BIW20404: WEB SERVICE TECHNOLOGY GROUP PROJECT

PERSONAL HEALTH & FITNESS CALCULATOR

BY GROUP 1

GROUP MEMBERS

NO	NAME	MATRIC NO
1.	ARUN KUMAR A/L MANIMARAN	CI230128
2.	KHOR HWEE SHIAN	CI230085
3.	SITI NUR ESYA BINTI BUWANG	AI220174
4.	YUVANESH A/L SHANMUGAM	CI230123
5.	VIVETHAN A/L MURUGAN	CI230090
6.	KUGAAN RAAJ A/L GOBALAN	CI230042

Table of Contents

Abstract

This group project is about building a Personal Health and Fitness Calculator—a simple, user-friendly tool designed to help people better understand and manage their health. The calculator brings together five core features: a BMI (Body Mass Index) Calculator to check if your weight is in a healthy range, a Body Fat Percentage Calculator to estimate how much of your body is made up of fat, a Calories Burn Rate Calculator to show how many calories you burn during different activities, a BMR (Basal Metabolic Rate) Calculator to figure out how many calories your body needs just to function at rest, and a Lean Body Mass (LBM) Calculator to help you understand the weight of everything in your body except fat.

All of these modules work together to give users a clear, personalized snapshot of their current health and fitness level. The goal is to make it easier for anyone, no matter their experience level to track progress, make informed decisions, and stay motivated on their health journey. In the future, we hope to add features like syncing with fitness trackers, progress charts, and customized tips to make the experience even more helpful.

1.0 Project Introduction

1.1 Project Background

These days, more people are trying to take better care of their health but finding the right tools to track things like fitness, weight or calories can be confusing. Lots of apps only focus on one thing or its too complicated to use. This is what inspired us to build this fitness calculator, a simple tool that gives users health insights based on their own personal data. This project was part of our course, where we had the chance to put what we've learned about SOAP into a system. Our calculator takes basic info from user like name, ID number, gender, weight and height to use that to calculate important health stats. We built three core modules that were required which are BMI (Body Mass Index) , Body Fat Percentage and Calories Burn Rate. Not only that, we added two extra modules BMR(Basal Metabolic Rate) and LBM (Lean Body Max).

This system also figures out the user's age automatically from their ID, checks for errors in their inputs and uses SOAP fault messages to let them know if something goes wrong. This project helped us strengthen our technical skills while creating something that's actually useful and maybe even a little motivating for people trying to improve their health.

1.2 Project Goals

i. Design a SOAP-based web service to manage user information, perform health-related calculations and handle input/output processing efficiently and securely.

ii. Develop a personal health and fitness calculator that allows user to access key health modules including BMI, Body Fat Percentage, Calories Burn Rate, Basal Metabolic Rate and Lean Body Mass while calculating age based on ID input.

iii. Create user-friendly client application that communicates with the web service to collect inputs, display result and guide users through each module clearly and interactively.

iv. Test the developed system thoroughly to ensure al modules work correctly in real use cases with a focus on providing accurate results to improve user experience.

1.3 Project Scope

The Personal Health and Fitness Calculator is a SOAP-based web service that helps users calculate and understand important health metrics. It is designed for general users who want to track their fitness and body composition using accurate personalized data. The system is simple and accepts inputs like name, ID number, gender, weight, height.

User Category	Module	Explanation
General User	User Information Display Module	Collect and display basic user information such as name, ID number, gender, weight, height and age where its automatically calculated from ID.
	Body Mass Index (BMI) module	Calculate BMI based on weight and height, and classify user into health categories which are underweight, normal, overweight and obese.
	Body Fat Percentage Module	Estimate body fat percentage using gender, age and body measurements to provide insights into body composition.
	Calories Burn Rate module	Calculate estimated number of calories burned during physical activities based on user's weight and activity duration/intensity.
	Basal Metabolic Rate (BMR) module	Determine number of calories the user's body needs at rest to maintain essential bodily functions.
	Lean Body Mass (LBM) module	Calculate fat-free portion of body weight to help users understand muscle mass and other non-fat components.
	Error Handling & Input Validation Module	Validate all user input and handle incorrect or missing data using SOAP fault message to ensure a smooth user experience.

Table 1.3.1 Module Explanation

1.4 Significance of the Project

This project offers several key benefits for users and developers of the Personal Health and Fitness Calculator:

i. Personalized Health Insights: Users receive tailored feedback based on their individual data, helping them to understand their body composition, calorie needs and fitness level more accurately.

ii. Improved Health Awareness: System encourages users to take a more active role in monitoring and managing their health.

iii. Time Saving and User Friendly: Users can quickly calculate health indicators without needing multiple apps or websites by streamlining process.

iv. Educational and Practical Value: For developers, this project strengthens skills in SOAP web services, XML communication and client-server interaction valuable knowledge for real-world applications.

2.0 Project Description

2.1 Overview of the Web Service

The Personal Health and Fitness Calculator is a SOAP-based web service designed to provide users with personalized health metrics through a simple and interactive client application

The web service offers five core health calculation modules:

- BMI(Body Mass Index) calculator - Calculates BMI based on weight and height, classifies the user into health categories and provides relevant health tips.
- Body Fat Percentage (BFP) Calculator - Estimates body fat percentage using age, gender and BMI. Provides classifications and personalized advice.
- Calories Burn Rate (CBR) Calculator - Estimates calories burned during physical activities based on weight, duration, and intensity (MET values).
- Basal Metabolic Rate (BMR) Calculator - Determines the nuber of calories needed at rest using weight, height, age and gender.
- Lean Body Mass (LBM) Calculator - Calculates the fat-free portion of the body using gender, height and weight. Provides classifications with fitness advice.

Each module includes robust input validation and uses SOAP fault messages to handle invalid inputs. The web service is fully described using WSDL (Web Services Description Language) and supports UDDI (Universal Description, Discovery and Integration) to enable easy discovery and integration with client applications.

2.2 Overview of the Client Application

The client application was built with a user-friendly graphical interface that allows users to :

- Enter personal information (name, ID, gender, weight, height).
- Automatically calculate age from the ID number.
- Access an interact with all five health calculation modules.

- View results in an understandable format with health recommendations.
- Reset or cancel operations easily through intuitive controls.

The client communicates with web service using SOAP messages, sending inputs and receiving structured responses for display. The application ensures a smooth user experience by validating inputs and guiding users step-by-step through each module.

2.3 Explaination of Modules Developed

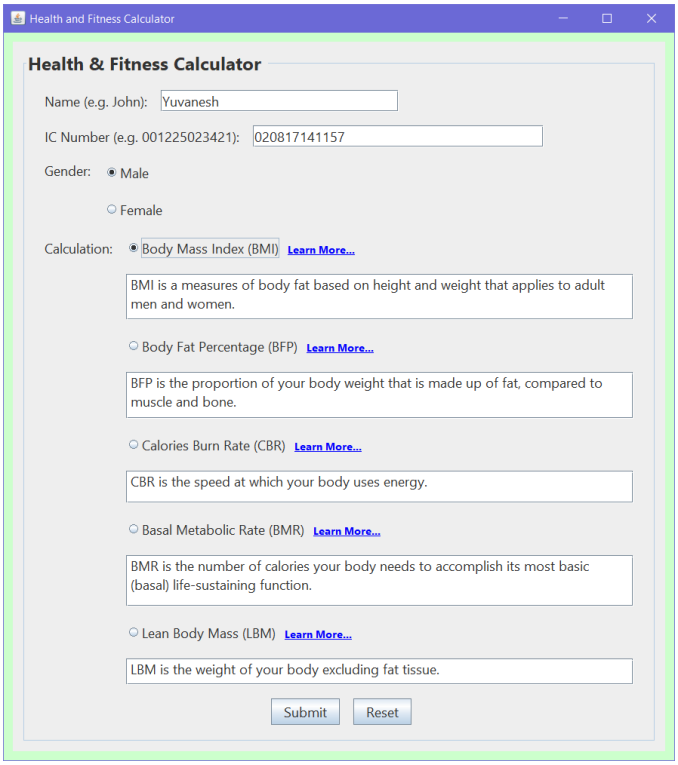

Figure 2.2.1 Main Page

The Personal Health & Fitness Calculator web service consists of five key health calculation modules, each designed to provide users with accurate and actionable health metrics. These modules were developed using scientifically validated formulas and incorporate robust input validation to ensure reliable results.

To determine a user's age based on their identification number, the system takes the first six digits of the IC number, which represent the user's birth date in the format YYMMDD. For example, if the user enters the IC number 020817141157, the birth date is extracted as 2002/08/17. The system then calculates the age by comparing this birth date with the current date, such as 2025/06/11. On the next page, the system will display the user's personal information, including their name, IC number, and calculated age.

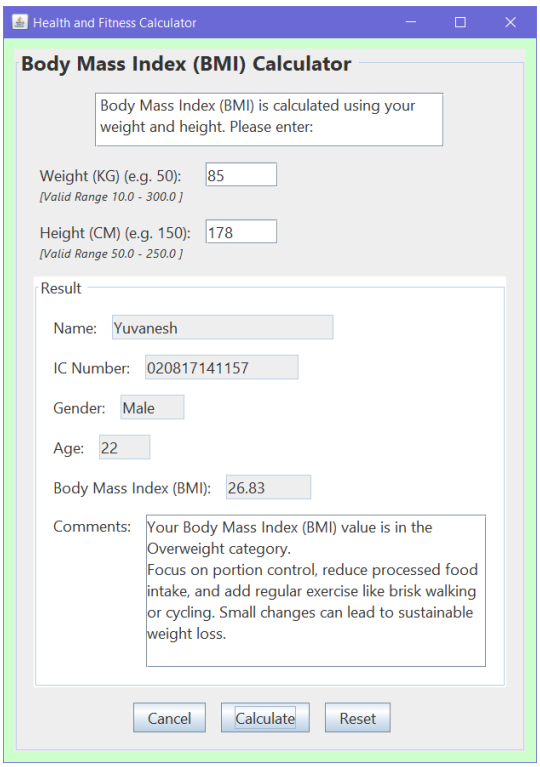

Figure 2.2.2 Body Mass Index (BMI) Calculator

The Body Mass Index (BMI) Calculator module calculates a user's BMI based on their weight and height, then classifies the result into standard health categories (Underweight, Normal, Overweight, or Obese). It includes validation to ensure weight and height inputs fall within reasonable ranges (0.1-500 kg for weight and 30-300 cm for height). The module also provides personalized health recommendations based on the BMI classification, helping users understand their results and take appropriate action.

