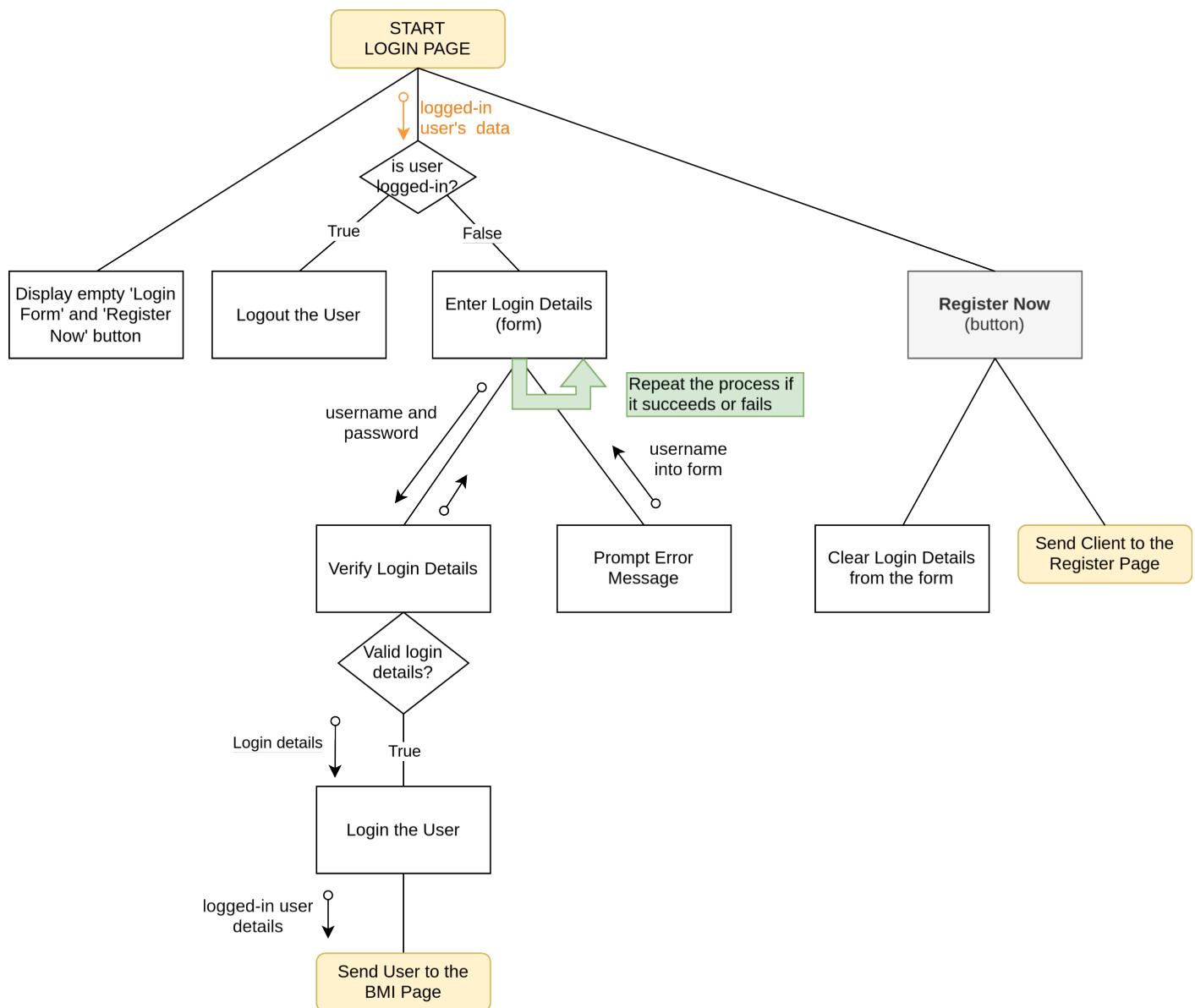
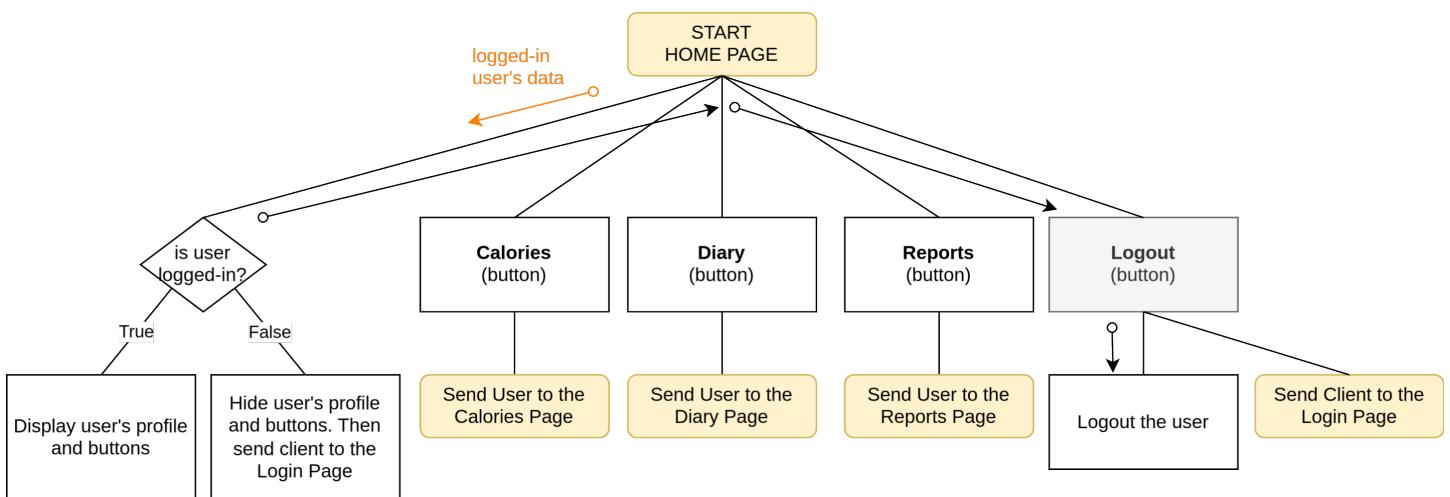
