



Milestone 2

- User Interface
 - Layout
 - Usability
- Roadmap
 - Features
 - Work Distribution
 - Collaborative work

Authors

Andrés Navarro Pedregal (100451730)

Daniel Toribio Bruna (100454242)

Alejo González García (100454351)





Introduction

The objective of our project is to create an app that can be used by runners so that they can measure their runs, compete, and challenge other friends, check the weather, and track their improvement.

During this weeks, we have followed the initial roadmap, according to the topics seen in class. We have developed most of the parts of the Back End of the application and some of the Front End. The code for the activities is already implemented, together with the login functionality and user management. The most difficult part has been the process of authenticating the users and finding the way to store their information.

Road Map Achievements

1. Backend Development.

Following our initial RoadMap, we have advanced with the 2º point, the Backend Development, where have been capable of correctly integrating Firebase within the app, been also able to Log in with any Google account.

This part of the application is handled in the “LoginActivity.java” file. We had some problems at the beginning until we really understood how the part of the database works correctly, but as soon as we have developed that part, the user handling is really easy and simple.

Another advantage of forcing the users to log in with their google account is that we do not need to handle with users and passwords, ie, having to securely safe user’s data.

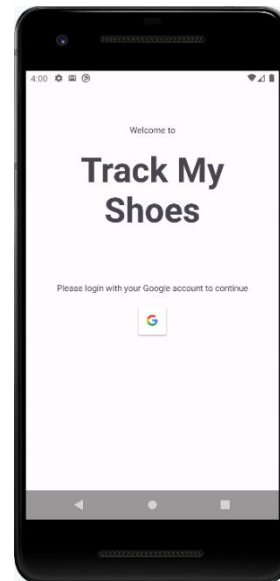
Buscar por dirección de correo electrónico, número de teléfono o UID de usuario					Agregar usuario		
Identificador	Proveedores	Fecha de creación ↓	Fecha de acceso	UID de usuario			
andream2949@gmail.c...		21 mar 2024	21 mar 2024	htrQsdcyW7TbZ8jc0SJaxVxA...			
lu34s3wer3@gmail.com		6 mar 2024	21 mar 2024	2XLPriJ7SxVKMuz5bzdbjT7p0...			
Filas por página: 50					1 – 2 of 2	<	>

		Agregar proveedor nuevo
Proveedor	Estado	
Correo electrónico/contraseña	Inhabilitado	
Google	Habilitado	





```
public class LoginActivity extends AppCompatActivity {  
  
    5 usages  
    private static final String TAG = "GoogleActivity";  
    2 usages  
    private static final int RC_SIGN_IN = 9001;  
  
    4 usages  
    private FirebaseAuth mAuth;  
  
    2 usages  
    private GoogleSignInClient mGoogleSignInClient;  
  
    3 usages  
    private ActivityLoginBinding binding;  
}
```



Login Activity code together with the Log In view in the android device.

Some of the Key files for the Back End Development that we have used:

1. ***“RunsAdapter.java”***

Defines a RecyclerView adapter called *RunsAdapter* for displaying a list of running activities. It inflates the layout for each item, binds data to the corresponding TextView, and implements an interface for handling item click events.

It is the element that allows us to display all the runs of a user one after another.

2. ***“SplashScreen.java”***

Briefly displays when the app is launched. After a delay of 1 second, it checks if the user is logged in using Firebase authentication.

Depending on the login status, it either navigates to the main activity (MainActivity) if the user is logged in or to the login activity (LoginActivity) if the user is not logged in.





3. “RunUtil.java”

This utility class RunUtil facilitates fetching runs associated with a user from Firebase. It defines a callback interface RunsOfUserCallback with methods for handling successful retrieval or failure.

