Technical Project Report - Android Module

App to Report Trash related Problems

Subject:	Universidade de Aveiro - Computação Móvel
Date:	29/01/2025
Students:	98359 : Gonçalo Fernandes Machado
Project abstract:	App that users can use to report trash related problems. They can submit photos, locations and see status. Admins can see problems and resolve them.

Report contents:

1 Application concept

Overview

This app is for citizens and the people in charge of the city to communicate and report problems related to trash. By using the app they could have a cleaner city and better organize cleaning the streets.

Essential journey map

1. **Login / Register Screen**
    - User logs in or creates a new account.
    - User role ('User' or 'Admin') is determined.
2. **Home Screen**
    - After login, the user sees the home page with its username and role and a logout option. There are also options for map and Reported Problems features.
3. **Map Screen**
    - **User**: Can see his own location, see markers of his previously reported problems, report new problems with location and photo.
    - **Admin**: Can view markers of problems reported by all users.
4. **Reported Problems**
    - Shows list of problems reported by users. Can sort by status or creation date. When a problem is clicked the details are shown.
    - **User**: Can view the problems he reported
    - **Admin**: Can view the problems all users reported.
5. **Logout**
    - User taps "Logout" to sign out and return to the login screen.

2 Implemented solution

Architecture overview (technical design)

The app is built with a focus on keeping the code organized and easy to manage. It uses ViewModels to handle the data and logic for each screen. Some ViewModels are shared across multiple screens to keep data consistent, while others are specific to a single screen. This helps keep the app responsive and makes it easier to add new features.

For storing data, the app uses Room, a local database, to save structured information.

Photos taken in the app are saved directly to the device's storage. To keep the screens updated automatically when data changes, the app uses Kotlin Flows, ensuring a smooth and dynamic user experience.

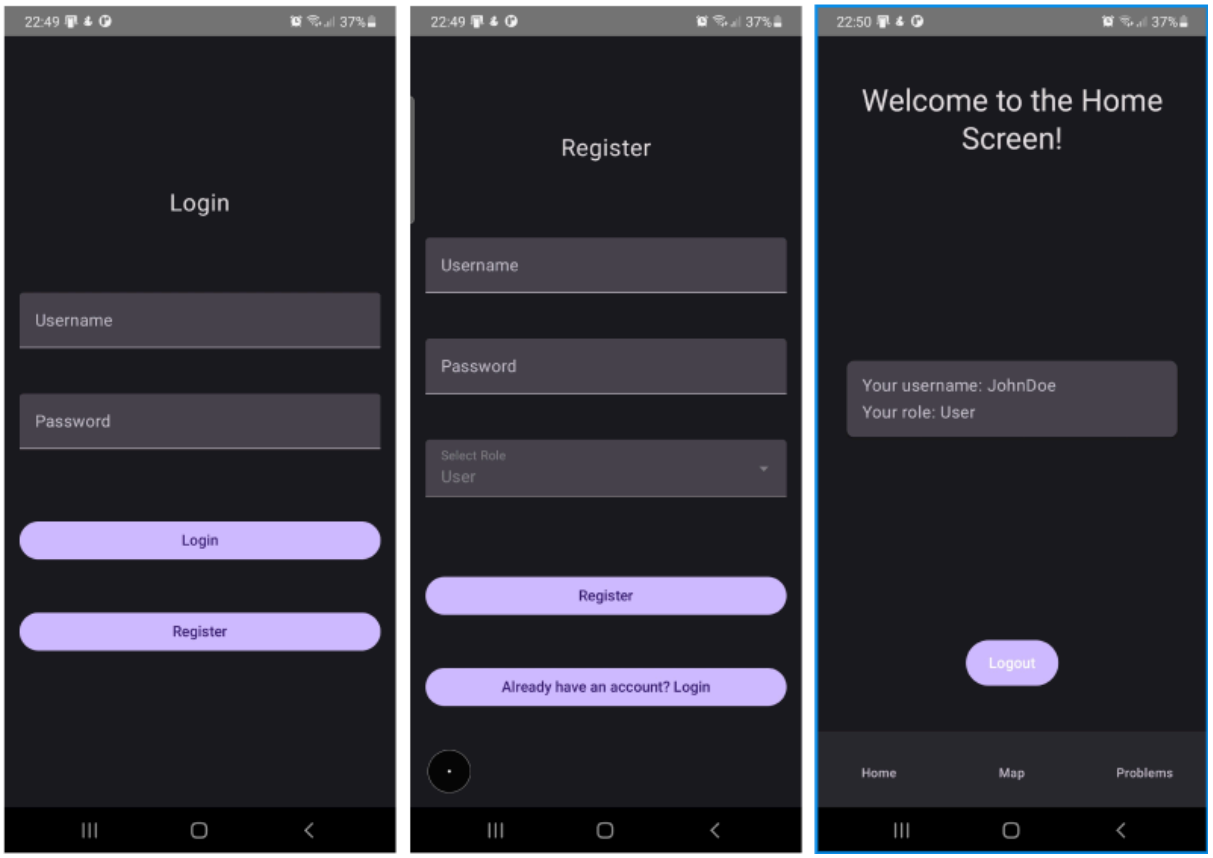
The app follows the Repository pattern to manage how the ViewModels access data. This makes it easier to work with the database and ensures the app's data handling is clean and efficient. Maps are displayed using OSMDroid, which allows the app to show locations and let users interact with them, such as adding markers.

