LE RAPPORT

Guide Développeur : -

C'est le MainActivity class

Il est responsable de l'animation au démarrage et amène l'utilisateur à la page de connexion

```java
public class MainActivity extends AppCompatActivity {

    // Widgets
    Button suivant;
    TextView t1,t2;
    Animation animate_btn,animate_txt, animate_txt2;

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

        suivant = findViewById(R.id.suivant);
        t1  = findViewById(R.id.logintextview);
        t2  = findViewById(R.id.registertextview);

        // Load the animation from the xml file
        animate_btn = AnimationUtils.loadAnimation( context: this, R.anim.animate_btn);
        animate_txt = AnimationUtils.loadAnimation( context: this, R.anim.animate_texts);
        animate_txt2 = AnimationUtils.loadAnimation( context: this, R.anim.animate_texts2);

        // Set the animation to the widgets
        suivant.setAnimation(animate_btn);
        t1.setAnimation(animate_txt);
        t2.setAnimation(animate_txt2);

        suivant.setOnClickListener(v -> {
            Intent i = new Intent(getApplicationContext(),LoginActivity.class);
            startActivity(i);
        });
    }
```

Cette classe est responsable de la connexion de l'utilisateur

```java
public class LoginActivity extends AppCompatActivity {

    TextView logintextview;
    Button toRegister,login;
    EditText email,password;
    Animation animate_btn,animate_txt, animate_txt2;

    //     Firebase Auth
    private FirebaseAuth firebaseAuth;
    private FirebaseAuth.AuthStateListener authStateListener;
    private FirebaseUser currentUser;

    //     FireBase Connection
    private final FirebaseFirestore db = FirebaseFirestore.getInstance();
    private final CollectionReference collectionReference = db.collection( collectionPath: "Users");
```

Il vérifie l'utilisateur avec l'authentification Firebase Database

```java
private void LoginEmailPasswordUser(String email, String password) {
    if(!email.isEmpty() && !password.isEmpty()){
        firebaseAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener(task -> {
            FirebaseUser user = firebaseAuth.getCurrentUser();
            assert user != null;
            final String currentUserId = user.getUid();

            collectionReference.whereEqualTo( field: "userId",currentUserId).addSnapshotListener((value, error) -> {
                if(error != null){
                    return;
                }
                assert value != null;
                if(!value.isEmpty()) {
    Getting all the data from the database
                    for (QueryDocumentSnapshot snapshot : value) {
                        JournalUser journalUser = JournalUser.getInstance();
                        journalUser.setUsername(snapshot.getString( field: "username"));
                        journalUser.setUserId(snapshot.getString( field: "userId"));

                          startActivity(new Intent(MainActivity.this, MyJournal.class));
                        startActivity(new Intent( packageContext: LoginActivity.this, JournalListAll.class));
                    }
                } else {
                    Toast.makeText( context: LoginActivity.this,  text: "No such user", Toast.LENGTH_SHORT).show();
                }
            });
        }).addOnFailureListener(e -> Toast.makeText( context: LoginActivity.this,  text: "Error: "+e.getMessage(), Toast.LENGTH_SHORT).show());
    } else {
        Toast.makeText( context: this,  text: "Please enter email and password", Toast.LENGTH_SHORT).show();
    }
```

