

Capstone Stage 1 - The Garden

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Screen 3](#)

[Screen 4](#)

[Screen 5](#)

[Screen 6](#)

[Screen 7](#)

[Screen 8](#)

[Screen 9](#)

[Screen 10](#)

[Screen 11](#)

[Screen 12](#)

[Screen 13](#)

[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.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

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

[Task 3: Create Data Layer 2 \(REST API, Cache\)](#)

[Task 4: Create Domain Layer](#)

[Task 5: Implement presenter to interact with the domain layer](#)

[Task 6: Implement UI for Some Activity and Fragment](#)

[Task 7: Connect views with business logic through the presenter](#)

[Task 8: Implement UI for Some Activity and Fragment](#)

[Task 9: Connect views with business logic](#)

[Task 10: Implement notifications](#)

[Task 11: Implement Google Play Services](#)

[Task 12: Create transitions between different views](#)

Capstone Stage 1 - The Garden

The Garden

Description

Problem:

Some people like to grow their own plants and would like to have more information about the different parameters (temperature, humidity, doses and frequency of nutrients, etc) which would allow them to improve the growing process. These parameters could be informed through sensors or be entered by the user.

Proposed Solution:

Design an app that allows the user to load data about the process of growing plants and to help him make decisions afterwards relying on it. The app will consume a REST API, which will provide the endpoints in order to persist, modify or delete the data. The API will also expose some business logic on top of the CRUD operations, like how many nutrients were used during one month or one growing period. At the end of one growing process, for example; the user will be able to know how much irrigation was needed during this time and compare it with other processes to improve performance. The collected data will be shown through charts or graphics to give a better lecture.

The app could be resumed as a Garden Administrator. Each garden will have its own views about irrigations, doses and charts. The app will support several gardens and each garden will have several plants. Each plant's data will be presented as a card with its own information. If the user wants more detail about the plant, like looking at some pictures of the process of the plant, he will be able to click over it and will see the rest of the info.

To sum up, the user will use this app to avoid making certain mistakes, either by using historical data or responding to certain alerts such as when a plant needs to be watered.

GitHub Username: [igiagante](#)

Capstone Stage 1 - The Garden

Intended User

This app is for someone who loves growing plants.

Features

List the main features of your app. For example:

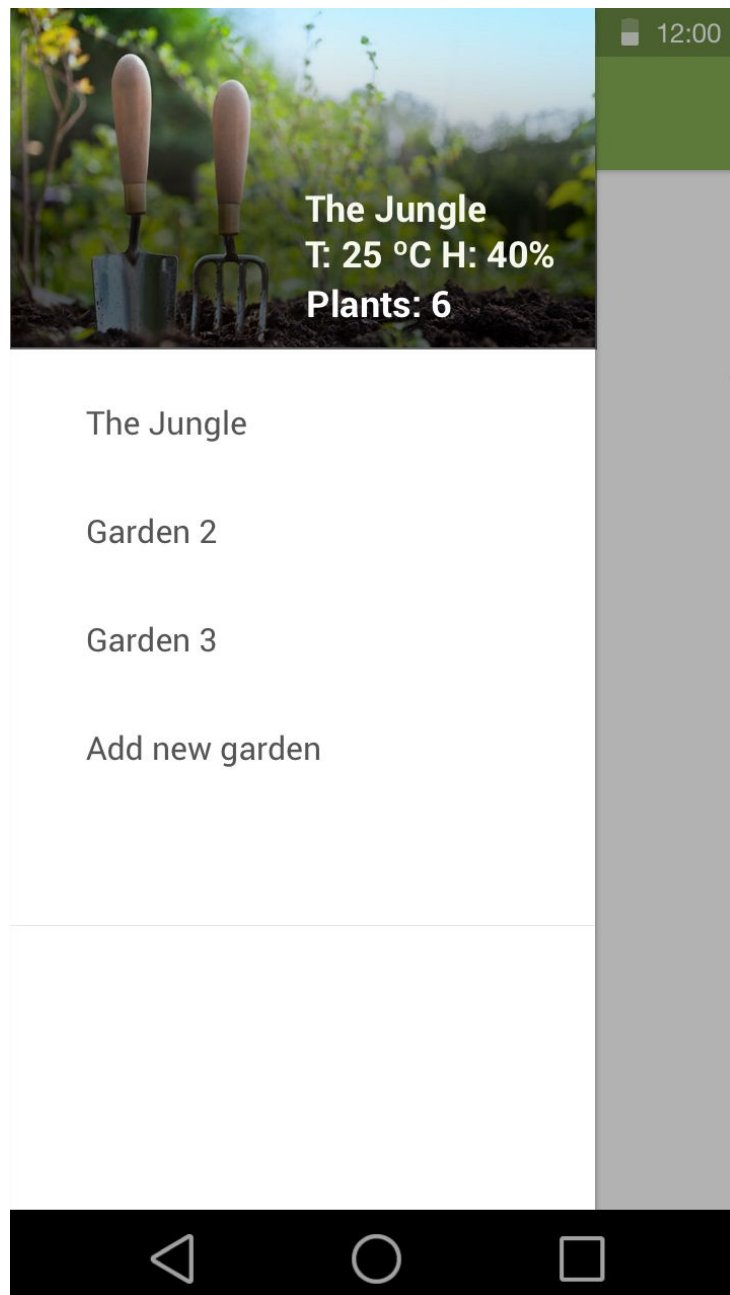
- Saves information
- Takes pictures
- Makes videos
- Loads images
- Uses a gallery to show the pictures or videos
- Shows information using graphic charts
- Sends notifications about different events
- Shares information

