

# ECE3156

## Software Engineering

### Project Assignment Report

### Trimester 1, 2022/2023

Name : Marawan Ashraf Fawzy Ahmed Eldeib

Student ID : 1181102334

Major : CE

Contact No : 01123782954

E-mail : 1181102334@student.mmu.edu.my

Signature :  Submission Date: 08/01/2020

Organization Name : SOG

Software Application Name : BodyMath

Software Application Theme : Fitness Application

Organization Role : SQA Manager

(E.g.: Project Manager, SQA/SCM Manager, Developer, Designer)

#### Other Team members:

Student ID	Name	Major	Organization Role
1191302231	Ali karimeh	CE	Project Manager
1181102921	Bashir Tawfiq	CE	SCM Manager / Lead programmer
1171103943	Yousef Yasser	CE	Lead Designer / Document Manager

2023

SOG Co.



**RC1: Phase 1 report  
(Organization  
profile, software  
proposal).**

## MEMBERS AND RESPONSIBILITIES

MEMBER	ROLE	RESPONSIBILITIES
ALI SALIM KARIMEH	Project Manager	<ul style="list-style-type: none"> <li>➤ Representing the project coordinator.</li> <li>➤ Coordination of the team members in carrying out the project activities.</li> <li>➤ Final say in decisions if the team is unable to reach a decision.</li> </ul>
	Designer	<ul style="list-style-type: none"> <li>➤ Representing the project coordinator.</li> <li>➤ Coordination of the team members in carrying out the project activities.</li> <li>➤ Final say in decisions if the team is unable to reach a decision.</li> </ul>
	Programmer	<ul style="list-style-type: none"> <li>➤ Coordination of software development tasks among developers.</li> <li>➤ Knowledge of programming language and tools.</li> </ul>
BASHIR TAWFIG BASHIR ABUGHARSA	Software Configuration Manager	<ul style="list-style-type: none"> <li>➤ Responsible for controlling changes that take place during project development.</li> </ul>



MARAWAN ASHRAF  
ELDEIB



### Designer

- Representing the project coordinator.
- Coordination of the team members in carrying out the project activities.
- Final say in decisions if the team is unable to reach a decision.

### Lead Programmer

- Coordination of software development tasks among developers.
- Knowledge of programming language and tools.

### Software Quality Assurance Manager

- Responsible for managing the quality of the project development.

### Designer

- Representing the project coordinator.
- Coordination of the team members in carrying out the project activities.
- Final say in decisions if the team is unable to reach a decision.

**YOUSSEF YASSER  
SALAHEDIN  
ELHAMMAMY**



### **Programmer**

- Coordination of software development tasks among developers.
- Knowledge of programming language and tools.

### **Lead Designer**

- Representing the project coordinator.
- Coordination of the team members in carrying out the project activities.
- Final say in decisions if the team is unable to reach a decision.

### **Programmer**

- Coordination of software development tasks among developers.
- Knowledge of programming language and tools.

### **Documentation Manager**

- Responsible for documentation activities.
- Coordination of document preparation and recording tasks.

# ORGANISATIONAL PROFILE

## ABOUT US

An expert web and mobile app developer based in Malaysia. Though we work mostly from Kuala Lumpur, our partners span the globe. For businesses worldwide, we design and create top-tier web and mobile applications.

Our passion is developing influential digital goods, and we do this by combining excellent user experience design with reliable software engineering. We put a lot of effort into exceeding expectations and assisting our clients in succeeding, while also offering straightforward, no-nonsense guidance along the road.

## OUR HISTORY

In 2009, we began creating amazing experiences. Business was started by a group of four co-founders. They borrowed \$500,000 from the bank. The first job they had with a company, for which they had a two-week deadline to complete an app called Shadow Light, was the most difficult. In 2015, the firm expanded and opened additional offices.

## MARKET CAPITAL



Sog has established itself as one of Malaysia's fastest growing companies, with a market capitalization of RM 90 million in 2019 and a future value projection of RM 150 million in 2027. Even though the company may not be as large as some of the industry giants, we have overcome numerous challenges and uncertainties that were initially stacked against us. The positive response across a 14-year period towards our growth is mostly due to the successful penetration Malaysia market. The COVID-19 pandemic's effect on the world economy, especially Malaysia's, is now predicted to have a significant negative impact on the company's market value.

## BOARD OF DIRECTORS



**James Lyons**

**Chairman / Independent Non-Executive Director**

**Nationality:** Italy

**Age:** 45

**Date of Appointment as Director of SOG:** 13 Jun 2014

### **Qualification**

Bachelor of Science in Economics, Sunway University (Malaysia, 1997)

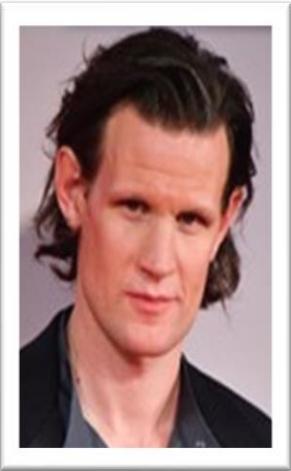
Master of Business, UCSI University (Malaysia, 2001)

### **Working Experience/Occupation**

SOG Co's current chairman is James Lyons. James Lyons was the Executive Chairwoman and CEO of Ant Financial Services Group prior to joining SOG. James Lyons, one of Alibaba Group's founders, led various business units, including marketing and services divisions, and also served as the company's Chief People Officer.

### **Board Committee / Directorship in other public or listed companies**

AL Bura Co.



**Matt Smith**

**Non-Executive Director**

**Nationality:** England

**Age:** 34

**Date of Appointment as Director of SOG:** 12 July 2020

#### **Qualification**

Bachelor of Commerce, University of Monash (Australia, 2005)

Chartered Accountancy, Institute of Chartered Accountants (Malaysia, 2001)

#### **Working Experience/Occupation**

Matt Smith is a Non-Executive Director at SOG Co., where he is in charge of the company's commercial portfolio, which includes managing strategic accounts, retail, and marketing solutions.

Matt was the Chief Executive Officer of Lazada Singapore before joining sog. He also co-founded and served as Managing Director of Lazada Philippines for two years, building the team and growing the company into one of the top eCommerce marketplaces in the country. Matt previously worked as a strategy consultant for Accenture in Australia, the United States, and the United Kingdom.

Matt holds a Bachelor of Commerce from the University of Monash.

#### **Board Committee / Directorship in other public or listed companies**

TeData Co.

Etisalat Co.

Meta Co.

## OUR SERVICES

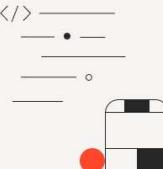
We provide business owners technology consultation services so they may understand how technology functions and how to turn their business into digital. We let our work to speak for itself, and thanks to our achievements, we are still able to do what we love.

OUR EXPERTISE



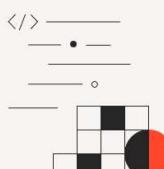
### Design

- Design Research
- Design Concepts
- Prototypes
- UI/UX Design
- UI/UX Production



### Mobile App Development

- iOS and tvOS App Development
- Android App Development
- Hybrid App Development
- CarPlay App Development
- Android Auto App Development



### Web Development

- Frontend Development
- Backend Development
- System Integration
- Cloud Managed Services

## OUR Products



Book Palace is the single destination for all the books you love — and all the ones you'll love next. Browse the Book Store to find the perfect book to read. Track what you've read and want to read, and set your own Reading Goals — all in one app



A food recommendation/review app displays recommendations from users who enjoy trying out new restaurants and can be relied upon to publish reviews and recommendations of what customers should order when they visit a certain restaurant.



The majority of lifestyle disorders can be avoided by drinking more water. With this software, you can be certain to consume the recommended amount of water each day.



This app is made to allow users to adjust and control all of their home's lighting online.



City Railway

"City Railway" is a unique train app that displays live train status and up-to-date schedules.

**ROZA**

It is a website that displays every available meals (menu) for a particular café or restaurant after scanning a QR code.

 SMARTHOUSE

Using a smartphone or tablet and an internet connection, a smart house application enables homeowners to remotely control lights, appliances, thermostats, and other devices.

 Lingo

One of the best app concepts for 2022 is a language learning app that provides users with some basic courses in several languages. The program has various difficulty levels, from the beginner level, which focuses on alphabets and fundamental letters, to the advanced level.

  
OASIS VILLAGE

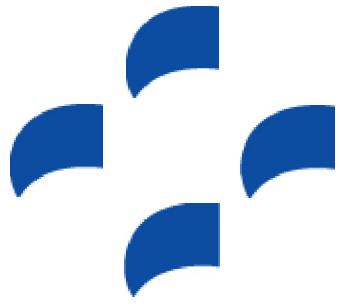
It is basically a resort map application that will direct you from one point to any other within the resort, such as the swimming pool, by indicating all the locations within it.



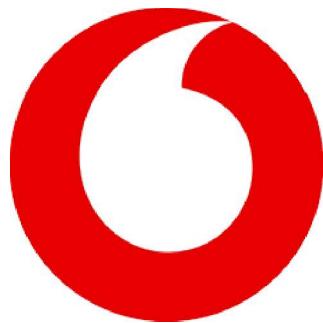
An app that allows you to water any plant you choose in the app in order to monitor the plants you have at home

## Our clients



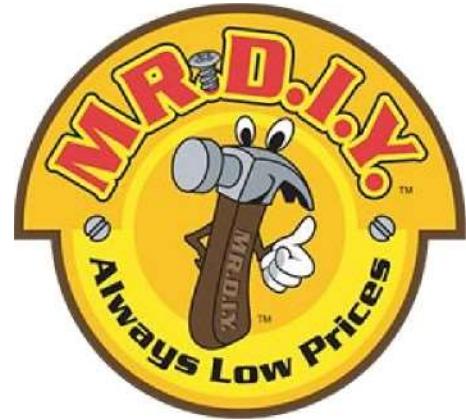


KUALA LUMPUR  
CONVENTION CENTRE



vodafone





**FUTVISE**  
INNOVATIONS FOR THE FUTURE

tealive

# Ali Karimeh



Solstice Cyberjaya, Selangor



01164206506



[ali-85-karima@hotmail.com](mailto:ali-85-karima@hotmail.com)



## Experience

### **November 2017–January 2018**

Game Designer • Intern • Station of Games

### **July 2018–May 2019**

IOT Engineer • Full time • Wyze Labs

### **July 2019–September 2021**

Junior Software Engineer • Full time • TM

### **October 2021-present**

Senior Software Engineer • Full time • Sog

## Education and Certificates

- Multimedia University, Cyberjaya, Malaysia  
B.E. (Hons) (Electronics-Comp) 2011-2016 Graduated  
with CGPA 3.60/4.00
- Oracle Java ME Mobile Application Developer –  
intermediate
- Front-End Web Development with React (Coursera)
- Meta Front-End Developer Professional Certificate

- Meta Back-End Developer Professional Certificate
- Oracle System Database.

## SKILLS

- Phyton
- HTML
- JavaScript
- Apache HTTP server
- XML
- Unix
- C/C++ object oriented
- MySQL and PHP
- Web and Mobile application
- Systems software and hardware configuration and troubleshooting.
- Microsoft Word, PowerPoint, and Excel.
- Software development

## Languages

- Arabic-Native
- English-fluent,
- Turkish beginner
- French-beginner



Bashir TawfigBashir Abugharsa

## Experience

### **2016**

Software Engineer- Internship -Dell – Malaysia

- Managed company's main database system.
- Wrote statistical reports about company performance.
- Managed company's social media accounts.

### **2017–2019**

Junior Software Engineer- Full time- Al Bura – Libya

- Design, development, bug-fixing, and maintenance of software solutions.
- Writing reports
- Collaborating with clients and cross-functional departments to offer project status updates.

### **2019–Present**

Senior software Engineer -Full time -Sog – Malaysia

- Assigning tasks to junior developers such as coding, testing, debugging, and analytics.
  - Design, development, bug-fixing, and maintenance of advanced software solutions.
- Games Library Database project using MySQL.
- Using an 8051 microcontroller and assembly language, create a car parking system.
- Capstone project: IoT-based home automation with mobile application control (Plant Watering System).



Solstice, Persiaran Bestari, Cyberjaya, Selangor.



01111637599



abugharsa@gmail.com



[www.linkedin.com/in/bashirabu](https://www.linkedin.com/in/bashirabu)

## Education

**Bachelor of Engineering (Honours) Electronics majoring in Computer, Multimedia University, Cyberjaya, Malaysia. 2013-2017**

- Merit international scholarship (50% discount on tuition fees) & dean's list.
- GPA- 3.3

**Foundation in Engineering, Multimedia University, Cyberjaya, Malaysia. 2012-2013**

- Merit international scholarship (50% discount on tuition fees) & dean's list.
- GPA- 3.43

**Ahmed Albahol High School, Misrata, Libya. 2009-2012**

## Certifications

- Front-End Web Development with React
- Meta Database Engineer Professional Certificate
- Meta Front-End Developer Professional Certificate
- Meta Back-End Developer Professional Certificate
- Front-End Web UI Frameworks and Tools

## Skills

- Able to program desktop applications using C++ and C.
- Able to develop mobile applications using Java and C#.
- Able to build and manage database system using SQL.
- Able to write scripts for websites and systems in Python
- Web and Mobile application Development - Excellent
- Simulation using MATLAB and GNU octave- Very Good
- Fluent in English, Arabic, Turkish, and Malay

## Extracurricular activities

- Member of IT club.
- Member of Robotics club.
- Board of Engineers Malaysia – Graduate Engineer
- Arab culture society, Multimedia University Cyberjaya, Malaysia.
- Member of Esports club.



# Marawan Ashraf Fawzy Ahmed Eldeib

Computer Engineer, Multimedia University

## Objective

Work on AI project, it becomes well-known and is widely utilised worldwide.

## Education

### **Bachelor of Engineering (Honours) Electronics majoring in Computer: June 2014-2018**

Multimedia University, Cyberjaya, Malaysia

During my studies:

- ♦ Merit international scholarship (50% discount on tuition fees) & dean's list.
- ♦ GPA- 3.68

### **Foundation in Engineering: November 2013-2014**

Multimedia University, Cyberjaya, Malaysia

During my studies I maintained:

- ♦ Merit international scholarship (25% discount on tuition fees) & dean's list.
- ♦ GPA- 3.31

## **High school: 2018**

International school of Pakistan, khaitan, Kuwait

4A,2B

### Extracurricular activities

- ❖ Cybertron club (Robotics).
- ❖ Member of IEEE Malaysia.
- ❖ Member of IEM Malaysia.
- ❖ Arab culture society, Multimedia University Cyberjaya, Malaysia - Participated in the planning of a few events.
- ❖ Board of Engineers Malaysia – Graduate Engineer

### Certifications

- ▲ IBM Full Stack Software Developer Professional Certificate
- ▲ Meta Database Engineer Professional Certificate
- ▲ Meta Front-End Developer Professional Certificate
- ▲ Meta Back-End Developer Professional Certificate

### Training & Experience

- ♦ Using an 8051 microcontroller and assembly language, create a car parking system.
- ♦ Bookstore Database project using MySQL
- ♦ Capstone project: IoT-based home automation with mobile application control (Home Security)
- ♦ Database Engineer- Internship -intel – Malaysia  
April 2017- June 2017
  - Database engineers design and maintain databases, increase data storage capacity, and troubleshoot database functionality issues. They also review database performance and user reports.
- ♦ Junior software Engineer- Full time- Vodafone – Egypt  
December 2018- December 2020
  - Design, development, bug-fixing, and maintenance of software solutions.
  - Writing reports

- 
- 
- Senior software Engineer -Full time -Sog – Malaysia January 2021-Present
    - Collaborating with clients and cross-functional departments to offer project status updates.
    - Assigning tasks to junior developers such as coding, testing, debugging, and analytics
    - Design, development, bug-fixing, and maintenance of advanced software solutions

### Skills and tools

- ❖ Microsoft Office- Excellent
- ❖ Advanced knowledge of programming languages including JavaScript, HTML5, C++ object oriented, Java, SQL, and PHP.
- ❖ Knowledge of system frameworks including .NET and GitHub.
- ❖ Web and Mobile application Development - Excellent
- ❖ Simulation using MATLAB and GNU octave- Very Good
- ❖ Fluent in English, Arabic, Deutsch, and Bahasa Malay

### Hobbies

- Travel & Volunteering
- Gym
- Learn new languages



# YOUSSEF YASSER SALAHELDIN ELHAMMAMY

---

## About Me

Diligent software engineer with 5+ years of experience in commercial application and software development. In previous roles, slashed downtime by 25% and ensured 98% on-time project completion. Also identified and dealt with a significant process bottleneck that boosted coding efficiency by 35% when resolved.

## Experience

**2019–Present**  
Software Engineer • SOG

- Engineered modern applications with Java, JavaScript, SQL Server, and NoSQL.
- Built innovative microservices and Web Services (incl. SOA/SOAP/REST/XML).
- Utilized Cloud Foundry for efficient building on top of Kubernetes.
- Efficiently deployed and integrated software engineered by team and updated integration/deployment scripts to improve continuous integration practices.
- Liaised with Product Managers to identify minimum viable product requirements and clearly defined feature sets into well-scoped user stories for individual team members.
- Maintained maximum uninterrupted flow of business-critical ops. Cut downtime by 25% and costs of warranty by up to 45%.

[linkedin.com/in/Yousef\\_yss](https://linkedin.com/in/Yousef_yss)

[Yousef32844\\_2@gmail.com](mailto:Yousef32844_2@gmail.com) 

2015–2019

#### Software Development Intern • X-Cite

- Supported X-Cite's testing and engineering processes.
- Confirmed that completed software met client requirements.
- Completed maintenance on existing programs.
- Worked closely with product team on scope of future projects and innovations.
- Carried out continuous identification, measurement, and improvement of processes.
- Identified and dealt with a significant process bottleneck that boosted coding efficiency by 35% when resolved.

#### Education

Bachelor of Electronics majoring in Computer, Multimedia University, Cyberjaya, Malaysia. 2012-2015

- Experienced in software development
- Heavy focus on full stack engineering with modules on both front and back-end processes.
- Built apps for various companies as intern.
- CGPA 3.9/4.0

#### Skills

- Software development
- Adaptability
- Teamwork
- Agile frameworks
- Interpersonal communication skills
- Java, HTML, CSS, SQL Server, NoSQL
- Status tracking frameworks (Jira and Rally)
- Practical knowledge of SQL and database concepts
- Solid knowledge of cybersecurity principles

#### Certifications

- Cloudera Certified Software Developer for Apache Hadoop (CCDH)
- Oracle Java Certifications Associate Professional
- Puppet Labs Puppet Developer Certification

# Project Proposal

## ➤ Problem Definition

One of our business partners, “Fitness Guys”, has requested a mobile application for their members that answers all their basic questions about their physical health. The app will have exclusive features for gym members that will analyze their physical state based on their inputs and return feedback such as routines to stick to improve or maintain their health. The name of the application will be “BodyMath”.

## ➤ Introduction

BodyMath is a fitness application that is designed to analyze the user’s body measurements and assist them based on their inputs. The application includes various fitness calculators such as:

- ❖ Body Mass Index (BMI) Calculator
- ❖ Body Fat Calculator
- ❖ Ideal Weight Calculator
- ❖ Calorie Intake Calculator

These services should give the gym members a better understanding of their own physical state followed by advice on how to improve or maintain it.

The best way to incorporate all these features in a convenient manner on a device that is used by almost the entire population is through a mobile application. The app will be accessible to the user with or without internet connection. The benefits of using the application with an internet connection is the added benefit of auto-filling previous measurements recorded in the database. By assigning each member a username and password, they can synchronize their measurements at any gym branch with their respective accounts.

## ➤ Product Overview

To efficiently and accurately develop the application, the IDE of our choice will be Android studio and to keep our programming clear and concise for future maintenance, we have decided to go with Kotlin as our programming language. To host our database, Firebase Realtime Database will be used for fast and secure hosting. Since both Android and Firebase are owned by the same company (Google), there should be no future inconveniences.

# BMI 25.2

## Overweight

Enter your weight (kg)

80

You cannot leave this bank\*

Enter your height (cm)

178

You cannot leave this bank\*

BMI



## BMI CALCULATOR

HEIGHT

**150** cm



WEIGHT

**60** kg



**CALCULATE**

*Final Preview*

# RC2: Project Plan document





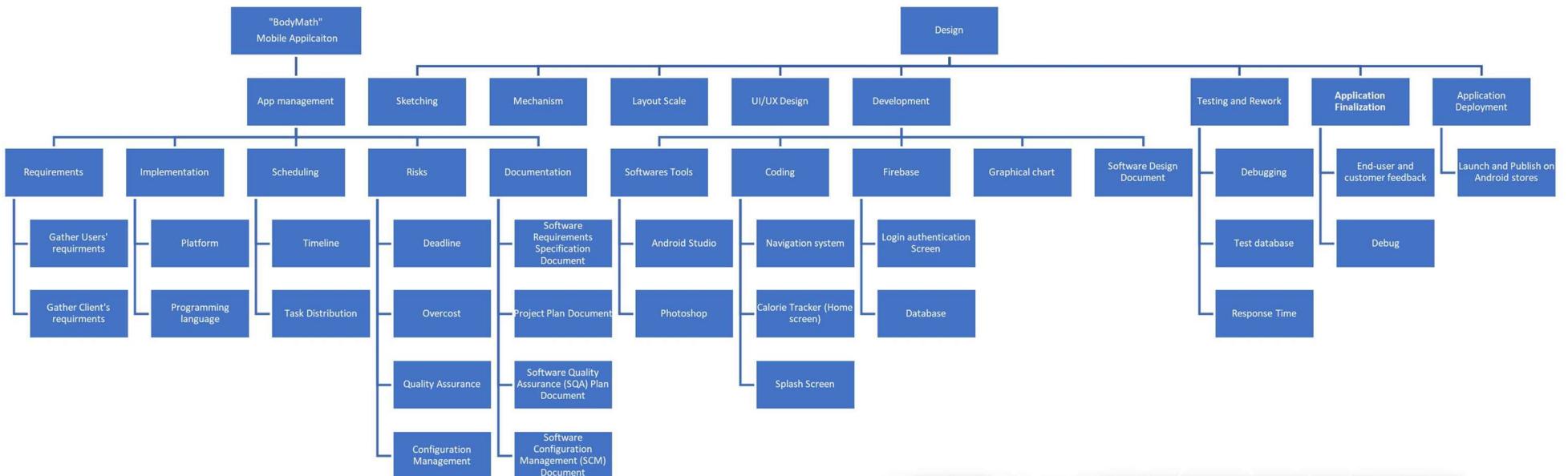
## 1- Objectives:

We intend to hire a team of 4 software developers and spend almost RM 100,000 to release the first beta version of the mobile application (Body math) to our client “Fitness Guys” by January 1, 2023.



