

## How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [ Select All → Copy → Paste into new document ]
2. Name your document file: “**Capstone\_Stage1**”
3. Replace the text **in green**

---

[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: **SnoutUp**

# Grocery Review App

## Description

With an ever growing supply of various food items available in stores, it's hard to remember products you liked or hated. Grocery Review application aims to solve this by providing an easy way to track things you eat, rate them and recommend the tastiest treats to others.

## Intended User

Grocery Review application can be used by anyone, who already started shopping for food. It could be kids spending their pocket money on crisps and candy, students trying to discover cheapest, yet still tasty, ramen or hotdogs or any adult who gets overwhelmed by options during grocery shopping.

## Features

- Saves product information including rating, picture and location of purchase
- Allows user to browse and search grocery item ratings of other application users
- Built-in shopping list functionality
- Shopping list widget

## 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](http://www.ninjamock.com), Paper by 53, Photoshop or Balsamiq.

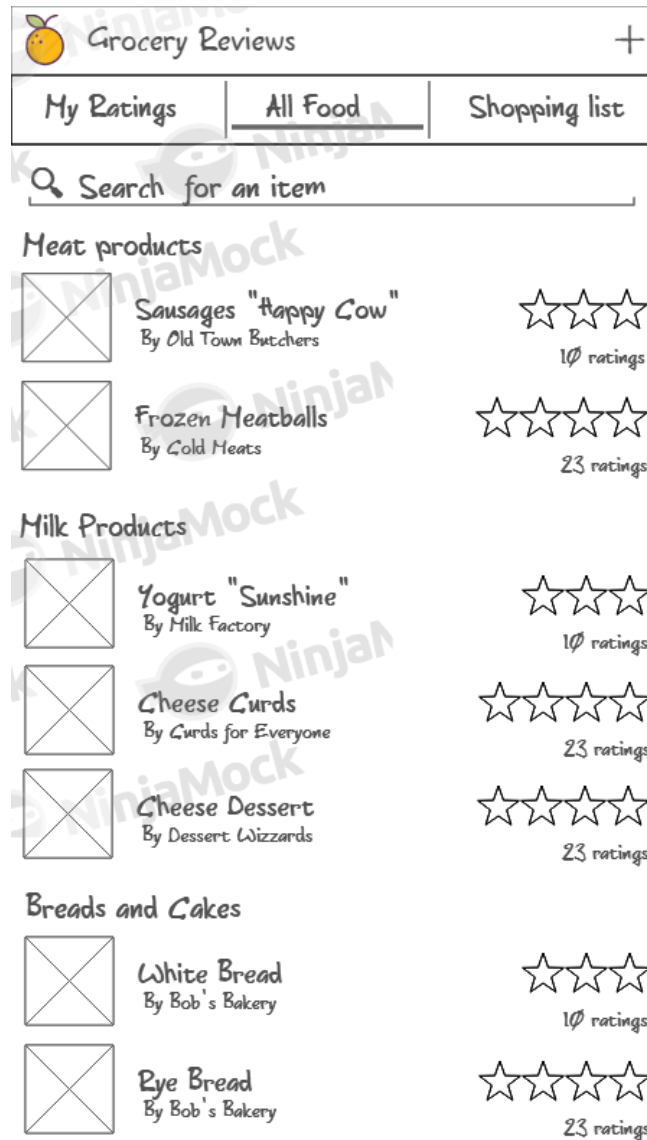
## Screen 1 - My Ratings List

 Grocery Reviews <span style="float: right;">+</span>		
My Ratings	All Food	Shopping list
	Sausages "Happy Cow" <i>Simply delicious</i>	☆☆☆
	Cheese "Milk Miracle" <i>Lacking fLavor, but great for cooking</i>	☆☆
	Sausages "Happy Cow" <i>Simply delicious</i>	☆☆☆
	Cheese "Milk Miracle" <i>Lacking fLavor, but great for cooking</i>	☆☆
	Sausages "Happy Cow" <i>Simply delicious</i>	☆☆☆
	Cheese "Milk Miracle" <i>Lacking fLavor, but great for cooking</i>	☆☆

+ Add Item

My ratings screen displays all the food items current user added themselves with star rating as well as short note describing the taste, value for the price or other properties. This is the screen where FloatingActionButton will be visible for quick access to most common action - adding new item. Tab-based navigation will help to quickly move around different lists described in more detail in other interface mocks.

## Screen 2 - All Food List



"All Food" section of the application contains all the products submitted by other users of this application, together with overall rating and amount of the votes. The list can be easily searched through to quickly find and use a certain item to create a new rating or add to shopping list.

### Screen 3 - Shopping List

 Grocery Reviews

My Ratings

All Food

Shopping list

+

Add an item

☒ Cheese "Milk Miracle"

✓

☐ Frozen Meatballs "Beef Bombs"

✓

☐ Yogurt "Smooth Udder"

^



By Milk Factory



10 ratings

A very tasty and cheap yogurt for whole family.