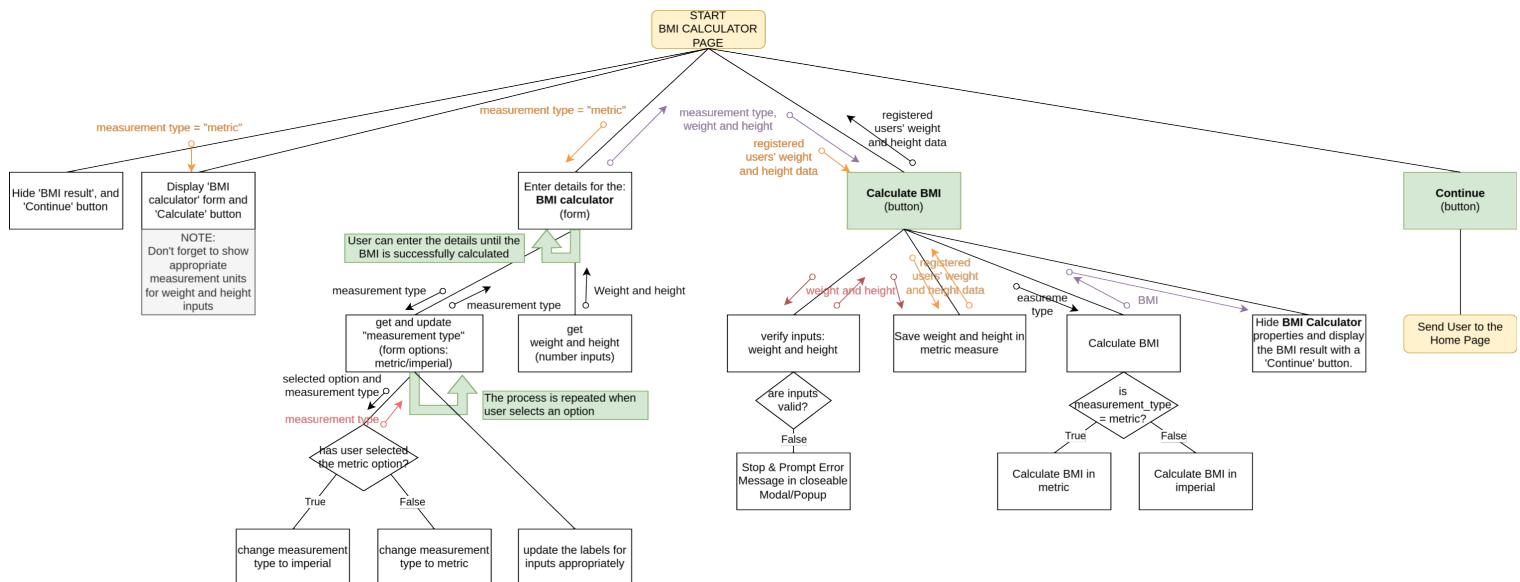
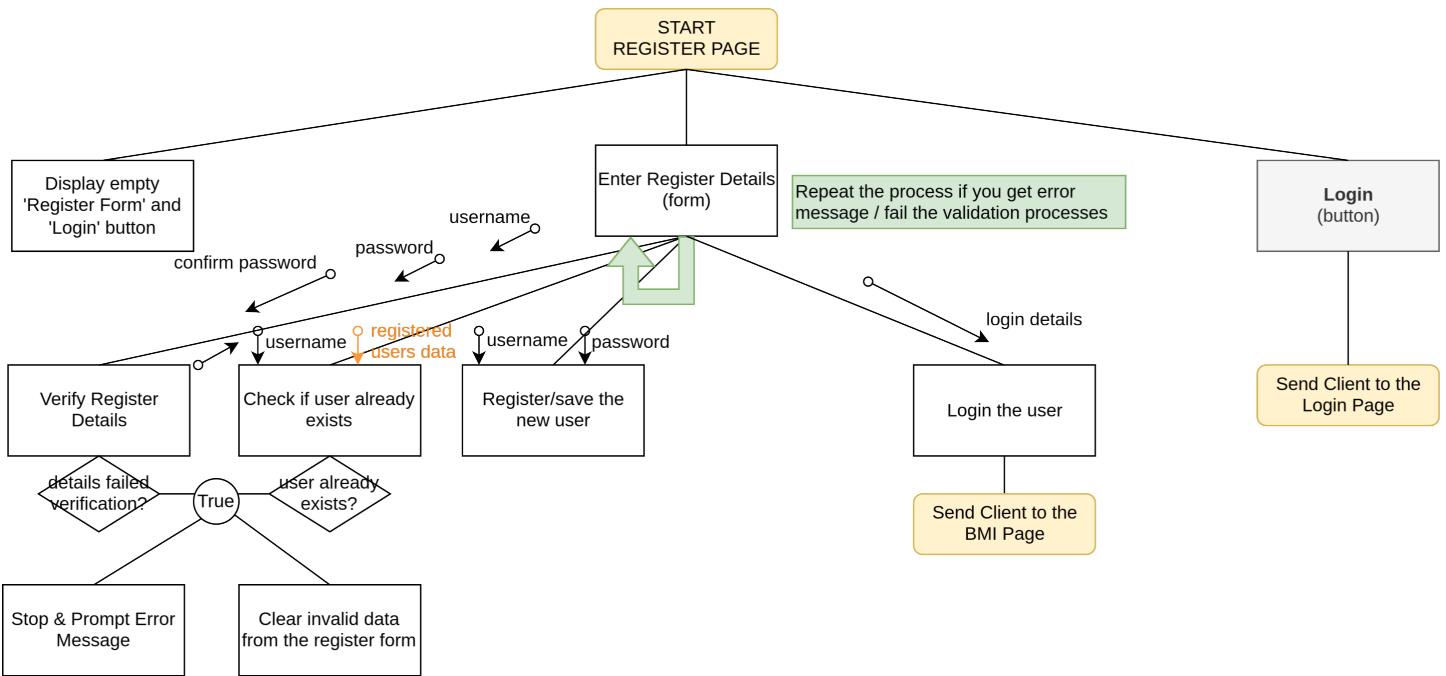


