

[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: Build the Base UI](#)

[Task 3: Implement the Authentication](#)

[Task 4: Restaurant List](#)

[Task 5: Menu Items](#)

[Task 6: Maps](#)

[Task 7: Add Menu Items](#)

[Task 8: Favorites](#)

[Task 9: Add Calculators](#)

**GitHub Username:** BytePair

## Keto Kodex

### Description

Shows low carb options from all of your favorite restaurants. Perfect for people following a low carb or ketogenic diet who want to eat out and stick to their macros without the guesswork.

### Intended User

Anyone following a low carb or ketogenic diet who eats out. Even though many restaurants have nutrition information posted, it is a pain to navigate to each restaurant's website, download a pdf, and then scan through and try to build a meal that fits into your daily calorie or carb limit.

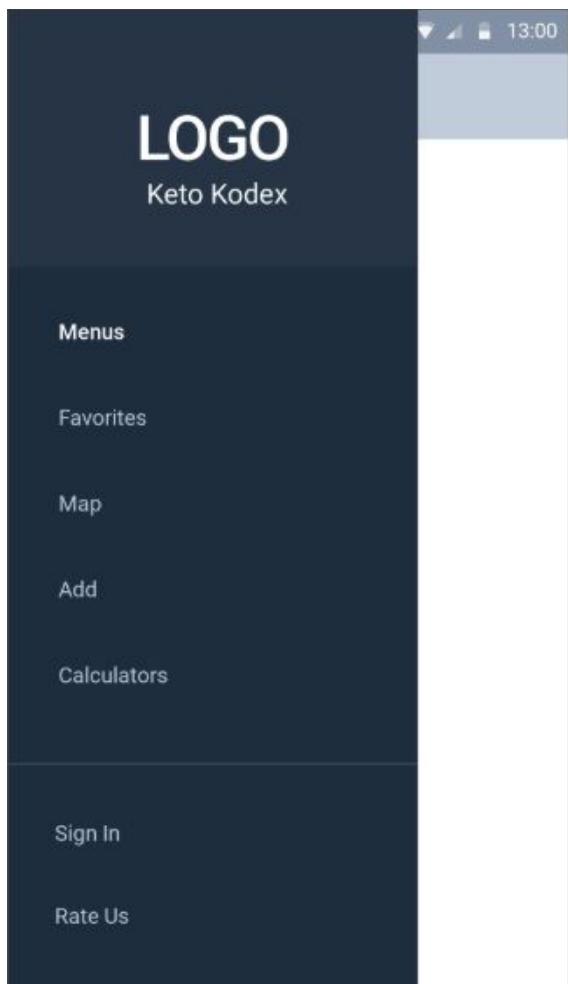
## Features

- Shows a list of restaurant menus with low carb food choices
- Items in menu have their own detailed breakdown (calories/carbs/fat/protein)
- Google maps integration to easily find something close
- Add custom meals if you find something not listed
- Save your favorites for later
- Calorie and macro calculator

