

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Profile summarize view](#)

[Profile view](#)

[Key Considerations](#)

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

[Describe any 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.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Implement Each Activity and Fragment](#)

[Task 4: Persistence layout](#)

[Task 5: Google Map implementation](#)

[Task 6: Create Widget](#)

GitHub Username: jyodroid

Eternal Friend

Description

Pets are important part of our lives, once they enter in our home, they become part of our family.

Our pets need more than only love; they need a healthy alimentation, they need vaccination plan, they need adequate hygiene habits and with an occupied agenda we usually forget take care of our friend needs.

The purpose of this app is to bind our eternal friend (Pets) with us using the technology. Is a moment's logs where we will store pictures with classify pictures by age and we can share each picture, medical history, remainders of our best friends needs. It also will include notifications and a widget with new york times "Pets" topic articles link, and use google maps to locate veterinary hospitals, pet shops and boutiques to buy food and toys. We can have a profile for several pets.

Intended User

The App is intended to be used for pets owners.

Features

- Create a pet profile
- We can add and remove pet profiles
- Take pictures of your pet and classify by age.
- Share pictures.
- Stores pet medical history
- Notifications of walk time and feed time.
- Widget with [The New York Times](#) “Pets” topic last article link.
- Vaccination plan with Google calendar integration.
- Locate nearest (5 kilometers radio) veterinarian clinic and pet shop.

User Interface Mocks

All views will contain a floating action button (Search button) that will show in a map the nearest veterinarian hospitals and pets shops.

Profile summarize view



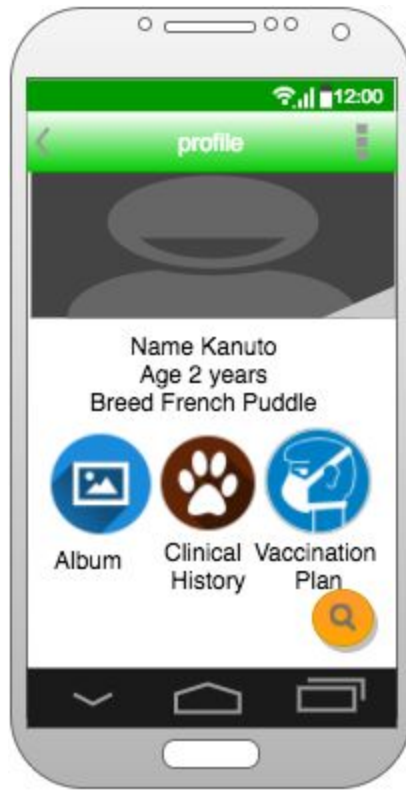
This view allows the user to see his different pets profile summarized and also can select a profile, create a new one or delete any profile. This view will be main view and will **NOT HAVE BACK BUTTON**. The “+” floating action button will leads to “Add pet profile view” and the “x” icon in each profile will popup a dialog with ok cancel option and the text “Are you sure you want to delete _pet name_ profile”