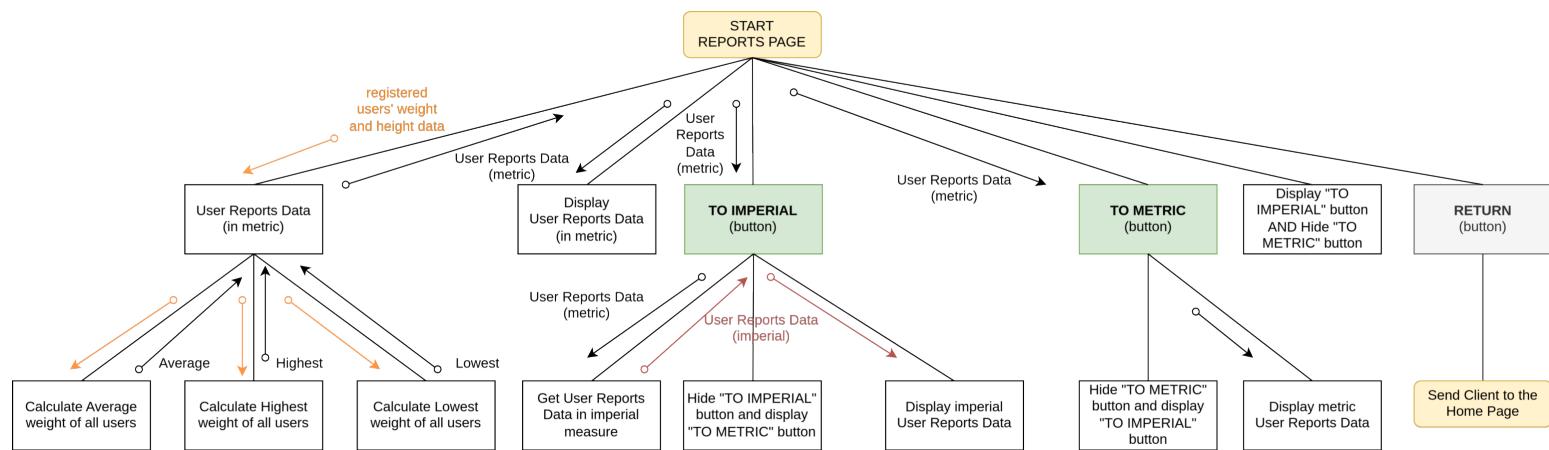
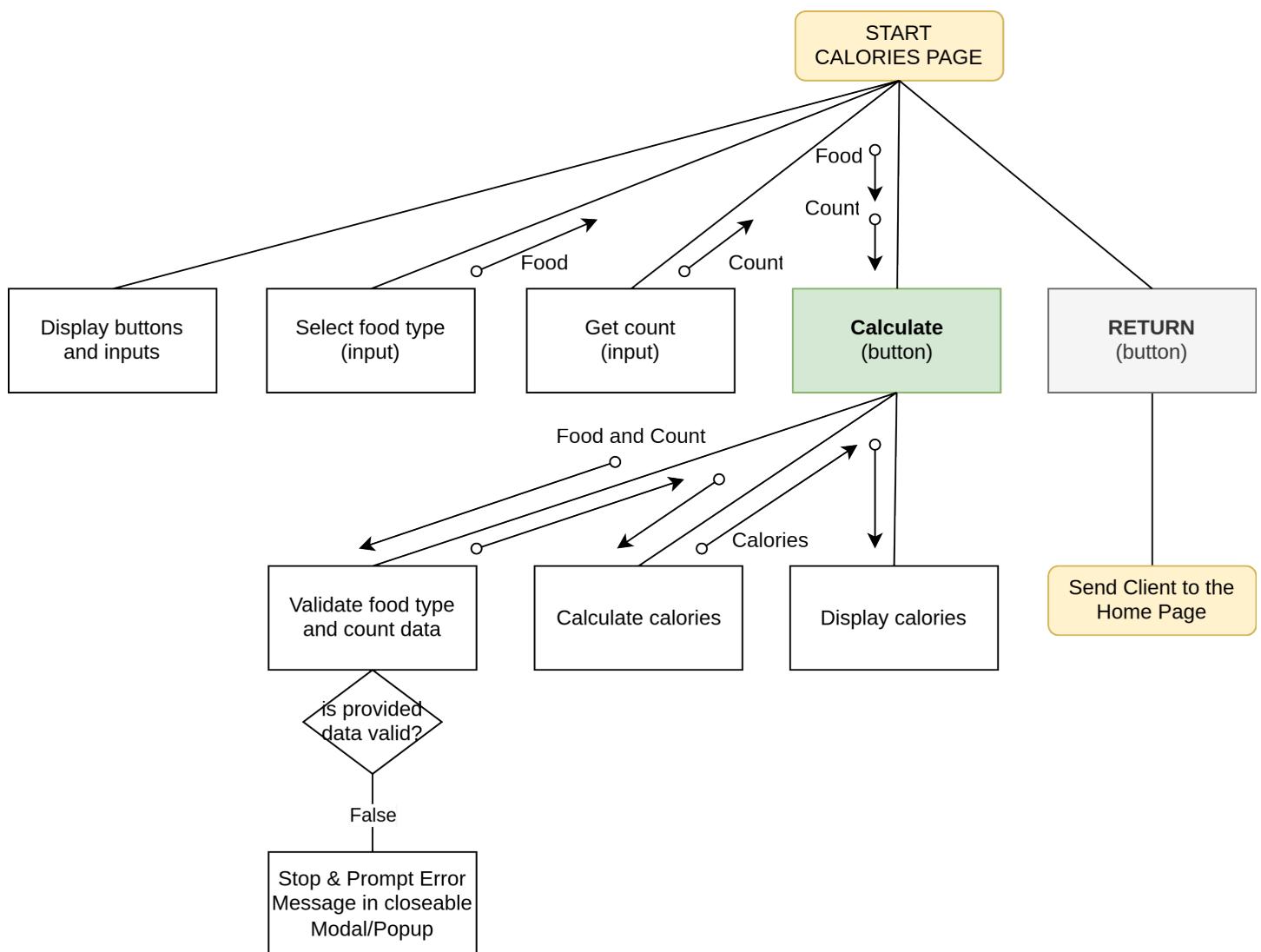
Design Document