## User Interface Mocks

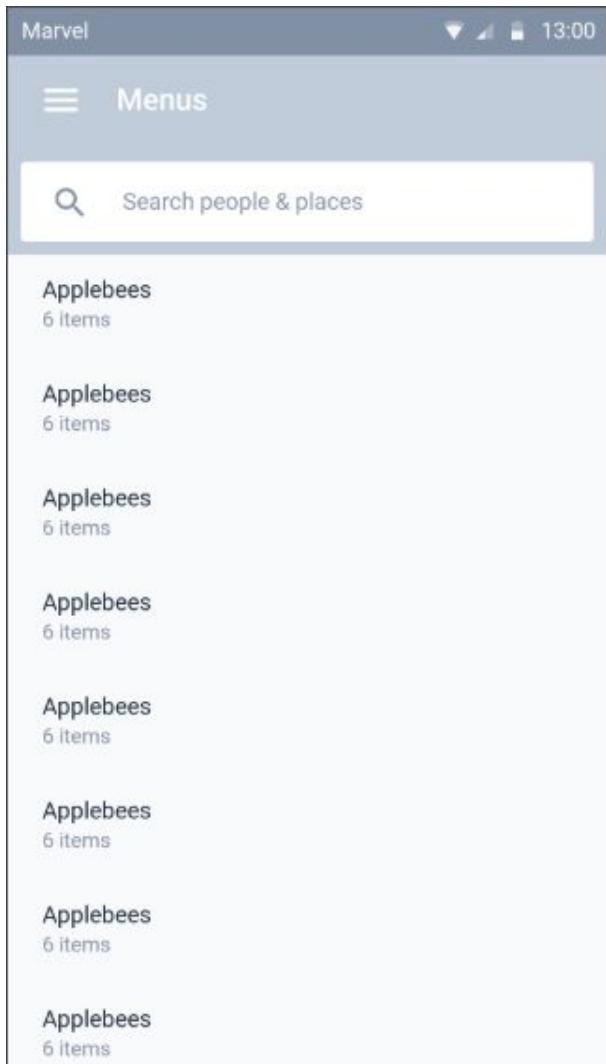
### Screen 1

Main menu of the app. Easy navigation to all features.



## Screen 2

Main point of the app, the list of restaurant menus. User can search if list seems too long. Each restaurant can be clicked on to view the available meals.



## Screen 3

After clicking on restaurant, user gets the menu with small description and vote counter.



## Screen 4

After clicking on menu item, user gets more details, and can add to favorites.

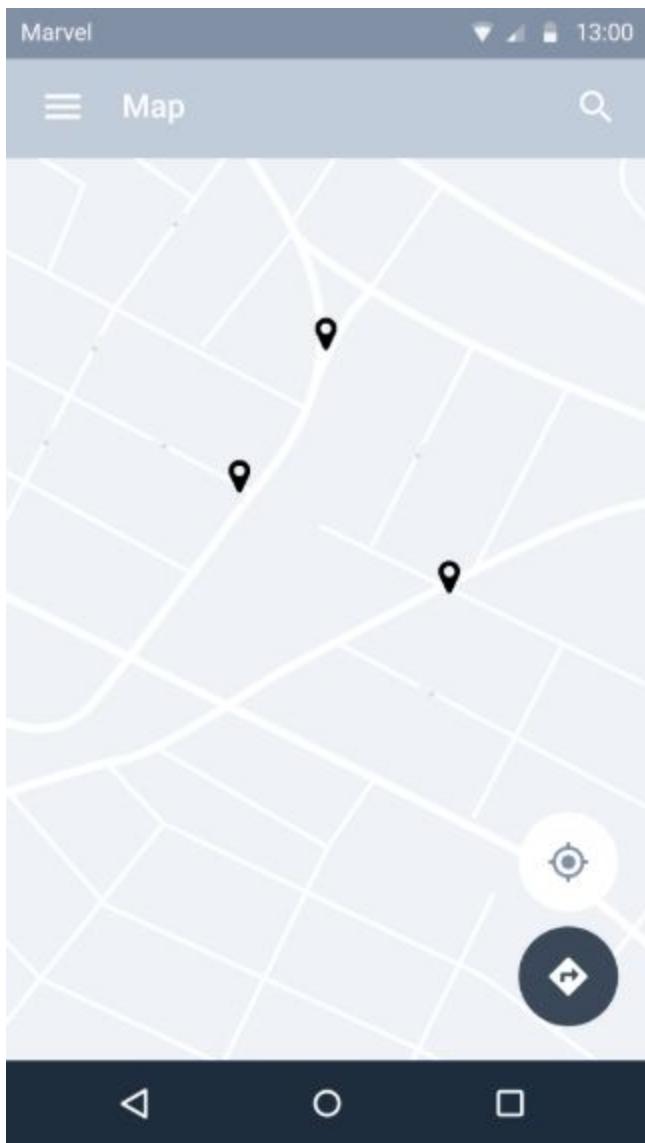


## Screen 5

Favorites is a list of menu items that user has added to favorites. See screen 3.

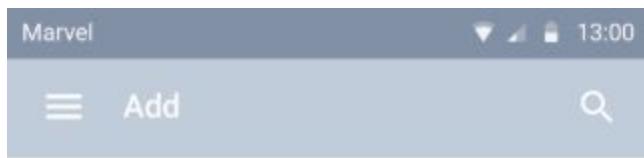
## Screen 6

Map should list all locations with a menu in the given area. User can search manually or click a button on the restaurant screen to search the map for individual places.



