GitHub Username: cminhtruong

Dalker

Description

Find the perfect dog lovers who can help you go out with your dogs

Too busy to get out with your friends. Take a look in Dalker, your solution is here. There are always a lot of guys who love this job and for sure they are pet lovers.

PRINCIPALE FEATURE

For dog owners:
- Find a dalker around
- Save the favorite dalker in their list.
- Give a note to review the service

For dalker:
- Create their own profile

In the next version:
- Messaging between dog owners and dalker
- Map view

Intended User

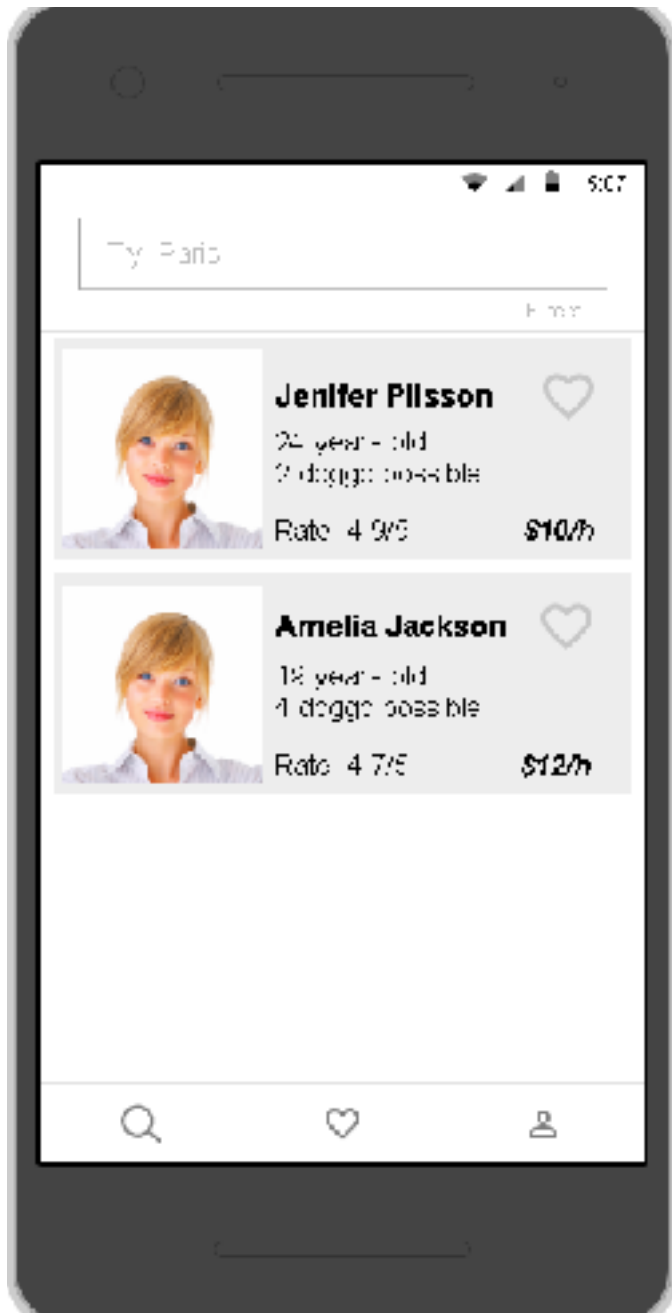Dog owners who are too busy and everyone who loves dogs.

Features

List the main features of your app. For example:

- CRUD dalker profile
- CRUD pet owners profile
- Search by location.
- Write a comment and give a note
- Make a call to dalkers
- Show favorite dalkers in widget

User Interface Mocks

Screen 1

TYPE TO ENTER A CAPTION.

This activity requires a list of service proposing.

Click on favorite icon to add the profile into favorite tab and the widget
Click on profile tab to open the settings activity
Type the location to search the service around, if not it's automatic

Click on filter button to choose filter criteria.

Screen 2

TYPE TO ENTER A CAPTION.

This screen will display a detail of the "dalker" with his/her rate and all reviews.