To calculate BMI, we use the formula: BMI = weight ÷ height². For example, if a person weighs 85 kilograms and is 178 centimeters tall, we first convert the height into meters: 178

cm is 1.78 meters. Then, we square the height: 1.78 × 1.78 = 3.1684. Next, we divide the weight by the squared height: 85 ÷ 3.1684 ≈ 26.83. So, the BMI is about 26.83, which falls in the overweight category according to the BMI chart. We take reference from https://www.calculatorsoup.com/calculators/health/bmi-calculator.php.

This is a BMI calculator where users enter their weight and height to get their BMI score. It shows the result (like "Overweight") and gives health tips, such as eating less processed food and exercising. It also displays personal details like name, age, and gender, with buttons to calculate, reset, or cancel.

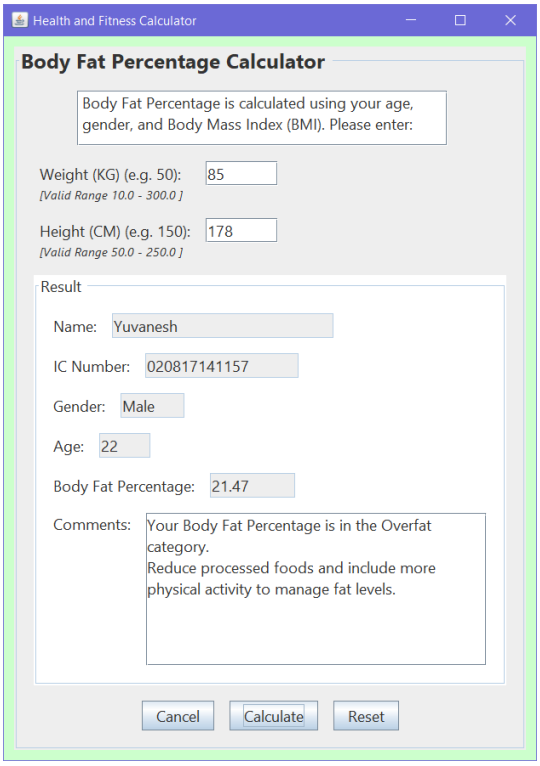

Figure 2.2.3 Body Fat Percentage Calculator

For body composition analysis, the Body Fat Percentage (BFP) Calculator estimates fat mass using BMI, age, and gender-specific formulas. This module applies different healthy ranges for males and females, classifying results as Underfat, Healthy, Overfat, or Obese. The calculation uses the Deurenberg equation, which has been validated in clinical studies. Input validation ensures the BMI value falls within a realistic range (10-60) before calculation.

To calculate body fat percentage, we use different formulas for men and women. For men, the formula is: (1.20 × BMI) + (0.23 × Age) – 16.2, and for women: (1.20 × BMI) + (0.23 × Age) – 5.4. For example, a 30-year-old man who weighs 85 kg and is 178 cm tall has a BMI of 26.8. Using the formula: (1.20 × 26.8) + (0.23 × 30) – 16.2 = 22.86% body fat. This helps estimate how much of a person's weight is from fat. We take reference from https://www.tgfitness.com/body-fat-percentage-calculator/#:~:text=The%20formula%20uses%20a%20person's,0.23%20x%20Age)%20%E2%80%93%205.4.

This is a body fat percentage calculator that estimates fat levels using weight, height, age, and gender. It displays the result (e.g., "Overfat") along with personalized advice like eating healthier and exercising. The user's details (name, ID, age, etc.) are shown, and buttons allow them to calculate, reset, or cancel.

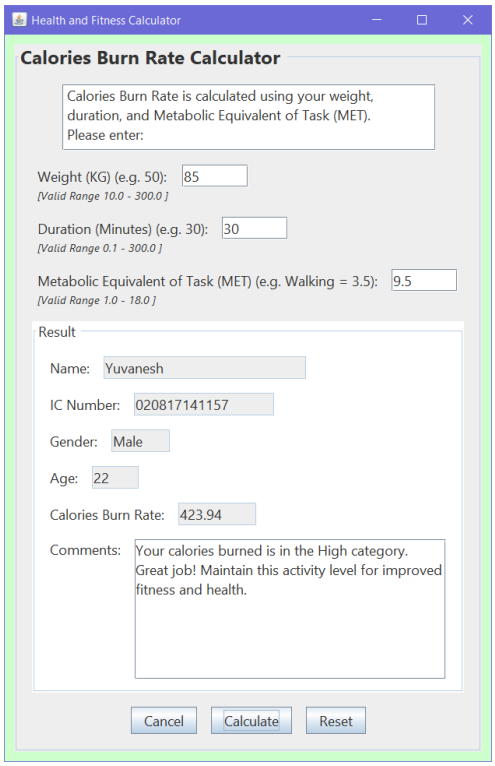

Figure 2.2.4 Calories Burn Rate Calculator

The Calories Burn Rate Calculator helps users estimate energy expenditure during physical activities. Using the Metabolic Equivalent of Task (MET) value, weight, and exercise duration,

it calculates total calories burned. The module includes comprehensive validation for all inputs and categorizes results by intensity level (Low to Very High), with suggestions for optimizing workouts. This provides valuable feedback for users tracking their fitness progress.

To calculate how many calories a person burns, we use the formula: Calories = Time × MET × Body Weight ÷ 200. For example, if someone weighs 85 kg, exercises for 30 minutes, and the activity has a MET value of 9.5, the calculation would be: 30 × 9.5 × 85 ÷ 200 = 121.13 calories burned. This means the person would burn about 121 calories during that 30-minute activity. We take reference from https://www.calculator.net/calories-burned-calculator.html.

This is a Calories Burn Rate calculator that estimates energy expenditure using weight, activity duration, and intensity (MET). It shows the calories burned (e.g., "Very High") with positive feedback like "Excellent work!" and reminders to stay hydrated and eat well. The user's personal details are displayed, and action buttons allow calculating, resetting, or canceling the operation.

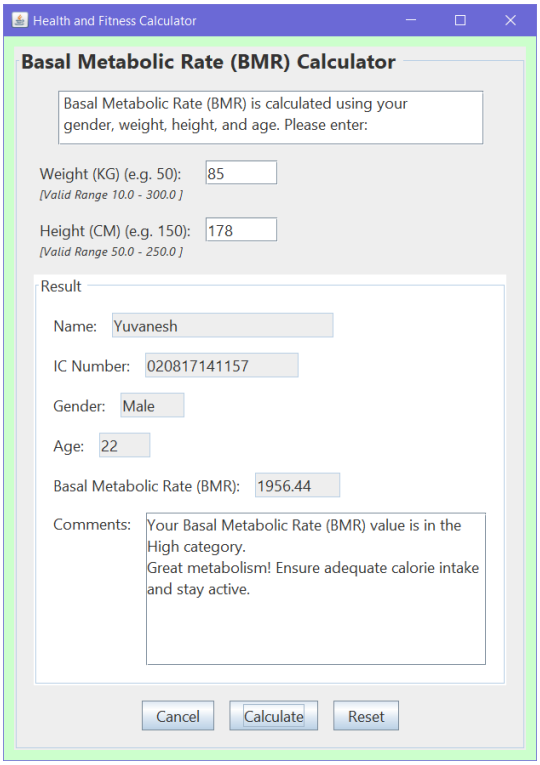

Figure 2.2.5 Basal Metabolic Rate (BMR) Calculator

To assess metabolic needs, the Basal Metabolic Rate (BMR) Calculator determines the calories required for basic bodily functions at rest. Using the Mifflin-St Jeor equation - currently considered the most accurate BMR formula - it accounts for weight, height, age, and gender differences. The results are classified into metabolic rate categories with corresponding nutritional advice, helping users understand their daily calorie requirements.

To calculate BMR (Basal Metabolic Rate), we use different formulas for men and women. For men, the formula is: 66.47 + (13.75 × weight in kg) + (5.003 × height in cm) − (6.755 × age in years). For example, a 30-year-old man who weighs 85 kg and is 178 cm tall would calculate like this: 66.47 + (13.75 × 85) + (5.003 × 178) − (6.755 × 30) = 66.47 + 1168.75 + 891.53 − 202.65 = 1924.1 calories. This means his body burns about 1924 calories per day at rest. We take reference from https://www.diabetes.co.uk/bmr-calculator.html.

This is a BMR calculator that estimates your body's calorie needs at rest using weight, height, age, and gender. It shows the result (e.g., "High" BMR) with feedback like "Great metabolism!" and tips to stay active. The user's details (name, ID, etc.) are displayed, and buttons allow calculating, resetting, or canceling.

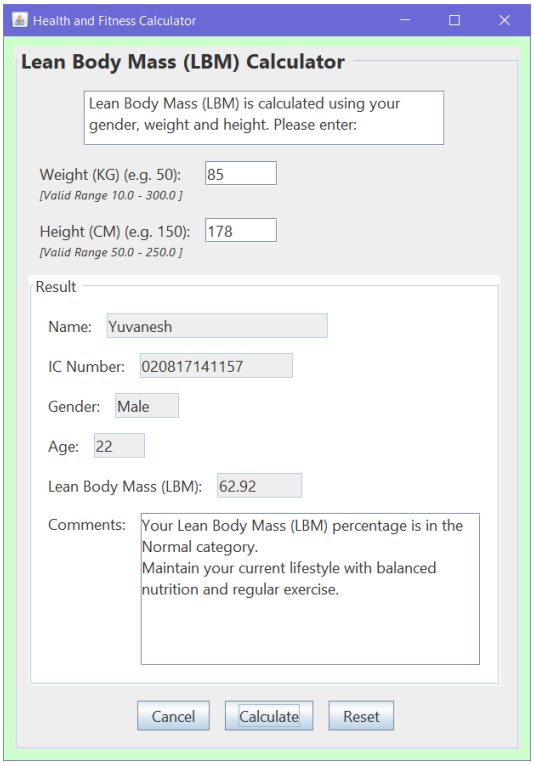

Figure 2.2.6 Lean Body Mass (LBM) Calculator

The Lean Body Mass (LBM) Calculator estimates fat-free mass (muscles, bones, organs) using the Boer formula, which is particularly accurate for the general population. It calculates LBM based on weight, height and gender, then presents the result as a percentage of total body weight. The module includes validation to ensure physiologically possible results and provides training recommendations based on whether the LBM percentage is Low, Normal, or High. We take reference from https://gymnation.com/fitness-calculators/lean-body-mass-calculator/.