## Screen 7

User can add their own custom menu items as long as they are signed in.



Restaurant

Menu Item

Calories

Carbs

Protein

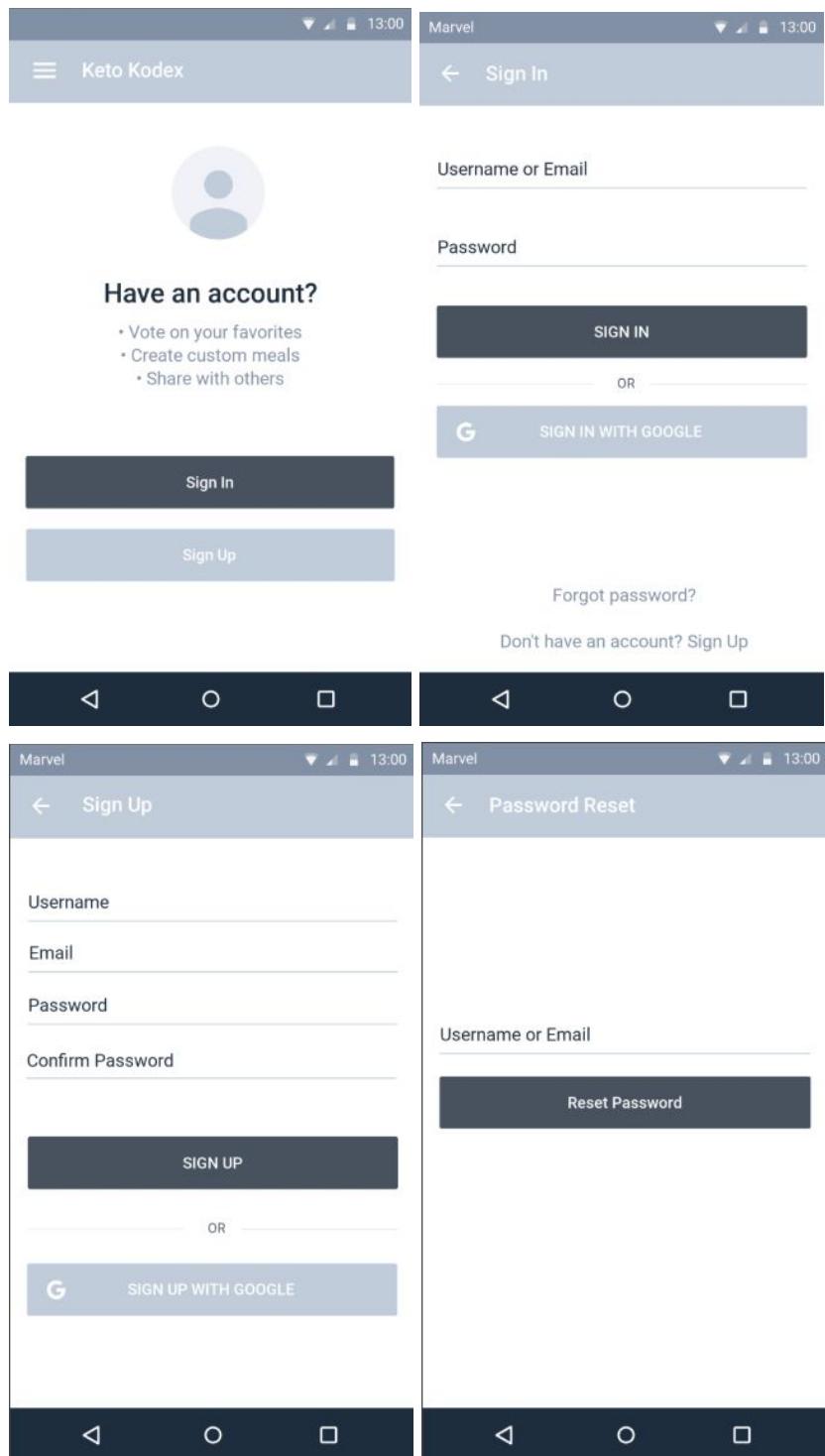
Fat

SAVE



## Screen 8-11

User authentication screens.