The getRunsOfUser method fetches runs stored in Firestore for the current user using asynchronous Firestore queries, populates an ArrayList with retrieved runs, and notifies the callback accordingly.

```
public static void getRunsOfUser(final RunsOfUserCallback callback){
    ArrayList<RunModel> runs = new ArrayList<>();

    CollectionReference documentReference = FirebaseFirestore.getInstance().collection(String.join( delimiter: "/", ...elem

    documentReference.get().addOnSuccessListener(new OnSuccessListener<QuerySnapshot>() {
        @Override
        public void onSuccess(QuerySnapshot querySnapshot) {
            try {
                for (DocumentSnapshot document : querySnapshot.getDocuments()) {
                    RunModel run = document.toObject(RunModel.class);
                    runs.add(run);
                }

                callback.onRunsOfUserLoaded(runs);
            } catch (RuntimeException exception) {
                callback.onFailed( errorMessage: "The run document cannot be assigned to the RunModel class");
            }
        }
    })
    .addOnFailureListener(new OnFailureListener() {
        @Override
        public void onFailure(@NonNull Exception e) {
            callback.onFailed( errorMessage: "Cannot get the run document");
        }
    });
};
```

2. Front End Development

Regarding the user interface, we have made lot of progress. We have developed the Main view, where the user can select all the available functions of the application.

Notice that we are storing all the runs for a user, and we loop through them to show them ordered in the main screen. (Using a Recycling View)

There is a button for each of the functions, one to go the view of the weather (We still must connect the API), another that takes you to the leaderboard, and you can see the top players, and another one for starting a new run.

We have implemented different views that we are going to require in order to fulfil the requirements of our application.





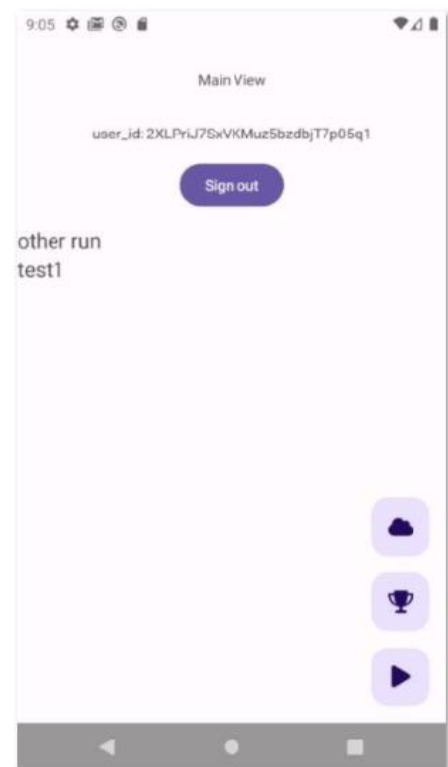
1. “MainActivity.java”

Initializes the main activity for a tracking application. It fetches the current user data and displays their email, loads and the user's runs in a RecyclerView, sets up buttons for signing out and navigating to other activities like tracking, leaderboard, and weather. It also implements an item click listener to handle interactions with the RecyclerView items.



2. “TrackActivity.java”

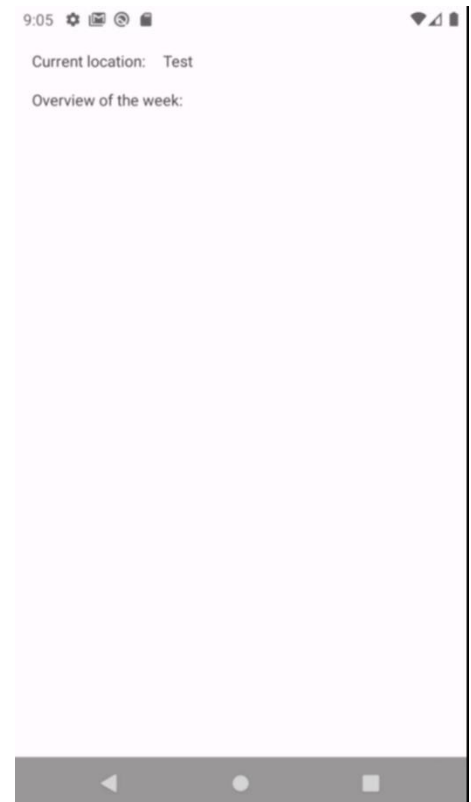
For the moment, this activity is responsible for displaying tracking-related functionalities within the application. We will further develop it.



