CSC207 Final Project: Calorie Calculator

Group #14

[1]Zhengyu Yi

[2]Max (Haotian) Xu

[3]Haoying Zhu

[4]Shourya Harsh Vardhan

[5]Yan Lam

[6]Rasyid Rafi Pamuji

November 24, 2025

User Stories (i.e., what functionality is in your team's MVP?)

- Explain what functionality your program provides.
- You can use whatever format you want — just make sure to clearly and concisely convey this information.
- Advice: avoid too many words here and don't plan to read a full paragraph of text on the slide when presenting — just highlight the key points.
- Focus on the WHAT and not the HOW!
- Recommended time limit: [1 minute]

API Usage

- What API(s) did your team use?
- Make sure to show appropriate information (briefly).
- Recommended time limit: [30 seconds]

Data Persistence

- What data is persistent in your program?
- Either here or in one of the use case demonstrations of functionality, data persistence should be demonstrated.
- Recommended time limit: [30 seconds]

TEMPLATE: Use Case Walkthrough (1 minute)

- Briefly state your user story and which associated use case you will focus on
- Show the before and after views for when the use case executes
- Show a UML class diagram for the use case; it should be clear from the diagram that your code adheres to CA!
- Show the code for your Use Case Interactor class.
- Discuss the flow of control when your use case executes.
- [This should be rehearsed so that it is around 1 minute per member]

Zhengyu Yi Use Case Walkthrough (1 minute)

- Briefly state your user story and which associated use case you will focus on
- Show the before and after views for when the use case executes
- Show a UML class diagram for the use case; it should be clear from the diagram that your code adheres to CA!
- Show the code for your Use Case Interactor class.
- Discuss the flow of control when your use case executes.
- [This should be rehearsed so that it is around 1 minute per member]

Max Xu Use Case Walkthrough (1 minute)

- Briefly state your user story and which associated use case you will focus on
- Show the before and after views for when the use case executes
- Show a UML class diagram for the use case; it should be clear from the diagram that your code adheres to CA!
- Show the code for your Use Case Interactor class.
- Discuss the flow of control when your use case executes.
- [This should be rehearsed so that it is around 1 minute per member]

Haoying Zhu Use Case Walkthrough (1 minute)

- Briefly state your user story and which associated use case you will focus on
- Show the before and after views for when the use case executes
- Show a UML class diagram for the use case; it should be clear from the diagram that your code adheres to CA!
- Show the code for your Use Case Interactor class.
- Discuss the flow of control when your use case executes.
- [This should be rehearsed so that it is around 1 minute per member]

Shourya Harsh Vardhan Use Case Walkthrough (1 minute)

- Briefly state your user story and which associated use case you will focus on
- Show the before and after views for when the use case executes
- Show a UML class diagram for the use case; it should be clear from the diagram that your code adheres to CA!
- Show the code for your Use Case Interactor class.
- Discuss the flow of control when your use case executes.
- [This should be rehearsed so that it is around 1 minute per member]

Yan Lam Use Case Walkthrough (1 minute)

- Briefly state your user story and which associated use case you will focus on
- Show the before and after views for when the use case executes
- Show a UML class diagram for the use case; it should be clear from the diagram that your code adheres to CA!
- Show the code for your Use Case Interactor class.
- Discuss the flow of control when your use case executes.
- [This should be rehearsed so that it is around 1 minute per member]

Rasyid Rafi Pamuji Use Case Walkthrough (1 minute)

- Briefly state your user story and which associated use case you will focus on
- Show the before and after views for when the use case executes
- Show a UML class diagram for the use case; it should be clear from the diagram that your code adheres to CA!
- Show the code for your Use Case Interactor class.
- Discuss the flow of control when your use case executes.
- [This should be rehearsed so that it is around 1 minute per member]

Design (Recommended time: 2.5 minutes)

- How does your program adhere to SOLID?
  - Should aim to talk about two specific examples present in your project; touching on at least two principles.
- How does your program adhere to the Clean Architecture?
  - This can be very short, or most likely skipped entirely, depending on what each member talked about in the previous use case part.
- What is the best example of a design pattern used in your program?
  - Aim to talk about one design pattern that your team introduced — don't talk about a pattern that was already implemented in the starter code to earn full marks here.
- As appropriate, your team should make use of diagrams and other visuals to convey this information.
  - Almost always, diagrams will be more effective than showing the details of the code!
    [Image of relevant design diagram]
- Roughly, your team might aim for something 2–4 slides in total about design, but the exact number can vary depending on how you are presenting the information.

Functionality Demonstration (Recommended time: 3.5 minutes)

- Make sure to time things out so that your team can demonstrate at least the core functionality of your MVP.
- Feel free to demo live or show recordings, but you should have a working version of the program available to demo specific functionality live immediately after the 15-minute presentation if there are questions.
- Make sure to especially rehearse the demo, as it can be easy to spend too much time here!
- Focus on the most interesting parts of the program (e.g., don't spend time on things like signing up a user, changing password, or similar unless it is central to the program)
- NOTE: depending on how your team decides to present, it might be appropriate to wait until the very end to do your demo.

Code Organization (Recommended time: 30 seconds)

- Talk about how your team organized your code.
- How did you package your code?
- Tip: make sure you follow proper naming conventions throughout your code, especially package names.

Code Quality (Recommended time: 30 seconds)

- Briefly explain how your team was able to maintain code quality.
  - You might discuss how you used Checkstyle or similar tools.
  - You might discuss your approach to pull requests and code reviews.
  - For the exceptional level, reminder that your team needs to explicitly draw attention to a representative pull request demonstrating your team's approach to code quality.

The End

- Time permitting, your team can highlight anything else about the project that you want to share.
- You'll likely fill the time with covering the required rubric elements, so only say more if you have extra time to fill.
- The recommended time limits would put your presentation around 15 minutes, so your team most likely won't have time for anything else.

Highlighting text

In this slide, some important text will be highlighted because it's important. Please, don't abuse it.

Remark

Sample text

Important theorem

Sample text in red box

Examples

Sample text in green box. The title of the block is "Examples".