GitHub Username: DKolev

Dog walks finder*

*work in progress app name

Description

The main idea here is that our best friends (talking about our dogs, of course) need to socialize with other dogs as much as their owners with other people. We always like to go for a coffee with friends, talking over dinner or going to the movies. For our four legged pets, the only opportunity to meet friends is when we take them out for a walk. And with our busy everyday schedule sometimes we tend to underestimate the need for social life of our dogs, just taking them out for a short amount of time to… you know… do their job.

My app aims to change this by offering the option for the user to schedule a walk (very short, short, medium or long) and invite other users to walk their dogs with him in a preselected area. If the walk is set to public, all users will be able to see it and request to join if they want.
All walks can be sorted by city, dog's breed, duration or neighbourhood.

Intended User

Obviously, the intended users are all dog owners.

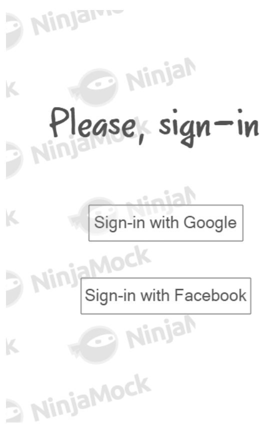
Features

List the main features of your app. For example:

- Creating and displaying a list of events (dog walks)
- Users can add one or more dogs to their pet list (and manage their profile)
- Users can choose to set the walk as private and invite only specific people to it
- Users can request to join in an already created public walk
- See a list of nearby dog playgrounds (usually in the big parks)

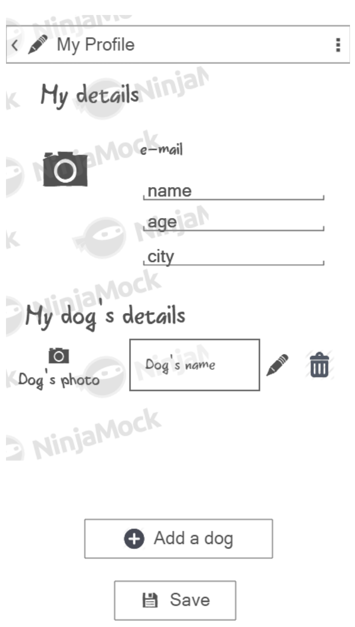
User Interface Mocks

Screen 1



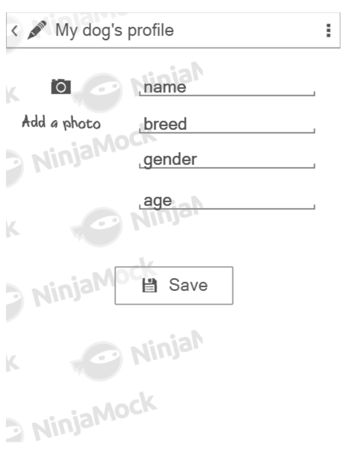
This is the Sign-in activity where the user can choose which service to use (for the sake of simplicity, I'll start just with a Google Sign-in option and later on will add Facebook or some other methods).

Screen 2



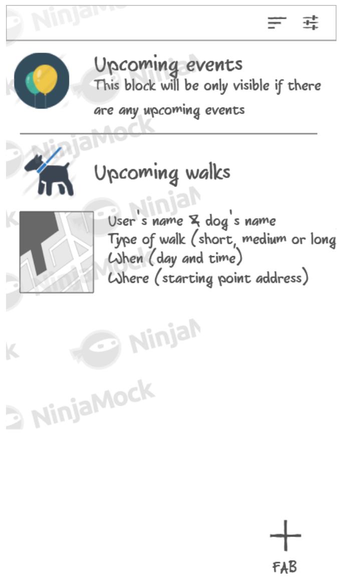
This is the User's Profile screen where the user can complete his profile as well as add one or more dogs to his/her pet list. When a dog is added, its profile can be edited or deleted.

