GitHub Username: almatr

Caloriya

Description

An application that will contain the recipes with the feature to search recipes by number of calories the recipe has. The application will display all of the recipes for the user but it will have the option for the user to limit the number of recipes displayed by the caloric amount.

This application will also give an approximate estimate of daily caloric intake for a person based on their weight and their weight goal (loose, gain or maintain weight).

Why this application? Many people are not happy with their weight and are not aware of the caloric values of the food they are cooking at home. Knowing the caloric values of your homecooked meals and being able to find recipes that meet your daily caloric needs can assist in achieving your weight goal.

Intended User

This application is intended for the following users:

- User that wants to loose weight
- User that wants to maintain the weight
- User that wants to gain weight
- User that is just browsing to get idea of what to cook today.
- User learning to cook.

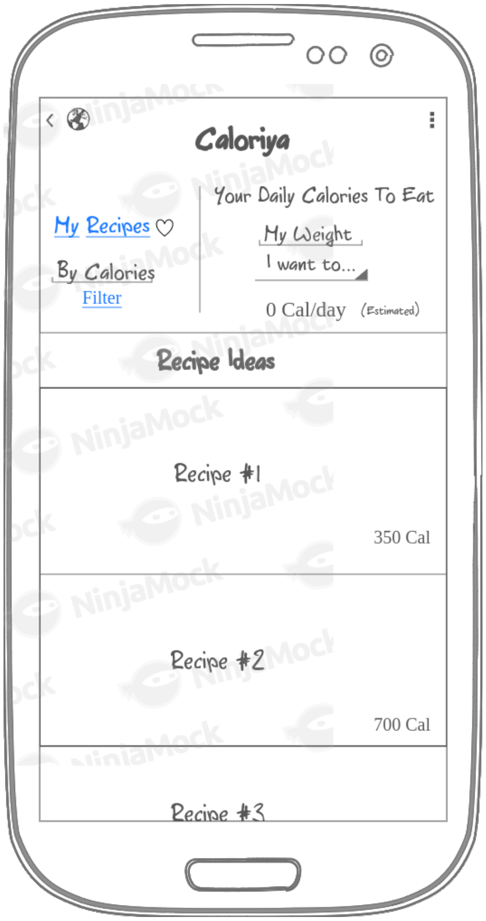
Features

Some of the features of the Caloriya application:

- Displays many different recipes, cuisines
- Displays calories per serving for each recipe
- User can calculate estimated daily calories to eat
- Spinner to select user’s weight goal (loose/maintain/gain weight)
- Filters recipes by caloric value
- Photo of every recipe
- Recipe instructions
- User can favorite or unfavorite the recipe
- Saves favorite recipes so user can view them under “My Recipes” link
- A widget that takes user to the favorites recipes home screen

User Interface Mocks

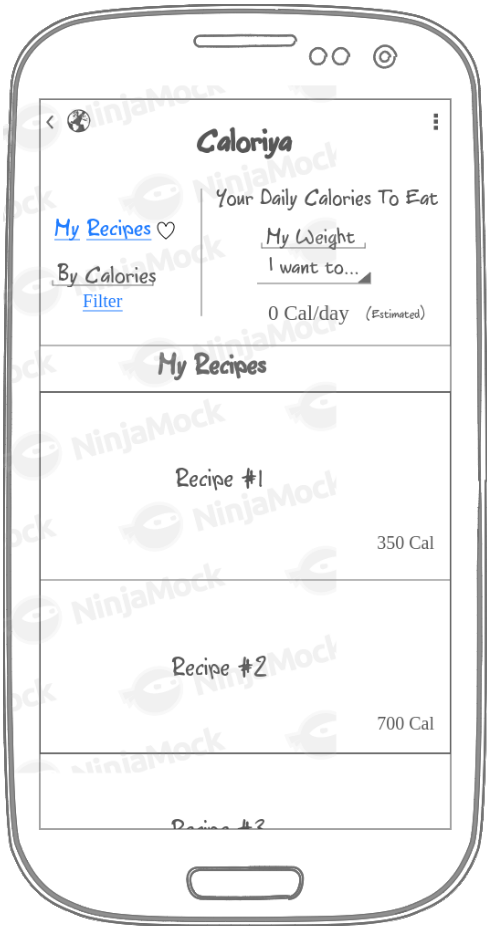
Screen 1



This is MainActivity.xml screen. With the start of the application, all of the recipes will be loaded as a list. To filter recipes by calories, a user can enter calories (as a numeric value) in the field “By Calories” and press the link “Filter” to display recipes that are less or equal to the entered numeric value. In addition, the user have an option to view their favorite recipes(previously favored by user) under “My Recipes” link. The list of recipes contains a field with an image, recipe name, caloric value and is clickable.

