MAN'S BEST FRIEND

Software Requirement Document

Created by:

Andrew Wallace

Joey Dyer

Uday Cherukuri

Disclaimer: This document is incomplete. More diagrams and application detail will be added in the future.

Table of Contents

1. Introduction:

Man's Best Friend is an Android app designed to assist users with taking care of their dog. This includes anything from helpful reminders to location features that aim to create a dog owner community where users can mutually benefit from each other. The basic principle of Man's Best Friend is that your dog has a profile in which information for general health care is based on their age, weight, breed, and previous experience.

2. Glossary:

Breed **-** a specific group of domestic animals having homogeneous appearance (phenotype), homogeneous behavior, and/or other characteristics that distinguish it from other organisms of the same species and that were arrived at through selective breeding.

Profile **-** Data component that encapsulates all information associated with a specific dog. This includes picture, age, weight, breed, allergies, medical conditions, etc.

Vaccine **-** Medication periodically given to dogs in order to keep them disease free and healthy.

3. Functional Requirements:

a. FR1: Profiles

Description:

Users should be able to create a dog profile. A blank template will be presented for the user to manually enter information.

Details:

- Profile features include biography, pictures, general information (age, breed, sex, etc.)
- Ability to support multiple profiles on a single device
- Dog profiles should be shareable to other users. On each profile there will be a share button that, when prompted, will ask the user to enter another user's name and share the profile with that user. Ex. User A clicks the share button on Spot's profile and enters User B's username. User B then receives a notification that User A has shared Spot's profile with them

- There will be a section of the app where a user can look at all the dog profiles that have been shared with them

b. FR2: Reminders

Description:

Users should be able to pick reminders. They will pick these from a list of available reminders, as well as be able to add custom reminders of their own

Details:

- Reminder frequency should be customizable from every n hours to every m years
- Reminders should have an optional message to display. Ex: “Medication Reminder: Spot needs the antibiotic today.”
- Reminders should be tied to an existing dog profile

c. FR3: Dog Database

Description:

The app should contain basic healthcare information for the top 50 most popular dog breeds. This will act as a suggestion for each dog’s profile.

Details:

- General information should be offered at the time of certain preset reminders that is relevant to the individual reminder. Ex: when the feeding reminder activates for a chihuahua profile, also display the information “Professionals recommend feeding chihuahuas puppy food due to higher nutritional value”.
- Informational reminders should have a customizable frequency setting, including “Always”, “Frequent”, “Less Frequent”, and “Never”

d. FR4: Barks

Description:

Users should be able to create and receive “Barks” (customized messages) from other users in their local area. These are messages sent out to every user within range that has their bark setting turned on.

Details:

- Barks must be labeled under a certain category such as adoption, safety, fun, etc.
- Users should be able to customize the distance threshold for receiving barks around them at the intervals 5, 10, 25, and 50 miles
- Users will have the option to like a bark, and each bark will display the current number of likes it has received
- Users will be able to filter the types of barks they see based on category and/or importance (5+ likes on a bark makes it important)
- There will be a section of the application that displays all barks within the past week, which can be ordered by category, number of likes, or recency

4. Non-Functional Requirements:

NFR 1: Information for dog breeds should be accurate.

NFR 2: The app should keep profile data isolated from other profile data.

NFR 3: The app should make efficient use of battery life.

5. Hardware Requirements:

- Android device with Android 6.0 (Marshmellow) or greater.
- Network communication for “Bark” feature.

6. Design Specifications:

a. Components

Main Menu - Show list of profiles and option to create a new profile.

- New Profile Button - Open new profile menu with blank template
- Previous Profile Selection - Opens the main menu for a previously saved profile

New Profile Menu - Show list of fields to be entered (Should specify what fields here). Contains back button.

- Finish Button - Adds profile to list of available profiles.

Profile Main Menu - Show buttons that will lead to the different functionalities of the app. Also contain a back button to select a different profile.

- Add New Activity Button - Opens up a list of fields where users can input reminder title, reminder frequency, and reminder optional message. Appears at the bottom of the list of reminders
- View Reminders Button - Takes user to a view of all reminders displayed in a list format
- View Barks Button - Takes the user to a view of all Barks within range within the last week. At the top of this view there are three buttons: Category (drop down list of categories which user will check or uncheck), Most Liked (will sort all currently shown barks by number of likes), and Most Recent (will sort all currently shown barks by time of creation)
- View Info for Dog Button - Takes the user to a screen where information about their dog and their dog's breed are displayed.

