BALANCE

Supervised by:Prof. Amjad Hudaib

Advanced Software Engineering

	Student Name	Student ID	No.
1	Leena Abu-Zakham	0224570	29
2	Eman Tegoan	0223779	27
3	Sara Al-Khrissat	0224675	31
4	Rama Nedal	0225141	32

Second Semester 2024/2025

Table of Contents

Table of Figures

Table of Tables

1.0 Chapter One

1.1 Project Overview

Balance is a website that provides personalized diet suggestions based on the nutritional needs of users. By providing food rich in vitamins and minerals for optimal health. The website also offers recipe ideas that include the recommended nutrients through integrated web search functionality.

1.2 Main Objectives

1. Developing an interactive interface for recipes and health tracking.

2. Helping housewives, the elderly, or those living in remote areas far from medical centers to have a healthy lifestyle and balance their nutrients.

3. Provide trusted and studied recommendations on food that meet nutritional and medical guidelines.

1.3 Main Motivations

The main motive for developing this website is to provide individuals with primary health checks and customized recipes online, limiting the costs and time required for consulting a nutritionist or the risk in relying on unreliable websites that fail to consider health conditions that may conflict with the recommended diets resulting in worsening the existing problems or creating a new one. Stakeholders include people who are motivated to improve and track their health, but face difficulties in accessing specialists, such as housewives, the elderly, or those living in remote areas far from medical centers.

1.4 Constraints

1. High costs of some products (food) that are needed for the diet.

2. Users lack commitment caused by unhealthy food attachment.

3. This website should be delivered within approximately 3 months.

4. The budget shouldn’t exceed 2000$.

1.5 Similar Projects

1) **Cronometer :**This app is a powerful health and fitness app that makes it easy to track your diet and exercise habits. You can easily log your meals and get a detailed breakdown of the calories, vitamins & minerals you're consuming, and you can easily enter your dietary preferences to receive personalized meal recommendations.

https://cronometer.com/

2)**Lifesum:** This app is more than a calorie counter, Lifesum helps you adopt nutritious diets that fit your lifestyle and taste. Achieve your weight loss goals while building healthy eating habits for life. Of its features: Macro tracker (protein, carbs, and fats) and food rating and Life Score test for personalized nutrition recommendations,etc.

https://lifesum.com/

3)**My Tummy**: this app encourages Intuitive eating and it is about creating a safe space to explore your relationship with food .It offers a food tracker, digestion timer, daily journal, and food tags. Instead of caloric metrics, the app guides you to tune into your hunger cues without labeling foods as “good” versus “bad.”

https://apps.apple.com/us/app/mytummy-intuitive-eating/id1560043349

1.6 Project Organization

The rest of the document is organized as follows:

Chapter two presents the feasibility study, the needed hardware and software, the schedule and estimated time, and the pert diagram.

Chapter three presents the stakeholders influenced by the system and the requirements of the software, and the architectural design.

Chapter four presents the project design, class, use case, and sequence diagrams.

Chapter five includes the implementation of the system.

Chapter six presents the project testing methodologies and test cases.

2.0 Chapter Two

2.1 Feasibility study

2.1.1 Technical Feasibility

Creating a website: Creating a healthcare website requires skilled front-end developers proficient in HTML, CSS, and JavaScript to ensure seamless and user-friendly experience. Each section of the website will be designed with clarity and simplicity to enhance usability.

A strong database management system is essential for efficiently handling various types of data, such as nutritional information and user-submitted dietary inputs, ensuring the software's optimal performance. The database structure must be designed to accommodate the growing volume of nutritional data and user-generated inputs.

Operational Speed: Each operation must be completed within a timeframe of less than 30 seconds to positively impact the website's efficiency.

2.1.2 Operational Feasibility

Some operational problems that the project may face:

Lack of users: Since this project targets elderly individuals and housewives as its main users, it may face challenges in attracting and retaining them. Effective advertising will be essential to create a positive impression of the application and increase the user base.

Limited Technological Knowledge Among Users: The targeted users might struggle with technology due to their age or lack of interest in this field, making it difficult for them to navigate the website. To address this issue, enhancing the simplicity and clarity of the interface will improve the user experience and ensure usability, preventing the loss of potential users.

Performance problems: Slow-loading pages can lead to user frustration and disengagement. To mitigate this, it is important to optimize website speed and overall performance, ensuring users remain engaged with the platform.

2.1.3 Legal Feasibility

The project confirms the legal and ethical requirements. It contains the privacy policy terms. There are some points that need to be clarified: the developers own this project, and any newly added feature will be in the next version.

2.2 The Needed Hardware and Software

Table 1: needed hardware and software

	Description	Quantity	Vindy Price	Total	Note
HW1	HP All-in-One	1 for each user	499	499	
HW2	Smartphone	1 for each user	200	200	needed if HW1 is not available
SW1	Windows 11	1 for each user	10	10	with HW1
SW2	Android	1 for each user	0	0	
HW3	TP-Link WiFi Router	1 can be used by multiple users	20	20	
Total			529		If Laptop device
			220		if Smartphone device

2.3 *The Schedule and The Estimated Time*

2.3.1 Schedule

Table 2:The Project Schedule

Task ID	Description	Duration Time (WD)	Dependence	Responsible partner's staff	Resources
T1	Data gathering & looking for relative resources	4 Days	None	All	Google
T2	Prepare a project management plan	7 Days	T1	Sarah	Software Engineering by Ian Sommerville book
T3	The feasibility study	5 Days	T1&T2(M1)	Eman, Leena, Rama	

Task ID	Description	Duration Time (WD)	Dependence	Responsible partner's, staff	Resources
T4	Questioner	2 Days	None	Sarah	Google forms
T5	Document software requirements (functional, non-functional and domain)	6 Days	T3&T4	All	
T6	Design GUI	7 Days	T5(M2)	Leena, Eman	Sketches
T7	Design software architecture	8 Days	T5	Sarah	
T8	Create the System Model (use case…)	9 Days	T5&T6	All	
T9	GUI implements	10 Days	T6&T7(M3)	Leena, Eman, Rama	Visual studio 2019
T10	Database implementation	5 Days	T9	Leena	SQL Server management studio (SSMS)
T11	Fix issues in the released patches	6 Days	T9	Rama	
T12	Test the System	7 Days	T9	Rama	

2.3.2 Pert Diagram

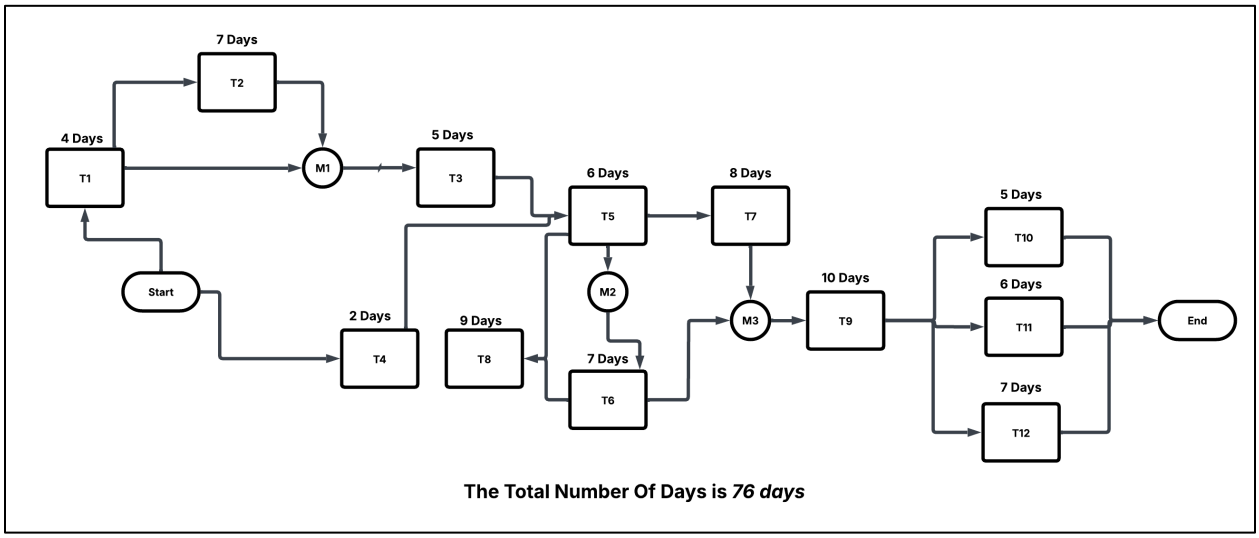


Figure 1: The Pert Diagram

3.0 Chapter Three

3.1 System Stakeholders

Table 3:The Project Stakeholders

Stakeholders ID	Stakeholders	Note
ST1	Users: people who want to customize diets online and cannot access them easily due to living in remote places or financial problems, as well as busy housewives and elders.	
ST2	Designers: are responsible for UI design and ensuring it meets the usability and user experience goals.	
ST3	Developers: The ones responsible for creating the website.	
ST4	Data Team: the ones who will create the database to store users' information.	
ST5	Regulatory Authorities: who are responsible for privacy laws and nutritional/medical regulations.	
ST6	Contact Team: people responsible for interacting with users and answering their questions.	

3.2 Data Gathering Techniques

3.2.1 Survey

Questionnaire: serves as a research tool comprising a series of inquiries or prompts designed to gather information from a respondent.

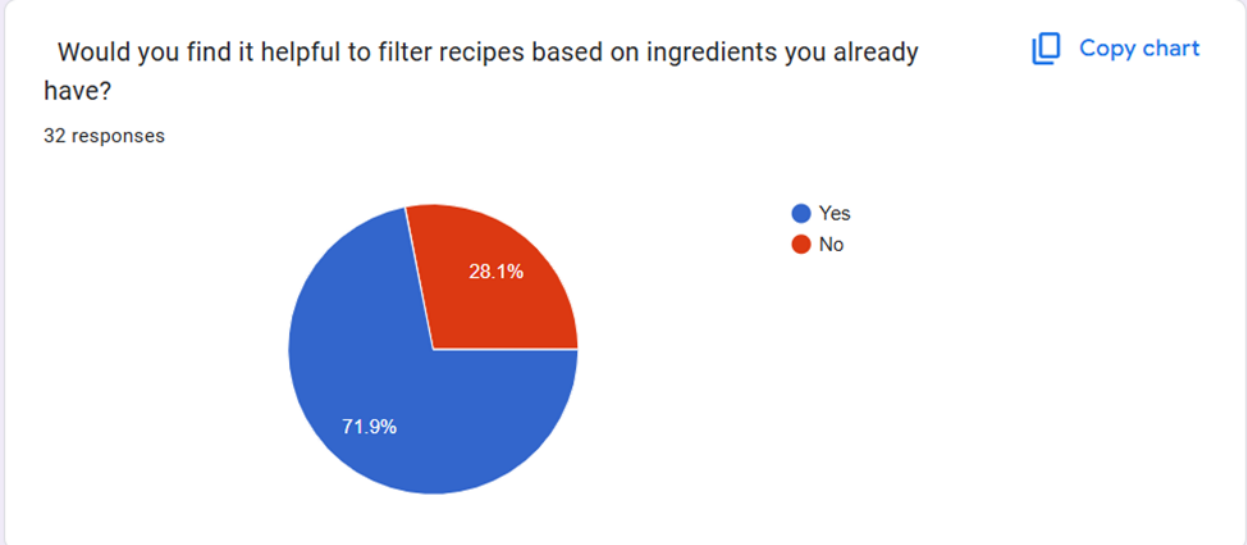


Figure 2:The Survey First Question Analyze

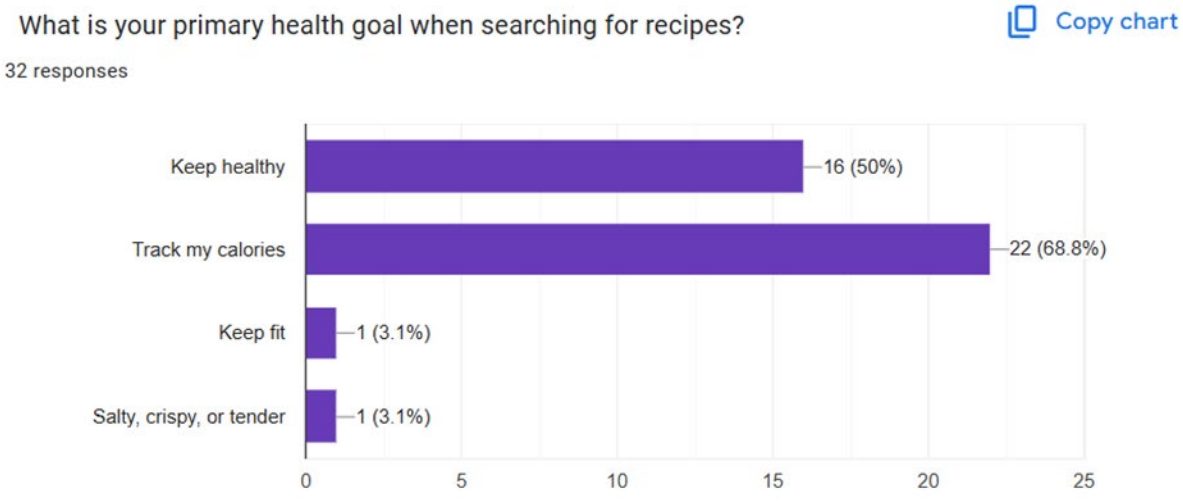


Figure 3:The Survey Second Question Analyze

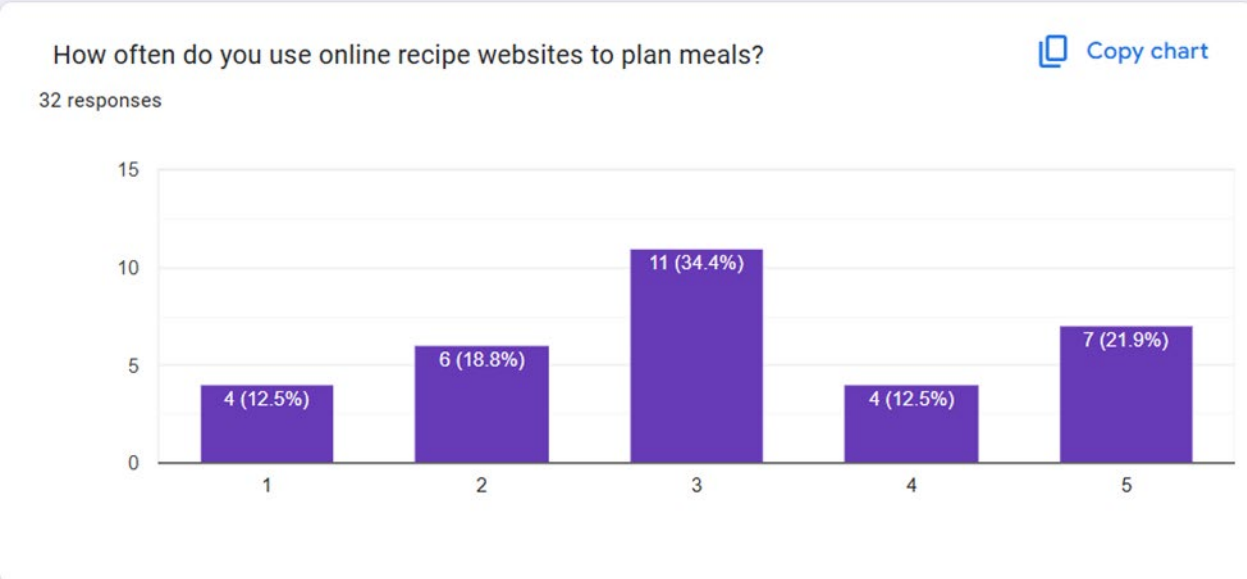


Figure 4:The Survey Third Question Analyze

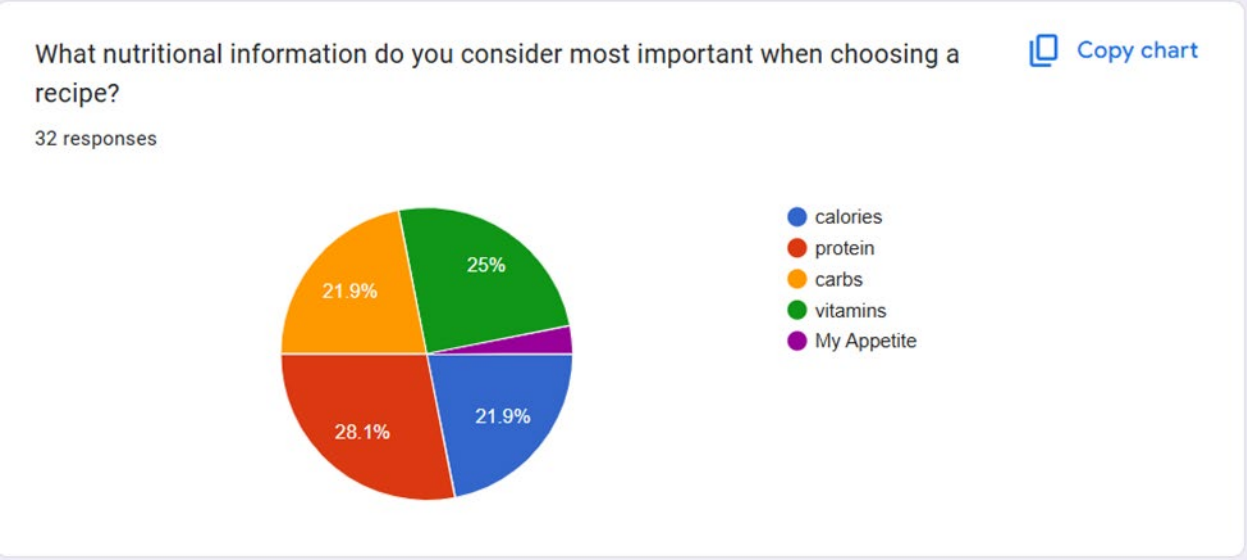


Figure 5:The Survey Fourth Question Analyze

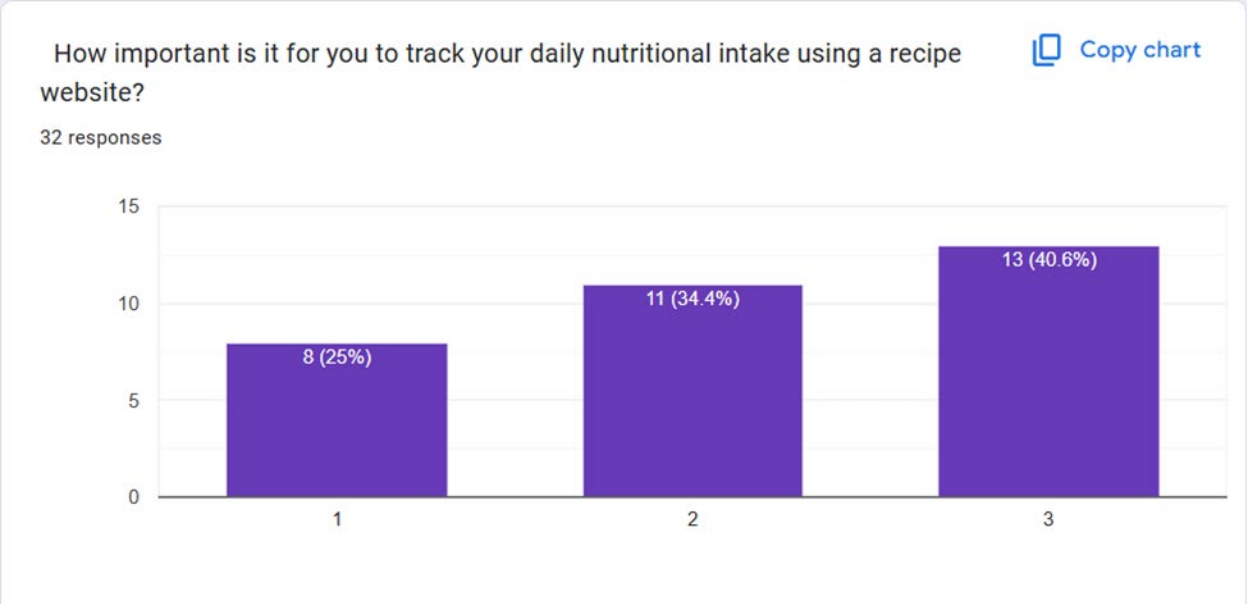


Figure 6:The Survey Fifth Question Analyze

3.3 Functional Requirements

Table 4:The Project Functional Requirements

Functional Requirement ID	Description	ST	Priority	Increment	Note
FR1	The first thing that appears to use is login page it includes the login function and a register link to go to the register page	ST1,ST2,ST3	5	1	
FR2	The login page is user friendly, where the user can easily enter their username and password	ST1,ST2,ST3	5	1	
FR3	The new passwords are added to the database	ST2,ST3,ST4,ST5	4	2	
FR4	The login button on the login page sends the information entered by the user to be checked in the database.	ST2,ST3,ST4	5	1	
FR5	A message indicating a failed login due to an error in username or password will appear in red.	ST2,ST3,ST4	4	2	
FR6	When clicking on the register link, the register function displays the registration page. Registration page is user-friendly so the users can easily write their username and password and confirm it.	ST1,ST2,ST3	5	1	
FR7	The register function checks that both password entries match and that the username has not appeared before in the database.	ST2,ST3,ST4	4	2	
FR8	The register function checks that both password and username are not sent empty, or else an error message occurs.	ST2,ST3	4	2	
FR9	When clicking on the register button with correct data, the user's information is sent to the database and a message indicating a successful registration appears.	ST1,ST2,ST3,ST4	3	2	
FR10	When clicking on the register button with incorrect data like a taken username, an error massage appears.	ST1,ST2,ST3,ST4	3	2	

