";**

**private FirebaseAuthRepo authRepo;**

**@Override**

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

**super.onCreate(savedInstanceState);**

**authRepo = new FirebaseAuthRepo();**

**Fragment fragment = null;**

**if(!authRepo.isSignedIn())**

**fragment = new LoginFragment();**

**else**

**fragment = new MainMenuFragment();**

**getFragmentManager().beginTransaction().replace(R.id.content,fragment,TAG).commit();**

**}**

**@Nullable**

**@Override**

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

**return inflater.inflate(R.layout.fragment\_loading, null);**

**}**

**}**

**public abstract class FragmentEvent extends Fragment{**

**public void onError(String why)**

**{**

**new AlertDialog.Builder(getContext()).setTitle("Error").setMessage(why).create().show();**

**}**

**}**

**public class DisconnectedFragment extends FragmentEvent {**

**public static final String TAG = "DisconnectedFragment";**

**@Override**

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

**super.onCreate(savedInstanceState);**

**}**

**@Nullable**

**@Override**

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

**return inflater.inflate(R.layout.fragment\_disconnected, null);**

**}**

**}**

**public class FirebaseDataScores {**

**public final static String TAG = "FirebaseAuthRepo";**

**private FirebaseDatabase database = null;**

**private DatabaseReference ref;**

**public final static String users = "users";**

**public FirebaseDataScores()**

**{**

**database = FirebaseDatabase.getInstance();**

**ref = database.getReference();**

**}**

**public void addUser(User user)**

**{**

**Map<String, Object> userData = new HashMap<String, Object>();**

**userData.put("loses", user.loses);**

**userData.put("wins", user.wins);**

**ref = ref.child(users).child(user.email);**

**ref.setValue(userData);**

**}**

**public void readTop(final ArrayList<User> userArrayList, FragmentEvent fragmentEvent)**

**{**

**ref.child(users).addListenerForSingleValueEvent(new ValueEventListener() {**

**@Override public void onDataChange(@NonNull DataSnapshot snapshot) {**

**for (Map.Entry<String, Object> entry : ((Map<String,Object>) snapshot.getValue()).entrySet()){**

**Map singleUser = (Map) entry.getValue();**

**User user = new User();**

**user.email = entry.getKey();**

**user.loses = (long) singleUser.get("loses");**

**user.wins = (long) singleUser.get("wins");**

**userArrayList.add(user);**

**}**

**}**

**@Override public void onCancelled(@NonNull DatabaseError error) { fragmentEvent.onError(error.getMessage()); }**

**});**

**}**

**}**

**public class FirebaseAuthRepo {**

**public final static String TAG = "FirebaseAuthRepo";**

**public FirebaseAuth mAuth = null;**

**public FirebaseAuthRepo()**

**{**

**mAuth = FirebaseAuth.getInstance();**

**}**

**public boolean isSignedIn()**

**{**

**return mAuth.getCurrentUser()!=null;**

**}**

**public void registerUser(Activity activity, User user,OnCompleteListener<AuthResult> task)**

**{**

**mAuth.createUserWithEmailAndPassword(user.email, user.password).addOnCompleteListener(activity,task);**

**}**

**public void signIn(Activity activity, User user,OnCompleteListener<AuthResult> task)throws Throwable**

**{**

**mAuth.signInWithEmailAndPassword(user.email, user.password).addOnCompleteListener(activity,task);**

**}**

**public void logout()**

**{**

**mAuth.signOut();**

**}**

**}**