

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[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: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

**GitHub Username:** [juancoob](#)

## Vegginer

### Description

Nowadays, we have already learnt nobody deserves to be discriminated against people by the color of their skin, sex, religion, ideology or sexual preference, but animals, on the mere fact that are not humans and be in our food chain, are not usually treated as beings with full rights.

If you have already considered veganism to change your consumption habits, fight against climate change and take part of relieving the world in hunger in your everyday life: Vegginer is here to guide you

## Intended User

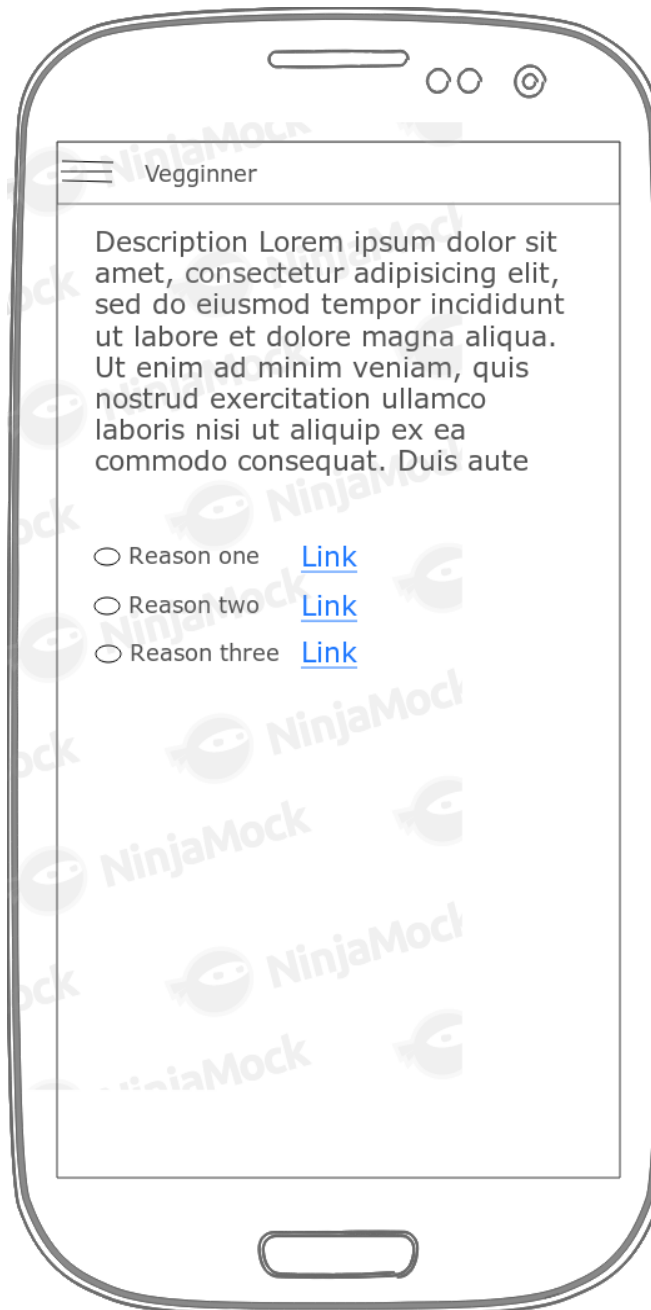
This app is intended for people who are curious and/or willing to be vegan

## Features

- Shows information about how we could change our habits
- Displays quick advices about how to tackle some common situations and questions
- Offers how to replace some animal products with vegan ones
- Proposes some recipes to make
- Marks in google maps some places to eat
- Searches vegan events near the user
- Recommends movies and documentaries to raise the awareness about veganism