View Activities Menu - Shows previously logged activities for the current profile. Contains Back Button.

Back Button - Switches the current screen to the previously displayed screen.

b. System Operation

Application Initialization - Application state data is loaded from the Android device's hard drive and the login screen is displayed. This occurs on application launch

New Profile Button Pressed - New profile object is initialized. This object contains variables which will store the different information stored by the user. Once the user has completed the form, this profile object is added to the app's list of profiles.

New Profile Finish Button Pressed - The current new profile object is added to the user object's list of profiles. This data is to remain persistent.

User Selects Profile - Previous profile state data is loaded. After this, a main menu is launched which will be linked to this profile.

Add New Activity Button Pressed - A new activity object is initialized. Once the user has completed filling the form, this object is added to the profile object's list of activities.

View Reminders Button Pressed - Get profile object's list of reminders. Then, create new View Reminders object which will be populated by the profile's list of reminders.

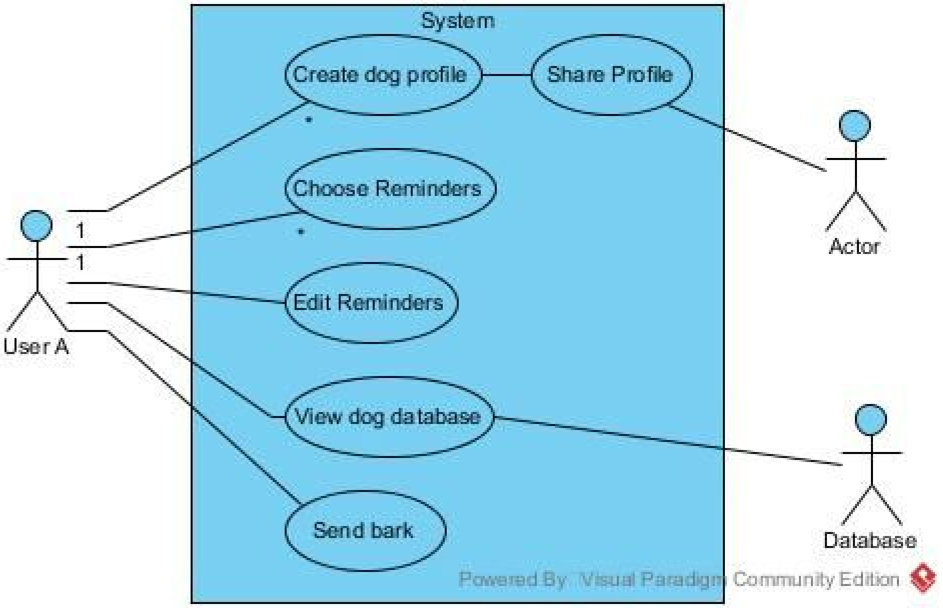
View Barks Button Pressed - Initialize a View Barks Object.

View Info for Dog Button Pressed - Initialize Info object. Then, get current profile's data that was entered by the user and add to the Info object. After this, if information is available for that specific breed, also add this data to the Info object.

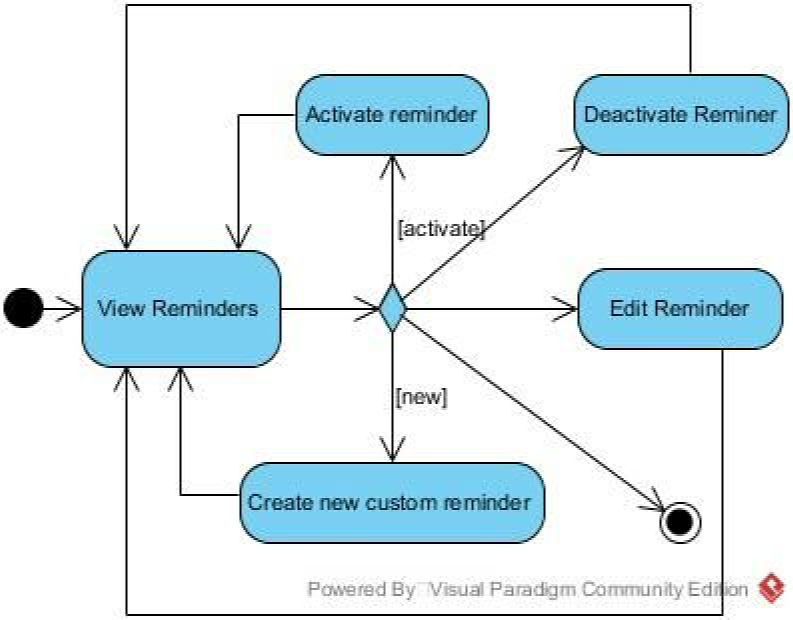
Back Button Pressed - Close current screen and make previous screen active. Should be implemented as a stack.