<i>Project Title</i>	BodyMath
<i>Project Manager</i>	Ali Salim Karimeh
<i>Project Sponsor</i>	Fitness Guys
<i>Project Start Date</i>	October 1, 2022
<i>Target End Date</i>	January 1, 2023

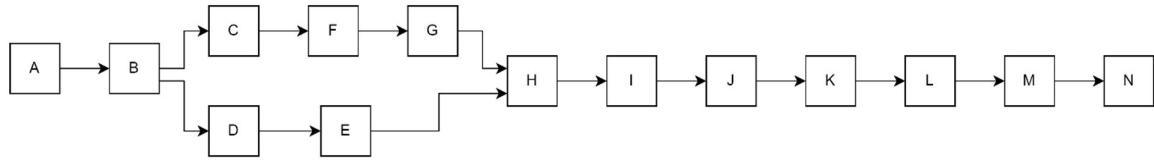
## 2- Work Breakdown structure (WBS):



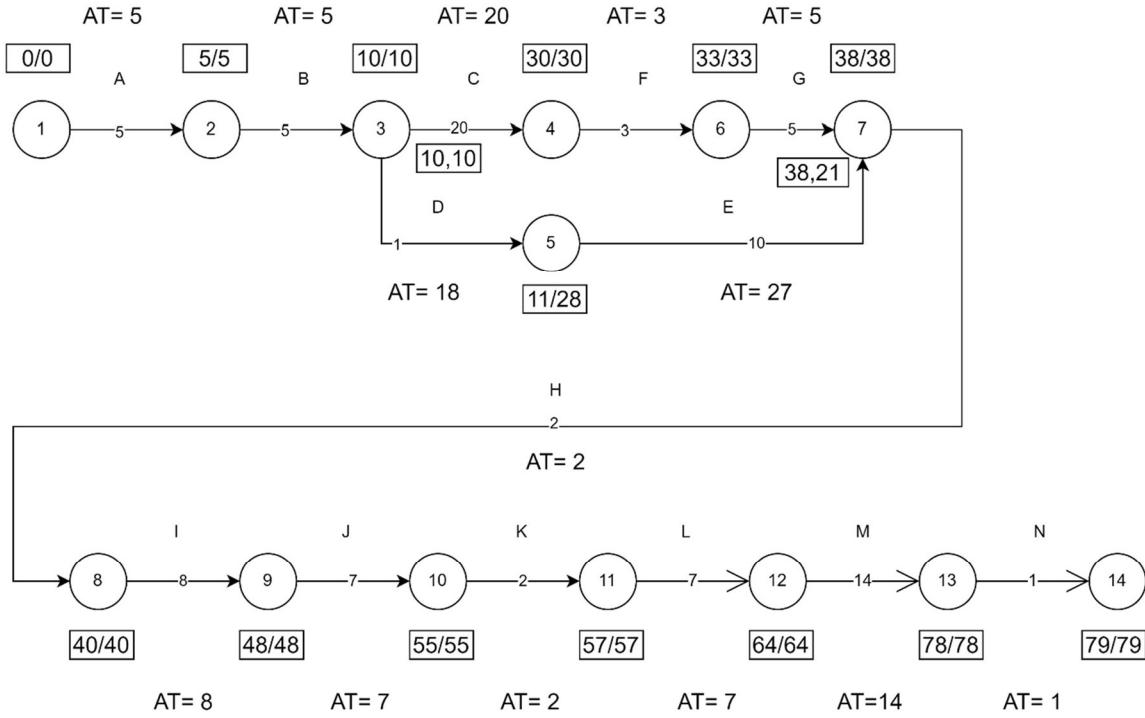
## **Activity Table:**

No.	Activity Description	Activity	Immediate Predecessor	Activity Time (Days)
<b>1</b>	<b>App management</b>			<b>10</b>
1.1	Gather requirements	A	-	5
1.2	Project proposal	B	A	5
<b>2</b>	<b>Design</b>			<b>20</b>
2.1	UI/UX Design, Mechanism Selection, and Layout Establishment with documentation	C	B	<b>20</b>
<b>3</b>	<b>Development</b>			<b>14</b>
3.1	Select Software Tools	D	B	<b>1</b>
3.2	Programming and Connection to Firebase	E	D	<b>10</b>
3.3	Content Creation	F	C	<b>3</b>
<b>4</b>	<b>Testing and Rework</b>			<b>7</b>
4.1	Load app and fix bugs	G	F	5
4.2	Test Firebase	H	E, G	2
<b>5</b>	<b>Application Finalization</b>			<b>15</b>
5.1	End-user and customer feedback	I	H	8
5.2	Debug	J	I	7
<b>6</b>	<b>Application Deployment</b>			<b>2</b>
6.1	Roll out	K	J	2
<b>7</b>	<b>Agile sprint</b>			<b>21</b>
7.1	Create Backlog from feedback	L	K	7
7.2	Implement Changes	M	L	14
<b>8</b>	<b>Application launch</b>			<b>1</b>
8.1	Launch and publish	N	M	1

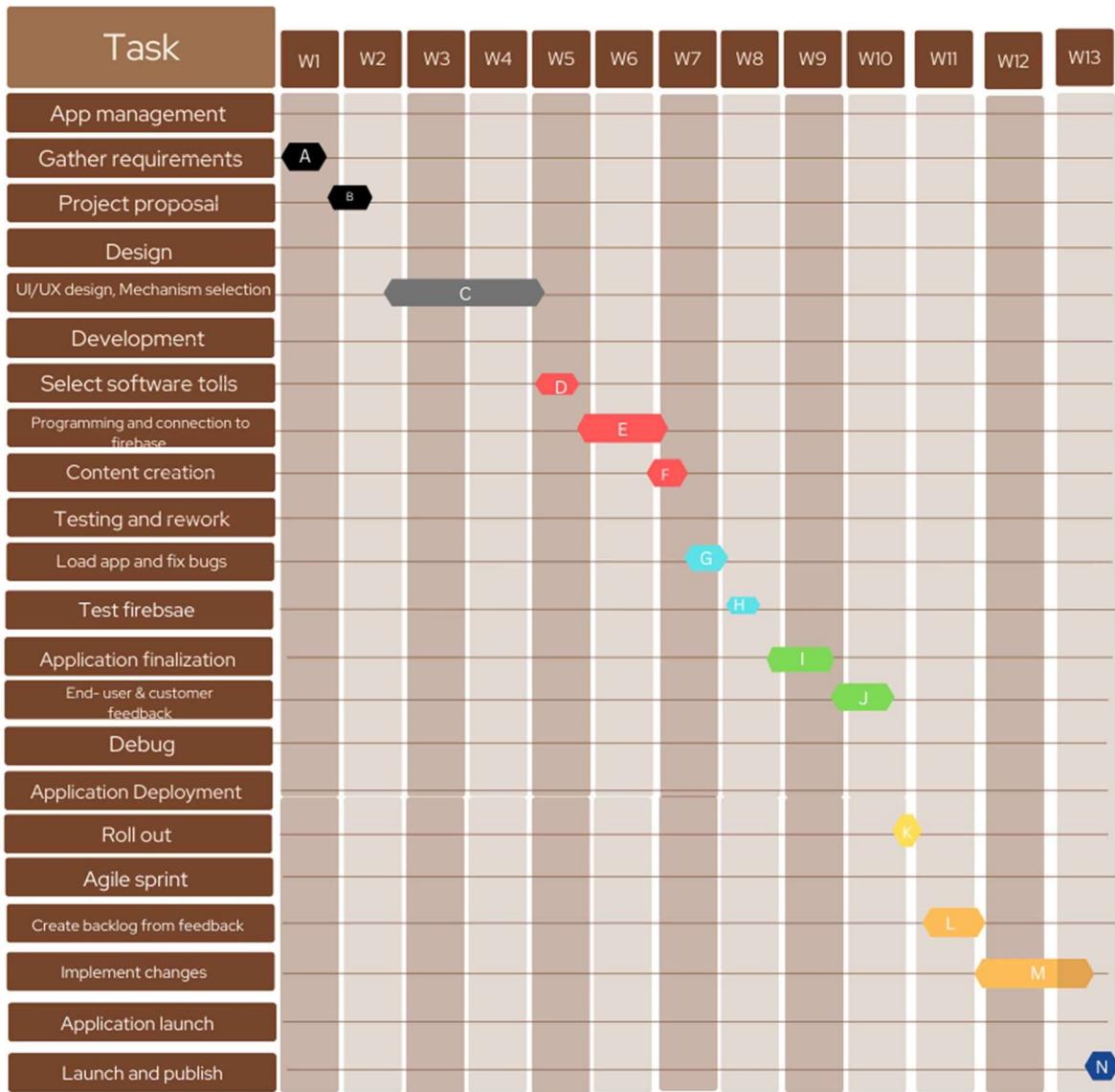
### 3.0 – Activity on Node (AON):



### 4.0 – Activity on Node (AON):



## 5 - Gantt chart



## **6 - Agile model**



The ability to work iteratively is the Agile method's primary area of concentration. Iterative work is completed in short cycles where only a small portion of the overall project is completed at once, hence emphasizing the quick delivery of business value.

Less work wasted is a huge advantage of iterative work. This guarantees that any updates to your project will require the least amount of rework possible. Teams are no longer required to advance too far in the development cycle just to discover that they must start over.

### **1) Strategy Meeting**

A brand-new Agile project begins by identifying a specific objective or business need that it aims to fulfill. Why are you doing what you're planning to accomplish? The fundamental principle you'll keep returning to during the development process is to look at the bigger picture.

### **2) Product Roadmap**

The product owner tries to translate that vision into a product roadmap in this step after verifying a strategy. With an approximate timeline for when each requirement will be developed, this gives your team a high-level picture of the project requirements.

### **3) Release Plan**

The product owner establishes a high-level timetable at this stage to show when usable software will be released. You will need to give more priority to the improvements that will bring you closer to launch first because Agile projects are expected to have frequent releases.

### **4) Sprint Planning**

Sprints are short development cycles in which certain goals and tasks will be completed. Each sprint generally lasts between one to four weeks—each should stay the same length throughout the whole project. At the beginning of a sprint cycle, your team needs to list down backlog items you think you can finish within a specific timeframe so you can create a functional final product.

### **5) Daily Standup**

In every sprint, you need opportunities to ensure that no roadblocks are getting in the way of accomplishing your objectives on time. This is where the so-called “standup” or daily meeting comes into play. A standup—in Agile-speak—is a 15-minute daily meeting where your team gathers to discuss three main points:

- The tasks you accomplished yesterday
- The tasks you are completing right now
- The roadblocks that may stand in your way

### **6) Review Sprints**

After working on each sprint, your team will check if everything has been accomplished as planned. By the end of the cycle, you should have a functioning software product that’s ready for release. At this point, you should review what was completed and show the evaluation to everyone on your team and important stakeholders. Simply put, this stage is an Agile show- and-tell where you discuss the final product.

---

## **7) Sprint Retrospective**

If you want Agile project management to work, each step must have a clear next step. A sprint retrospective can be used to establish this. It's now time to determine what gets done after a sprint is finished and all features have been shown.

## **7.0 - Software Metrics:**

A software metric is a numerical measurement of how much a system component or process possesses a specific software property. Software metrics are crucial since they serve as a gauge for bettering the software development process. Software metrics are used to analyze how the current process model might be improved and how the organization attempts to accomplish the improvement objectives in software process improvement.

We use project metrics to monitor the project's status, identify potential risks, and evaluate the project team's capacity to regulate the quality of the software work products. Size-oriented metrics, function-oriented measurements, object-oriented metrics, and use-case-oriented metrics are some of the different types of project metrics.

Both size-oriented and function-oriented metrics will be used in this project. Size-oriented metrics centre their measurements of project data, costs, and effort on the size of the software that is developed. Function-oriented metrics are based on the functionality that will be provided by this software. It is an indirect measurement that makes use of function points (FP), which can be calculated in the early stages of software development before the exact number of lines of code can be known.

Both metrics have their own set of benefits and drawbacks. Size-oriented metrics have the advantage of being quickly calculated and relevant for projects with similar surroundings, but function-oriented metrics have the advantage of being language independent and based on data available early in the project. Size-oriented metrics, on the other hand, have the problem of being language and programmer dependent, whilst function-oriented metrics have the disadvantage of being more complex to calculate, harder to collect data, and having no physical significance.

---

## 7.1 - Size-Oriented Metrics

Project	Effort (Person-month)	Cost (RM)	kLOC	Doc. (pgs)	Errors	People
A	30	700,000	11.1	965	39	3
B	30	650,000	9.25	1224	50	5
C (Current)	6	300,000	2.5	200	10	4

The formulae are:

- ✚ Productivity = size / effort = kLOC / person-month
- ✚ Quality = Errors / Size = Errors / kLOC
- ✚ Cost = Total cost / Size = RM / kLOC
- ✚ Documentation = Pages / Size
- ✚ Total project duration = Effort / no. of members

Project	Productivity	Quality	Cost	Documentation	Duration
A	0.370	3.514	63063.06	86.93694	10
B	0.308	5.405	70270.27	132.3243	6
C (Current)	0.416	4.00	120000	80	1.5

Project C is more productive than the other two projects in terms of productivity, demonstrating resource efficiency while maintaining good quality.

## **7.2 - Function-Oriented Metrics:**

Parameter	Count	Simple	Average	Complex
<b>Inputs</b>	5	3	4	6
<b>Outputs</b>	8	4	5	7
<b>Inquiries</b>	7	3	4	6
<b>Files</b>	10	7	10	15
<b>Interfaces</b>	3	5	7	10

### **Calculating the raw FP value:**

Parameter	Simple	Average	Complex
<b>Inputs x Count</b>	$5 \times 3 = 15$	$5 \times 4 = 20$	$5 \times 6 = 30$
<b>Outputs x Count</b>	$8 \times 4 = 32$	$8 \times 5 = 40$	$8 \times 7 = 56$
<b>Inquiries x Count</b>	$7 \times 3 = 21$	$7 \times 4 = 28$	$7 \times 6 = 42$
<b>Files x Count</b>	$10 \times 7 = 70$	$10 \times 10 = 100$	$10 \times 15 = 150$
<b>Interfaces x Count</b>	$3 \times 5 = 15$	$3 \times 7 = 21$	$3 \times 10 = 30$
<b>Raw FP</b>	<b>153</b>	<b>209</b>	<b>308</b>

For each complexity adjustment factor, a rating on a scale of 0 to 5 is given, as shown below:

Complexity adjustment factors	Rate
F1: Does the system require reliable backup and recovery?	4 - Significant
F2: Are data communications required?	3 - Average
F3: Are there distributed processing functions?	2 - Moderate
F4: Is performance critical?	4 - Significant
F5: Will the system run in an existing, heavily utilized operational environment?	5 - Essential
F6: Does the system require on-line data entry?	4 - Significant
F7: Does the on-line data entry require the input transaction to be built over multiple screens or operations?	0 - No influence
F8: Are the master files updated on-line?	3 - Average
F9: Are the inputs, outputs, files, or inquiries complex?	1 - Incidental
F10: Is the internal processing complex?	2 - Moderate
F11: Is the code designed to be reusable?	4 - Significant
F12: Are conversion and installation included in the design?	2 - Moderate
F13: Is the system designed for multiple installations in different organizations?	3 - Average
F14: Is the application designed to facilitate change and ease of use by the user?	5 - Essential
SUM (CAV)	42

#### **Rate Level:**

1 – Incidental 2 – Moderate 3 – Average 4 – Significant 5 – Essential

Calculating the FP of each Simple, Average & Complex using the formula below:

$$FP = \text{Raw FP} \times [0.65 + (0.01 \times CAV)]$$

- Simple:  $FP = 153 \times [0.65 + (0.01 \times 42)] = 163.71$
- Average:  $FP = 209 \times [0.65 + (0.01 \times 42)] = 223.63$
- Complex:  $FP = 308 \times [0.65 + (0.01 \times 42)] = 329.56$

Calculating the productivity, quality, cost, and documentation of the following projects:

- Productivity = Function Point/Effort
- Quality = Errors/Function Point
- Cost = Cost / Function Point
- Documentation = Pages/Function Point

### Project A

Metrics	Simple	Average	Complex
<b>Productivity</b>	$163.71/30 = 5.46$	$223.63/30 = 7.45$	$329.56/30 = 10.99$
<b>Quality</b>	$39/163.71 = 0.24$	$39/223.63 = 0.174$	$39/329.56 = 0.12$
<b>Cost</b>	$700,000/163.71 = 4275.85$	$700,000/223.63 = 3130.17$	$700,000/329.56 = 2124.04$
<b>Documentation</b>	$965/163.71 = 5.89$	$965/223.63 = 4.32$	$965/329.56 = 2.93$

### Project B

Metrics	Simple	Average	Complex
<b>Productivity</b>	$163.71/30 = 5.46$	$223.63/30 = 7.45$	$329.56/30 = 10.99$
<b>Quality</b>	$50/163.71 = 0.305$	$50/223.63 = 0.22$	$50/329.56 = 0.1517$
<b>Cost</b>	$650,000/163.71 = 3970.44$	$650,000/223.63 = 2906.58$	$650,000/329.56 = 1972.32$
<b>Documentation</b>	$1224/163.71 = 7.47$	$1224/223.63 = 5.473$	$1224/329.56 = 3.714$

### Project C

Metrics	Simple	Average	Complex
<b>Productivity</b>	$163.71/6 = 27.285$	$223.63/6 = 37.27$	$329.56/6 = 54.92$
<b>Quality</b>	$10/163.71 = 0.061$	$10/223.63 = 0.044$	$10/329.56 = 0.0303$
<b>Cost</b>	$300,000/163.71 = 1832.5$	$300,000/223.63 = 1341.5$	$300,000/329.56 = 910.30$
<b>Documentation</b>	$200/163.71 = 1.22$	$200/223.63 = 0.89$	$200/329.56 = 0.60$

Finally, it is shown that our current project C is more productive than projects A and B. Furthermore, according to the table, the cost of developing project C is the lowest, indicating that the project's resources are being used efficiently.

## **Cost estimation of the project:**

We have 4 developers working on the application. Each developer salary is RM6000.

Salaries for the 4 employees per month =  $4 * 6000 = \text{RM}24000$ .

Project duration is almost 3.5 months.

Total salaries for the whole project:  $3.5 * 24000 = \text{RM}84000$ .

Backend and hosting price: RM5000.

Testing price: RM10000.

Total costs:  $84000 + 5000 + 10000 = \text{RM}99,000$ .

### **For More details:**

Selling price: 300,000RM.

Project cost: 99,000RM Profit: SP - CP =  $300,000 - 99,000 = \text{RM } 201,000$ .

Profit Percentage:  $(\text{Profit} / \text{CP}) * 100\% = (201,000 / 300,000) * 100\% = 67\%$ .

**Our company has a contract with a Fitness company that states that we are responsible for maintenance and troubleshooting for a period of one year, starting from the day that the app is published.**

Type of Work	Backend Development (hours)	Mobile Application (hours)
Features	150	290
UI/UX design	-	125
Frameworks & libraries	133	200
Project management	90	180
Quality assurance	110	250
Total time	483	1045

## **8.0 - Project Risk Management**



A risk is a potential problem – it might happen, and it might not.

### **Step 1: Possible risk identification**

The first step in the risk management process avoiding or controlling them when possible. There are different kinds of risks associated with our projects (technical & technology risks, customer Risks and business risks).



**Project risks:** They pose a threat to the project plan, if they actually happen, it is likely that the project schedule will slip and that costs will rise as well.

**Technical & Technology Risks (TTR):** They threaten the quality and timeliness of the software to be produced, if they occur, implementation may become difficult or impossible.

- For the time being, there is no need for additional algorithms because the resources already are sufficient. With the technologies at hand, this project can be finished. Additionally, the staff members have prior experience with the technologies required by this software.
- Particularly size-oriented metrics and function-oriented metrics needed to be gathered.

**Customer Risks (CR):** The project's software requirements specification reflected the vision or concept that both parties had mutually agreed upon.

- The customer is well-aware of the requirements, but does he comprehend the software development process? Despite being knowledgeable about the product field, is he prepared to put in time in formal requirements collecting meetings to determine the project's scope and take part in technical reviews?

**Business Risks (BR):** They pose a risk to the software's viability and, if they happen in the future, put the project or product in risk.

- Given that the project's scope is maintained, and the original plan has not changed, the date for delivery, which is January 1st, seems appropriate.

- For the client to use the built mobile application, clear and informative documentation should be created. This is because the purchaser will have full access to the mobile application settings and database, thus it must be user-friendly.

- The PDPA states that users must be protected in order to prevent any abuse of the storage or processing of personal data of individuals, public and private sectors in Malaysia for commercial activities.

- Fitness Guys will be the only recipient of the software, and the price will be as
-

agreed upon with the customer. The company's revenue will rise as a result. The project's earnings will be used to pay the workers. Customer dissatisfaction from a delayed application delivery could affect negatively on the sales .

---

## **Step 2: Risk Table**

Risk Summary	Risk Category	Probability (%)	Impact (1-4)
Delay in Delivery	BR	10	1
Changes in Requirements	CR	14	2
Database instability	TTR	20	1
Poor Quality Documentation	BR	60	3
Deviation from Software Engineering Standards	TTR	9	2
Poor Comments in Code	TTR	5	4
Additional funding needed	BR	7	3

### **Impact Level:**

1 – Catastrophic    2 – Critical    3 - Marginal    4 – Negligible

---

### **Step 3: Risk Mitigation, Monitoring and Management**



#### **Delay in Delivery**

- **Mitigation :** Assigning tasks to the team while taking into account the number of individuals available for the project as well as their talents, strengths, and weaknesses will be one of the procedures we take to estimate the project scope and determine the deadline. It is critical that team members communicate their progress and deal with any issues they may encounter.
- **Monitoring :** During the planning phase, a schedule was created for project monitoring. The team will carefully follow plan, and if delivery is likely to be delayed, it will be signaled by running behind schedule.
- **Management:** When the worst-case scenario happens, that the project cannot be finished within the time allocated, the only option would be to ask the customer for an extension, which could be negatively affecting the company's reputation and the relationships with its clients.

## **Poor Quality Documentation**

- **Mitigation:** The manager or documentation team should be in frequent communication with the development team. The project manager must keep an eye on the establishment of document standards and methods to make sure that they are created on schedule. Meetings should be conducted frequently to discuss the paperwork. Any questions should be answered right away. If a developer discovers through these sessions that a feature is missing from the documentation, it may be added after careful study. Additionally, the testing team needs to run tests using this documentation to inform the documentation team of the software's documentation's quality.
- **Monitoring:** The development and testing teams should continually be on the lookout for any topics about the application that have not been covered.
- **Management:** If low quality is found, the entire team should make an effort to raise the standard of the documentation by adding any necessary subjects or eliminating any unnecessary ones.

## **Additional funding needed**

- **Mitigation:** In the future, there can be a need for extra cash as a result of unpredictable events or poor planning. There might be a reserve fund set aside as a precaution for such unforeseen events.
  - **Monitoring:** The finance team should continually keep an eye on the project's cash flow and notify the project team if it exceeds the budget amount.
  - **Management:** If additional funding is needed, the project manager should meet with the project sponsor and make the request. However, given that the team has successfully completed a few projects of a similar type, this is highly unlikely to occur.
-

## **Changes in Requirements**

- **Mitigation:** There will be multiple client meetings during the project to improve communication understanding between the two sides in an effort to reduce this risk. This makes certain that our product will stand up to client expectations.
- **Monitoring:** Each time a meeting is held with the client, the meeting minutes should be appropriately prepared and handled on both sides. The meeting minutes should then be approved by both parties prior to the start of the subsequent meeting. Customers and developers will be well-informed about the project using this strategy.
- **Management:** If the requirements change, appropriate action should be done to remedy the situation as fast and effectively as feasible. To resolve this issue, a meeting should be called between the project team and the customer.

## **Database instability**

- **Mitigation:** The developer who oversees the database should always check for errors (Testing stage). When issues are identified, they should be reported to the team and remedied as soon as possible.
- **Monitoring:** Without the developer who maintains the database's control, any user who accesses it shouldn't make any significant changes. Anytime an error is discovered, it must be reported to the team.
- **Management:** The team should get together to investigate the problem and develop potential remedies if the database is found to be unstable.

## **Deviation from Software Engineering Standards**

- **Mitigation:** It is highly unlikely to occur since the team working on this project is highly qualified and has a deep understanding of software engineering standards.
- **Monitoring:** Technical reviews will be conducted on a regular basis to ensure that the documentation and the actual project are in sync. To ensure the team's compliance with software engineering standards, all necessary documentation should be handed out as precisely as possible.
- **Management:** To address this issue, the project must be updated so that it meets the standards. Technical reviews can be used to identify how to bring the project up to modern standards.

## **Poor Comments in Code**

- **Mitigation:** The lead programmer should establish certain commenting guidelines to ensure high-quality code is generated.
  - **Monitoring:** Regular code reviews will help the reviewers decide whether the comments made for the code are suitable. This guarantees the creation of high-quality code.
  - **Management:** If the code quality is found to be poor, programmers should take the required steps to improve the code comments. Constantly monitoring the code will limit the probability of improper code comments. Programmers could easily fix this problem by simply modifying or introducing additional comments.
-

## **Step 4: Risk Map**

Managers may find it challenging to comprehend all the risks that their organization faces. A real discussion can easily go from one risk to risk and even making a list makes it hard to truly analyze which risks are significant, which may be accepted and how risks relate to one another.

A risk map is one technique to utilize data visualization to make risks simpler to grasp. A risk map shows the risks that an organization is exposed to graphically. The complete map can be thought of as the organization's risk universe, or the collection of all the risks that the organization faces. Individual risks are detected on the risk map and their location on the map is determined. A risk's additional elements, such as its type or the department in charge of managing it, might be represented by other visual attributes, such as color or size.

Legend					
Risk Rating	Low	Moderate	Significant	High	Extreme

Risk Map					
↑ Level of Likelihood →	Almost certain				
	Likely		BR2		
	Possible			TTR1	BR1
	Unlikely	TTR3		CR1, BR3	
	Rare			TTR2	
		4- Negligible	3- Marginal	2- Critical	1- Catastrophic
Level of Impact →					

Tasks		Risk Level	Level of Likelihood	Level of Impact
<b>Business risks (BR)</b>				
BR1	Delay in Delivery	High	Possible	Catastrophic
BR2	Poor quality documentation	High	Likely	Marginal
BR3	Additional funding needed	Moderate	Unlikely	Critical
<b>Customer Risks (CR)</b>				
CR1	Changes in requirements	Significant	Unlikely	Critical
<b>Technical &amp; technology risks</b>				
TTR1	Database instability	High	Possible	Critical
TTR2	Deviation from software engineering standards	Low	Rare	Critical
TTR3	Poor comments in code	Low	Unlikely	Negligible

Probability Rescale Table	
Pre-defined probability (%)	Level of Likelihood
>=70	Almost certain
>=40	Likely
<=30	Possible
<=20	Unlikely
<=10	Rare

#### Level of Impact → :

1 – Catastrophic    2 – Critical    3 - Marginal    4 – Negligible

## **Conclusion**

**Project Risk Rating:** Moderate Risk

(0 Extreme , 1 High, 2 Significant, 3 Moderate, 1 Low)

Recommendations that the project manager assess the project scope and set the deadline by assigning tasks to the team while keeping the team size in mind. The documentation team will collaborate with the development and testing teams closely. Meetings should be held on a regular basis. Checking for database errors on a regular basis and, if any, resolving them as soon as possible and setting up a reserve fund as a precaution.



## RC3: Software Quality Assurance (SQA) Plan Document



## **Foreword**

This document is a “BodyMath” Mobile Application Project controlled document and adheres to IEEE 730-2002, the IEEE Standard for Software Quality Assurance Plans. Changes to this document require prior approval of the “BodyMath” Mobile Application Project Configuration Control Board (CCB). The Gym Mobile Application Software Quality Assurance Manager must receive any suggested changes, together with any supporting documentation that explains why the change is necessary.