This is a Lean Body Mass (LBM) calculator that estimates muscle and bone weight using gender, height, and weight. It displays the result (e.g., "Normal" LBM) with advice to maintain a healthy lifestyle. The user's details (name, ID, etc.) are shown in a table format, and buttons allow calculating, resetting, or canceling the operation.

To summarize, all modules feature comprehensive input validation and return SOAP fault messages for invalid entries, ensuring data integrity. They provide not just numerical results but also clear classifications and practical health recommendations, making the service both

informative and actionable for users. The calculations use precise formulas from established medical and fitness research, with results rounded to two decimal places for consistency. This combination of accurate metrics and user-friendly interpretation makes the web service a valuable tool for health monitoring and fitness planning.

2.4 Explanation of SOAP Messaging, WSDL, and UDDI Used

2.3.1 SOAP (Simple Object Access Protocol)

(a) Body Mass Index (BMI) Calculator and Categories

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" 
   <soapenv:Header/>
   <soapenv:Body>
      <heal:calcBMIValue>
         <weight>90</weight>
         <height>170</height>
      </heal:calcBMIValue>
   </soapenv:Body>
</soapenv:Envelope>
```

Figure 2.3.1.7 SOAP Request for BMI Calculator

The SOAP request is in XML message enclosed in a **<soapenv:Envelope>**, which defines the SOAP namespace. The optional **<soapenv:Header>** is empty, which shows that there are no additional metadata such as authentication. The main part of the request is in the <soapenv:Body>, which contains the **<heal:calcBMIValue>** method call. This method requires 2 inputs which is **<weight>** and **<height>**. These measurements are important inputs for the server to perform the body mass index calculation accurately.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:calcBMIValueResponse xmlns:ns2="http://healthwebservice.khor.com/">
         <return>31.14</return>
      </ns2:calcBMIValueResponse>
   </S:Body>
</S:Envelope>
```

Figure 2.3.1.8 SOAP Response for BMI Calculator

The soap response for Body Mass Index has the **<S:Body>** tag which contains the **<ns2:calcBMIValueResponse>**, where the xmlns:ns2 attribute specifies the namespace for the healthcare service (http://healthwebservice.khor.com/). The **<return>** tag represents the computed BMI value based on the user input that is derived from the formula of the BMI. This level of BMI value suggests the patient falls into a certain category and highlights the service's role in providing actionable health metrics.

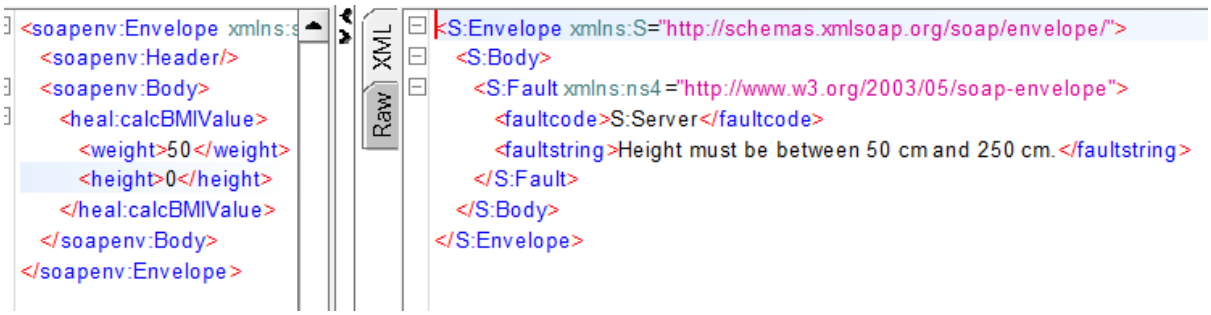

Figure 2.3.1.9 SOAP Fault Message Error BMI Calculator

In BMI calculator methods, The SOAP fault will handle error if the user enter wrong weight and height. For example. If user enter height as 0, system will return “Height must be between 50 cm and 250 cm.”

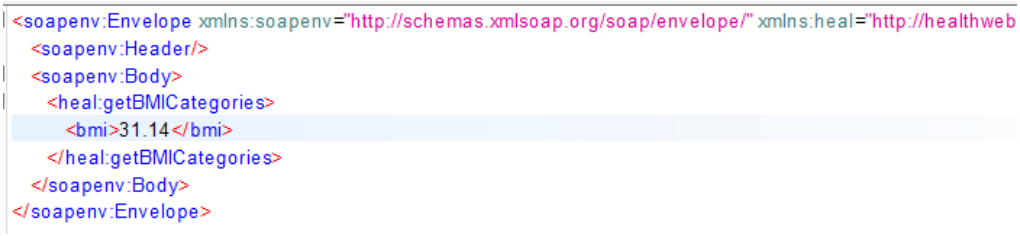

Figure 2.3.1.10 SOAP Request for BMI Categories

The request begins with the optional **<soapenv:Header>** which is empty, indicating no additional metadata such as authentication or session details are included. This SOAP request resides in the **<soapenv:Body>**, which contains the **<heal:getBMICategories>** method call. This method takes a single parameter, **<bmi>** , with the value that is calculated according to the user input by getting passed on calcBMIValue() method. This structured

XML format ensures the server receives the necessary data to determine the corresponding BMI category.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:getBMICategoriesResponse xmlns:ns2="http://healthwebservice.kh
         <return>Your Body Mass Index (BMI) value is in the Obesity categ
Consider a comprehensive lifestyle change. Adopt a healthy, low-calorie d
      </ns2:getBMICategoriesResponse>
   </S:Body>
</S:Envelope>
```

Figure 2.3.1.11 SOAP Response for BMI Categories

The response also starts in the **<S:Body>** which contains the **<ns2:getBMTCategoriesResponse>**. The **<return>** tag provides the result of a human-readable message classifying the Body Mass Index value that is calculated based on the user input into a certain category and recommending lifestyle changes to the users. The response from the server is shortened, but it clearly communicates health advice based on the Body Mass Index calculation for specific individuals.

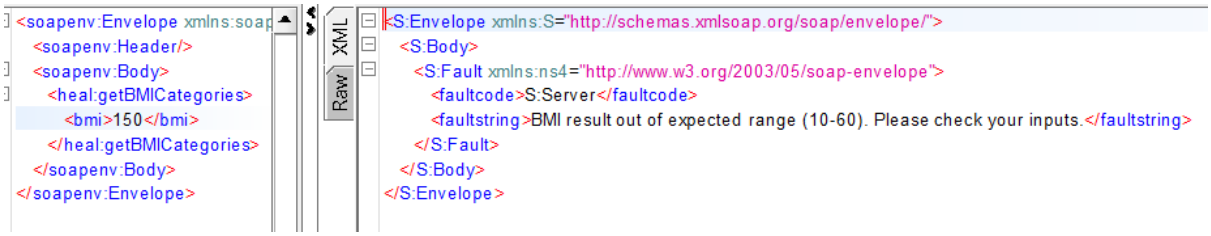

Figure 2.3.1.12 SOAP Fault Message Error for BMI Categories

In BMI categories methods, The SOAP fault will handle error if the user enter wrong bmi. For example. If user enter bmi as 150, system will return “BMI result out of expected range (10-60). Please check your inputs.”

(b) Body Fat Percentage Calculator and Categories

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" x
   <soapenv:Header/>
   <soapenv:Body>
      <heal:calcBFPValue>
         <age>22</age>
         <!--Optional:-->
         <gender>male</gender>
         <bmi>27.8</bmi>
      </heal:calcBFPValue>
   </soapenv:Body>
</soapenv:Envelope>
```