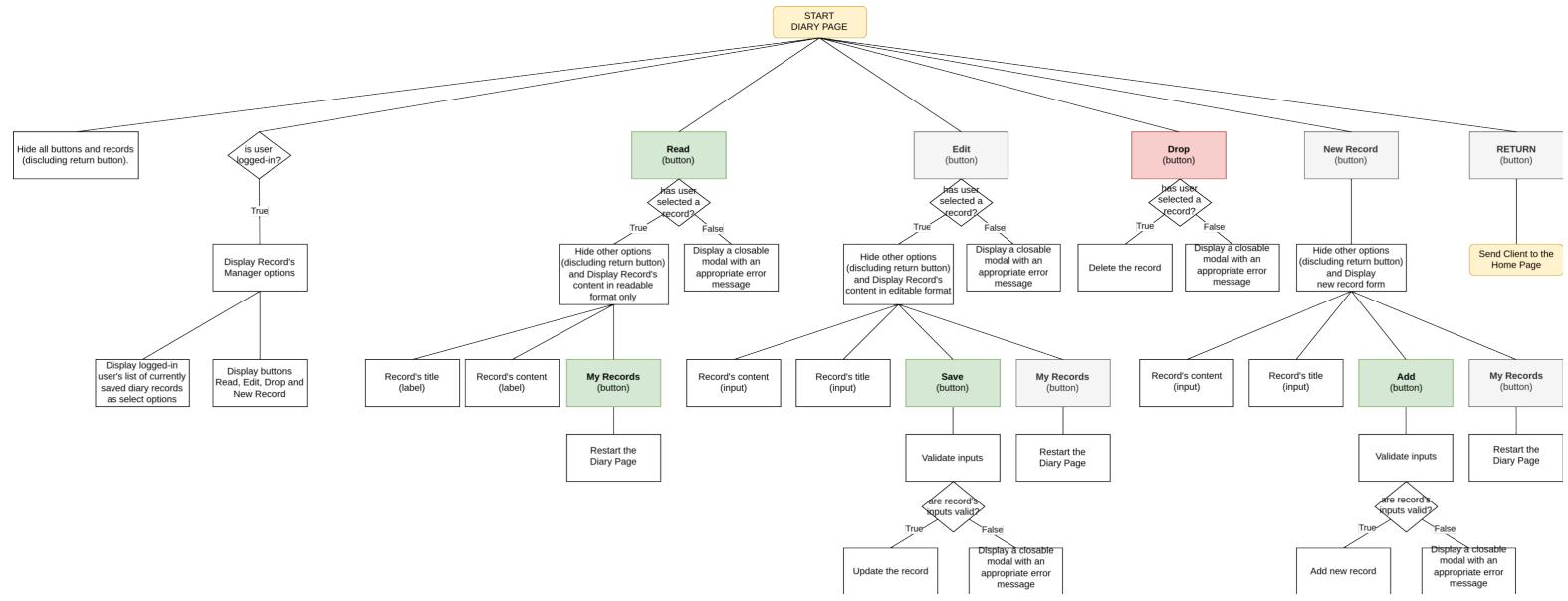
by Gabriel Ksiazek (2030089)