Questions or comments concerning this document should be addressed to the Assurance Management Office:

BodyMath, Marawan Ashraf Eldeib

Building 5 , Room 7

Mail stop 17:30

Shaftsbury Square

Cyberjaya, Cyber 6,

63000 Persiaran

Multimedia, Malaysia

Or via Email :[SQA@sog.com](mailto:SQA@sog.com)

## Signature Page

Prepared by,



-----  
Marawan Ashraf Eldeib

16th November 2022

Software Quality Assurance Manager

Reviewed by,



-----  
Bashir Tawfig Abugharsa

18th November 2022

Lead Programmer

Reviewed by,



-----  
Youssef Yasser Elhammamy

21st November 2022

Lead Designer

Approved by,



-----  
Ali Salim Karimeh

23rd November 2022

Project Manager

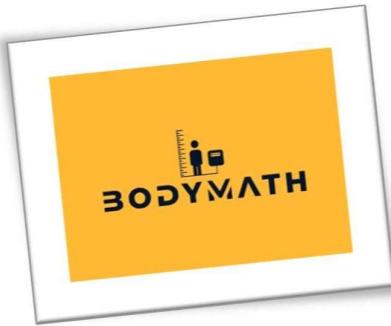
Gym Mobile Application Project Document

DOCUMENT CHANGE RECORD      Sheet: 1 of 1

REV/ VER LEVEL	DESCRIPTION OF CHANGE	APPROVED BY	DATE APPROVED
1.0	Initial Publication of SQA document	Ali Salim Karimeh	23rd November 2022

## **1.0 - Introduction**

The purpose of this Software Quality Assurance (SQA) Plan is to establish the goals, processes, and responsibilities required to implement effective quality assurance functions for the “BodyMath” Mobile Application Project.



The “BodyMath” Mobile Application Software Quality Assurance Plan provides the framework to ensure a consistent approach to software quality assurance throughout the project life cycle. The systematic monitoring of “BodyMath” Mobile Application products, processes, and services will be evaluated to ensure they meet requirements and comply with mobile

Application policies, standards, and procedures, as well as applicable Institute of Electrical and Electronic Engineers (IEEE) standards.

This plan covers SQA activities throughout the implementation phases of the Appointment “BodyMath” Mobile Application mission.

This document explains the Software Quality Assurance Plan (SQA) for “BodyMath” Mobile Application. This project will create and develop a mobile application based on android. This project will come out with a mobile application that calculates the user's Body Mass Index (BMI), calories , ideal weight and bodyfat and saves it in the database.

## **2.0 – Management**



This section will summarize the roles and responsibilities and the software quality task to be performed by Software Quality Assurance Manager and relative personals.

The “BodyMath” Mobile Application project is supported by numerous personnel from different departments.

The relevant personnel or entities are stated below:

- **Ali Salim Karimeh** - Project Manager
- **Marawan Ashraf Eldeib** - Software Quality Assurance Manager
- **Youssef Yasser Elhammamy** - Lead Designer
- **Bashir Tawfig Abugharsa**- Lead Programmer

All these personals are responsible for managing the objectives prescribed by “BodyMath” Mobile Application Project Plan.

## **3.0 - Roles and responsibilities of Software Quality Assurance Department**



The following are all the Software Quality Assurance Department's responsibilities:

### **Software Quality Assurance Manager**

The SQA manager's responsibilities watches for the entire team and personally supervises the software testing processes.

Furthermore, provide software status updates on a weekly and quarterly basis to grant evaluations as a general guidance to his SQ professionals who oversee software quality activities. In addition to that, he supports SQ Personnel in resolving any noncompliances, problems, or risks discovered during software quality operations and then reports any everything to project management.

### **Software Quality Personnel**

The responsibilities of the SQA Personnel include developing and maintaining metrics in addition to creating and maintaining the project's software quality assurance plan and task schedule. Conducting software assurance assignments in coordination with the Safety and Reliability teams. By performing rigorous testing, they guarantee a quality product to the end- user.

Moreover, using objective requirements, they will complete the process and product evaluations indicated in this plan. They design and execute systems to check for issues by perform manual and automated testing. Identify, isolate, and track bugs throughout testing.

Furthermore, identify any potential problems that users might encounter and how he should deal with them. Then, share the evaluations results with stakeholders

Finally, figure out how to remedy or fix any noncompliance and highlight any issues that cannot be rectified within the scope of the project. Then, review them and recognize any new insights discovered that can help in the development of methods for upcoming releases or products.

## **4.0 - Details of the software quality assurance (SQA)**

**To summarize, the SQA will carry out the following activities:**

- ▲ Utilization of Technical Techniques.
- ▲ Software Testing and conduct technical reviews.
- ▲ Enforcement of Standards (ISO 9000)
- ▲ Change Management
- ▲ Documentation, report and maintaining

### **4.1 - Utilization of Technical Techniques**

There are **2 main types** of techniques or strategies that can be adopted during this activity:

#### **i. Modeling**

The model, such as a UML diagram, will be used by SQA personnel to concentrate on what the system should and shouldn't do. Using these models makes it simpler and more precise to grasp a system's operation and other details while analyzing software needs. The approach enables SQA to assess the thoroughness, accuracy, and consistency of the specifications and designs.

#### **ii. Partitioning**

When an error or problem happens, it is too complicated to consider it as a whole system because a whole system is made up of numerous components such as software, database, and so on. As a result, SQA should determine and divide the problem into smaller components so that designers and programmers can grasp it.

---

## **Other**

A designer should employ approaches such as frequent interviews to produce high quality requirements using information acquired to gain a deeper understanding of what the client wants.

### **4.2 - Software testing and conduct technical reviews**

A test case design will be created to be applied on the software to uncover errors in the software development process. The SQA team will next utilize this test case to evaluate and audit the software consistently. Audits can be scheduled or unscheduled. TSQA team will set up various checkpoints for technical audit and reviews for both scheduled and unscheduled.

Scheduled audits are audits that are performed at the completion and beginning of each development stage in accordance with the WBS (work breakdown structure), AOA, and AON. An audit report must include all views, opinions, and suggestions. Unscheduled audits imply that the SQA will audit the system at random to ensure that all corrections and debugging are carried out, based on the report generated on each scheduled audit.

Following that, a monthly SQA status report will be prepared and submitted for technical evaluation. Before the mobile application can be deployed to the Client's users, technical reviews are used to analyze or appraise the quality and design of the prototype. The technical staff is consulted at meetings (physical or virtual) to discuss the prototype's design and software quality. This evaluation aids in finding faults that non-technical staff cannot notice (SQA team).

---

### **4.3 - Enforcement of Standards (ISO 9000 , 730-2002 - IEEE Standard for Software Quality Assurance Plans)**



ISO 9000 and 730-2002 – IEEE should be followed throughout the mobile application development process. ISO 9000 and 730-2002 – IEEE is a set of standards used to validate the quality process in the development of an organization. ISO 9000 is defined as a set of international standards on quality management and quality assurance developed to help companies effectively document the quality system elements needed to maintain an efficient quality system. First and foremost, it's customer focus. This means the project should

always align the objectives with customers' needs and expectations. This can be accomplished through meeting customer requirements, maintaining frequent meetings and communication with the client, performing client-side surveys, and so on. Leaders from each team from SOG should define a vision and direction for their teams according to the goals. Leaders should also interact with their subordinates to acknowledge and provide feedback on employee contributions. Leaders must also guarantee that employees' strengths are recognized and utilized. We must use all our personnel and brainpower to maximize the quality of products and services we provide to our customers. This standard assists in determining the content and preparation of Software Quality Assurance Plans and provides a standard against which such plans can be prepared and assessed. It is directed toward the development and maintenance of software.

### **4.4 - Control of change management**

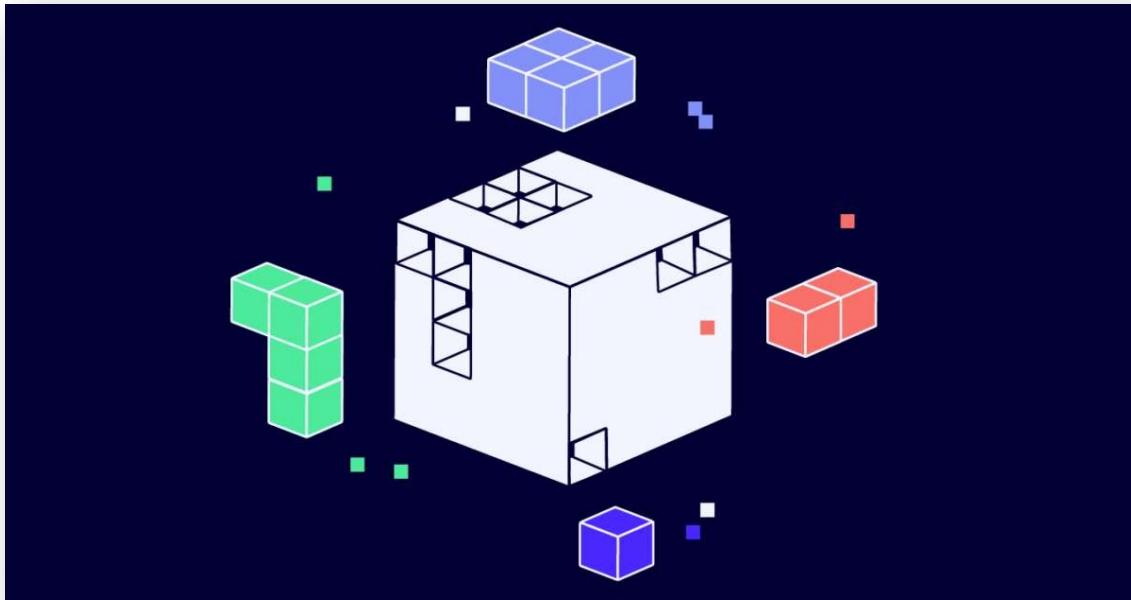
Changes are managed in accordance with software configuration management (SCM). By incorporating all members of the project team and also stakeholders in selecting modifications to software configurations, a shared responsibility model is implemented. The SCM specifies who is involved and what their roles are. The Configuration Process, which includes the establishment of a Change Control Authority (CCA) is the

formal channel for adding any further software changes. Any change request will necessitate the submission of an Engineering Change Proposal (ECP) for CCA approval. To apply the improvements, the CCA will issue an Engineering Change Order (ECO) to the relevant department. Once the ECO is issued, the require a verification process involving Status Accounting and Configuration Auditing. In Status Accounting, all the changes, ECO, and Audit reports are recorded. The Configuration Audit ensures proper implementation of the changes, including auditing of changes and its Formal Technical Review (FTR).

#### **4.5 - Record, keeping and reporting**



The developers are required to perform weekly software metrics calculation, like the Size- Oriented Metrics and Function-Oriented Metrics. These metrics will be submitted to the Software Quality Assurance Manager. These test, audit, review report results will all be documented . These documents will be presented to stakeholders, when needed or during Annual Stakeholder Meeting.



# **RC4: Software Configuration Management (SCM) Document**



---

Gym Mobile Application Project Document

DOCUMENT CHANGE RECORD      Sheet: 1 of 1

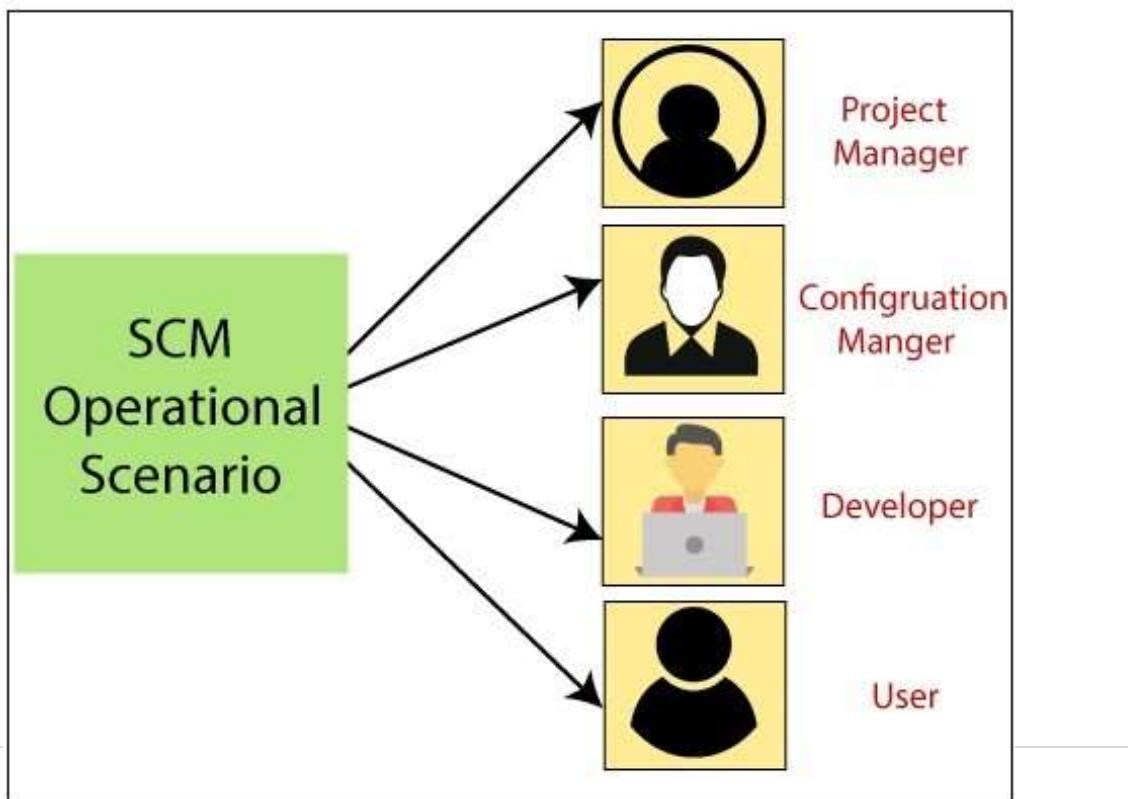
REV/ VER LEVEL	DESCRIPTION OF CHANGE	APPROVED BY	DATE APPROVED
1.0	Initial Publication of SCM document	Ali Salim Karimeh	24th November 2022

# **1 - Identification**

## **1.1 - Document overview**

Software configuration management is the discipline of managing the evolution of complex software systems [IEEE STD 1987]. It is also defined as 'the process of controlling and monitoring change to work products' [Herbert 1999]. Software Configuration Management (SCM) is a method in software engineering that systematically manages, organizes, and controls changes in documents, codes, and other entities across the Software Development Life Cycle. The main objective is to boost production while reducing errors.

The software configuration management plan for the "BodyMath" Mobile Application Software is given in this document. Software configuration management is an approach for the project team to track and monitor changes to the software. There are multiple layers to the management process, including identification, change control, version control, configuration auditing, and status reporting. The layers will be thoroughly highlighted in this document.



## **1.2 - Abbreviations and Glossary**

### **1.2.1 - Abbreviations**

<b>SCM</b>	Software Configuration Management
<b>CSCI</b>	Computer Software Configuration Item
<b>SCI</b>	Software Configuration Item
<b>ECP</b>	Engineering Change Proposal
<b>ECO</b>	Engineering Change Order
<b>CCA</b>	Configuration Control Authority
<b>FTR</b>	Formal Technical Review
<b>SE</b>	Software Engineering
<b>PDR</b>	Preliminary Design Review
<b>CDR</b>	Critical Design Review
<b>TRR</b>	Test Readiness Review

---

## **1.2.2 - Glossary**

<b>Baseline</b>	A specification or product that has undergone formal review and agreement, is then used as the foundation for additional development, and changes can only be made in accordance with formal change control procedures.
<b>SCI</b>	A set of hardware, software, or both that has been given a configuration management designation and is handled as a single entity during the configuration management process.
<b>Identification</b>	Separately naming each CSCI and then organizing it in the SCM repository using an object-oriented approach.
<b>Change Control</b>	A procedure that guarantees the quality and consistency of configuration object modifications.
<b>Version</b>	A system instance that differs from other instances in terms of functionality, performance, and so on.
<b>Revision</b>	Change to a version that only remedies design/code issues and has no impact on the documented functionality.
<b>Release</b>	Official release of the approved version.
<b>Configuration Audit</b>	A procedure used to ensure that a Configuration Identification, or a group of CIs that comprise a baseline, meets a predefined standard or criteria. This covers audits of both functional and physical configuration.

## **1.3 References**

### **1.3.1 Project References**

#	Document Identifier	Document Title
R1	C215	Customer Requirements
R2	M155	Mobile Application Essentials

### **1.3.2 Standard and regulatory References**

#	Document Identifier	Document Title
STD1	M25	Mobile Application standards (Kotlin)
STD2	O42	Operating Systems Standards
STD3	U27	User Data Standards
STD4	D33	Database Standards

---

## **2 - Organization**

The software configuration is managed by members of the project, with specific tools.

The responsibilities are shared between the:

- i. Project Manager/Documentation Manager
- ii. Software Configuration Manager
- iii. Software Quality Manager
- iv. Software Engineer
- v. Software Designer
- vi. Client
- vii. Customer service

### **2.1 - Activities and responsibilities:**

The following table lists the duties connected to the roles in managing the configuration of software projects:

SCM is the person in charge of locating the configuration items, installing the tool for the bug repository, and setting up the database during the project's setup phase. Set up the database and install the software configuration repository tool. Organize and manage the reference area. Ensure the configuration processes are stable.

<b>Activities during the project lifecycle</b>	<b>Person responsible</b>
Export elements for modification, test, or delivery.	SCM
Set under control validated components.	SCM
Create the version and the delivery document.	SCM
Approve reference configurations.	Project manager
Check the version that will be delivered and approve release.	Project manager
Backup spaces.	SCM
Performing configuration audits.	Quality Manager
Examine configuration files.	Quality Manager
Archive reference version.	SCM

<b>Management activities</b>	<b>Person responsible</b>
Manage versions and archives.	SCM
Manage configuration records.	SCM
Publish data and reports.	SCM
Manage reference space and its access control list.	SCM
Manage backup spaces and media archive.	SCM
Manage quality reports.	Quality Manager

### **2.1.1 - Decisions process and responsibilities**

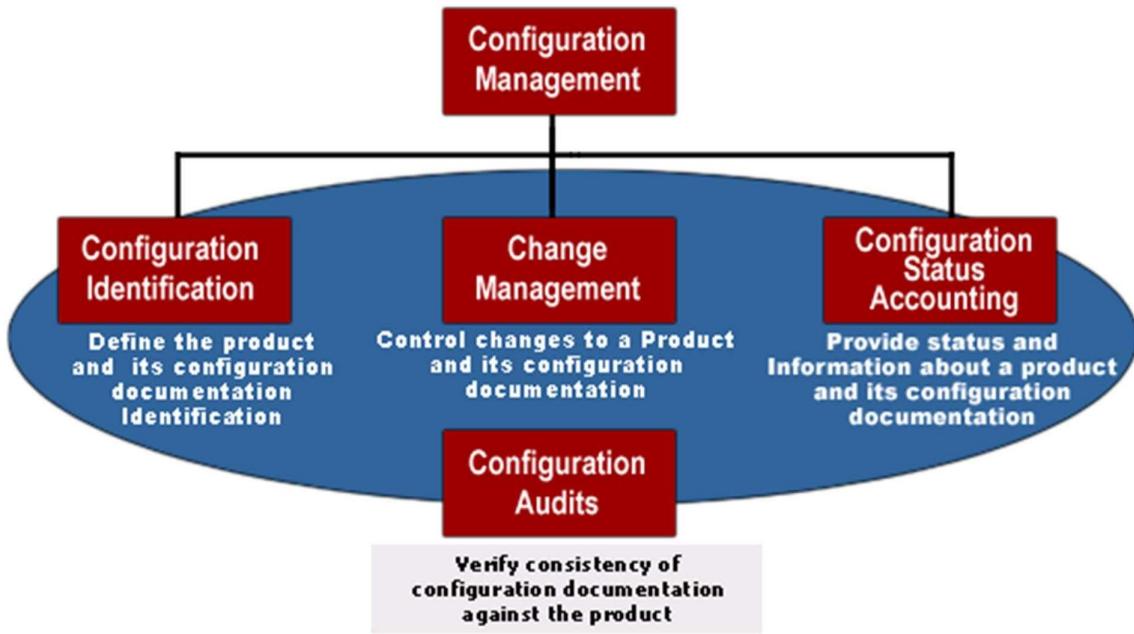
The following is a list of the duties dependent upon the person in control during reviews, audits, and approvals:

Upon completion of a project activity:

<b>Activities</b>	<b>Person Responsible</b>
Consolidate the configuration.	SCM
Show the components' configuration states that were affected by the activity.	SCM
Show the components affected by the activity in their current documented state.	SCM

During an assessment of the configuration management process:

<b>Activities</b>	<b>Person Responsible</b>
Perform a process audit for configuration management	Project Manager
Display the report for the configuration management procedure.	SCM
Show the configuration management process's quality records.	Quality Manager
Display the documentation management process's records.	SCM



### **3 - Configuration identification**

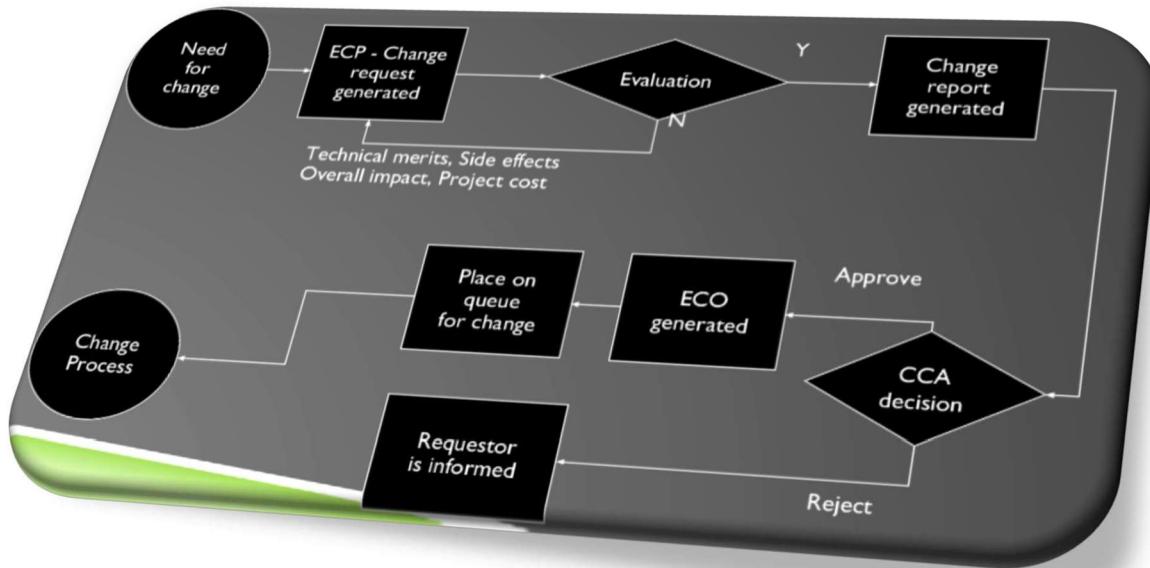
The SCM process' main layer is found here. The identification layer of the software configuration management will give each CSCI a name using an object-oriented methodology. The SCM repository will subsequently be used to organize them. This layer has several duties, including defining the process for identifying SCIs, defining the process for depicting relationships between SCIs, defining the process for identifying SCIs, defining the process for establishing baselines and the associated SCIs, and defining the process for acquiring SCIs in the project repository.

### **4 - Configuration control**

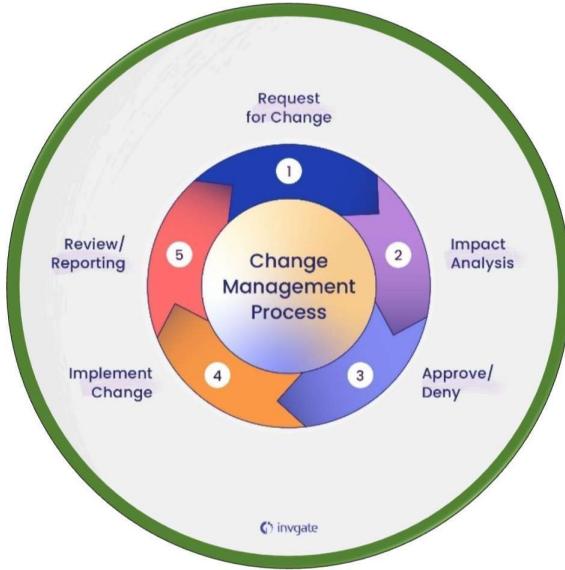
A subsystem of the overall project management system. It is a collection of formal documented procedures used to apply technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of a product, result, service, or component; control any changes to such characteristics; record and report each change and its implementation status; and support the audit of the products, results, or components to verify conformance to requirements.

Configuration Control is a procedural activity that ensures quality and consistency is maintained as modifications are made to the configuration object. It oversees modifications to various SCI configurations to define and request a change, an ECP, a document containing proposed modifications, reason(s), baselines & SCIs affected, cost, scheduling impacts, and procedures for change, will be used. After reviewing the submitted proposal, the CCA determines whether to accept it or reject it.

The SCM repository is used to get the baselined CSCI. Only authorized software engineers are allowed access to and the ability to edit the configuration object thanks to access control. The use of synchronization control prevents concurrent modifications from being overwritten.



