PAL

A mobile application for adopting dogs and cats.

Ben Peretz.
Shay Solomon.
Daniel Semerjian.
Benny Batash.

Pal

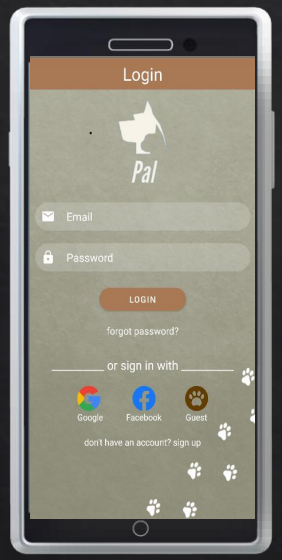
About

Pal is an Android application developed using Kotlin that helps users find and adopt pets. The app offers two options for users to sign in - as a registered user or as a guest. Registered users have the added advantage of being able to add pets to their favorites for easy access later.

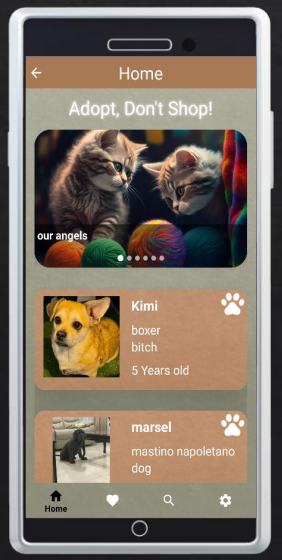
As a guest, users can still use the app almost like a registered user and browse through the available pets. The app provides detailed information about different dog and cat breeds, including whether the pet is suitable for kids, their shedding level, and other relevant information to help users make informed decisions.

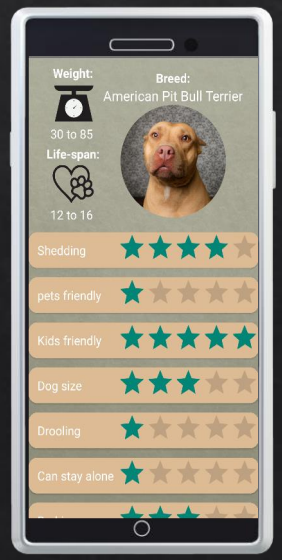
Whether you are a registered user or a guest, Pal makes it easy and convenient to search for your next furry friend. With its user-friendly interface and comprehensive information about different pet breeds, Pal is the perfect companion for anyone looking to adopt a pet.

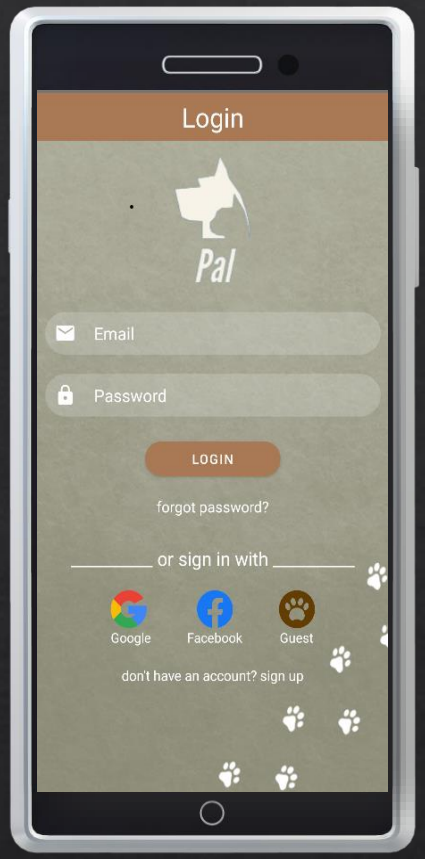
Screens


Login
Pal
Email
Password
LOGIN
forgot password?
or sign in with
Google
Facebook
Guest
don't have an account? sign up


Welcome
How about adopting a pet?
Welcome to Pal
Dog
Cat
Pal

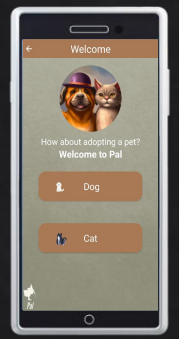

Home
Adopt, Don't Shop!
our angels
Kimi
boxer
bitch
5 Years old
marsel
mastino napoletano
dog
Home