Structure Charts



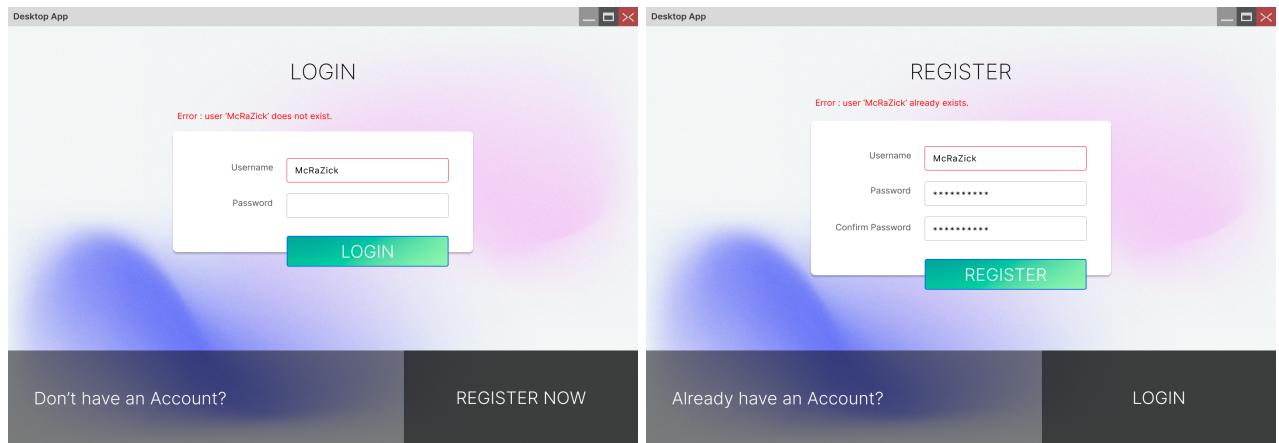






Wireframes

Login & Register page



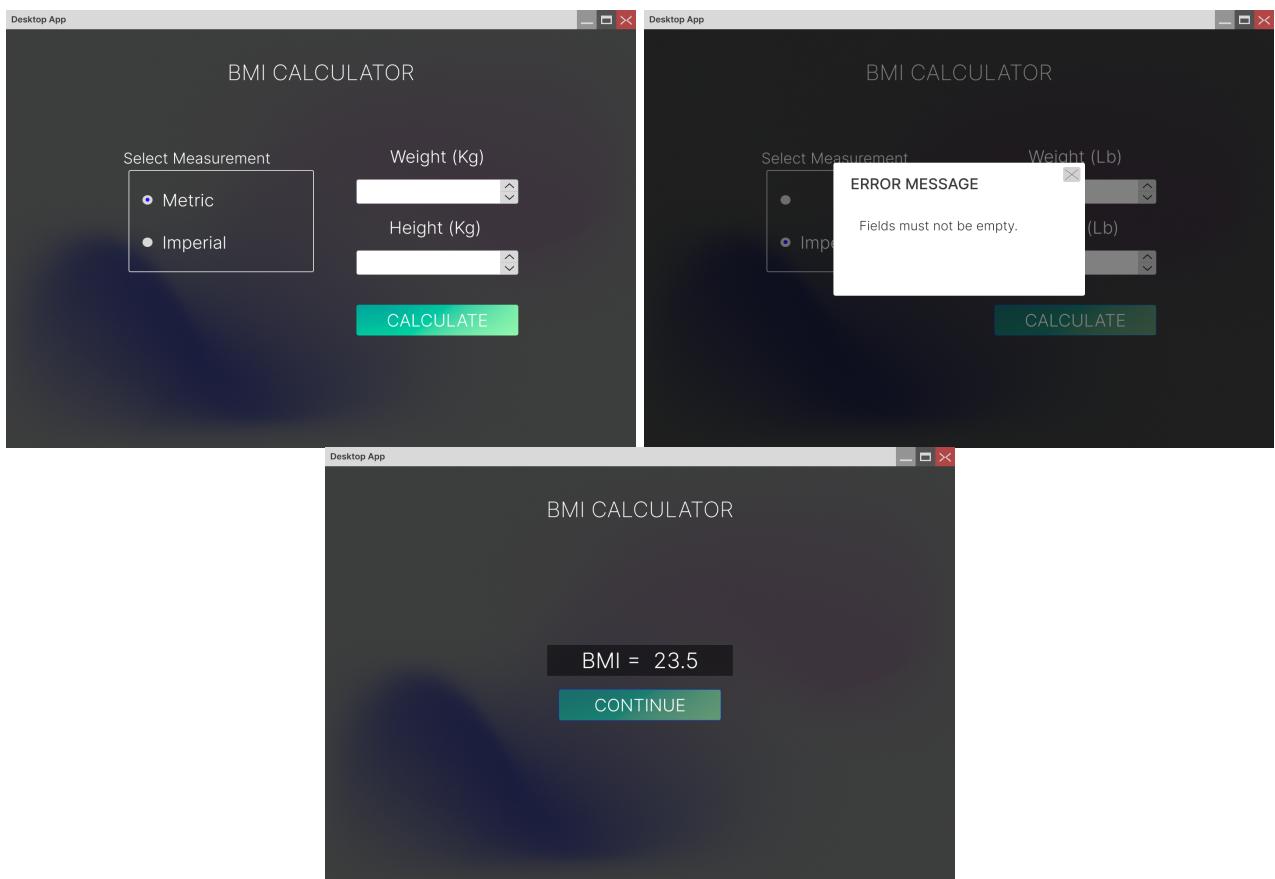
The wireframes show two side-by-side windows for a desktop application. Both windows have a title bar 'Desktop App' and standard window controls (minimize, maximize, close).