Click the button review to insert your review.
Click on each item tab to open the detail of the person who proposes the service.
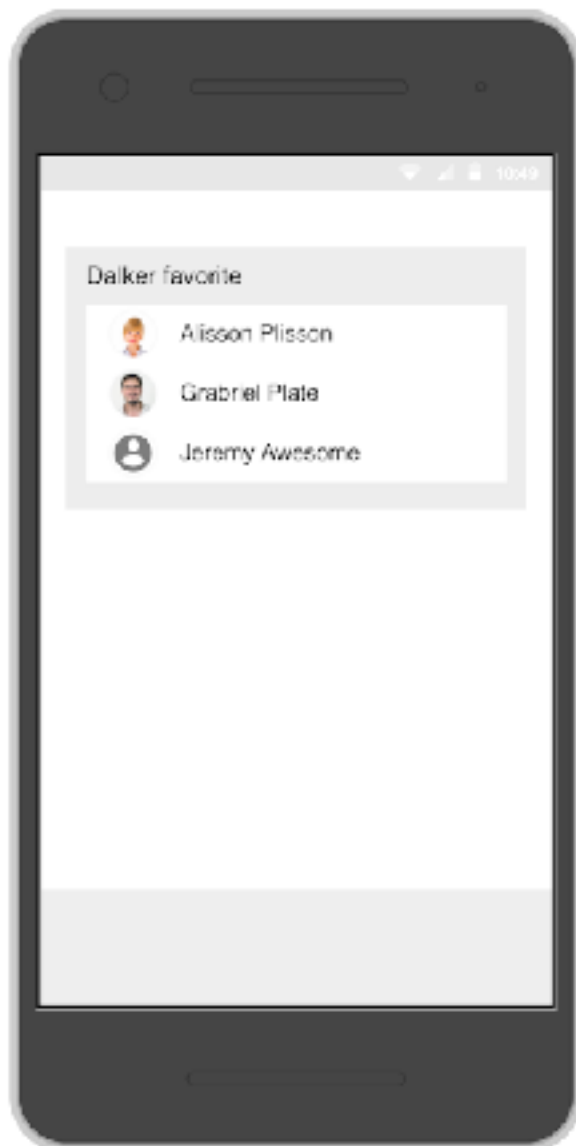Click the button call to make a call.

Screen 3

TYPE TO ENTER A CAPTION.

This screen will display every options to modify the user profile include the profile editor and change the statut to become a "dalker".

Screen 4

Dalker favorite

- 👤 Alisson Plisson
- 👤 Grabriel Plate
- 👤 Jeremy Awesome

TYPE TO ENTER A CAPTION.

This screen will display the widget which contain your favorite dalker profile.
Click on each contact to open the profile detail

Key Considerations

How will your app handle data persistence?

Using Room to store the favorite "dalker".
Using Firebase Realtime Database to store all profiles.
Using the SharePreferences to save the login session

Describe any edge or corner cases in the UX.

The application provides a fixe tab to go to anywhere.

Describe any libraries you'll be using and share your reasoning for including them.

Butterknife v8.8.1
Timber v4.7.1
Gradle v3.1.1
Android Studio v3.1.1
RecyclerView + CardView + NestedScrollView 27.1.0
Image or Photo: Glide v4.7.1
Data persistence: Room v1.1.1
Java language will be used for development
App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.

(may be more …)

Describe how you will implement Google Play Services or other external services.

Describe which Services you will use and how.
I think to use Admod, FCM to push the notification to dalker when they receive a review. The Firebase Database Realtime/ Cloud Firestore is use to store the data.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:
- Configure libraries
- Configure Github repo
- Implement a integration continue tool
- Implement all tasks in Trello
- Configure all Google Play Services using in this application

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:
- Build UI for MainActivity
- Build UI for DetailActivity
- Build UI for ProfileActivity
- Build UI for SettingActivity
- Build UI for LoginActivity

Task 3: Create the LoginActivity

Describe the next task. List the subtasks. For example:
- Create a table in Firebase Database
- Get a connection into Firebase
- Create a SharePreferences which save the session for the next time.
- Think about Google Oauth and Facebook Oauth if have free time.
- If it needs to pull or send data to/from a web service or API only once, or on a per request basis (such as a search application), app

uses an IntentService to do so.

Task 4: Fake data for testing the application

Describe the next task. List the subtasks. For example:
- Insert fake data via randomuser.
- All fake data are required for testing but not for call feature. They'll be removed before pushing in store.

Task 5: Implement widget

Describe the next task. List the subtasks. For example:
- Display and implement UI for the list of dalker's favorite

Task 6: Make the lien between logic business and UI

Describe the next task. List the subtasks. For example:
- Search via location around
- See all detail about the dalker
- Review the service
- Filter the search via criteria
- Pass a call
- Get the notification when receive a review

Task 7: Test UI

Describe the next task. List the subtasks. For example:
- Test all UI possible