## User Interface Mocks

## Screen 1

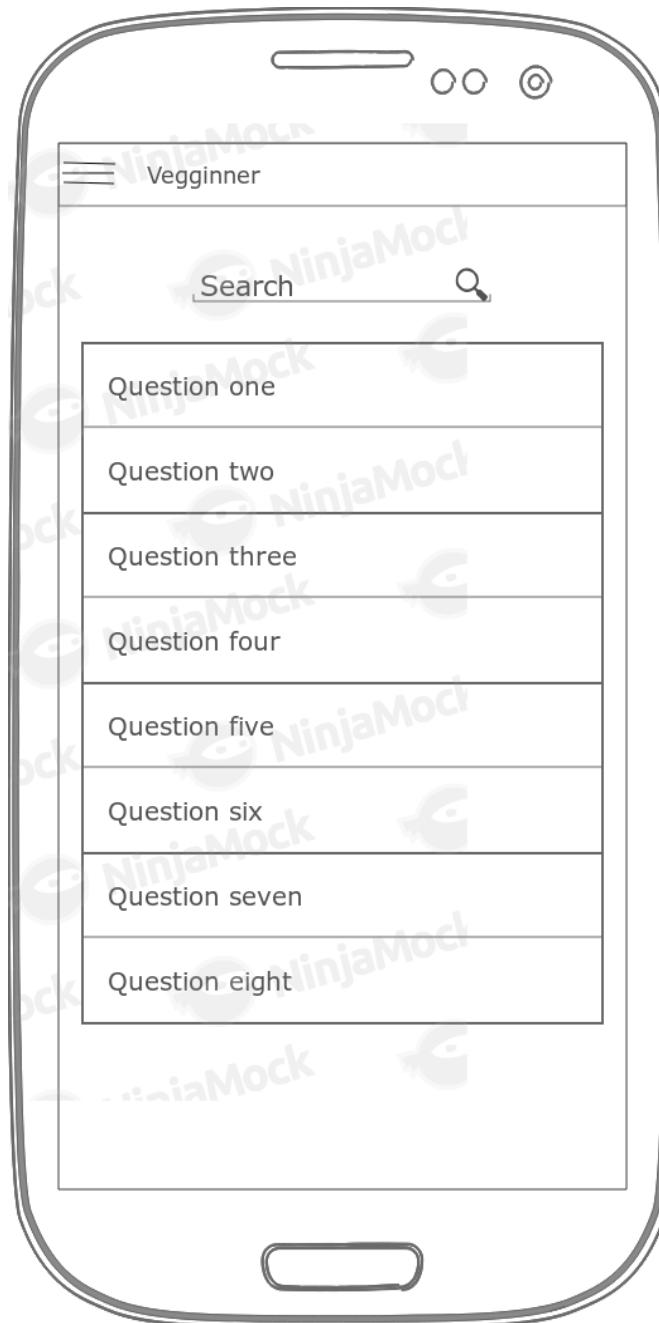


This is the main activity where the app shows a summary explaining the vegan lifestyle advantages. It has some links which redirect to other web pages explaining some facts more detailed.