Left Window (Login): The title is 'LOGIN'. It contains a form with 'Username' (McRaZick) and 'Password' fields, and a 'LOGIN' button. Below the form is an error message: 'Error : user 'McRaZick' does not exist.' At the bottom are links 'Don't have an Account?' and 'REGISTER NOW'.

Right Window (Register): The title is 'REGISTER'. It contains a form with 'Username' (McRaZick), 'Password', and 'Confirm Password' fields, and a 'REGISTER' button. Below the form is an error message: 'Error : user 'McRaZick' already exists.' At the bottom are links 'Already have an Account?' and 'LOGIN'.

BMI page

- Users are automatically logged-in after successful registration, and then are redirected to the BMI page.
- Upon successful submission, the BMI is displayed with the 'continue button', which redirects user to their **home page**.



The wireframes show three states of a BMI calculator application:

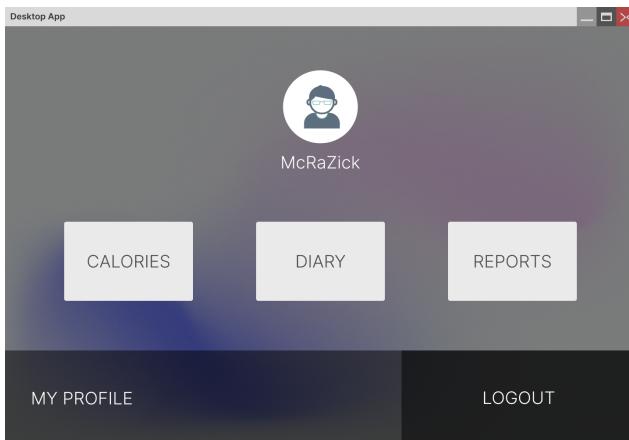
Initial State: The title is 'BMI CALCULATOR'. It has a 'Select Measurement' section with radio buttons for 'Metric' (selected) and 'Imperial'. Below it are 'Weight (Kg)' and 'Height (Kg)' input fields with up/down arrows, and a 'CALCULATE' button.

Error State: An 'ERROR MESSAGE' box appears, stating 'Fields must not be empty.' The 'Weight (Lb)' field is highlighted with a red border.

Result State: The title is 'BMI CALCULATOR'. It displays the result 'BMI = 23.5' and a 'CONTINUE' button.

Home page

- I decided to include this page to improve navigability.
- User can find navigation buttons to head to other pages.
- They can also logout from their profile.



Reports page

Note that the 'left-arrow' button is present on many different pages, it serves a function of returning the user back to the **home page**.

Two side-by-side screenshots of a desktop application window titled "Desktop App". Both screens show a "USER REPORTS" section with three data points: "Average weight of all users - 150 Kg", "Highest weight of all users - 300 Kg", and "Lowest weight of all users - 53 Kg".

The left screen has a "TO IMPERIAL" button in the top right corner and a left arrow button in the bottom left corner.

The right screen has a "TO METRIC" button in the top right corner and a left arrow button in the bottom left corner.

The data in the right screen is:
Average weight of all users - 330 Lb
Highest weight of all users - 661 Lb
Lowest weight of all users - 116 Lb

Calories page

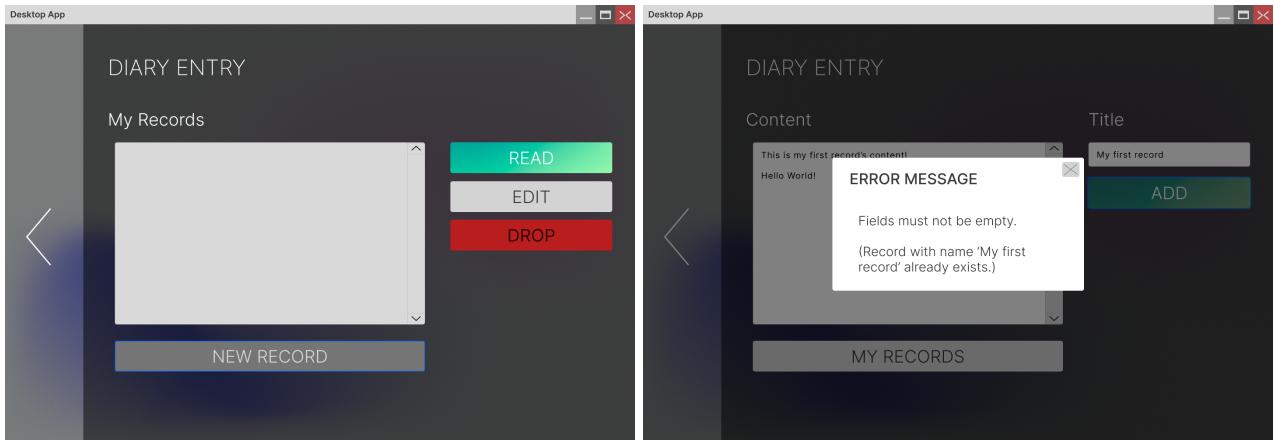
The screenshots illustrate the workflow of a calorie counter application:

- Screenshot 1:** Shows an "ERROR MESSAGE" dialog box with the text "Fields must not be empty." The "CALCULATE" button is visible.
- Screenshot 2:** Shows a dropdown menu with food items: Apple, Banana, Egg, Burger King, and Whopper. The "Count" input field shows "1".
- Screenshot 3:** Shows the result of the calculation: "4 x Apples (65 calories each) = 260 calories".