7. Software Models:

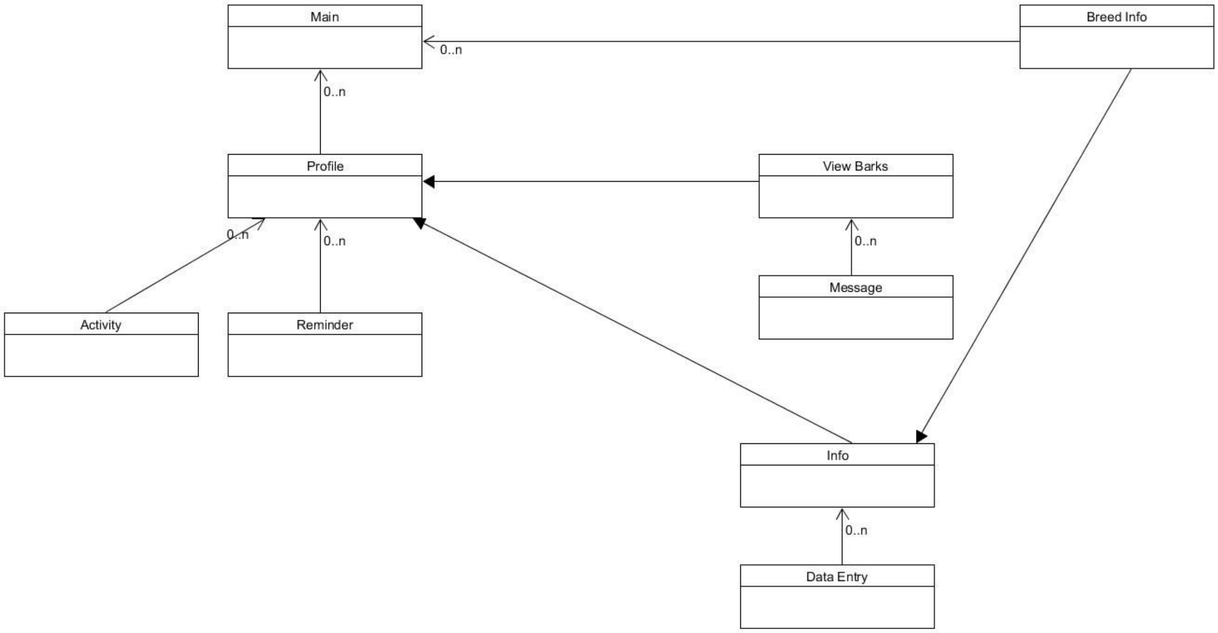
a. Use Case Diagram:



b. Activity Diagram (Reminder):

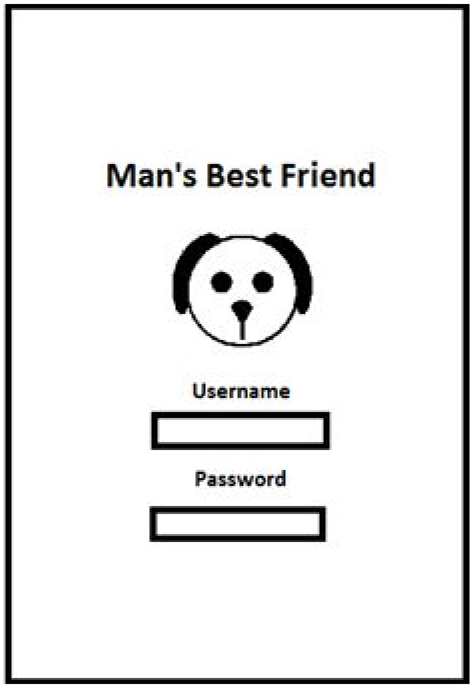


c. Basic Class Diagram:

