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: github.com/ConnorSinnott

SelfImage

Description

Most weight management apps on the marketplace help users target a preferred weight at some point in the future, and then make recommendations for eating plans based on progress toward that goal. However, many users don't choose to alter their weight because of a number on their scale; they chose to make an adjustment because they prefer to look different. SelfImage is an application that revolves around the idea that giving a user visual feedback of their change in image over time is more valuable than a change in numbers.

Intended User

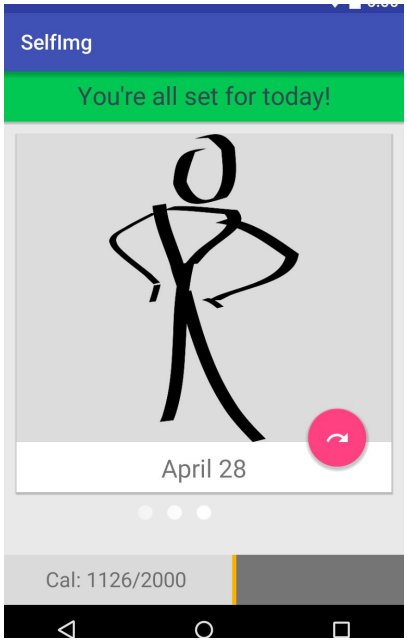
The intended users for SelfImage are individuals who are looking to change their physique. This is a large target market which frequently receives new users. These users should be willing to open this application a minimum of once a day.

Features

- Take pictures
- Generate meal-plans
- Create a timelapse of user progress
- Notifications
- Access USDA.gov's nutritional database

User Interface Mocks

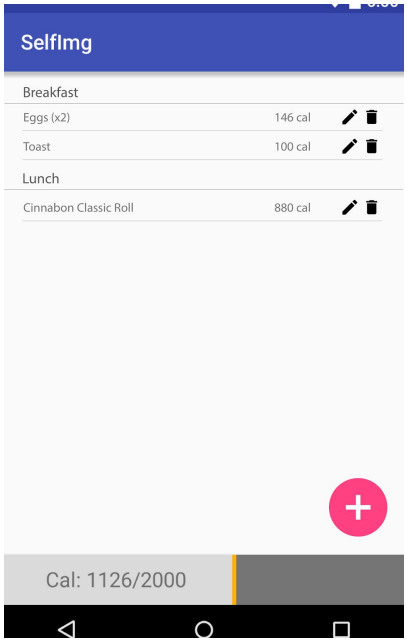
Main Activity: Current Image Viewer and Archive



This is the main activity for SelfImage. Here the user will be prompted to take their daily picture. The goal of this app is to collect enough pictures of the user that the user can then quickly cycle through all of the previous pictures, revealing any progress the user has made towards their goal.

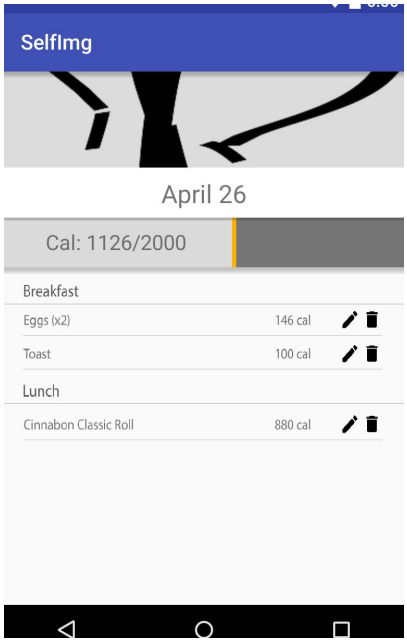
On the bottom of this screen is a calorie progress bar. This displays how many calories the user has logged in their meal plan. The calories in their meal plan will be compared against their quota and their progress will be shown here.

Meal Diary



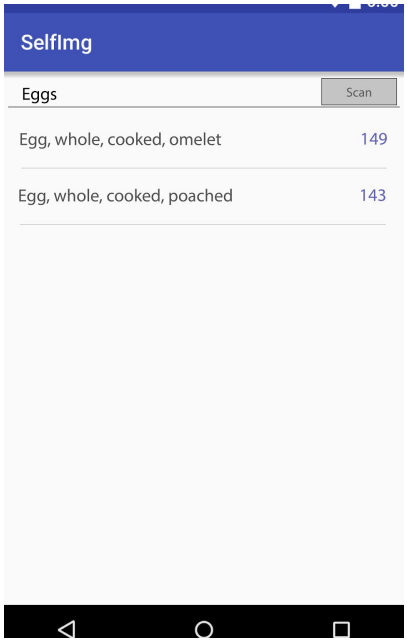
This is where the user will manage their meal diary. By pressing the FAB on the bottom right, the user will be able to log a new food item into their plan.

