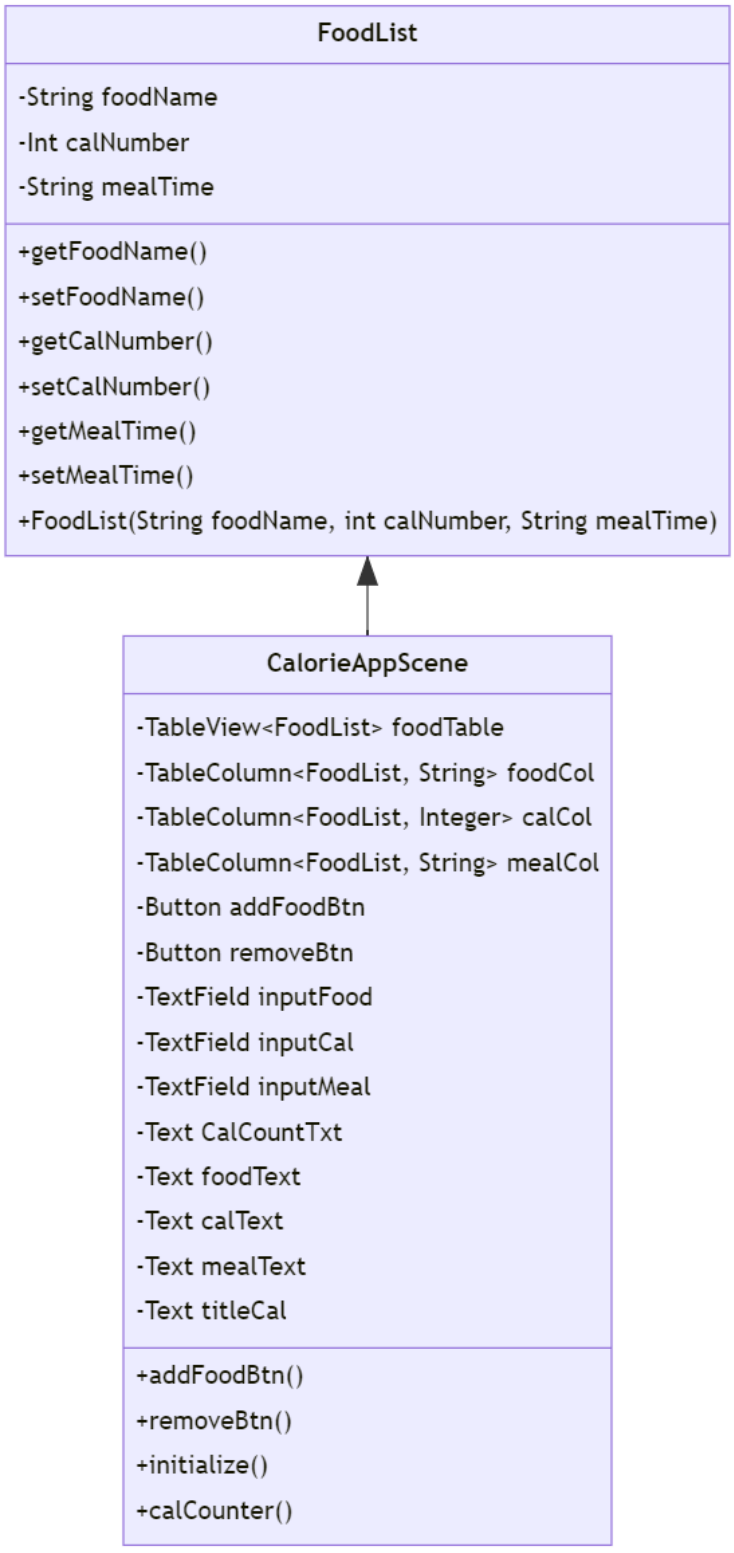
Healthy and Fit Implementation Manual

UML Class Diagram:



This application was made using Java, JavaFX, CSS, and SceneBuilder. The application allows users to input a food name, the numbers of calories associated with that food, and the mealtime (This can be breakfast, lunch, dinner, snack, or a specific time of day e.g., 11:30 AM). Originally the goal was to make the back end first and the front-end after. This changed after realizing most of the methods and variables were going to be in the controller class (CalorieAppScene).

Using SceneBuilder was helpful to make the GUI look clean and user-friendly. Also, the ability to see changes made instantly was beneficial because it allowed for more time to spend on the functionality of the application. The FXML controller skeleton that SceneBuilder generated made working on the application more efficient. It generated most of the variables used for the application in the controller class.

FoodList Class:

This class holds the variables for the food name, calorie number, and mealtime. The purpose of this class was to create a constructor that would initialize objects in the CalorieAppScene class.

CalorieAppScene Class:

initialize (method) - The purpose of this method was using the setCellValueFactory to set the TableColumns that made use of the FoodList constructor. It also used ResourceBundle to load distinct data files from the package used to make this application. URL is also used to communicate with files in the package.

addFoodBtn (ActionEvent) – The purpose of this ActionEvent was to add the food name, number of calories, mealtime, and count the total number of calories consumed. To do this an object was created and that would get the text from the TextFields. An ObservableList was used to handle changes made from the food table. Using the object and ObservableList, the data could then be placed in the table. Also, the clear method was used to reset the TextFields after clicking the add button.

removeBtn (ActionEvent) – The purpose of this ActionEvent was to remove a certain index from the food table. This in turn subtracts the total number of calories from the number of calories deleted. The change in the total number of calories will reflect instantly.

calCounter (method) – This method counts the number of calories from the table. It uses IntegerBinding to complete this task. The binding calculates the value depending on more than one source. Since the table has more than one source, this is used to count the total number of calories and assign it to a piece of text.