Add pet Profile view



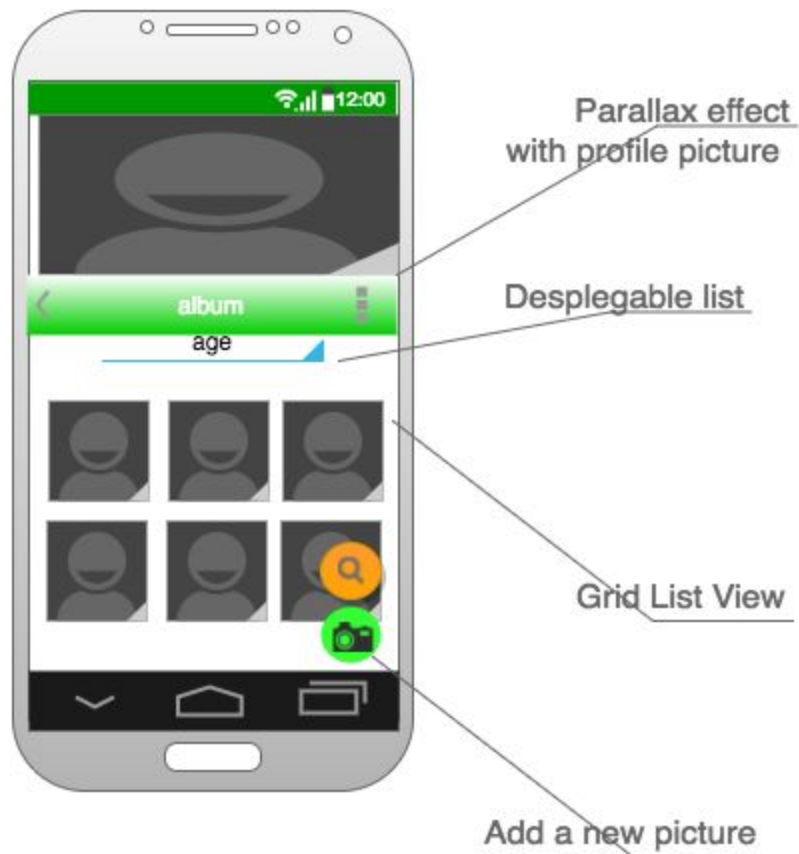
This view will present a form to create a new pet profile. On click on the picture thumbnail will lead to a dialog to open camera app or gallery app of the device. On “add” button press will lead to a ok/cancel dialog to confirm that the user want to save the profile.

Profile view



This view will show the pet's profile and the action button to see pictures album, clinical history and vaccination plan.

Album view



This view will show a gridview with all the pictures of the pet and includes a filter that classify pictures by age. Also includes an floating action button that will take the user to device camera app and adds the picture to the grid view with the adequate age.

Clinical History view



This view will show a listview with clinical events logs sorted by date. As album appBar will include parallax effect and the “+” floating action button will leads to a form that will add a log to the list.

Add Clinical History view

The image shows a mobile application interface for adding clinical history. The screen displays a form with the following fields:

- Attendance date
- Veterinarian hospital name
- Veterinarian pronostic
- Treatment

At the bottom of the form is a green button labeled "Add". To the right of the "Attendance date" field, a calendar pop-up is visible, showing the month of October 2014. The date 24 is highlighted in red, indicating it is the selected date.

October 2014						
Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

This view will show a form to add a clinical history log to the clinical history. Once the user enter the data he should press "add" button and this will show an ok/cancel dialog before he can store the new entrance.

Vaccination Plan view



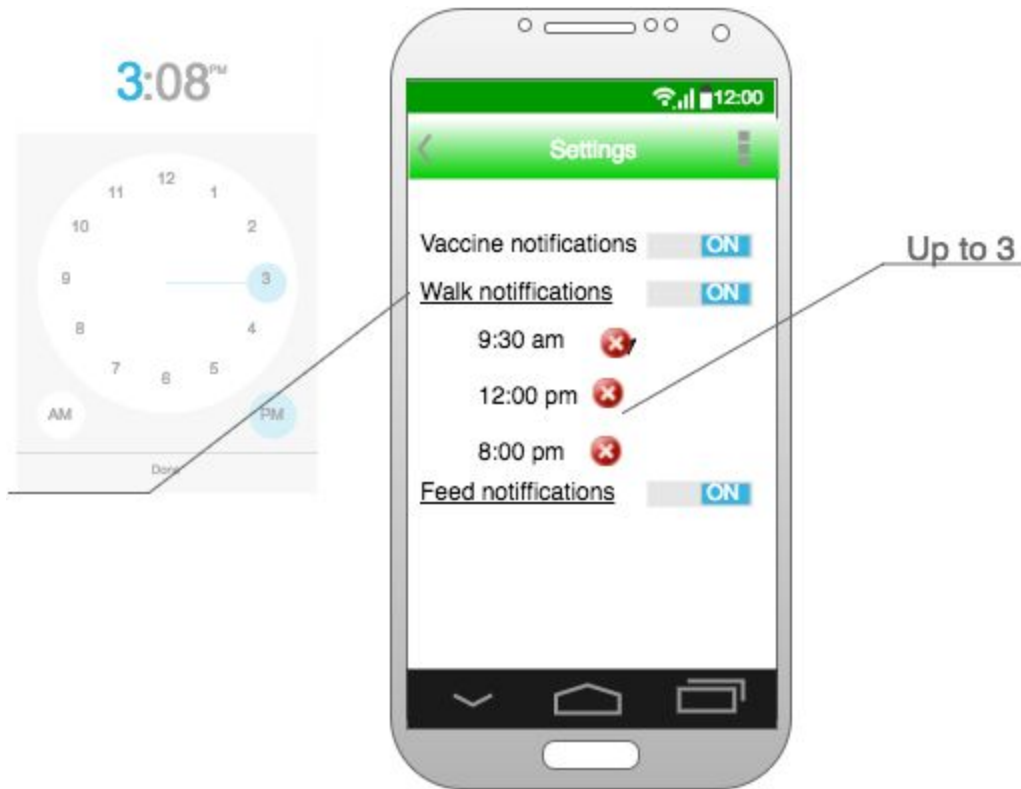
This view will show a listview with vaccination plan sorted by date and showing in the top the no status indicator. As album appBar will include parallax effect and the “+” floating action button will leads to a form that will add a vaccination to the list. Once the vaccine is done the user should touch the status indication and this will change from red “X” to green check as the image above. By default the status indicator will be a red “X”.

