

walkiewalkie

An app for professional dog walkers

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Dog/Owner Management Screen](#)

[Owner Details](#)

[Dog Details](#)

[Start Walk](#)

[Walk Status screen](#)

[Walk widget](#)

[Walk completed](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any edge or corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services or other external services.](#)

[Next Steps: Required Tasks](#)

[Tasks](#)

[Project Setup](#)

[Implement UI for Each Activity and Fragment](#)

[Create the data model](#)

[Implement adding Dogs and Owners](#)

[Implement walks](#)

[Implement location tracking](#)

[Implement in-walk reminders](#)

[Implement the walk widget](#)

GitHub Username: macarus0

WalkieWalkie

Description

This app makes it easy for dog walkers to manage their dogs and customers. Easily send walk summaries with photos as you walk, keeping your clients informed and happy.

Intended User

This app is for people who walk dogs for a living. Dog walkers are used to give dogs exercise while their owners are away from the home and often do not interact with the dog owners, just the dogs. Dog walking customers want to see pictures and read reports of what kind of walk their dog took. Sending these updates consistently keeps customers happy and encourages referrals.

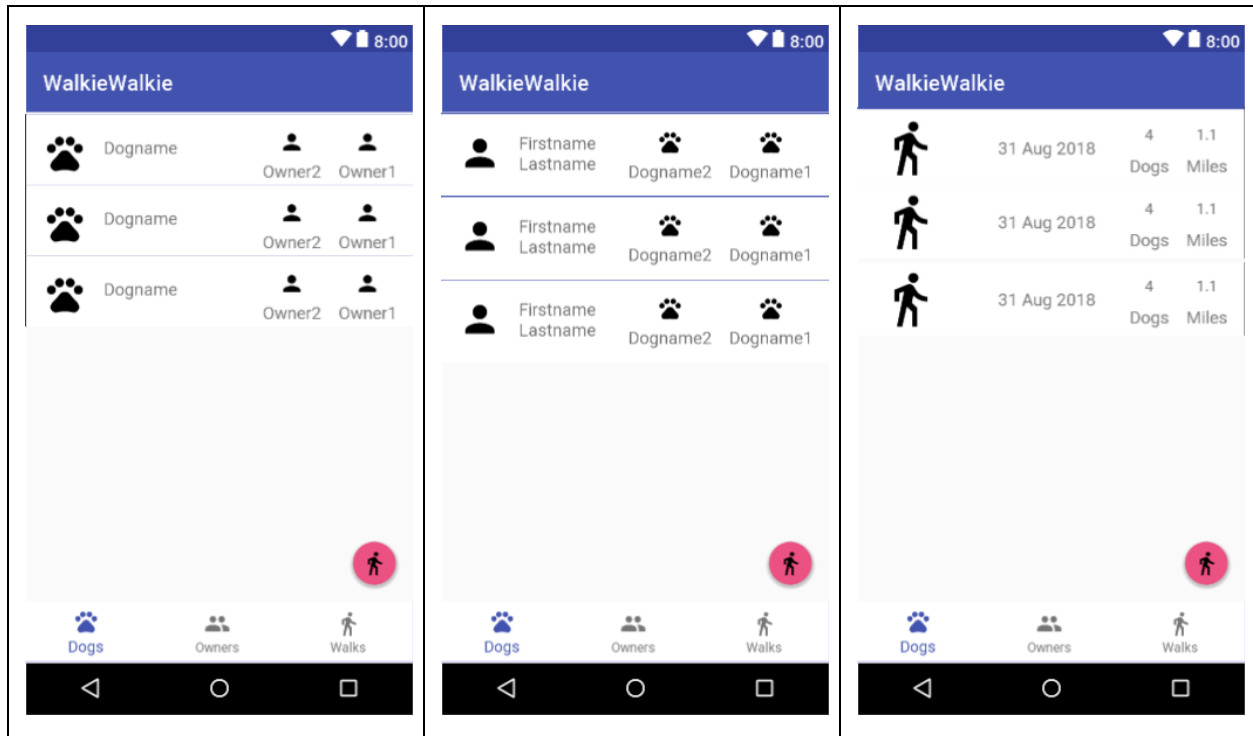
Features

Stores photos of the dogs, their owner(s), their address, contact information, and any special notes. Group dogs together on a walk, track the walk's distance and set reminders to take photos. Once the walk is complete, WalkieWalkie generates a summary that you can send to all of the owners so they know that their pet was taken care of.