Functional Requirement ID	Description	ST	Priority	Increment	Note
FR11	When logging in the root function displays the main page in the website the my Data page. if it was the first time logging in the main form in the page will require the user to enter some information to be used after.	ST1,ST2,ST3,ST4	5	1	
FR12	The data Save button sends user-entered data to the database to store and check it.	ST2, ST3, ST4	4	1	
FR13	The user can add values to the nutrition list of specified elements including Ferritin, B12, Omega3, Zinc, Calcium And magnesium.	ST1,ST2,ST3,ST4	4	1	
FR14	Enter the current weight and goal weight function for users to fill in their weight at the moment and the goal weight they are aiming for.	ST1,ST2,ST3,ST4	5	1	
FR15	Entering height function which will be used in the calculation of BMI.	ST1,ST2,ST3,ST4	2	2	
FR16	Filling the nutrition will provide a color code and a message in the label nutrition state, it will appear green if the elements are at normal levels and red if levels are extremely low.	ST2,ST3	2	2	
FR17	Progress percentage circle showing how many days they have been following the diet based on their clicking follow days button and it will be reset after a month.	ST1,ST2,ST3,ST4	3	2	
FR18	Entering gender and age functions.	ST1,ST2,ST3,ST4	2	2	
FR19	A message representing the weight progress is indicated by the colors green if the goal is in the goal range or yellow if the aim is still not reached with the kgs needed to reach the aim.	ST2,ST3	2	2	
FR20	BMI (body mass index) value will appear after calculating it using height and current weight to provide users with an estimation of the amount of body fat.	ST2,ST3,ST4	2	3	
FR21	If the BMI value is normal it will appear in green with a message that says “normal” or else in red with a message either “underweight” or “overweight”.	ST2,ST3,ST4	2	3	

Functional Requirement ID	Description	ST	Priority	Increment	Note
FR22	The diet recipes will include approximate nutritional breakdowns	ST2,ST3	3	2	
FR23	Users can browse or search for healthy recipes that align with their nutrition goals.	ST1,ST2,ST3	4	1	
FR24	The recipes page will include a list of ingredients, preparation instructions, and a breakdown of nutrients.	ST2,ST3	4	1	
FR25	The recipes page will include a function to choose based on the nutrition a user wants.	ST2,ST3	4	1	
FR26	The food page will include a list of vegetables and fruits breakdown of nutrients.	ST2,ST3	4	1	
FR27	Food page that will provide different food types that have the nutrients user's need as they can filter it based on nutrients.	ST2,ST3	4	2	
FR28	Logout element to allow users to log out from the website.	ST1.ST2,ST3	5	1	
FR29	The system will provide a way for users to get answers or complain about problems they face by filling in the name, email, and message fields.	ST1,ST2,ST3,ST5	4	2	
FR30	Send message button which will send a user message to the contact team.	ST3,ST5	4	3	
FR31	The system will provide a way for users to reach contact team directly by clicking on their number or email.	ST2,ST3,ST5	4	3	

3.4 Non-Functional Requirements

Table 5:The Project Non-Functional Requirements

Non-Functional Requirements ID	Description	Priority	Note
NFR1	Security: The system ensures robust security through user authentication and authorization, requiring a username, password validation and session control.	5	
NFR2	Ease of Use: Our system features an intuitive, user-friendly interface, making it accessible and easy to navigate for a wide range of users.	4	
NFR3	Ease of learn: Our system does not have advanced features and functionalities which may delay using and performance.	5	

3.5 Domain Requirements

Table 6:The Project Domain Requirements

Domain Requirements ID	Description	Note
DR1	Availability: The system must provide monthly information on the updated information (new nutrient values, new weight).	
DR2	Accessibility: The user can access the system from any device (smartphone, laptop, etc.) at any time.	
DR3	Internet Connection: To fully access the system, you need to be connected to the internet.	

3.6 Requirements Analysis and the Architecture Design Components:

C1: **Registration**: {FR1,FR6,FR7,FR8,FR9,FR10}

C2: **Login**: {FR1,FR2,FR4,FR5}

C3: **Logout**:{FR28}

C4: **Validation**:{FR4,FR5,FR7,FR8,FR10,FR12,FR13,FR16,FR17,FR19,FR21,FR28}

C5: **Storing Information** :{FR3,FR9,FR12,FR13,FR14,FR15,FR17,FR18}

C6: **Contact**: {FR29,FR30,FR31}

C7: **Personal Data** :{FR11,FR12,FR14,FR15,FR18,FR19,FR20,FR21}

C8: **Nutritional Data** :{FR11,FR12,FR13,FR16}

C9:**Progress**:{FR11,FR12,FR17}

C10:**Filtring** :{FR22,FR23,FR24,FR25,FR26,FR27}

Groups:

G1(Registration)

{C1,C2,C3}

G2(Database)

{C4,C5}

G3(User Data)

{C7,C8,C9,C10}

G4(Contact)

