Technical Project Report - Android Module

MovingCampus

Subject:	Introdução à Computação Móvel
Date:	Aveiro, 19/04/2023
Students:	104247: Pedro Lima Baía Coelho 98652: Bernardo Alves Leandro
Project abstract:	Application that allows Erasmus students to explore the university of Aveiro, which relies on the use of location, QR Code reader and text recognition.

Report contents:

1 Application concept

This application aims to help the users, more specifically Erasmus students, to learn about the campus of the University of Aveiro. With our app, students can ask for directions to their classroom, learn about the buildings around them by scanning their QR Codes, and even view and add events available to all students to help with their integration.

2 Implemented solution

Architecture overview (technical design)

Architecture

We based our application around an MVC architecture where we have the models created in “classes” folder, the views are the layouts created in “layouts” section and the file DatabaseInfo.kt acts as a controller, getting information from the database and passing it to the views.

Database

Our application data is stored in an SQLite database with the objective of being later deployed into Firebase Realtime Database. In our database we have five tables:

- “users”: storing the user’s information such as name, e-mail, password, etc.
- “events”: stores the information about events. In this table we have the attributes – name, location, date, etc.
- “cSchedule”: associates a classroom to its schedule where we can retrieve the information about what classes are occurring at a given time.
- “sSchedule”: associates a student to his schedule where we can retrieve the information about what classes he is taking and where and when they will occur.
- “friends”: this table will associate two users showing that they are friends, using both of their NMECs. This feature is not yet implemented.

MLKit

We used MLKit API’s vision, to read QR Codes and recognize text. The QR Code reader is used for the students to read QR Codes that are displayed in each campus’ building, getting information about them. The other AI tool is used for the students to scan the number from a classroom to see its schedule. We used the androidx.camera.view.PreviewView to display a preview of the camera in our fragment so we can aim for the desired content.

API TomTom

This API is used for us to display and navigate throw the campus' map. With this API we can insert a specific classroom (ex:04.02.01) and the API will return the rout to its location. Also we defined markers that represent all departments from UA and others important buildings like cafeteria and residences. We used the androidx.fragment.app.FragmentContainerView layout design to display the map in our fragment (PathToClassroomFragment).

Implemented interactions

Main activity

This activity is responsible for the initialization of the application where we are routed to our first fragment (LoginFragment). Also, this activity is responsible for the initialization of the RecyclerViews, FloatingActionButton, BottomNavigationView and NavHostFragment. The navigation between fragments is made via NavController.navigate().

1. Registration Interaction

   - Firstly, we start by clicking the Register button. After, we are redirected to another fragment, the RegisterFragment, where we can write our information, and after clicking Register, if some information is missing or is incorrectly filled, we can see error boxes. If everything is good, we are redirected back to the LoginFragment and our information is stored in the database, as we are ready to login.