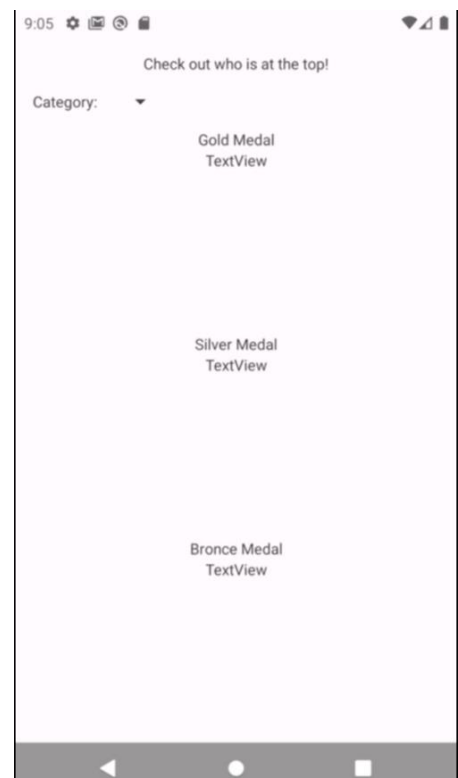


3. “WeatherAcitivity.java”

As the previous activity, this is created indeed, to represent the Weather view. We still have to connect the API and show the content. Further development is required.



- ### 4. “LeaderboardActivity.java” is the file where we are going to add all the information related to the Leaderboard, for the moment we are just showing the Top 3 rank players. Our idea is to store the data of each run and show the top 3 best performing runners on the leaderboard.





Road Map

For the rest of the course, we are going to keep developing our application:

- We must further develop the Secondary views related to the Weather (connect to external services like the API).
- The tracking, to further test with more examples the correctness of its functionality.
- Regarding the Leaderboard, we must also connect the users correctly to the top, making sure that only the top ones are shown and that the users are updated correctly and displayed as intended.
- As soon as most of the main functionalities are working, we will start adding the secondary features such as the Language versioning and a map with GPS location.
- Regarding Testing, we **MUST** ensure the correct functionality of all the features and components of the application, and we are planning to do automatic tests by using the app while running and trying to find possible cases in which the app does not behave as expected.
- Another feature is going to be the implementation of a chronometer, to measure the time of each run correctly.

Work Distribution

For the moment, thanks to the use of GitHub, we have been capable of work with the versioning of the code all the three of us correctly.

We were using the GitHub Desktop application, but it is true that Android Studio already allows us to use GitHub directly without the need to import the files to other location.

We will continue using this collaborative tool as it saves lot of time, specially, when you want to go back to a previous version or whenever we try to merge different pieces of code into the same project.

Encountered Problems

For the moment, we just had to get used and a bit familiarized with Firebase at the beginning, and as we just mentioned, we had a few problems regarding the versioning of the code.

Sometimes we have some errors regarding the Firebase login by using Google accounts as when we log in with the account it raises an error, and we are doing a bypass at the moment to solve it. We are trying to fix it as it is quite rare and specific, but sooner or later we will have it fixed!

Apart from these issues, we think we are doing right and on the correct direction.





Google Drive Code

Source Code:

https://drive.google.com/file/d/1F1izPxaNiUE_qyNHWjCTzh_1lc3ZFH0I/view?usp=sharing

Video Of the App:

https://drive.google.com/file/d/121Z9kNQ_iFxR62UZO1ofXh2ixETZn_g0A/view?usp=sharing

Considerations to execute the app

In order to execute the app in an AVD you must consider using Pixel 2 device and because of Firebase we decided to use the “Pie” version, so with a Pixel 2 API 28 or above you should be fine.

Conclusion

For this first part of the project, most of the work has been concentrated into the Back end. It is necessary to have the database and the main processes of the application correctly working so that we can continue developing the app and create a more user-friendly environment, ie, working in the Front-End development and the appearance of the layouts.

We are happy with the results we are achieving, and as time goes on, we are getting closer to the final result that we want to achieve.

References

Up to the moment, for this project we have only used 3 sources:

- Github-Copilot that offers a free licence for students (as we saw in class)
- ChatGpt – For code errors fixing mainly
- C2.350.16504-96* Mobile Applications 23/24 Contents. This is from where we have mainly taken the code, as expected.