Capstone Stage 1 - The Garden

User Interface Mocks

Screen 1

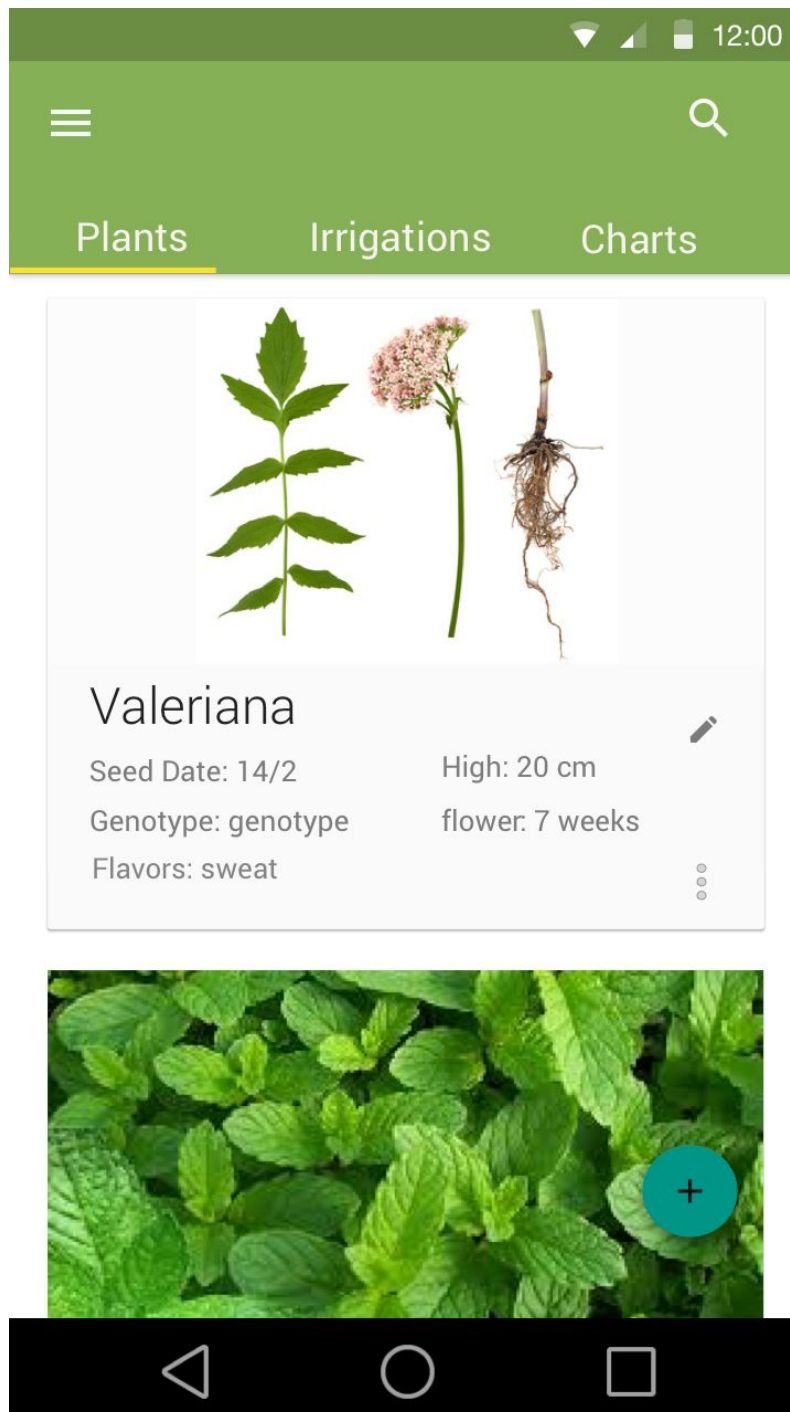
This screen is the main menu of the app. Here, the user can select any of his gardens. The last garden that the user viewed is automatically selected. A panel is displayed on the top showing some information like the name of the garden, how many plants it has, etc.