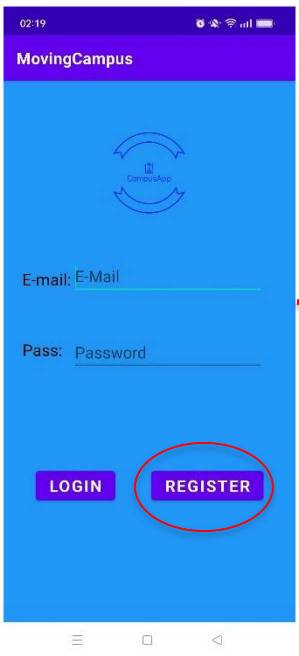


Figure 1: Login Fragment

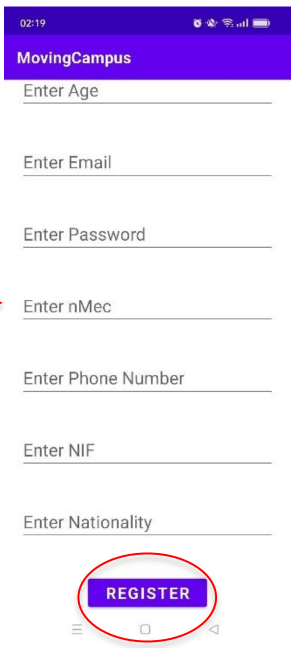


Figure 1: Register Fragment

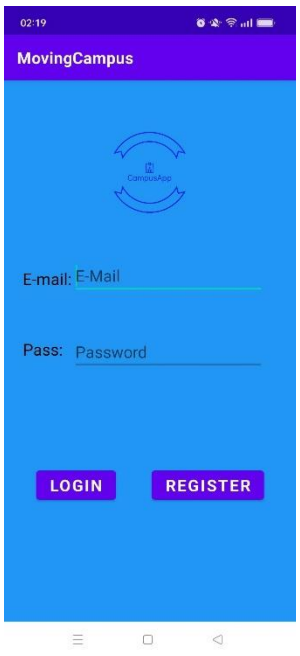


Figure 3: Login Fragment, back from Register

2. Login Interaction

- When we login, if our login information is incorrect, we will see error boxes, if not, we will be redirected to the DashboardFragment where we can see events created by the users. These events are displayed with a RecyclerView containing CardViews.

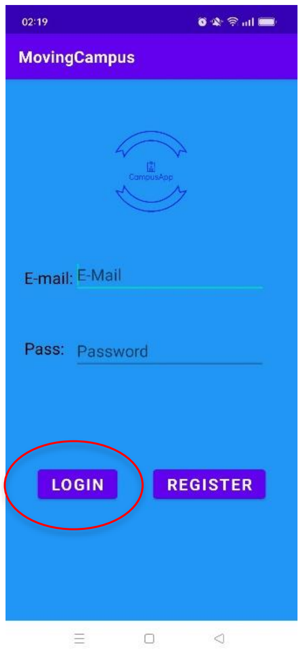


Figure 4: LoginFragment

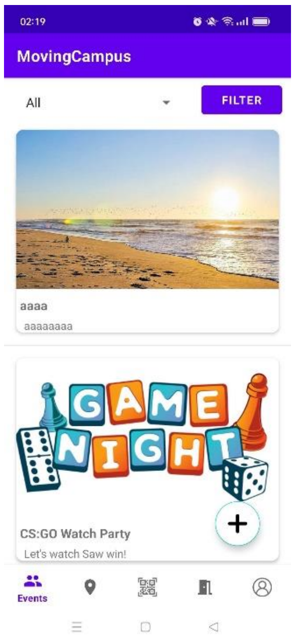


Figure 5: DashboardFragment

3. Bottom Navigation Menu

- In the bottom navigation menu, we can switch between DashboardFragment, PathToClassroomFragment, FindCampusFragment, ClassroomScheduleFragment and ProfileFragment. We will explain these fragments next.

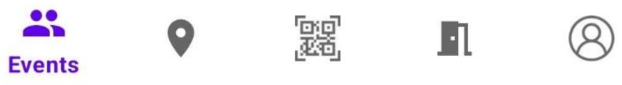


Figure 6: Bottom Navigation Menu

4. Filter Events Interaction

- We can select a filter to search for the events. After we click the filter button we get the events that belong to that category

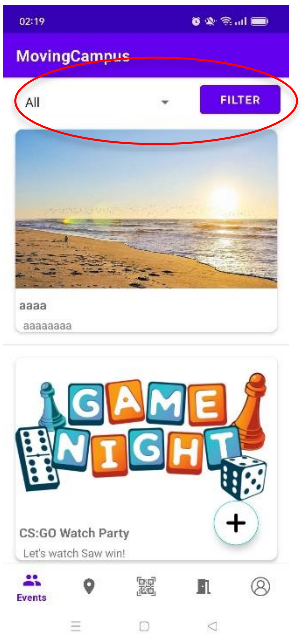


Figure 7: All Events shown

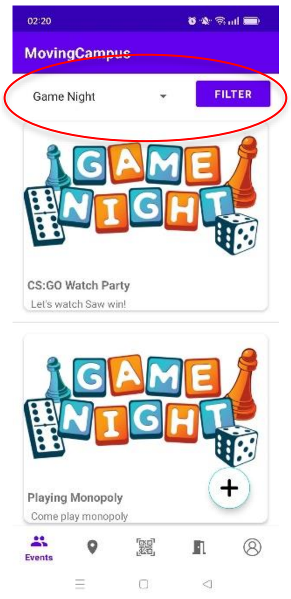


Figure 8: Filtered Events shown

5. Check event Information Interaction

   - If we click on an event we are redirected to the event's information page, where we can see location, date and who created the event. We can also see more information about who created said event.

