How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: wwSchrader

Peak Fresh

Description

Peak Fresh provides an easy way to track perishable goods at home. Users will save money by getting alerts when a product is approaching its expiration date to promote the consumption of that product rather than letting it go to waste.

Intended User

The intended user is anyone in a household responsible for purchasing, tracking and/or consuming perishable goods.

Features

- Saves product name and selected expiration date
- Takes pictures of products.
- Provide notification of expiring products.
- Add expiration dates to calendar

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 Photoshop or Balsamiq.

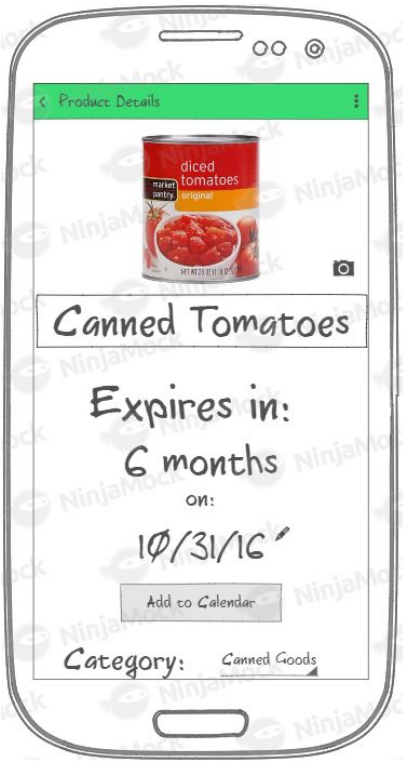
Main Screen - List

A view listing products with their names and time remaining until expiration. Users can select a product to launch the details screen or tap the fab to create a new product. Users can select a category to filter the list.

Main Screen - Icons

Similar to the main screen showing the lists. It displays just the icons and the expiration dates.

Details Screen



A detail view of one of the products. This is the same screen where a user would add info to the product. Users can tap the camera icon to use their camera to take a picture. Users can tap the “Add to Calendar” button to add the expiration date event to their personal calendar. Users can edit the expiration date and select the category the product belongs to.

Date Selector

A date selector window would pop up if the user chooses to edit the date.

Settings

Settings screen where a user can toggle notifications on or off. Users can set the number of days in advance to receive notifications. Can switch between icon and list view.

Key Considerations

How will your app handle data persistence?

A content provider will be built to store the data.

Describe any corner cases in the UX.

Name field will be checked for null input. Image will have standard default image unless one is provided by a camera. Existence of a camera and calendar will be checked before attempting launching of intent.

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

I will be using Glide for loading images and Simonvt/schematic for generating content provider for database.

Describe how you will implement Google Play Services.

I will use Google Analytics to track the number of users, time in app, number of products a user will add, whether they add a picture, how frequently a product expires, if the notification is switched on or off, if a crash has occurred.

I will use AdMob to place ads on each screen using smart banners.

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

- Configure Libraries
- Create Content Provider
- Create a cursor recyclerview adapter

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Main Screen - List & Icons
- Build UI for Detail Screen
- Build UI for Settings

Task 3: Implement using camera for photos

- Build intent to launch phone's camera if available
- Grab photo and store reference in database

Task 4: Handle error cases

- Check for null value in product name.
- Check for existence of camera on phone.
- Create error message if no phone exists.

Task 5: Create Widget

- Create widget showing listview of products organized by expiration date

Task 6: Optimize for tablets

- Add fragments optimized for tablet view

Task 7: Optimize for Material Design

- Add surfaces to main screen
- Add transitions between main screen and detail screen

Task 8: Accessibility

- Add accessibility features such as content descriptions to each view.
- Optimize for TalkBack support

Task 9: Test for bugs

- Test app looking for unexpected results.

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"