Figure 2.3.1.13 SOAP Request for BFP Calculator

The SOAP request is in an XML message enclosed in a **<soapenv:Envelope>**, which defines the SOAP namespace. The optional **<soapenv:Header>** is empty, which shows that there is no additional metadata such as authentication. The main part of the request is in the **<soapenv:Body>**, which contains the **<heel:calcBFPValue>** method call. This method requires 3 inputs which are **<age>**, **<gender>** and **<bmi>**. These measurements are important inputs for the server to perform the body mass index calculation accurately.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:calcBFPValueResponse xmlns:ns2="http://healthwebservice.khor.com/">
         <return>22.82</return>
      </ns2:calcBFPValueResponse>
   </S:Body>
</S:Envelope>
```

Figure 2.3.1.14 SOAP Response for BFP Calculator

The soap response for Body Fat Percentage(BFP) has the **<S:Body>** tag which contains the **<ns2:calcBFPValueResponse>**, where the xmlns:ns2 attribute specifies the namespace for the healthcare service (http://healthwebservice.khor.com/). The **<return>** tag represents the computed BFP based on the user input that is derived from the formula of the Body Fat Percentage. This level of Body Fat Percentage value suggests the patient falls into a certain category and highlights the service's role in providing actionable health metrics.

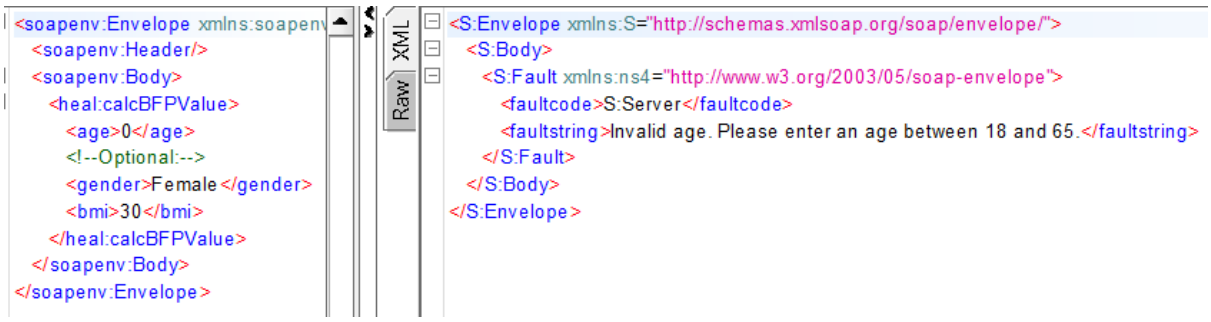

Figure 2.3.1.15 SOAP Fault Message Error for BFP Calculator

In BFP calculator methods, The SOAP fault will handle error if the user enter wrong age, gender and bmi. For example. If user enter age as 0, system will return “Invalid age. Please enter an age between 18 and 65.”

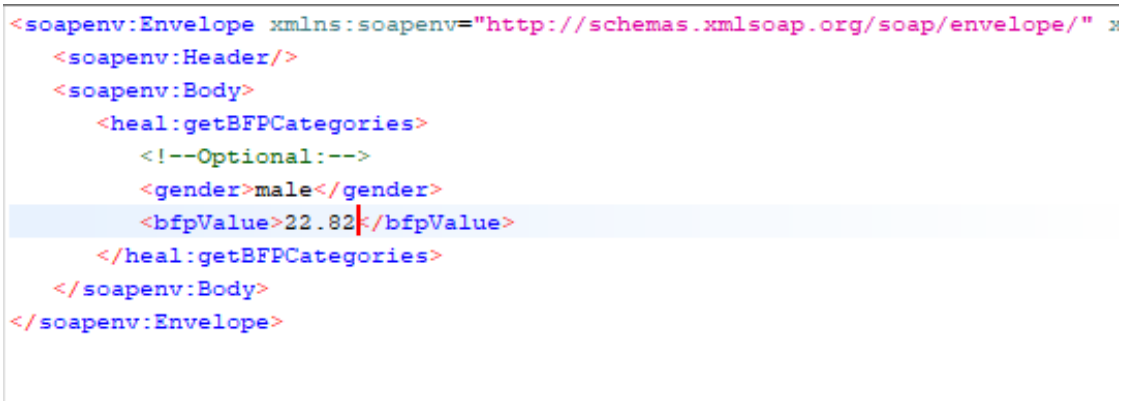

Figure 2.3.1.16 SOAP Request for BFP Categories

The request begins with the optional **<soapenv:Header>** which is empty, indicating no additional metadata such as authentication or session details are included. This SOAP request resides in the **<soapenv:Body>**, which contains the **<heal:getBFPCategories>** method call. This method takes two parameters which are **<gender>** and **<bfpValue>** with the value that is calculated according to the user input which is passed from calcBFPValue() This structured XML format ensures the server receives the necessary data to determine the corresponding BFP category.

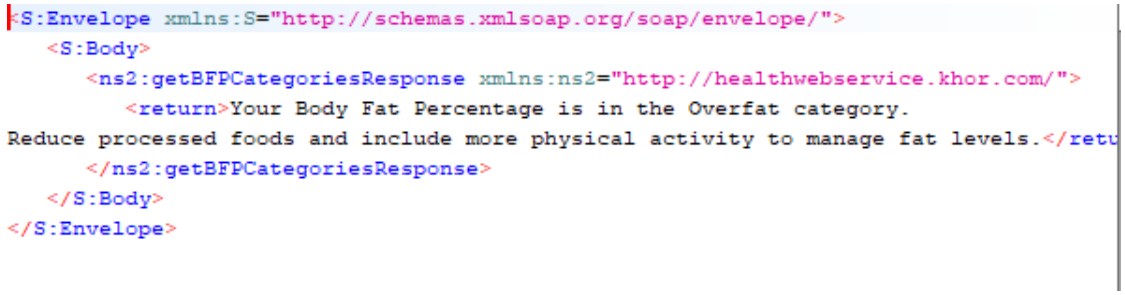

Figure 2.3.1.17 SOAP Response for BFP Categories

The response also starts in the **<S:Body>** which contains the **<ns2:getBFPCategoriesResponse>**. The **<return>** tag provides the result of a human-readable message classifying the Body Fat Percentage value that is calculated based on the user input into a certain category and recommending lifestyle changes to the users. The response from the server is shortened, but it clearly communicates health advice based on the Body Fat Percentage calculation for specific individuals.

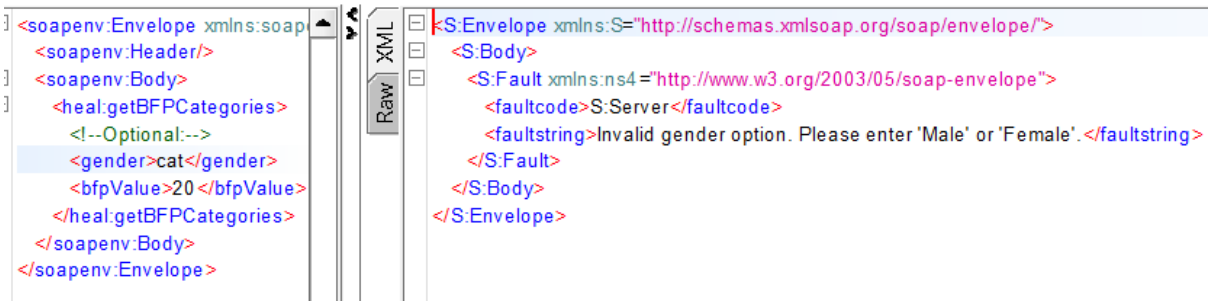

Figure 2.3.1.18 SOAP Fault Message Error for BFP Categories

In BFP categories methods, The SOAP fault will handle error if the user enter wrong gender and bfpValue. For example. If user enter bfpValue as cat, system will return “Invalid gender option. Please enter ‘Male’ or ‘Female’.”

(c) Calories Burn Rate Calculator and Categories

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" :
   <soapenv:Header/>
   <soapenv:Body>
      <heal:calcCaloriesValue>
         <weight>90</weight>
         <duration>156</duration>
         <met>10.5</met>
      </heal:calcCaloriesValue>
   </soapenv:Body>
</soapenv:Envelope>
```

Figure 2.3.1.19 SOAP Request for CBR Calculator

The SOAP request is in an XML message enclosed in a **<soapenv:Envelope>**, which defines the SOAP namespace. The optional **<soapenv:Header>** is empty, which shows that there is no additional metadata such as authentication. The main part of the request is in the **<soapenv:Body>**, which contains the **<heal:calcCaloriesValue>** method call. This method requires 3 inputs which are **<weight>**, **<duration>** and **<met>**. These measurements are important inputs for the server to perform the calories calculation accurately that is needed for the user's body.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:calcCaloriesValueResponse xmlns:ns2="http://healthwebservice.khor.com/">
         <return>2579.85</return>
      </ns2:calcCaloriesValueResponse>
   </S:Body>
</S:Envelope>
```

Figure 2.3.1.20 SOAP Response for CBR Calculator

The soap response for Calories Burned Has the **<S:Body>** tag which contains the **<ns2:calcCaloriesValueResponse>**, where the xmlns:ns2 attribute specifies the namespace for the healthcare service (http://healthwebservice.khor.com/). The **<return>** tag represents the computed Calories based on the user input that is derived from the formula of the Calories Burned. This level of Calories value suggests the patient falls into a certain category and highlights the service's role in providing actionable health metrics.

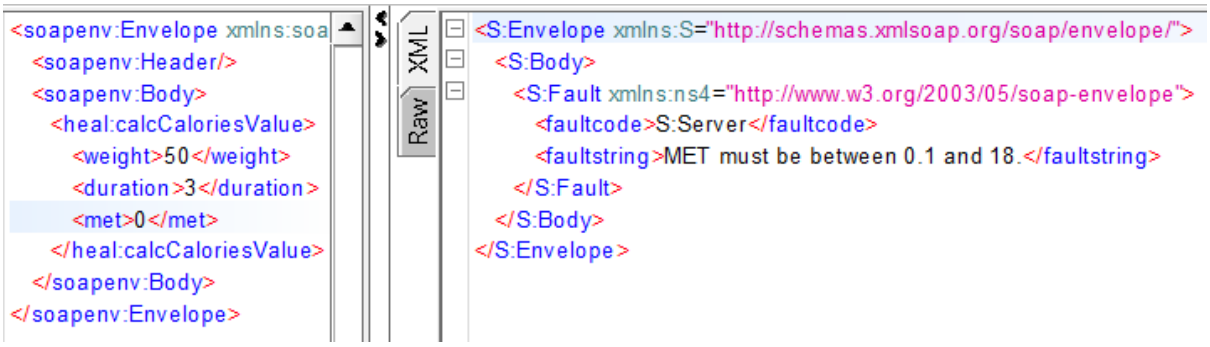

Figure 2.3.1.21 SOAP Fault Message Error for CBR Calculator

In CBR calculator methods, The SOAP fault will handle error if the user enter wrong weight, duration and met. For example. If user enter met as 3, system will return “MET must be between 0.1 and 18.”

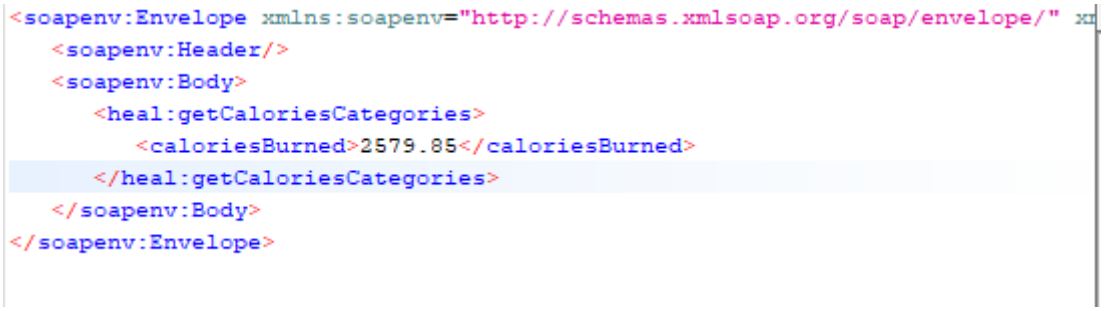

Figure 2.3.1.22 SOAP Request for CBR Categories

The request begins with the optional **<soapenv:Header>** which is empty, indicating no additional metadata such as authentication or session details are included. This SOAP request resides in the **<soapenv:Body>**, which contains the **<heal:getCaloriesCategories>** method call. This method takes a single parameter hich is **<caloriesBurned>** with the value that is calculated according to the user input which is passed from calcCaloriesValue() This structured XML format ensures the server receives the necessary data to determine the corresponding Calories Burned Category.

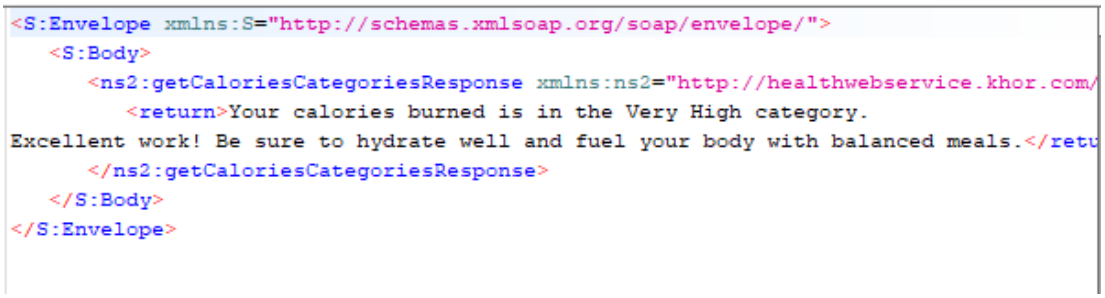

Figure 2.3.1.23 SOAP Response for CBR Categories

The response also starts in the **<S:Body>** which contains the **<ns2:getCaloriesCategoriesResponse>**. The **<return>** tag provides the result of a human-readable message classifying the Calories Burned value that is calculated based on the user input into a certain category and recommending lifestyle changes to the users. The response from the server is shortened, but it clearly communicates health advice based on the Calories Burned calculation for specific individuals.

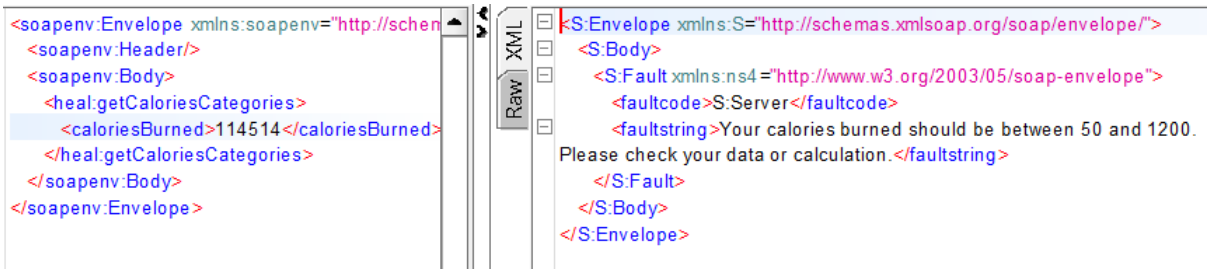

Figure 2.3.1.24 SOAP Fault Message Error for CBR Categories

In CBR categories methods, The SOAP fault will handle error if the user enter wrong caloriesBurned. For example. If user enter caloriesBurned as 114514, system will return “Your calories burned should be between 50 and 1200. Please check your data or calculation.”

(d) Basal Metabolic Rate (BMR) Calculator and Categories

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xm
   <soapenv:Header/>
   <soapenv:Body>
      <heal:calcBMRValue>
         <!--Optional:-->
         <gender>male</gender>
         <weight>90</weight>
         <height>170</height>
         <age>22</age>
      </heal:calcBMRValue>
   </soapenv:Body>
</soapenv:Envelope>
```