Diary page

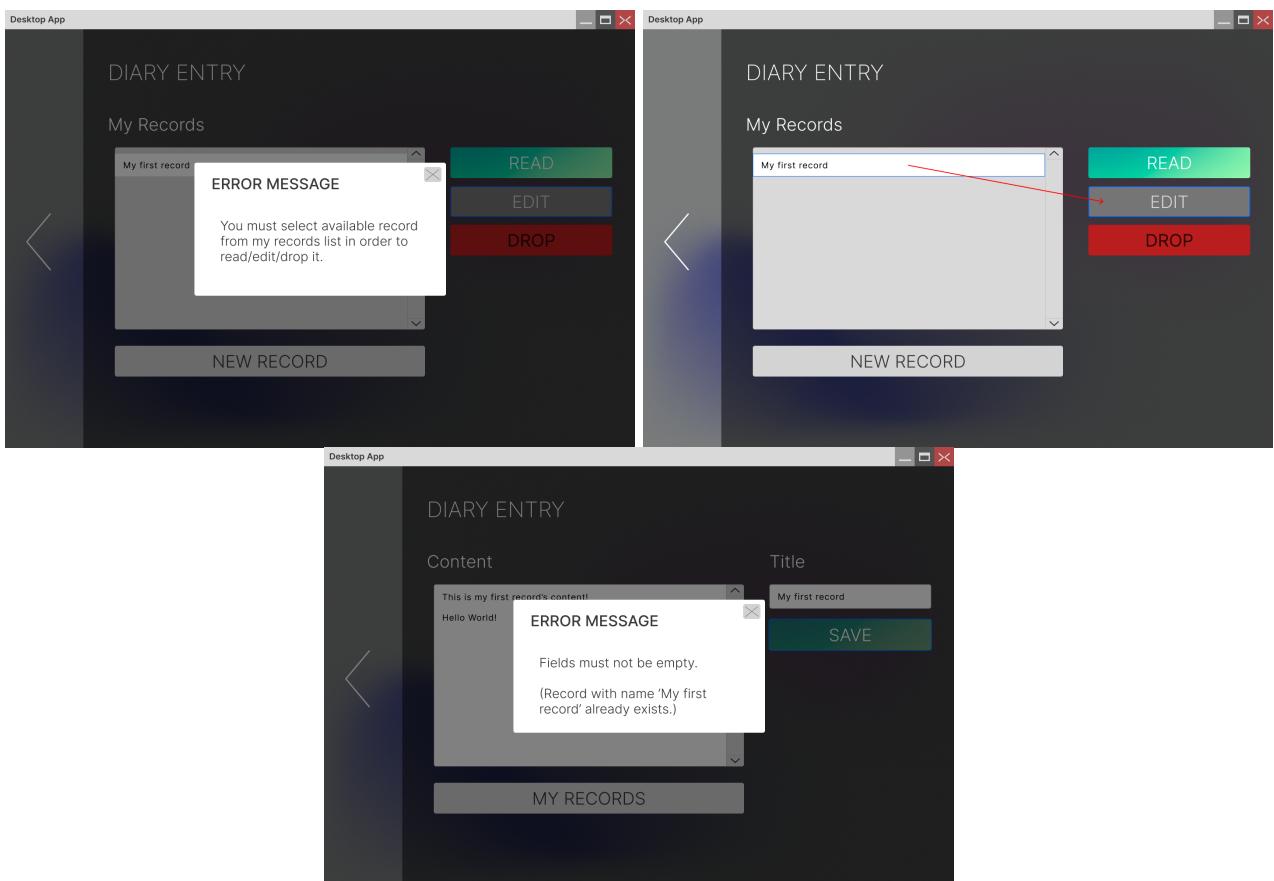
Add Diary Record:

(Press 'New record' button, fill-out the form, and press 'Add' button to create a record.)