## **4.1 - Change Management**



Change management is the process, tools and techniques to manage the people side of change to achieve the required business outcome. Change management incorporates the organizational tools that can be utilized to help individuals make successful personal transitions resulting in the adoption and realization of change.

The project manager submits change requests in accordance with the procedure for resolving issues. A branch is made in the SCM when a modification request is approved by the project manager.

### **Conclusion:**

For developing software systems, configuration management is essential. It is inevitable that many major software systems will change. Careful planning is necessary, considering things like the parts that will probably change, any other subsystems that might need to change for the alteration of the current system, cost analysis, etc. CM ensures that these modifications have no unintended effects on the other subsystems.



## RC5: Software Requirements Document



## **1.0 Introduction**

### **1.1 Purpose & Scope**

The purpose of this project is to develop a mobile application that calculates the user's Body Mass Index (BMI), body fat, ideal weight, and calories intake. This mobile application supports multiple users. It also has an administrator account with access to the database and the ability to manage users accounts.

The application is connected to a Firebase non-tabular database that stores the customers and admins information (username, email, date of birth, password, and gender). The backend server has a database of customers personal information. This application will run on mobile phones, and tablets.

### **Abbreviations**

Abbreviations	Definition
SRS	Software Requirement Specifications
BMI	Body Mass Index

### **1.2 Overview**

Software Requirements Specification (SRS) is document that outlines the characteristics of a project, piece of software, or application. SRS document is a project documentation that should be created before a project or application is launched. This document is also referred to as a software document and SRS report. A software document is often created for a project, piece of software, or application of any type.

The software requirement specification document must be prepared according to a set of rules. This covers the project's goal, scope, functional, nonfunctional requirements, software, and hardware specifications. Additionally, it includes details on the necessary environmental parameters, security and safety standards, project-specific software quality criteria.

### **1.3 General Description**

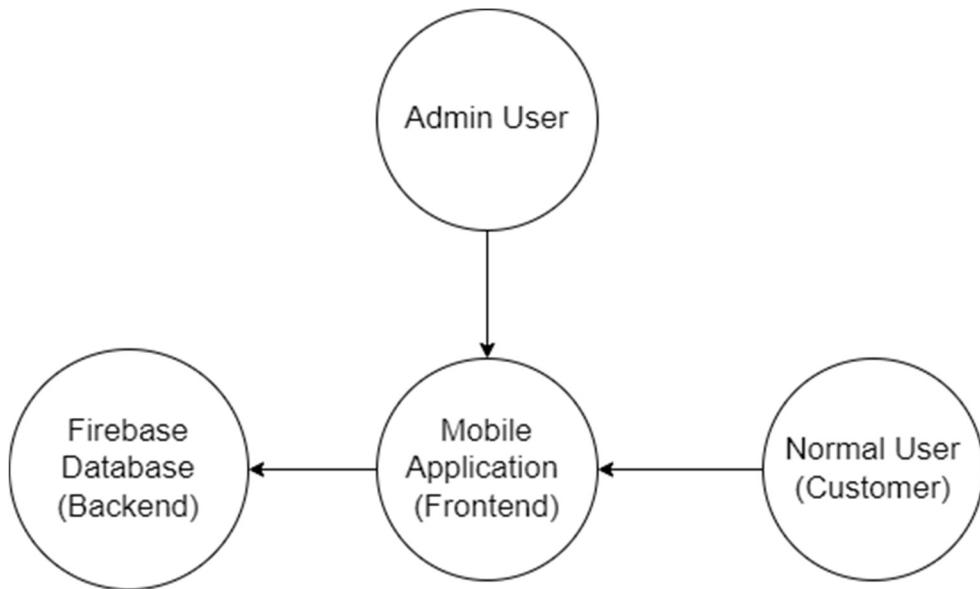
The concepts and features that the clients requested are displayed below together with the solutions that our team has suggested. It also describes the problems and difficulties our team encountered while

---

developing the program. This section also lists the system requirements needed for the application to function properly.

## 2.0 Product Perspective

This mobile application will be specifically designed for our client Fitness Guys. The application calculates BMI, body fat, ideal weight, and calories intake. The users of this application can be either a customer or an admin. The application is connected to Firebase database, which saves the users information.



## 3.0 Product Functions

The application is like other health and fitness applications that can be found on the Google Play Store. In addition, the users need to make an account to use the application. Having an account makes them able to login to the application from any device that can run the application.

The administrator (admin) has the authority to add and delete consumer accounts.

Customers can create an account and their account details are stored in the Firebase database. They can calculate their Body Mass Index (BMI), body fat, ideal weight, and calories intake. In addition, they will get an advice depending on the calculated values.

## 4.0 User Characteristics

---

The users of the application are classified in two groups

**Admin:**

- the person in charge of administering the application.
- Adding, and deleting users' accounts.

**User:**

- Uses the application to calculate their BMI, body fat, ideal weight, and calories intake.

## **5.0 General Constraints**

There are various restrictions when creating a mobile application that uses a backend server to hold user data. The mobile application should utilise as little power as possible so that users' mobile phones' batteries aren't drained. The mobile application must also adjust for various screen sizes and orientations. In order for consumers to access their data at any time, the server backend also has to be available 24 hours a day.

## **6.0 Assumptions and Dependencies**

The application is being developed using flutter framework, which can run on android and iOS devices. We therefore assumed that the majority of consumers would not have any issues using our application.

## **7.0 Functionality and Features**

### **7.1 Requirements**

#### **-Login**

By entering their personal details, the customer can login to the application.

#### **-BMI Calculator**

Customers can calculate their BMI value by providing their height (in cm), and their weight (in kg). Then the application calculates their BMI.

#### **-Ideal Weight Calculator**

Customers can calculate their ideal weight value by providing their height (in cm), and weight (kg).

---

Then the application calculates their ideal weight.

#### **-Calories Intake Calculator**

Customers can calculate their calories intake value by providing their height (in cm), weight (in kg).

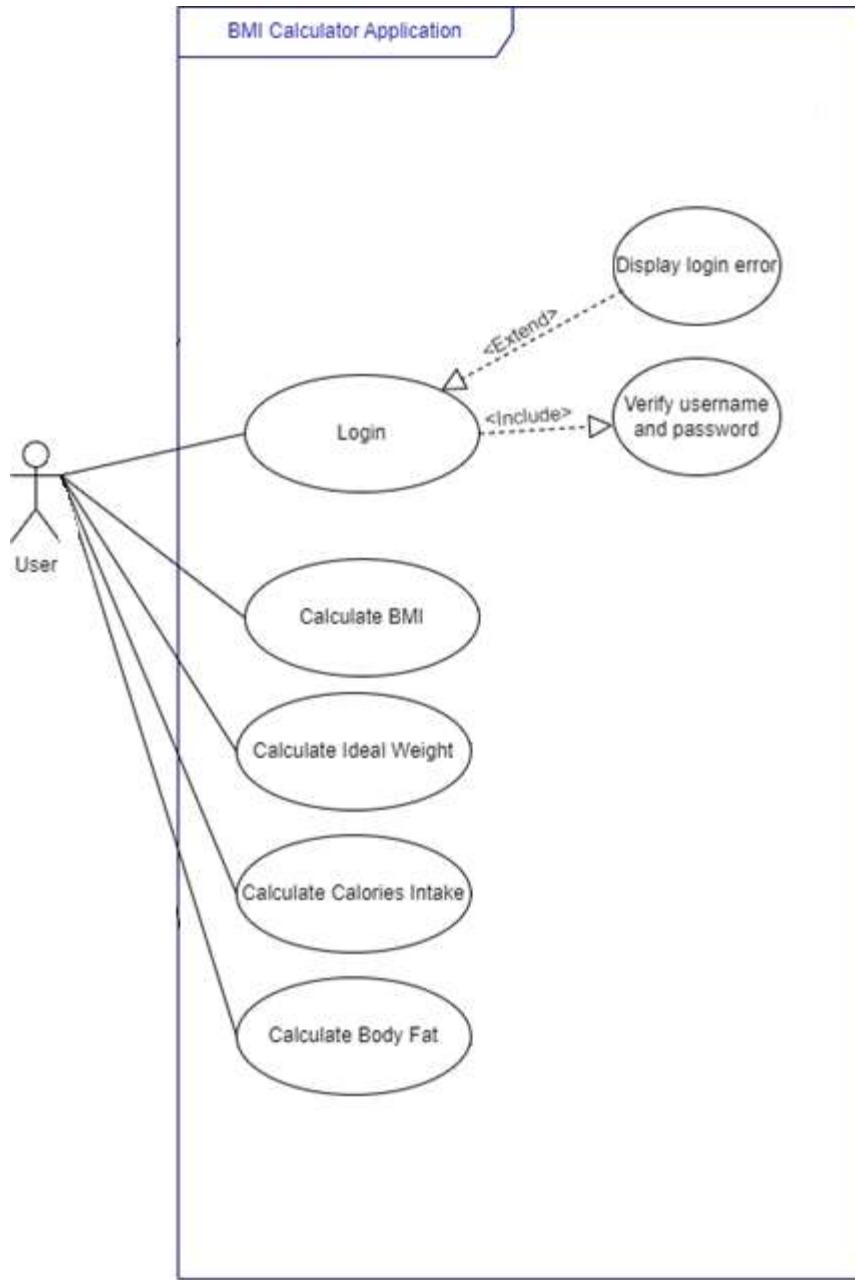
Then the application calculates their calories intake.

#### **-Body Fat Calculator**

Customers can calculate their body fat value by providing their height, hip, neck, waist measurements (in cm), and weight (in kg). Then the application calculates their calories intake.

### **7.2 Use Case**

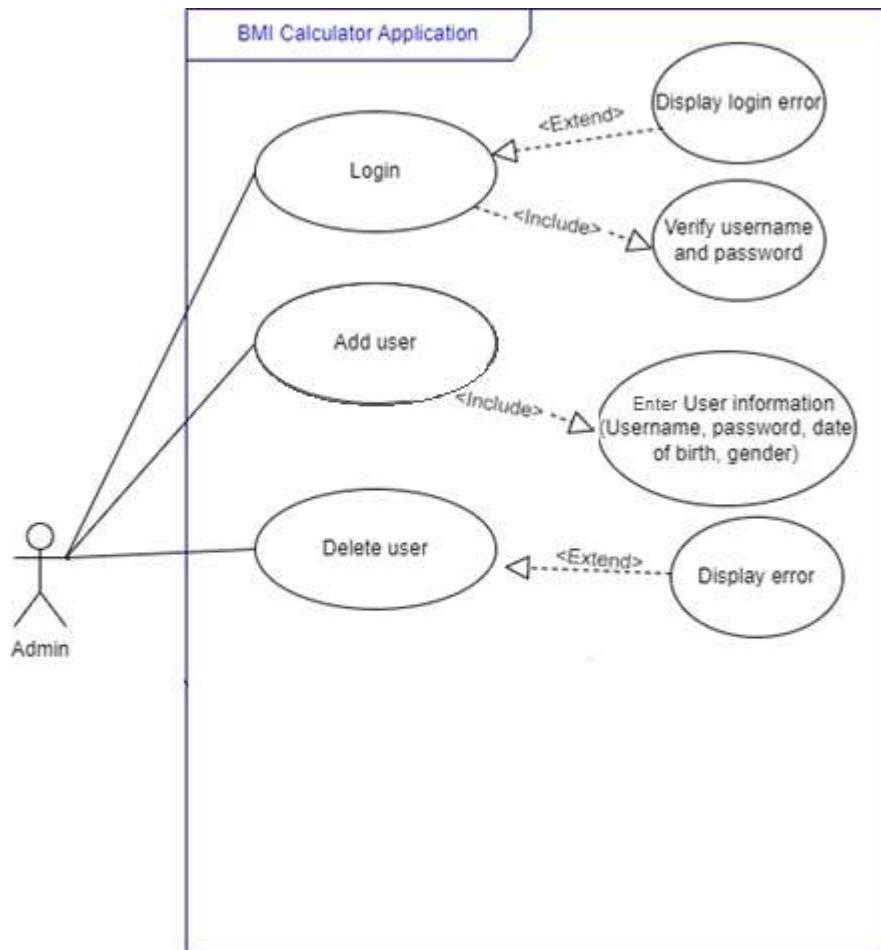




Use case diagram for user's point of view

Name Description	User use cases
<b>Actors</b>	<ul style="list-style-type: none"> <li>User</li> </ul>
<b>Main Success Scenario</b>	<ul style="list-style-type: none"> <li>An existing user can login using their username and password.</li> <li>User can calculate their BMI.</li> <li>User can calculate their Ideal Weight.</li> <li>User can calculate their Calories Intake.</li> <li>User can calculate their Bod Fat.</li> </ul>

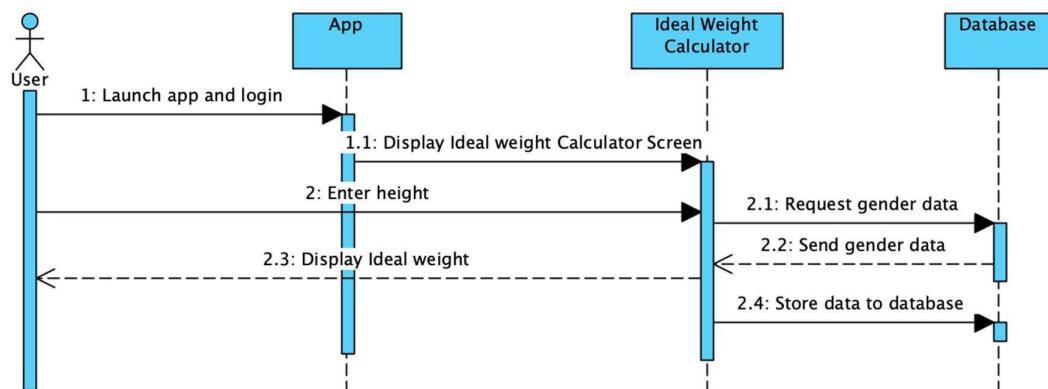
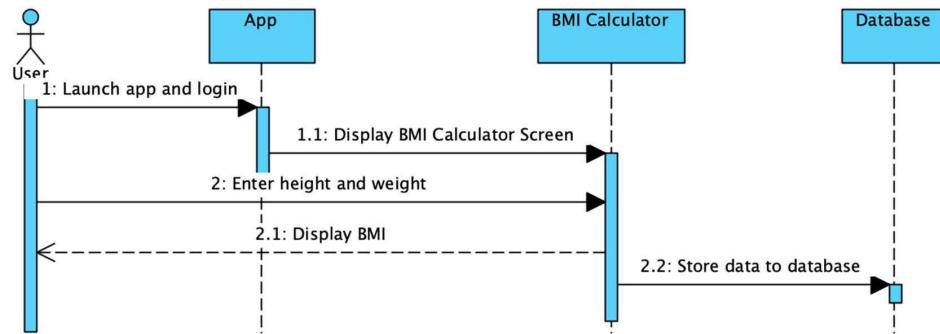
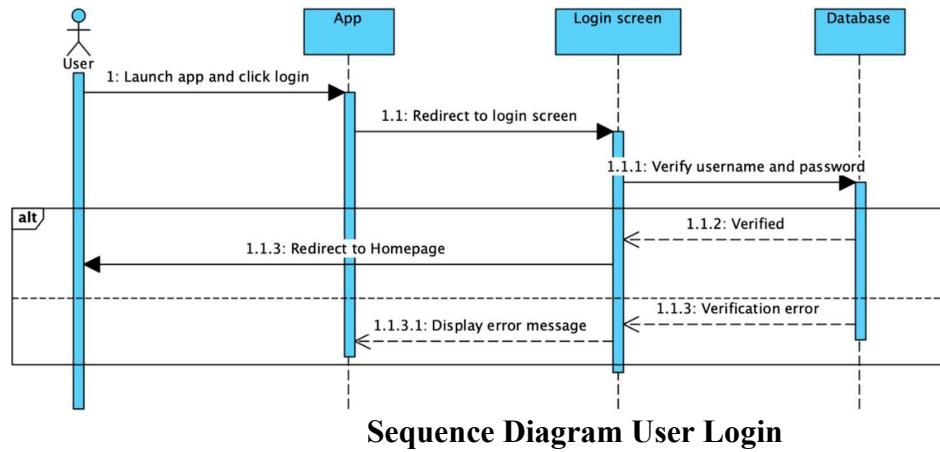
<b>Alternative Flows</b>	<ul style="list-style-type: none"> <li>If the user enters incorrect username or password, the application will display an error message.</li> </ul>
--------------------------	---

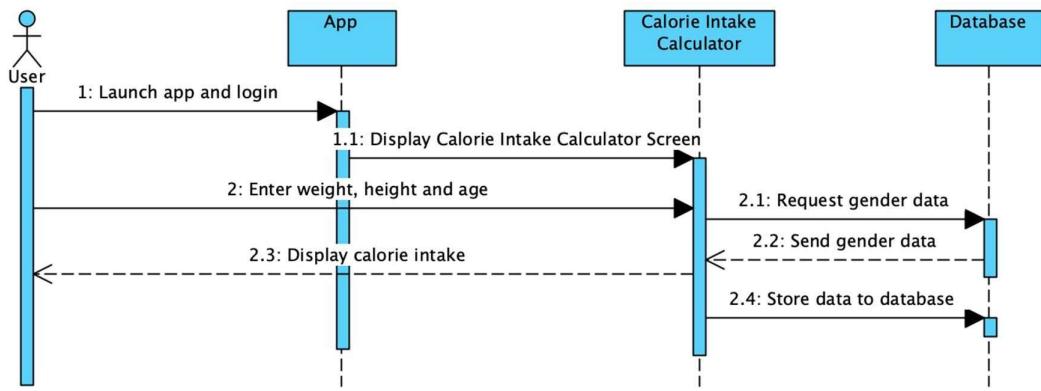


Use case diagram for admin's point of view

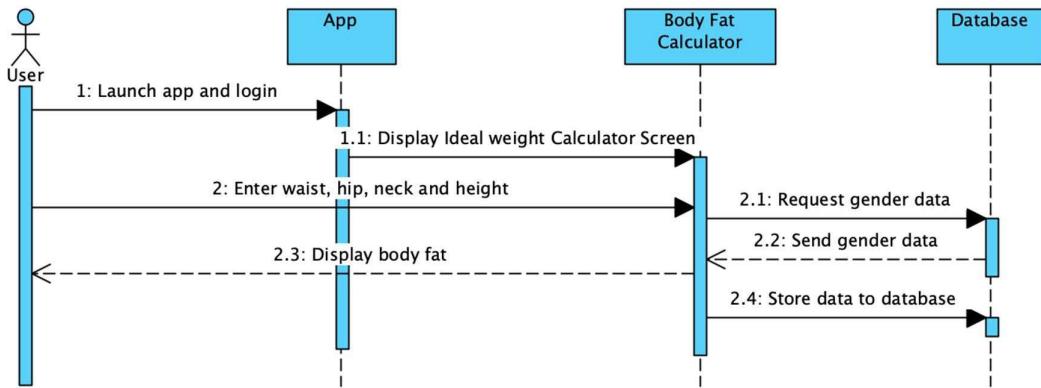
Name Description	User use cases
<b>Actors</b>	<ul style="list-style-type: none"> <li>Admin</li> </ul>
<b>Main Success Scenario</b>	<ul style="list-style-type: none"> <li>Admin can login using their username and password.</li> <li>Adding new user to the application.</li> <li>Deleting an existing user.</li> </ul>
<b>Alternative Flows</b>	<ul style="list-style-type: none"> <li>If the admin enters incorrect username or password, the application will display an error message.</li> <li>If admin deletes non-existing user, an error message will pop up.</li> </ul>

### 7.3 Sequence Diagrams

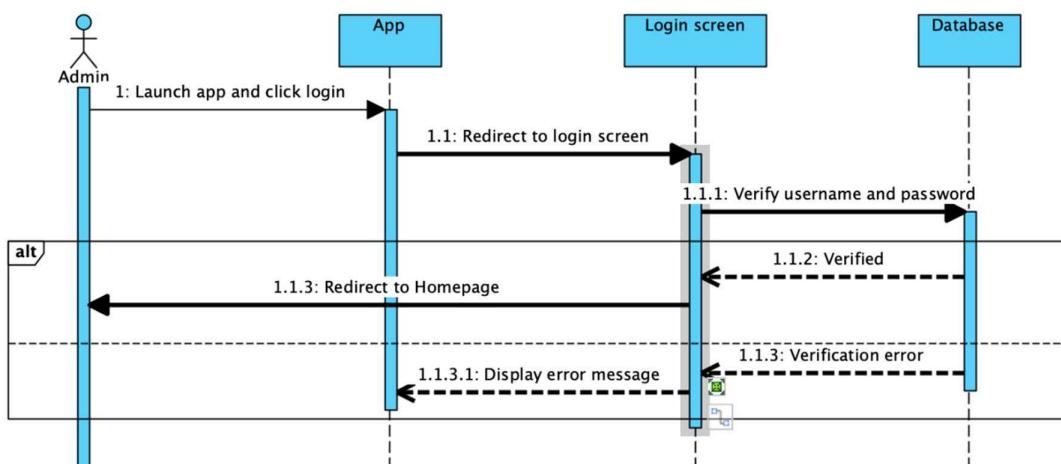




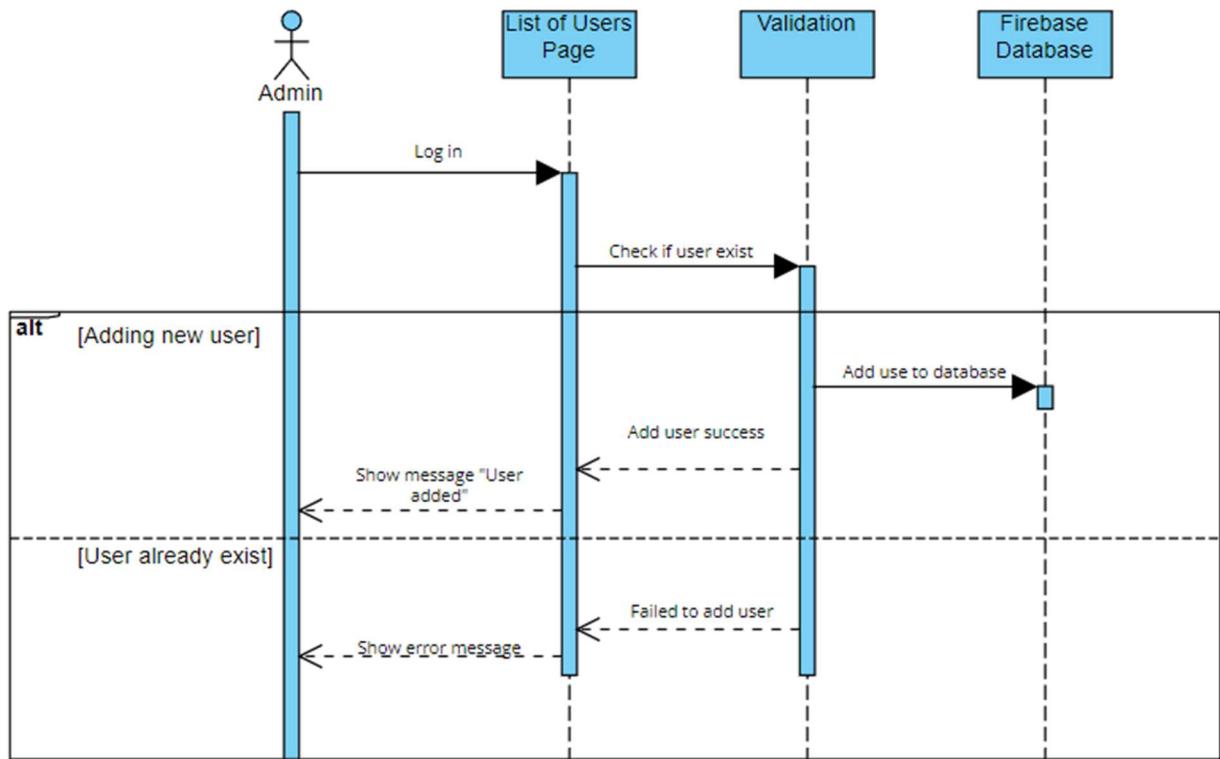
### Sequence Diagram Calories Intake Calculator



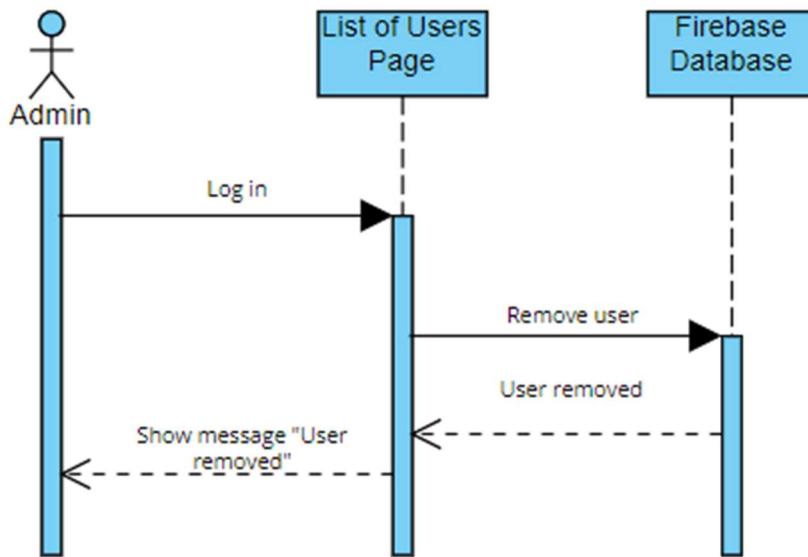
### Sequence Diagram Body Fat Calculator