Capstone Stage 1 - The Garden

Screen 2

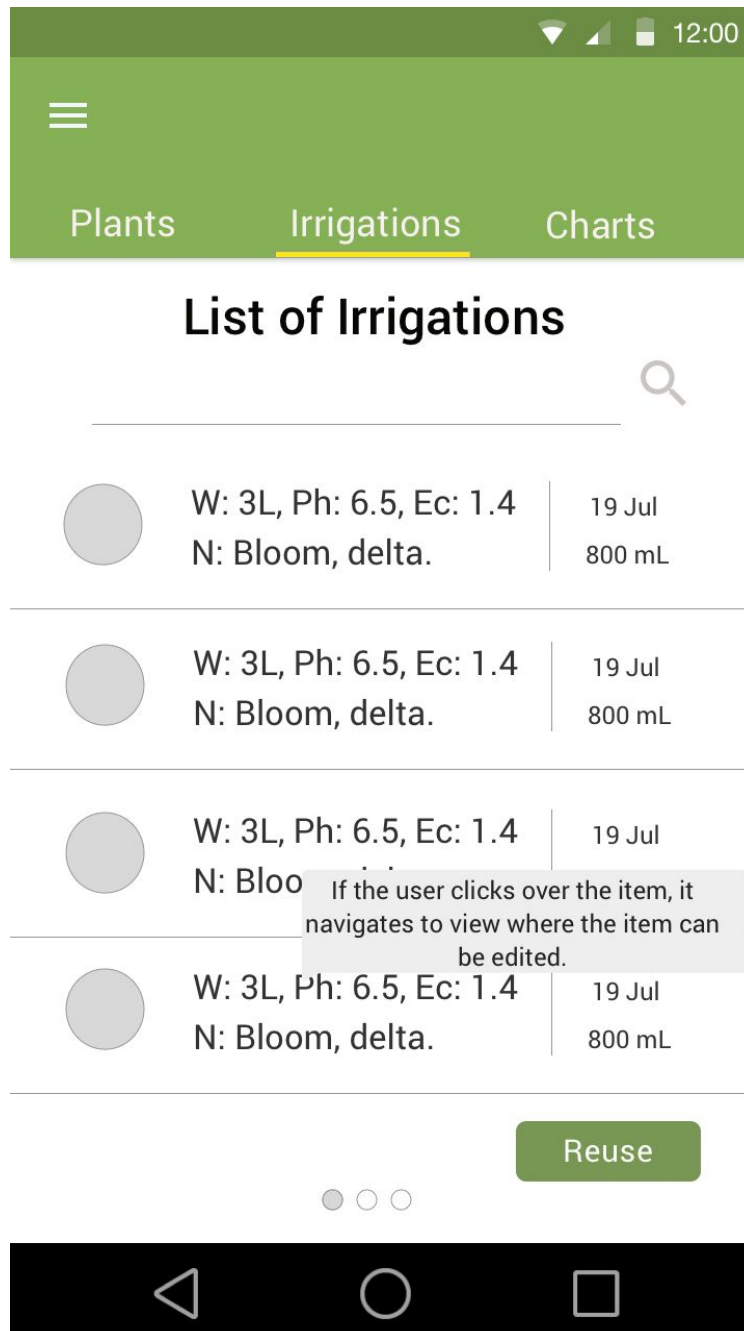
This screen represents the second navigation of the app. After a garden is selected, it's information will be shown in a set of tabs. There are four tabs. The first tab "Plants" shows a list of plants with their respective information. The Floating Action Button allows the user to create and include a new plant to the garden.



Capstone Stage 1 - The Garden

Screen 3

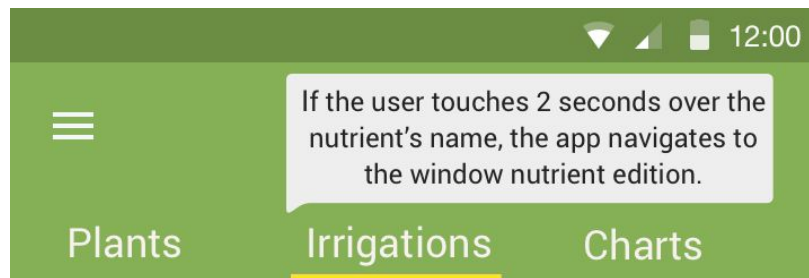
At the Irrigations tab, which is a ViewPager, the user can select an irrigation and reuse it. A dosis is formed by different nutrients, level of water, ph and ec. If the user decides to reuse an irrigation, the app navigates to the last page of the ViewPager where it is possible to edit irrigations' values. The user will be able to edit the dose only if it hasn't been used by more than one irrigation. In case that none of the existing doses satisfies the requirement of the user, a new dose can be created by moving to the next page of the ViewPager.



Capstone Stage 1 - The Garden


Screen 4

At the second page of the ViewPager, the user can choose one or more nutrients to create a new dose. If the nutrient doesn't exist, it can be created by clicking the "add nutrient" button and navigate to that view in order to create the nutrient.



Choose nutrients

If the user touches over the nutrient's name, a dialog is shown with its information.



Nutrients

☐

Delta10

☒

Bloom9

☐

Delta0

+

▼

Water

3 L

Ph Dosis

8 mL

Ph

6.5

Ec

1.4

Capstone Stage 1 - The Garden

Screen 5

At this view, the user will be able to create a new nutrient. Once it's been created, the navigation will come back to the last view and the recently created nutrient will be shown in the list.


12:00


Nutrients


Name Flora Ph 6.5 N P K


How to use

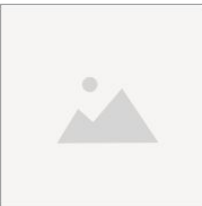
Add image











Save Cancel

Capstone Stage 1 - The Garden

Screen 6

This is the last view of the ViewPager. It shows the dose summary and the user should confirm that all the data is correct.

12:00

Plants Irrigations Charts

Create a new Irrigation

Date Water per plant

Fri, Feb 10 2014 800mL

Dose Resumen

Water: 3L	Ph Dosis: 8mL	Ph: 6.5	Ec: 1.4
-----------	---------------	---------	---------