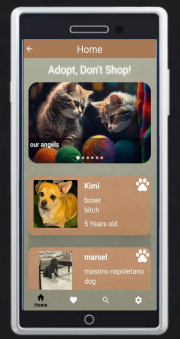

Weight:
30 to 85
Life-span:
12 to 16
Breed:
American Pit Bull Terrier
Shedding
pets friendly
Kids friendly
Dog size
Drooling
Can stay alone

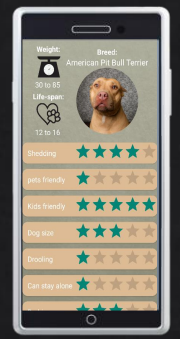


Login screen

The first time the app is launched, you will see this screen in which you must sign in with your registered user account, or create a new user account, or log in as a guest.

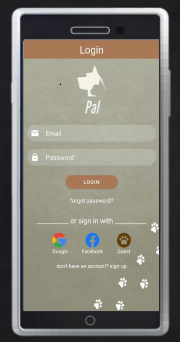


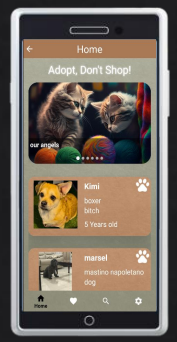


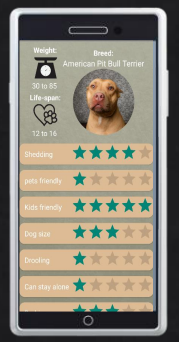


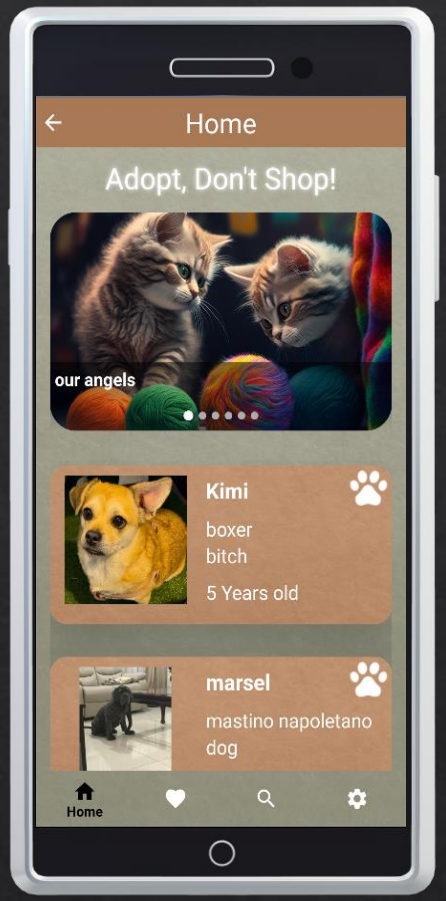
Welcome screen

Once you've logged in to the app, you'll be prompted to select a pet type to move to the home screen.