Figure 2.3.1.25 SOAP Request for BMR Calculator

The SOAP request is in an XML message enclosed in a **<soapenv:Envelope>**, which defines the SOAP namespace. The optional **<soapenv:Header>** is empty, which shows that there is no additional metadata such as authentication. The main part of the request is in the **<soapenv:Body>**, which contains the **<heal:calcBMRValue>** method call. This method requires 3 inputs which are **<gender>**, **<weight>** and **<height>**. These measurements are important inputs for the server to perform the calories calculation accurately that is needed for the user's body.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:calcBMRValueResponse xmlns:ns2="http://healthwebservice.khor.com/">
         <return>1985.03</return>
      </ns2:calcBMRValueResponse>
   </S:Body>
</S:Envelope>
```

Figure 2.3.1.26 SOAP Response for BMR Calculator

The soap response for Basal Metabolic Rate has the **<S:Body>** tag which contains the **<ns2:calcBMRValueResponse>**, where the xmlns:ns2 attribute specifies the namespace for the healthcare service (http://healthwebservice.khor.com/). The **<return>** tag represents the computed BMR based on the user input that is derived from the formula of the Basal Metabolic Rate. This level of Basal Metabolic Rate value suggests the patient falls into a certain category and highlights the service's role in providing actionable health metrics.

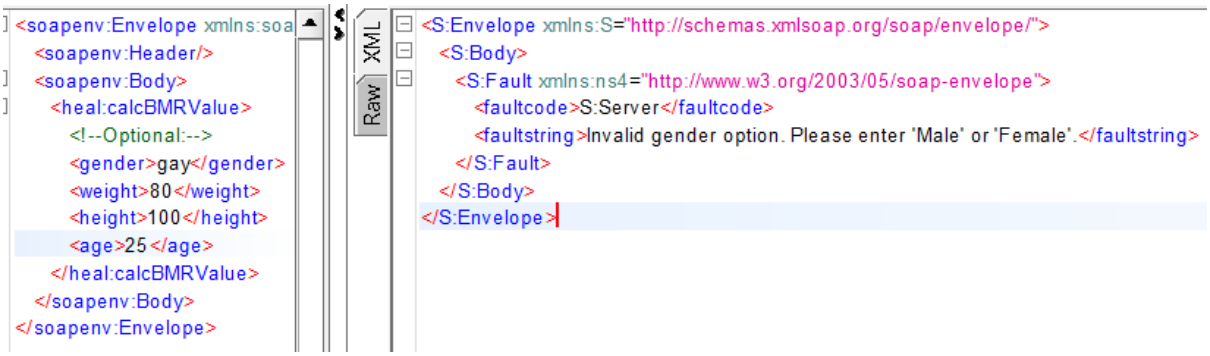

Figure 2.3.1.27 SOAP Fault Message Error for BMR Calculator

In BMR calculator methods, The SOAP fault will handle error if the user enter wrong gender, weight, height and age. For example. If user enter gender as gay, system will return “Invalid gender option. Please enter ‘Male’ or ‘Female’.”

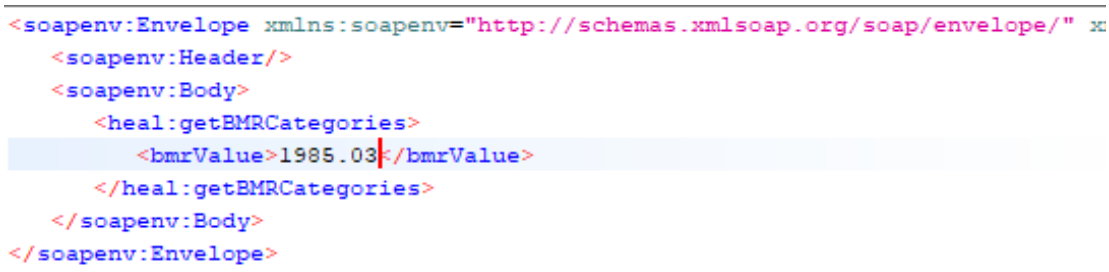

Figure 2.3.1.28 SOAP Request for BMR Categories

The request begins with the optional **<soapenv:Header>** which is empty, indicating no additional metadata such as authentication or session details are included. This SOAP request resides in the **<soapenv:Body>**, which contains the **<heal:getBMRCategories>** method call. This method takes single parameter which is **<bmrValue>** with the value that is calculated according to the user input which is passed from calcBMRValue() This structured XML format ensures the server receives the necessary data to determine the corresponding Basal Metabolic Rate Category.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:getBMRCategoriesResponse xmlns:ns2="http://healthwebservice.khor.com/">
         <return>Your Basal Metabolic Rate (BMR) value is in the High category.
Great metabolism! Ensure adequate calorie intake and stay active.</return>
      </ns2:getBMRCategoriesResponse>
   </S:Body>
</S:Envelope>
```

Figure 2.3.1.29 SOAP Response for BMR Categories

The response also starts in the **<S:Body>** which contains the **<ns2:getBMRCategoriesResponse>**. The **<return>** tag provides the result of a human-readable message classifying the Basal Metabolic Rate value that is calculated based on the user input into a certain category and recommending lifestyle changes to the users. The response from the server is shortened, but it clearly communicates health advice based on the Basal Metabolic Rate calculation for specific individuals.

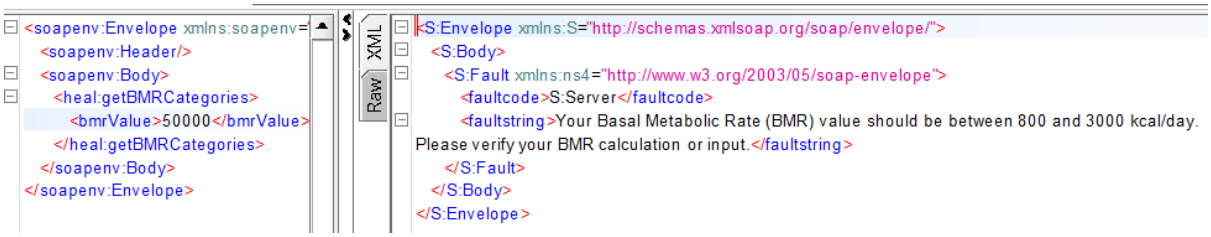

Figure 2.3.1.30 SOAP Fault Message Error for BMR Categories

In BMR categories methods, The SOAP fault will handle error if the user enter wrong bmrValue. For example. If user enter bmrValue as 50000, system will return “Your Basal Metabolic Rate (BMR) value should be between 800 and 3000 kcal/day. Please verify your BMR calculation or input.”