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Dog/Owner Management Screen



Lists out dogs and owners

- FAB starts a walk and launches 'Start Walk'
- Clicking a dog/walk/owner shows details screen
- Clicking bottom bar select dog/walk/owner list

Owner Details



The image shows a mobile application interface for "WalkieWalkie". At the top, there is a blue header bar with the app name "WalkieWalkie" in white. Below the header, the interface is divided into several sections. The first section contains a black silhouette of a person's head and shoulders, followed by two input fields labeled "Firstname" and "Lastname". The second section contains a phone icon followed by the number "213-555-1212". The third section contains an email icon followed by the address "email@example.com". The fourth section contains the address "123 Some Street", "City, State", and "12345". The fifth section contains two identical white boxes, each with a black paw print icon and the label "Doggy" below it. At the bottom of the screen, there is a black navigation bar with three white icons: a back arrow, a circle, and a square.

WalkieWalkie

Firstname

Lastname

213-555-1212

email@example.com

123 Some Street
City, State
12345

Doggy

Doggy

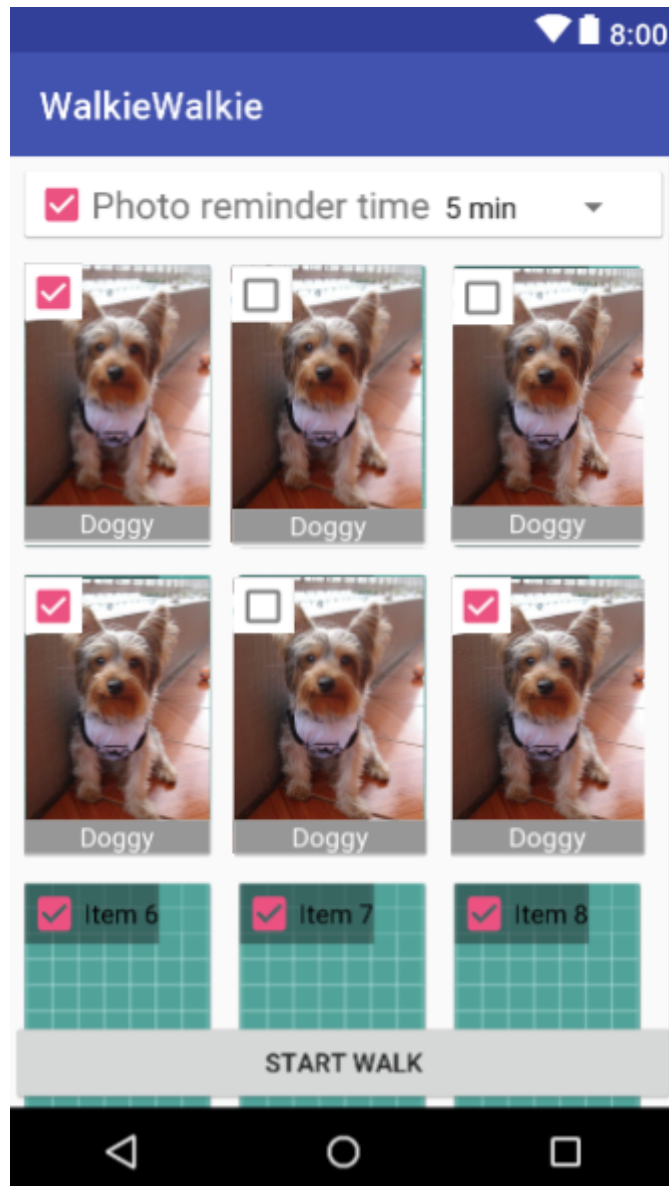
Edit the owner contact information and associate dogs with the owner