To handle tasks like querying the database or setting up listeners, the app uses lifecycle-aware components like LaunchedEffect and DisposableEffect. These ensure that tasks start and stop at the right times, avoiding unnecessary resource usage.

Implemented interactions

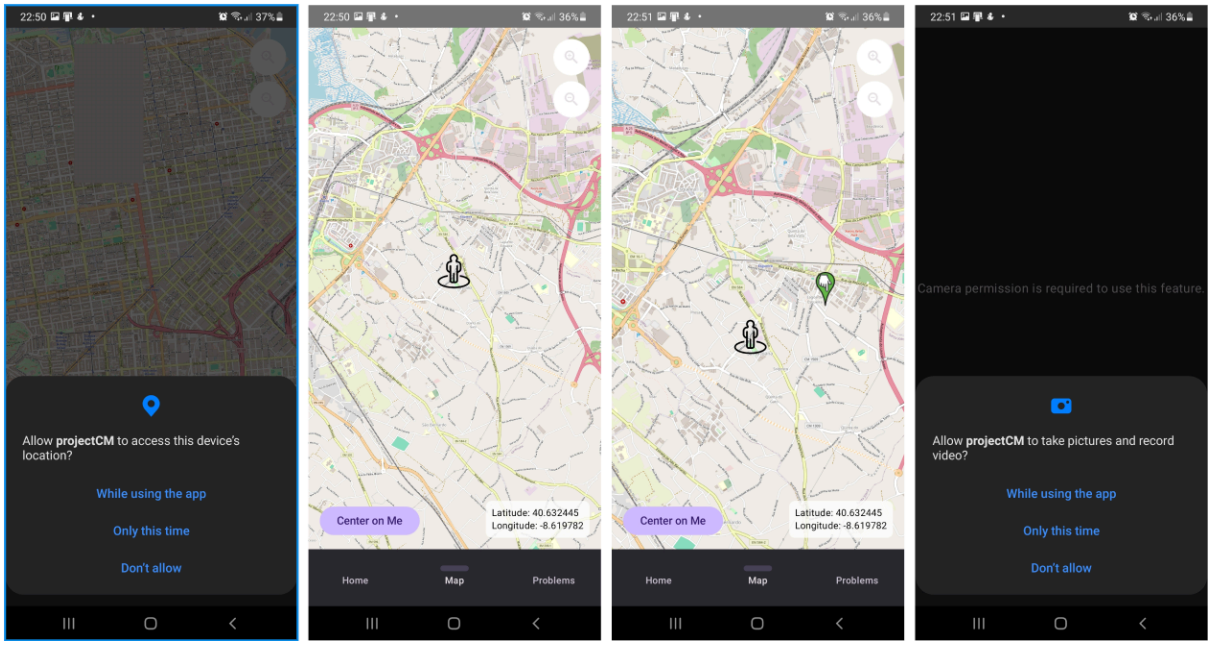
Login/Register

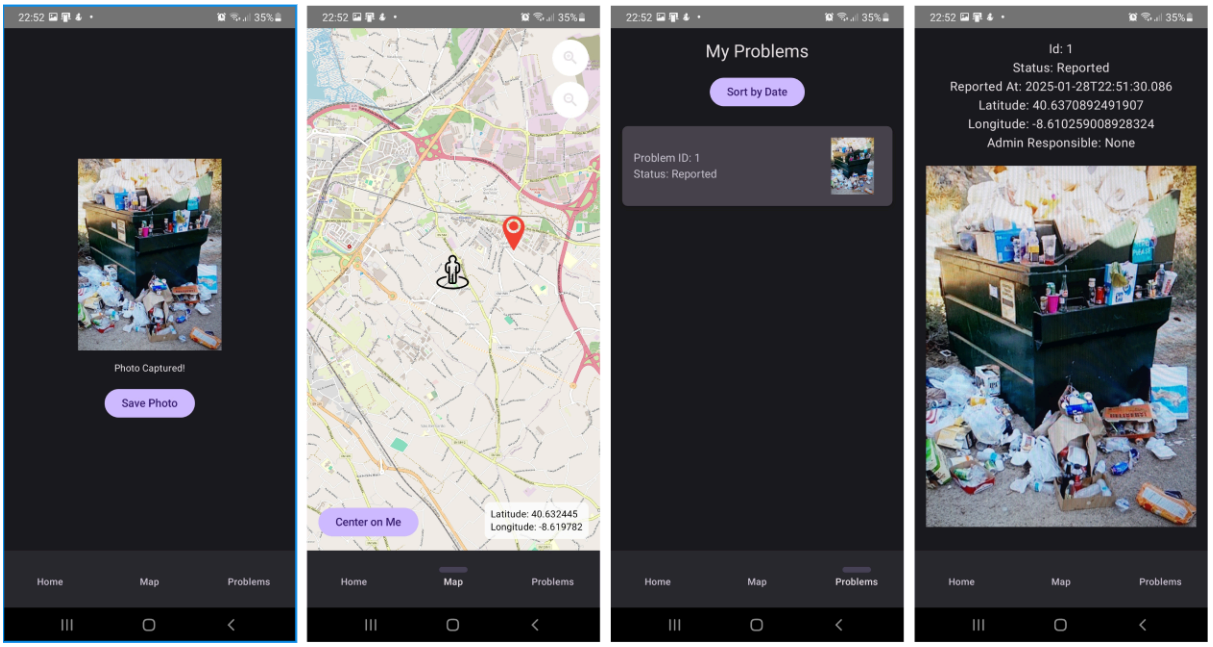
In this page the user can login if he has an account or register if he does not. He can register as an User or an Admin. After logging in or registering, the user will go to the home screen.