{C6}

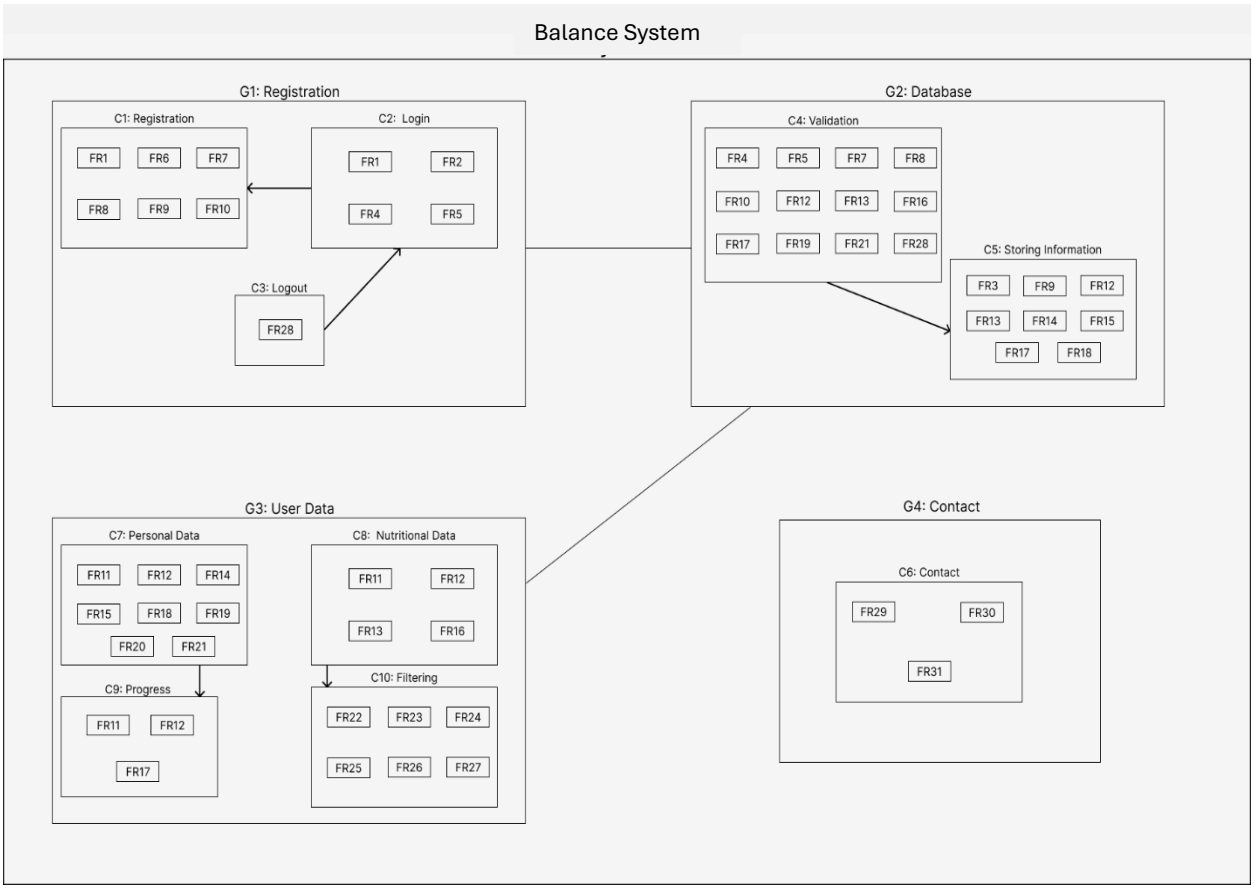


Figure 7:The project Architectural Design

4.0 Chapter Four

4.1 Class Diagram

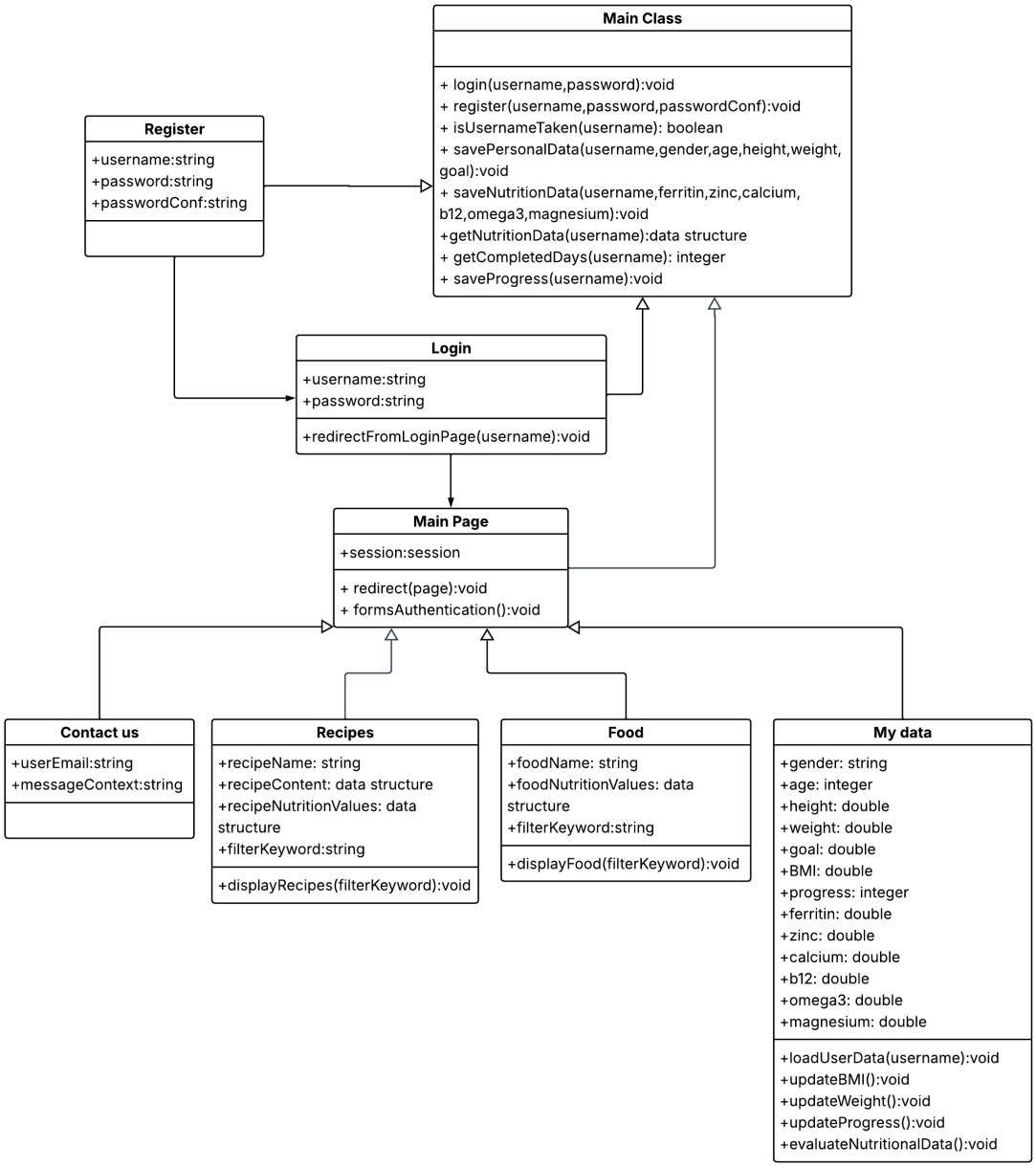


Figure 8: The Project Class Diagram

4.2 Use-Case Diagram

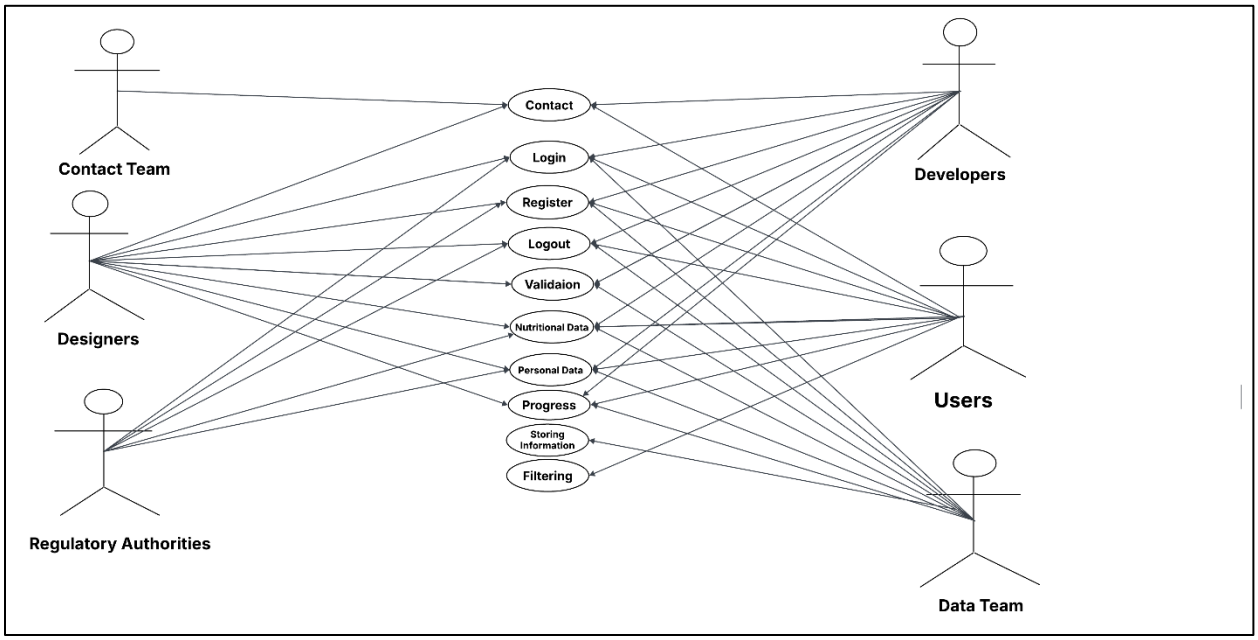


Figure 9:The Full Use Case Diagram

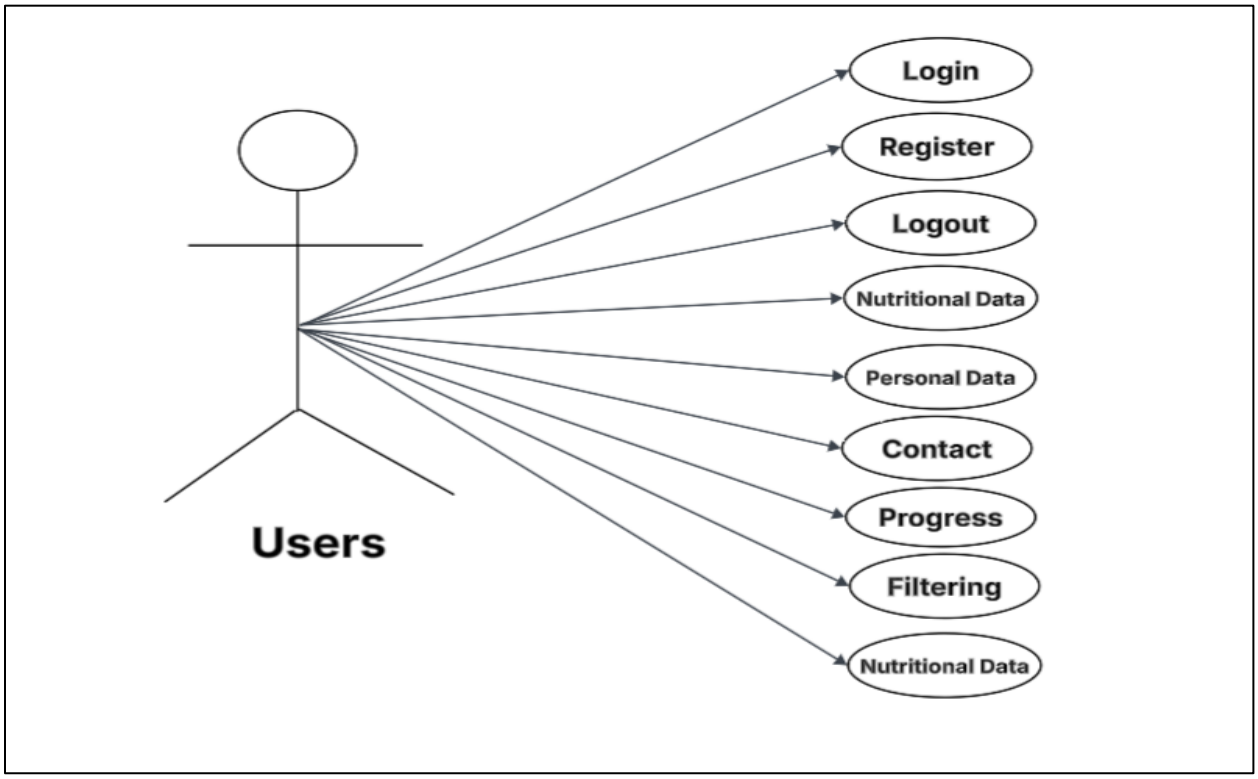


Figure 10:The Users Use Case Diagram

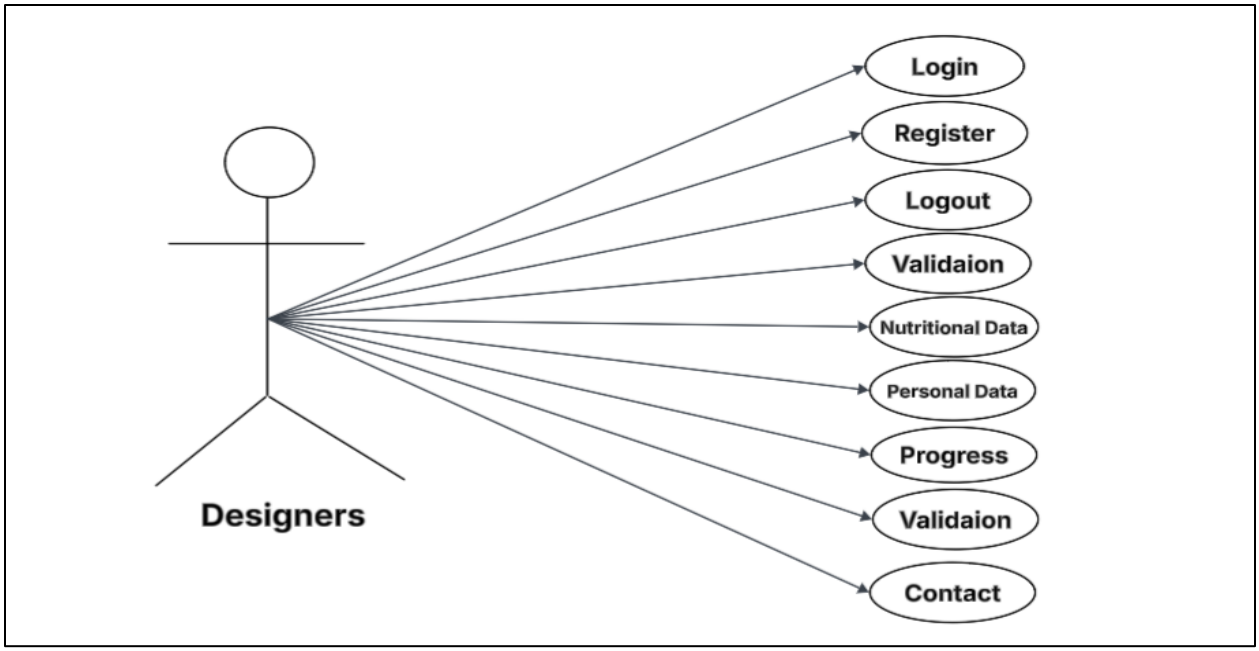


Figure 11:The Designers Use Case Diagram

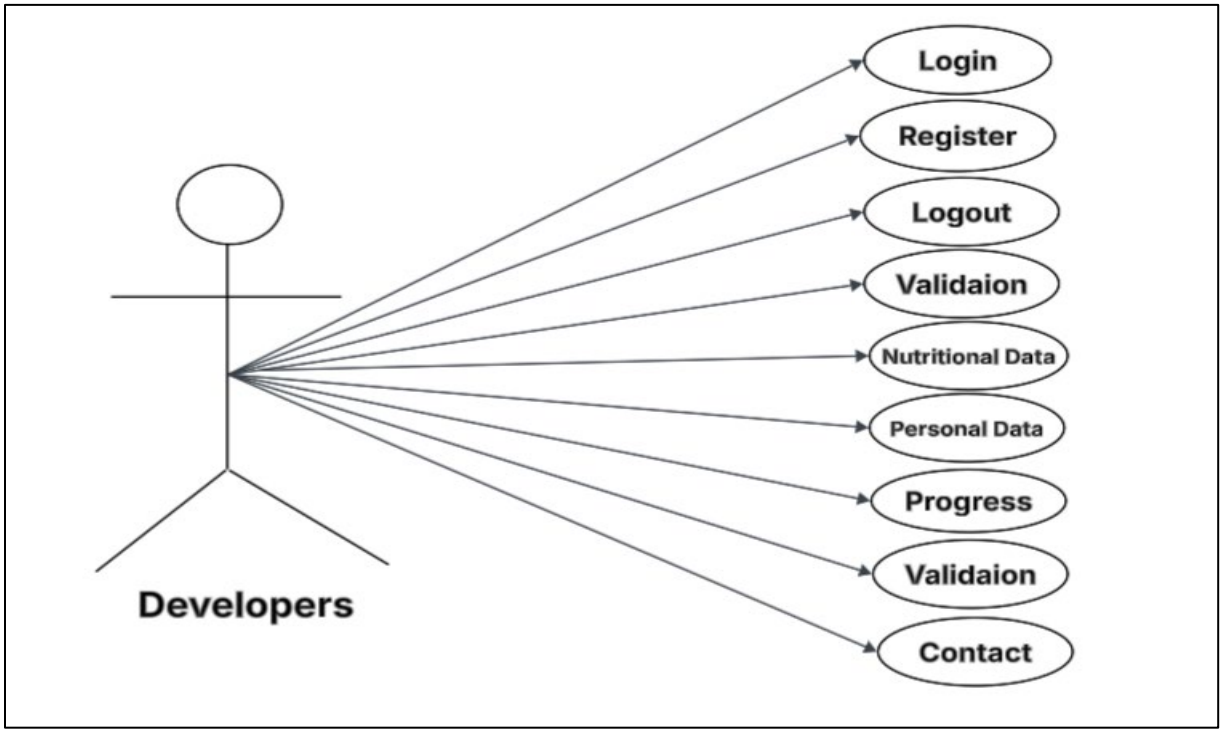


Figure 12:The Developers Use Case Diagram

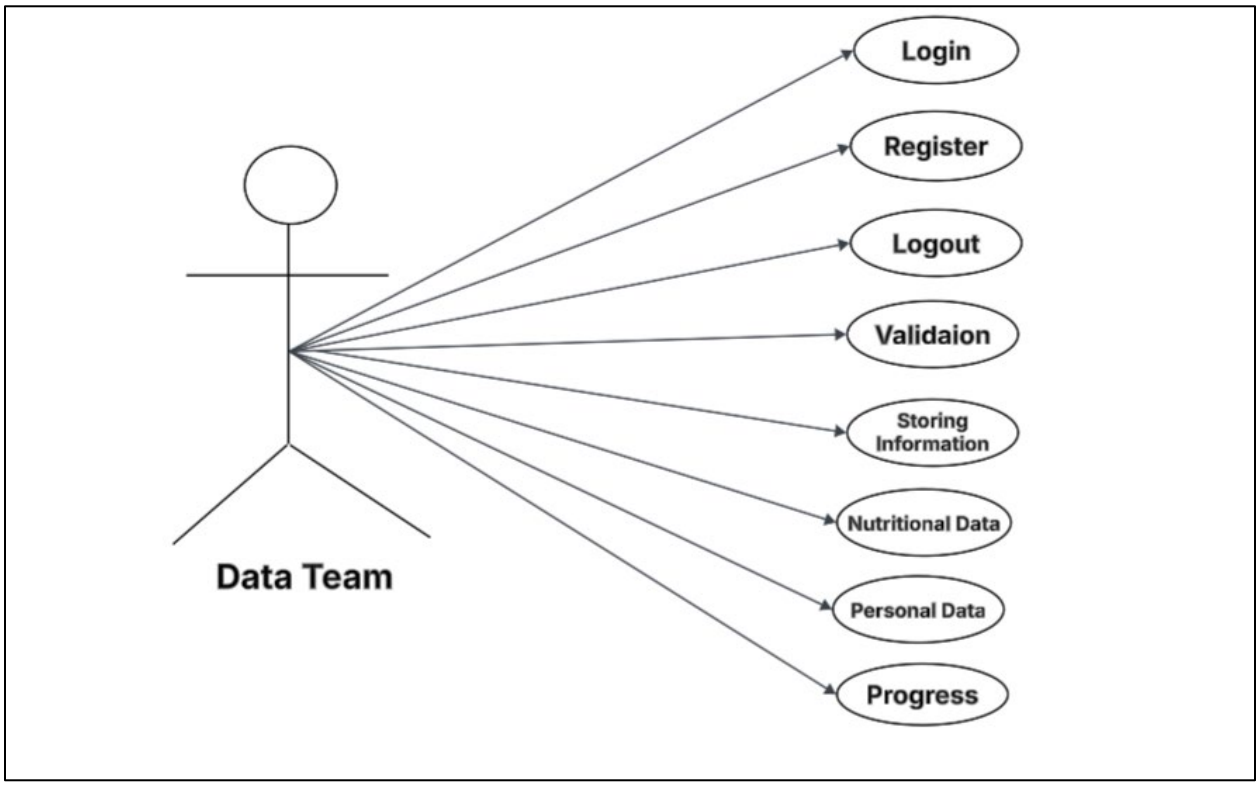


Figure 13:The Data Team Use Case Diagram

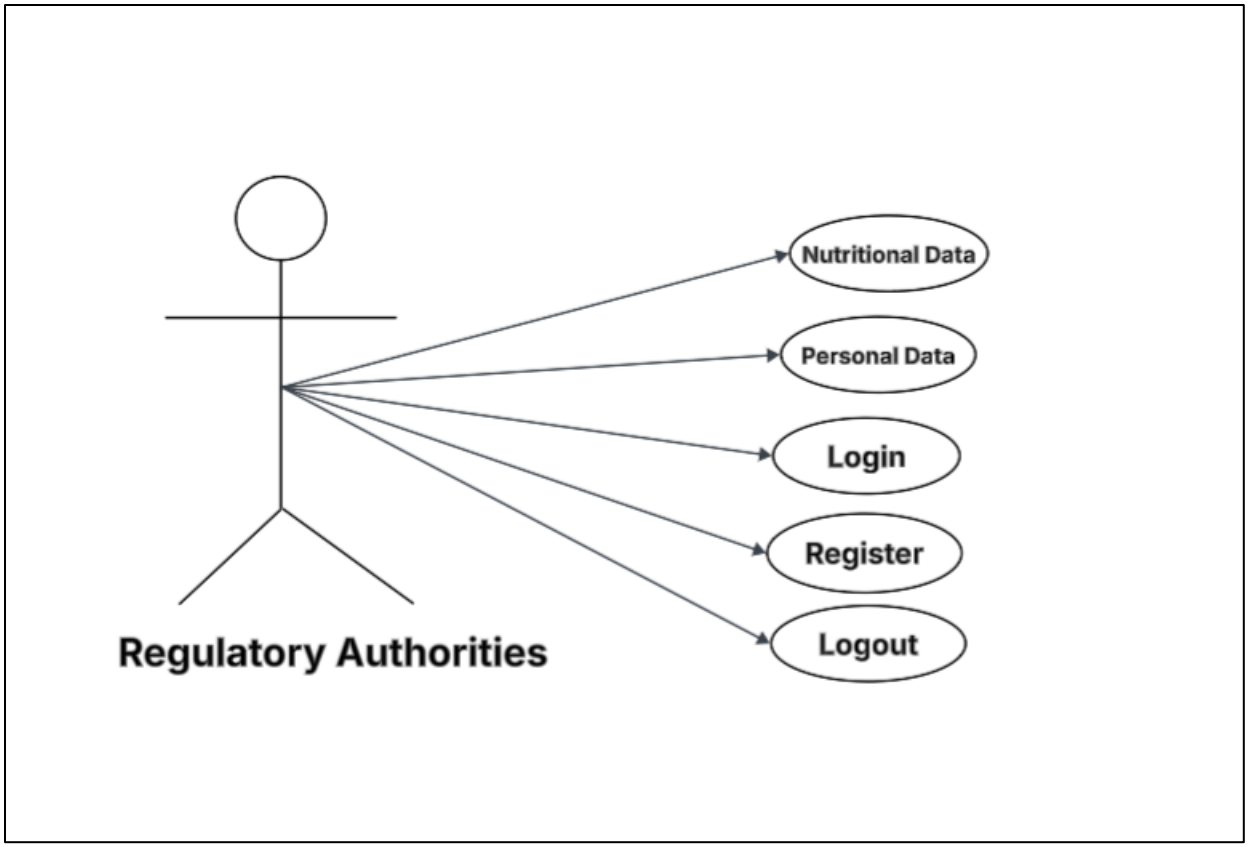


Figure 14:The Regulatory Authorities Use Case Diagram

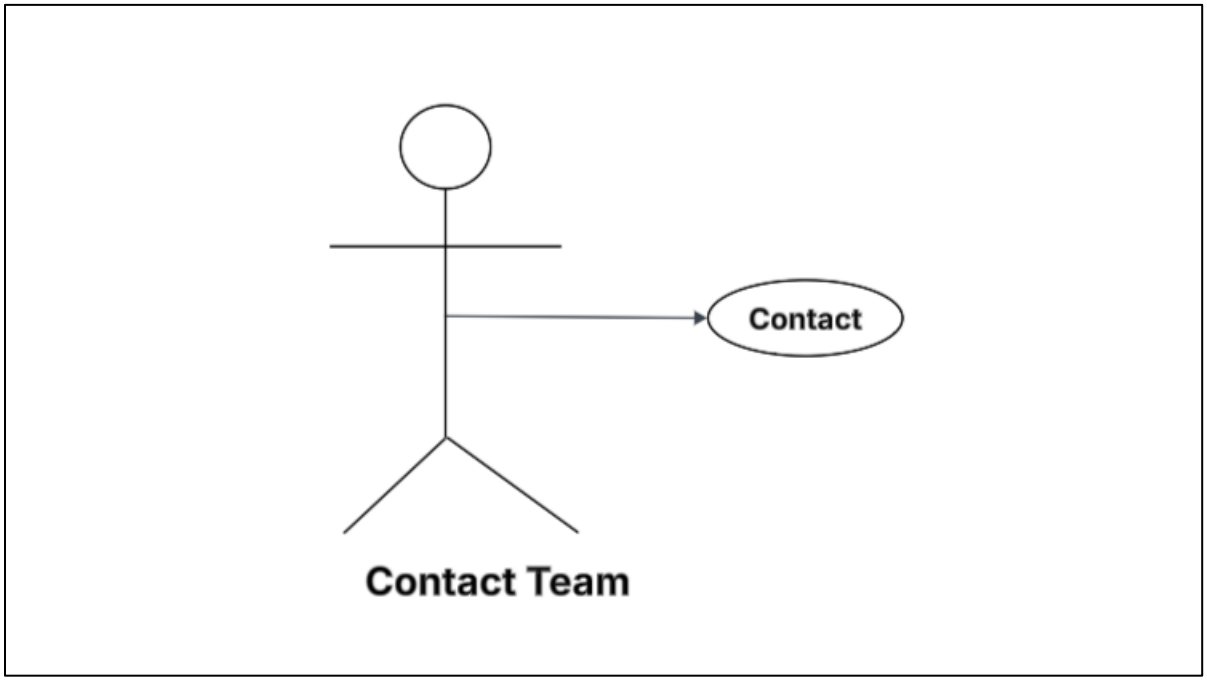


Figure 15:The Contact Team Use Case Diagram

4.2.1 Use-Case Description and Sequence Diagrams

Table 7:The Login Use-Case Description

Login	
Actors	Users, Designers, Developers, Data team, and Regulatory Authorities.
Description	User enters his/her username and password, then he clicks on **"Login"** button to go to the home page if he/she doesn't have an account, he must click on **"register here"** to create his account.
Data	Username and password.
Stimulus	The user fills in the email and password fields on the Login page, then clicks the **"Login"** button. If the credentials are correct, the login proceeds. If the credentials are incorrect, a red error message appears: *"Wrong Username Or Password."*
Response	On correct input: The system matches the entered data with the database and displays the home page. On incorrect input: The system displays a red error message, and the user remains on the login page.
Components	C2

*C2 **satisfies** these functional requirements: FR1, FR2, FR4, FR5.

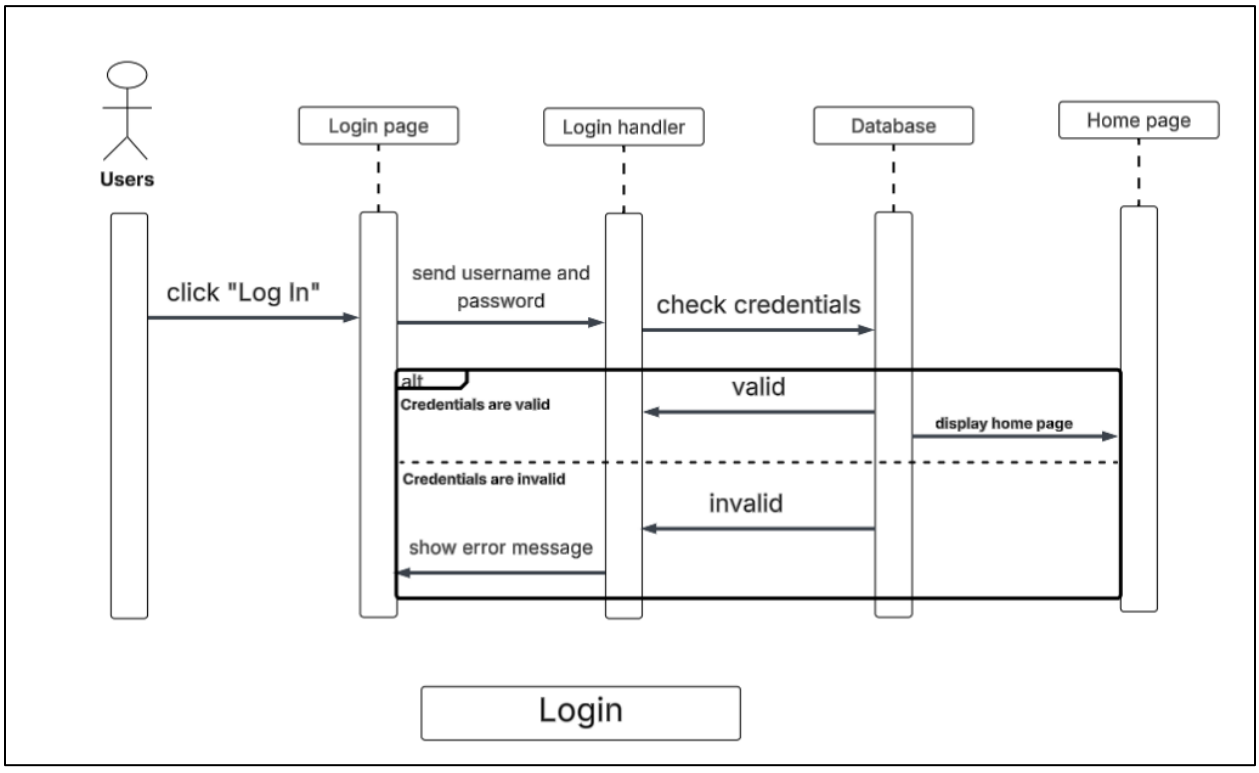


Figure 16:The login Sequence Diagram

Table 8: The Logout Use-Case Description

Logout	
Actors	Users, Designers, Developers, Data team, and Regulatory Authorities.
Description	The logout process is an action that terminates the user's active session with the web application, ensuring that no unauthorized access can occur after the user leaves the system.
Data	No entered data.
Stimulus	The user clicks the **"Log Out"** button.
Response	The system ends the user's session by invalidating the session token and redirects the user to the Login page.
Components	C3

*C3 **satisfies** these functional requirements: FR28.