**Screen 1B**

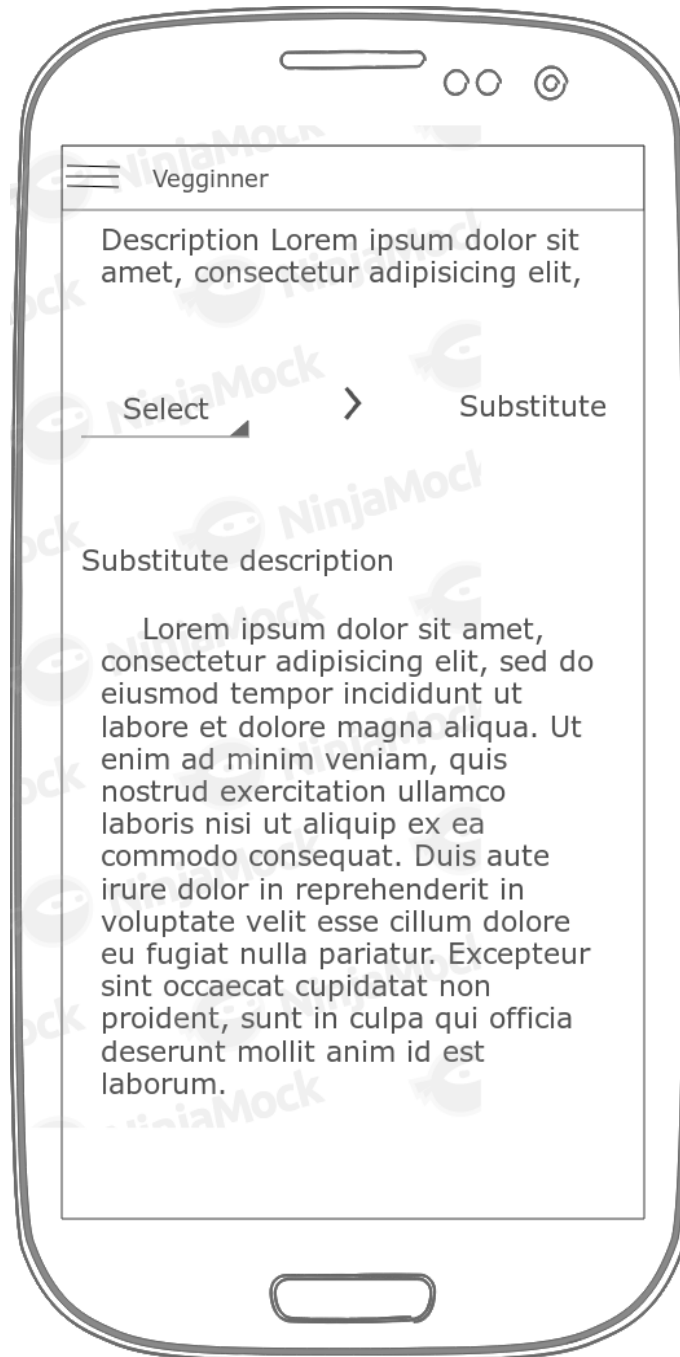
This is the main menu where the user can navigate in the app over sections.

## Screen 2



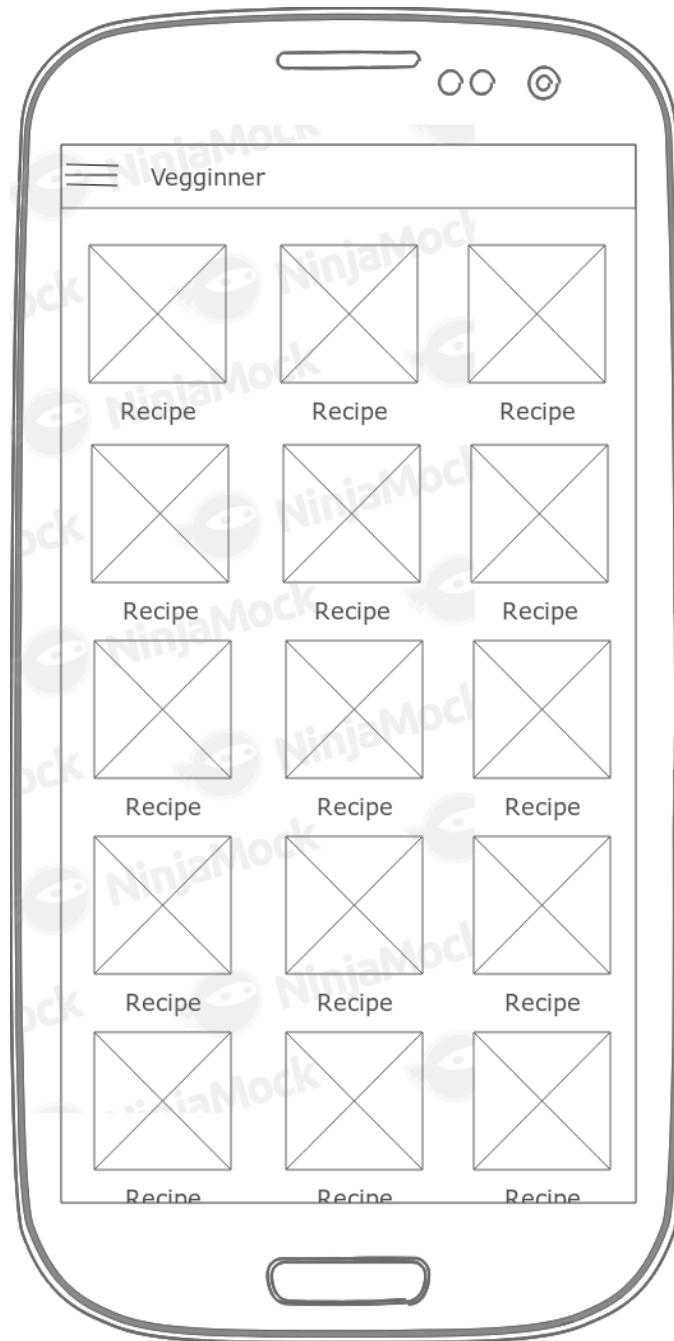
This is the advices screen which shows a searchable list of expandable advices to help the user with common questions or situations.