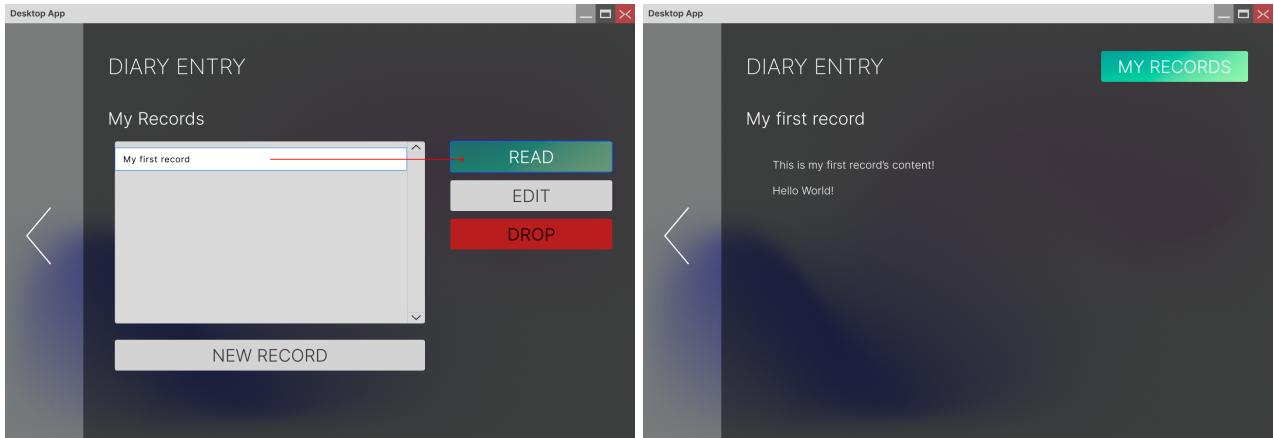


Edit Diary Record:

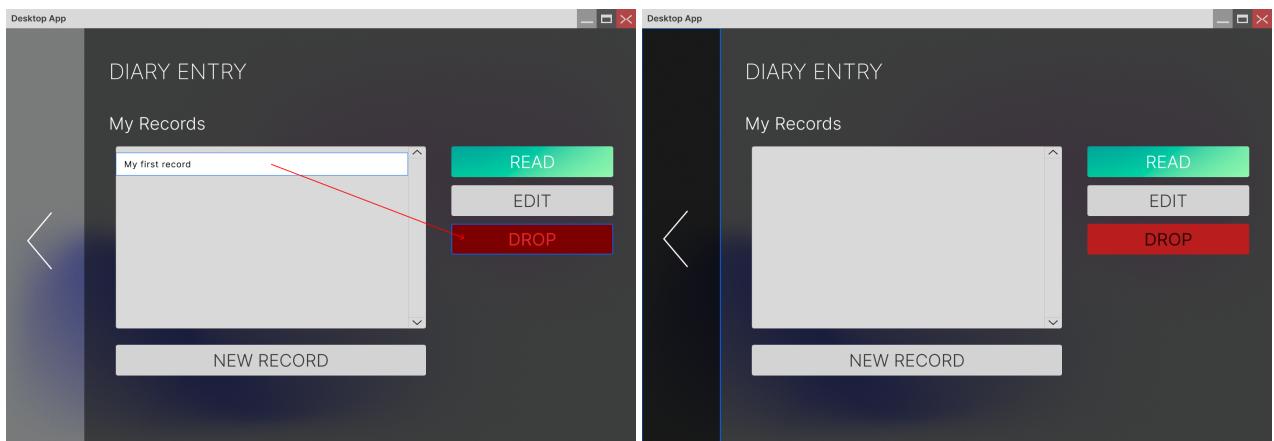
(Select a record from my records list, and press 'Edit' option at the right side.)



Read Diary Record:
(Select a record from my records list, and press 'Read' option at the right side.)



Drop Diary Record:
(Select a record from my records list, and press 'Drop' option at the right side.)



Pseudocode

```
### NOTE: "weight_list" is a list of weight's of all registered users

### START OF STANDARD_ALGORITHMS -----
-----

DEFINE FUNCTION get_average_weight(weight_list) AS {
    SET total TO 0
    SET count TO 0

    REPEAT FOR EACH weight FROM weight_list {
        SET total TO total + weight
        SET count TO count + 1
    }

    RETURN total / count
}

DEFINE FUNCTION get_highest_weight(weight_list) AS {
    SET highest_weight TO 0

    REPEAT FOR EACH weight FROM weight_list {
        IF weight IS_GREATER_THAN highest_weight THEN {
            SET highest_weight TO weight
        }
    }

    RETURN highest_weight
}

DEFINE FUNCTION get_lowest_weight(weight_list) AS {
    SET lowest_weight TO get_highest_weight(weight_list)

    REPEAT FOR EACH weight FROM weight_list {
        IF weight IS_LOWER_THAN lowest_weight THEN {
            SET lowest_weight TO weight
        }
    }

    RETURN lowest_weight
}

### END OF STANDARD_ALGORITHMS -----
```

```
### START OF CONVERSION_ALGORITHMS -----  
  
DEFINE FUNCTION kg_to_lb(kg) AS {  
    SEND ERROR IF kg IS_NOT_A_NUMBER  
    RETURN kg MULTIPLY_BY 2.20462262  
}  
  
DEFINE FUNCTION lb_to_kg(lb) AS {  
    SEND ERROR IF lb IS_NOT_A_NUMBER  
    RETURN lb DIVIDE_BY 2.20462262  
}  
  
DEFINE FUNCTION lb_weight_list_to_kg(weight_list) AS {  
    SET new_weight_list TO EMPTY_LIST  
  
    REPEAT FOR EACH weight FROM weight_list {  
        ADD ITEM lb_to_kg(weight) TO new_weight_list  
    }  
  
    RETURN new_weight_list  
}  
  
DEFINE FUNCTION kg_weight_list_to_lb(weight_list) AS {  
    SET new_weight_list TO EMPTY_LIST  
  
    REPEAT FOR EACH weight FROM weight_list {  
        ADD ITEM kg_to_lb(weight) TO new_weight_list  
    }  
  
    RETURN new_weight_list  
}  
  
### END OF CONVERSION_ALGORITHMS -----
```