☒ Cheese "Milk Miracle"

✓

☐ Frozen Meatballs "Beef Bombs"

✓

☒ Cheese "Milk Miracle"

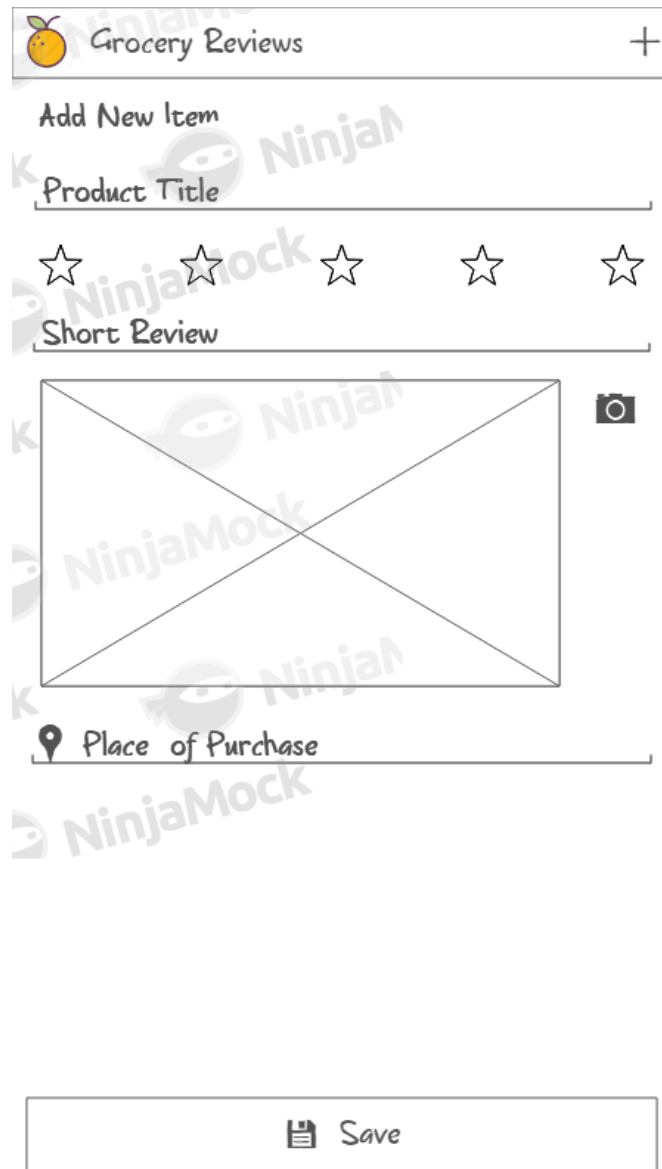
✓

☐ Frozen Meatballs "Beef Bombs"

✓

"Shopping list" displays a certain temporary grocery list which can be created before going to the story. List item expands to show more details.

## Screen 4 - New Item Form



The form is titled "Grocery Reviews" with an orange icon and a plus sign. It includes a section for "Add New Item" with a "Product Title" input field. Below this is a star rating system with five empty stars. A "Short Review" input field follows. There is a large rectangular area for a picture, marked with a large 'X' and a camera icon to its right. Below the picture area is a "Place of Purchase" input field with a location pin icon. At the bottom is a "Save" button with a document icon.

Grocery Reviews +

Add New Item

Product Title

☆ ☆ ☆ ☆ ☆

Short Review

Place of Purchase

Save

New item form used to create a new entry to grocery rating system. Includes title, short review, star rating, picture and an option to add location where item was bought.

## Key Considerations

### How will your app handle data persistence?

All user data will be saved to Firebase Realtime Database.

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

- Application should keep working offline and sync newly added data when connection becomes available
- Shopping list section of the application will contain auto-completing search list to quickly add rated items to the grocery list. Pressing “back” while search is active will hide the result list.
- Keeping proper fragment back stack will allow correctly navigate through the tab history using device’s back button.

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

- Firebase Database library for used for data persistence.
- Firebase Authentication library to ease user creation and authentication
- Picasso to load and process images if needed
- Google Places to search through surrounding stores
- Admob library for ads

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

- Google Places service will be used in “New Item” form to search and assign grocery store to an item
- Admob will be used to monetize the application and show ads in certain sections
- Firebase Database service will be used to store data
- Firebase Crash Reporting to track application stability and crashes

## 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.

## Task 1: Project Setup

- Create new Firebase project
- Configure Firebase Realtime Database
- Setup Gradle dependencies for required libraries

## Task 2: Data Model

Creating data classes to match Firebase Realtime Database structure.

- User
- FoodItem
- FoodCompany
- ItemRating
- ShoppingListItem

## Task 3: Implement UI for Each Activity and Fragment

- MainActivity User Interface
  - CollapsingActionBar layout
  - TabLayout
  - FloatingActionButton to navigate to “New Item” activity
  - “My Ratings” fragment (RecyclerView with custom list items)
  - “All Items” fragment (grouped RecyclerView with custom items and search with autocomplete)
  - “Shopping List” fragment (RecyclerView with checkbox items and expandable details)
- NewItemActivity
  - Text inputs
  - An option to start camera intent or browse for the picture in the gallery
  - Image preview
  - Place search

## Task 4: Connecting data to UI

- Implementing Firebase Realtime Database
- Querying and displaying items for different lists (My Ratings, All Items, search results, shopping list)



## **Task 5: Implementing other services**

- Adding Admob ads to the user interface
- Connecting Google Place to “New Item” grocery store search field
- Implementing Firebase Crash Reporting to track any issues which may arise

## **Task 6: Application Widget**

Creating a widget to show current shopping list on the home screen of the device.

## **Task 7: Polish**

- Smooth transitions between fragments and activities
- Shared element transitions
- Parallax scrolling
- Animations

## **Task 8: Accessibility**

- Providing RTL-layout support
- Providing accessibility support (content descriptions, D-pad navigation)
- Localization support

## **Task 9: Testing**

- Trying out the application on different Android devices and emulator setups
- Trying out all activities and fragments in different orientations
- Testing offline functionality
- Espresso automated UI tests

## **Task 10: Preparing application for release**

- Creating required application icons
- Setting up paid and free flavors
- Cleaning up the code