### Sequence Diagram Admin Login



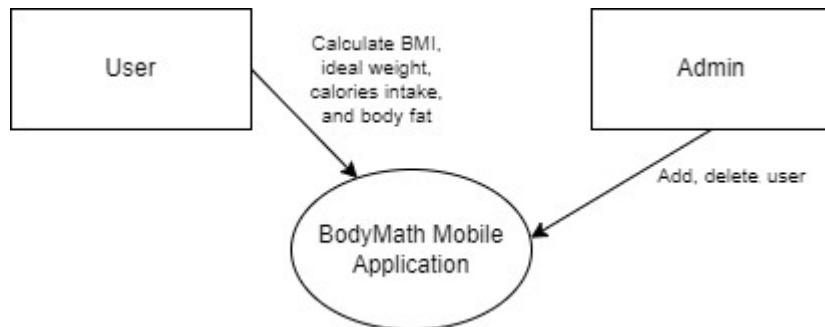
Sequence Diagram Adding User



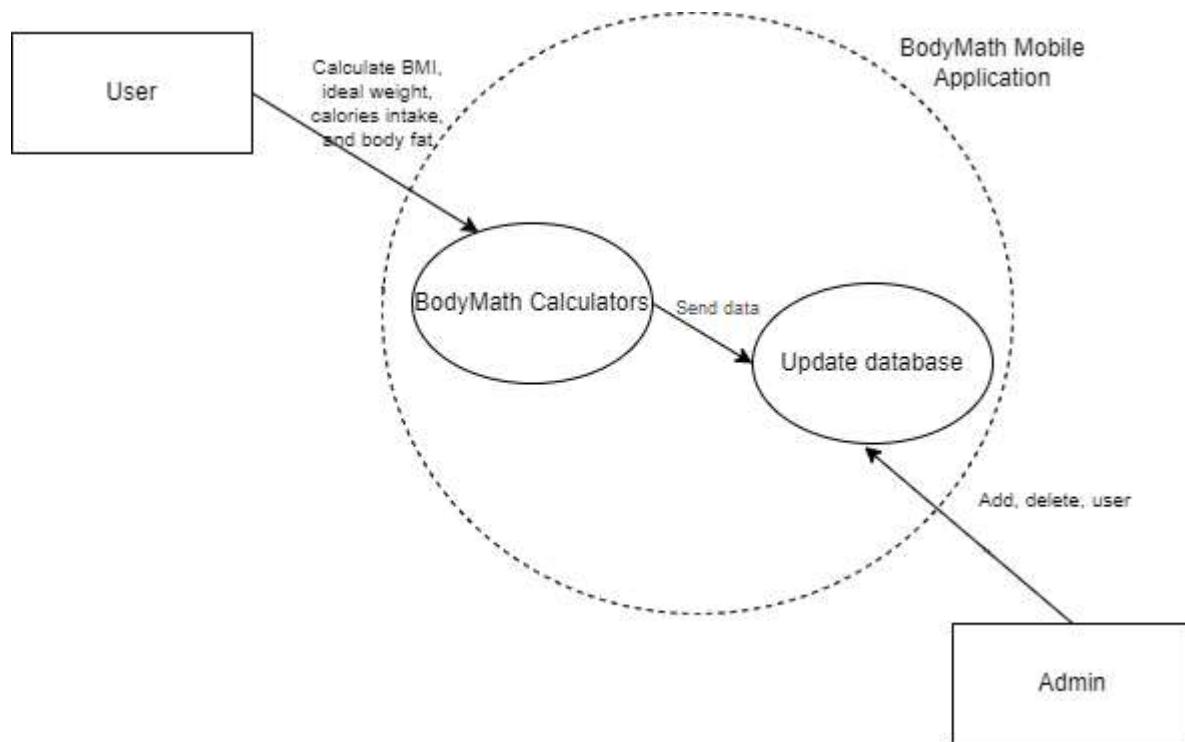
Sequence Diagram Removing User

## 7.4 Analysis Models

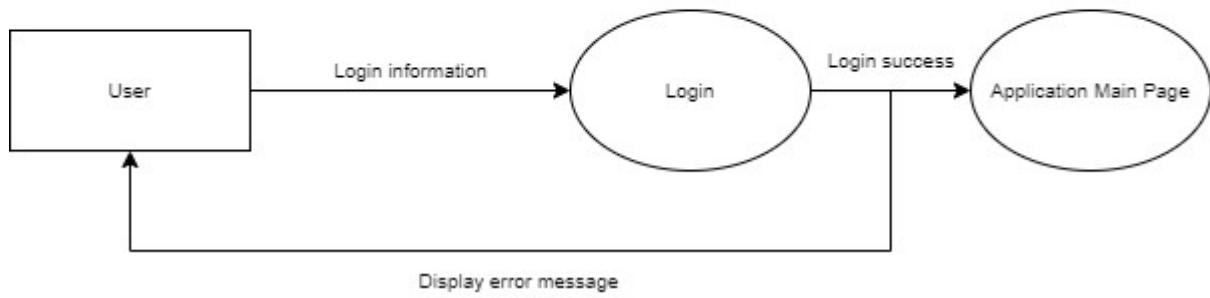
### Data Flow Diagrams



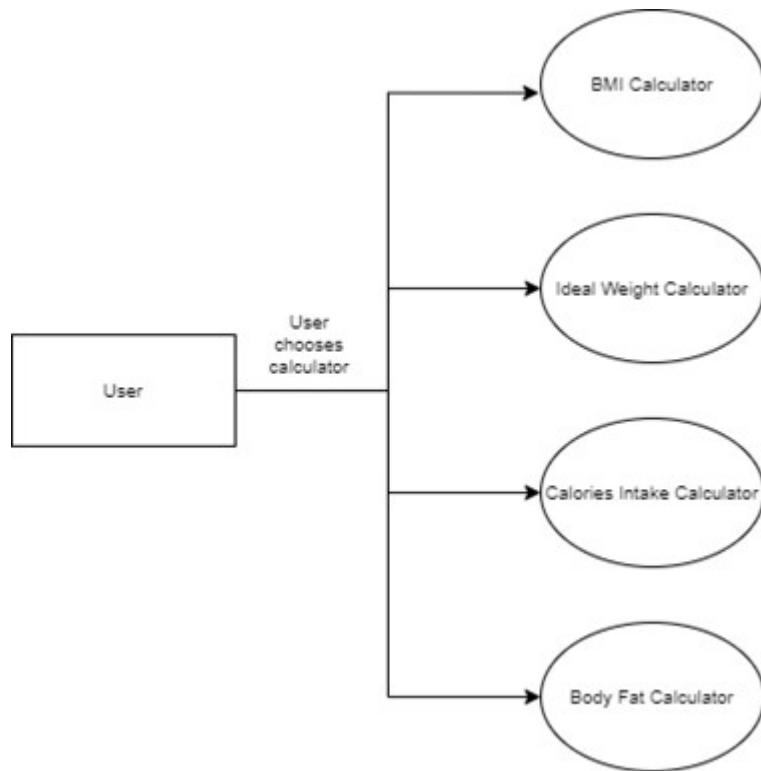
**Level 0 DFD**



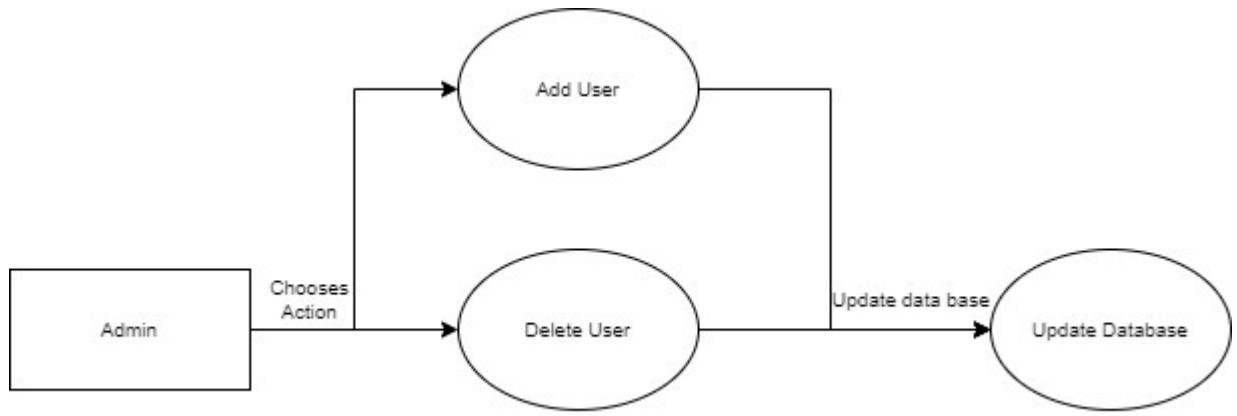
**Level 1 DFD**



**Login Level 1 DFD**

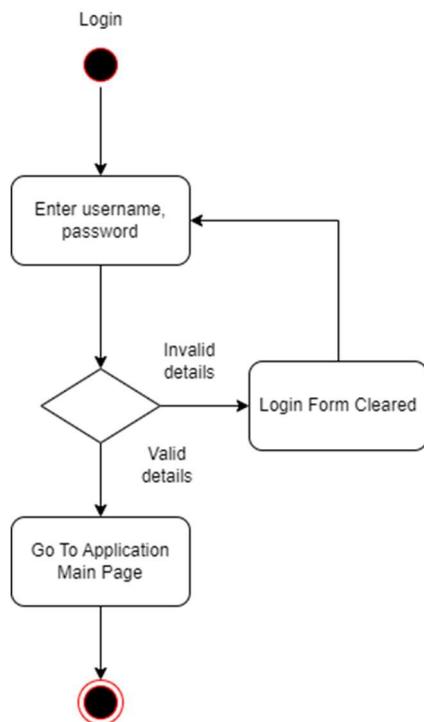


**BodyMath Calculators Level 1 DFD**

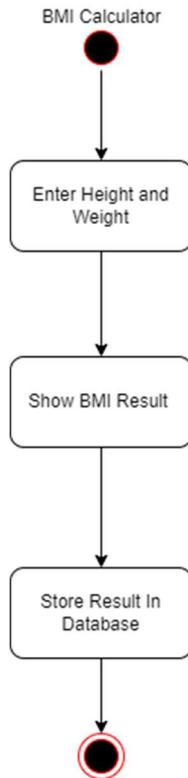


**Add, Remove User Level 1 DFD**

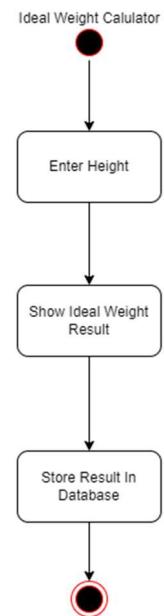
## 7.5 State-Transition Diagrams (STD)



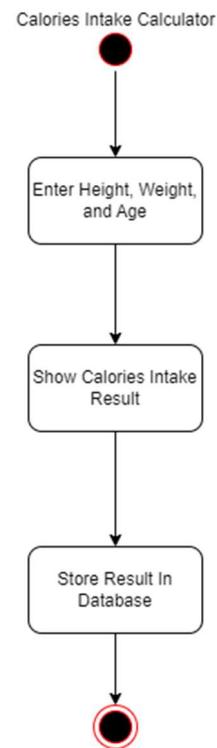
**UML Diagram for Login**



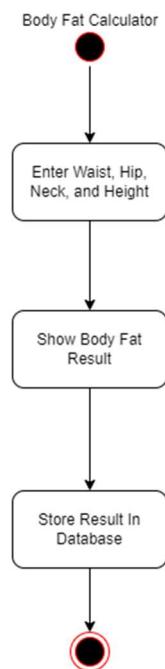
**UML Diagram for BMI Calculator**



**UML Diagram for Ideal Weight Calculator**

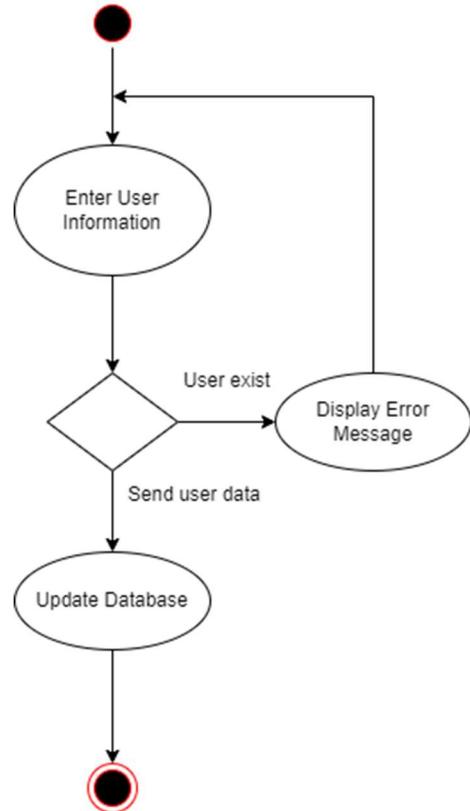


**UML Diagram for Calories Intake Calculator**



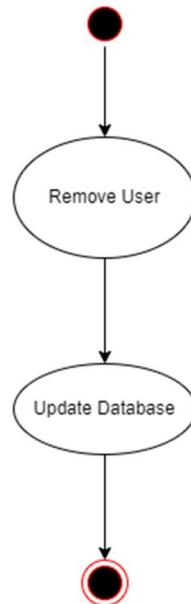
**UML Diagram for Body Fat Calculator**

Admin Add User



UML Diagram for Admin Add User

Admin Remove User



UML Diagram for Admin Remove User

## 8.0 Specific Requirements

The purpose of the mobile application is to provide an effective tool for determining BMI, ideal

weight, calories intake, and body fat values. The system should have a straightforward user interface to prevent confusion. Additionally, the system enables the administrator to manage the application.

## **8.1 External Interface Requirements**

### **8.1.1 User Interfaces**

- This mobile application's user interface is uncomplicated, convenient, and easy to use.
- Each operation should require the fewest possible clicks on the user interface.

### **8.1.2 Hardware Interfaces**

- Application should work on most modern mobile devices.
- Application should adapt to all screen resolutions and sizes.
- Application should be power efficient, so it does not drain the user's mobile phone battery.

### **8.1.3 Software Interfaces**

- The administrators only can modify the database.
- Mobile application communicates with database in order to save user account information.

### **8.1.4 Communications Interfaces**

- Users need internet connection to use the application.
- The server needs 24-hours uptime, so the users can access their data at any time.

## **9.0 Non-Functional Requirements**

### **9.1 Performance**

- The application should use as low hardware resources as possible, so the application can run any android device.

### **9.2 Reliability**

- The calculation must be accurate, so the users can rely on its data.
- The application should be bug free.

### **9.3 Availability**

- The application should be available in most modern mobile devices with internet connection.
- The application should be available all the time.

### **9.4 Security**

- The entities of the database cannot be changed by anyone other than administrators.
- All communications between the server and the mobile application must be encrypted.

## 9.5 Maintainability

- The Application will get consistent updates to fix bugs and add features to the application base on user's feedback.

## 9.6 Portability

- The application should run on all android devices.

## 10. Meeting Minutes with Fitness Guys

One meeting were held to discuss and gather the requirements of the application. In the meeting, the client explains what features their application need to have.

First Meeting		
11 <sup>th</sup> of November 2022	09:00 AM – 10:00 AM	Google Meet
Note taker	Abdullah Mohammed (Fitness Guys) Bashir Abugharsa (SOG)	
Attendees	Fitness Guys -Khalid Ibrahim (CEO) -Hakeem Mahmood (Operation Manager) -Abdullah Mohammed (Main Representative) SOG -Bashir Abugharsa (Software Configuration Manager) -Marawan Eldeib (Software Quality Assurance Manager) -Yousef Yasser (Documentation Manager) -Ali Kharimeh (Project Manager)	
Mobile Application Specifications by Fitness Guys		
30 minutes	Khalid Ibrahim	
Discussion	1. The mobile application should calculate BMI, ideal weight, calories intake, and body fat of the user. 2. The mobile application should have an administrator account, that manages the application.	
SOG Review & Recommendations		
30 minutes	Marawn ELdeib	
Discussion	1. The mobile application should be developed to work	

	<p>on IOS and android mobile application.</p> <ol style="list-style-type: none"> <li>2. The users should be able to create accounts, so they can access their data from any android device.</li> <li>3. The user information and data should be stored on a backend server database.</li> </ol>	
Meeting Conclusion	<ol style="list-style-type: none"> <li>1. Both parties agreed to work together to develop the application.</li> <li>2. Both parties agreed to schedule another meeting on 30 of October to discuss the proposal of the mobile application development.</li> </ol>	
Action Item	Person Responsible	Deadline
Meeting notice to be send to all via E-mail.	Ali Kharimeh	1 <sup>st</sup> of January 2022

Second Meeting		
20 <sup>th</sup> of November 2022	09:00 AM – 10:00AM	Google Meet
Note taker	Abdullah Mohammed (Fitness Guys) Bashir Abugharsa (SOG)	
Attendees	Fitness Guys -Khalid Ibrahim (CEO) -Hakeem Mahmood (Operation Manager) -Abdullah Mohammed (Main Representative) SOG -Bashir Abugharsa (Software Configuration Manager) -Marawan Eldeib (Software Quality Assurance Manager) -Yousef Yasser (Documentation Manager) -Ali Kharimeh (Project Manager)	
Mobile Application Project Proposal Review		
30 Minutes	Ali Kharimeh	
Discussion	<ol style="list-style-type: none"> <li>1. Presentation of the project proposal by SOG to Fitness Guys.</li> <li>2. System diagram presentation of the mobile application.</li> </ol>	

	<ul style="list-style-type: none"> <li>3. Presentation on frontend, and backend architecture of the system.</li> </ul>				
Conclusion	<ul style="list-style-type: none"> <li>1. SOG will develop a BodyMath calculator application for Fitness Guys.</li> <li>2. Fitness Guys were pleased with the proposal.</li> </ul>				
Fitness Guys Review					
30 Minutes	Khalid Ibrahim				
Discussion	<ul style="list-style-type: none"> <li>1. Presentation by Fitness Guys.</li> <li>2. They were pleased by the work done by SOG.</li> <li>3. They liked the architecture that SOG used to develop the application.</li> </ul>				
Meeting Conclusion	<ul style="list-style-type: none"> <li>1. Fitness Guys approved SOG approval.</li> <li>2. SOG will prepare Software Requirements Document according to Fitness Guys requirements.</li> </ul>				
Action Items	<table border="1"> <thead> <tr> <th>Person Responsible</th><th>Deadline</th></tr> </thead> <tbody> <tr> <td>SOG</td><td>1<sup>st</sup> of January 2022</td></tr> </tbody> </table>	Person Responsible	Deadline	SOG	1 <sup>st</sup> of January 2022
Person Responsible	Deadline				
SOG	1 <sup>st</sup> of January 2022				
Software Requirements Document					



## RC6: Software Design Document



---

## **Introduction**

SOG handed in their proposal to Fitness Guys HQ on the 11<sup>th</sup> of November 2022. The proposal has been accepted on the 20<sup>th</sup> of November 2022. A “Fitness Calculator” program will be developed by SOG and will in sync with Fitness Guys’ Databases.

## **Purpose**

The purpose of this document is to give the developer a comprehensive understanding of the system design in order to reduce any difficulties with the actual execution of the finished product. The end-user characteristics of the software, hardware, and related frameworks used in the creation are described in the software design document. This section of the document has several purposes, including:

- To aid in determining what should be built and how to do it
- To analyze and evaluate the completed job
- To assess customer compliance
- To provide developers with a functional point of view
- To assist in simulating the environment when executing test cases
- **Scope**
- The graphical/high abstraction diagram can be discussed in the document in a practical way.
- Details on the relevant hardware, software, and framework will be provided.
- 

## **Audience**

- The development team, the quality assurance team, design team, and testing team are to use the document.

## Overview

The simplified design structure of BodyMath is by providing users a user-friendly interface via an app to either connect to their database records or use the offline BMI calculator.

Administrators can sign in to modify the database and:

- Create member accounts
- Delete member accounts
- Edit member's personal information

Calculated data from online users will utilize the database to fetch data records such as:

- Gender
- Age
- Other fixed traits

And store records such as:

- Calculated BMI
- Body fat percentage
- Various other calculations

## Design Considerations: Assumptions

Android 5.0 (Lollipop) and up are presumptively used to execute the application. Beyond that, it is expected that the user is able to use the application on any Android device with or without internet connection. The main presumption is that the application will be expanded to enable a new functionality depending on our client's needs. The suggested application will undoubtedly get more features in the future.

## Design Considerations: Dependencies

Regarding dependencies, the two servers are necessary for the suggested application to be able to store the necessary data. One of the servers is utilized for data backup, and also contains a domain that the gym can use for its interface with potential clients. The interface must be granted permission to use the gym's capacity database at a specific time.

---

## Goals and Guidelines

The suggested application must be implemented with the following objectives in mind:

- The system must offer a solid user experience and be simple to use. To easily perform calculations, the user interface must be simple enough for the user. The number of different pages in the app needed to complete each transaction completed by the user should be kept to a minimum.
- The proposed system's features must be expandable, allowing users to enable additional functions and application changes as needed.
- The developers will conduct routine checks to ensure that the information provided by the system is accurate. Additionally, this will guarantee that the application runs smoothly and that users won't be inconvenienced by any inconsistencies.
- Users won't have to worry about data breaches, which could damage our product's reputation, as they can submit their information details safely. By limiting access to the information to just those who are permitted, such as administrators, we hope to protect the data of the users.
- The application must be accessible to users on any Android device including tablets and smartphones. The system should be created to have the same interface on any Android device.
- To guarantee user satisfaction and enhance the user experience, any future development, such as modifications and new features in response to consumer feedback, will be implemented into the system.
- Any future developments, such as modifications and new features based on customer requests or feedback, will be integrated into the system to ensure user satisfaction and improve the user experience.

## **Architectural Strategies**

The design tactics that have an impact on the appointment application system's higher-level structures and overall organization are discussed in this part.

The system can be expanded and improved further at the request of the customer to include additional features and functionalities. This is crucial to ensure that our application remains updated and that the introduction of new features enhances the client experience. However, it's crucial to try out the features first to make sure our target audience can utilize them smoothly and without any interruptions.

The user must have an Internet connection in order to sign in and use the exclusive application features from an interface perspective. In the meantime, the database must be continuously updated for each calculation made by users in order for the software interface to function, and for security reasons, only authorized users, such as administrators, are permitted access to the database via the application in order to perform modifications.

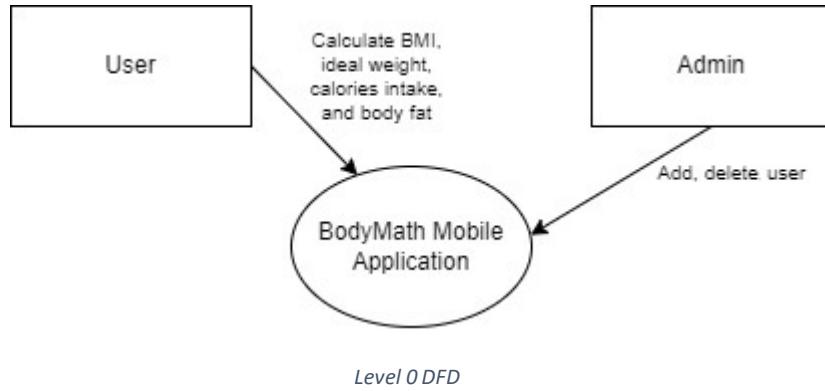
In the meantime, a test case design will be created and applied to the software for error detection and recovery. The SQA team will next evaluate, test, and audit the software application consistently using the test case.

## **System Architecture**

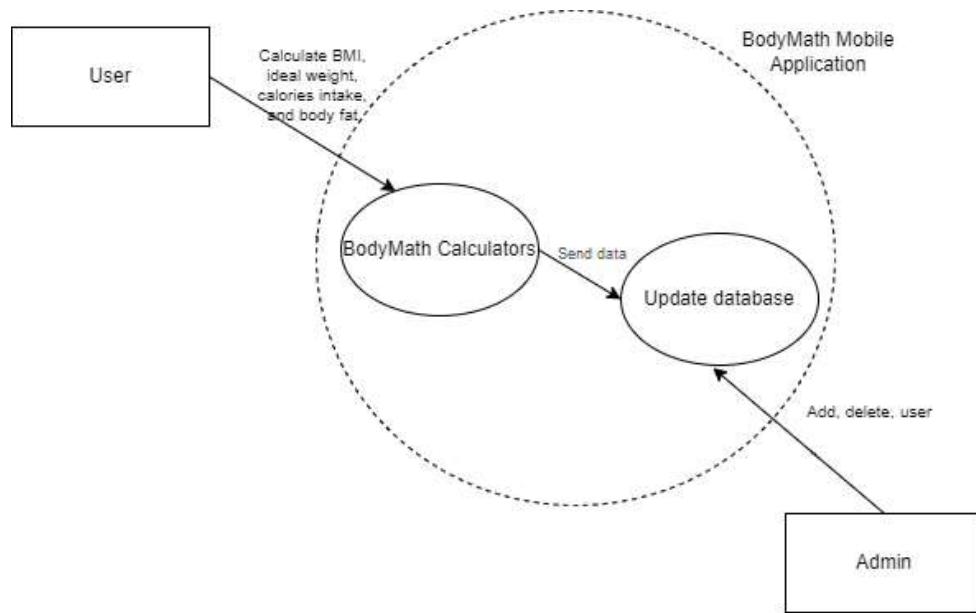
To outline the main duties of Fitness Guys, this section will give a high-level summary of the application being developed. The document then includes a thorough description of each component as a Data Flow Diagram. The main goal of this strategy is to give a general grasp of how app technology works, how and why the system is formed as it is, and how various components interact to form a comprehensive system of interdependence. The Software Requirement Specification (SRS) document is referred to in the DFD that is shown in the following subsections.