Screen 3



This is the Dog's profile screen where the user can add details about his dog such as photo, name, breed, gender and age.

Screen 4

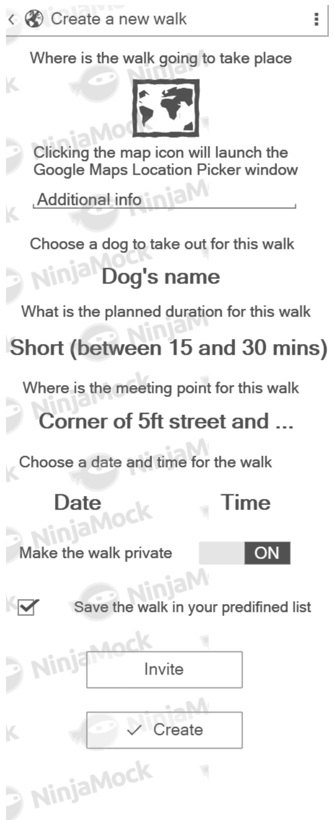

Upcoming events
This block will be only visible if there are any upcoming events
Upcoming walks
User's name & dog's name
Type of walk (short, medium or long
When (day and time)
Where (starting point address)
FAB

This is the MainActivity of the app. Once the user has completed setting his profile, this screen will appear and will show all upcoming public walks created by other users. He/She can create their own walk by tapping on the FAB button.

Here I'm planning a functionality similar to the Inbox app by Google - on the next screen where the user fills in the details about the walks he/she wants to create, there will be an option to save this walk to a predefined list. So when the user taps the FAB button later, he can choose a walk from this list to save time (very useful if for example he often goes for a very short walk with his dog in the same area).

Also, in the App Bar there will be an option to sort the walks by city, neighbourhood, dog's breed or duration.

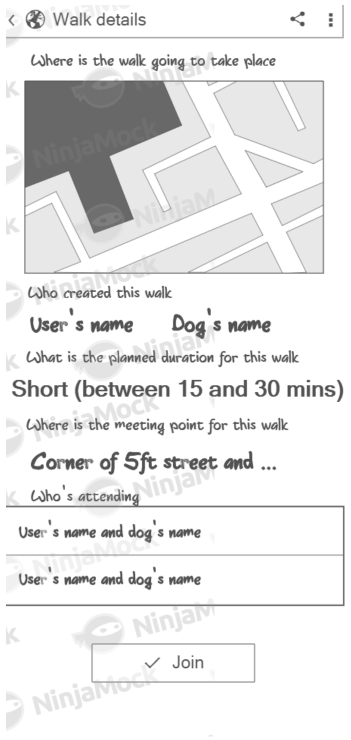
Screen 5


Create a new walk
Where is the walk going to take place
Clicking the map icon will launch the Google Maps Location Picker window
Additional info
Choose a dog to take out for this walk
Dog's name
What is the planned duration for this walk
Short (between 15 and 30 mins)
Where is the meeting point for this walk
Corner of 5ft street and ...
Choose a date and time for the walk
Date
Time
Make the walk private
ON
Save the walk in your predifined list
Invite
Create

This is the screen where the user can create a new walk. He can choose:

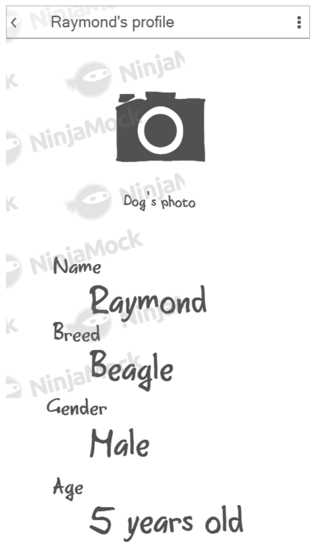
- The location (using Google Location Picker… for example Central Park),
- To add additional info for the walk
- One or more of his dogs to take out
- The planned duration for the walk
- The meeting point for the walk
- Date and time for the walk
- The option to save the walk to a predefined list (for quick creation later)
- If the walk is public or private
- To invite specific users to the walk (they should get a notification)