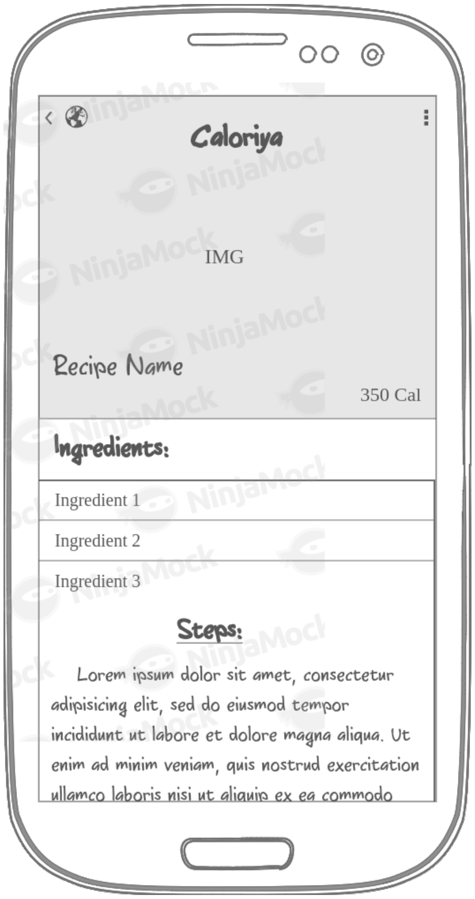
The user also has an option to estimate his/her daily caloric needs based on his/her current weight and weight goal. This is a simple calorie calculator that does not take in consideration other factors such as activity level, person’s height, gender, age, and for this reason the word “estimated” has shown in parenthesizes next to the calorie result.

Screen 2



This is also MainActivity.xml screen but showing favorite recipes after user clicked on “My Recipes” link. The only change on this screen is the title above recipe list that now shows “ My recipes” as the label, as well as the recipe list that is now showing only recipes that were favored by user. All other features and functionalities are the same as in the Screen 1.

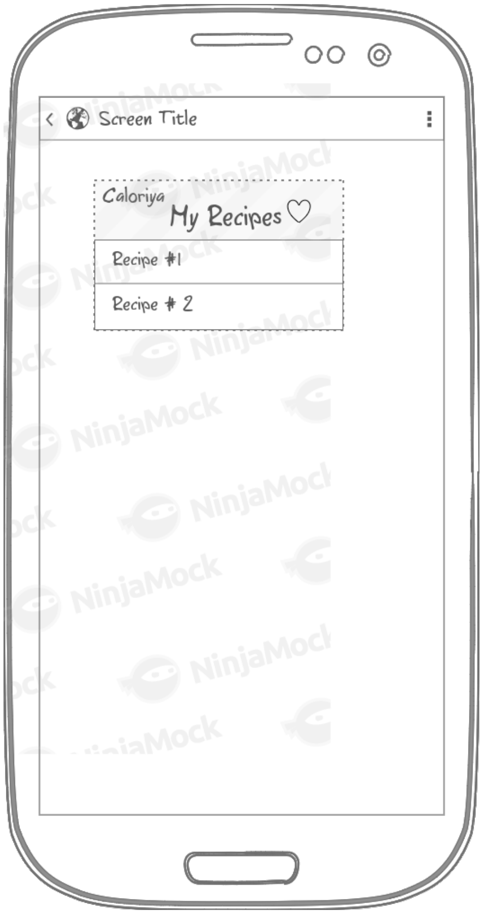
Screen 3



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo

This is a DetailActivity.xml screen. It displays the recipe information after user clicks on the recipes in the main screen. The user can see the photo of the recipe item along with scrollable ingredients and steps information.