---

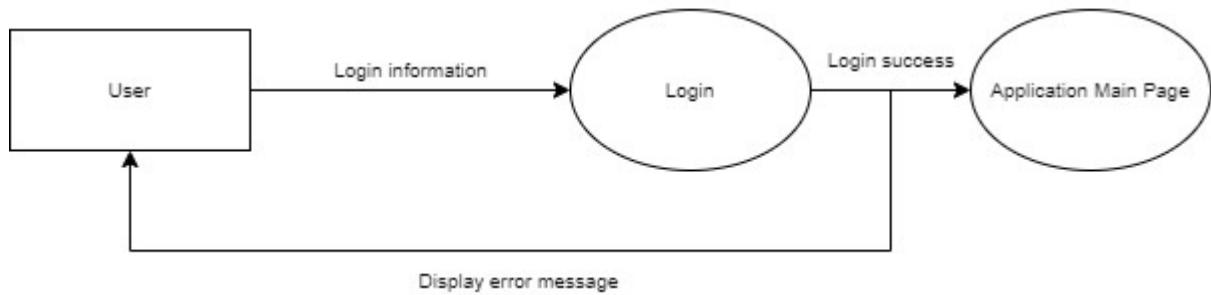
## Data Flow Diagrams



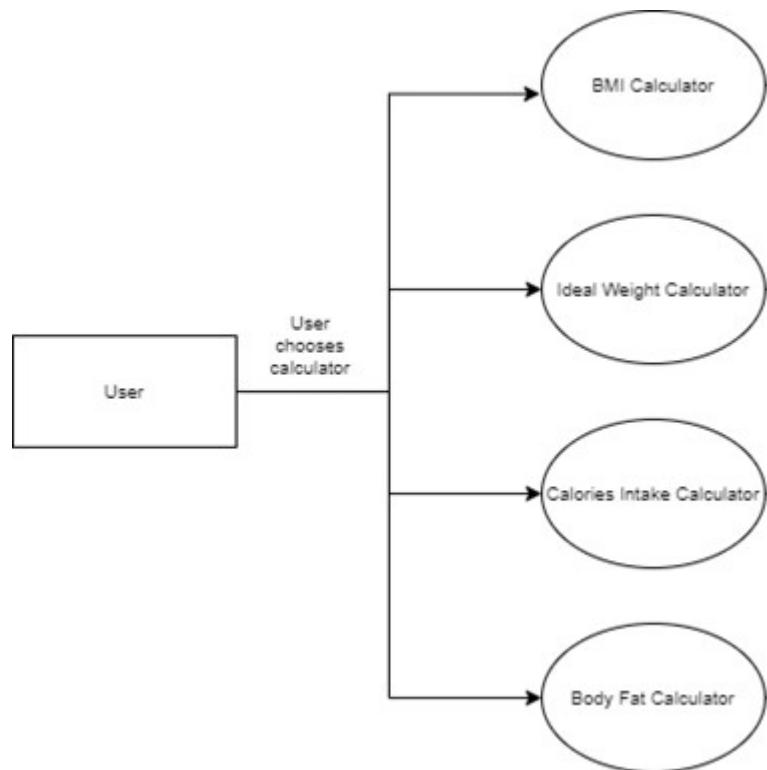
Level 0 DFD



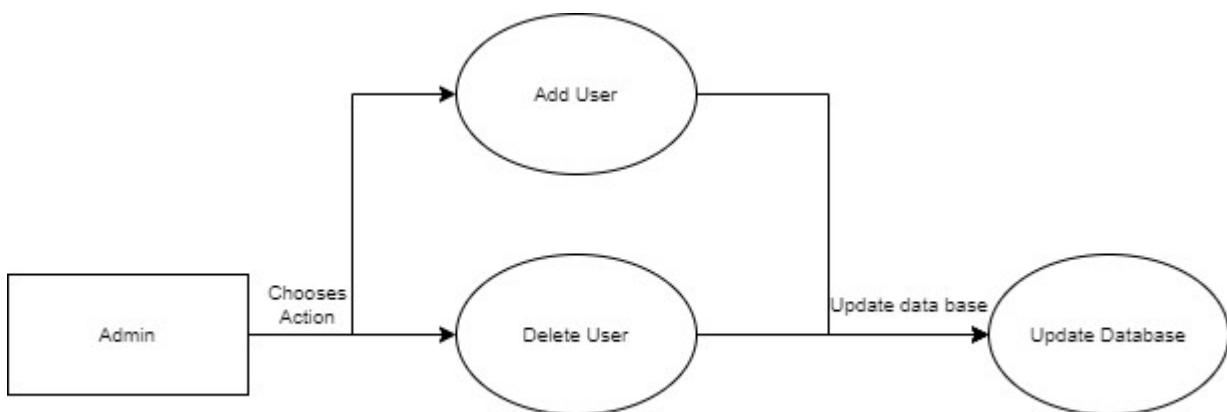
Level 1 DFD



*Level 1 DFD (Login)*



Level 1 DFD (Calculators)

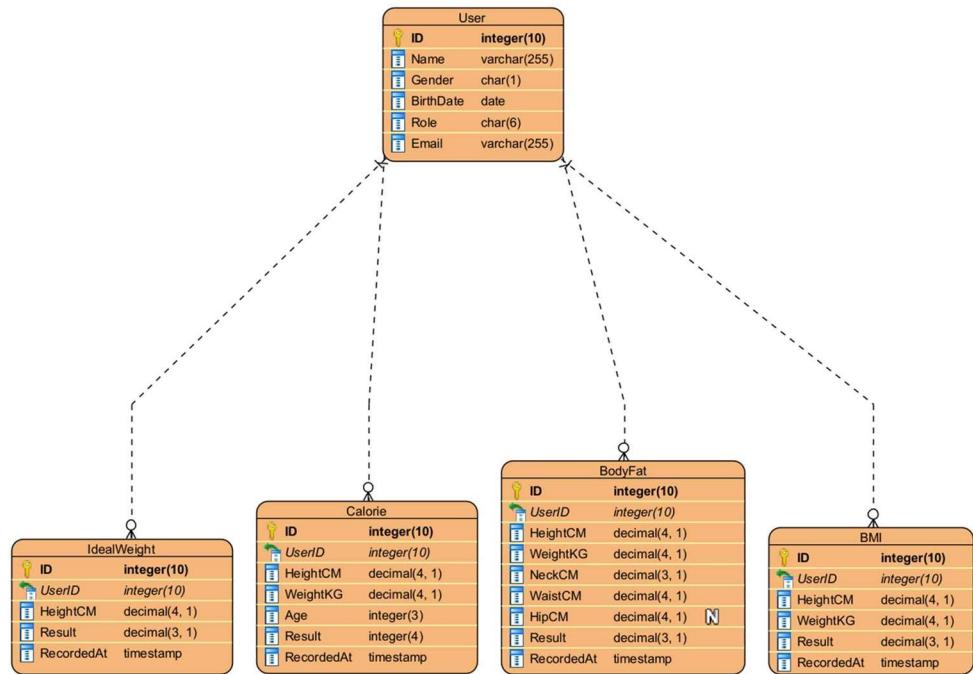


Level 1 DFD (Admin View)

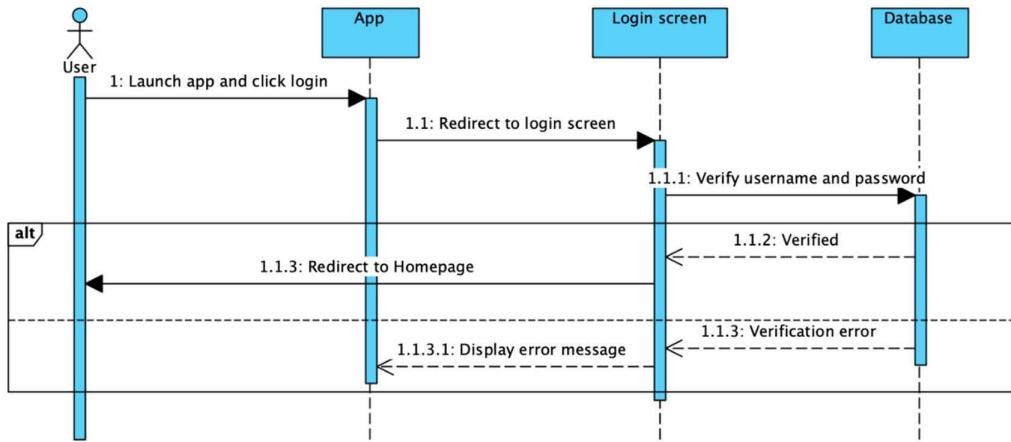
## Data Dictionary

Table	Name	Type	Nullable	Example
User	ID (PK)	Integer (10)	No	1234567890
	Name	Varchar (255)	No	John Doe
	Gender	Char (1)	No	M
	BirthDate	Date	No	2002-01-05
	Role	Char (6)	No	Member
	Email	Varchar (255)	No	johndoe@email.com
IdealWeight	ID (PK)	Integer (10)	No	1234567890
	UserID (FK)	Varchar (255)	No	1234567890
	HeightCM	Decimal (4, 1)	No	170.5
	Result	Decimal (3, 1)	No	70.2
Calorie	ID (PK)	Integer (10)	No	1234567890
	UserID (FK)	Integer (10)	No	1234567890
	HeightCM	Decimal (4, 1)	No	170.5
	WeightKG	Decimal (4, 1)	No	111.2
	Age	Integer (3)	No	26
	Result	Integer (4)	No	2102
	RecordedAt	Timestamp	No	2022-11-09 19:05:22
BodyFat	ID (PK)	Integer (10)	No	1234567890
	UserID (FK)	Integer (10)	No	1234567890
	HeightCM	Decimal (4, 1)	No	170.5
	WeightKG	Decimal (4, 1)	No	111.2
	NeckCM	Decimal (3, 1)	No	30.3
	WaistCM	Decimal (4, 1)	No	40.2
	HipCM	Decimal (4, 1)	Yes	52.5
	Result	Decimal (3, 1)	No	15.2
	RecordedAt	Timestamp	No	2022-11-09 19:05:22
BMI	ID (PK)	Integer (10)	No	1234567890
	UserID (FK)	Integer (10)	No	1234567890
	HeightCM	Decimal (4, 1)	No	170.5
	WeightKG	Decimal (4, 1)	No	111.2
	Result	Decimal (3, 1)	No	25.2
	RecordedAt	Timestamp	No	2022-11-09 19:05:22