Screen 6


Walk details
Where is the walk going to take place
Who created this walk
User's name
Dog's name
What is the planned duration for this walk
Short (between 15 and 30 mins)
Where is the meeting point for this walk
Corner of 5ft street and ...
Who's attending
User's name and dog's name
User's name and dog's name
Join

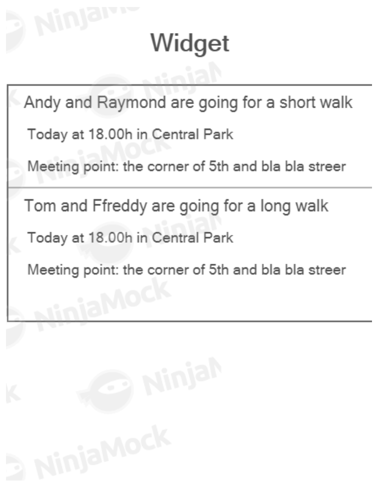
This is the Details screen for an already created walk. When a user clicks on a public walk from the list (in the MainActivity), this screen will show all the details about it and also an option to request to join the walk.

Screen 7



When a user clicks on a dog's name from a details screen about an already created walk, he/she can see additional info about this dog (like breed, gender, age).

Screen 8


Widget

Andy and Raymond are going for a short walk
Today at 18.00h in Central Park
Meeting point: the corner of 5th and bla bla streer

Tom and Ffreddy are going for a long walk
Today at 18.00h in Central Park
Meeting point: the corner of 5th and bla bla streer

The app's widget should display a list of walks and the user should be able to customize it (showing only walks in his area or walks with dogs from a particular breed). Tapping on a list item (a walk) should open the detailed info about it where the user can request to join the walk.

Key Considerations

How will your app handle data persistence?

I'm going to use Firebase Realtime Database to store additional info about each user as well as storing each walk a user creates and all details related to it.

Describe any edge or corner cases in the UX.

I'm not sure if this is the place to mention it but I'm planning on using a hamburger menu as a main way for the user to navigate through the app with the following sub-menus:

- Like in the G+ app - a profile pic of the user and a smaller one for his dog/s
- Profile - to edit his profile info
- Public Walks - to display all available public walks

- Events - to see all big future events (like a monthly group meetings and etc.)
- Playgrounds (list of special areas for the dogs in the public parks)
- Lost / Found (I have this idea to include a place where people can post about lost/found dogs though I'm not sure if I'm going to complete this for the Capstone Project or leave it for the future to work on).
- Settings - to edit settings like a default sorting of the walks in the MainActivity, syncing interval and others

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

- Glide for handling the loading and caching of images
- Butterknife to avoid some boilerplate code
- Firebase Authentication (com.google.firebase:firebase-auth:11.6.2) for authenticating the user
- Firebase Realtime Database (com.google.firebase:firebase-database:11.6.2) to handle writing and reading data (for user's profile and the created walks)
- CardView (com.android.support:cardview-v7:27.0.2) to display a group of public walks together (or a single walk, I'm not yet sure where exactly I'm going to need this)
- RecyclerView (com.android.support:recyclerview-v7:27.0.2) to display each walk in a list, as well as other elements
- Design Support Library (com.android.support:design:27.0.2) to add some material design components as FAB and others
- AppCompat Library (com.android.support:appcompat-v7:27.0.2) for the ActionBar and AppCompatActivity
- ConstraintLayout (com.android.support.constraint:constraint-layout:1.0.2) as a main parent layout for all activities
- FirebaseJobDispatcher (com.firebase:firebase-jobdispatcher:0.8.5)
- Android Architecture Components (ViewModel and LiveData)
- Firebase UI