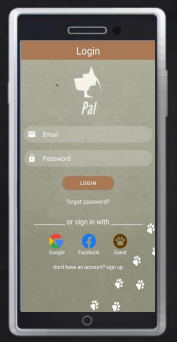


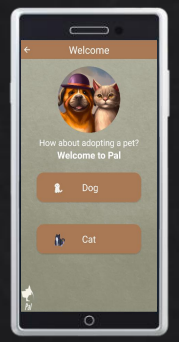


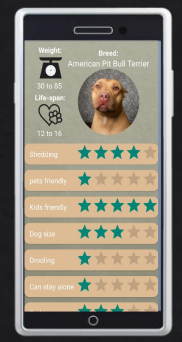


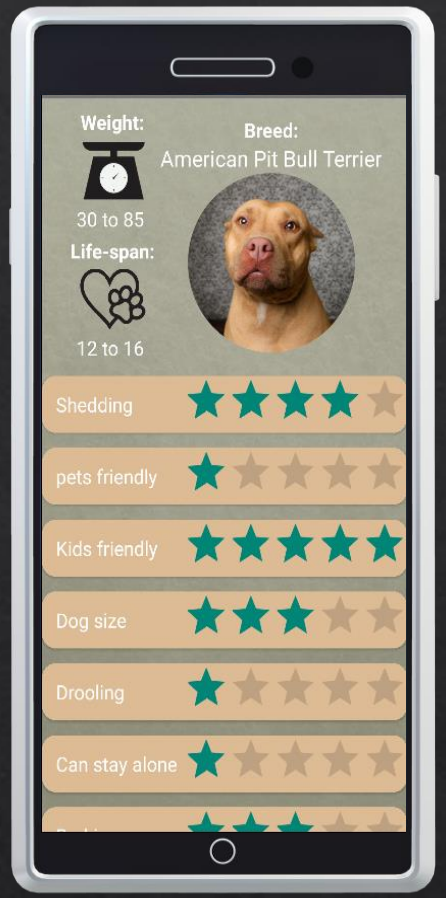
Home screen

Pets for adoption will appear at the home screen, you can scroll and look for your new friend, if you press a pet, you will be taken to a new page with all the information about the pet, you can save him in your favorites to explore more without losing him.



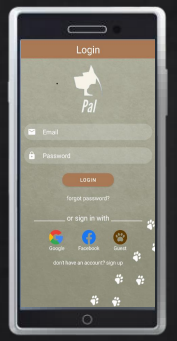


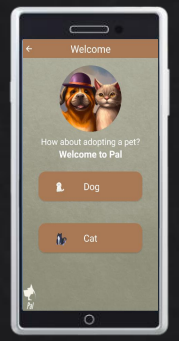


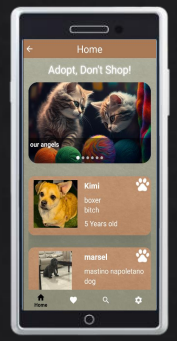


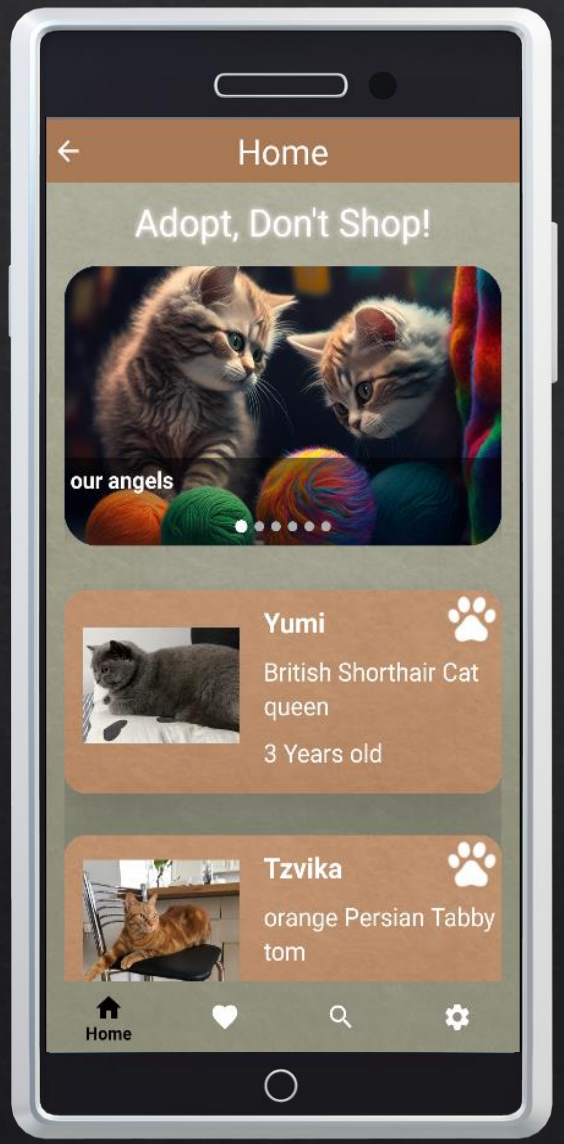
Search screen

On this page you can search specific breeds to find out more information about them, such as if they are kid-friendly or other dog-friendly, if they have any health issues, etc.









Currently, the application supports two languages, English and Hebrew, in which we store our pet information in both languages, as can be seen in the screenshots, the app functions the same in both languages.

Libraries

Retrofit – a type-safe HTTP client for Android and Java that makes it easy to connect to REST web services.