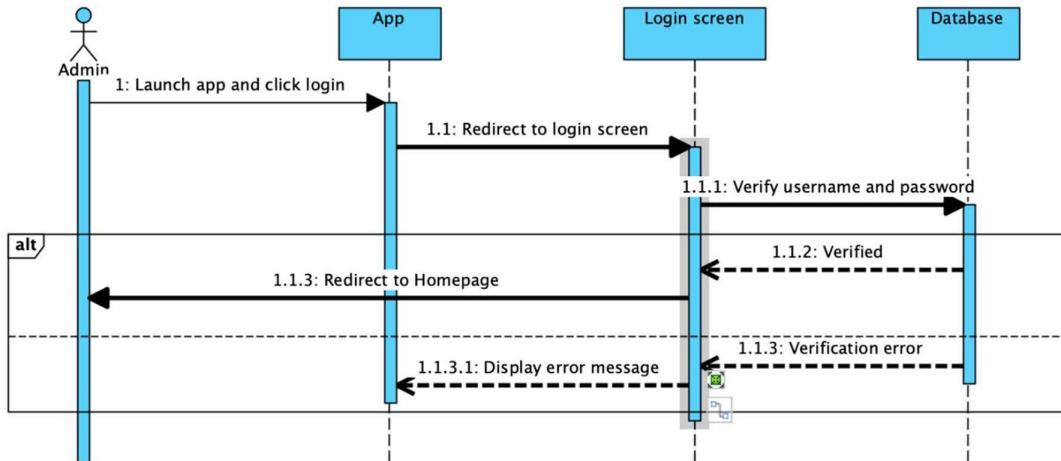
# Entity Relationship Diagram



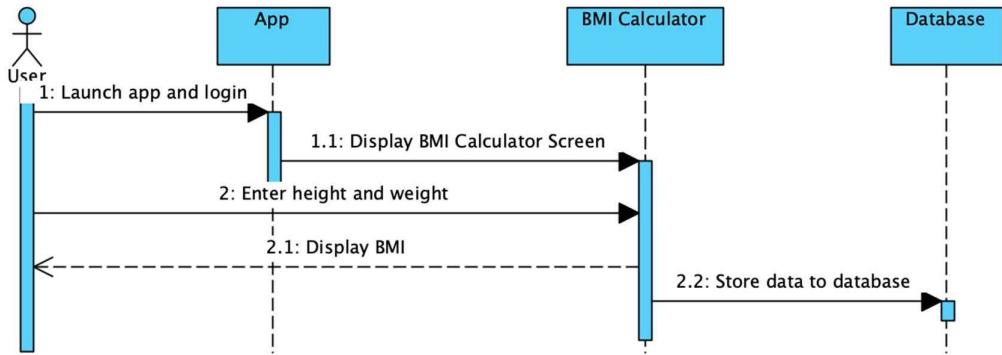
# Sequence Diagrams



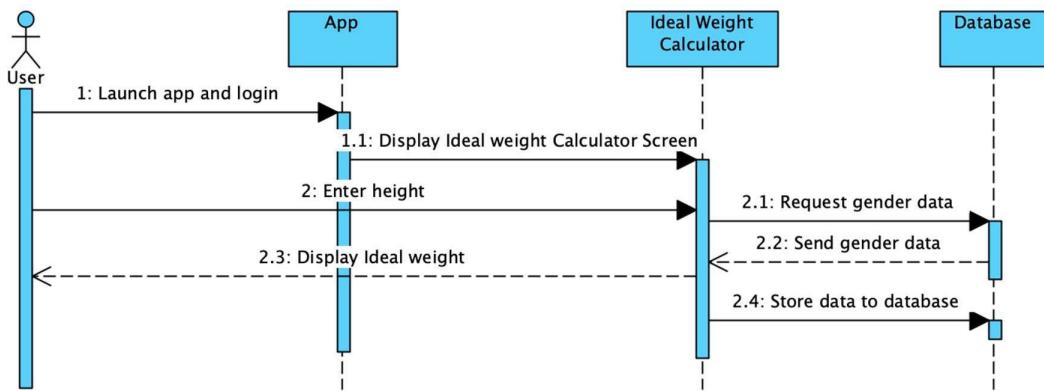
User Login Sequence



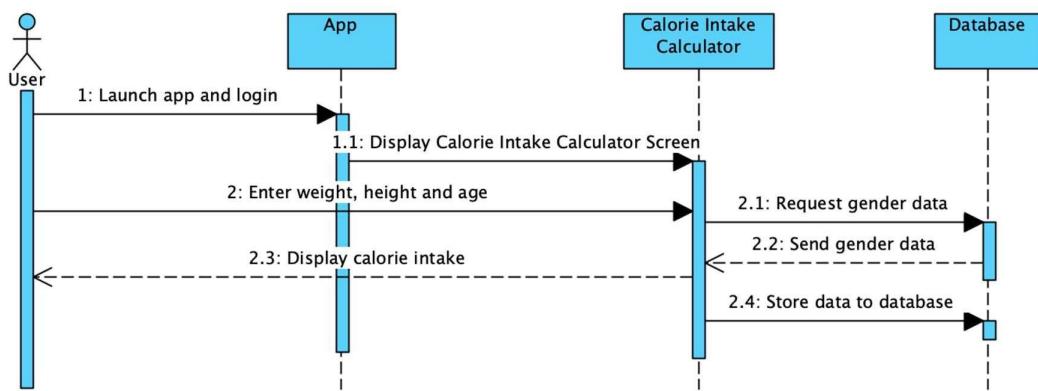
Admin Login Sequence



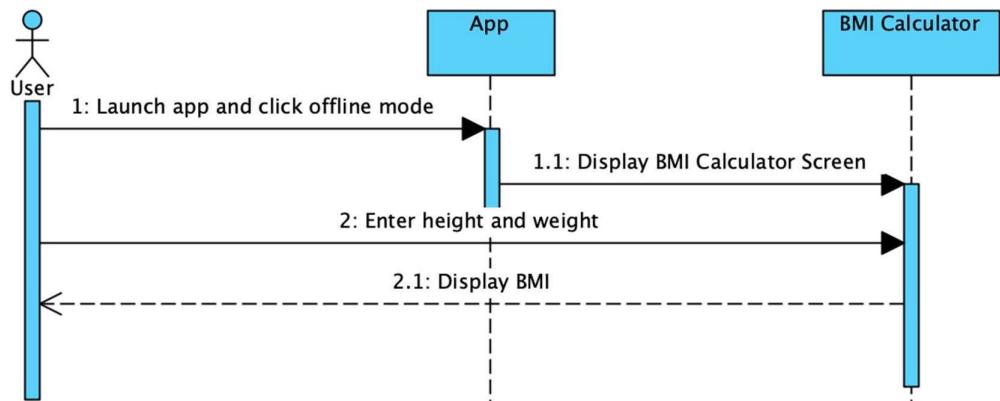
BMI Calculator Sequence



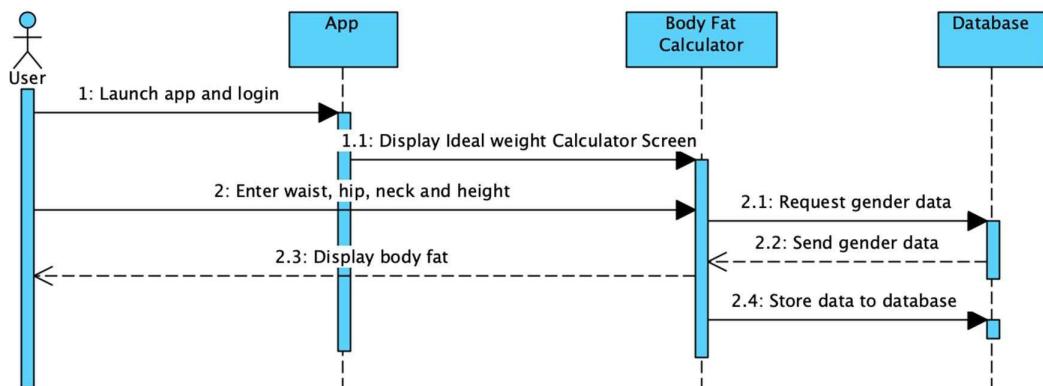
Ideal Weight Sequence



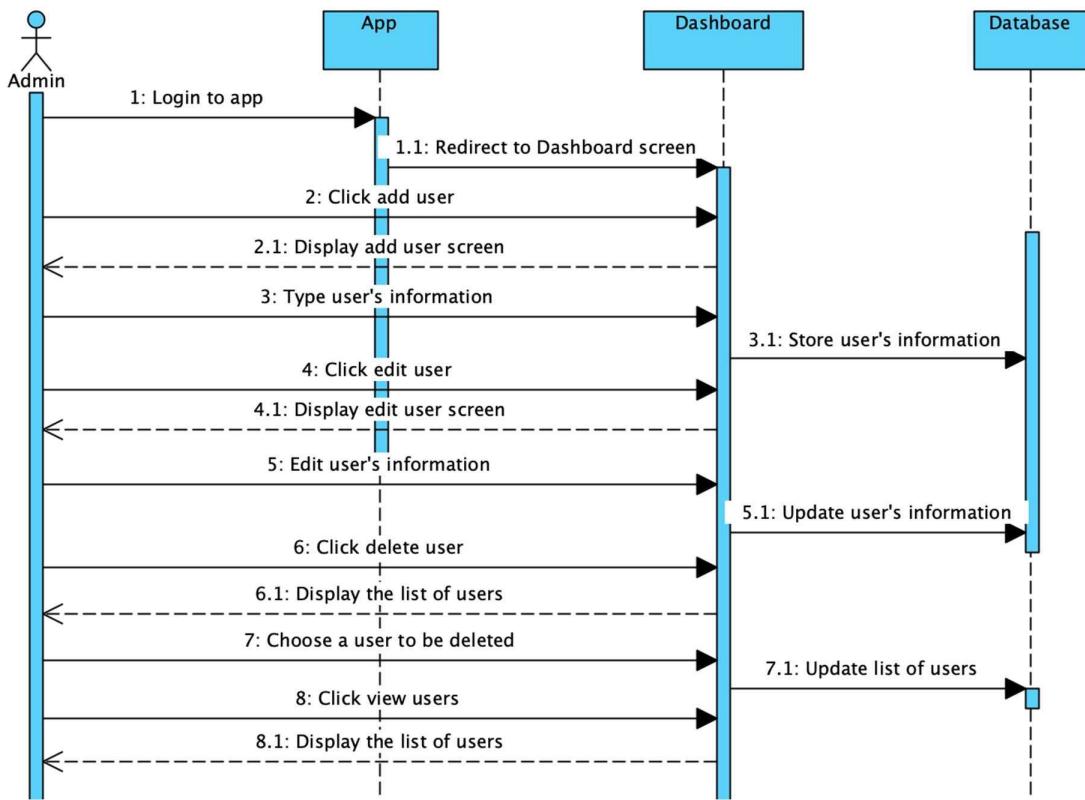
Calorie Intake Sequence



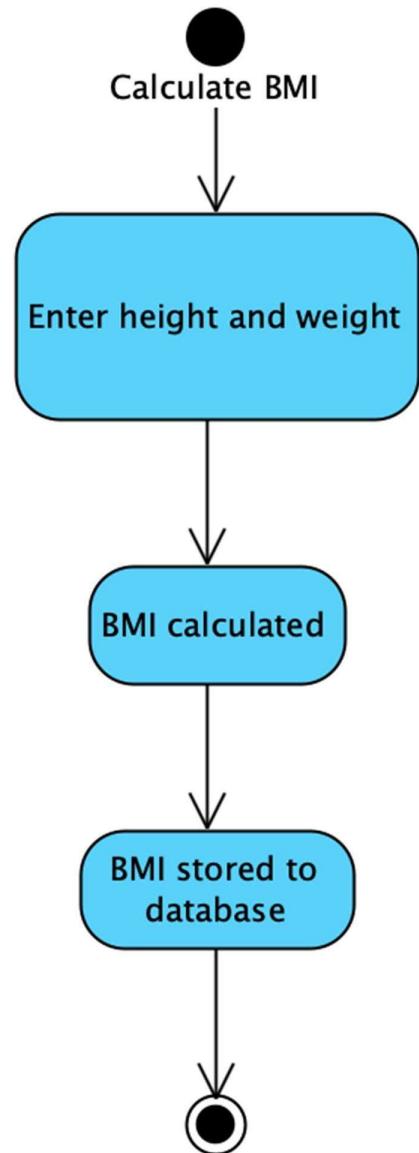
*Offline Mode Sequence*



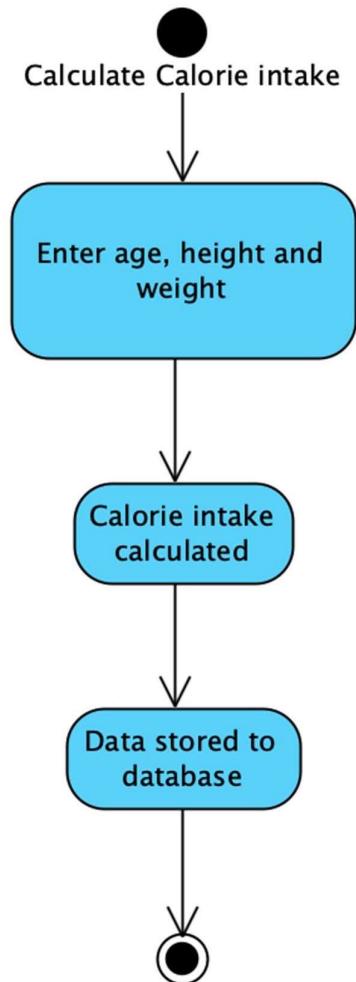
*Body Fat Calculator Sequence*



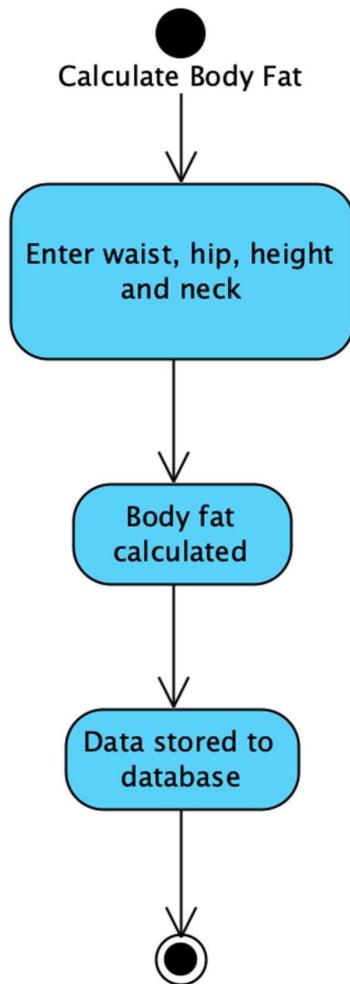
## Activity Diagrams



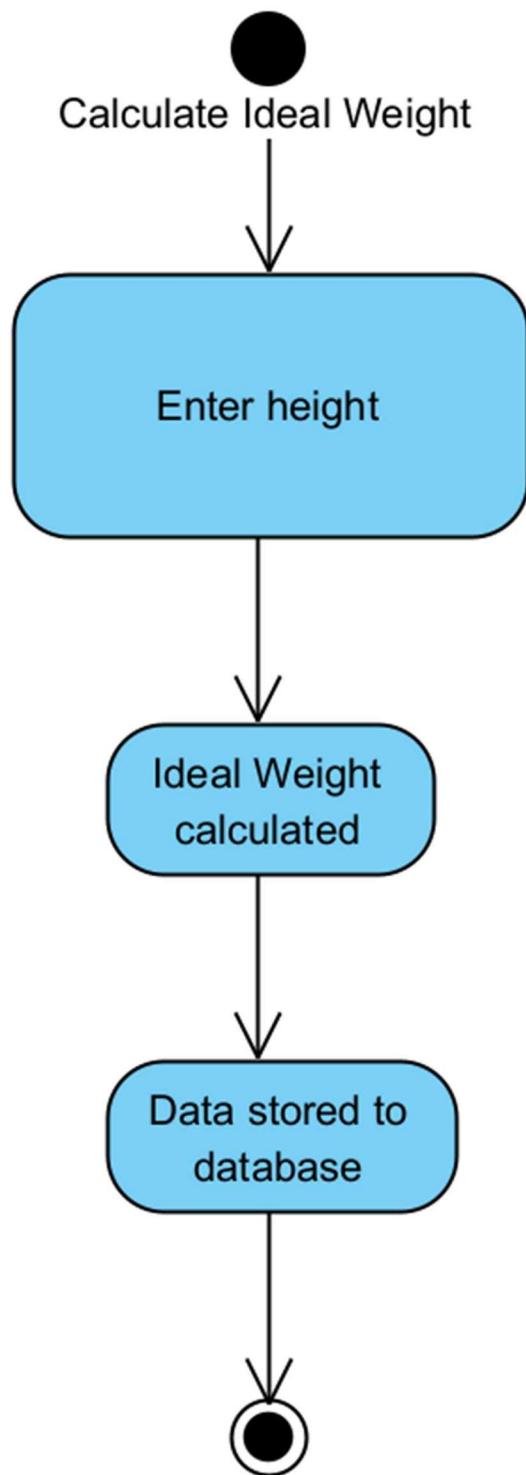
*BMI Calculator Activity*



*Calorie Intake Activity*



*Body Fat Activity*



*Ideal Weight Activity*

Add user

Type user's information

User added to database

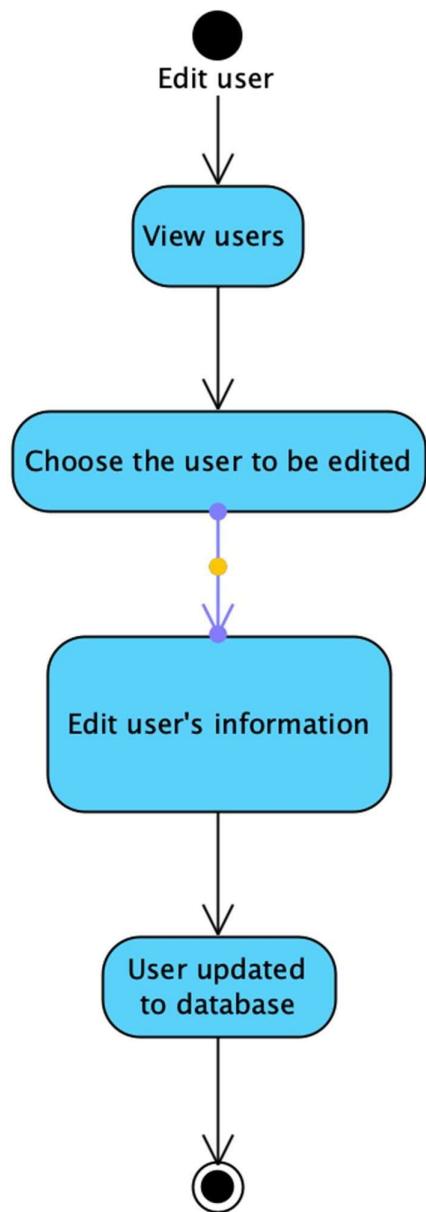
*Add User Activity*

View users

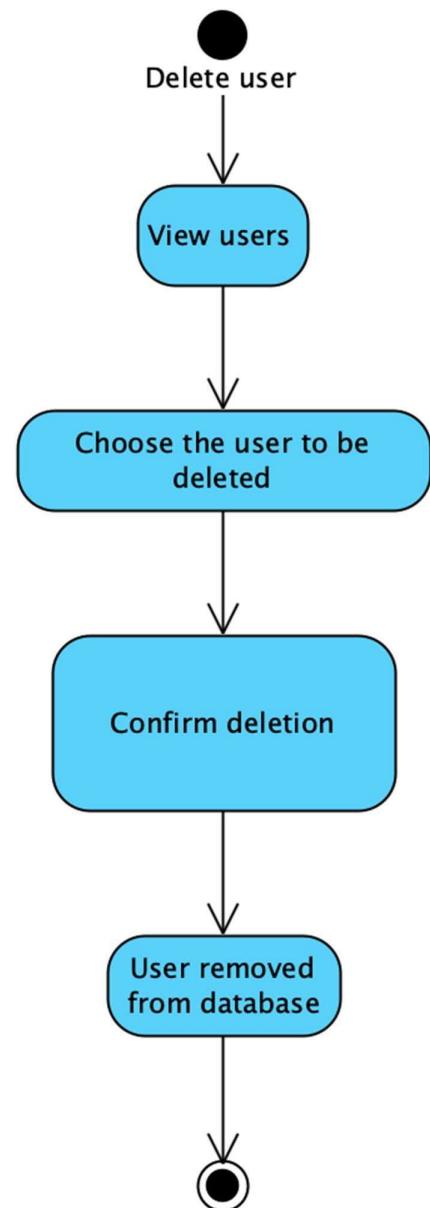
View users

Display the list of users

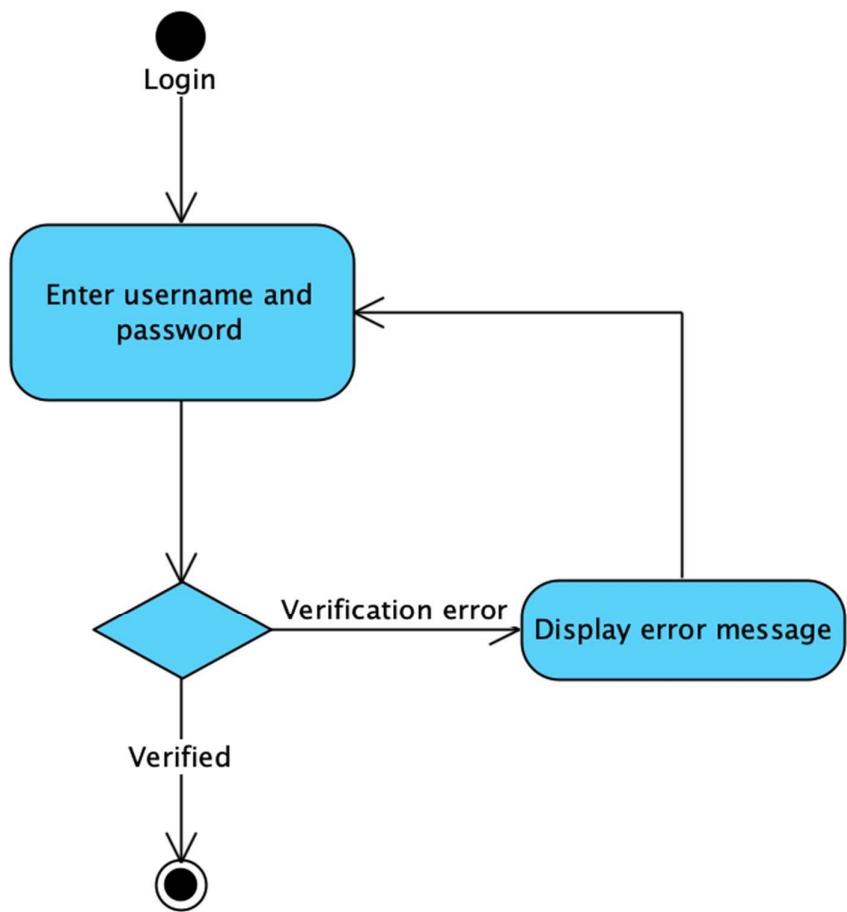
*View User Activity*



*Edit User Activity*



*Delete User Activity*

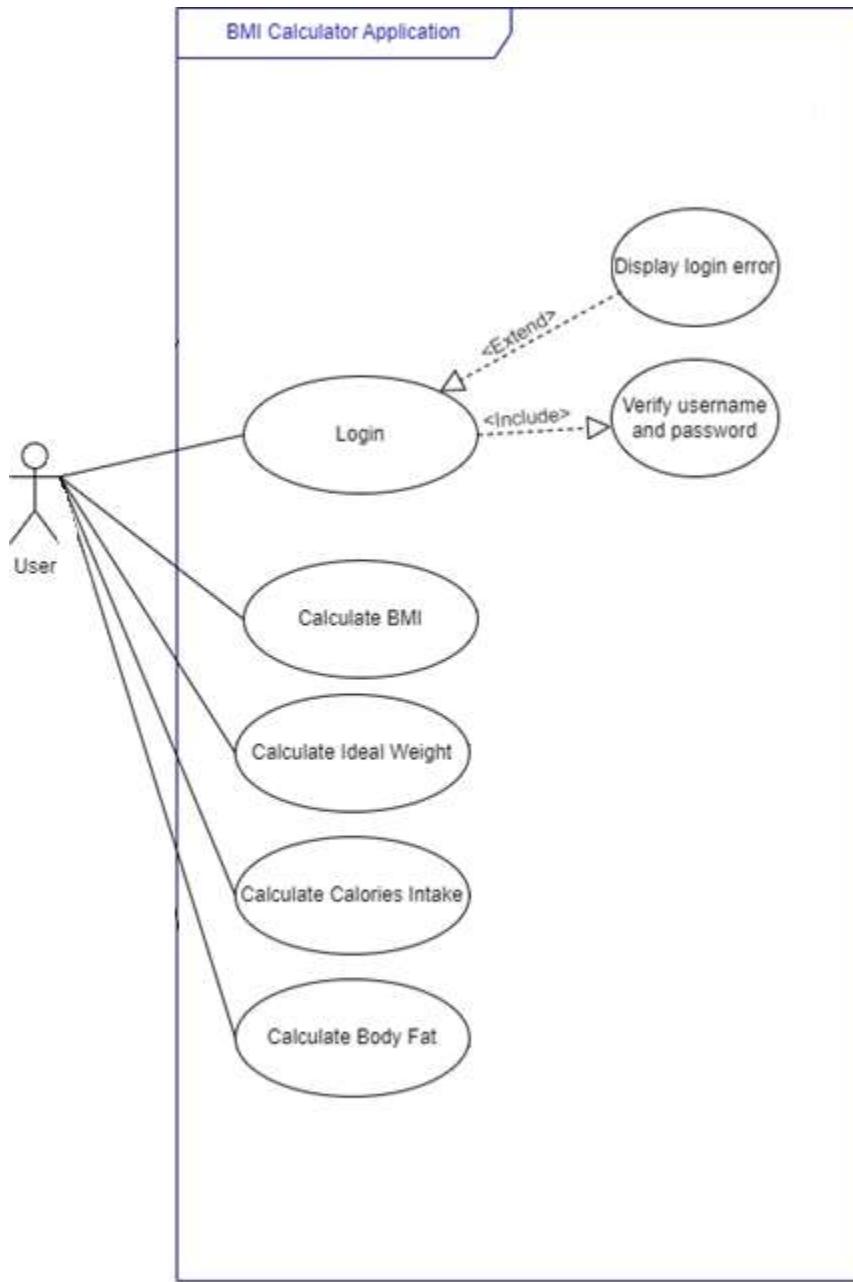


*Login Activity*

## Actors

The following actors are used in the use case diagrams:

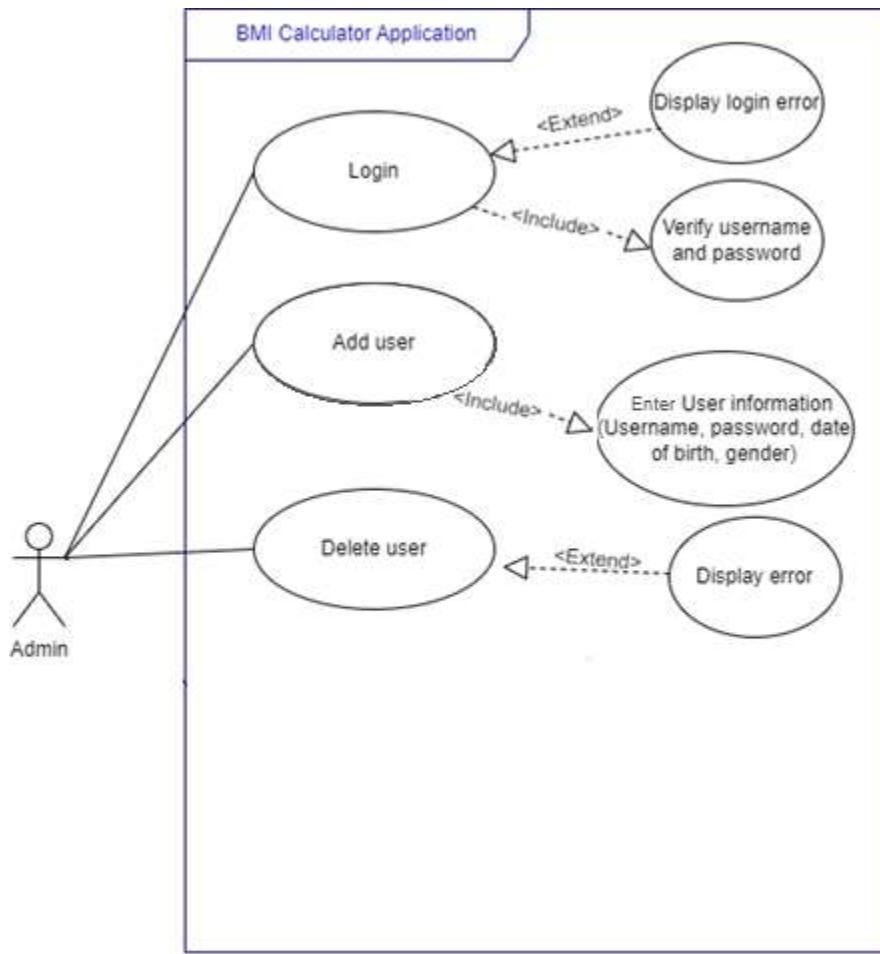
- User (Member)
- Admin



Use case diagram for user's point of view

Name Description	User use cases
<b>Actors</b>	<ul style="list-style-type: none"> <li>• User</li> </ul>
<b>Main Success Scenario</b>	<ul style="list-style-type: none"> <li>• A new user can register their personal details.</li> <li>• An existing user can login using their username and password.</li> <li>• User can calculate their BMI.</li> <li>• User can calculate their Ideal Weight.</li> <li>• User can calculate their Calories Intake.</li> <li>• User can calculate their Bod Fat.</li> </ul>
<b>Alternative Flows</b>	<ul style="list-style-type: none"> <li>• If the user enters incorrect username or password, the application will display an error message.</li> </ul>

---



Use case diagram for admin's point of view

Name Description	User use cases
<b>Actors</b>	<ul style="list-style-type: none"> <li>Admin</li> </ul>
<b>Main Success Scenario</b>	<ul style="list-style-type: none"> <li>Admin can login using their username and password.</li> <li>Adding new user to the application.</li> <li>Deleting an existing user.</li> </ul>
<b>Alternative Flows</b>	<ul style="list-style-type: none"> <li>If the user enters incorrect username or password, the application will display an error message.</li> <li>If admin deletes non-existing user, an error message will pop up.</li> </ul>

## **Design Principles**

- The design should appear to be anticipating the user's subsequent action.
- The user interface (UI) should depict the state of a process or user action.
- For the app, the UI design, aesthetic, and functionality should remain consistent.
- Ensuring that important data access has a mechanism of authentication to prevent data integrity issues.
- The user's demand and action should come first in the content that is provided on the application. Therefore, it must have the pertinent information in the center of the screen.
- Users don't need to worry about how the system works internally. If customers must wait, let them know by using some graphical information, such as spinning wheels, a loading indicator, etc.
- Regardless of the user's age, any text present should be readable.

# **BODYMATH APP**

## **BY SOG CO.**



**RC7: Software Test Case Document &  
Software version control (GIT)**



## **1.0 Introduction**

This document outlines the requirements and specifications for verifying the integrity of the BodyMath App and its database. It describes the test cases used to evaluate the performance of the system in various scenarios and risk factors that may impact its integrity. The purpose of this document is to ensure that the BodyMath system meets the necessary standards for reliability and security.

## **2.0 The functions to be checked for functionality of the mobile application**

**are as follows:**

### **2.1 Welcome page**

- Verify that the “sign in” page is displayed as expected.
- Verify that the email and password fields are functional and can accept user input.
- Verify that the sign in button correctly directs the user to the appropriate page based on their role (admin or member).
- Verify that the app displays an error dialog if the login credentials are incorrect or do not exist.

### **2.2 Administration View**

- Verify that the “ADMIN VIEW” page is displayed upon pressing the “sign in” button in the sign in page.
- Verify that “Create New User” page is displayed upon pressing the “ADD USER” button In the admin view page.
- Verify that the textboxes can receive correct input for Name, Email, Password and Date of Birth.
- Verify that dropdown menu can select gender either male or female.
- Verify that “Create User” button store the new user records to database and navigate him to menu password of member view.

- Verify that “go back” button icon in the “Create New User” page navigates you back to admin view.
- Verify that “Delete User” page is displayed upon pressing the “DELETE USER” button in the admin view page.
- Verify that textbox can receive input for Email.
- Verify that “Delete User” button delete the whole user records in database and displays a message “The user has been deleted successfully”.
- Verify that “go back” button icon in the “Delete User” page navigates you back to admin view.
- Verify that the “Welcome” page is displayed upon scrolling back.

## 2.3 Member View

- Verify that the “Main” page is displayed upon pressing the “sign in” button in the sign in page.
- Verify that “BMI CALCULATOR” page is displayed upon pressing the “BMI CALCULATOR” button in the Main page.
- Verify that the slider to take the height and weight as input in “BMI CALCULATOR” page is working smoothly and accordingly.
- Verify that “go back” button icon in the “BMI CALCULATOR” page navigates you back to main page.
- Verify that the “CALCULATE” button in “BMI CALCULATOR” page navigate to Result page.
- Verify that the calculated value, the status, the advice message, and the Gif are displayed for all 5 conditions of BMI when pressed the “CALCULATE” button.
- Verify that “go back” button icon in the Result page navigates you back to “BMI CALCULATOR” page.
- Verify that the “CALCULATE AGAIN” button in Result page navigate back to “BMI CALCULATOR” page.

- Verify that “Body Fat Calculator” page is displayed upon pressing the “BODY FAT” button in the Main page.
- Verify that “go back” button icon in the “Body Fat Calculator” page navigates you back to main page.
- Verify that increase and decrease button in each card works smoothly as input for height, weight, neck, waist, hip in “Body Fat Calculator” page.
- Verify that the “CALCULATE” button in “Body Fat Calculator” page navigate to Result page.
- Verify that the calculated value, the status and the advice message are displayed for all 4 conditions of Body Fat Calculator when pressed the “CALCULATE” button.
- Verify that “go back” button icon in the Result page navigates you back to “Body Fat Calculator” page.
- Verify that the “CALCULATE AGAIN” button in Result page navigate back to “Body Fat Calculator” page.
- Verify that “IDEAL WEIGHT” page is displayed upon pressing the “IDEAL WEIGHT” button in the Main page.
- Verify that “go back” button icon in the “IDEAL WEIGHT” page navigates you back to main page.
- Verify that the slider to take the height and weight as input in “IDEAL WEIGHT” page is working smoothly and accordingly.
- Verify that the “CALCULATE” button in “IDEAL WEIGHT” page navigate to Result page.
- Verify that the calculated value and the advice message are displayed for both conditions of ideal weight calculator when pressed the “CALCULATE” button.
- Verify that “go back” button icon in the Result page navigates you back to “IDEAL WEIGHT” page.
- Verify that the “CALCULATE AGAIN” button in Result page navigate back to “IDEAL WEIGHT” page.
- Verify that “CALORIE INTAKE” page is displayed upon pressing the “CALORIE INTAKE” button in the Main page.

- Verify that “go back” button icon in the “CALORIE INTAKE” page navigates you back to main page.
- Verify that the slider to take the height and weight as input in “CALORIE INTAKE” page is working smoothly and accordingly.
- Verify that the “CALCULATE” button in “CALORIE INTAKE” page navigate to Result page.
- Verify that the calculated value are displayed for “CALORIE INTAKE” calculator when pressed the “CALCULATE” button.
- Verify that “go back” button icon in the Result page navigates you back to “CALORIE INTAKE” page.
- Verify that the “CALCULATE AGAIN” button in Result page navigate back to “CALORIE INTAKE” page.

## **3.0 Function Test Case Dependencies/ Assumptions/ Limitations**

### **3.1 Dependencies**

- Firebase Database to save records and to choose role for every account as admin or member.
- Internet connection.

### **3.1 Limitations**

- Due to time constraints, not all functions intended to be implemented could be implemented.
- Not all invalid inputs could be accounted for and have error message allocated to them.

### **3.3 Assumptions**

- Any records that update database information entered into the application is correct.  
Assuming that certain input data has been correctly formatted and no human error.

**Test Cases:**

WELCOME PAGE				
ID	Test Case Description	Expected Result	Status	Remarks
TC1	Verify that the “sign in” page is displayed as expected	“Sign in” page is displayed as expected		
TC2	Verify that the email and password fields are functional and can accept user input.	Email and password fields are functional and can accept user input.		
TC3	Verify that the sign in button correctly directs the user to the appropriate page based on their role (admin or member).	Sign in button correctly directs the user to the appropriate page based on their role (admin or member).		
TC4	Verify that the app displays an error dialog if the login credentials are incorrect or do not exist.	The app displays an error dialog if the login credentials are incorrect or do not exist.		

**Administration View:**

Administration View				
ID	Test Case Description	Expected Result	Status	Remarks
TC5	Verify that the “ADMIN VIEW” page is displayed upon pressing the “sign in” button in the sign in page	The “ADMIN VIEW” page is displayed upon pressing the “sign in” button in the sign in page		
TC6	Verify that “Create New User” page is displayed upon pressing	“Create New User” page is displayed upon pressing the “ADD		

	the “ADD USER” button in the admin view page.	“USER” button In the admin view page.		
TC7	Verify that the textboxes can receive correct input for Name, Email, Password and Date of Birth.	The textboxes can receive correct input for Name, Email, Password and Date of Birth.		
TC8	Verify that dropdown menu can select gender either male or female.	Verify that dropdown menu can select gender either male or female.		
TC9	Verify that “Create User” button store the new user records to database and navigate him to menu password of member view.	“Create User” button store the new user records to database and navigate him to menu password of member view.		
TC10	Verify that “go back” button icon in the “Create New User” page navigates you back to admin view.	“go back” button icon in the “Create New User” page navigates you back to admin view.		
TC11	Verify that “Delete User” page is displayed upon pressing the “DELETE USER” button in the admin view page.	“Delete User” page is displayed upon pressing the “DELETE USER” button in the admin view page.		
TC12	Verify that textbox can receive input for email.	Textbox can receive input for email.		
TC13	Verify that “Delete User” button delete the whole user records in database and displays a message “The user has been deleted successfully”.	“Delete User” button delete the whole user records in database and displays a message “The user has been deleted successfully”.		

TC14	Verify that “go back” button icon in the “Delete User” page navigates you back to admin view.	“go back” button icon in the “Delete User” page navigates you back to admin view.		
TC15	Verify that the “Welcome” page is displayed upon scrolling back.	Verify that the “Welcome” page is displayed upon scrolling back.		