Add Vaccine view



This view will show a form to add a vaccine to the vaccination plan. Once the user enter the vaccine name and select in the date picker a vaccination date he should press “add” button and this will show an ok/cancel dialog before he can store the new entrance to vaccination plan.

Settings view



The settings view allows to disable notifications and add time for timely notifications feed and walk .

Widget view



The widget will show the most recent article in “Pets” section of the New York times.

Key Considerations

How will your app handle data persistence?

The data will be stored in a relational database and will be implemented with its own content provider.

Describe any corner cases in the UX.

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?

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

- **Glide** will be include to bind the ImageView with the images restored from database. This way i have not to implement cache management or error callbacks.
- **Volley** Will be used to call API so i have not to implement call queues or exceptions handling.
- **Gson** Will map API elements to the model so i have not to do so.

Describe how you will implement Google Play Services.

- Localization API will help to determine the user's position
- Map API will help to show nearest interest places to pets owners.

Next Steps: Required Tasks

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

Task 1: Project Setup

- Create Github repository for Capstone named **"Capstone Project"**
- Create a new Android Studio project.
- Add to build.gradle on app module the libraries.
- Add Github repository origin to local git.
- Do the first commit.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Profile summarize view
- Build UI for Add pet Profile view
- Build UI for Profile view
- Build UI for Album view
- Build UI for Clinical History view
- Build UI for Add Clinical History view
- Build UI for Vaccination Plan view
- Build UI for Add Vaccine view
- Build UI for Settings view

Task 3: Implement Each Activity and Fragment

- Build Activity for Profile summarize view
- Build Fragment for Profile summarize view
- Build Fragment for Add pet Profile view
- Build Activity for Profile view
- Build Fragment for Profile view
- Build Fragment for Album view
- Build Fragment for Clinical History view
- Build Fragment for Add Clinical History view
- Build Fragment for Vaccination Plan view
- Build Fragment for Add Vaccine view
- Build Activity for Settings view

- Implement Data Models
- Implement Adapters for listViews an GridView on Album View

Task 4: Persistence layout

- Create relational model for database
- Create Contract
- Create DBHelper
- Create content provider
- Implement CursorLoader
- Add data on activities, adapters and fragments

Task 5: Google Map implementation

- Install Google Play Services SDK
- Create a Google Maps Activity
- Create an xml file
- Obtain API Key
- Implement Model for API response
- Create OnMapReadyCallback
- Obtain device Location
 - Define location listener
 - Get location manager
 - Add permissions to manifest
 - Add permission management on runtime for Marshmallow devices
- Personalization of markets to show only Veterinarian and pets shops

Task 6: Create Widget

- Obtain NYT API key
- Create a model for NYT API response
- Create the Widget layout
- Bind the data from the API with the widget