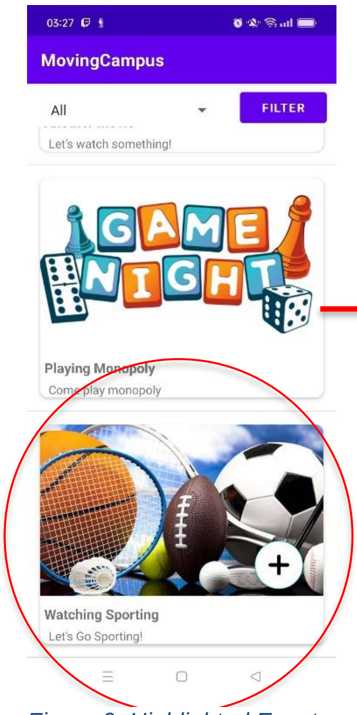
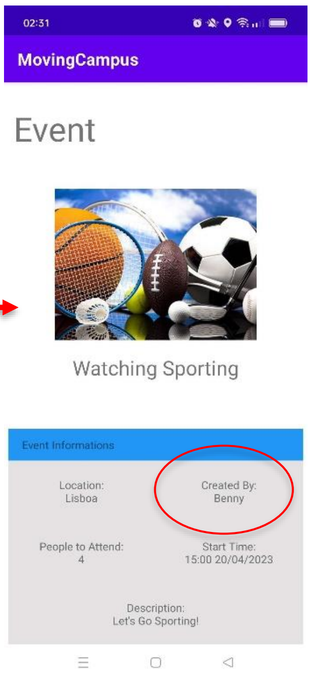
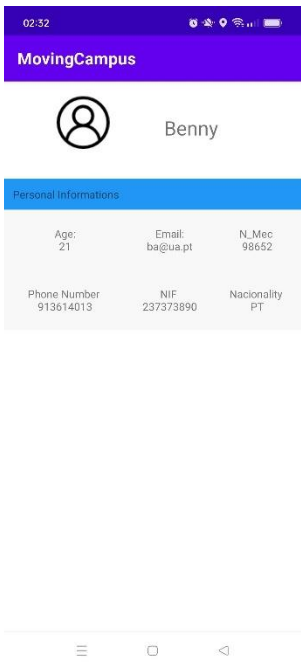


Figure 9: Highlighted Event



Location: Lisboa
Created By: Benny
People to Attend: 4
Start Time: 15:00 20/04/2023
Description: Let's Go Sporting!

Figure 10: Highlighted Event's details



Age: 21
Email: ba@ua.pt
N_Mec 98652
Phone Number 913614013
NIF 237373890
Nacionality PT

Figure 11: Highlighted Event's creator

6. Add Event Interaction

- By clicking the floating action button, we will be routed to the AddEventFragment, where we can create our fragment. Being redirected back to the page where we can see the events.