Previous Days



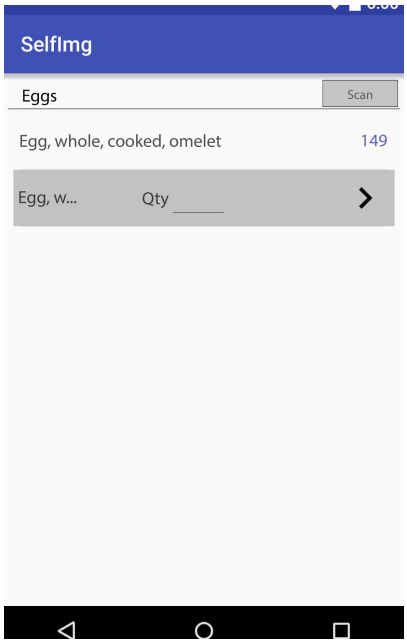
By clicking an image of a previous day, the user is able to see their meal diary for that day, as well as their total calories consumed.

Food Item Search



This screen is is where the user will search for food items. By typing keywords into the search bar on the top, the user can retrieve a list of similar food items from USDA.gov's database and then add which applies to their meal diary.

Food Item: Add Quantity



This mock is of the previous screen, however, it showcases how the user will specify how many food items they intend to add to the meal diary. Such as when the user wants to add 2 eggs.

Key Considerations

How will your app handle data persistence?

SelfImage will handle data persistence by using a Content Provider alongside an SQLite database. The database will retain previous meal diaries, and custom food items which the user might add. Images will also be stored locally on the device.

Describe any corner cases in the UX.

SelfImage will utilize the standard toolbar for navigation and will not override any system buttons.

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

Libraries this project will include:

- ZXing
    - ZXing will be used to allow users to scan food items instead of searching for them from an online database.
- Android Image Slider - Daimajia
    - The image slider library will be used within the SelfImage's main activity. While it might be possible to achieve the same results with a viewpager, using Daimajia's library will take much of the leg work out of its design while keeping it visually pleasing.
- Logger - Orhanobut
    - This will primarily be used during development, but as an external library it should be listed.

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

1. Create a new Android "Hello World" application.

2. Change all relevant string variables to reflect the new app. For example, set app_name to SelfImage.
3. Import all libraries into the project through a Gradle sync and validate their functionality.

Task 2: Implement UI for Each Activity and Fragment

1. Build UI for Main Activity
2. Build UI for meal diary
3. Build UI for Food Item Search
4. Design custom list items as needed
5. Do not incorporate Material Design until the application can be navigated.

Task 3: Establish MainActivity functionality

1. Create a week of dummy data for MainActivity
2. Set up a ViewPager or other view which will allow the user to traverse through all previous images.
3. Rig the FAB to launch the camera activity using an intent, and use the result for the current image.

Task 4: Create Notifications

1. Create notifications which prompt the user to take their daily image if the have not done so already. Pressing this notification will bring up MainActivity.
2. Add options relevant to these notifications in the options activity.

Task 5: Establish Food Search Functionality

1. Rig the calorie progress bar at the bottom of main activity to launch the meal diary activity.
2. Rig the FAB in the meal diary activity to switch to the add food fragment.
3. Setup the search bar to display food item results into the list below.
4. Setup the scan button to launch ZXing and search for food by barcode.

Task 6: Establish Meal Diary Functionality

1. Create the SQLite database along with a ContentProvider which will allow the app to update and retrieve meal diary data.
2. Setup the meal diary fragment to display the current meal diary.

3. Once this has been established, allow the user to add food using the add food fragment, and remove food using a UI button displayed next to the listed food items.

Task 7: Implement Material Design

1. Create transitions between activities.
2. Allow the user to click on previous images to show that day's meal diary.

Task 8: Create Timelapse Ability

1. Create an activity that will traverse through all images in quick succession.

Task 8: Finalize App

1. Check for major issues (ie. Internet connectivity related crashes)
2. Sign and submit application

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**"