Nutrient One
Quantity: 3 mL

Nutrient Two
Quantity: 4 mL

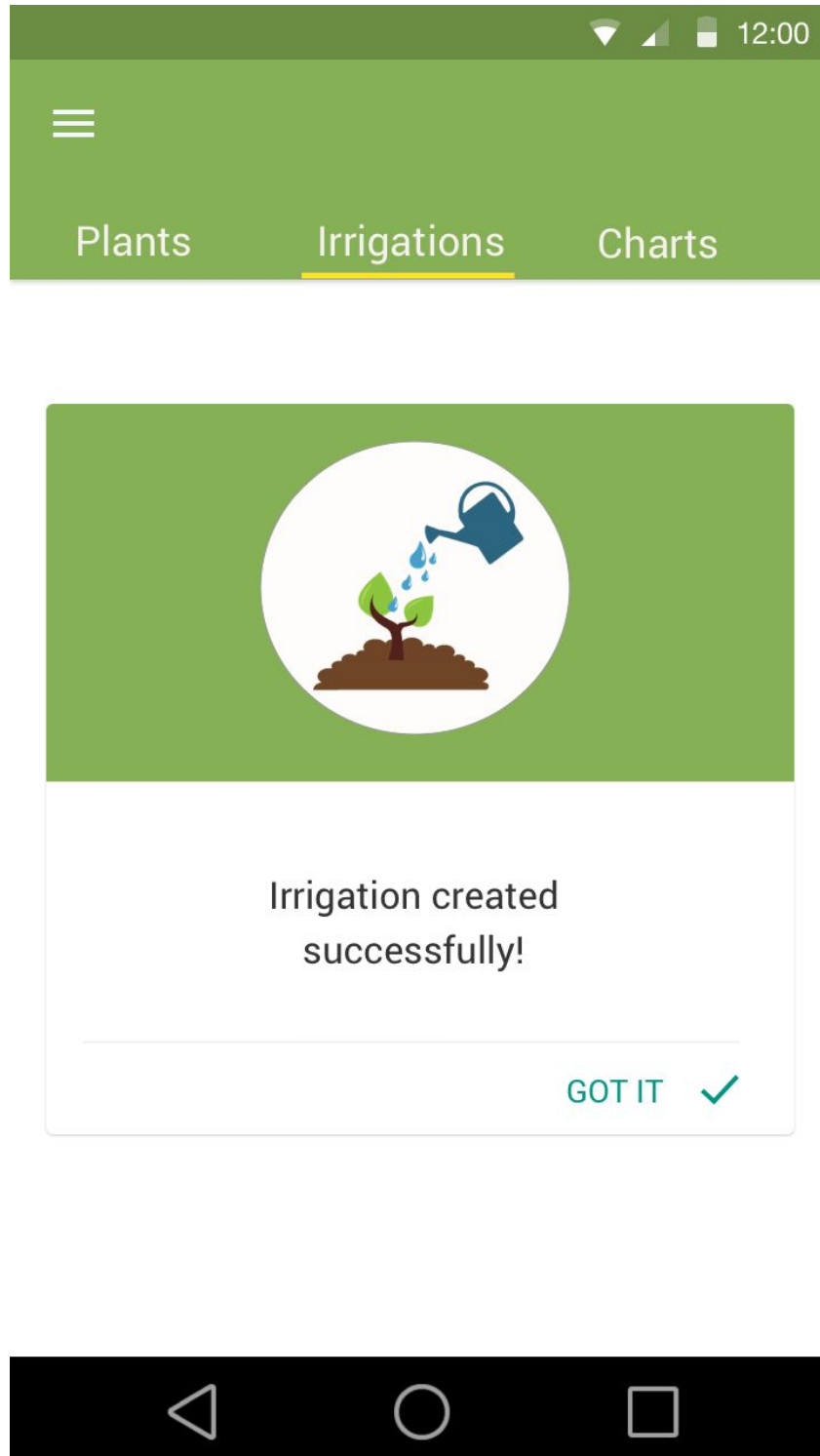
Nutrient Three
Quantity: 3 mL

Save Cancel

Capstone Stage 1 - The Garden

Screen 7

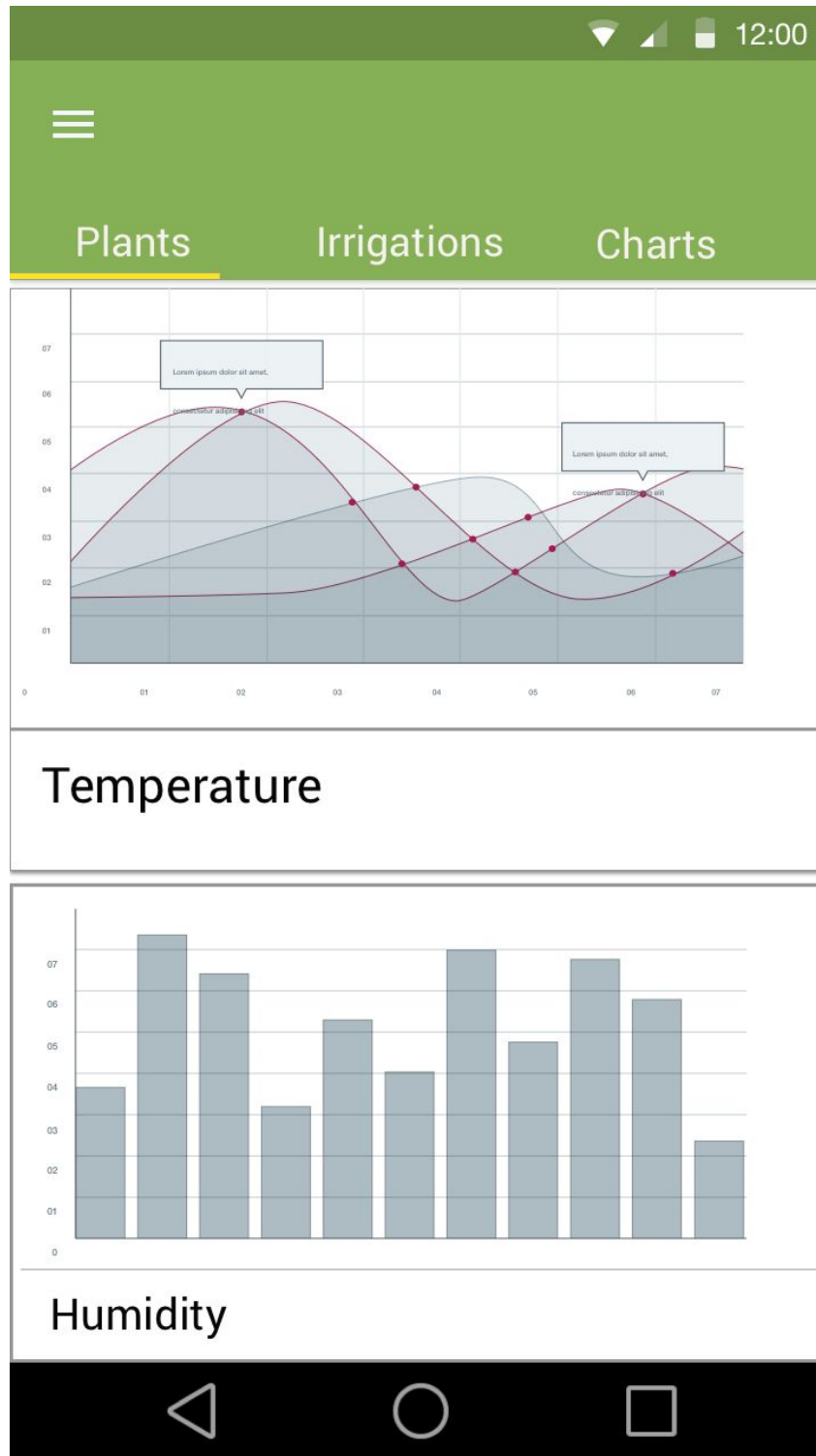
This view appears after the user clicked the “save” button at the view “Create New Irrigation”.



Capstone Stage 1 - The Garden

Screen 8

At the third tab, all the data is shown to the user in the form of graphs.



Capstone Stage 1 - The Garden

Screen 9

This screen allows the user to create a new plant inside of a garden.

12:00

☰

Plant's Name

Name
Valeriana


Ph Soil
6.5


Ec Soil
1.4

Flowering Time
13 weeks

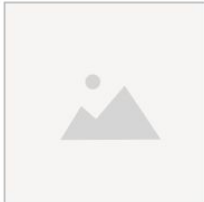
Genotype
genotype

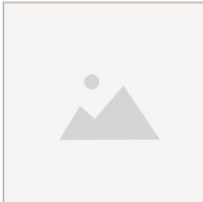
High
30







Add
image










Flavors









Capstone Stage 1 - The Garden

Screen 10

This screen continues with the last view. The user can rate in percentages any properties of the plant that he would like to stand out.

The screenshot shows a mobile application interface for 'Screen 10'. At the top, there is a green header bar with a white hamburger menu icon on the left and the text 'Plant's Name' in the center. Above the header, a status bar shows a Wi-Fi icon, a signal strength