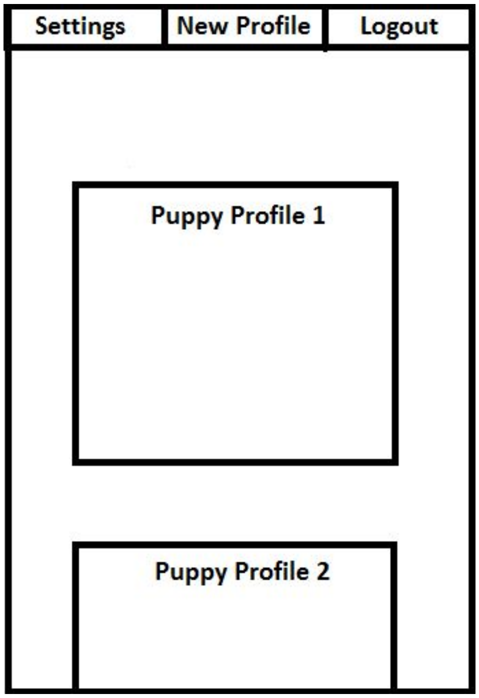


8. UI Layout:

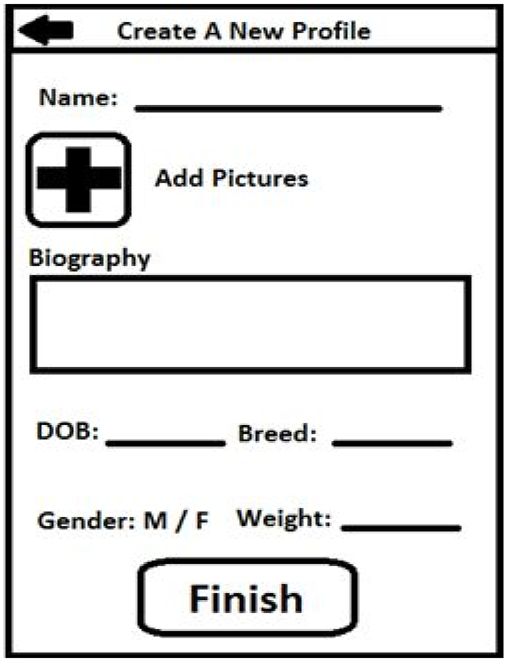
Login Screen:



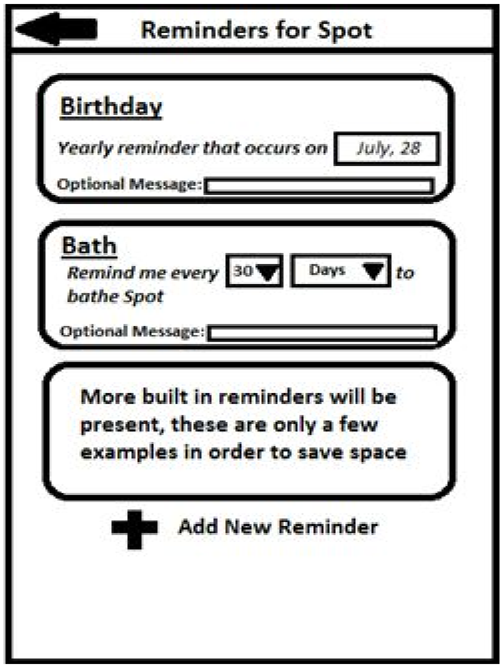
Profile Selection Screen (most likely also the main menu):



New Profile Creation Screen:



Reminder Screen:



Yearly reminder that occurs on July, 28

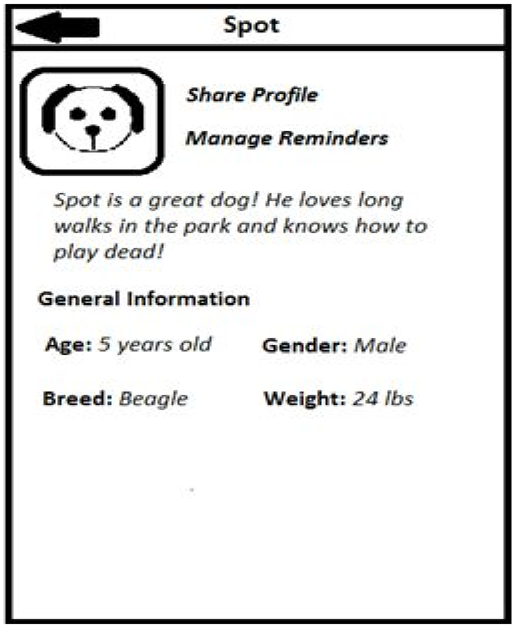
Optional Message:

Remind me every 30 Days to bathe Spot

Optional Message:

More built in reminders will be present, these are only a few examples in order to save space

Profile Screen:



Spot is a great dog! He loves long walks in the park and knows how to play dead!

General Information

Age: 5 years old

Gender: Male

Breed: Beagle

Weight: 24 lbs

Credits

puppy by John Winowiecki from the Noun Project