User - Reporting a problem

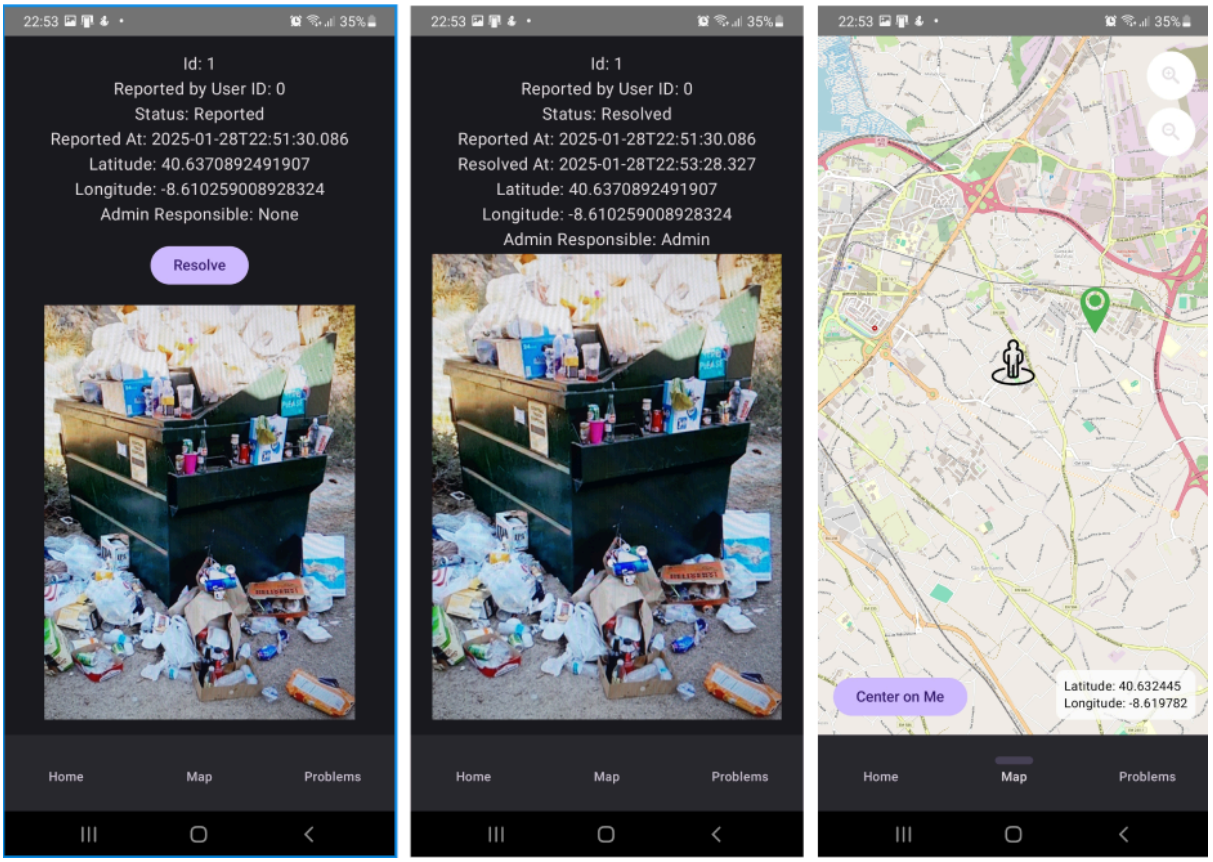
The user can access the map, where he is prompted to give permission to use location. In the map he can see his location, his coordinates and has the “Center on Me” button that centers the map on the user's location. If the user presses and holds in a location, a marker is created which represents the location of a trash related problem, like a full dumpster. If he clicks the marker he will be prompted to give access to use the camera. After giving permission, the user can take the photo and save it, completing the report, and is redirected back to the map. In the map the problems reported by the user are shown, with red being used for problems not yet resolved and green for problems already resolved. The user can see the problems reported in the “Problems” tab. The user can sort the problems by status or creation date. When clicking on a marker on the “Map” tab or on a card in the “Problems” tab the details of the problem can be seen.





Admin- Resolving a Problem

The admin can see the problems from all users and can resolve them. On the details page of a problem, the admin can resolve them by clicking the “Resolve button”. The status is updated and the date and admin name are added. In the “Map” tab, the marker of the problem is now green.



Project Limitations

The database is local, meaning that it could not be used to manage several users. The organization of the code is also not the best and should be improved in the future. The design needs a lot of improvement.

It was planned to implement push notifications to update users when a problem they reported was solved.

3 Conclusions and supporting resources

Lessons learned

The biggest problems were utilizing Room to save data. Creating the login and register page was more difficult than it was supposed to be. The dependency management was also a major problem, as some versions did not work well with one another. Unexpected recomposition also took a lot of the development time, which was solved using debug tools and logging.

Work distribution within the team

Taking into consideration the overall development of the project, the contribution of each team member was distributed as follows: Gonçalo Machado did 100% of the work.

Reference materials

OSM Droid - https://github.com/osmdroid/osmdroid

Room - https://developer.android.com/jetpack/androidx/releases/room

Project resources

Resource:	**Available from:**
Code repository:	https://github.com/goncalo-machado/CM-AndroidProject
Ready-to-deploy APK:	https://github.com/goncalo-machado/CM-AndroidProject/blob/main/app-debug.apk
App Store page:	
Demo video:	