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

- Google Account Login (com.google.android.gms:play-services-auth:11.6.0) so the user can log in with his/hers Google account
- Google Location and Activity Recognition (com.google.android.gms:play-services-location:11.6.0) to allow the user to use the Google Place Picker for the location of the walk

Next Steps: Required Tasks

I’m going to divide the work on this project into 3 big parts/stages:

- **Stage 1 -** here I’m going to work on everything concerning the user profile (login with Google, adding details and dogs to his/hers list of pets and saving this into the Firebase Realtime Database.
- **Stage 2 -** here I’m going to work on everything concerning creating, storing and querying the database for walks.
- **Stage 3 -** overall UI polish, making the app Material (add some simple animations), make the widget

Stage 1:

Task 1: Project Setup

- Configure libraries

Task 2: Implement UI for the Sign-in Activity, the User’s profile Activity and Adding a new dog Activity (only for phones, I’m going to add tablet layouts when I have everything working on a phone)

- Build UI for the Sign-in Activity
- Build UI for the User’s profile Activity
- Build UI for the Adding a new dog Activity

Task 3: Authenticate Using Google Sign-In on Android

(following this guide https://firebase.google.com/docs/auth/android/google-signin)

- Go to the Firebase console and connect the app the a new Firebase project
- Enable Google Sign-in in the Sign-in Methods tab
- Integrate Google Sign-in following this guide https://developers.google.com/identity/sign-in/android/sign-in

- Follow the rest of the steps

Task 4: Add additional info about the user in the User's profile Activity

- Use an instance of FirebaseAuth to get the user's account data (like user's name, email and photo URL) and update the corresponding views

Task 5: Set up Firebase Realtime Database

(following this guide https://firebase.google.com/docs/database/android/start)

Task 6: Structure the data

(following this guide https://firebase.google.com/docs/database/android/structure-data)

- Make the JSON tree (just the part concerning user's profile)

Task 7: Set the rules for the Firebase Realtime Database

(following this guide https://firebase.google.com/docs/database/security)

- .write - Writing access should have only the currently authenticated user
- .read - Reading access should have every user who is authenticated
- I should also use .validate to make sure each value is in the right format:
  - * name, e-mail, user's profile pic URL, city are STRINGS; age is INTEGER;
  - * dog's name, breed, gender are STRINGS; age is INTEGER
- Try the rules with the Simulator and make sure they work as intended

Task 8: Create a ViewModel class for the User's profile activity

Task 9: Create a ViewModel class for the Adding a new dog Activity

Task 10: Create a model class Dog file with all the details needed for the dog's profile

Task 11: Verify that adding a dog to the user's pet list works

- Add a dog's photo and load it into the ImageView following this guide https://github.com/firebase/FirebaseUI-Android/blob/master/storage/README.md
- Add the rest details about the dog
- Save the dog to the Realtime Database
- Going back to the User's profile Activity should automatically display the dog so:

- Query the Database if the user has added a dog and use RecyclerView adapter to display the dog's name (one or more) (following this guide https://github.com/firebase/FirebaseUI-Android/blob/master/database/README.md)

Task 12: Try everything on a phone

- Install the app on a phone and try if everything works as it should
- Sign-in
- Add additional info about the user
- Save the user's profile and add it to the Firebase Realtime Database (verify that the entry is in the database through the console)

Stage 2:

Task 1: Implement UI for the Main Activity (displaying the walks using RecyclerView), for the CreateANewWalk Activity, DisplayAWalk Activity and the hamburger menu

- Build UI for the Main Activity
- Build UI for the CreateANewWalk Activity
- Build UI for the DisplayAWalk Activity
- Build UI for the hamburger menu

Task 2: Create a ViewModel class for the Main Activity activity

Task 3: Make the FAB button launch the CreateANewWalk Activity

Task 4: Structure the data

- Add a new node to the JSON tree for the walks and then all the fields which a walk should have (place, user's name, dog's name, starting point, time and date etc.)