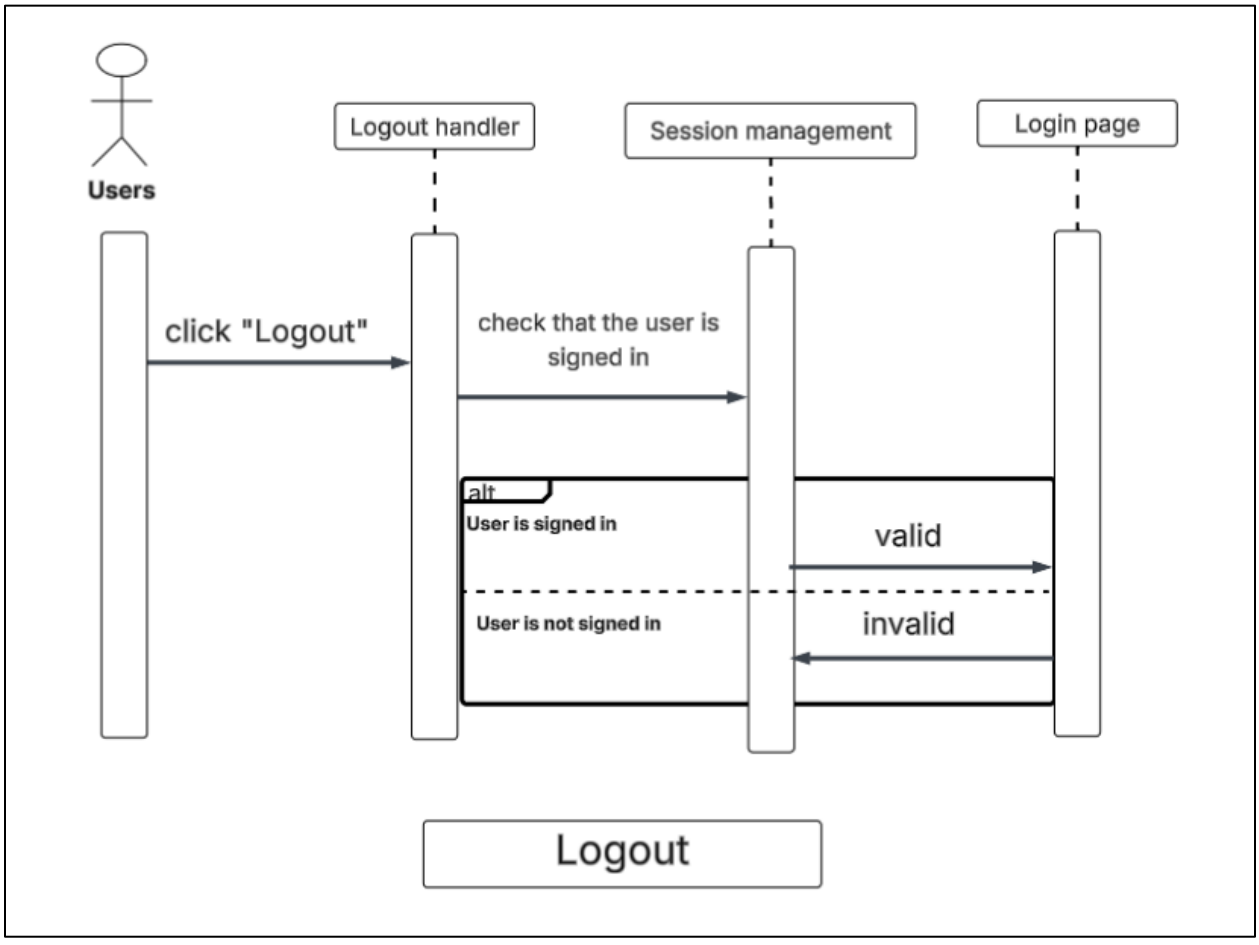


Figure 17: The Logout Sequence Diagram

Table 9: The Registration Use-Case Description

Registration	
Actors	Users, Designers, Developers, Data team, and Regulatory Authorities.
Description	The user enters a username, password, and confirms the password. Then, the user clicks the **"Register"** button. The system checks the input and, if valid, saves the information in the database.
Data	Username ,password and confirm this password
Stimulus	The user enters the required information and if there are no errors click the register button. If the username already exists, or the password and confirmation do not match, or any field is left empty, the system displays a relevant error and prevents registration.
Response	If the input is valid, the system saves the user's data to the database and displays a success message: "Registration successful" However, if there are any validation errors such as an existing username, mismatched passwords, or empty fields the system displays the appropriate error message and does not save the data.
Components	C1

*C1 **satisfies** these functional requirements: FR1, FR6, FR7, FR8, FR9, FR10.

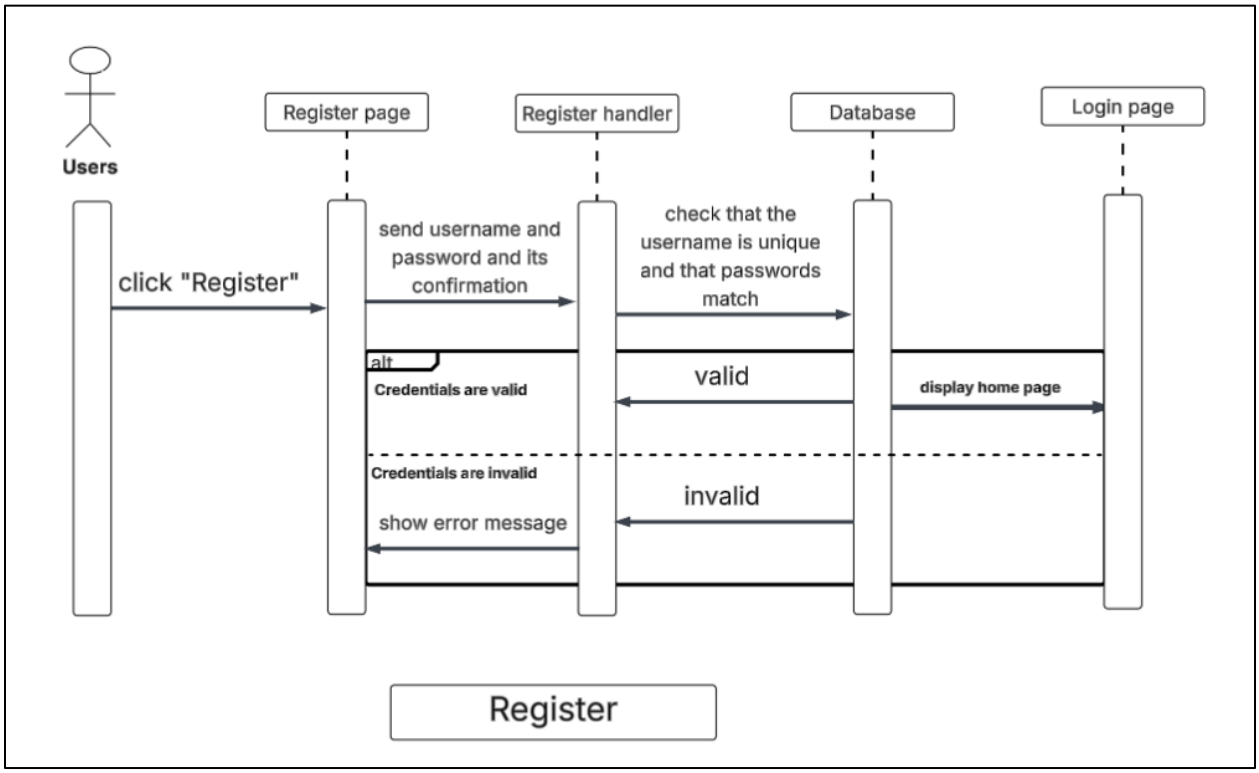


Figure 18:The Register Sequence Diagram

Table 10: The Validation Use-Case Description

Validation	
Actors	Designers, Developers, Data team
Description	The log in button sends the username and password to check if they are valid if not it sends error message to the user, and when the user wants to register it check that the name is not taken by someone else and check that the password is correct and sends the user data to the database and it response colored text and represent the wight progress.
Data	Username, password, personal user data
Stimulus	The user enters the required information and sends them to the database to check if there are valid or not
Response	If the input is valid, it will log in to the account and if it is not, it will send error message to the user to check the username or the password to correct them. And if the user enters the required information about their diet plan it will response colored text depending on the information they enter.
Components	C4

* C4 **satisfies** these functional requirements: FR4, FR5, FR7, FR8, FR10, FR12, FR13, FR16, FR17, FR19, FR21, FR28

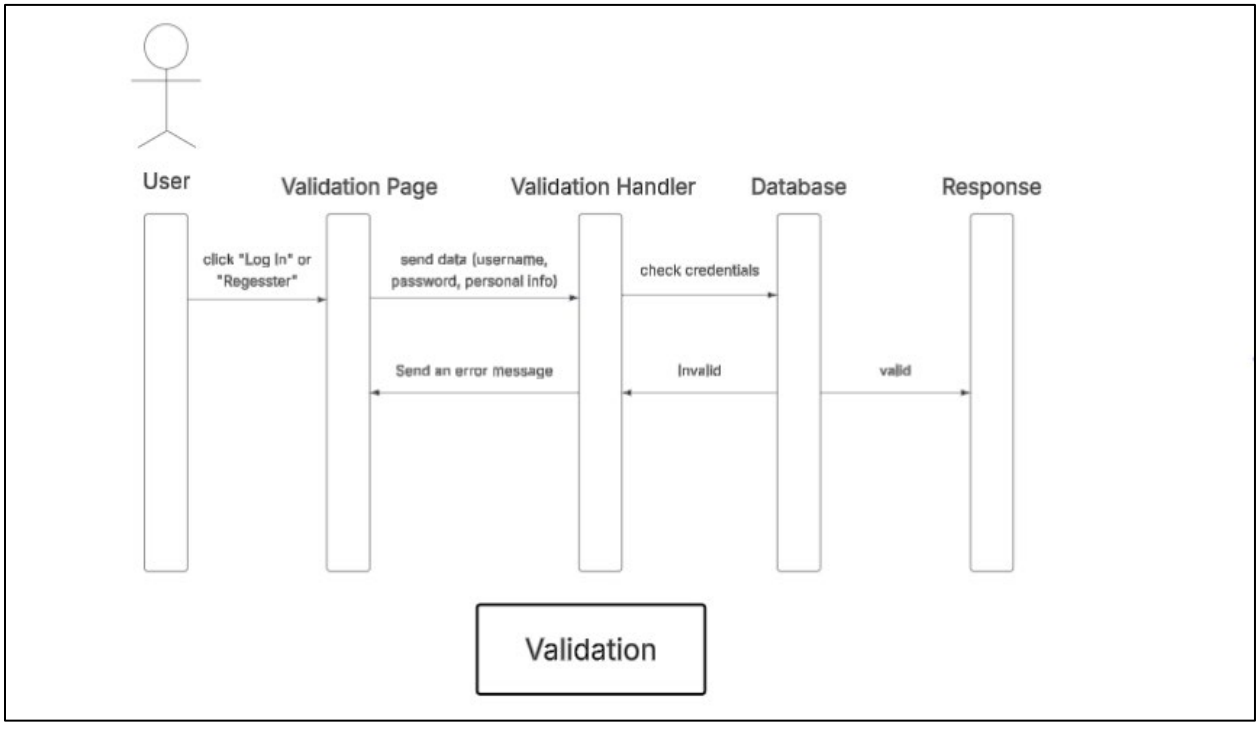


Figure19: The Validation Sequence Diagram

Table 11: The Storing Information Use-Case Description

Storing Information	
Actors	Data team
Description	When the user enters the information like username, password, or personal data it will be sent to the database to check or store the information.
Data	Username, password, and personal user data.
Stimulus	If the user enters the data whatever data type it will be sent to the database to decide if to save or reject it and send message to the user if the data is ok or not.
Response	The database will response an error message if the data is incorrect and if it is a correct data, it will response progress percentage circle and colored text (red or green) depends on their information.
Components	C5

*C5 **satisfies** these functional requirements: FR3, FR9, FR12, FR13, FR14, FR15, FR17, FR18.

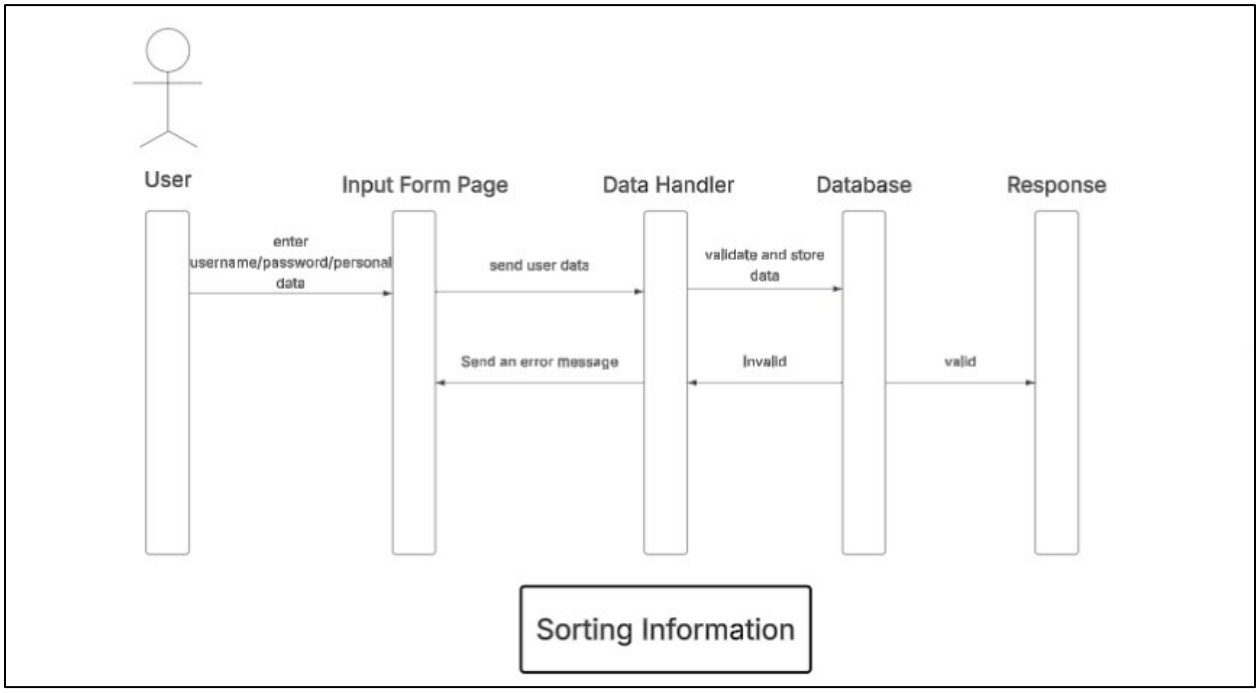


Figure 17: The Storing Information Sequence Diagram

Table 12: The Filtering Use-Case Description

Filtering	
Actors	Users, Designers, Developers
Description	Users can view the food and recipes recommended with instructions to make recipes and nutritional values in each food or recipe and can filter results based on nutrients chosen from a drop-down list then click the filter button to obtain desired results.
Data	Images for recipes and food, instructions, notes and nutritional values .
Stimulus	The user selects the food or recipes tab, then he /she will be able to view recommendations and then when an item from the drop-down list is selected and the filter button is clicked, the recommendation will only include recipes or food having the nutrient selected.
Response	The filter process will compare recipes or food having the nutrient selected form the dropdown list and limit recommendations based on it.
Components	C10

*C10 **satisfies** these functional requirements:FR22, FR23, FR24, FR25, FR26, FR27.

