

## 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**”

---

[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.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

**GitHub Username:** MaryiaR

# Calories Counter

## Description

This app helps you on your way to get in better shape! Set your goals, track your calorie food and lose weight.

## Intended User

Everyone who cares about healthy lifestyle.

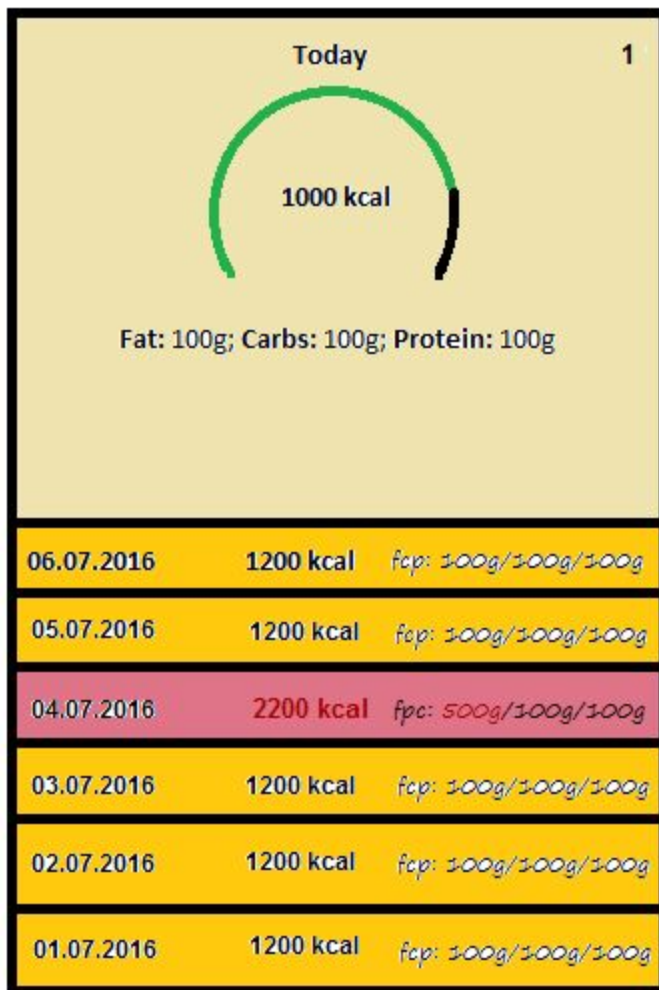
## Features

- Simple and user-friendly interface
- Quick search through a big product database
- Detailed statistic for a week
- Ability to create your own products

## 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.

### Screen 1



Main screen with today progress and weekly statistic. Screen 2 is opened by clicking on list item or today view.

1 - settings icon, opens Screen 6

## Screen 2



Details screen with list of food

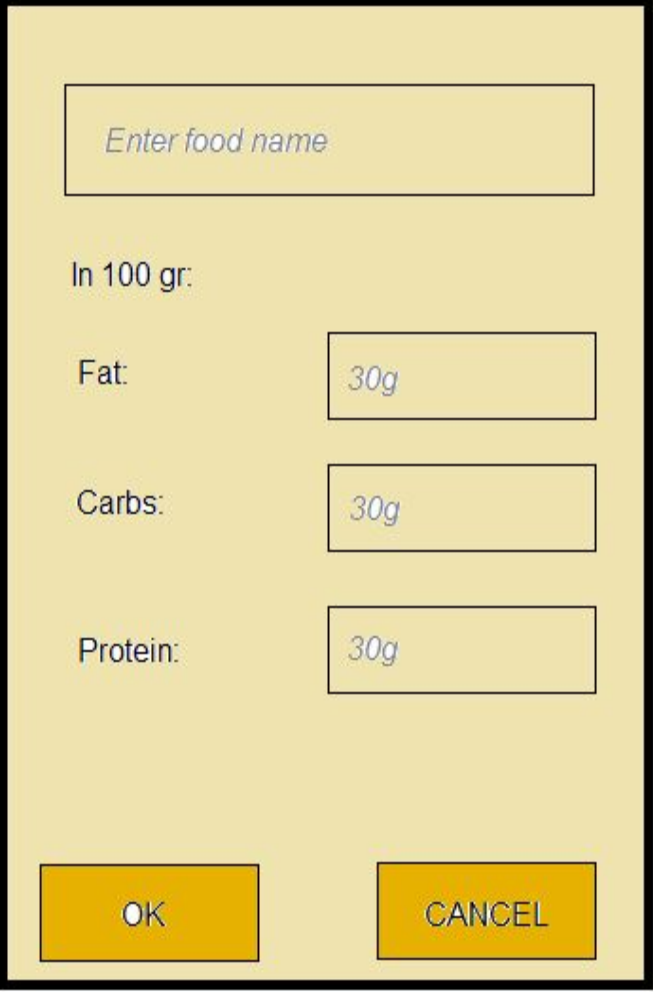
“+” button is visible only for today, opens Search Product screen (Screen 3)