(e) Lean Body Mass (LBM) Calculator and Categories

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xm
   <soapenv:Header/>
   <soapenv:Body>
      <heal:calcLBMValue>
         <weight>90</weight>
         <height>170</height>
         <!--Optional:-->
         <gender>male</gender>
      </heal:calcLBMValue>
   </soapenv:Body>
</soapenv:Envelope>
```

Figure 2.3.1.31 SOAP Request for LBM Calculator

The SOAP request is in an XML message enclosed in a **<soapenv:Envelope>**, which defines the SOAP namespace. The optional **<soapenv:Header>** is empty, which shows that there is no additional metadata such as authentication. The main part of the request is in the **<soapenv:Body>**, which contains the **<heel:calcLBMValue>** method call. This method requires 3 inputs which are **<weight>**, **<height>** and **<gender>**. These measurements are important inputs for the server to perform the calories calculation accurately that is needed for the user's body.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:calcLBMValueResponse xmlns:ns2="http://healthwebservice.khor.com/">
         <return>62.82</return>
      </ns2:calcLBMValueResponse>
   </S:Body>
</S:Envelope>
```

Figure 2.3.1.32 SOAP Response for LBM Calculator

The soap response for Lean Body Mass has the **<S:Body>** tag which contains the **<ns2:calcLBMValueResponse>**, where the xmlns:ns2 attribute specifies the namespace for the healthcare service (http://healthwebservice.khor.com/). The **<return>** tag represents the computed value based on the user input that is derived from the formula of the Lean Body Mass. This level of Lean Body Mass value suggests the patient falls into a certain category and highlights the service's role in providing actionable health metrics.

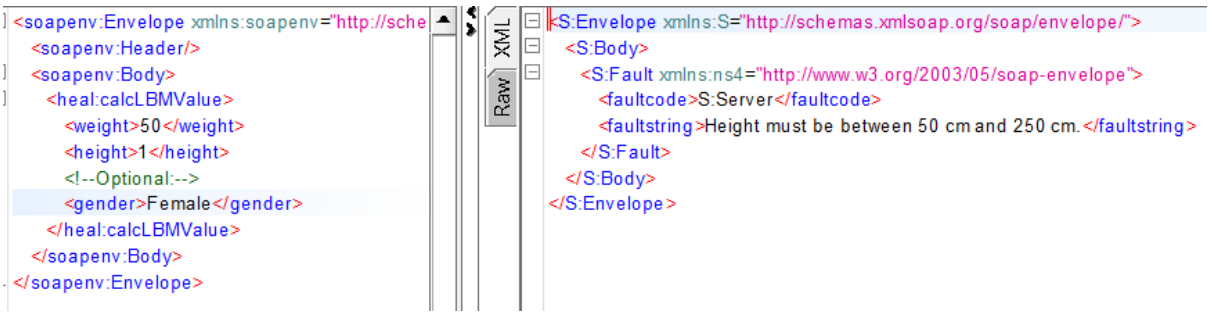

Figure 2.3.1.33 SOAP Fault Message Error for LBM Calculator

In LBM calculator methods, The SOAP fault will handle error if the user enter wrong weight, height and gender. For example. If user enter height as 1 cm, system will return “Height must be between 50 cm and 250 cm.”

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" x
   <soapenv:Header/>
   <soapenv:Body>
      <heal:getLBMCategories>
         <lbm>62.82</lbm>
         <weight>90</weight>
      </heal:getLBMCategories>
   </soapenv:Body>
</soapenv:Envelope>
```

Figure 2.3.1.34 SOAP Request for LBM Categories

The request begins with the optional **<soapenv:Header>** which is empty, indicating no additional metadata such as authentication or session details are included. This SOAP request resides in the **<soapenv:Body>**, which contains the **<heal:getLBMCategories>** method call. This method takes two parameters which are **<lbm>** and **<weight>** with the value that is calculated according to the user input which is passed from calcLBMValue() This structured XML format ensures the server receives the necessary data to determine the corresponding Lean Body Mass Category.

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
   <S:Body>
      <ns2:getLBMCategoriesResponse xmlns:ns2="http://healthwebservice.khor.com/">
         <return>Your Lean Body Mass (LBM) value is in the Normal category.
Maintain your current lifestyle with balanced nutrition and regular exercise.</return
      </ns2:getLBMCategoriesResponse>
   </S:Body>
</S:Envelope>
```

Figure 2.3.1.35 SOAP Response for LBM Categories

The response also starts in the **<S:Body>** which contains the **<ns2:getLBMCategoriesResponse>**. The **<return>** tag provides the result of a human-readable message classifying the Lean Body Mass value that is calculated based on the user input into a certain category and recommending lifestyle changes to the users. The response from the server is shortened, but it clearly communicates health advice based on the Lean Body Mass calculation for specific individuals.

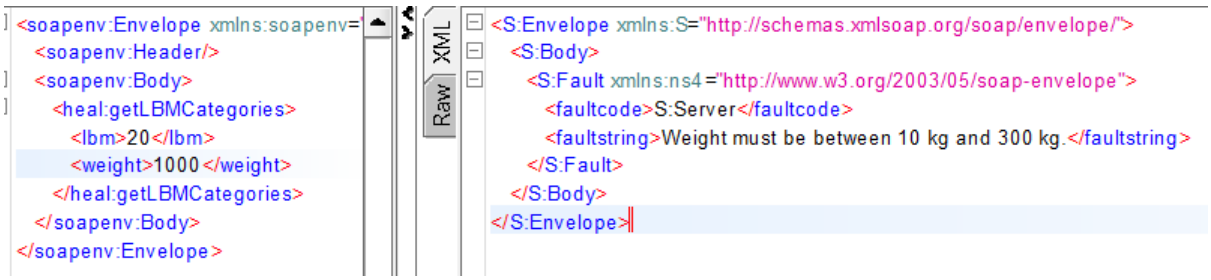

Figure 2.3.1.36 SOAP Fault Message Error for LBM Categories

In LBM categories methods, The SOAP fault will handle error if the user enter wrong lbm and weight. For example. If user enter weight as 1000 kg, system will return “Weight must be between 10 kg and 300 kg.”