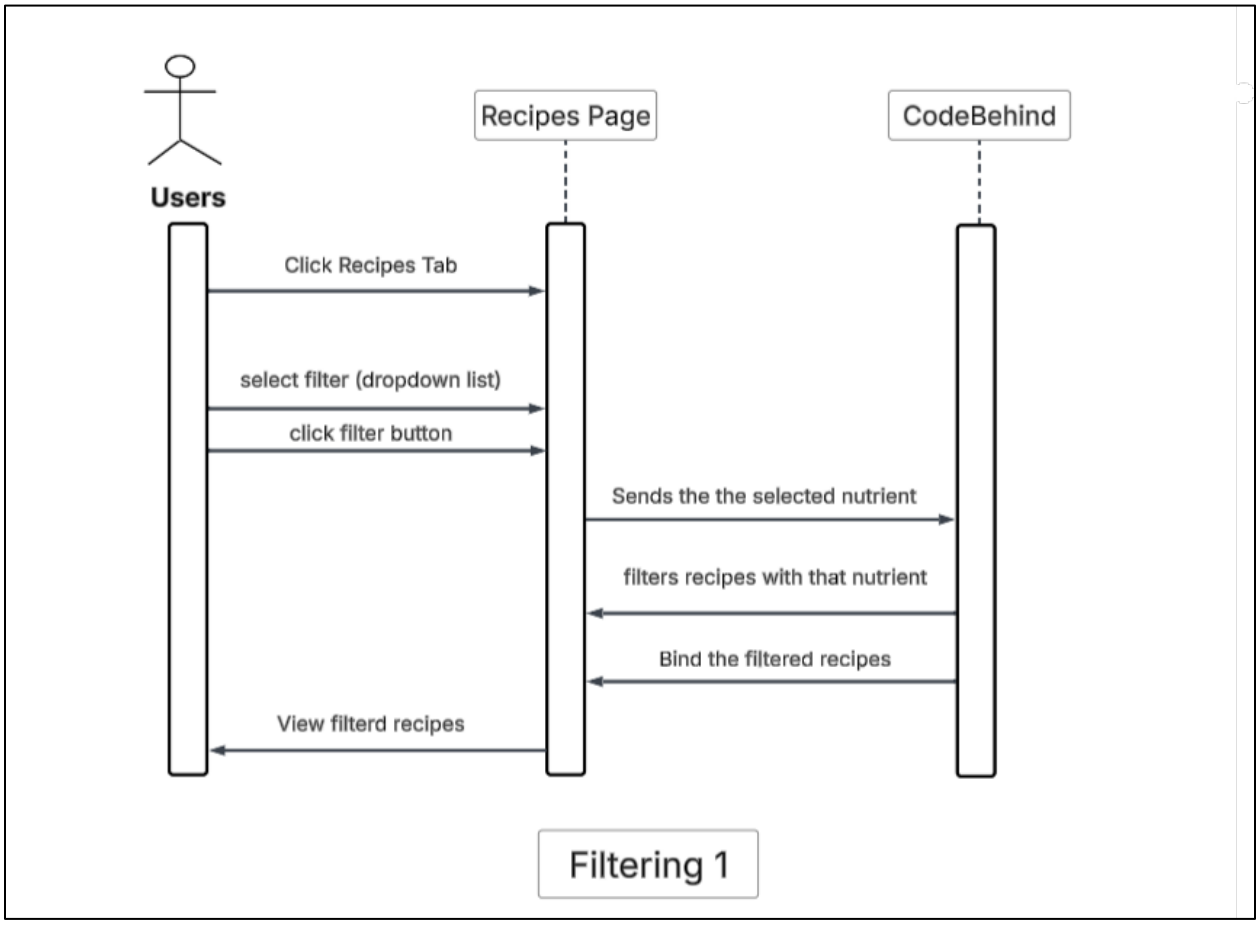


Figure 21: The Filtering Sequence Diagram part1

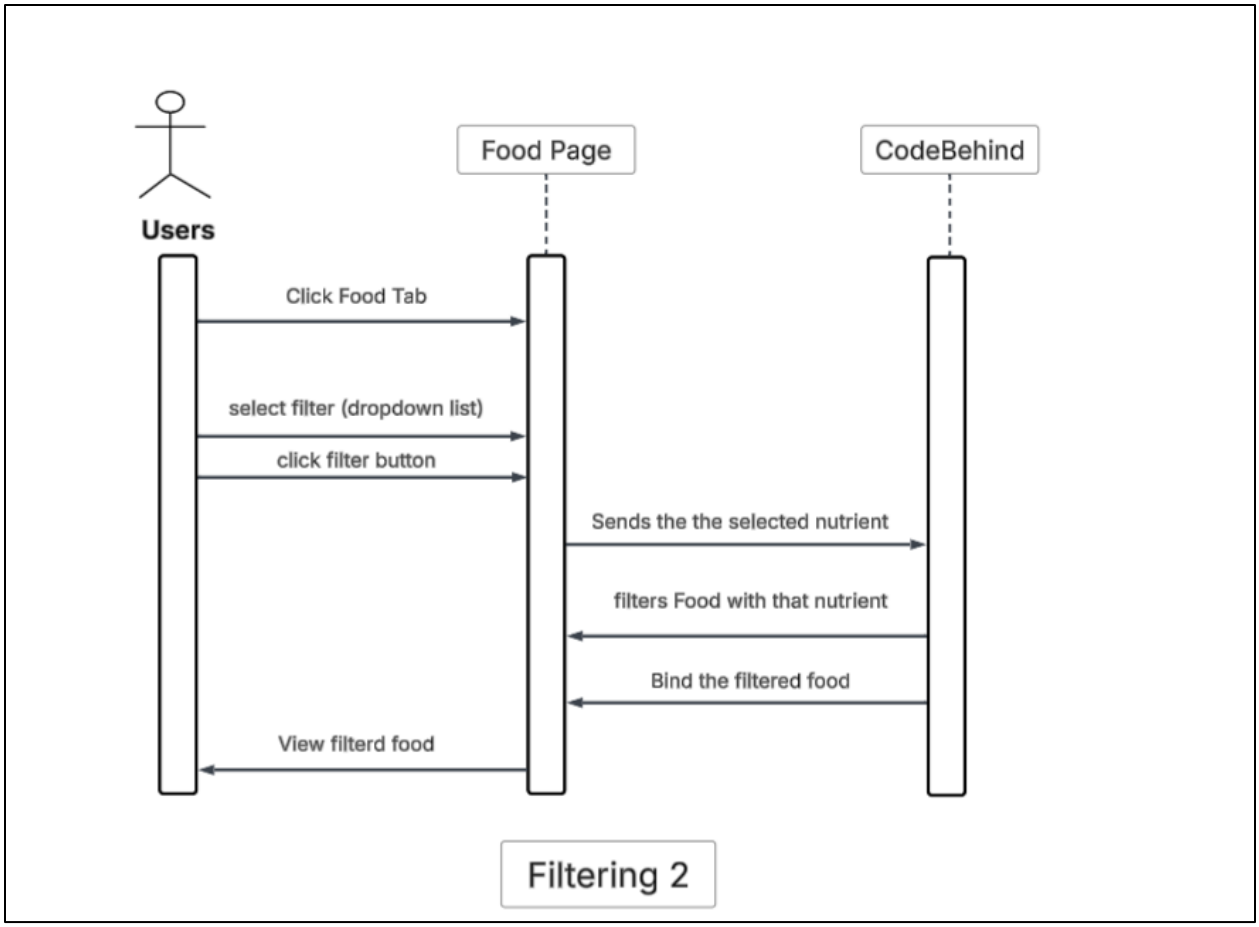


Figure 22: The Filtering Sequence Diagram part2

Table 13: The Personal Data Use-Case Description

Personal Data	
Actors	Users, Developers, Data Teams, and Designers.
Description	Users enter their data (nutrition levels, current and goal weight, height, gender, and age) after login. They then click the" Save" button, which sends this information to be stored, checked in the database.
Data	User's nutrition rate, current weight, Goal weight, height, gender and age.
Stimulus	After the user enters his/her data or edit them, the user clicks the "Save" button to submit the information to the database, the user clicks" Mark as followed today" button to measure diet progress.
Response	The system validates the user's input, the data stored in the database, the system provide feeds back which are BMI result, progress color, and confirmation message.
Components	C7

*C7 **satisfies** these functional requirements:FR11, FR12, FR14, FR15, FR18, FR19, FR20,FR21.

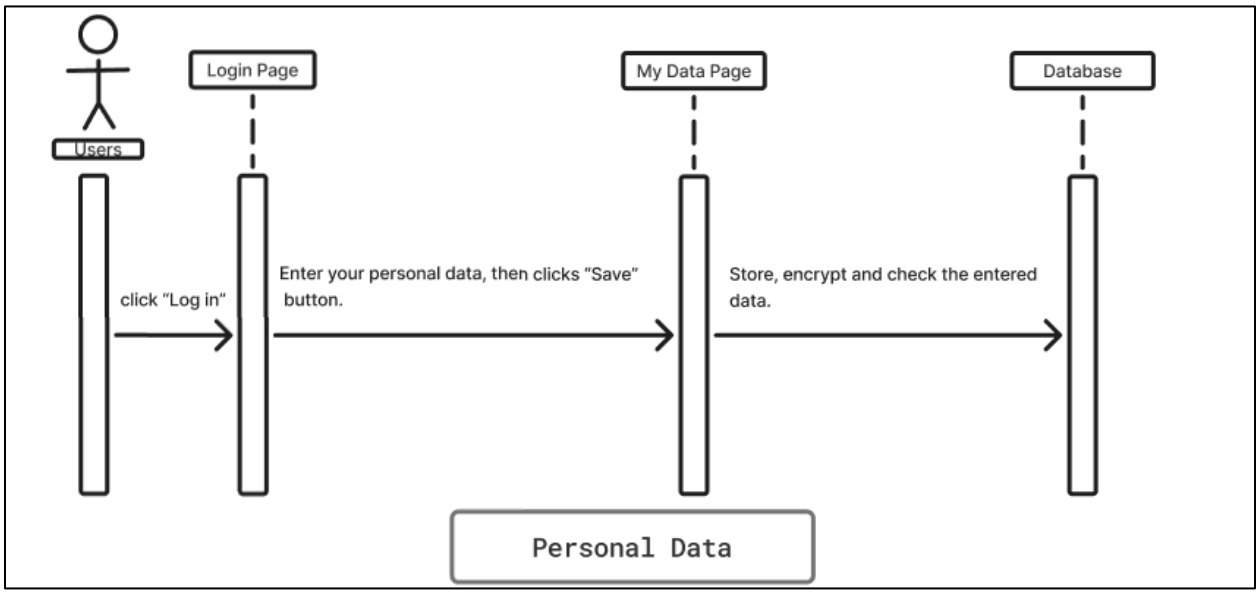


Figure 23: The Personal Data Sequence Diagram

Table 14: The Nutritional Data Use-Case Description

Nutritional Data	
Actors	Users, designers, developers, data team, regulatory Authorities, contact team.
Description	Users can input specific nutritional values, including Ferritin, Vitamin B12, Omega-3, Zinc, Calcium, and Magnesium. Upon clicking the "Save" button, the system validates the data, encrypts it, and securely stores it in the database and a visual feedback feature analyzes the data and provides a real-time nutritional status indicator using color-coded labels: green for normal levels and red for critically low values. This helps users monitor their health effectively and make informed decisions.
Data	Nutritional values (Ferritin, B12, Omega-3, Zinc, Calcium, Magnesium)
Stimulus	User enters nutritional data, the user clicks on button" Save".
Response	Validates and encrypts user data before storing, shows a color-coded label with a message about nutritional status.
Components	C8

* C8 **satisfies** these functional requirements: FR11, FR12, FR13, FR16.

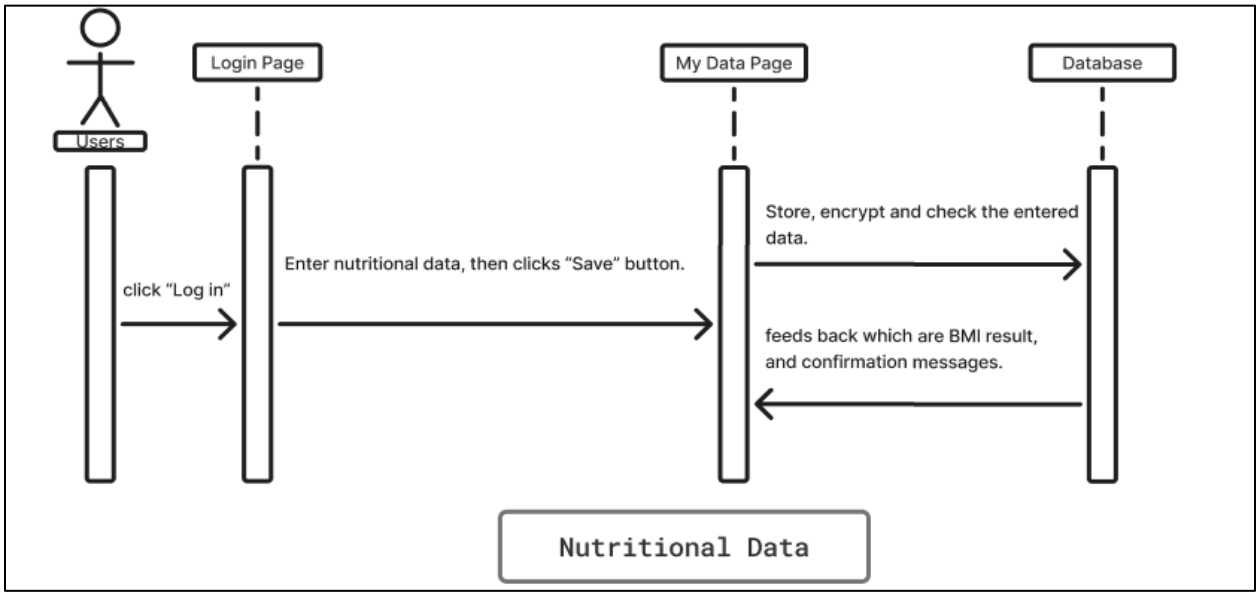


Figure 24: The Nutritional Data Sequence Diagram

Table 15: The Progress Use-Case Description

Progress	
Actors	Users, designers, developers, data team.
Description	The system tracks the user's adherence to their diet by displaying a progress percentage circle. This circle visually indicates how many days the user has followed the diet, based on their clicking of a “follow days” button. The progress resets every month to encourage continuous tracking.
Data	User login credentials and profile information, nutrition data entered by the user, and the diet adherence data (days followed, reset period).
Stimulus	User logs into the system, User clicks the” Save“ button to store nutrition data, User clicks the “follow days” button to log diet adherence.
Response	Shows the “My Data” page or first-time form based on login status, validates, encrypts, and stores nutrition data securely, Updates and displays a progress circle showing diet adherence percentage, resetting monthly.
Components	C9

* C9 **satisfies** these functional requirements: FR11, FR12, FR17.

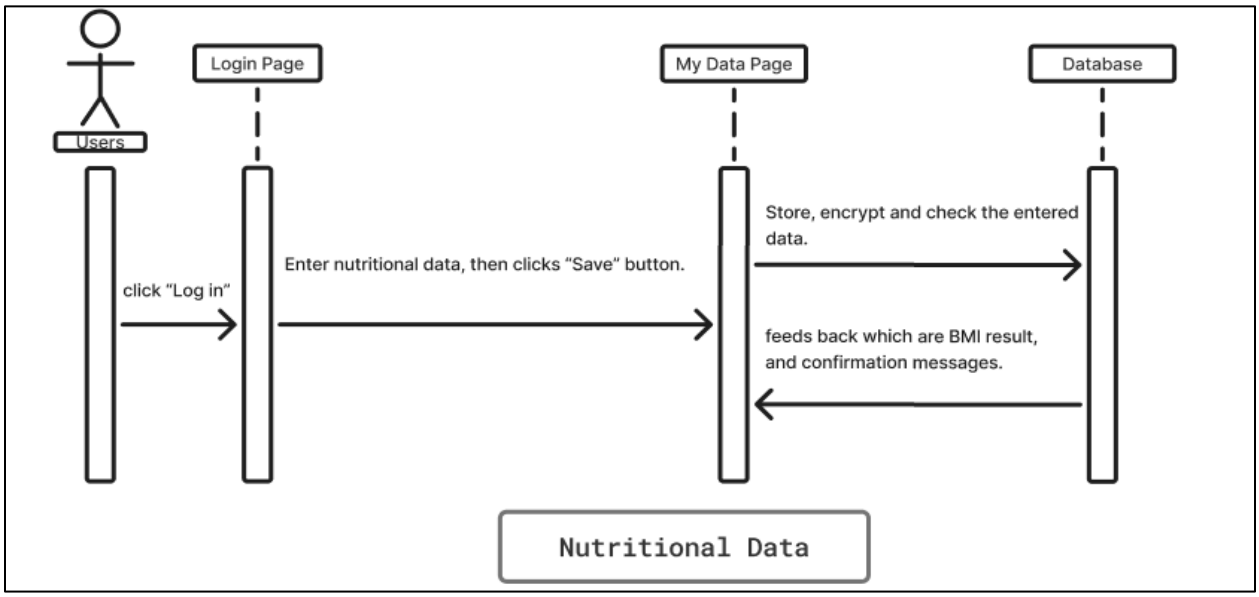


Figure 25: The Nutritional Data Sequence Diagram

Table 16: The Contact Use-Case Description

Contact	
Actors	Users, Developers, Designers, Contact Team
Description	The system will provide a way for users to get answers or complaints about problems they face by filling in the name, email, and message fields or reaching the contact team directly by clicking on their number or email.
Data	Uname, email, and message or complaint they wish to tell the contact team about.
Stimulus	Users will fill in their names and email then in the message field they will enter the message or complaint, and press send message.
Response	The contact team will respond to them with an email message.
Components	C6

* C6 **satisfies** these functional requirements: FR29, FR30, FR31.

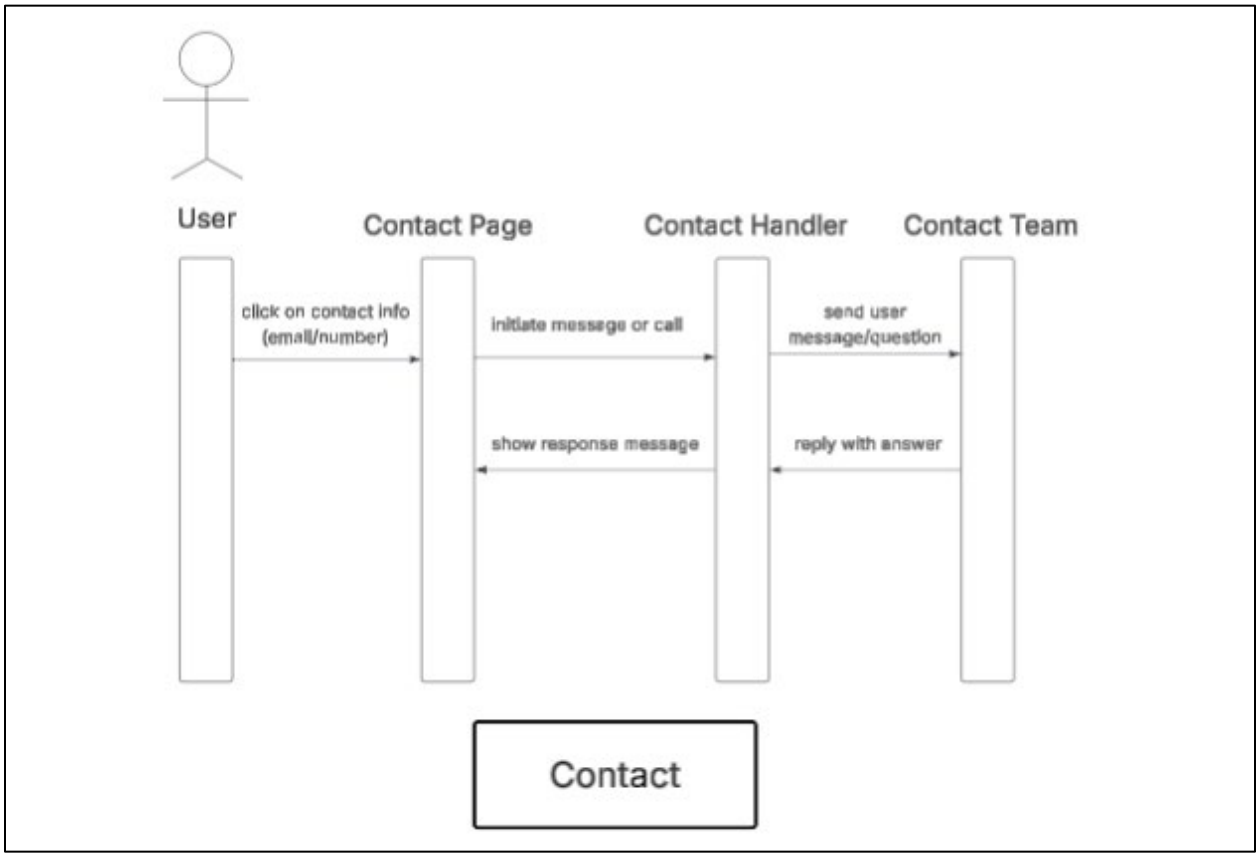


Figure 26: The Contact Sequence Diagram

5.0 Chapter Five

5.1 The Programming Language

We have used VB.NET to write the code with HTML , and CSS to style our website.