Cette classe est responsable de l'enregistrement des utilisateurs dans la base de données Firestore Firebase

```
public class RegisterActivity extends AppCompatActivity {

    TextView registertextview;
    EditText password_create, email_create, username_create;
    Button register, backToLogin;
    Animation animate_btn,animate_txt, animate_txt2;

//  Firebase Auth
    private FirebaseAuth firebaseAuth = FirebaseAuth.getInstance();
    private FirebaseAuth.AuthStateListener authStateListener;
    private FirebaseUser currentUser;

//  FireBase Connection
    private final FirebaseFirestore db = FirebaseFirestore.getInstance();
    private final CollectionReference collectionReference = db.collection( collectionPath: "Users");

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

        registertextview = findViewById(R.id.registertextview);
        register = findViewById(R.id.register);
        backToLogin = findViewById(R.id.backtologin);
        password_create = findViewById(R.id.password_create);
        email_create = findViewById(R.id.email_create);
        username_create = findViewById(R.id.username_create);
```

```
private void createUserEmailAccount(String email, String password, String username) {
    if(!email.isEmpty() && !password.isEmpty() && !username.isEmpty()){
        firebaseAuth.createUserWithEmailAndPassword(email,password)
                .addOnCompleteListener(task -> {
                    if(task.isSuccessful()){
                        currentUser = firebaseAuth.getCurrentUser();
                        assert currentUser != null;
                        String currentUserId = currentUser.getUid();

Create a User Map so we can create a user in the user Collection in Firestore Database
                        Map<String,String> userObj = new HashMap<>();
                        userObj.put("userId",currentUserId);
                        userObj.put("username",username);

Save to our Firestore Database
                        collectionReference.add(userObj)
                                .addOnSuccessListener(documentReference -> documentReference.get()
                                        .addOnCompleteListener(task1 -> {
                                            if(Objects.requireNonNull(task1.getResult()).exists()){
                                                String name = task1.getResult().getString( field: "username");

if the user is registered successfully then we start the MyJournal Activity
Getting the Global Journal user
                                                JournalUser journalUser = JournalUser.getInstance();
                                                journalUser.setUserId(currentUserId);
                                                journalUser.setUsername(name);

                                                Toast.makeText( context: RegisterActivity.this,  text: "Welcome " + name, Toast.LENGTH_SHORT).show
                                                Intent intent = new Intent( packageContext: RegisterActivity.this, MyJournal.class);
                                                intent.putExtra( name: "usernname",name);
```

Il s'agit de la classe Model pour le poste,

Il est utilisé comme classe abstraite pour tous les messages

```java
public class Journal {
    private String title;
    private String thoughts;
    private String imageUrl;
    private String userId;
    private String userName;
    private String timeAdded;

    public Journal() {
//        Empty Constructor Needed for Firestore
    }

    public Journal(String title, String thoughts, String imageUrl, String userId, String userName, String timeAdded) {
        this.title = title;
        this.thoughts = thoughts;
        this.imageUrl = imageUrl;
        this.userId = userId;
        this.userName = userName;
        this.timeAdded = timeAdded;
    }

//    Getters and Setters

    public String getTitle() { return title; }

    public void setTitle(String title) { this.title = title; }
```

Cette classe est responsable de l'affichage des messages des utilisateurs

```java
public class JournalRecyclerAdapter extends RecyclerView.Adapter<JournalRecyclerAdapter.ViewHolder> {

    private Context context;
    private List<Journal> journalList;

    public JournalRecyclerAdapter(Context context, List<Journal> journalList) {
        this.context = context;
        this.journalList = journalList;
    }

    @NonNull
    @Override
    public JournalRecyclerAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
        View view = LayoutInflater.from(context)
                .inflate(R.layout.journal_row, viewGroup,  attachToRoot: false);
        return new ViewHolder(view, context);
    }

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

        Journal journal = journalList.get(position);
        String imageUrl;

        holder.title.setText(journal.getTitle());
        holder.thoughts.setText(journal.getThoughts());
        holder.name.setText(journal.getUserName());
        imageUrl = journal.getImageUrl();
```

