Technical Project Report - Android Module

Pet Friends App

Subject:	Computação Móvel
Date:	Aveiro, 19/04/2023
Students:	103645: Manuel Diaz 654321: Leonardo Freitas
Project abstract:	A Pet Sitting app that allows users to find and book for pet sitting services. Key achievements of the app include a user-friendly interface where users can create a profile, add pets, search for sitters and see the tracking routes between user and pet sitter.

Report contents:

1 Application concept

Our pet sitting app is designed to help pet owners find reliable and trustworthy pet sitters in their local area. The app is aimed at busy pet owners who are unable to look after their pets due to work, travel, or other commitments.

The target users of the app include pet owners who want to ensure that their pets are cared for by experienced and qualified professionals while they are away.

By using our app, pet owners can benefit from the peace of mind that comes with knowing their pets are in good hands. Overall, our pet sitting app aims to make the process of finding and booking a pet sitter simple, easy, and stress-free for both pet owners.

2 Implemented solution

Architecture overview (technical design)

The architecture devised for our software solution includes several components and layers. We use the MVVM (Model-View-ViewModel) architectural pattern which separates the user interface (View) from the business logic (ViewModel) and the data model (Model).

To interact with the database, we use a DBHelper class that creates the tables and has methods to manipulate the database (Model layer). The methods are called from the activities or fragments, not from a specific file for that, but they act like the ViewModel layer, and then the data is displayed in the user interface (View Layer).

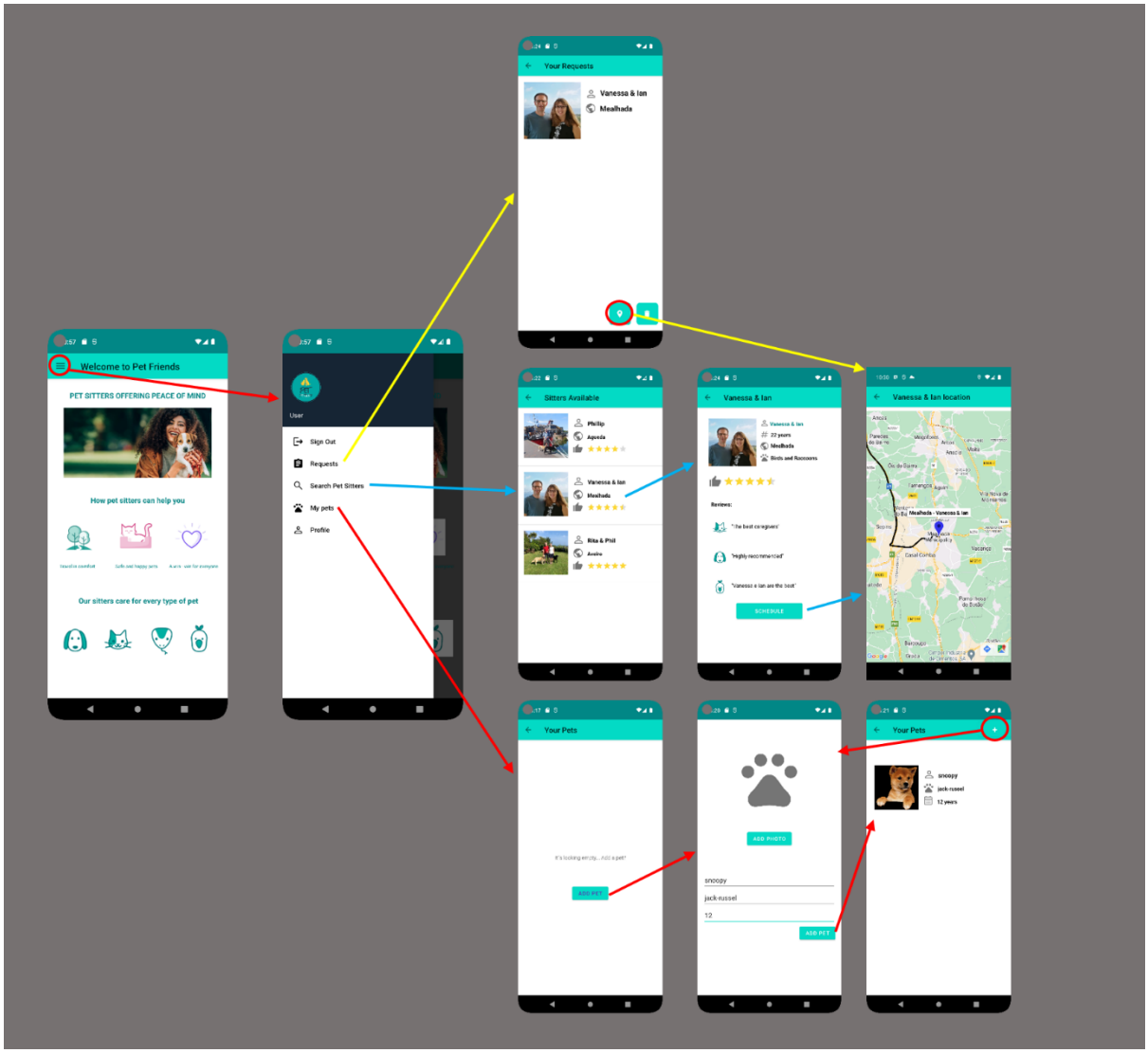
In addition, we integrate with Google Maps API, specifically the Directions API, Geocoding API, and Maps SDK for Android. This allows us to provide accurate and up-to-date location information to our users, as well as directions between locations.