5.1.1 Significant Parts in Code

Login Page

```
2 references
Public Class login
    Inherits System.Web.UI.Page
    Dim obj As New mainclass
    0 references
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

    End Sub

    0 references
    Protected Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim obj As New mainclass()
        Dim user As String = obj.login(UserName.Text, Password.Text)

        If user <> "Error" Then

            Session("UserName") = user


            FormsAuthentication.RedirectFromLoginPage(user, False)


            Response.Redirect("home.aspx")
        Else
            msg1.Visible = True
            msg1.Text = "Wrong UserName Or Password"
        End If
    End Sub



End Class
```

Figure 27: The code for Login Page 1

```
<%@ Page Title="Login" Language="vb" AutoEventWireup="false" MasterPageFile="~/logInM.Master" CodeBehind="login.aspx.vb" Inherits="advSoftware.login" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server" >
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <h2>Login to Your Health Portal</h2>

    <label for="UserName">Username:</label>
    <asp:TextBox ID="UserName" runat="server" />

    <label for="Password">Password:</label>
    <asp:TextBox ID="Password" runat="server" TextMode="Password" />

    <asp:Button ID="Button1" runat="server" Text="Login" CssClass="btn" />

    <asp:Label ID="msg1" runat="server" CssClass="error" />

    <a href="Register.aspx">Don't have an account? Register here</a>
</asp:Content>

```

Figure 28: The code for Login Page 2

```
<%@ Master Language="VB" AutoEventWireup="false" CodeBehind="logInM.master.vb" Inherits="advSoftware.logInM" %>

<!DOCTYPE html>
<html>
<head runat="server">
    <title>Health & Diet Portal</title>
    <asp:ContentPlaceHolder ID="head" runat="server">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </asp:ContentPlaceHolder>

    <style>
        body {
            background: linear-gradient(to right, #a8e063, #56ab2f);
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
            margin: 0;
        }

        .login-box {
            background: #ffffff;
            padding: 40px;
            border-radius: 12px;
            box-shadow: 0 8px 20px rgba(0, 0, 0, 0.2);
            text-align: left;
            width: 100%;
            max-width: 400px;
        }

        .login-box h2 {
            margin-bottom: 20px;
            color: #388e3c;
            text-align: center;
        }
```

Figure 29: The code for Login Page 3

Register Page

```
<%@ Page Title="Register" Language="VB" AutoEventWireup="false" MasterPageFile="~/logInM.Master" CodeBehind="Register.aspx.vb" Inherits="advSoftware.Register" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server" >
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <h2>Create Your Account</h2>

    <label for="usern">Username:</label>
    <asp:TextBox ID="usern" runat="server" />

    <label for="pass1">Password:</label>
    <asp:TextBox ID="pass1" runat="server" TextMode="Password" />

    <label for="pass2">Confirm Password:</label>
    <asp:TextBox ID="pass2" runat="server" TextMode="Password" />

    <asp:Button ID="Button1" runat="server" Text="Register" CssClass="btn" />
    <div>
    <asp:Label ID="msg1" runat="server" CssClass="error"/>
    </div>
    <asp:Label ID="msg2" runat="server" CssClass="error" ForeColor="#33CC33" Visible="False"/>
    <a href="login.aspx">

    Already have an account? Login here</a>
</asp:Content>
```

Figure 30:The code for Register Page 1

```
Public Class Register
    Inherits System.Web.UI.Page
    Dim obj As New mainclass
    0 references
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

    End Sub



    0 references
    Protected Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        ' Check for empty username or password
        If String.IsNullOrWhiteSpace(usern.Text) OrElse
           String.IsNullOrWhiteSpace(pass1.Text) OrElse
           String.IsNullOrWhiteSpace(pass2.Text) Then

            msg1.Visible = True
            msg2.Visible = False
            msg1.Text = "Username and password fields cannot be empty."
            Exit Sub
        End If

        ' Check if passwords match
        If pass1.Text = pass2.Text Then
            ' Check if username already exists
            If obj.IsUsernameTaken(usern.Text) Then
                msg1.Visible = True
                msg2.Visible = False
                msg1.Text = "Username already exists. Please choose another one."
            Else
                ' Attempt to create account
                Dim success As Boolean = obj.createacc(usern.Text, pass1.Text, pass2.Text)
                If success Then
                    msg2.Visible = True
                    msg1.Visible = False
                    msg2.Text = "Registration successful"
                Else
                    msg1.Visible = True
                    msg2.Visible = False
                    msg1.Text = "An error occurred during registration"
                End If
            End If
        Else
            msg1.Visible = True
            msg2.Visible = False
            msg1.Text = "Passwords do not match"
        End If
```

Figure 31:The code for Register Page 2

My Data Page

```
2 references
Public Class home
    Inherits System.Web.UI.Page

    0 references
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        If Not IsPostBack Then
            If Session("StartDate") Is Nothing Then
                Session("StartDate") = DateTime.Today
            End If
            If Session("FollowedDays") Is Nothing Then
                Session("FollowedDays") = 0
            End If
            If Session("UserName") IsNot Nothing Then
                Dim usr As String = Session("UserName").ToString()
                LoadUserData(usr)
            Else
                Response.Redirect("login.aspx")
            End If

            UpdateDietProgress()
            UpdateWeightProgress()
            UpdateBMI()
            EvaluateNutritionLevels()
        End If
    End Sub

    1 reference
    Private Sub LoadUserData(userName As String)
        Dim mc As New mainclass()
        Dim dt1 As DataTable = mc.GetPersonalData(userName)
        Dim dt2 As DataTable = mc.GetNutritionData(userName)

        If dt1.Rows.Count > 0 Then
            ddlGender.SelectedValue = dt1(0)("Gender").ToString()
            txtAge.Text = dt1(0)("Age").ToString()
            txtHeight.Text = dt1(0)("HeightCm").ToString()
            txtCurrentWeight.Text = dt1(0)("CurrentWeight").ToString()
            txtGoalWeight.Text = dt1(0)("GoalWeight").ToString()
        End If
```

Figure32:The code for My Data Page 1

```
139         If followedDays = 30 Then
140             followedDays = 0
141         End If
142     End Sub
143
            0 references
144     Protected Sub btnFollowedToday_Click(ByVal sender As Object, ByVal e As EventArgs)
145         Dim usr As String = Session("UserName").ToString()
146         Dim mc As New mainclass()
147
148         Dim lastFollowDate As Date = mc.GetLastFollowDate(usr)
149
150         If lastFollowDate = Date.Today Then
151             lblFollowedDays.Text = "Already marked today!"
152             lblFollowedDays.CssClass = "status-yellow"
153             Return
154         End If
155
156         ' Increment followed days
157         Dim followedDays As Integer = mc.GetFollowedDays(usr) + 1
158         mc.SaveFollowProgress(usr, followedDays, Date.Today)
159
160         ' Update session/cache for UI
161         Session("FollowedDays") = followedDays
162         ViewState("LastFollowDate") = Date.Today
163
164         UpdateDietProgress()
165     End Sub
166
167
168
            3 references
169     Private Sub EvaluateNutritionLevels()
170         Try
171             Dim ferritin = Val(txtFerritin.Text)
172             Dim zinc = Val(txtZinc.Text)
173             Dim b12 = Val(txtB12.Text)
174             Dim calcium = Val(txtCalcium.Text)
175             Dim omega3 = Val(txtOmega3.Text)
176             Dim magnesium = Val(txtMagnesium.Text)
177
178             Dim danger = ""
```

Figure 33:The code for My Data Page 2

```
148     </style>
149 </asp:Content>
150
151 <asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
152     <div class="tracker-container">
153         <h2>User Health Tracker</h2>
154
155         <div class="tracker-form">
156             <table class="auto-style1">
157                 <tr>
158                     <td class="auto-style2"><label>Gender</label></td>
159                     <td><label>Age</label></td>
160                     <td><label>Height (cm)</label></td>
161                 </tr>
162                 <tr>
163                     <td class="auto-style2">
164                         <asp:DropDownList ID="ddlGender" runat="server">
165                             <asp:ListItem Text="Male" Value="Male" />
166                             <asp:ListItem Text="Female" Value="Female" />
167                         </asp:DropDownList>
168                     </td>
169                     <td><asp:TextBox ID="txtAge" runat="server" /></td>
170                     <td><asp:TextBox ID="txtHeight" runat="server" /></td>
171                 </tr>
172                 <tr>
173                     <td class="auto-style2"><label>Current Weight (kg)</label></td>
174                     <td><label>Goal Weight (kg)</label></td>
175                 </tr>
176                 <tr>
177                     <td class="auto-style2"><asp:TextBox ID="txtCurrentWeight" runat="server" /></td>
```

Figure 34:The code for My Data Page 3

```
<%@ Page Title="Health Tracker" Language="vb" AutoEventWireup="false" MasterPageFile="~/MainPage.Master" CodeBehind="home.aspx.vb" Inherits="advSoftware.home" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
    <style>
        html, body {
            margin: 0;
            padding: 0;
            width: 100%;
            overflow-x: hidden;
            font-family: 'Segoe UI', sans-serif;
            background-color: #f9f9f9;
            box-sizing: border-box;
        }

        *, *::before, *::after {
            box-sizing: inherit;
        }

        .tracker-container {
            max-width: 1000px;
            margin: 0 auto;
            padding: 30px 20px;
        }

        h2 {
             font-size: 40px;
            font-weight: bold;
            color: #2e7d32;
            margin-bottom: 25px;
            text-align: center;
            max-width: 1000px;
            margin: 0 auto;
            padding: 30px 20px;
            /*font-size: 32px;
            color: #2e7d32;
            margin-bottom: 25px;
            text-align: center;*/
        }

        .tracker-form {
```

Figure 35:The code for My Data Page 4

```
                    <td><label>Ferritin</label></td>
                    <td><label>Zinc</label></td>
                </tr>
                <tr>
                    <td><asp:TextBox ID="txtFerritin" runat="server" /></td>
                    <td><asp:TextBox ID="txtZinc" runat="server" /></td>
                </tr>
                <tr>
                    <td><label>B12</label></td>
                    <td><label>Calcium</label></td>
                </tr>
                <tr>
                    <td><asp:TextBox ID="txtB12" runat="server" /></td>
                    <td><asp:TextBox ID="txtCalcium" runat="server" /></td>
                </tr>
                <tr>
                    <td><label>Omega 3</label></td>
                    <td><label>Magnesium</label></td>
                </tr>
                <tr>
                    <td><asp:TextBox ID="txtOmega3" runat="server" /></td>
                    <td><asp:TextBox ID="txtMagnesium" runat="server" /></td>
                </tr>
            </table>

            <asp:Label ID="lblNutritionStatus" runat="server" CssClass="centered" />

            <div class="btn-group">
                <asp:Button runat="server" Text="Save" CssClass="auto-style3" OnClick="btnSave_Click" Width="685px" />
                <asp:Button ID="btnUpdate" runat="server" Text="Update" CssClass="btn-action" Enabled="false" Visible="False" OnClick="btnUpdate_Click" />
            </div>

            <div class="section-title">Diet Progress</div>
            <asp:Button ID="btnFollowedToday" runat="server" Text="Mark as Followed Today" CssClass="btn-action" OnClick="btnFollowedToday_Click" Width="1086px" />
            <asp:Label ID="lblFollowedDays" runat="server" CssClass="centered" />

            <div class="circle">
                <asp:Label ID="lblProgressCircle" runat="server" Text="0%" />
            </div>
        </div>
    </div>
</asp:Content>
```

Figure 36:The code for My Data Page 5

Food Page

```
        }
    }
</style>
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <div class="food-container">
        <div class="food-header">Fruits & Vegetables</div>

        <div class="filter-container">
            <label for="ddlNutrient">Filter by Nutrient:</label>
            <asp:DropDownList ID="ddlNutrient" runat="server">
                <asp:ListItem Text="Select a Nutrient" Value="" />
                <asp:ListItem Text="Omega 3" Value="Omega3" />
                <asp:ListItem Text="Ferritin" Value="Ferritin" />
                <asp:ListItem Text="Zinc" Value="Zinc" />
                <asp:ListItem Text="Magnesium" Value="Magnesium" />
                <asp:ListItem Text="Calcium" Value="Calcium" />
                <asp:ListItem Text="B12" Value="B12" />
            </asp:DropDownList>
            <asp:Button ID="btnFilter" runat="server" Text="Filter" CssClass="btn-filter" OnClick="btnFilter_Click" />
        </div>

        <div class="food-list">
            <asp:Repeater ID="foodRepeater" runat="server">
                <ItemTemplate>
                    <div class="food-card">
                        <img src='<%# Eval("ImageUrl") %>' class="food-image" />
                        <div class="food-name"><%# Eval("FoodName") %></div>
                        <div class="food-instructions"><%# Eval("Notes") %></div>
                        <div class="food-nutrients">
                            <strong>Nutrients:</strong><br />
                            Omega 3: <%# Eval("Omega3") %><br /><%-- to extracts values from the data source bind to the repeater--%>
                            Ferritin: <%# Eval("Ferritin") %><br />
                            Zinc: <%# Eval("Zinc") %><br />
                            Magnesium: <%# Eval("Magnesium") %><br />
                            Calcium: <%# Eval("Calcium") %><br />
                            B12: <%# Eval("B12") %><br />
                        </div>
                    </div>
                </ItemTemplate>
            </asp:Repeater>
        </div>S
    </div>
</asp:Content>

```

Figure 37:The code for Food Page 1

```
<%@ Page Title="" Language="VB" AutoEventWireup="false" MasterPageFile="~/MainPage.Master" CodeBehind="Food.aspx.vb" Inherits="adv5oftware.Food" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
    <style>
        .food-container {
            display: flex;
            flex-direction: column;
            align-items: flex-start;
        }

        .food-header {
            font-size: 40px;
            font-weight: bold;
            color: #2e7d32;
            margin-bottom: 25px;
            text-align: center;
            max-width: 1000px;
            margin: 0 auto;
            padding: 30px 20px;
        }

        .filter-container {
            margin-bottom: 20px;
        }

        .btn-filter {
            background-color: #2e7d32;
            color: white;
            padding: 10px 20px;
            font-size: 16px;
            border-radius: 8px;
```

Figure 38:The code for Food Page 2

```
1   Imports System.Collections.Generic
2
    24 references
3   Public Class FoodInfo
        22 references
4       Public Property FoodName As String
        22 references
5       Public Property ImageUrl As String
        22 references
6       Public Property Omega3 As String
        22 references
7       Public Property Ferritin As String
        22 references
8       Public Property Zinc As String
        22 references
9       Public Property Magnesium As String
        22 references
10      Public Property Calcium As String
        22 references
11      Public Property B12 As String
        22 references
12      Public Property Notes As String
13  End Class
14
    2 references
15  Public Class Food
16      Inherits System.Web.UI.Page
17
18      Private foodsList As New List(Of FoodInfo) From {
19          New FoodInfo With {
20              .FoodName = "Spinach",
21              .ImageUrl = "Resources/spinach.jpg",
22              .Omega3 = "370mg",
23              .Ferritin = "2.7mg",
24              .Zinc = "0.5mg",
25              .Magnesium = "79mg",
26              .Calcium = "99mg",
27              .B12 = "0mcg",
28              .Notes = "Rich in iron and magnesium. Great for salads and smoothies."
29          },
```

Figure 39:The code for Food Page 3

```
259             .Notes = "Contain small amounts of B12 when sun-dried."
260         }
261     }
262
        0 references
263     Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
264         If Not IsPostBack Then
265             BindFoods()
266             If Session("UserName") IsNot Nothing Then
267                 Dim usr As String = Session("UserName").ToString()
268             Else
269                 Response.Redirect("login.aspx")
270             End If
271         End If
272     End Sub
273
        2 references
274     Private Sub BindFoods(Optional nutrient As String = "")
275         Dim filteredFoods As List(Of FoodInfo) = foodsList
276
277         If Not String.IsNullOrEmpty(nutrient) Then
278             filteredFoods = foodsList.Where(Function(f)
279                                                 Dim nutrientValue As String = f.GetType().GetProperty(nutrient).GetValue(f, Nothing).ToString()
280                                                 Return Not String.IsNullOrEmpty(nutrientValue) AndAlso nutrientValue <> "0mg" AndAlso nutrientValue <> "0mcg"
281                                             End Function).ToList()
282         End If
283
284         foodRepeater.DataSource = filteredFoods
285         foodRepeater.DataBind()
286     End Sub
287
        0 references
288     Protected Sub btnFilter_Click(sender As Object, e As EventArgs) Handles btnFilter.Click
289         Dim selectedNutrient As String = ddlNutrient.SelectedValue
290         BindFoods(selectedNutrient)
291     End Sub
292 End Class
293
```

Figure 40:The code for Food Page 4

Recipes Page

```
                flex: 0 0 100%;
            }
        }
    </style>
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
    <div class="recipe-container">
        <div class="recipe-header">Recipes</div>

        <!-- Filter Dropdown -->
        <div class="filter-container">
            <label for="ddlNutrient">Filter by Nutrient:</label>
            <asp:DropDownList ID="ddlNutrient" runat="server">
                <asp:ListItem Text="Select a Nutrient" Value="" />
                <asp:ListItem Text="Omega 3" Value="Omega3" />
                <asp:ListItem Text="Ferritin" Value="Ferritin" />
                <asp:ListItem Text="Zinc" Value="Zinc" />
                <asp:ListItem Text="Magnesium" Value="Magnesium" />
                <asp:ListItem Text="Calcium" Value="Calcium" />
                <asp:ListItem Text="B12" Value="B12" />
            </asp:DropDownList>
            <asp:Button ID="btnFilter" runat="server" Text="Filter" CssClass="btn-filter" OnClick="btnFilter_Click" />
        </div>

        <!-- Recipe List -->
        <div class="recipe-list">
            <asp:Repeater ID="recipeRepeater" runat="server">
                <ItemTemplate>
                    <div class="recipe-card">
                        <img src="<%# Eval("ImageUrl") %>" class="recipe-image" />
```

Figure 41: The code for Recipes Page 1

```
    .RecipeName = "Cabbage Stir Fry",
    .ImageUrl = "Resources/cabbage-stir-fry-recipe.jpg",
    .Omega3 = "50mg",
    .Ferritin = "5mcg",
    .Zinc = "2mg",
    .Magnesium = "60mg",
    .Calcium = "180mg",
    .B12 = "0mcg",
    .Instructions = "Stir fry cabbage with garlic, soy sauce, and sesame oil."
}
}

' When the page loads, bind data to the repeater
0 references
Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
    If Not IsPostBack Then
        BindRecipes()
        If Session("UserName") IsNot Nothing Then
            Dim usr As String = Session("UserName").ToString()
        Else
            Response.Redirect("login.aspx")
        End If
    End If
End Sub

' Bind the recipes to the repeater
2 references
Private Sub BindRecipes(Optional nutrient As String = "")
    Dim filteredRecipes As List(Of RecipeInfo) = recipesList

    ' If a nutrient filter is selected, filter the recipes
    If Not String.IsNullOrEmpty(nutrient) Then
        filteredRecipes = recipesList.Where(Function(r)
                                                Dim nutrientValue As String = r.GetType().GetProperty(nutrient).GetValue(r, Nothing).ToString()
                                                ' Exclude recipes with a "0" value for the selected nutrient
                                                Return Not String.IsNullOrEmpty(nutrientValue) AndAlso nutrientValue <> "0mcg" AndAlso nutrientValue <> "0mg"
                                            End Function).ToList()
    End If

    ' Bind the filtered list to the repeater
    recipeRepeater.DataSource = filteredRecipes
    recipeRepeater.DataBind()
End Sub
```

Figure 42: The code for Recipes Page 2

Contact Page

```
103  }
104
105      </style>
106  </asp:Content>
107
108  <asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
109      <div class="contact-container">
110          <div class="contact-content">
111              <h1>Contact Us</h1>
112              <div class="contact-form">
113      <asp:Label ID="lblName" runat="server" Text="Your Name" CssClass="form-label"></asp:Label>
114      <asp:TextBox ID="txtName" runat="server" CssClass="form-input" placeholder="Enter your name"></asp:TextBox>
115
116
117      <asp:Label ID="lblEmail" runat="server" Text="Your Email" CssClass="form-label"></asp:Label>
118      <asp:TextBox ID="txtEmail" runat="server" CssClass="form-input" placeholder="Enter your email" TextMode="Email"></asp:TextBox>
119
120
121      <asp:Label ID="lblMessage" runat="server" Text="Your Message" CssClass="form-label"></asp:Label>
122      <asp:TextBox ID="txtMessage" runat="server" CssClass="form-input" TextMode="MultiLine" Rows="6"
123          placeholder="Enter your message"></asp:TextBox>
124
125      <asp:Button ID="btnSubmit" runat="server" Text="Send Message" CssClass="btn-submit" OnClick="btnSubmit_Click" />
126
127      <asp:Label ID="lblStatus" runat="server" ForeColor="Green" Font-Bold="true" />
128  </div>
129
130              <div class="info">
131                  <p>Or reach us directly:</p>
132                  <p><i class='bx bx-phone'></i> Phone: <a href="tel:+1234567890">+1 (234) 567-890</a></p>
133                  <p><i class='bx bx-envelope'></i> Email: <a href="mailto:contact@balance.com">contact@balance.com</a></p>
134              </div>
135          </div>
136      </div>
137  </asp:Content>
138
```