### Screen 3



This is the equivalencies screen where the user could discover equivalencies using the spinner to change animal products with vegetal ones on any recipe. It explains the advantages of the substitute.

## Screen 4



This is the recipe screen which shows vegan recipes to do thanks to the Edamam API.

## Screen 5



This is the recipe details screen where the user could see the selected recipe image, maximum servings number and add ingredients to the shopping list widget.

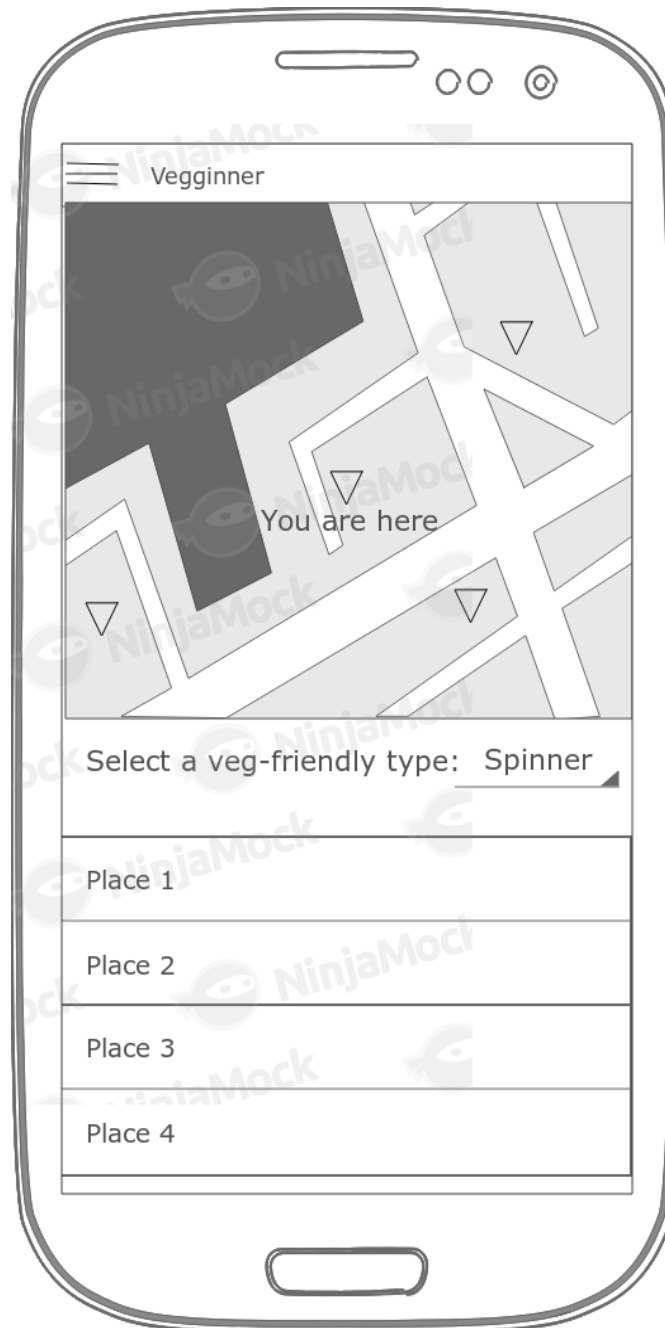


## Screen 6



This is the recipe steps screen where the user could see the original web page to get the steps and make the recipe.

## Screen 7



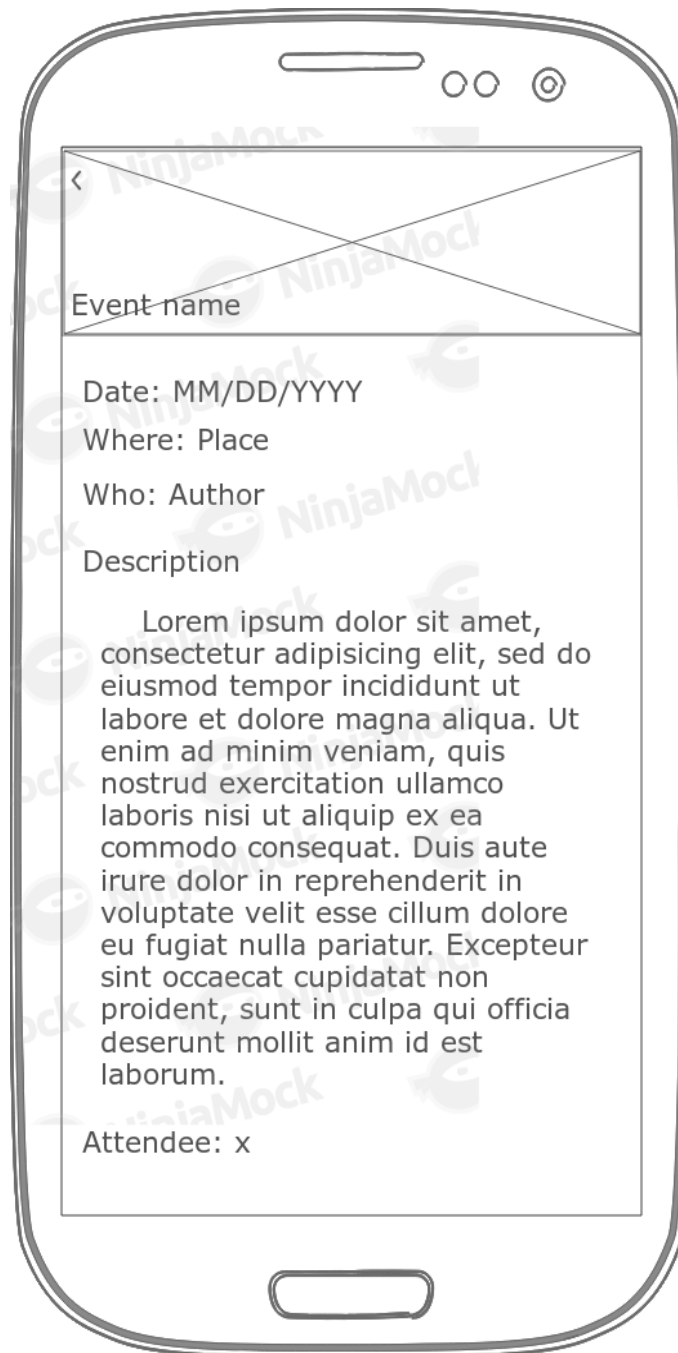
This is the places screen which shows where the user is and offers veg-friendly options to list them thanks to google maps and the Location service.