Task 5: Set the rules and validate the values

Task 6: Create a ViewModel class for the CreateANewWalk Activity

Task 7: Add a Google Place Picker

(following the steps in this codelab https://codelabs.developers.google.com/codelabs/fire-place/index.html?index=..%2F..%2Findex#0)

- Register my app on the Google API Console and obtain a Google API Key
- Add the Place Picker
- Store the picked place into a variable which later will be saved to the Realtime Database

Task 8: Store the additional info about the walk into a variable which later will be saved to the Realtime Database

Task 9: Add the option for the user to choose a dog from his/hers pet list

- Query the database for the user's pet list
- When I get the dog/s name/s store them/it into a variable and use that to display a dropdown menu so the user can choose from it

Task 10: Add the option for the user to choose the length of the walk

- Make a list (very short, short, medium or long walk) with additional description for each and use it to display a dropdown menu
- Store the chosen option into a variable which later will be saved to the Realtime Database

Task 11: Add the option for the use to choose a meeting point for the walk

- Add another Place Picker
- Store the picked place into a variable which later will be saved to the Realtime Database

Task 12: Add the option for the user to choose time and a data for the walk

(following this guide https://developer.android.com/guide/topics/ui/controls/pickers.html)

- Add the time and data pickers
- Store the two values into a variables which later will be saved to the Realtime Database

Task 13: Add the option for the user to choose if the walks he/she is creating should be public

- Add a switch toggle which will return a value of TRUE or FALSE
- Store the value into a variable which later will be saved to the Realtime Database and will be used when querying for walks in the Main Activity (only public walks should be displayed for all users, but if the walk is private it should appear only for the user who created it in a separate section called My Walks)

Task 14: Add the option for the user to save this walk to his/hers list with predefined walks

- Add a child for this list in the Realtime Database

- If the user wants to save the walk a popup window will appear asking for a name for this walk

Task 15: Install the app and save the walk to the Realtime Database

- Clicking on the Create button should save the walk to the Realtime Database with all its details
- Verify that the walk is saved through the Firebase Console

Task 16: Invite people

- When the creation of a walk works go back and work on inviting other users
- When the user clicks on the Invite button, a new Activity will open where he/she can search the database for other users based on name or email (this option will show only the users who have the app installed). The invited user should receive a notification
- Otherwise, I may add a share button to share the walk details with other services (email, Messenger or other)

Task 17: Implement the UI for the RecycleView single walk item

- It should contain the user's name, dog's name, place for the walk, meeting point, date and time, and (maybe) a small picture of the location on the map

Task 18: Create a model class Walk file with all the details needed for a single walk (like user, dog, place, data & time etc) and the getters and setters methods

Task 19: Create a WalkAdapter which extends FirebaseRecyclerAdapter

(following this guide
https://github.com/firebase/FirebaseUI-Android/blob/master/database/README.md)

- Handle the click - when a user clicks on a walk, a new Activity should open and display the details about this walk
- I may remove the detail about who is attending the walk and the request to join button (or implement them later on). After all, if a user creates a public walk, the presumption is he/she wants other users to join.

Task 20: In the MainActivity attach the adapter to the RecyclerView and set a LayoutManager and see if everything works

Task 21: Manage the lifecycle of the FirebaseRecyclerAdapter

- Automate the process using Android Architecture Components and LifecycleOwner

Task 22: Handle the new Data and error events

- Override onDataChanged() and onError() methods in the adapter

Task 23: Create a Firebase JobDispatcher

https://github.com/firebase/firebase-jobdispatcher-android

Stage 3:

Task 1: Make sure everything else works - from creating and editing user's profile to creating, editing and displaying a walk & etc.

Task 2: Go to https://material.io/guidelines/ and start reading. Respect the guidelines

Task 3: Make some simple animations using this guide

https://youtu.be/OHcfs6rStRo

Task 4: Optimize the layout for tablets