Firebase – a comprehensive app development platform that provides backend services, real-time databases, and user authentication services.

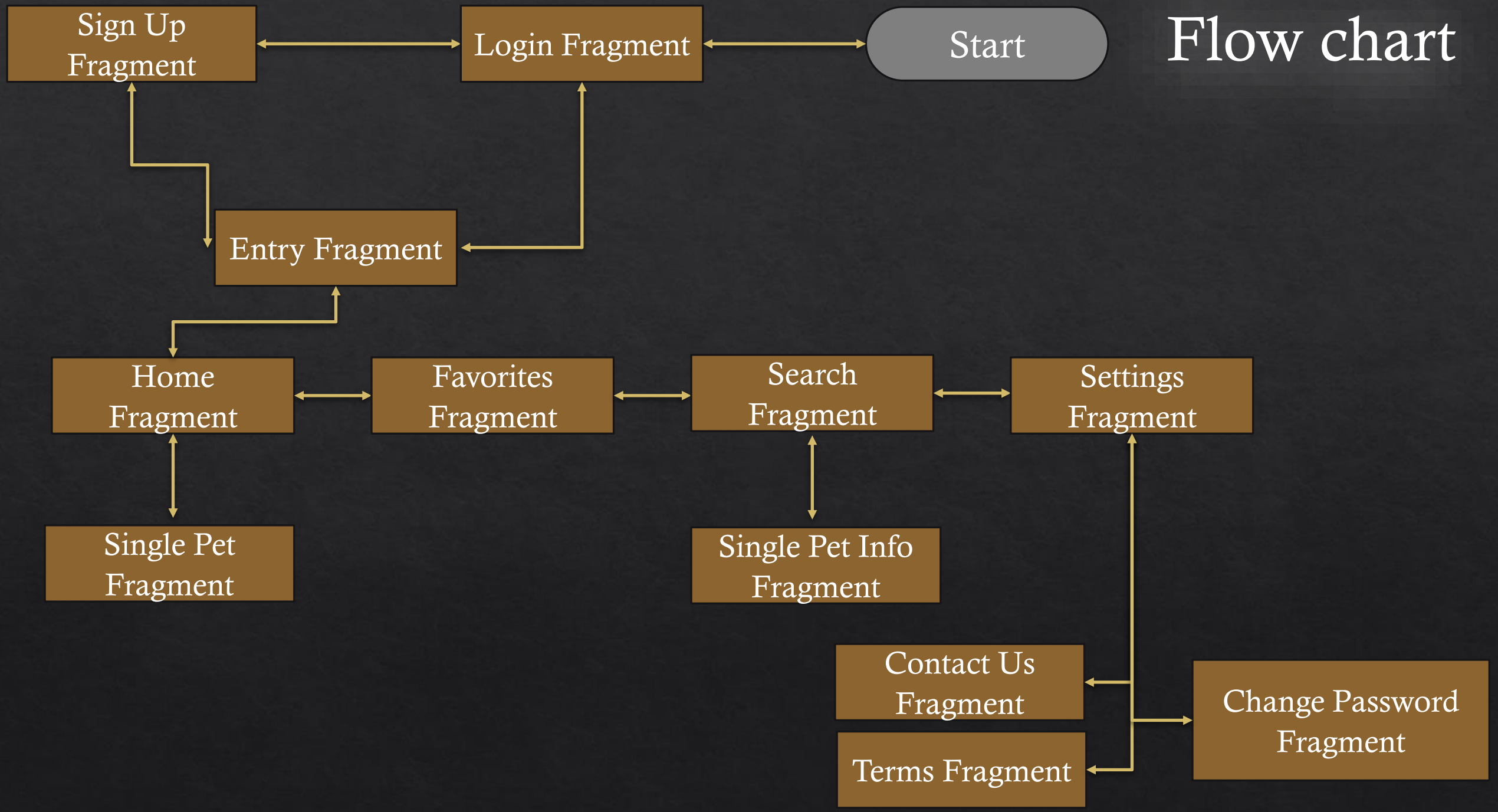
Glide – an image loading library for Android that can handle a wide range of image sources, from local resources to remote URLs.