icon, a battery icon, and the time '12:00'. Below the header, the 'Effects' section contains three buttons: 'Euphoric 50%' with a plus-minus icon, 'Relaxed 10%' with a plus-minus icon, and 'Energetic 10%' with a plus-minus icon. The 'Medicinal' section contains six buttons arranged in two rows: 'Stress 50%', 'Pain 10%', 'Insomnia 10%' in the first row, and 'Heart 50%', 'Liver 10%', 'Kidney 10%' in the second row. Each button has a plus-minus icon. The 'Syntoms' section contains three buttons: 'Headache 50%', 'Sleppy 10%', and 'Insomnia 10%', each with a plus-minus icon. At the bottom of the screen is a black navigation bar with three white icons: a triangle, a circle, and a square.

Plant's Name

Effects

Euphoric 50% \pm Relaxed 10% \pm Energetic 10% \pm

Medicinal

Stress 50% \pm Pain 10% \pm Insomnia 10% \pm

Heart 50% \pm Liver 10% \pm Kidney 10% \pm

Syntoms

Headache 50% \pm Sleppy 10% \pm Insomnia 10% \pm

Capstone Stage 1 - The Garden

Screen 11

This screen continues with the last view. After saving the plants data, the navigation comes back to the list of plants.

12:00

Plant

The shadow indicates that the plague wasn't choose.

Plagues

Fly White Cochineal Spider

Description

How many do you want to create?