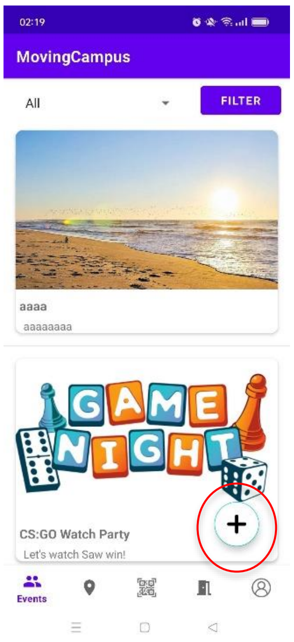


Figure 12: Highlighted Floating Action Event

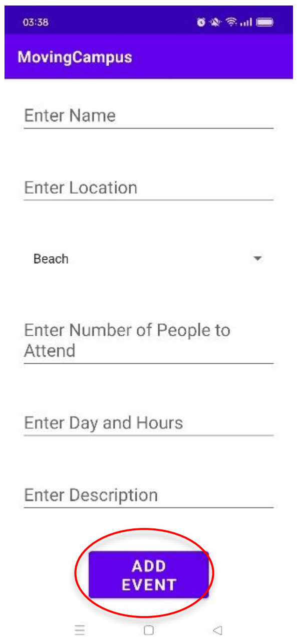


Figure 13: Add event fragment

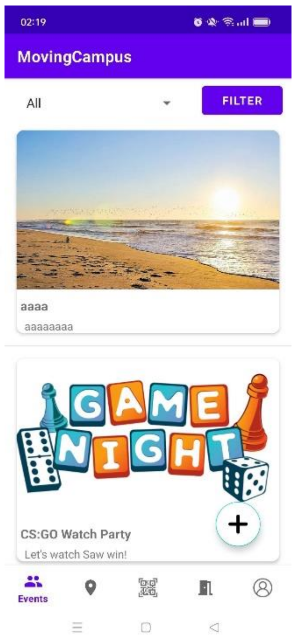
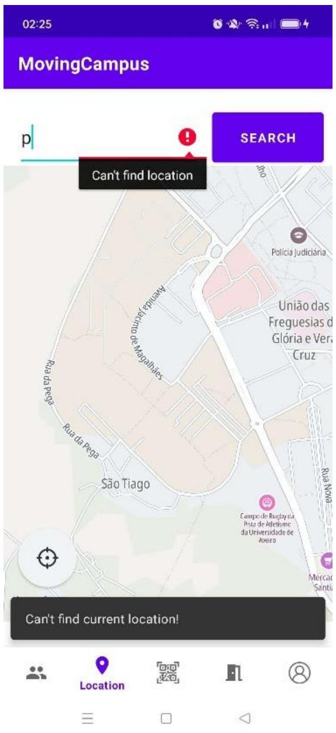


Figure 14: Back to all events fragment

7. Location Interaction

- When we go to the PathToClassroom, we can see a map we markers representing every important campus building for Erasmus Students. Here we can search for a classroom or a bulding. If our location or the classroom/building we search for is invalid, we get error messages. If they are valid we get a route from our location to that classroom/building. We can also click on the markers to find out what the building is. This was done with a CustomBalloonViewAdapter



Can't find location

Can't find current location!

Figure 16: Error on locations

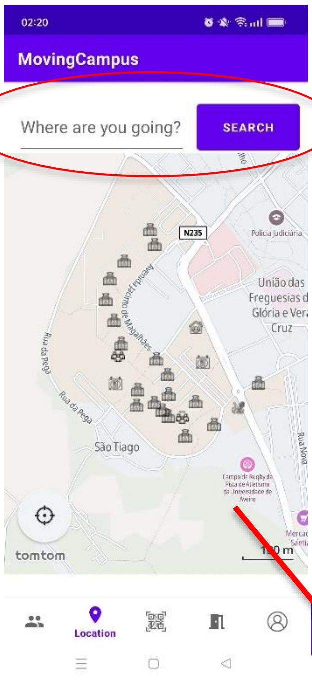
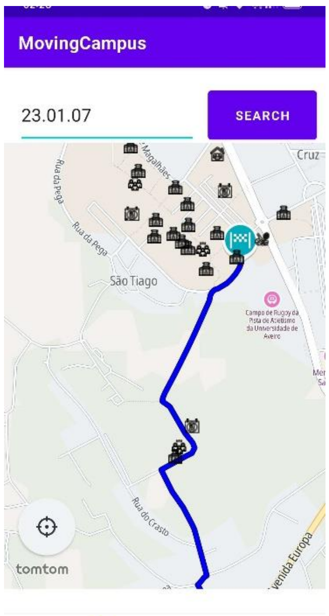


Figure 15: Map with markers



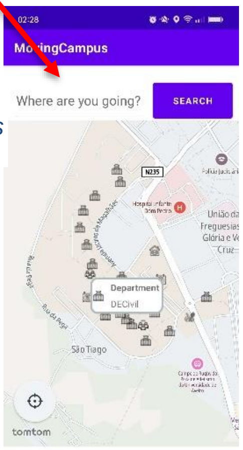
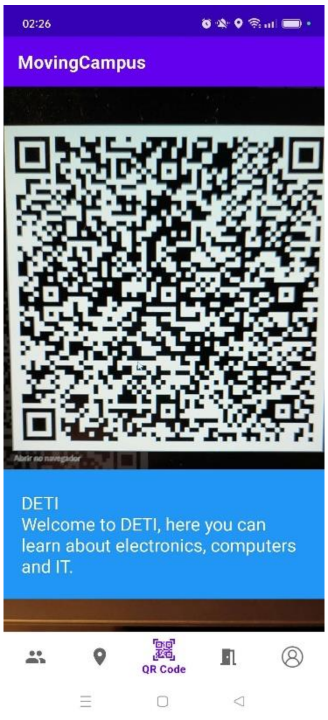


Figure 18: Markers balloon view

Figure 17: Route found and shown

8. QR code Scan

- In this Fragment, we can read QR codes. The QR codes will be displayed in each building, giving information about them, so Erasmus students can learn more.