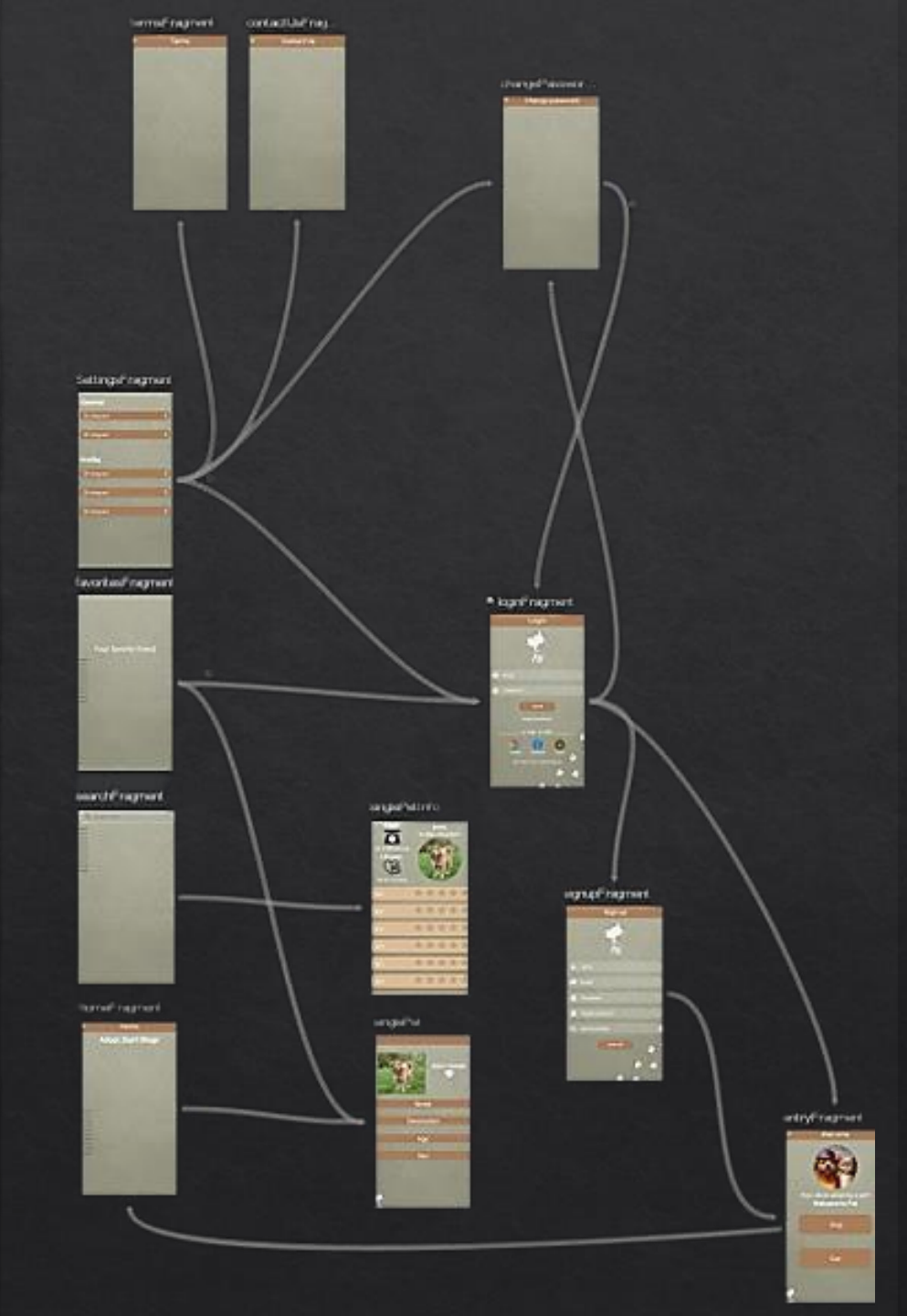
Room – a persistence library for Android that provides an abstraction layer over SQLite to make it easier to work with databases in Android apps.

Hilt – a dependency injection library for Android that provides a standard way to manage dependencies in an Android app.

Lottie File - a library for Android that allows developers to add animations to their apps, made possible by loading and displaying animations created with the Bodymovin plugin for Adobe After Effects.

