

[Title](#) 

user guide for Calorie Tracker

# Koh Ming En's Project PortFolio Page

# Koh Ming En's Project PortFolio Page

## Project: LifeTracker

LifeTracker is a desktop application used for encouraging users to lead a healthier life by tracking their net calorie intake. The user interacts with it using a CLI. LifeTracker is written in Java 11 and has about 4 kLoC.

Given below are my contributions to the project.

### Databases

The databases generated by LifeTracker stores all of the users information. The primary file type used in this project is csv and I used the OpenCSV v5.6 library to read and write to the databases. The databases I created are:

1. Food Database
2. Meal Database
3. Exercise Database

The sections below will discuss each database, its implementation and its contribution to the project.

### Food Database

- What it does: Stores a copy of food sold in Technoedge and its nutritional value
- Justification: This enables a more streamlined user experience as the user did not need to recall the exact food eaten and its nutritional value before inputting it in LifeTracker. The database also included methods to filter the foods for use in other parts of the project.
- Highlights: The database was designed to be as general as possible, such that it could be “plugged” into any part of the project and used.

### Meal Database

- What it does: Stores the meals that the user previously ate, including date of meal, type of meal (Breakfast, Lunch, Dinner) and food eaten.
- Justification: The information stored is used to calculate the calories consumed per day so that the user is able to better manage their weight.

### Exercise Database

- What it does: Stores the exercises that the user previously did, the amount of calories burnt and the date of exercise
- Justification: The information stored is used to calculate the calories burnt per day so that the user is able to better manage their weight.
- Highlights: The enhancement complemented the existing meal database, which tracked the meals that the user ate. The database was planned and implemented such that changes to existing code was minimised, thus ensuring low coupling. The existing command to track calories which had to be updated, was done in a way to not break its usage in other parts of the code.

## Entities

To complement the database, I also created entities to hold the information from the databases. These are the entities I created and a brief description of what they do.

1. Food: Holds the name of food, the store that sells it, and the nutritional content of the food. Has 3 child classes that further divides the food
  - Dish: A main course
  - Side: A side dish
  - Ingredient: Combined to make a full dish
2. Meal: Holds the date of meal eaten, the type of meal, and foods eaten.
3. User: Holds the user current information

## Exceptions

For error handling, I also created some custom exceptions. These are the exceptions I created and a brief description of their purpose.

1. LifeTrackerException: Base Custom Exception that the rest of the exceptions inherits from
2. InvalidArgumentsException: Raised when the user included arguments in the command that cannot be parsed.
3. InvalidDateException: Raised when the user included dates that cannot be parsed.
4. InvalidIndexException: Raised when the user selected an option that is out of selectable range.
5. InvalidMealException: Raised when the user entered an invalid type of meal
6. MissingArgumentException: Raised when the user did not include some arguments for the command
7. UnableToSaveDatabaseException: Raised when the information entered could not be saved.

## Code Contributed

[RepoSense Link](#)

## Project Management

- Released v1.0 on Github

## Enhancements to existing features

- Implemented Sorting for meals by date (PR [#75](#))
- Implemented single line CLI option for add command (PR [#77](#))
- Enhanced tracking to include exercises on top of meals (PR [#99](#))

## Documentation

- User Guide:
  - Added documentation for add, update, view, delete, filter, nutrition, exercise and track (PR [#100](#))
- Developer Guide:
  - Added documentation for Storage components (PR [#100](#))

## Community:

- Helped my groupmates with their parts of the project (PR [#55](#), [#86](#))

# Tools

- Integrated a third party library ([OpenCSV](#)) to the project (PR [#41](#))
- 