DETI
Welcome to DETI, here you can learn about electronics, computers and IT.

Figure 19: Qr code scanner and text shown

9. Classroom Schedule Search

- In this Fragment we are presented with two buttons. One searches the inserted classroom and will give you it’s schedule. The other opens up a camera to read text. Once you feed it a classroom number, it will give you said classroom’s schedule. If the classroom is not found, we will go to an empty frgament. To present the schedule we used a RecyclerView with TableLayout.

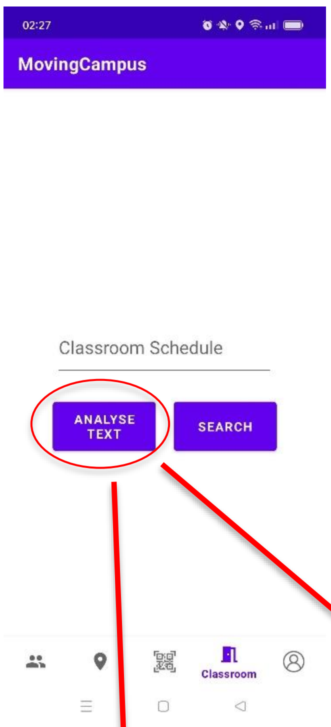


Figure 20: Classroom search by text recognition

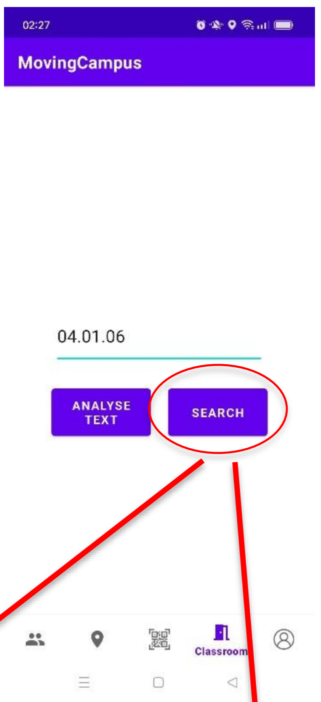


Figure 21: Classroom search by button

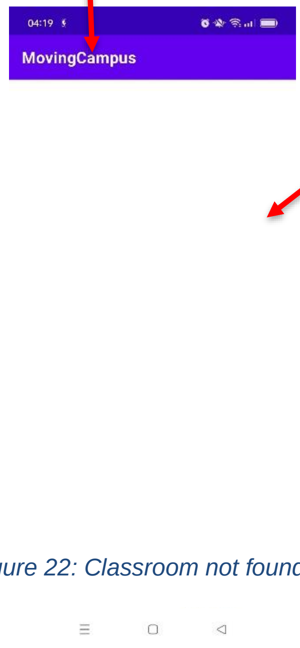
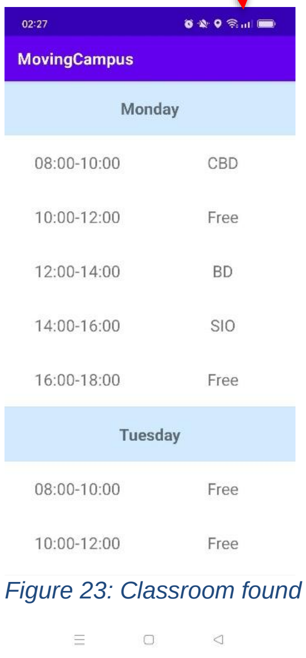