### Screen 3

The wireframe for Screen 3 is a vertical rectangle with a black border. It is divided into four horizontal sections. The top section has a light beige background and contains a text input field with the placeholder text "Start entering food name". In the top right corner of this section is a black plus sign (+). The second section is a solid yellow rectangle containing the text "autocomplete option 1". The third section is also a solid yellow rectangle containing the text "autocomplete option 2". The fourth section is a solid yellow rectangle containing the text "autocomplete option 3...". The bottom section is a large, empty light beige rectangle.

Search food screen with autocomplete. Clicking on list item opens dialog from Screen 5  
“+” button opens Add New Product screen (screen 4)

## Screen 4



The image shows a mobile application screen titled "Screen 4" with a light yellow background and a black border. At the top, there is a text input field with the placeholder text "Enter food name". Below this, the text "In 100 gr:" is displayed. Underneath, there are three rows of input fields for nutritional information: "Fat:", "Carbs:", and "Protein:". Each of these labels is followed by a text input field containing the value "30g". At the bottom of the screen, there are two yellow buttons with black text: "OK" on the left and "CANCEL" on the right.

Add New Product screen

"OK" button opens dialog from Screen 5

"Cancel" button returns to Screen 3

## Screen 5

Product weight dialog for selected product. "Product1" - name of selected product

