Table of contents

GitHub Username: SarahEhabMostafa

Carb Calculator

Description

Carb Calculator is an application that solves a problem that I face every day at least twice a day. Well, me and thousands of other diabetics all around the world. The problem I'm talking about is counting the number of carbohydrates in a meal to calculate the amount of insulin one should take for this meal. Carb Calculator will solve this problem by providing the user a wide range of items to choose from to add to a meal and it will calculate the number of carbohydrates in this meal. It will also provide the user with an option to add new items if they're not available on the list.

Intended User

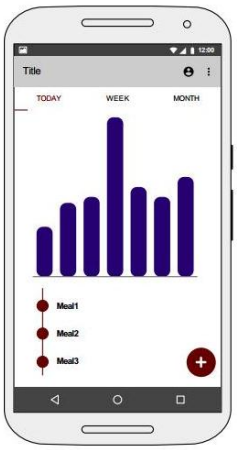
This application is for diabetics who take insulin or anyone who wishes to calculate and track the amounts of carbohydrates that he eats

Features

- Create a meal by choosing food/beverage items from the list
- Add new items to the list if it doesn't exist
- Favorite items
- Saves user's meals
- Tracks the user's meals over the days
- Provides visual graphs that show the user's meals daily/weekly/monthly

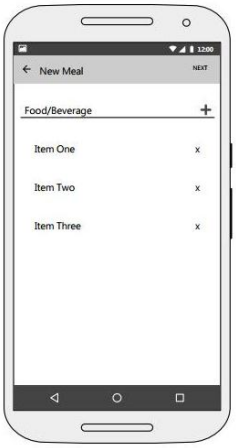
User Interface Mocks

Main Screen



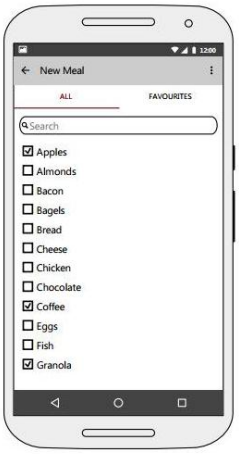
This is the main screen of the application. Here the user will see a visualization of the amounts of carbs that he'd eaten. He can choose to view today, this week or this month.
There is also a list of the meals that the user ate today. Clicking on one of these meals will take him to view more details about the meal (Meal Details Screen)
From this screen the user can also create a new meal by clicking on the Floating Action Button.

New Meal - Step 1



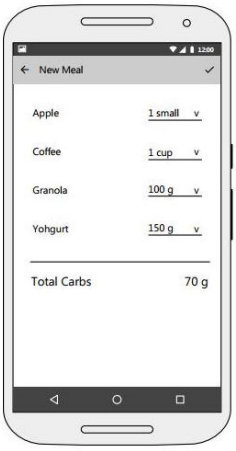
This is the screen that the user will see after clicking on the Floating Action Button (in the Main Screen). Here is where the user will add the food/beverage items to his meal. The user can start typing an item's name to see suggestions or he can click on the little (+) sign in the text field to open a full list of items to view and select from. If the user wishes to remove an item after adding it, he can press on the small (x) next to the item.

New Meal - Step 1(2)



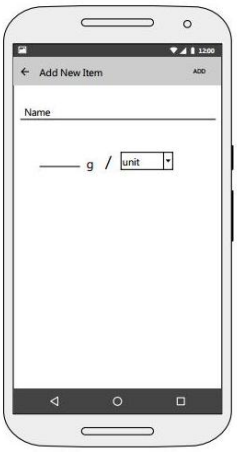
This is the full list of food/beverage items that the user can skim through or search to find the item he's looking for and select it to be added to the meal. He can select to view all the items or just his favorite items. Items can be marked as favorite by swiping an item in the full list of items. If the user doesn't find an item that he's looking for, he can select 'ADD ITEM' from the overflow menu to add an item to the list of items.

New Meal - Step 2



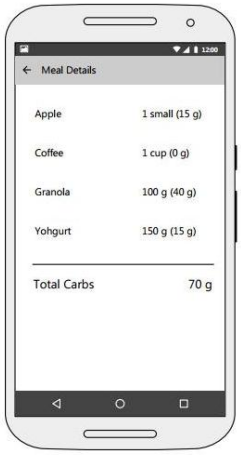
In this screen, the user specifies the amount of each item and he'll see the total number of carbohydrates for all the items in the end of the screen. After selecting the items and entering the amounts and pressing done, the user can now enjoy his meal without having to worry about any calculations. The meal can now be viewed from the Main Screen.

Add Item



If the user has searched for an item and it was not on the list and he decided to add it to the list, this is the screen that he will see. All he has to do is specify a name and the amount of carbohydrates per a unit (He can add more than one unit if he wishes). And that's it! The newly added item is now on the list.

Meal Details



The user can press on one of the meals in the Main Screen to view its details

Key Considerations

How will your app handle data persistence?

Application will use a Content Provider to persist the data. I will build one.

Describe any corner cases in the UX.

To mark an item as favorite, user should swipe the item

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

- WilliamChart to visualize the amounts of carbohydrates consumed over the days as graphs
- Steppers to represent today's meals in a timeline-like view
- Android support library

Next Steps: Required Tasks

Task 1: Project Setup

- Create project
- Configure all needed libraries

Task 2: Data Storage preparation

- Build the database
- Build the content provider
- Build an interface class to ease the communication with the content provider

Task 3: Implement UI for Each Activity and Fragment

- Build UI for MainActivity and fragment
- Build UI for NewMeal fragments and activities
- Build UI for AddItem fragment and activity
- Build UI for MealDetails fragment and activity

Task 4: Integrate the data

- Integrate the data on the Main Screen to get today, this week and this month's meals
- Integrate the data on the Main Screen to get the list of today's meals
- Integrate the data on the New Meal Screen to get the food list
- Integrate the data on the New Meal Screen to handle adding and removing from favorites
- Integrate the data on the Add Item Screen to handle saving an item to the database

Task 5: Accessibility

- Implement content descriptions for all items to support Talkbalk
- Handle error cases