2.3.2 Web Service Description Language (WSDL)

This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
<!-- Published by JAX-WS RI (http://jax-ws.java.net). RI's version is Metro/2.4.0 (wsit240-7e98ff4; 2017-08-03T21:19:54+0200) JAXWS-RI/2.3.0 JAXB-RI/2.3.0 JAXB-API/2.3.0 svn-revision#unknown. -->
<!-- Generated by JAX-WS RI (http://javaee.github.io/metro-jax-ws). RI's version is Metro/2.4.0 (wsit240-7e98ff4; 2017-08-03T21:19:54+0200) JAXWS-RI/2.3.0 JAXB-RI/2.3.0 JAXB-API/2.3.0 svn-revision#unknown. -->
<definitions xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsp="http://www.w3.org/ns/ws-policy" xmlns:wsp1_2="http://schemas.xmlsoap.org/ws/2004/09/policy"
 xmlns:wsam="http://www.w3.org/2007/05/addressing/metadata" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://healthwebservice.khor.com/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://healthwebservice.khor.com/" name="HealthWebService">
  <types>
    <xsd:schema>
      <xsd:import namespace="http://healthwebservice.khor.com/" schemaLocation="http://localhost:9080/HealthFitnessCalc/HealthWebService?xsd=1"/>
    </xsd:schema>
  </types>
  <message name="getUserInfo">
    <part name="parameters" element="tns:getUserInfo"/>
  </message>
  <message name="getUserInfoResponse">
    <part name="parameters" element="tns:getUserInfoResponse"/>
  </message>
  <message name="calcCaloriesValue">
    <part name="parameters" element="tns:calcCaloriesValue"/>
  </message>
  <message name="calcCaloriesValueResponse">
    <part name="parameters" element="tns:calcCaloriesValueResponse"/>
  </message>
  <message name="getCaloriesCategories">
    <part name="parameters" element="tns:getCaloriesCategories"/>
  </message>
  <message name="getCaloriesCategoriesResponse">
    <part name="parameters" element="tns:getCaloriesCategoriesResponse"/>
  </message>
  <message name="getBMICategories">
    <part name="parameters" element="tns:getBMICategories"/>
  </message>
  <message name="getBMICategoriesResponse">
    <part name="parameters" element="tns:getBMICategoriesResponse"/>
  </message>
  <message name="calcAge">
    <part name="parameters" element="tns:calcAge"/>
  </message>
  <message name="calcAgeResponse">
    <part name="parameters" element="tns:calcAgeResponse"/>
  </message>
  <message name="calcLBMValue">
    <part name="parameters" element="tns:calcLBMValue"/>
  </message>
  <message name="calcLBMValueResponse">
    <part name="parameters" element="tns:calcLBMValueResponse"/>
  </message>
  <message name="getLBMCategories">
    <part name="parameters" element="tns:getLBMCategories"/>
  </message>
  <message name="getLBMCategoriesResponse">
    <part name="parameters" element="tns:getLBMCategoriesResponse"/>
  </message>
  <message name="calcBMRValue">
    <part name="parameters" element="tns:calcBMRValue"/>
  </message>
  <message name="calcBMRValueResponse">
    <part name="parameters" element="tns:calcBMRValueResponse"/>
  </message>
  <message name="calcBMIValue">
    <part name="parameters" element="tns:calcBMIValue"/>
  </message>
  <message name="calcBMIValueResponse">
    <part name="parameters" element="tns:calcBMIValueResponse"/>
  </message>
  <message name="calcBFPValue">
    <part name="parameters" element="tns:calcBFPValue"/>
  </message>
  <message name="calcBFPValueResponse">
    <part name="parameters" element="tns:calcBFPValueResponse"/>
  </message>
  <message name="getBFPCategories">
    <part name="parameters" element="tns:getBFPCategories"/>
  </message>
  <message name="getBFPCategoriesResponse">
    <part name="parameters" element="tns:getBFPCategoriesResponse"/>
  </message>
  <message name="getBMRCategories">
    <part name="parameters" element="tns:getBMRCategories"/>
  </message>
  <message name="getBMRCategoriesResponse">
    <part name="parameters" element="tns:getBMRCategoriesResponse"/>
  </message>
  <portType name="HealthWebService">
    <operation name="getUserInfo">
      <input wsam:Action="http://healthwebservice.khor.com/HealthWebService/getUserInfoRequest" message="tns:getUserInfo"/>
      <output wsam:Action="http://healthwebservice.khor.com/HealthWebService/getUserInfoResponse" message="tns:getUserInfoResponse"/>
    </operation>
    <operation name="calcCaloriesValue">
      <input wsam:Action="http://healthwebservice.khor.com/HealthWebService/calcCaloriesValueRequest" message="tns:calcCaloriesValue"/>
      <output wsam:Action="http://healthwebservice.khor.com/HealthWebService/calcCaloriesValueResponse" message="tns:calcCaloriesValueResponse"/>
    </operation>
    <operation name="getCaloriesCategories">
      <input wsam:Action="http://healthwebservice.khor.com/HealthWebService/getCaloriesCategoriesRequest" message="tns:getCaloriesCategories"/>
      <output wsam:Action="http://healthwebservice.khor.com/HealthWebService/getCaloriesCategoriesResponse" message="tns:getCaloriesCategoriesResponse"/>
    </operation>
    <operation name="getBMICategories">
      <input wsam:Action="http://healthwebservice.khor.com/HealthWebService/getBMICategoriesRequest" message="tns:getBMICategories"/>
      <output wsam:Action="http://healthwebservice.khor.com/HealthWebService/getBMICategoriesResponse" message="tns:getBMICategoriesResponse"/>
    </operation>
    <operation name="calcAge">
      <input wsam:Action="http://healthwebservice.khor.com/HealthWebService/calcAgeRequest" message="tns:calcAge"/>
    </operation>
  </portType>
  <binding name="HealthWebServicePortBinding" type="tns:HealthWebService">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="getUserInfo">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="calcCaloriesValue">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="getCaloriesCategories">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="getBMICategories">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="calcAge">
      <soap:operation soapAction=""/>
      <input>
        <soap:body use="literal"/>
      </input>
```

Figure 2.3.2.1 WSDL File for Health and Fitness Calculator

The HealthWebService is a SOAP-based web service that offers a range of health and fitness calculations to help users better understand their physical well-being. The service

begins with detailed XML namespace definitions and includes a **<types>** section that imports an external XSD schema from the same endpoint. This ensures strong type safety and accurate data structures for all request and response messages.

Among the key functionalities provided by the service are calculations for Body Mass Index (calcBMIValue), Basal Metabolic Rate (calcBMRValue), Lean Body Mass (calcLBMValue), Body Fat Percentage (calcBFPValue) and Caloric Burned(calcCaloriesValue). Each of these results are provided by a corresponding categorization method such as getBMICategories or getBMRCategories which can translate the values into easy-to-understand health classifications like "Low," "Normal," or "High." This approach transforms raw data into practical health insights, making the service both informative and user-friendly.

The HealthWebService is built for compatibility and ease of integration, using well-established web service standards. The **<binding>** section of the WSDL specifies that all operations use document SOAP binding over HTTP, which can ensure smooth interoperability across different platforms. The **<portType>** section acts as the service's interface, outlining the input and output messages for each function the service performs.

Key technical features from this WSDL file can ensure all messages can be validated against the XML schema, and the use of a consistent request-response pattern for every operation. The soapAction fields are left empty, meaning there are no extra processing instructions required. Although the WSDL doesn’t define specific security policies, it's clear that the service is intended to rely on transport-level protection like HTTPS for secure communication. This design makes the service suitable for integration with health applications or fitness trackers requiring reliable metric computations like calculating Body Mass Index, Basal Metabolic Rate, Lean Body Mass, Body Fat Percentage and Caloric Burned.