## Screen 8



This is the events screen which list vegan-friendly events by date near the user thanks to the meetup API.

## Screen 9



This is the event description screen to show the selected event details as date, place, author, description and number of attendees.

## Widget



This is the widget on an empty desktop screen and shows an interactive ingredient list which elements selected previously to buy. The user could click on an ingredient to mark it as bought and click it again to undo changes.

## Key Considerations

### How will your app handle data persistence?

The app is going to handle data using Room, LiveData and ViewModel

### Describe any edge or corner cases in the UX.

- If the user have no internet, a message is displayed.
- If there is no match on any search box, the app shows an empty list.
- If some recipe have no steps, the user will be notified.
- If the user select a place type and get no results, an empty list is displayed.
- If there are no events near the user, an empty list is displayed.
- If there are no ingredients added to the list, it will show an empty list.

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

- Picasso v2.71828 to handle images
- Butterknife v8.8.1 to bind views
- Retrofit 2 v2.4.0 to manage API requests
- Maps v15.0.1 to show the map
- Place v15.0.1 to show places near the user
- RecyclerView v27.1.1 to list elements from lists
- Constraint-layout v1.1.2 to construct robust layouts
- Design v27.1.1 to make the app visually attractive

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

- Google Maps to integrate maps on the app and shows places near the user as restaurants and bars.
- Location will be implemented to locate the user in the current position.

## Next Steps: Required Tasks

### Task 1: Project Setup

- Create the project with an empty activity
- Update gradle and use the recent and stable version

- Implement project libraries

## **Task 2: Implement UI for Each Activity and Fragment**

- Build UI for the main screen
- Build UI for the main menu
- Build UI for the advice screen
- Build UI for the equivalencies screen
- Build UI for the recipes screen
- Build UI for the recipe details screen
- Build UI for the places screen
- Build UI for the events screen
- Build UI for the event details screen

## **Task 3: Create a recyclerview to show advices**

- Create the item layout
- Create the RecyclerView class
- Implement the filtering

## **Task 4: Create a search process with Room in the equivalencies screen**

- Create the model, DAO, database and local repository
- Setup click listeners to show data

## **Task 5: Implement retrofit 2 to get recipes**

Create the recipe list screen

- Setup retrofit 2
- Create endpoints to get recipes thanks to Edamam API.
- Create a recyclerview to show recipes
- Design an item for the recyclerview
- Setup click listeners

Create the recipe detail screen

- Implement the ViewPager
- Configure the toolbar to show an image and the title
- Show servings
- Show ingredients
- Implement the click event to add any ingredient to the widget
- Setup the webView to show the author web page and recipe steps if they are available

## Task 6: Implement maps and location services

- Setup google maps to show the map
- Implement the location to pinpoint the user
- Create the spinner logic to select the place type
- Create the recyclerView to show the places retrieved
- Design a recyclerView item
- Implement retrofit 2 to use the location API to get near places from the user as the selected for the spinner

## Task 7: Create the events screen

Create the event list screen

- Create the recyclerView to show the events retrieved
- Design a recyclerView item
- Implement retrofit 2 to use the meetup API to get near events from the user

Create the event detail screen

- Configure the toolbar to show an image and the title
- Show the event details as date, place, author, description and attendees

## Task 8: Implement the widget

Create the interactive shopping list to add it on the screen

## Task 9: Create tests

Create espresso tests for UI testing

## Common project requirements

- App is written solely in the Java Programming Language
- App utilizes stable release versions of all libraries, Gradle (v3.1.3), and Android Studio (v3.1.3).
- App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.
- App validates all input from servers and users. If data does not exist or is in the wrong



- format, the app logs this fact and does not crash.
- App includes support for accessibility including content descriptions.
- It use an AsyncTask when the user makes a search on the equivalencies screen