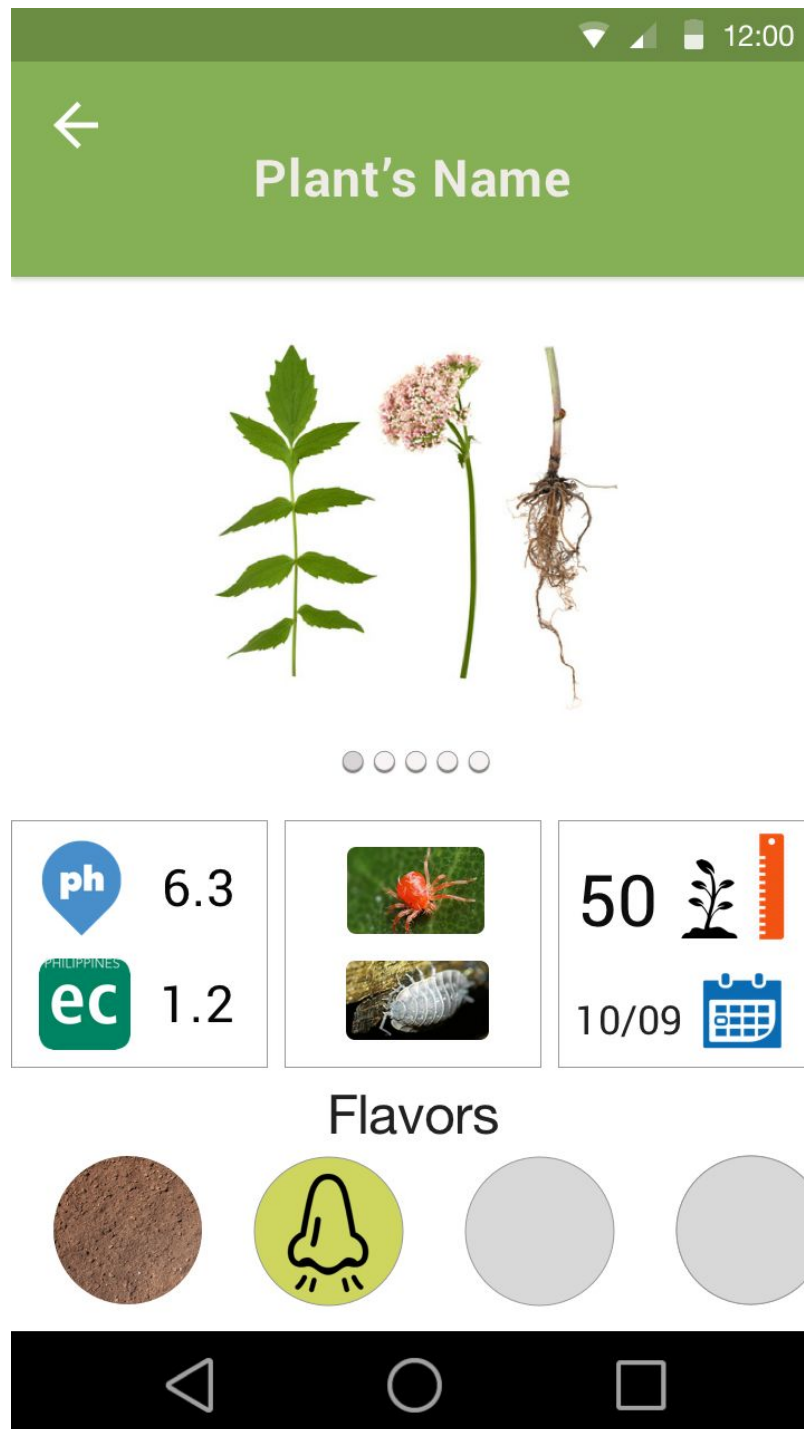
plants: 6

Save Cancel

Capstone Stage 1 - The Garden

Screen 12

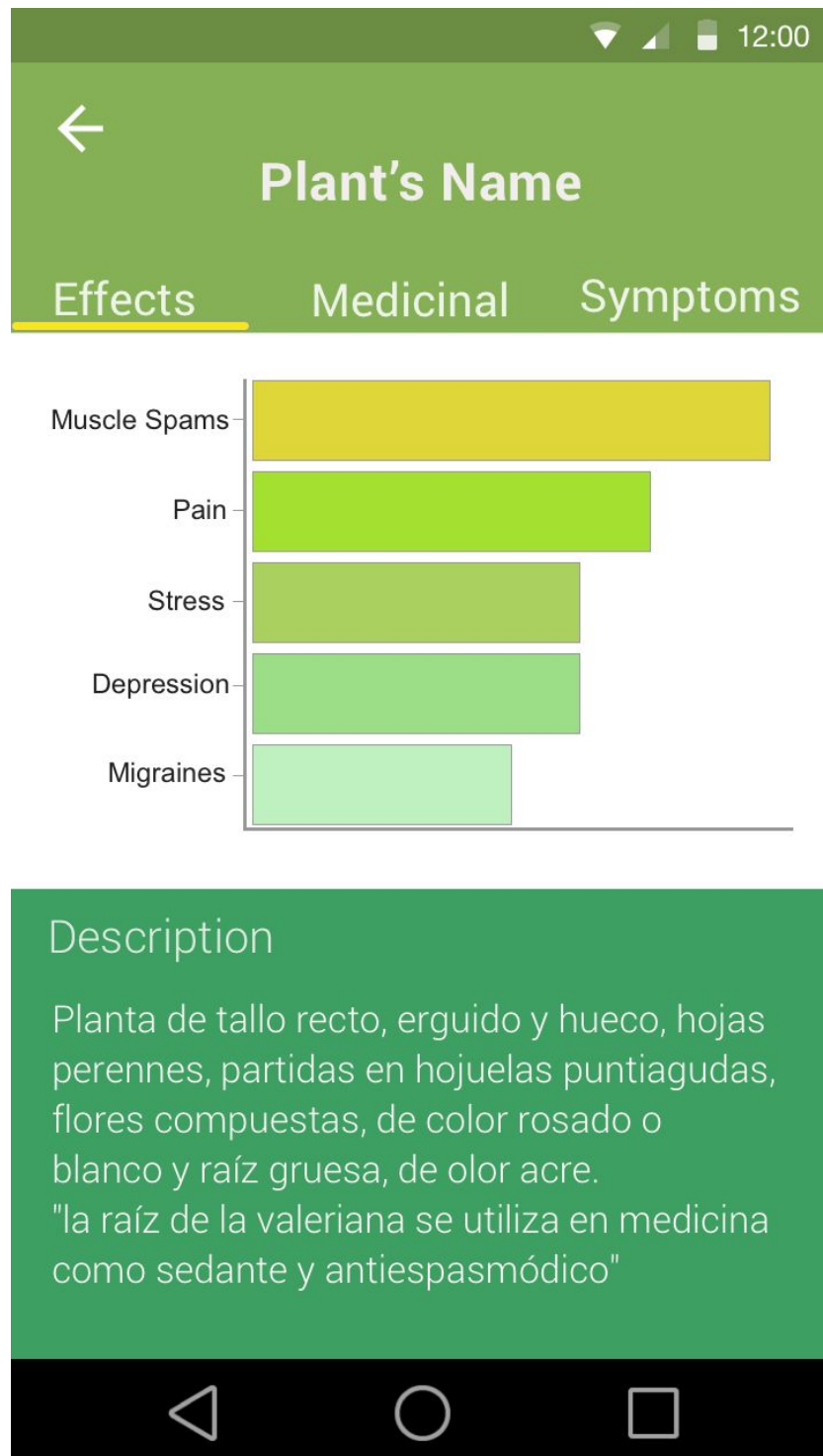
This screen will show all the information about a plant.