In terms of connected devices, our app does not read from sensors or use NFC technology.

The DBHelper class is used to store and manage users, pets and pet sitters related data. It contains several methods for inserting, selecting, updating and deleting data from SQLite database.

When it comes to integrating with Internet-based external services, one of the strategies that we've utilized is leveraging the Google Maps API. By doing so, we were able to provide our app's users with a seamless experience of searching for, visualizing, and interacting with maps and location-based data. The Google Maps API offers a wealth of features and functionalities that developers can leverage, such as geocoding, routing, distance matrix and more. It also provides SDKs and plugins for different platforms and languages, making it easier to integrate with your app's tech stack.

Implemented interactions



(red flow) - User selects “My pets”, because the user has no pets yet, it will prompt the user to add a pet. So he Selects “Add Pet”. After pressing the app will show a dedicated menu to add pets. After this process the user will be presented with its own pet list.

(blue flow) - Then the user can search pet sitters from the menu, pressing “Search Pet Sitters”, he will be presented with the petsitter options and can read their profile and reviews and if he wants to schedule a session he can do so with the "Schedule" button. That will show the sitter on the map.

(yellow flow) - If the user wants to verify his sessions he can do so with the "Requests" button on the menu and open the map to show the sitter location.

Project Limitations

Currently, our pet care application is missing some key features that we have planned but, unfortunately, we did not have time to implement. One of these features is the ability for users to sign up and log in as a pet sitter, which would allow them to offer their services to pet owners in need.

Another important feature that we plan to implement is a notification system. This will help keep pet owners informed about important updates related to their pets, such as when a pet sitter has accepted a booking or when their pet's feeding schedule has been updated.

Finally, we would have liked to implement a kind of gallery for the user, where he or she could publish pictures of their pets, acting as a kind of social network. But this feature was not so important, but just an extra to improve the user experience.

Certainly, with more time and if we have the opportunity in the future to continue improving our platform, we are committed to implement the features in fault and providing the best possible experience for our users.

About limitations in the features we implemented, we had some problems with the Directions API of google maps, since whenever we tried to get the route between our location and the pet sitter's it gave an error, because supposedly there was no route, but we couldn't understand it, since both our location and the pet sitter's location were recognized by the Geocode API. So we had to, at least for now, assume the user's location as University of Aveiro.

New features & changes after the project presentation

We recognize the value that the key features that we said before can bring to our platform and in the future, if we have time, we would like very much to implement them.

Another feature that we would like to implement is to integrate our local storage with Firebase. By doing so, we can take advantage of Firebase's real-time database capabilities, which will enable us to sync data across multiple devices and provide a better user experience. This feature is still in the planning stage, but we hope to implement it soon.

3 Conclusions and supporting resources

Lessons learned

Throughout the development of this app, we faced several major challenges and obstacles that required careful consideration and creative problem-solving. One of the biggest hurdles was designing a seamless user experience that balanced functionality with simplicity. We learned that it's important to prioritize features and functionality based on user needs and feedback, and to continually test and iterate on the app's design to ensure it meets those needs.

Integrating Google Maps API’s in a mobile application can be a challenging task, but it can also add a significant amount of value to the application. Our team faced some obstacles when integrating the API’s into our application, but we were able to overcome them. We needed to carefully design the UI to make sure that the map was properly integrated and that the user experience was seamless. We also needed to make sure that the map was properly configured to handle user interactions, such as zooming in and out, and that the

application was properly handling map-related errors.

Overall, we think that we have learned a lot from developing this application and have gained skills to develop a much more complex application in the future and with a longer development time.

Work distribution within the team

Taking into consideration the overall development of the project, the contribution of each team member is distributed as follow: Manuel Diaz did 50% of the work, and Leonardo Freitas contributed with 50%.

Project resources

Resource:	**Available at:**
Code repository:	https://github.com/Manuel-Diaz17/Pet-Friends_ICM
Ready-to-deploy APK:	In the repository above

Reference materials

Google Maps API’s : https://console.cloud.google.com/google/maps-apis/api-list