Figure 43: The code for Contact Page 1

```
1   <%@ Page Title="Contact Us" Language="vb" AutoEventWireup="false" MasterPageFile="~/MainPage.Master" CodeBehind="Contact.aspx.vb" Inherits="advSoftware.Contact" %>
2
3   <asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
4       <style>
5           /* the whole content container to center evrything*/
6           .contact-container {
7               display: flex;/*to center the content both vertically and horizontally*/
8               justify-content: center;
9               align-items: center;
10              width: 100%;
11              height: 100vh; /* to fill the whole screen */
12              box-sizing: border-box; /* Prevent overflow */
13          }
14
15          /* Content Box that fills the screen */
16          .contact-content {
17              width: 100%;
18              height: 100%; /* to make sure it takes full height and width of the container */
19              box-sizing: border-box;
20              display: flex;
21              flex-direction: column;
22              justify-content: center;
23          }
24
25          h1 {
26              color: #2e7d32;
27              font-size: 40px;
28              text-align: center;
29              font-weight: bold;
30          }
31
32          .contact-form {
33              display: flex;
34              flex-direction: column;
35              gap: 16px;/*for spacing*/
36              width: 100%;
```

Figure 44: The code for Contact Page 2

Main Class code

```
Imports System.Data.SqlClient
6 references
Public Class mainclass
    Dim con As New SqlConnection("Server=THINKBOOKI5\TESTINSTANCE;Database=Advsoftware;User Id=sa;Password=le-2022;")
    10 references
    Sub OpenDB()
        Try
            If con.State = 0 Then
                con.Open()
            End If
        Catch ex As Exception
        End Try
    End Sub
    10 references
    Sub CloseDB()
        Try
            If con.State = 1 Then
                con.Close()
            End If
        Catch ex As Exception
        End Try
    End Sub
    1 reference
    Function login(usr As String, pw As String) As String
        Try
            Dim ad As New SqlDataAdapter("SELECT   a.userName , a.userPw FROM users as a  where a.userName =@usr and a.userPw =@pw ", con)
            ad.SelectCommand.Parameters.Add("@usr", SqlDbType.VarChar, 100).Value = usr
            ad.SelectCommand.Parameters.Add("@pw", SqlDbType.VarChar, 100).Value = pw
            Dim ds As New DataSet

            OpenDB()
            ad.SelectCommand.Prepare()
            ad.Fill(ds)
            If ds.Tables(0).Rows.Count > 0 Then
                Return ds.Tables(0).Rows(0).Item(0)
            Else
                Return "Error"
            End If
        Catch ex As Exception
            Return "Error"
```

Figure 45: The code for Main Class 1

```
    1 reference
    Sub SavePersonalData(user As String, gender As String, age As Integer, height As Decimal, weight As Decimal, goal As Decimal)
        Dim q As String = "IF EXISTS (SELECT 1 FROM UserPersonalData WHERE UserName=@u)
        UPDATE UserPersonalData SET Gender=@g, Age=@a, HeightCm=@h, CurrentWeight=@w, GoalWeight=@gw, DateUpdated=GETDATE() WHERE UserName=@u
    ELSE
        INSERT INTO UserPersonalData (UserName, Gender, Age, HeightCm, CurrentWeight, GoalWeight)
        VALUES (@u, @g, @a, @h, @w, @gw)"

        Dim cmd As New SqlCommand(q, con)
        cmd.Parameters.AddWithValue("@u", user)
        cmd.Parameters.AddWithValue("@g", gender)
        cmd.Parameters.AddWithValue("@a", age)
        cmd.Parameters.AddWithValue("@h", height)
        cmd.Parameters.AddWithValue("@w", weight)
        cmd.Parameters.AddWithValue("@gw", goal)

        OpenDB()
        cmd.ExecuteNonQuery()
        CloseDB()
    End Sub

    1 reference
    Sub SaveNutritionData(user As String, ferr As Decimal, zinc As Decimal, b12 As Decimal, cal As Decimal, omega As Decimal, mag As Decimal)
        Dim q As String = "IF EXISTS (SELECT 1 FROM UserNutritionData WHERE UserName=@u)
        UPDATE UserNutritionData SET Ferritin=@f, Zinc=@z, B12=@b, Calcium=@c, Omega3=@o, Magnesium=@m, DateUpdated=GETDATE() WHERE UserName=@u
    ELSE
        INSERT INTO UserNutritionData (UserName, Ferritin, Zinc, B12, Calcium, Omega3, Magnesium)
        VALUES (@u, @f, @z, @b, @c, @o, @m)"

        Dim cmd As New SqlCommand(q, con)
        cmd.Parameters.AddWithValue("@u", user)
        cmd.Parameters.AddWithValue("@f", ferr)
        cmd.Parameters.AddWithValue("@z", zinc)
        cmd.Parameters.AddWithValue("@b", b12)
        cmd.Parameters.AddWithValue("@c", cal)
        cmd.Parameters.AddWithValue("@o", omega)
        cmd.Parameters.AddWithValue("@m", mag)

        OpenDB()
        cmd.ExecuteNonQuery()
        CloseDB()
```

Figure 46: The code for Main Class 2

Data Base

```
CREATE TABLE UserPersonalData (
    Id INT PRIMARY KEY IDENTITY,
    UserName VARCHAR(100) FOREIGN KEY REFERENCES users(userName),
    Gender VARCHAR(10),
    Age INT,
    HeightCm DECIMAL(5,2),
    CurrentWeight DECIMAL(5,2),
    GoalWeight DECIMAL(5,2),
    DateUpdated DATETIME DEFAULT GETDATE()
)
CREATE TABLE UserNutritionData (
    Id INT PRIMARY KEY IDENTITY,
    UserName VARCHAR(100) FOREIGN KEY REFERENCES users(UserName),
    Ferritin DECIMAL(6,2),
    Zinc DECIMAL(6,2),
    B12 DECIMAL(6,2),
    Calcium DECIMAL(6,2),
    Omega3 DECIMAL(6,2),
    Magnesium DECIMAL(6,2),
    DateUpdated DATETIME DEFAULT GETDATE()
)
```

Figure 187: The code for the Database

5.1.2 Overview

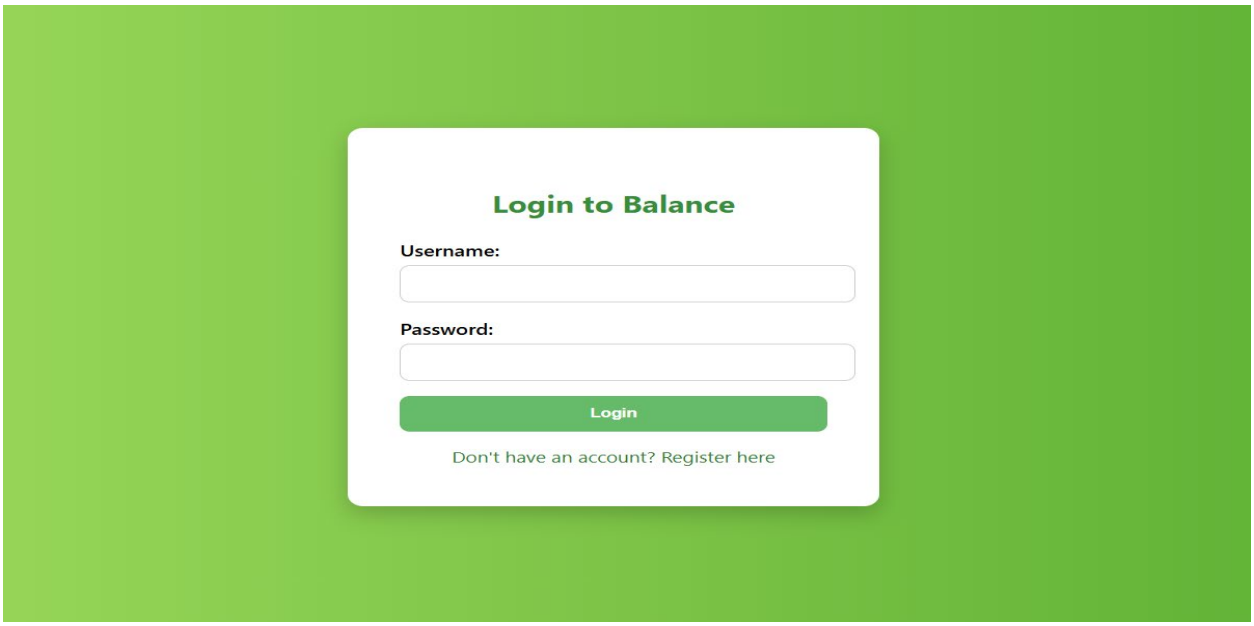


Figure 48:The Login Page Overview

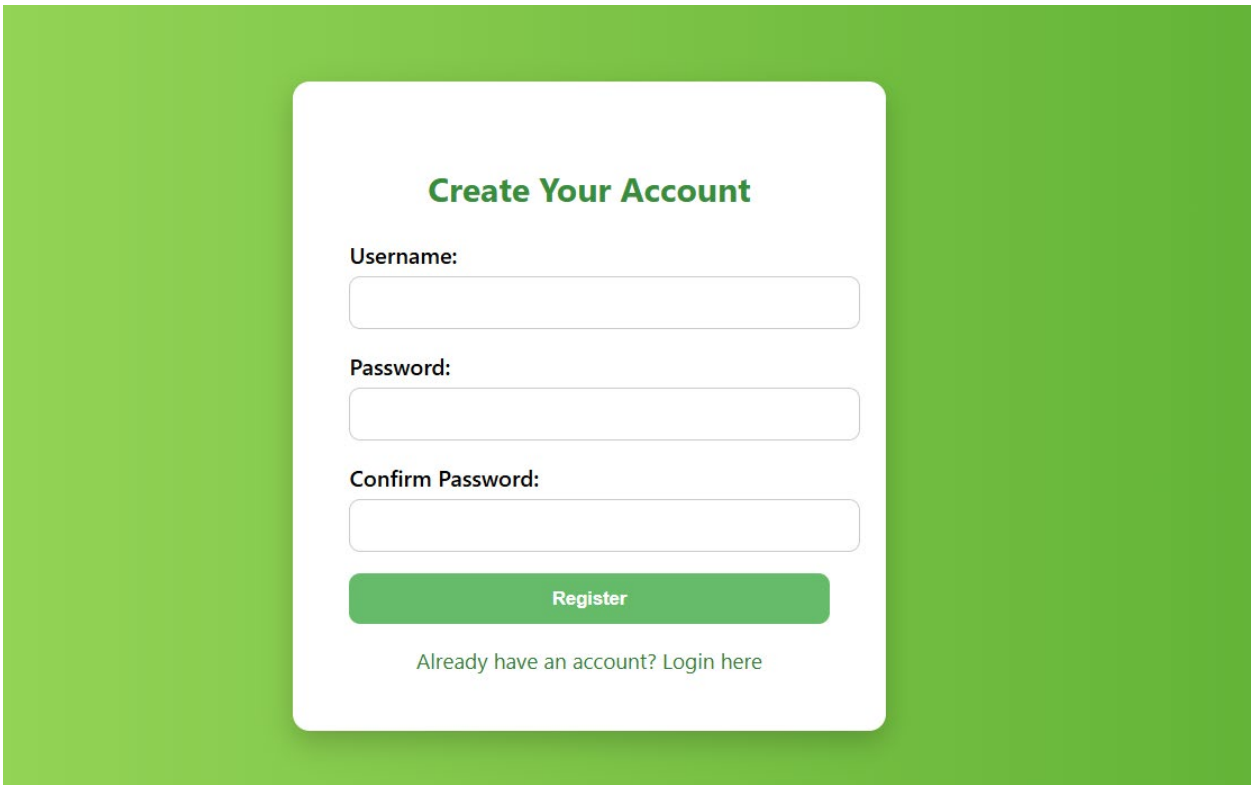


Figure 49: The Register Page Overview

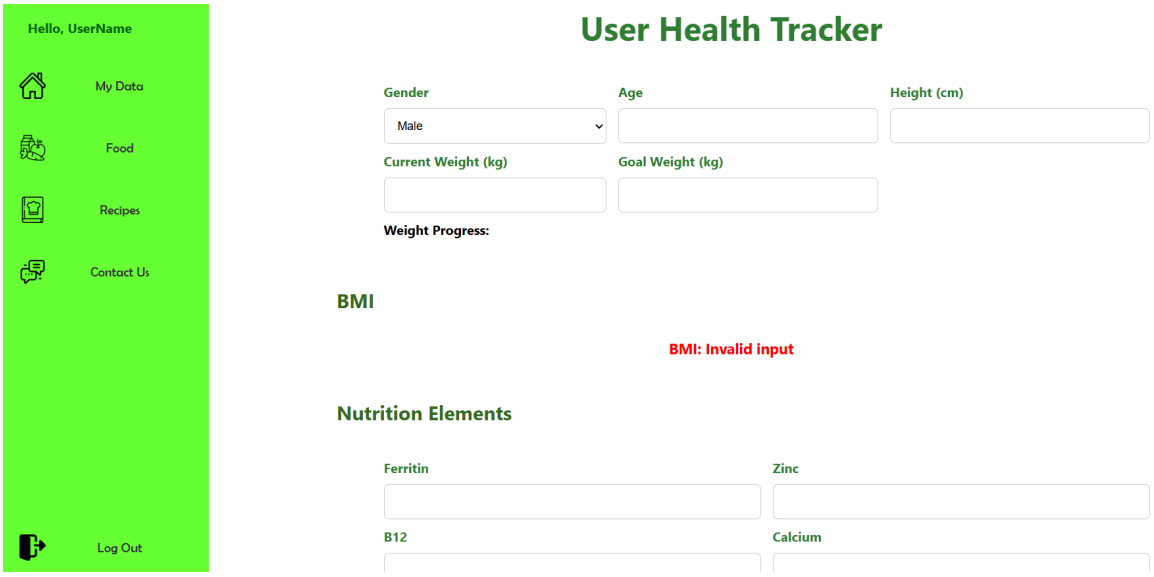
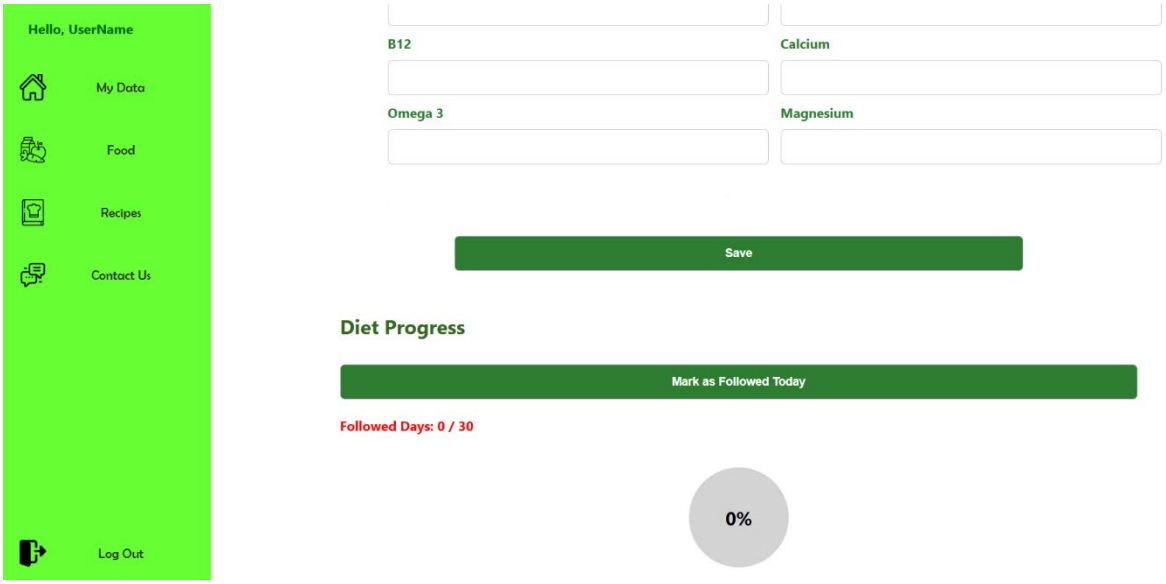