Dog Details



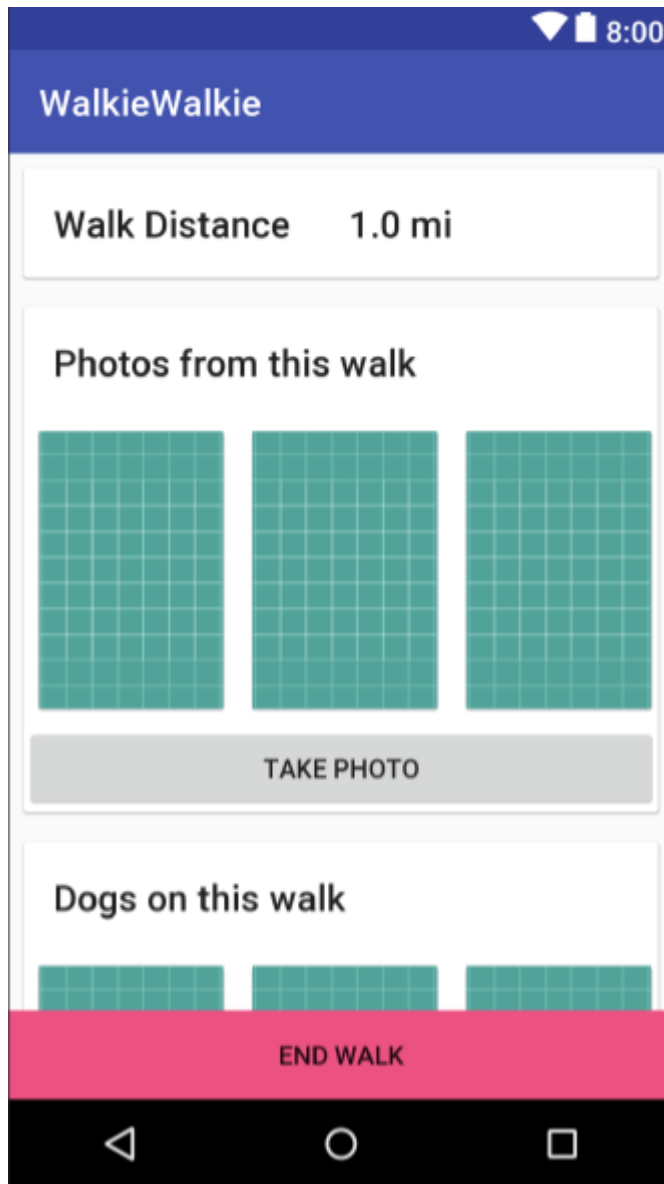
Edit information about the dog and associate owners with the dog.

Start Walk



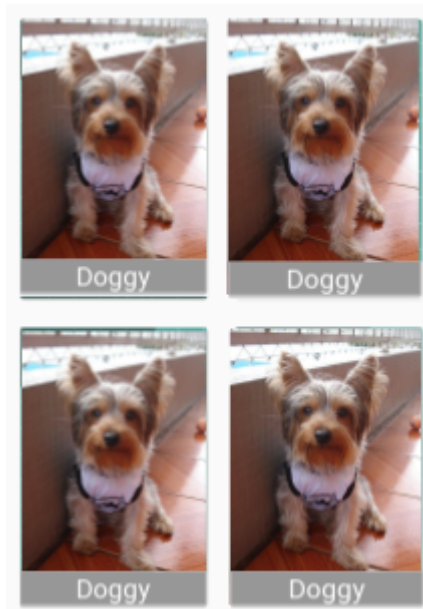
- Select the dogs that will go on the walk
- Set the timer for notifying you to take photos