Cette classe est chargée de sauvegarder les messages dans la base de données

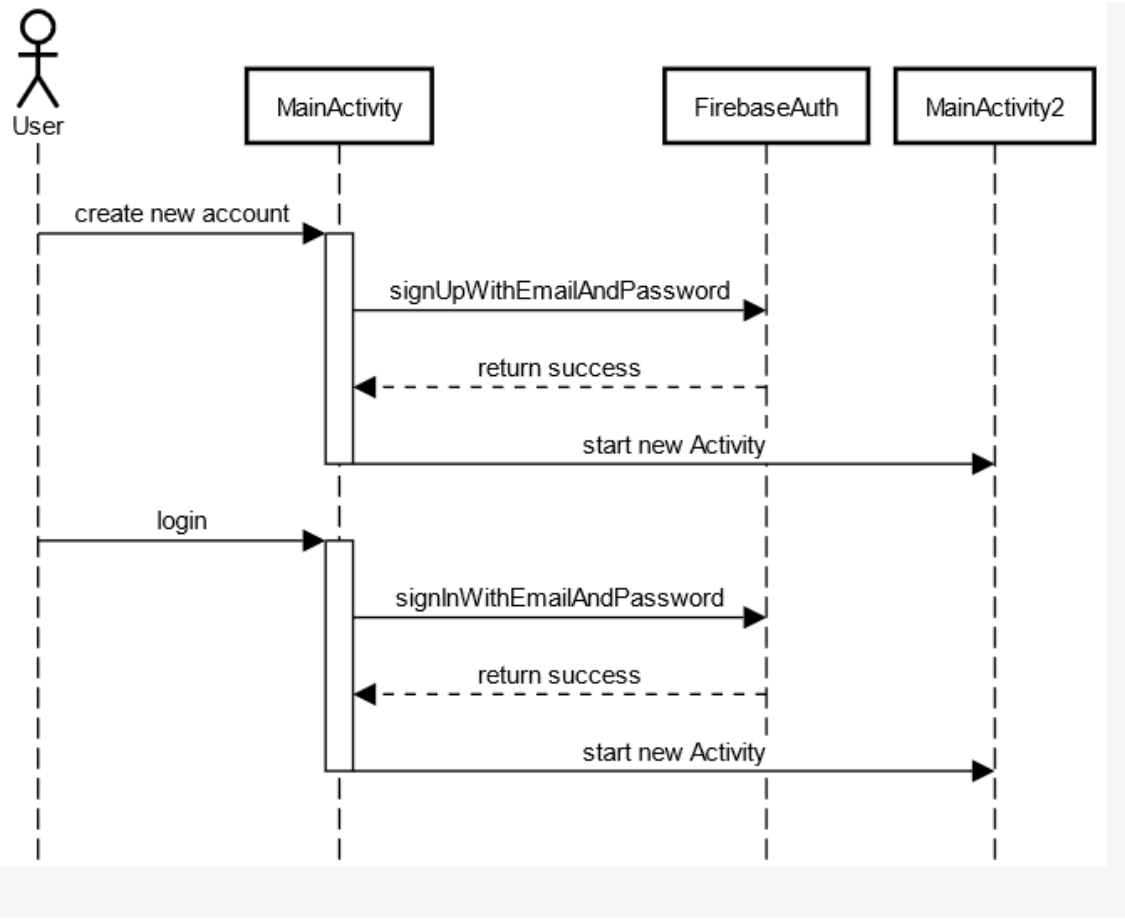
```java
public class MyJournal extends AppCompatActivity {

    private static final int GALLERY_CODE = 1;
    private TextView welcomeUser;
    private Button postSave, backToDashboard;
    private ProgressBar progressBar;
    private ImageView addImageButton, imageView;
    private EditText title, thoughts;

//    User Id & Username
    private String currentJournalUserId;
    private String currentUsername;
    /**
     * Firebase Authentification
     */
    private FirebaseAuth firebaseAuth;
    private FirebaseAuth.AuthStateListener authStateListener;
    private FirebaseUser currentUser;

    /**
     * Firebase Firestore
     * FireBase Connection
     */
    private final FirebaseFirestore db = FirebaseFirestore.getInstance();
    private StorageReference storageReference;
    private final CollectionReference collectionReference = db.collection( collectionPath: "Users");
    private final CollectionReference collectionReferenceJournal = db.collection( collectionPath: "Journal");
    private Uri imageUri;
```

Diagramme de sequence pour l'application


User
MainActivity
FirebaseAuth
MainActivity2
create new account
signUpWithEmailAndPassword
return success
start new Activity
login
signInWithEmailAndPassword
return success
start new Activity