2.3.3 Universal Description, Discovery and Integration (UDDI)

UDDI, which stands for Universal Description, Discovery, and Integration, plays an important role in the Health and Fitness Calculator system. It acts like a central directory where different calculator services such as the Body Mass Index Calculator, Body Fat Percentage Calculator, Calories Burned Calculator, Basal Metabolic Rate Calculator and Lean Body Mass Calculator. This is because UDDI follows a platform-independent, XML-based standard which allows these services to be easily listed, found, and used by other systems.

By registering each module in the UDDI registry, the system makes it possible for clients or other services to discover and connect to these calculators without needing to know their exact location ahead of time. This flexible discovery can make the applications to search the registry based on the type of service they need such as Body Mass Index or calorie burned tracking, and immediately get the necessary details like the service's description and endpoint URL to start using it immediately.

This process makes integration much smoother, especially in environments where services may be updated or moved around. Clients don't need to manually adjust their configurations as they simply rely on UDDI to always find the latest version of the system. In a Nutshell, UDDI helps make the Health and Fitness Calculator system more flexible, easier to expand, and better suited for integration across different platforms or health-related applications for the users.

3.0 Task Distribution

3.1 Team Members and Responsibilities

Group Member	Task
Arun Kumar A/L Manimaran	• Web service: module Body Mass Index (BMI) Calculator • Report: 4.0 Conclusion
Khor Hwee Shian	• Client app: Create and Build Client App GUI Design • Report: 3.0 Task Distribution
Siti Nur Esya Binti Buwang	• Web service: Module Body Fat Percentage Calculator • Report: 1.0 Introduction
Yuvanesh A/L Shanmugam	• Web service: Module Calories Burn Rate Calculator • Report: 2.1 Overview of the Web Service and Client Application
Vivethan A/L Murugan	• Web service: Module Basal Metabolic

	Rate (BMR) Calculator • Report: 2.2 Explanation of Modules Developed
Kugaan Raaj A/L Gobalan	• Web service: Module Lean Body Mass (LBM) Calculator • Report: 2.3 Explanation SOAP Messaging, WSDL and UDDI used

Figure 3.1.1 Team Members and Responsibilities

3.2 Meeting Schedules and Proof

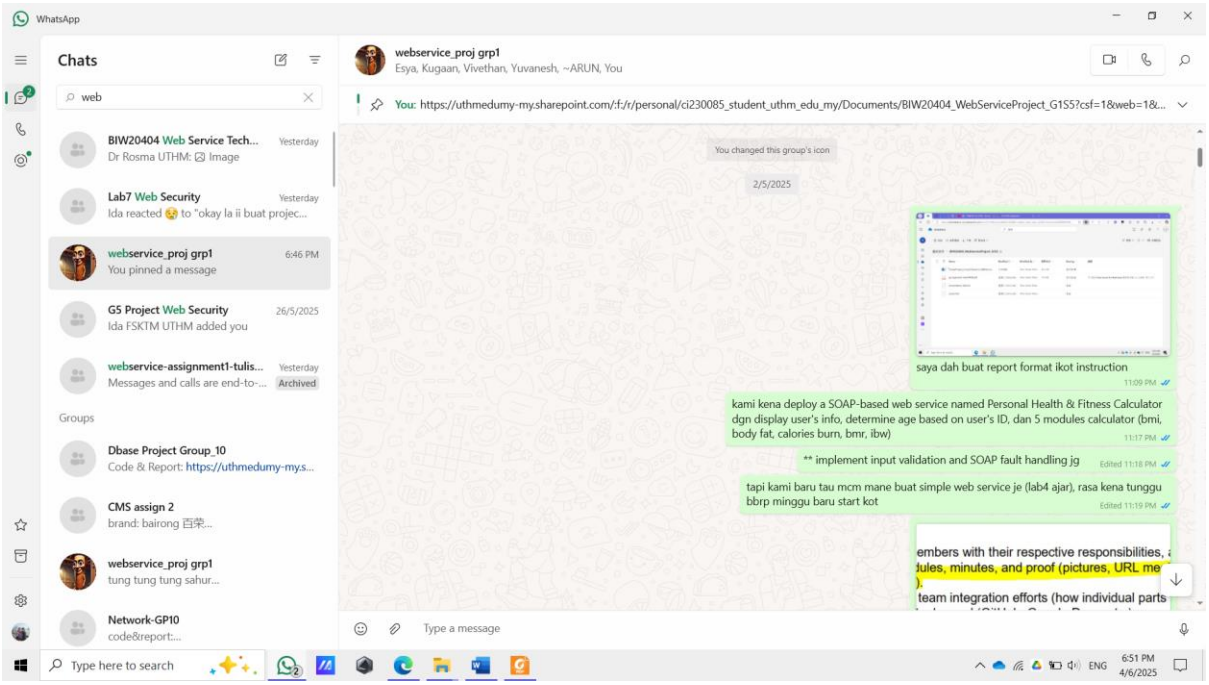

Figure 3.2.1 Whatsapp Log for 2nd May, 2025

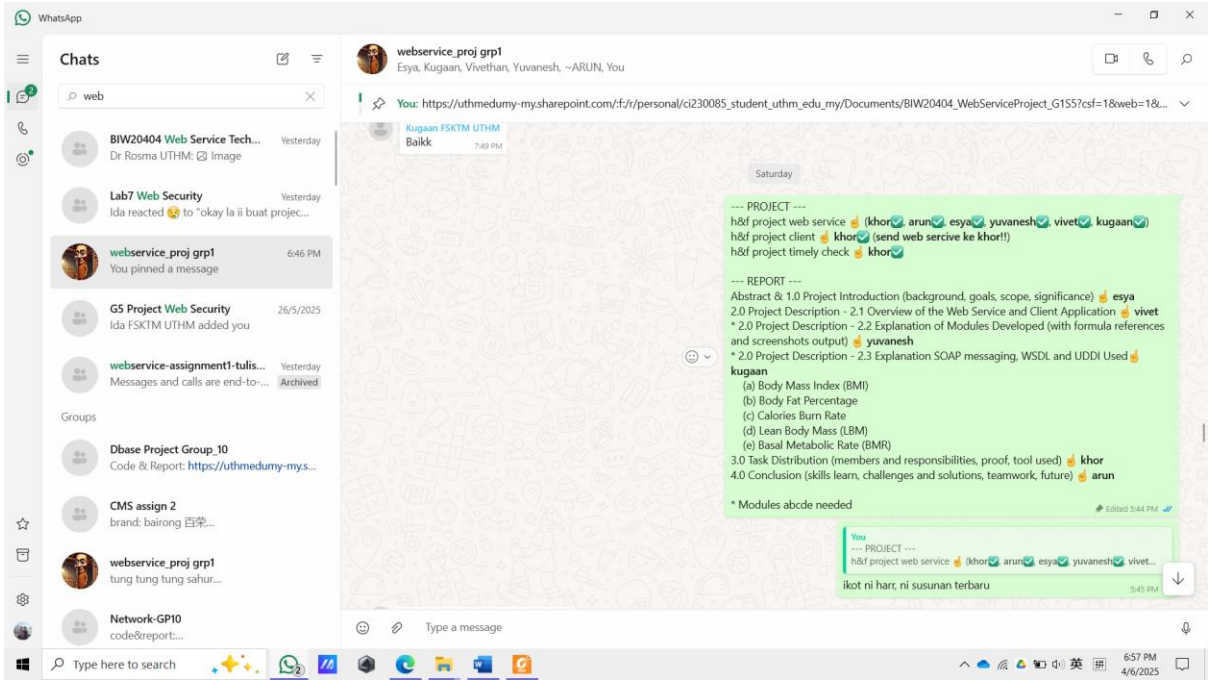

Figure 3.2.2 Whatsapp Log for 26th May, 2025

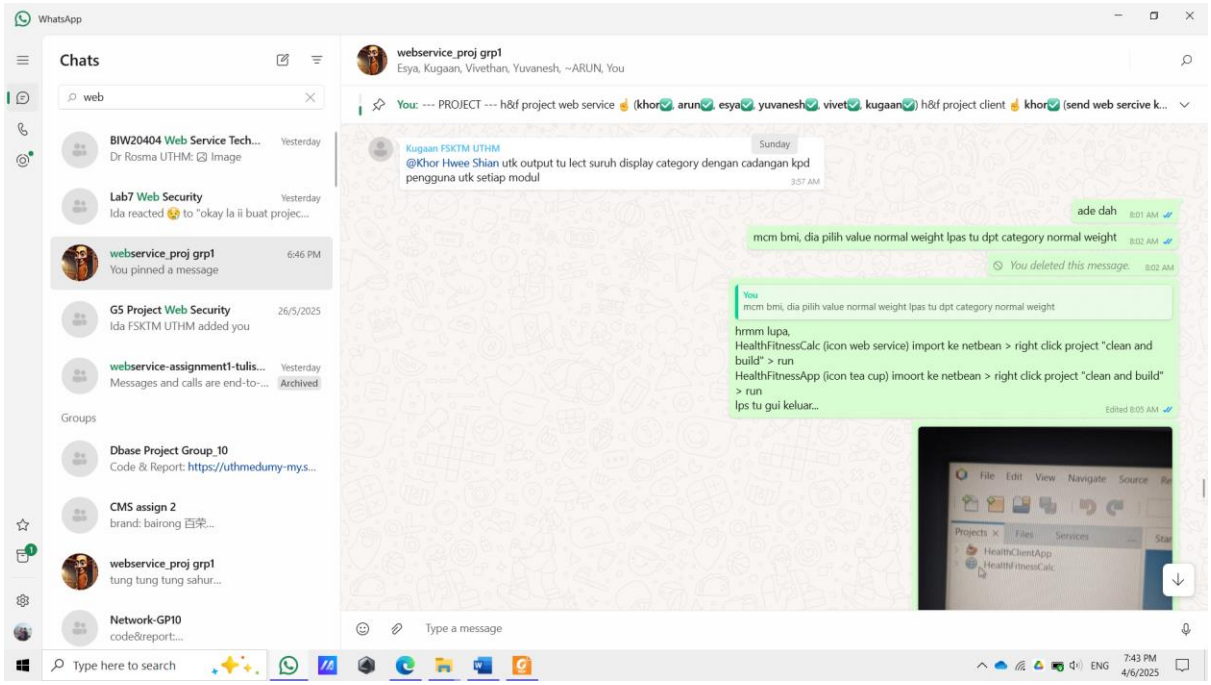

Figure 3.2.3 Whatsapp Log for 1st June, 2025

Figures 3.2.1, 3.2.2 and 3.2.3 above shows how our group uses WhatsApp for discussions and task distribution. Most of the simpler discussions, such as task assignments and project confirmations, are carried out through WhatsApp. For example, Khor is responsible for the client app GUI design, Arun handles the BMI module, Esya works on the Body Fat

Percentage module, Yuvanesh is in charge of the Calories Burn Rate module, Vivethan is responsible for the Lean Body Mass (LBM) module, and Kugaan handles the Basal Metabolic Rate (BMR) module. In addition, some members help remind the group about UI design issues; for instance, when the lecturer was scheduled to review our project progress on May 27th, Kugaan took responsibility for checking and reminding the team. Lastly, we also discussed how to divide the report writing. We decided that all six of us would contribute: Arun is responsible for Chapter 4, Esya for Chapter 1, Khor for Chapter 3, and Yuvanesh, Vivethan, and Kugaan will work together on Chapter 2.

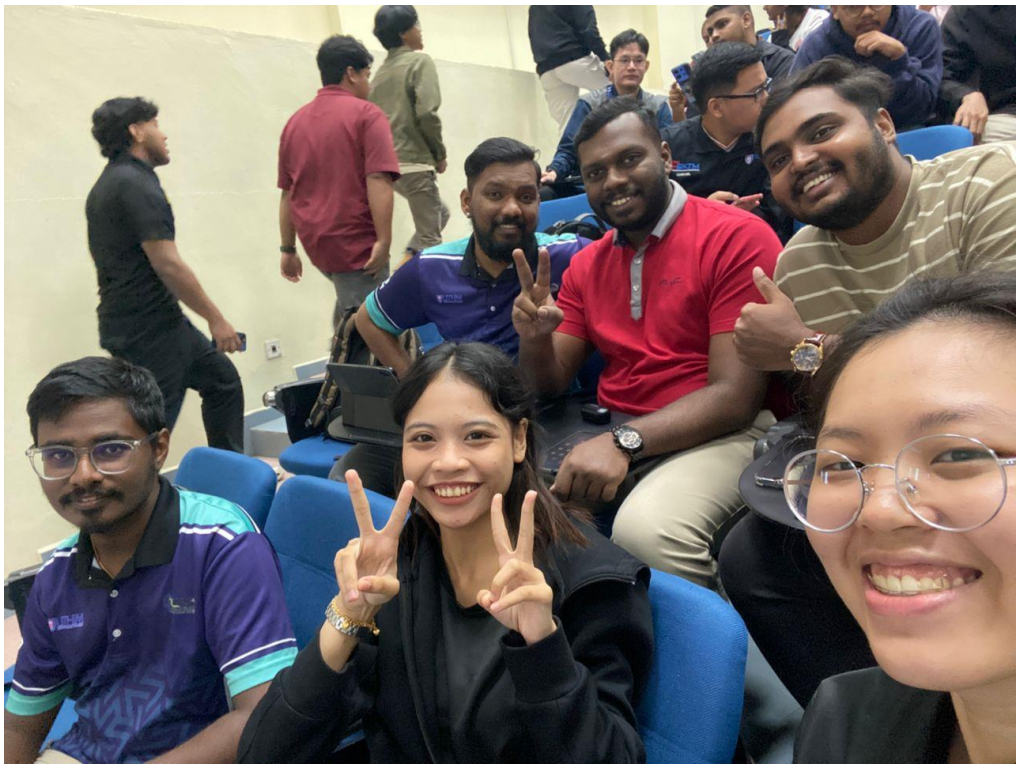

Figure 3.2.4 Whatsapp Log for 22th May, 2025

Figure 3.2.4 above shows our group having a face-to-face discussion about the extra modules and the task of finding references. For example, Khor and Esya were responsible for explaining the project requirements, while Yuvanesh, Kugaan, and Vivethan searched online to find suitable extra modules. In addition, some members needed to clarify the specific details of the GUI design done by Khor.

3.3 Integration and Collaboration Tools Used

- We use **OneDrive** to store the report and project backups.
- We use **GitHub** to store and regularly update the project.
- We use **Microsoft Word** to write the report.
- We use **WhatsApp** for communication.
- We use **Apache NetBeans** to create the web service and design the client app UI.
- We use **SOAP UI** to test the SOAP Request and Response.
- We use **YouTube and Google Chrome** to search for relevant information.

4.0 Conclusion

4.1 Knowledge and Skills Gained

- We learned the principles behind SOAP-based web services, including WSDL, XML communication, and service-client architecture.
- By implementing the service using Java in NetBeans, we strengthened our coding skills, especially in modularizing code and handling user inputs.
- We researched and applied scientifically validated formulas for BMI, Body Fat Percentage, Calories Burn Rate, Basal Metabolic Rate (BMR) and Lean Body Mass (LBM), enhancing our ability to integrate real-world calculations into programming.

4.2 Challenges and Solutions

CHALLENGES	SOLUTION
Understanding SOAP structure and WSDL configuration.	We studied multiple tutorials from various online platform to solve this challenge.
Applying correct health formulas.	We verified multiple external sources to get the information.
Handling invalid or missing inputs.	Input validation and SOAP fault handling were implemented to provide meaningful error messages to the user.

Figure 4.2.1 Challenges and Solutions

4.3 Teamwork and Technical Lessons

- We learned that effective communication and task delegation are crucial in a team project. Each member took responsibility for specific modules and regularly updated others.
- Pair programming and code reviews helped us improve each other's logic and efficiency.

- We utilized Github and Microsoft Word to manage tasks and documentation collaboratively.

4.4 Future Recommendations

- For better scalability and modern application support, consider developing a RESTful version of the calculator for broader integration.
- Create a mobile-friendly frontend using Android Studio to make the service more accessible.
- Add user authentication and history tracking using a MySQL backend to provide long-term health tracking.

5.0 References

[1] **Calculatorsoup. (n.d.).** *BMI Calculator.* https://www.calculatorsoup.com/calculators/health/bmi-calculator.php

[2] **Cleveland Clinic. (2023, December 11).** *Body mass index (BMI): What it is & how to calculate*. https://my.clevelandclinic.org/health/articles/9464-body-mass-index-bmi

[3] **TG Fitness. (n.d.).** *Body Fat Percentage Calculator*. https://www.tgfitness.com/body-fat-percentage-calculator/#:~:text=The%20formula%20uses%20a%20person's,0.23%20x%20Age)%20%E2%80%93%205.4.

[4] **Healthline. (2025, June 11).** *Ideal body fat percentage: For men and women*. https://www.healthline.com/health/exercise-fitness/ideal-body-fat-percentage

[5] **Calculator.net. (n.d.).** *Calories Burned Calculator*. https://www.calculator.net/calories-burned-calculator.html

[6] **Cleveland Clinic. (2023, June 9).** *Here's how many calories you naturally burn in a day*. https://health.clevelandclinic.org/calories-burned-in-a-day

[7] **Diabetes.co.uk. (n.d.).** *BMR Calculator*. https://www.diabetes.co.uk/bmr-calculator.html

[8] **Cleveland Clinic. (2024, December 11).** *Basal metabolic rate (BMR)*. https://my.clevelandclinic.org/health/body/basal-metabolic-rate-bmr

[9] **GymNation. (n.d.).** *Lean Body Mass Calculator*. https://gymnation.com/fitness-calculators/lean-body-mass-calculator/

[10] **Bodybuilding Wizard. (n.d.).** *Lean Body Mass (LBM) Calculator*. Retrieved June 11, 2025, from https://bodybuilding-wizard.com/lean-body-mass-lbm-calculator/