Walk Status screen



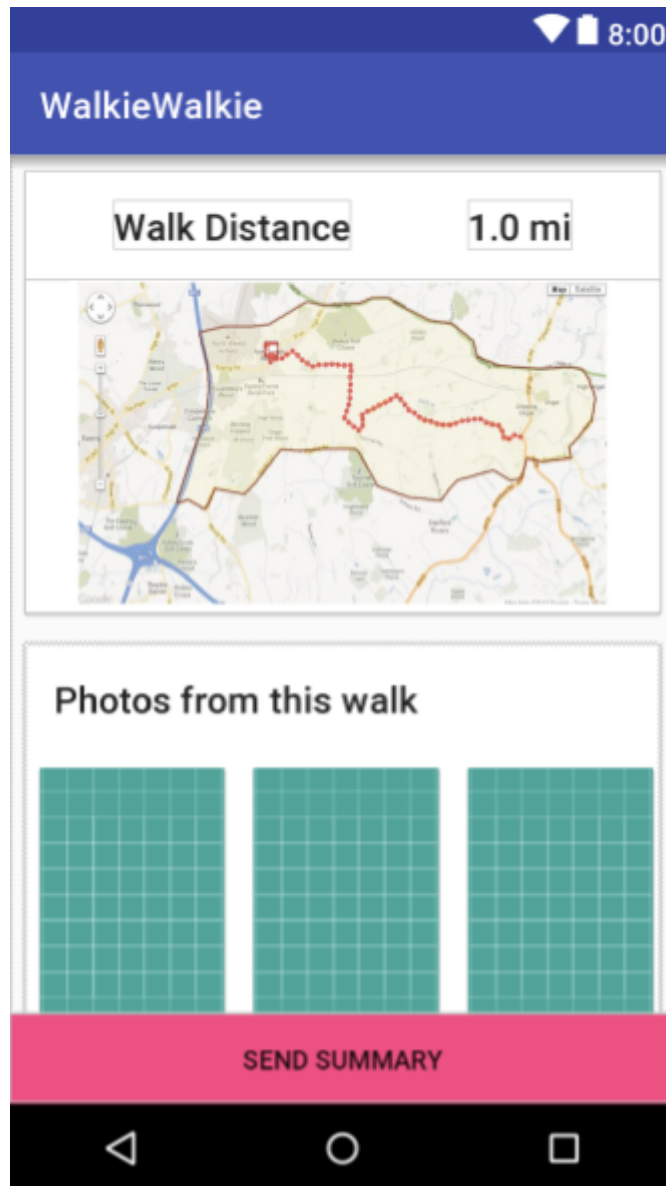
- Summary of dogs on walk
- Distance traveled
- Take Photo
- Complete walk
- During a walk, a widget will list the dogs that are on the walk with their photos

Walk widget



Tapping on a dog will bring up the dog contact activity

Walk completed



- Date of walk
- Distance
- Duration
- Dogs
- Photos
- Option to Send Summary

Key Considerations

How will your app handle data persistence?

A Room database to store the customer information, photos, and records of walks that the dogs went on.

Describe any edge or corner cases in the UX.

Hitting the back button on the walk summary screen should take you back to the walk listing screen.

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

Room: Data persistence

ConstraintLayout: Used to make custom card layouts

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

Location: Tracks distance of walks and uses that information for sharing and record keeping.

Analytics: Log how many walks are taken, how many dogs are added per walk, and how many email summaries are sent.

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.

Tasks

1. Project Setup

Configure the necessary libraries and create a dummy main Activity.

2. Implement UI for Each Activity and Fragment

Create UI for the following activities:

MainActivity

CustomerDetails

EditCustomer

EditDog

WalkHistory

WalkDetails
WalkStart
WalkStatus
WalkComplete

3. Create the data model

Add support for stored data:

Dogs
Owners
Walks

4. Implement adding Dogs and Owners

Implement the flow to add a new dog and associate it with owners or add a new owner. This involves storing photos and the relationships between dogs and owners

5. Implement walks

Implement the flow for starting a walk, adding dogs to it and concluding it with a summary page. Store the walks and list them on the main page with a walk summary.

6. Implement location tracking

Add walk distance to the status page and a path graphic to the summary page.

7. Implement in-walk reminders

Add a notification as a reminder to take photos during the walk

8. Implement the walk widget

Add a list of current dogs on the walk to the home screen

Submission Instructions

If using GitHub:

- Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
- Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”