"OK" button adds product to today's list

"Cancel" button returns to Screen 3 or Screen 4

## Screen 6

Daily Calorie Needs: **1600 kcal**

Age

Years

Gender

☒ Female ☐ Male

Current Weight

Kilos

Height

cm

Exercise level

3 times/week

▼

Goal

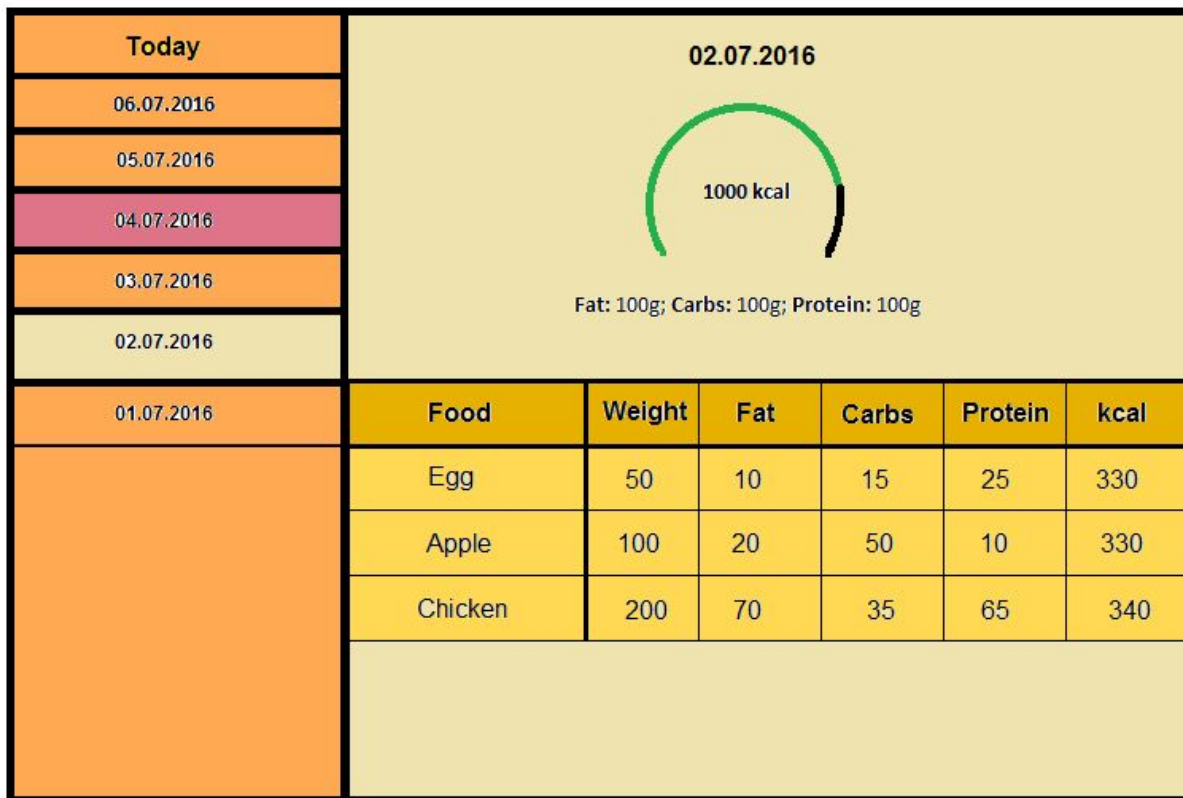
Fat Loss

▼

CALCULATE

Settings screen for calculating calorie need.

"Calculate" button updates existing calorie need value

**Tablet screen:****Key Considerations****How will your app handle data persistence?**

Content Provider will be used to store user daily reports and user created products. Personal data for calculating calorie need will be stored in Shared preferences. App will backup all saved data with the help of BackupAgentHelper.

**Describe any corner cases in the UX.**

Navigation in the app is based on buttons and clicking on daily views. Back button returns to previous screen or closes the app if it's main screen.

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



OkHttp library will be used to handle communication with the FatSecret Platform API for using autocomplete in search.

## 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

- Create a project
- Configure libraries
- Get familiar with the FatSecret Platform API
- Plan the structure of storing data in Content Provider

### Task 2: Implement UI for Each Activity and Fragment

- Build UI for Main screen
- Build UI for Details screen
- Build UI for searching product in DB and from FatSecret API
- Build UI for adding new product
- Build UI for user profile screen

### Task 3: Implement searching foods with FatSecret API

- Add OkHttp library
- Create Helper class for making requests
- Create model classes for responses
- Handle error cases and cases of no internet

### Task 4: Adding content provider and Shared preferences, implement data backup

- Create content provider for storing data
- Create tables for daily history, user products
- Setup auto-delete history that is older than 7 days
- Create Helper class to manage all communication with Content Provider
- Create shared preferences for storing profile data
- Implement BackupAgentHelper and request backup every time when data is changed

### **Task 5: Enable calculating Daily Calorie Needs**

- Define formula for calculating calorie need depending on user data
- When screen is opened fill all fields with previously saved data
- Update data in shared prefs after calculating new value

### **Task 6: Connect UI to logic**

- Load data from db on the main screen and details screen on start
- Enable autocomplete search through db and FatSecret API
- Enable adding new products to db

### **Task 7: Implement Google services**

- Integrate Admob - add test ads
- Integrate Analytics - add tracking changes of calories need and exceeding calories allowed value

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**"