Capstone Stage 1 - The Garden

Screen 13

This screen is part of the last view.



Capstone Stage 1 - The Garden

Key Considerations

This app will be implemented using the MVP pattern in order to avoid having business logic in the activities. In order to restrict blocking the UI Main Thread, the app will use Reactive (RxJava) communication to inform some results, which will be running in background mode.

How will your app handle data persistence?

The app will use a content provider in order to persist the plants.

Describe any corner cases in the UX.

If the user wants to edit an irrigation, it should click over one list's item (irrigation). Then, the user navigates to the view "Edit Irrigation".

After one Irrigation is created, the navigation comes back to the first page of the ViewPager at the irrigations tab.

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

Picasso to handle the loading and caching of images.

RxJava & RxAndroid to manage some operations, which should run in a new Thread in order to avoid blocking the Main UI Thread.

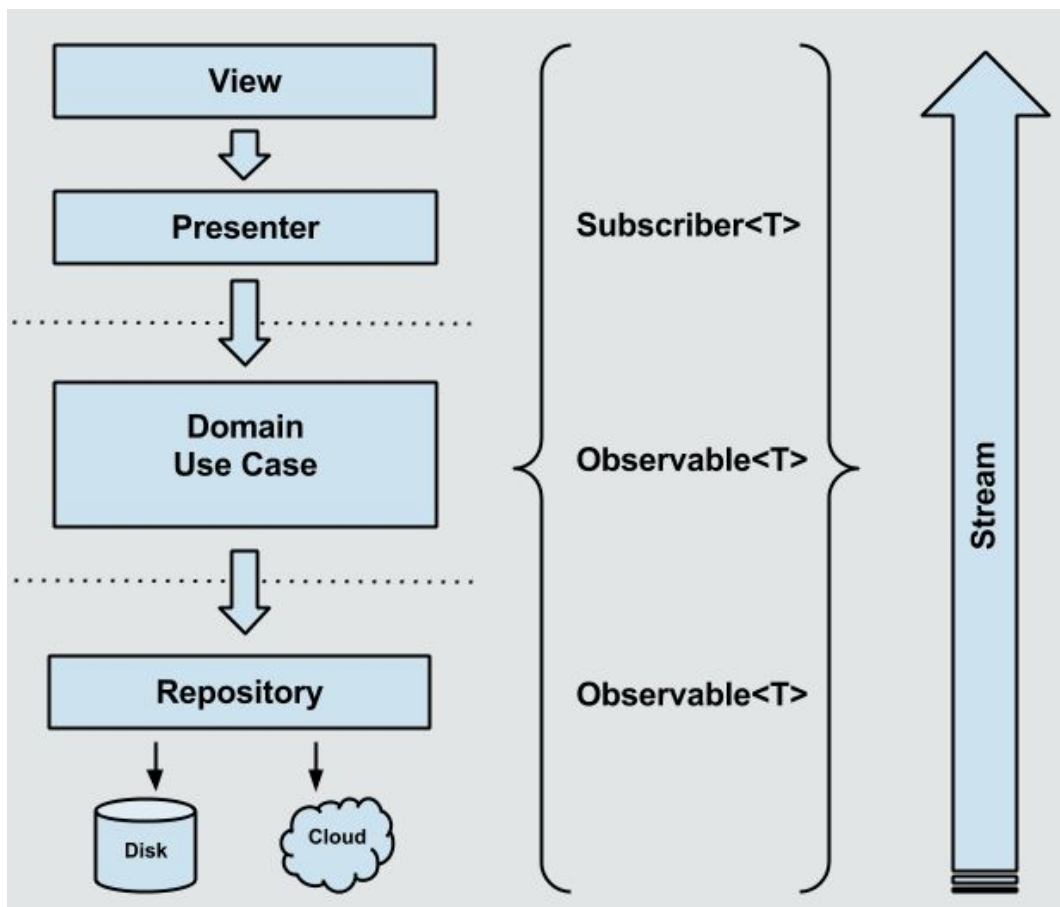
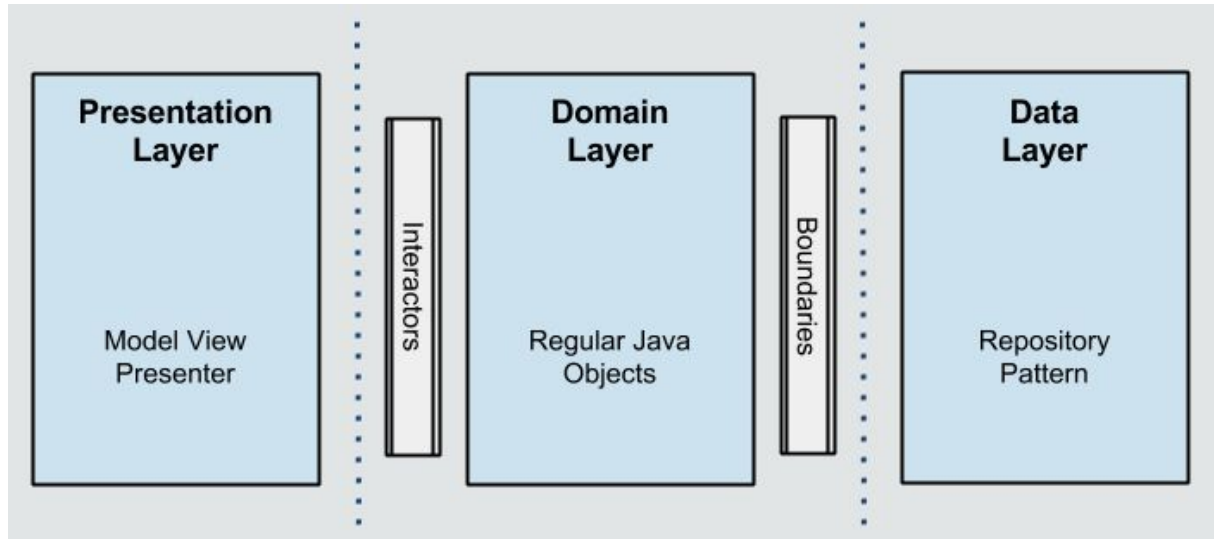
Square's Retrofit project to handle HTTP requests and responses.