Screen 4



This is a screen with the widget for Caloriya application. This widget will show favorite recipes with a small, scrollable list. When the user clicks on the widget, the user will be taken to the favorite recipes in the application.

Key Considerations

How will your app handle data persistence?

The application will use the SQL database along with content provider to store favorite recipes for the user. The user’s personal calculated calories will be stored using SharedPreferences.

Describe any edge or corner cases in the UX.

The user will be presented with the list of recipes regardless of recipes being filtered by their caloric values or not. By clicking on one of them, the user will be taken to the recipe detail screen. In the detail screen, a user can give thumps up or down to favorite or unfavorite the recipe.

The user will be able to go back to main screen either by clicking the back button on the phone or back arrow in the toolbar.

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

To design my application I will be using Android Support Library. For Google Play services I will be using AdMob and Analytics. AdMob library will be used for display of ads while Analytics will be used to track the user's activity such as number of screen views.

The application will use Picasso to display photos of recipes.

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

Google Mobile Ads – this will be used to get revenues from ads

Google Analytics – this will be used to collect data about screen views

Next Steps: Required Tasks

Task 1: Project Setup

- Create the blank project in Android studio
- Set up gradle libraries and dependencies, minSdkVersion(17), targetSdkVersion(27)
- Edit Manifest file to include application name, permissions and child activity declaration.
- Edit colors.xml and styles.xml to include appropriate color and themes for the application
- RTL to be supported through the application (manifest file and layout attributes)

Some of the dependencies and their versions used:

```
implementation 'com.android.support:appcompat-v7:27.1.1'

implementation 'com.android.support:cardview-v7:27.1.1'

implementation 'com.android.support:recyclerview-v7:27.1.1'
implementation 'com.squareup.picasso:picasso:2.5.2'
implementation 'com.android.support:design:27.1.1'
implementation 'com.google.android.gms:play-services-ads:11.8.0'
implementation com.google.android.gms:play-services-analytics:16.0.3
```

Task 2: Implement UI for Each Activity and Fragment

1 . Build UI for MainActivity

- implement toolbar with correct application name and link to show favorite recipes
- under toolbar implement options to calculate user's daily caloric needs based on their weight and weight goal(ex. Loose, maintain, gain weight)
- add recyclerview to show list of recipes (all)

2. Build UI for DetailActivity

- implement toolbar with back arrow
- add ImageView with Recipe name and calories showing over image (FrameLayout)
- add thumps up and thumps down to enable favoring
- add recipe ingedianets as a list
- add recipe steps TextView

Task 3: API and java files

Sign up to get API key to get recipe data in json (https://developer.edamam.com/edamam-docs-recipe-api)

This application will be written in Java language.

1. MainActivity.java

- Work on MainActivity.java to set up calorie calculator and Recyclerview
- create dropdown menu for the user to pick their weigh goal
- using asyncTask, get the recipes
- display clickable pictures of recipes in Recyclerview using Picasso
- implement intent activity for recipe clicks
- implement SharedPreferences to save user's daily calculated calories

2. DetailActivity.java

- In FrameLayout display picture of the clicked recipe using Picasso
- pick up recipe info from the intent to display in the DetailView

Task 4: SQLite and recipe favoring

- Create SQL database with content provider and loader
- in DetailView, implement thumbs up/down button to work with SQL database to add/remove recipes
- Under recipe DetailView, click on the thumbs up button to favorite recipe
- Navigate to MainActivity screen and click on “Favorite recipes” to check if SQL database sincs with thumbs up/down button

Task 5: Widget

- Implement widget to display favorite recipes main screen when clicked
- The widget will appear as a list of favorite recipes

Task 6: Other

- Implement Google Play Services (AdMob and Analytics)
- Implement savedInstanceState to avoid unnecessary call to AsnycTask on phone rotation
- Error handling
- Application accessibility (RTL, navigation using D-pad)
- Implement toast for connection loss
- Clean up code and edit strings.xml file to use resources from there
- Sign application for publishing