## Screen 12

BMR Calculator

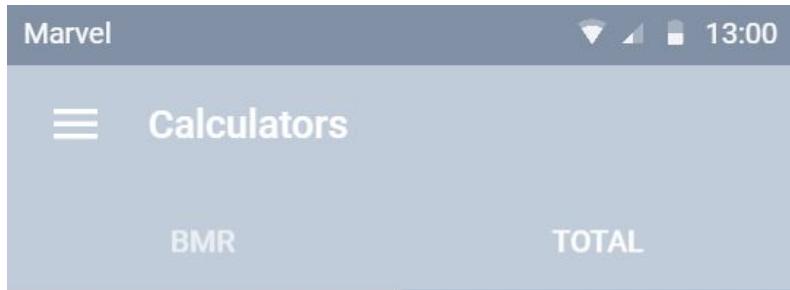
The screenshot shows a mobile application interface for a BMR calculator. At the top, there is a header bar with the text "Marvel" on the left and a battery icon with the time "13:00" on the right. Below the header, the title "Calculators" is displayed next to a three-line menu icon. A horizontal navigation bar at the bottom of the screen has two tabs: "BMR" on the left and "TOTAL" on the right. The main content area contains several input fields and controls:

- Age:** The value "24" is entered into a text field.
- Gender:** A radio button for "Female" is selected.
- Units:** A radio button for "Imperial" is selected.
- Height:** The value "5" is entered into one text field and "7" is entered into another adjacent text field.
- Weight:** The value "178" is entered into a text field.
- BMR:** The calculated result "1248" is displayed prominently in large, bold, dark blue text.

At the bottom of the screen is a dark blue navigation bar with three white icons: a triangle pointing left, a circle, and a square.

## Screen 13

Total daily calorie requirement calculator



Activity Level

- Sedentary
- Lightly Active
- Moderately Active
- Very Active
- Athlete/Bodybuilder

Total Daily Calories:

**1972**



## Key Considerations

### How will your app handle data persistence?

All data will be stored remotely. User authentication and features that require it will depend on Google Firebase. The main tables in the db are going to be restaurants, menu items, and favorites.

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

Navigation is the main concern for now. There are lots of ways to click into a menu item and I have to keep track of where the user came from and make sure they are returned to the same place.

Also need to handle some edge cases in the UI like if a restaurant's logo is missing or cannot load, or how to hide/show/disable things if the user is not signed in.

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

- Picasso will be used to handle any image loading for the logos since I have experience with it in this course.
- Retrofit will be used to handle api calls since it is easy to add endpoints when/if the app requires it in the future.

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

Other than firebase, the only other required google service is going to be location and the maps api. Users will be able to search around their current location for a restaurant and if one is found, they should be able to open it in google maps / waze / etc. for navigation.

## 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

- App is written solely in the Java programming language.
- Submission must use stable release version of all libraries, Gradle, and Android Studio. Debug/beta/canary versions are not acceptable.

## Task 2: Build the Base UI

- Create new project with navigation drawer
- Set min/target versions
- Set colors

## Task 3: Implement the Authentication

- Build UI for main sign in/up page
- Build UI for 'Sign In' page
- Build UI for 'Sign Up' page
- Build UI for password reset page
- Implement firebase and hook up pages

## Task 4: Restaurant List

- Build UI for restaurant list
- Build Retrofit API
- Build recyclerview, adapter, presenter, etc. for restaurants

## Task 5: Menu Items

- Build UI for menus
- Build recyclerview, adapter, presenter, etc. for menus
- Build UI for menu item detail

## Task 6: Maps

- Build UI for Map
- Get user's location and handle asking for permissions
- Populate map with available restaurants
- Add search from search bar and restaurant page

## Task 7: Add Menu Items

- Build UI for add menu item
- Ensure it is only available if user is authenticated
- Ensure new menu item shows up in main list

## Task 8: Favorites

- Add FAB for add to favorites
- Add endpoints for add/remove/fetch favorite(s) to retrofit api
- Build UI for favorites page
- Build recyclerview, adapter, presenter, etc. for favorites

## Task 9: Add Calculators

- Build UI for BMR/BMI calculators
- Ensure it is only available if user is authenticated
- Ensure new menu item shows up in main lis