Dagger for DI (Dependency Injection).

Butterknife to find and cast automatically fields and view IDs at the corresponding view layout.

Capstone Stage 1 - The Garden

Note: This project's architecture is based in [fernando cejas - clean architecture](#). The tasks described in the next session may be executed more than once, since the software development will be based on an incremental process.



Capstone Stage 1 - The Garden

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 project
- Create repository in Github
- Setup libraries
- Create the core app using the MVP pattern
- Apply design support library
- Apply Coordinator Layout for the main activity and consistent color theme.

Task 2: Create Data Layer (Content provider, Contract, DBHelper)

- Analyse which data should be persisted
- Create Contract class
- Create ContentProvider class
- Create tests to check that the data layer is working correctly

Task 3: Create Data Layer 2 (REST API, Cache)

- Analyse which data should be requested in order to create the REST API for consumers' classes
- Design and Implement REST Client API

Task 4: Create Domain Layer

- Implement interactors (Use cases). All the business logic will be presented at this layer.

Task 5: Implement presenter to interact with the domain layer

- Implement presenter for the interaction between the view (Activity or Fragment) and model (Domain Layer). Each presenter is composed with an interactor, which performs the job in a separate Thread.

Capstone Stage 1 - The Garden

Task 6: Implement UI for Some Activity and Fragment

- Build Navigation Drawer for main navigation
- Build Tabs for the second navigation
- Build UI for the first tab (plants)
 - It will include a RecyclerView with cards, each one represents a plant
 - Each card has a button to edit the plants' information and another one for share it
 - This UI will have a Floating Action Button in order to create a new plant for a garden
- Build UI to create a new plant
- Build UI to see the plants' information

Task 7: Connect views with business logic through the presenter

- Connect the first tab with it's Business Logic (plants)
- Connect the view "create a new plant" with it's Business Logic
- Connect the view "plant's information" with it's Business Logic

Task 8: Implement UI for Some Activity and Fragment

- Build UI for the second tab (Irrigations).
 - This tab should have a ViewPager, which will be representing the steps to create a new Irrigation
 - The first page of the ViewPager will show a list of doses in case the user wants to reuse one of them.
 - At the second page of the ViewPager, it will have a RecyclerView (horizontal) with cards (nutrients). It is possible to create a new nutrient in case none of them satisfy the user's requirements.
 - The last page of the ViewPager will have two input data and a list, which summarizes what is the dose chosen or created by the user.
 - Once the user has created the irrigation, a congrats card will be shown.
- Build UI for the third tab (Charts)
 - Implement a RecyclerView with some cards in order to show some graphics data.

Capstone Stage 1 - The Garden

Task 9: Connect views with business logic

- Connect the second tab with its Business Logic (Irrigations)
 - Connect each view from the ViewPager with its Business Logic
- Connect the third tab with its Business Logic (Charts)

Task 10: Implement notifications

- Send notification to user
 - Each notification should have an icon, main text and description

Task 11: Implement Google Play Services

- Implement Google Adds on.

Task 12: Create transitions between different views

- Create transition when the user navigates from one view to another
 - Add some animations during the transition