Flow chart


Sign Up Fragment
Login Fragment
Start
Entry Fragment
Home Fragment
Favorites Fragment
Search Fragment
Settings Fragment
Single Pet Fragment
Single Pet Info Fragment
Contact Us Fragment
Terms Fragment
Change Password Fragment



Navigation

The Navigation component is built on top of the Android Jetpack library and is part of the Android KTX project. It is designed to handle everything from simple navigation to complex navigation, making it easier for developers to manage the navigation structure in their apps.

The Navigation component in Android simplifies the development process by providing a clear and consistent way to manage the navigation structure in your app. This helps you make the right connections between the fragments, making your app much easier to develop.

Design

For our application, we utilized Dalle and Mid Jurnee to generate images. The AI generated stunning images, and we have incorporated some of them into the application.

And for the app architecture we used The Model-View-ViewModel (MVVM) architecture is a pattern for designing Android apps that separates the user interface (view), data (model), and the logic that manipulates the data (view model). MVVM facilitates a clean separation of concerns, allowing developers to focus on writing code for the view model and making it easier to test the app. It also promotes efficient communication between the view and the model through data binding, leading to a more responsive and user-friendly app.

Struggles and solutions

Dogs Breed Info – utilized a web crawling technique using Python to extract information from the dog breed profiles on the https://dogtime.com/dog-breeds/profiles website. The data was then stored in Firebase Firestore, a cloud-based NoSQL database.

Cats Breed Info – our website's API, which was responsible for providing data on cats, had gone down. Thus, we had to gather information about cats and store it on a web server based on Express.js. To ensure that the data was accessible and reliable, the server was hosted on Render, a cloud platform for web applications.

```python
def dogs_stars_info():
    for dog in dogs_breed:
        try:
            time.sleep(1)
            driver.get(f"{dogs_page}{dog}#")
            time.sleep(2)

            # the box that contains the Height Weight & Life_Span.
            dogs_vital_stats = driver.find_element(By.CLASS_NAME, r'breed-vital-stats-wrapper')
            dogs_vital_stats_info = dogs_vital_stats.find_elements(By.CLASS_NAME, r'vital-stat-box')

            dogs_data["Height"].append(dogs_vital_stats_info[1].text)
            dogs_data["Weight"].append(dogs_vital_stats_info[2].text)
            dogs_data["Life_Span"].append(dogs_vital_stats_info[3].text)

            dogs_about = driver.find_element(By.CLASS_NAME, 'breeds-single-intro')
            dogs_about_p = dogs_about.find_elements(By.)

            # the box that contains the rest of the info.
            dogs_quality_table = driver.find_element(By.CLASS_NAME, 'breeds-single-content')
            dogs_quality_stars = dogs_quality_table.find_elements(By.CLASS_NAME, r'characteristic-star-block')

            dogs_data["Apartment_Living"].append(dogs_quality_stars[1].text)
            dogs_data["Tolerates_Being_Alone"].append(dogs_quality_stars[4].text)

            dogs_data["Kid_Friendly"].append(dogs_quality_stars[9].text)
            dogs_data["Other_Dogs_Friendly"].append(dogs_quality_stars[10].text)
            dogs_data["Strangers_Friendly"].append(dogs_quality_stars[11].text)

            dogs_data["Shedding"].append(dogs_quality_stars[13].text)
            dogs_data["Drooling_Level"].append(dogs_quality_stars[14].text)
            dogs_data["Size"].append(dogs_quality_stars[18].text)

            dogs_data["Easy_To_Train"].append(dogs_quality_stars[20].text)
            dogs_data["Intelligence"].append(dogs_quality_stars[21].text)
            dogs_data["Barking_Level"].append(dogs_quality_stars[24].text)


        except:
            time.sleep(1)
            driver.get(f"{dogs_page}{dog}{dogs_page_secoond_type}")
            time.sleep(2)
```

index.js cats.js

index.js

```javascript
const express = require("express");
const app = express();
const { catsList } = require("./cats");

app.get("/cats", (req, res) => {
    const { name } = req.query
    let result = catsList
    if (name) {
        result = catsList.find((cat) => cat.name === name)
        res.status(200).json(result)
    } else {
        res.status(200).json(catsList)
    }
```

References

- https://kotlinlang.org/docs/home.html
- https://www.youtube.com/
- https://www.gool.co.il/
- https://square.github.io/retrofit/
- https://developer.android.com/
- https://github.com/
- https://openai.com/dall-e-2/