Figure 22: Classroom not found

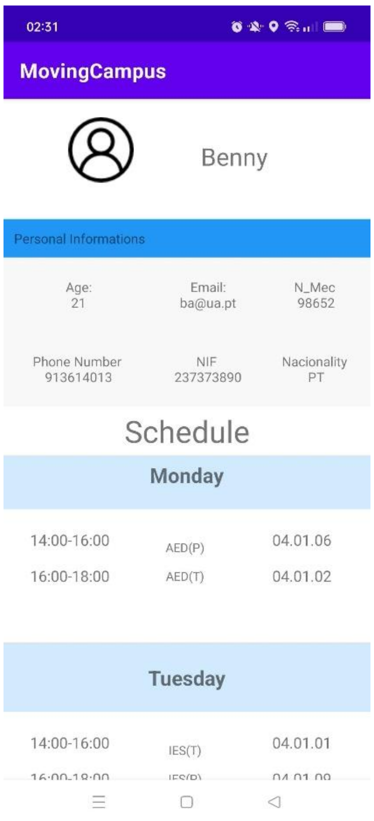


Monday	
08:00-10:00	CBD
10:00-12:00	Free
12:00-14:00	BD
14:00-16:00	SIO
16:00-18:00	Free
Tuesday	
08:00-10:00	Free
10:00-12:00	Free

Figure 23: Classroom found

10. Profile and User Schedule

- In this Fragment we can see our personal informations, such as our e-mail, phone number, nationality, etc. We can also see our schedule. To present the schedule we used a RecyclerView with TableLayout, just like we did for the classroom's schedule.


02:31

MovingCampus

Benny

Personal Informations

Age: 21	Email: ba@ua.pt	N_Mec 98652
Phone Number 913614013	NIF 237373890	Nacionality PT

Schedule

Monday

14:00-16:00	AED(P)	04.01.06
16:00-18:00	AED(T)	04.01.02

Tuesday

14:00-16:00	IES(T)	04.01.01

Figura 24: Profile Fragment: Personal information and schedule

Project Limitations

Despite having several functionalities, we were not able to implement all the desired ones at the beginning of the application development. Unfortunately, as previously mentioned, we were unable to place our database online through FireBase, we did not develop the user's friends tab, the route on the map does not update according to the user's progress and we still lacked the development of Augment Reality.

For future work, we would like to implement all of the mentioned above. Besides those, we would also like to filter events that where created by the user’s friends.

3 Conclusions and supporting resources

Lessons learned

Throughout the development of our application, we encountered several problems, some foreseen, but others much more complex than we anticipated. One of them was the development of the map using the TomTom API. In this module we had some problems in calculating the route through the insertion of a classroom. After solving this matter using markers (in each department) with a geolocation assigned to each one, we had problems in displaying this same one. To solve this problem, an investigation was made through the TomTom documentation and several videos on YouTube until we reached the solution.

Another problem found was in extracting data from the database to TableLayouts. In addition to taking time to extract all the data we wanted from the database, we had difficulties in treating this data in order to be presented correctly in the TableLayouts.

Finally, another major problem encountered at the beginning of the application's development was the use of RecyclerView to display events. First, the RecyclerView was not updated after a new event was inserted. After resolving this situation, the RecyclerView did not show all the events (sometimes it didn't show the first one and sometimes it only showed one).

After this project we saw how important good documentation can be, and we now know that we can learn how to surpass certain errors by reading lots of documentation.

Work distribution within the team

Bernardo Leandro – 50%

Pedro Coelho - 50%

Project resources

Resource:	Available at:
Code repository:	https://github.com/PedroC55/ICM_Project1
Ready-to-deploy APK:	In repository

Reference materials

Tomtom maps sdk:
https://developer.tomtom.com/android/maps/documentation/overview/introduction
MLKit QR Code scanner:
https://developers.google.com/ml-kit/vision/barcode-scanning/android?hl=pt-br
MLKit Text Recognition:
https://developers.google.com/ml-kit/vision/text-recognition/android?hl=pt-br
CameraX getting started tutorial from Google Developers:
https://developer.android.com/codelabs/camerax-getting-started?hl=pt-br#0
QR Code Generator:
https://qr.io/?gclid=CjwKCAjw8-OhBhB5EiwADyoY1SRra-K3A3oPfNHxcydEX3zMXPYYzdZHJn7TxZY-kAVxEblxPQ84JRoCa3oQAvD_BwE
Icons used from:
https://www.flaticon.com/