Figure 50: My Data Page Overview Part 1



`

Figure 51:My Data Page Overview Part 2

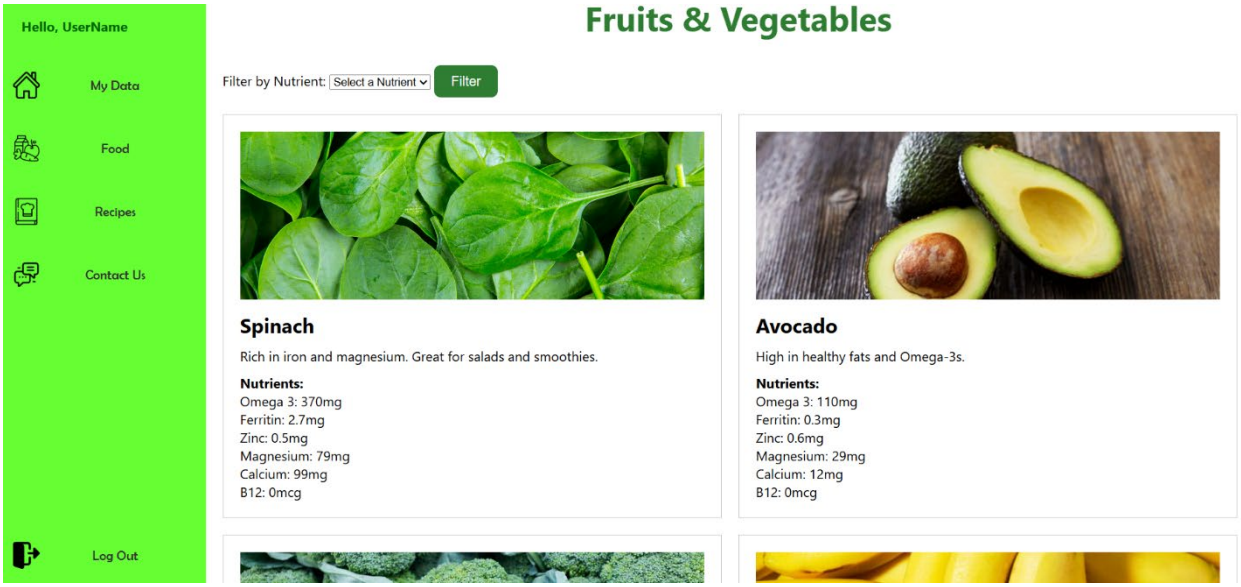


Figure 52:The Food Page Overview

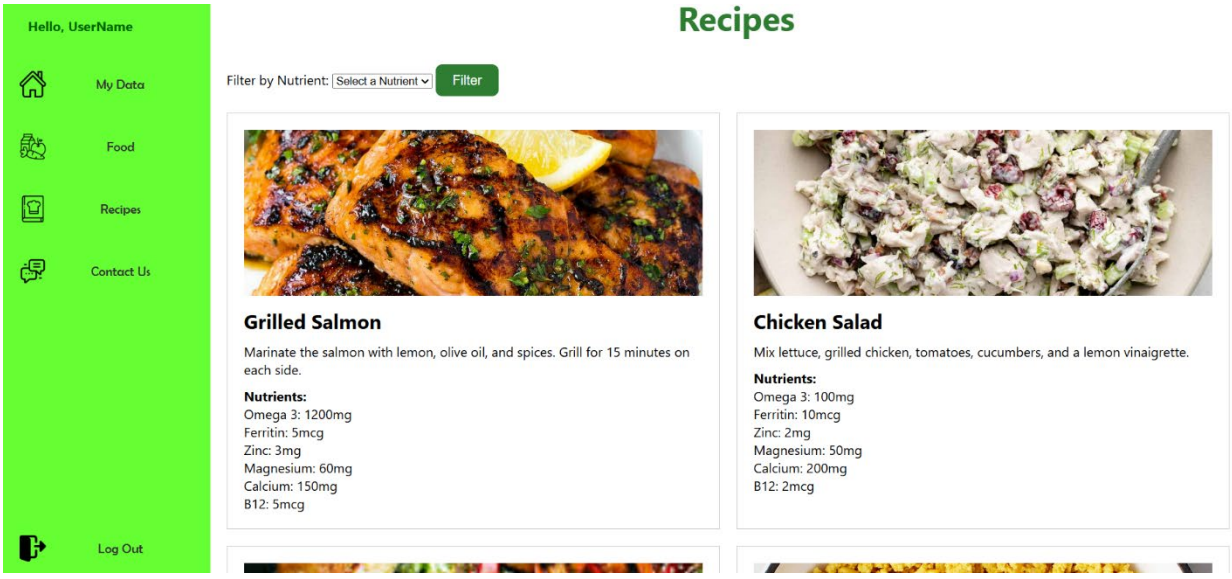


Figure 53:The Recipes Page Overview

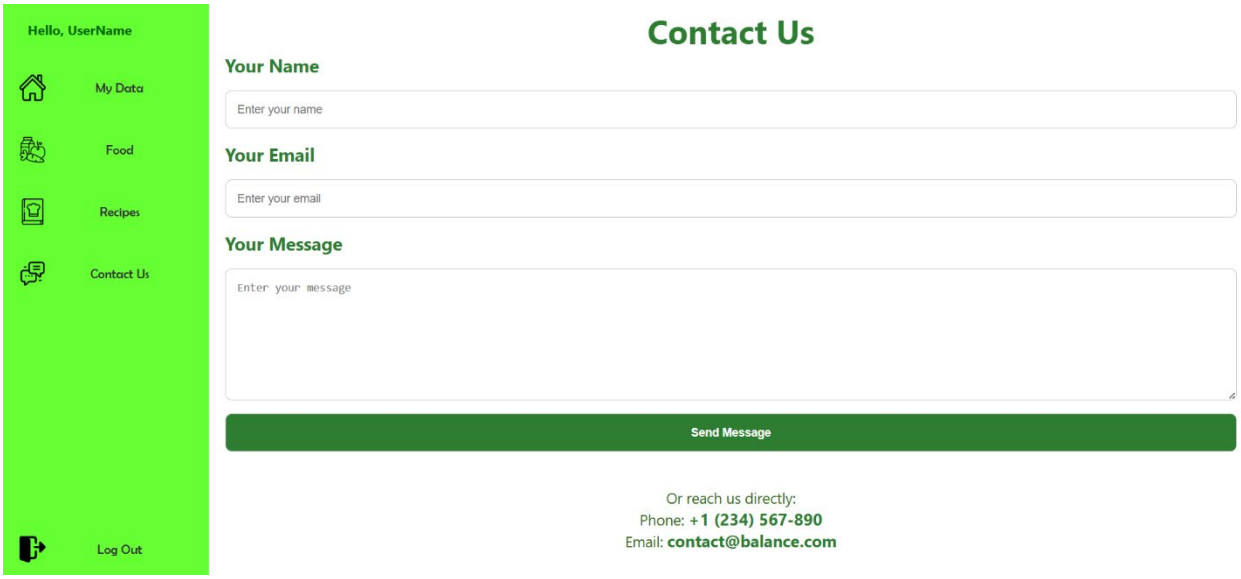


Figure 54:The Contact Page Overview

5.1.3 User Manual

Accessing the Website

- Open a browser and enter the official Balance website.
- The first thing the user will see is the **Login Page**.

Registering a New Account

Steps:

1. On the **Login Page**, click the **"Register"** link.
2. Enter:
    - A unique **Username**
    - A **Password**
    - **Confirm Password**
3. Click the **"Register"** button.

- If passwords match and the username is unique:
    - You will see "**Registration Successful**" and be redirected to the **Login Page**.
- If inputs are invalid:
    - You will see one of the following error messages:
        - "**Username already exists**"
        - "**Passwords do not match.**"
        - "**Fields cannot be empty**"

Logging In

Steps:

1. On the **Login Page**, enter:
    - Your **Username**
    - Your **Password**
2. Click the **"Login"** button.

- On correct credentials:

- You are directed to the **My Data Page**.

- On incorrect credentials:
  - A red error message appears: "**Wrong Username or Password**".

My Data Page

First-Time Users:

- You must enter personal data:
  - Age
  - Gender
  - Height
  - Current Weight
  - Goal Weight
  - Nutritional Levels (Ferritin, B12, etc.)
- Click **"Save"** to submit.

Features:

- View **BMI Calculation**
- View **Progress Circle** based on diet adherence
- Nutrition status color indicator (green: normal, red: low)

Recipes Page

- Browse healthy recipes based on your needs.
- Filter by nutrients (e.g., Calcium-rich recipes) by choosing from the dropdown list and clicking the filter button.
- User will see :
  - Ingredients
  - Nutrient breakdown
  - Instructions

Food Page

- Browse fruits and vegetables rich in specific nutrients.
- Filter by nutrients (e.g., Calcium-rich recipes) by choosing from the dropdown list and clicking the filter button.
- The user will see:
    - Nutrient breakdown
    - notes

Contact Page

- Fill in your:
    - Name
    - Email
    - Message
- Click **“Send Message”** for help or complaints.
- Alternatively, click on contact numbers/emails for direct communication.

Logging Out

- Click the **“Log Out”** button on any page to end your session.
- You will be redirected back to the **Login Page**.

Troubleshooting

Problem	Solution
Can’t register	Ensure all fields are filled, passwords match, and the username is unique
Can’t log in	Recheck credentials, ensure Caps Lock is off
Page not loading	Check your internet connection

Security & Privacy

- All user data is securely stored and encrypted.
- Sessions are protected and automatically logged out after inactivity.

Support

For technical issues, contact the support team via the **Contact Page** or directly by email/phone.

6.0 Chapter six

6.1 Test cases

This test suite was developed using MSTest, a popular unit testing framework for .NET, to validate the key functionalities of the mainclass used in a health tracking web application. The class handles user login, registration, and database operations related to personal data, nutrition data, and diet progress tracking.

The primary goals of these unit tests are:

1.To **verify the correctness** of login and registration logic.

2.To **ensure database write and read consistency** for personal and nutritional user data.

3. To validate the tracking mechanism for user follow-up days and last interaction date.

Test Cases Included:

1. Login Tests

Login_ValidCredentials_ReturnsUserName :

Verifies that a valid username and password return the correct username.

Login_InvalidCredentials_ReturnsError :

Ensures that invalid login attempts return the string "Error".

2. Account Creation Test

CreateAccount_ExistingUser_ReturnsFalse :

Tests that the function prevents registering an already existing username.

3. Personal Data Test

SavePersonalData_AndRetrieve_AreConsistent

Saves mock data for gender, age, height, weight, and goal weight.

Retrieves the data to check consistency.

4. Nutrition Data Test

SaveNutritionData_AndRetrieve_AreConsistent

Saves and reads values for ferritin, zinc, vitamin B12, calcium, omega-3, and magnesium.

Confirms that the values stored match those retrieved.

Assumptions:

The database contains a test user with credentials:

Username = "testuser"

Password = "testpassword"

Tables such as users, UserPersonalData, and UserNutritionData exist and are structured properly.

Test data is isolated and doesn't affect production values.

6.1.1 Significant Parts in testing code

HealthAppTests | MainClassTests | TestLogin_ValidUser_ReturnsUsername

```
Imports Microsoft.VisualStudio.TestTools.UnitTesting
Imports advSoftware


<TestClass>
0 references
Public Class MainClassTests

    <TestMethod>
    0 references
    Public Sub TestLogin_ValidUser_ReturnsUsername()
        ' Arrange
        Dim main As New mainclass()
        Dim username As String = "testuser"
        Dim password As String = "password"

        ' Act
        Dim result As String = main.login(username, password)

        ' Assert - using full namespace to avoid ambiguous reference
        Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual(username, result)
    End Sub

    <TestMethod>
    0 references
    Public Sub TestLogin_InvalidUser_ReturnsError()
        ' Arrange
        Dim main As New mainclass()
        Dim username As String = "invaliduser"
        Dim password As String = "wrongpassword"

        ' Act
        Dim result As String = main.login(username, password)

        ' Assert
        Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual("Error", result)
    End Sub

End Class
```

Figure 55: The code for the testing part 1

```vb
0 references
Public Class MainClassTest

    Dim mc As New mainclass()


    <TestMethod()>
    0 references
    Public Sub TestLogin_Failure()
        Dim result = mc.login("wronguser", "wrongpassword")
        Assert.AreEqual("Error", result)
    End Sub

    <TestMethod()>
    0 references
    Public Sub TestCreateAccount_AlreadyExists()
        Dim result = mc.createacc("testuser", "1234", "1234")
        Assert.IsFalse(result) ' testuser already exists
    End Sub

    <TestMethod()>
    0 references
    Public Sub TestSavePersonalData()
        mc.SavePersonalData("testuser", "Female", 25, 160D, 60D, 55D)
        Dim dt As DataTable = mc.GetPersonalData("testuser")
        Assert.AreEqual(1, dt.Rows.Count)
        Assert.AreEqual(25, dt.Rows(0)("Age"))
    End Sub

    <TestMethod()>
    0 references
    Public Sub TestSaveNutritionData()
        mc.SaveNutritionData("testuser", 30.5D, 12.3D, 450D, 900D, 1.1D, 300D)
        Dim dt As DataTable = mc.GetNutritionData("testuser")
        Assert.AreEqual(1, dt.Rows.Count)
        Assert.AreEqual(30.5D, dt.Rows(0)("Ferritin"))
    End Sub


End Class
```

Figure 56: The code for the testing part2

6.1.2 The results of testing

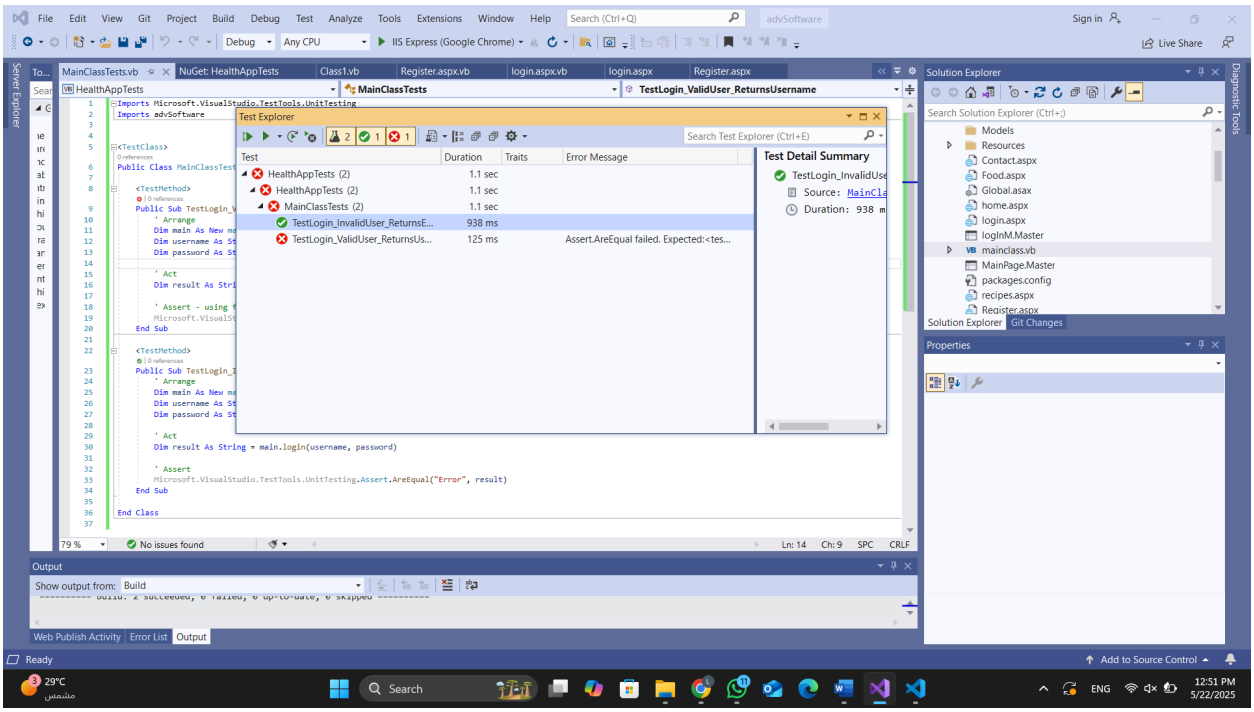


Figure 57:The result of first test

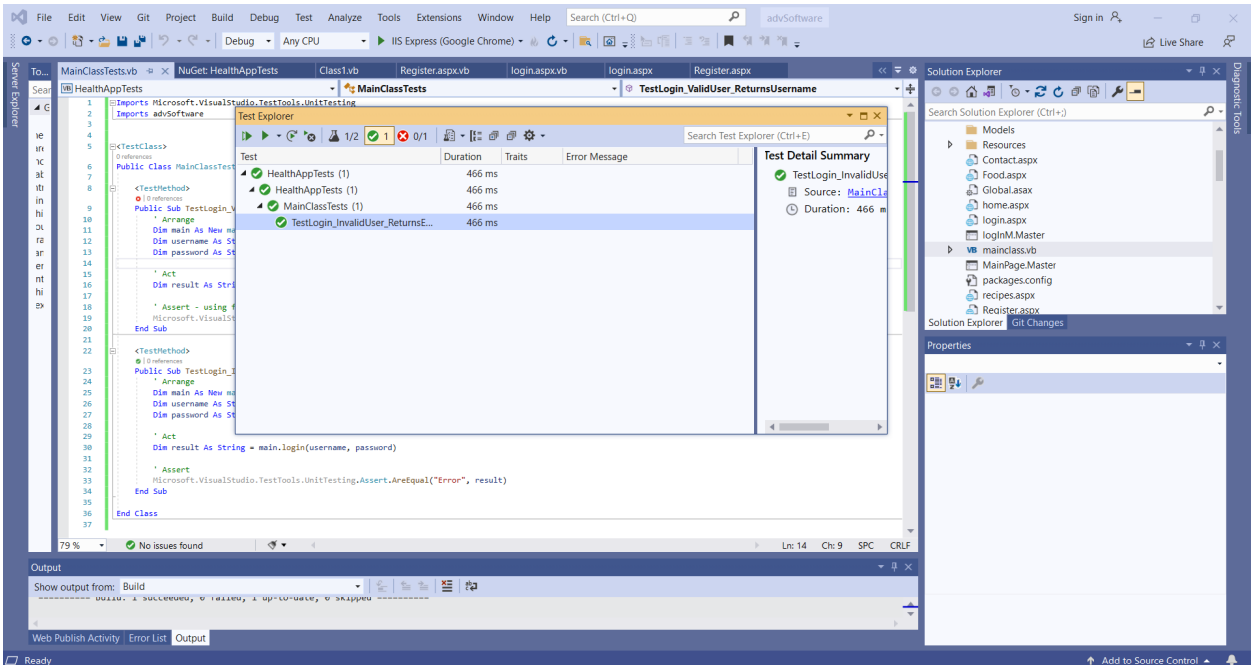


Figure 58:The final testing results part1

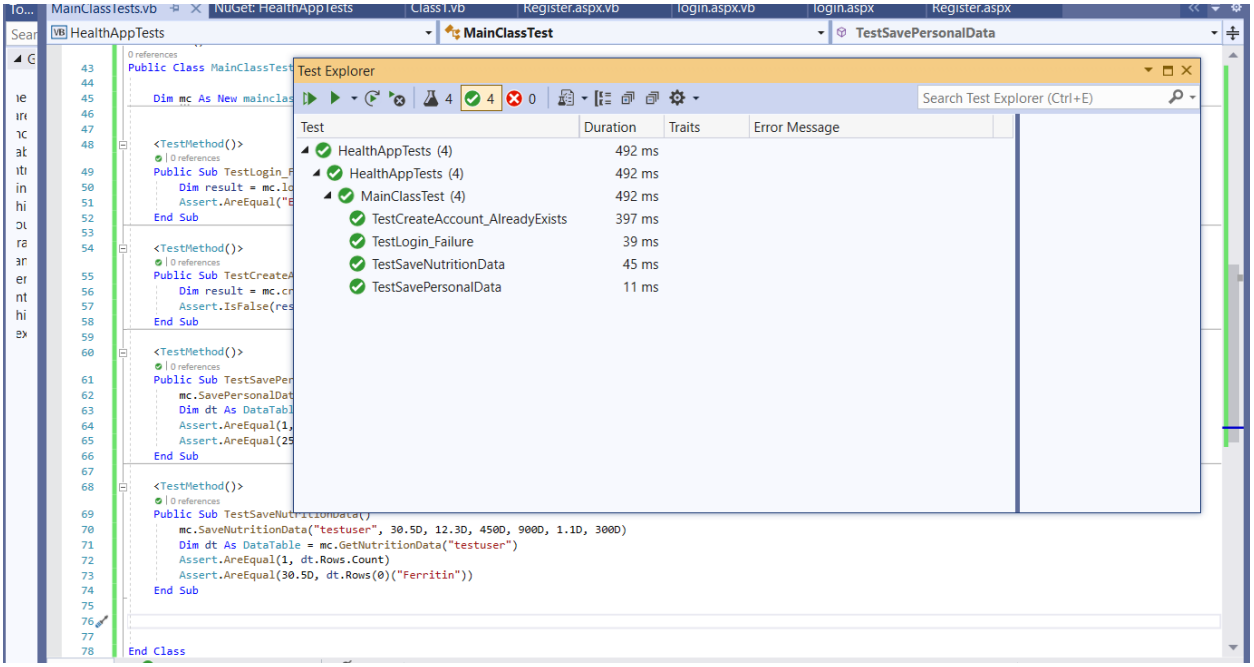


Figure 59:The final testing results part2

References

Sommerville, Ian, Software engineering, 10th edition, 2015, Welly.

Hudaib, Amjad, Lecture Notes of Advanced Software Engineering Course, Second Semester, 2024.

Appendix

Appendix A: System Features Overview

Feature	Description
User Registration	**Secure user signup with validation of username and password confirmation**
User Login	**Authenticates existing users and directs them to their personalized dashboard**
Personal Data Entry	**Allows users to input height, weight, age, gender, and dietary goals**
Nutrition Tracking	**Inputs and tracks nutrients like Ferritin, B12, Omega-3, etc.**
Recipe Recommendation	**Suggests healthy recipes based on user's dietary needs**
Filtering Food Options	**Filters foods and recipes based on selected nutritional preferences**
Progress Tracking	**Displays a visual progress indicator based on dietary adherence**
Contact Page	**Enables users to send feedback or questions to the support team**
Logout Functionality	**Ends the session and returns the user to the login screen**

Appendix B: Technologies Used

Component	Technology
Frontend	**HTML, CSS**
Backend	**VB.NET**
Database	**SQL Server Management Studio (SSMS)**
Development Tool	**Visual Studio 2019**
Testing Framework	**MSTest**

Appendix C: Use Case Mapping

Component	Related Functional Requirements
Registration	**FR1, FR6, FR7, FR8, FR9, FR10**
Login	**FR1, FR2, FR4, FR5**
Data Validation	**FR4, FR5, FR7, FR8, FR10, FR12, FR16**
Nutrition & Personal Data	**FR11, FR12, FR13, FR14, FR15, FR18**
Progress Tracking	**FR17, FR19, FR20, FR21**
Recipes & Food Filter	**FR22, FR23, FR24, FR25, FR26, FR27**
Contact System	**FR29, FR30, FR31**
Logout	**FR28**

Appendix D: Testing Summary

Testing Framework: MSTest

Key Tests:

- Valid and invalid login

- Existing user registration check

- Data consistency in nutrition and personal entries

Result: All core functions passed the defined test cases successfully.