### Member View:

Member View				
ID	Test Case Description	Expected Result	Status	Remarks
TC16	Verify that “BMI CALCULATOR” page is displayed upon pressing the “BMI CALCULATOR” button in the Main page.	“BMI CALCULATOR” page is displayed upon pressing the “BMI CALCULATOR” button in the Main page.		
TC17	Verify that the slider to take the height and weight as input in “BMI CALCULATOR” page is working smoothly and accordingly.	The slider to take the height and weight as input in “BMI CALCULATOR” page is working smoothly and accordingly.		
TC18	Verify that “go back” button icon in the “BMI CALCULATOR” page navigates you back to main page.	“go back” button icon in the “BMI CALCULATOR” page navigates you back to main page.		
TC19	Verify that the “CALCULATE” button in “BMI CALCULATOR” page navigate to Result page.	“CALCULATE” button in “BMI CALCULATOR” page navigate to Result page.		
TC20	Verify that the calculated value, the status, the advice message,	The calculated value, the status, the advice message, and the Gif		

	and the Gif are displayed for all 5 conditions of BMI when pressed the “CALCULATE” button.	are displayed for all 5 conditions of BMI when pressed the “CALCULATE” button.		
TC21	Verify that “go back” button icon in the Result page navigates you back to “BMI CALCULATOR” page.	“go back” button icon in the Result page navigates you back to “BMI CALCULATOR” page.		
TC22	Verify that the “CALCULATE AGAIN” button in Result page navigate back to “BMI CALCULATOR” page.	The “CALCULATE AGAIN” button in Result page navigate back to “BMI CALCULATOR” page.		
TC23	Verify that “Body Fat Calculator” page is displayed upon pressing the “BODY FAT” button in the Main page.	“Body Fat Calculator” page is displayed upon pressing the “BODY FAT” button in the Main page.		
TC24	Verify that “go back” button icon in the “Body Fat Calculator” page navigates you back to main page.	“go back” button icon in the “Body Fat Calculator” page navigates you back to main page.		
TC25	Verify that increase and decrease button in each card works smoothly as input for height, weight, neck, waist, hip in “Body Fat Calculator” page.	Increase and decrease button in each card works smoothly as input for height, weight, neck, waist, hip in “Body Fat Calculator” page.		
TC26	Verify that the “CALCULATE” button in “Body Fat Calculator” page navigate to Result page.	The “CALCULATE” button in “Body Fat Calculator” page navigate to Result page.		
TC27	Verify that the calculated value, the status and the advice message are displayed for all 4	The calculated value, the status and the advice message are displayed for all 4 conditions of		

	conditions of Body Fat Calculator when pressed the “CALCULATE” button.	Body Fat Calculator when pressed the “CALCULATE” button.		
TC28	Verify that “go back” button icon in the Result page navigates you back to “Body Fat Calculator” page.	“go back” button icon in the Result page navigates you back to “Body Fat Calculator” page.		
TC29	Verify that the “CALCULATE AGAIN” button in Result page navigate back to “Body Fat Calculator” page.	The “CALCULATE AGAIN” button in Result page navigate back to “Body Fat Calculator” page.		
TC30	Verify that “IDEAL WEIGHT” page is displayed upon pressing the “IDEAL WEIGHT” button in the Main page.	The “IDEAL WEIGHT” page is displayed upon pressing the “IDEAL WEIGHT” button in the Main page.		
TC31	Verify that “go back” button icon in the “IDEAL WEIGHT” page navigates you back to main page.	“go back” button icon in the “IDEAL WEIGHT” page navigates you back to main page.		
TC32	Verify that the slider to take the height and weight as input in “IDEAL WEIGHT” page is working smoothly and accordingly.	Verify that the slider to take the height and weight as input in “IDEAL WEIGHT” page is working smoothly and accordingly.		
TC33	Verify that the “CALCULATE” button in “IDEAL WEIGHT” page navigate to Result page.	The “CALCULATE” button in “IDEAL WEIGHT” page navigate to Result page.		
TC34	Verify that the calculated value and the advice message are displayed for both conditions of ideal weight calculator when	The calculated value and the advice message are displayed for both conditions of ideal weight		

	pressed the “CALCULATE” button.	calculator when pressed the “CALCULATE” button.		
TC35	Verify that “go back” button icon in the Result page navigates you back to “IDEAL WEIGHT” page.	“go back” button icon in the Result page navigates you back to “IDEAL WEIGHT” page.		
TC36	Verify that the “CALCULATE AGAIN” button in Result page navigate back to “IDEAL WEIGHT” page.	The “CALCULATE AGAIN” button in Result page navigate back to “IDEAL WEIGHT” page.		
TC37	Verify that “CALORIE INTAKE” page is displayed upon pressing the “CALORIE INTAKE” button in the Main page.	“CALORIE INTAKE” page is displayed upon pressing the “CALORIE INTAKE” button in the Main page.		
TC38	Verify that “go back” button icon in the “CALORIE INTAKE” page navigates you back to main page.	“go back” button icon in the “CALORIE INTAKE” page navigates you back to main page.		
TC39	Verify that the slider to take the height and weight as input in “CALORIE INTAKE” page is working smoothly and accordingly.	The slider to take the height and weight as input in “CALORIE INTAKE” page is working smoothly and accordingly.		
TC40	Verify that the “CALCULATE” button in “CALORIE INTAKE” page navigate to Result page.	The “CALCULATE” button in “CALORIE INTAKE” page navigate to Result page.		
TC41	Verify that the calculated value is displayed for “CALORIE INTAKE” calculator when	The calculated value is displayed for “CALORIE INTAKE” calculator when pressed the “CALCULATE” button.		

	pressed the “CALCULATE” button.			
TC42	Verify that “go back” button icon in the Result page navigates you back to “CALORIE INTAKE” page.	“go back” button icon in the Result page navigates you back to “CALORIE INTAKE” page.		
TC42	Verify that the “CALCULATE AGAIN” button in Result page navigate back to “CALORIE INTAKE” page.	The “CALCULATE AGAIN” button in Result page navigate back to “CALORIE INTAKE” page.		

## 5.0 Software version control (GIT):

Github Page:

The screenshot shows the GitHub repository page for 'bmi-calculator-software'. The repository is private, has 4 branches, and 1 tag. The master branch has 25 commits. The README.md file contains the following content:

```
yyss-yss finished project

fix login screen, navigation problem
images and launcher icon
fix login screen, navigation problem
finished project
test commit
test commit
test commit
fix login screen, navigation problem
fix login screen, navigation problem
```

The repository has 0 stars, 1 watching, and 0 forks. It includes a 'About' section with no description, website, or topics provided. There is a 'Releases' section with one release named 'BodyMath First Release' (Latest, 2 days ago). The 'Packages' section indicates no packages published. The 'Contributors' section lists yyss-yss, BashirAbu Bashir, and Marawan2002 Marawan. The 'Languages' section shows Dart at 99.4% and Other at 0.6%.

master			
Commits on Jan 5, 2023			
<b>finished project</b> yss-yss committed 2 days ago			
<b>images and launcher icon</b> Marawan2002 committed 2 days ago			
<b>fix login screen, navigation problem</b> Marawan2002 committed 2 days ago			
<b>minor fix</b> yss-yss committed 2 days ago			
<b>all calculators complete</b> yss-yss committed 2 days ago			
Commits on Jan 3, 2023			
<b>Calorie Intake Calculator</b> BashirAbu committed 4 days ago			
<b>bmi, body fat, and ideal weight complete</b> yss-yss committed 4 days ago			
Commits on Jan 2, 2023			
<b>added gender fetching for body fat</b> yss-yss committed 5 days ago			
Commits on Jan 1, 2023			
<b>ideal weight</b> Marawan2002 committed last week			
<b>Merge branch 'ysP' into bashir</b> BashirAbu committed last week			
<b>typo</b> BashirAbu committed last week			
<b>organizing stuff</b> BashirAbu committed last week			
<b>body fat calculator (working now)</b> BashirAbu committed last week			
<b>added methods that allows data to be read from firestore</b> yss-yss committed last week			

Commits on Dec 30, 2022

- bodyfat(not working)  
BashirAbu committed last week
- testing  
BashirAbu committed last week
- Layout of selection page  
yys-yss committed last week

Commits on Dec 29, 2022

- Implementation of layout and basic functionality of BMI Calculator page  
yys-yss committed last week

Commits on Dec 27, 2022

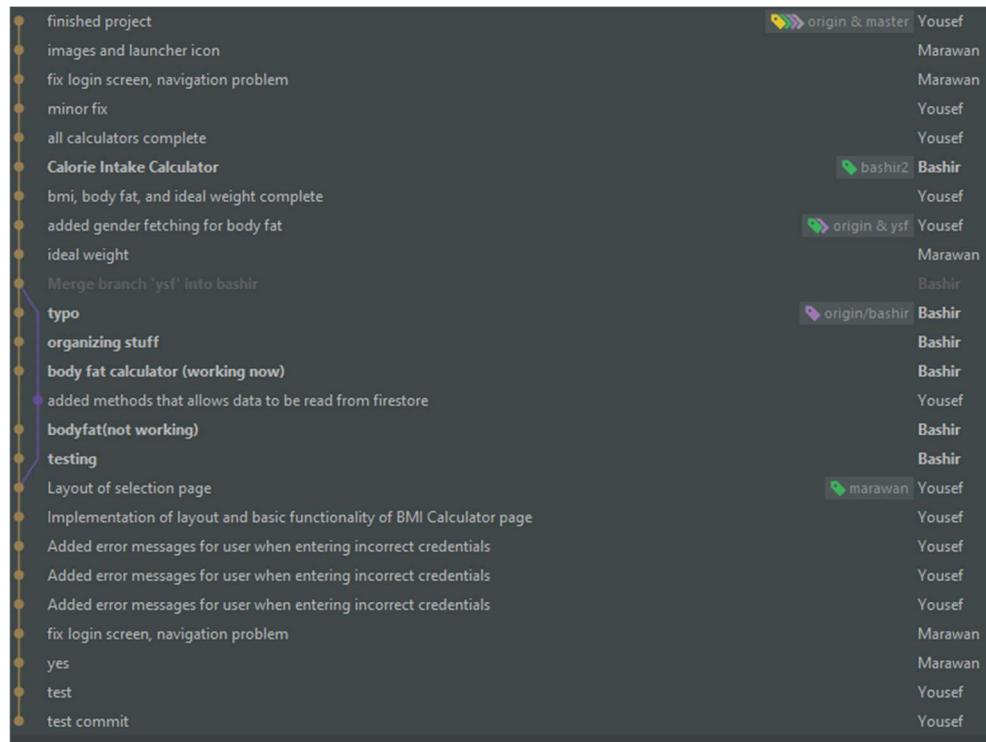
- Added error messages for user when entering incorrect credentials  
yys-yss committed 2 weeks ago
- Added error messages for user when entering incorrect credentials  
yys-yss committed 2 weeks ago
- Added error messages for user when entering incorrect credentials  
yys-yss committed 2 weeks ago
- fix login screen, navigation problem  
Marawan2002 committed 2 weeks ago
- yes  
Marawan2002 committed 2 weeks ago

Commits on Dec 24, 2022

- test  
yys-yss committed 2 weeks ago
- test commit  
yys-yss committed 2 weeks ago

[Newer](#) [Older](#)

## 5.1 Git Branch Graph:



# **BODYMATH APP**

## **BY SOG CO.**



## **RC8: Software Test Case Execution**

### **Document**



## **1.0 Introduction**

This document outlines the requirements and specifications for verifying the integrity of the BodyMath App and its database. It describes the test cases used to evaluate the performance of the system in various scenarios and risk factors that may impact its integrity. The purpose of this document is to ensure that the BodyMath system meets the necessary standards for reliability and security.

## **2.0 The functions to be checked for functionality of the mobile application are as follows:**

### **2.1 Welcome page**

- Verify that the “sign in” page is displayed as expected.
- Verify that the email and password fields are functional and can accept user input.
- Verify that the sign in button correctly directs the user to the appropriate page based on their role (admin or member).
- Verify that the app displays an error dialog if the login credentials are incorrect or do not exist.

### **2.2 Administration View**

- Verify that the “ADMIN VIEW” page is displayed upon pressing the “sign in” button in the sign in page.
- Verify that “Create New User” page is displayed upon pressing the “ADD USER” button In the admin view page.
- Verify that the textboxes can receive correct input for Name, Email, Password and Date of Birth.
- Verify that dropdown menu can select gender either male or female.
- Verify that “Create User” button store the new user records to database and navigate him to menu password of member view.
- Verify that “go back” button icon in the “Create New User” page navigates you back to admin view.

- Verify that “Delete User” page is displayed upon pressing the “DELETE USER” button in the admin view page.
- Verify that textbox can receive input for Email.
- Verify that “Delete User” button delete the whole user records in database and displays a message “The user has been deleted successfully”.
- Verify that “go back” button icon in the “Delete User” page navigates you back to admin view.
- Verify that the “Welcome” page is displayed upon scrolling back.

## 2.3 Member View

- Verify that the “Main” page is displayed upon pressing the “sign in” button in the sign in page.
- Verify that “BMI CALCULATOR” page is displayed upon pressing the “BMI CALCULATOR” button in the Main page.
- Verify that the slider to take the height and weight as input in “BMI CALCULATOR” page is working smoothly and accordingly.
- Verify that “go back” button icon in the “BMI CALCULATOR” page navigates you back to main page.
- Verify that the “CALCULATE” button in “BMI CALCULATOR” page navigate to Result page.
- Verify that the calculated value, the status, the advice message, and the Gif are displayed for all 5 conditions of BMI when pressed the “CALCULATE” button.
- Verify that “go back” button icon in the Result page navigates you back to “BMI CALCULATOR” page.
- Verify that the “CALCULATE AGAIN” button in Result page navigate back to “BMI CALCULATOR” page.
- Verify that “Body Fat Calculator” page is displayed upon pressing the “BODY FAT” button in the Main page.

- Verify that “go back” button icon in the “Body Fat Calculator” page navigates you back to main page.
- Verify that increase and decrease button in each card works smoothly as input for height, weight, neck, waist, hip in “Body Fat Calculator” page.
- Verify that the “CALCULATE” button in “Body Fat Calculator” page navigate to Result page.
- Verify that the calculated value, the status and the advice message are displayed for all 4 conditions of Body Fat Calculator when pressed the “CALCULATE” button.
- Verify that “go back” button icon in the Result page navigates you back to “Body Fat Calculator” page.
- Verify that the “CALCULATE AGAIN” button in Result page navigate back to “Body Fat Calculator” page.
- Verify that “IDEAL WEIGHT” page is displayed upon pressing the “IDEAL WEIGHT” button in the Main page.
- Verify that “go back” button icon in the “IDEAL WEIGHT” page navigates you back to main page.
- Verify that the slider to take the height and weight as input in “IDEAL WEIGHT” page is working smoothly and accordingly.
- Verify that the “CALCULATE” button in “IDEAL WEIGHT” page navigate to Result page.
- Verify that the calculated value and the advice message are displayed for both conditions of ideal weight calculator when pressed the “CALCULATE” button.
- Verify that “go back” button icon in the Result page navigates you back to “IDEAL WEIGHT” page.
- Verify that the “CALCULATE AGAIN” button in Result page navigate back to “IDEAL WEIGHT” page.
- Verify that “CALORIE INTAKE” page is displayed upon pressing the “CALORIE INTAKE” button in the Main page.
- Verify that “go back” button icon in the “CALORIE INTAKE” page navigates you back to main page.

- Verify that the slider to take the height and weight as input in “CALORIE INTAKE” page is working smoothly and accordingly.
- Verify that the “CALCULATE” button in “CALORIE INTAKE” page navigate to Result page.
- Verify that the calculated value are displayed for “CALORIE INTAKE” calculator when pressed the “CALCULATE” button.
- Verify that “go back” button icon in the Result page navigates you back to “CALORIE INTAKE” page.
- Verify that the “CALCULATE AGAIN” button in Result page navigate back to “CALORIE INTAKE” page.

## **3.0 Function Test Case Dependencies/ Assumptions/ Limitations**

### **3.1 Dependencies**

- Firebase Database to save records and to choose role for every account as admin or member.
- Internet connection.

### **3.1 Limitations**

- Due to time constraints, not all functions intended to be implemented could be implemented.
- Not all invalid inputs could be accounted for and have error message allocated to them.

### **3.3 Assumptions**

- Any records that update database information entered into the application is correct.  
Assuming that certain input data has been correctly formatted and no human error.

**Test Cases:**

WELCOME PAGE				
ID	Test Case Description	Expected Result	Status	Remarks
TC1	Verify that the “sign in” page is displayed as expected	“Sign in” page is displayed as expected	Pass	
TC2	Verify that the email and password fields are functional and can accept user input.	Email and password fields are functional and can accept user input.	Pass	
TC3	Verify that the sign in button correctly directs the user to the appropriate page based on their role (admin or member).	Sign in button correctly directs the user to the appropriate page based on their role (admin or member).	Pass	
TC4	Verify that the app displays an error dialog if the login credentials are incorrect or do not exist.	The app displays an error dialog if the login credentials are incorrect or do not exist.	Pass	

**Administration View:**

Administration View				
ID	Test Case Description	Expected Result	Status	Remarks
TC5	Verify that the “ADMIN VIEW” page is displayed upon pressing the “sign in” button in the sign in page	The “ADMIN VIEW” page is displayed upon pressing the “sign in” button in the sign in page	Pass	
TC6	Verify that “Create New User” page is displayed upon pressing	“Create New User” page is displayed upon pressing the “ADD	Pass	

	the “ADD USER” button in the admin view page.	“USER” button In the admin view page.		
TC7	Verify that the textboxes can receive correct input for Name, Email, Password and Date of Birth.	The textboxes can receive correct input for Name, Email, Password and Date of Birth.	Pass	
TC8	Verify that dropdown menu can select gender either male or female.	Verify that dropdown menu can select gender either male or female.	Pass	
TC9	Verify that “Create User” button store the new user records to database and navigate him to menu password of member view.	“Create User” button store the new user records to database and navigate him to menu password of member view.	Pass	
TC10	Verify that “go back” button icon in the “Create New User” page navigates you back to admin view.	“go back” button icon in the “Create New User” page navigates you back to admin view.	Pass	
TC11	Verify that “Delete User” page is displayed upon pressing the “DELETE USER” button in the admin view page.	“Delete User” page is displayed upon pressing the “DELETE USER” button in the admin view page.	Pass	
TC12	Verify that textbox can receive input for email.	Textbox can receive input for email.	Pass	
TC13	Verify that “Delete User” button delete the whole user records in database and displays a message “The user has been deleted successfully”.	“Delete User” button delete the whole user records in database and displays a message “The user has been deleted successfully”.	Pass	

TC14	Verify that “go back” button icon in the “Delete User” page navigates you back to admin view.	“go back” button icon in the “Delete User” page navigates you back to admin view.	Pass	
TC15	Verify that the “Welcome” page is displayed upon scrolling back.	Verify that the “Welcome” page is displayed upon scrolling back.	Pass	

### Member View:

Member View				
ID	Test Case Description	Expected Result	Status	Remarks
TC16	Verify that “BMI CALCULATOR” page is displayed upon pressing the “BMI CALCULATOR” button in the Main page.	“BMI CALCULATOR” page is displayed upon pressing the “BMI CALCULATOR” button in the Main page.	Pass	
TC17	Verify that the slider to take the height and weight as input in “BMI CALCULATOR” page is working smoothly and accordingly.	The slider to take the height and weight as input in “BMI CALCULATOR” page is working smoothly and accordingly.	Pass	
TC18	Verify that “go back” button icon in the “BMI CALCULATOR” page navigates you back to main page.	“go back” button icon in the “BMI CALCULATOR” page navigates you back to main page.	Pass	
TC19	Verify that the “CALCULATE” button in “BMI CALCULATOR” page navigate to Result page.	“CALCULATE” button in “BMI CALCULATOR” page navigate to Result page.	Pass	
TC20	Verify that the calculated value, the status, the advice message,	The calculated value, the status, the advice message, and the Gif	Pass	

	and the Gif are displayed for all 5 conditions of BMI when pressed the “CALCULATE” button.	are displayed for all 5 conditions of BMI when pressed the “CALCULATE” button.		
TC21	Verify that “go back” button icon in the Result page navigates you back to “BMI CALCULATOR” page.	“go back” button icon in the Result page navigates you back to “BMI CALCULATOR” page.	Pass	
TC22	Verify that the “CALCULATE AGAIN” button in Result page navigate back to “BMI CALCULATOR” page.	The “CALCULATE AGAIN” button in Result page navigate back to “BMI CALCULATOR” page.	Pass	
TC23	Verify that “Body Fat Calculator” page is displayed upon pressing the “BODY FAT” button in the Main page.	“Body Fat Calculator” page is displayed upon pressing the “BODY FAT” button in the Main page.	Pass	
TC24	Verify that “go back” button icon in the “Body Fat Calculator” page navigates you back to main page.	“go back” button icon in the “Body Fat Calculator” page navigates you back to main page.	Pass	
TC25	Verify that increase and decrease button in each card works smoothly as input for height, weight, neck, waist, hip in “Body Fat Calculator” page.	Increase and decrease button in each card works smoothly as input for height, weight, neck, waist, hip in “Body Fat Calculator” page.	Pass	
TC26	Verify that the “CALCULATE” button in “Body Fat Calculator” page navigate to Result page.	The “CALCULATE” button in “Body Fat Calculator” page navigate to Result page.	Pass	
TC27	Verify that the calculated value, the status and the advice message are displayed for all 4	The calculated value, the status and the advice message are displayed for all 4 conditions of	Pass	

	conditions of Body Fat Calculator when pressed the “CALCULATE” button.	Body Fat Calculator when pressed the “CALCULATE” button.		
TC28	Verify that “go back” button icon in the Result page navigates you back to “Body Fat Calculator” page.	“go back” button icon in the Result page navigates you back to “Body Fat Calculator” page.	Pass	
TC29	Verify that the “CALCULATE AGAIN” button in Result page navigate back to “Body Fat Calculator” page.	The “CALCULATE AGAIN” button in Result page navigate back to “Body Fat Calculator” page.	Pass	
TC30	Verify that “IDEAL WEIGHT” page is displayed upon pressing the “IDEAL WEIGHT” button in the Main page.	The “IDEAL WEIGHT” page is displayed upon pressing the “IDEAL WEIGHT” button in the Main page.	Pass	
TC31	Verify that “go back” button icon in the “IDEAL WEIGHT” page navigates you back to main page.	“go back” button icon in the “IDEAL WEIGHT” page navigates you back to main page.	Pass	
TC32	Verify that the slider to take the height and weight as input in “IDEAL WEIGHT” page is working smoothly and accordingly.	Verify that the slider to take the height and weight as input in “IDEAL WEIGHT” page is working smoothly and accordingly.	Pass	
TC33	Verify that the “CALCULATE” button in “IDEAL WEIGHT” page navigate to Result page.	The “CALCULATE” button in “IDEAL WEIGHT” page navigate to Result page.	Pass	
TC34	Verify that the calculated value and the advice message are displayed for both conditions of ideal weight calculator when	The calculated value and the advice message are displayed for both conditions of ideal weight	Pass	

	pressed the “CALCULATE” button.	calculator when pressed the “CALCULATE” button.		
TC35	Verify that “go back” button icon in the Result page navigates you back to “IDEAL WEIGHT” page.	“go back” button icon in the Result page navigates you back to “IDEAL WEIGHT” page.	Pass	
TC36	Verify that the “CALCULATE AGAIN” button in Result page navigate back to “IDEAL WEIGHT” page.	The “CALCULATE AGAIN” button in Result page navigate back to “IDEAL WEIGHT” page.	Pass	
TC37	Verify that “CALORIE INTAKE” page is displayed upon pressing the “CALORIE INTAKE” button in the Main page.	“CALORIE INTAKE” page is displayed upon pressing the “CALORIE INTAKE” button in the Main page.	Pass	
TC38	Verify that “go back” button icon in the “CALORIE INTAKE” page navigates you back to main page.	“go back” button icon in the “CALORIE INTAKE” page navigates you back to main page.	Pass	
TC39	Verify that the slider to take the height and weight as input in “CALORIE INTAKE” page is working smoothly and accordingly.	The slider to take the height and weight as input in “CALORIE INTAKE” page is working smoothly and accordingly.	Pass	
TC40	Verify that the “CALCULATE” button in “CALORIE INTAKE” page navigate to Result page.	The “CALCULATE” button in “CALORIE INTAKE” page navigate to Result page.	Pass	
TC41	Verify that the calculated value is displayed for “CALORIE INTAKE” calculator when	The calculated value is displayed for “CALORIE INTAKE” calculator when pressed the “CALCULATE” button.	Pass	

	pressed the “CALCULATE” button.			
TC42	Verify that “go back” button icon in the Result page navigates you back to “CALORIE INTAKE” page.	“go back” button icon in the Result page navigates you back to “CALORIE INTAKE” page.	Pass	
TC42	Verify that the “CALCULATE AGAIN” button in Result page navigate back to “CALORIE INTAKE” page.	The “CALCULATE AGAIN” button in Result page navigate back to “CALORIE INTAKE” page.	Pass	

## Firebase (Database):

### Application Accounts:

The screenshot shows the Firebase Authentication console. On the left, there's a sidebar with project settings like "Project Overview", "Build", "Release and monitor", "Analytics", "Engage", and "All products". Below that is a "Customise your navigation" section. At the bottom, it says "Spark No cost \$0/month" and "Upgrade". The main area is titled "Authentication" and has tabs for "Users", "Sign-in method", "Templates", "Usage", and "Settings". A search bar at the top says "Search by email address, phone number or user UID". Below it is a table with columns: Identifier, Providers, Created, Signed in, and User UID. The table lists ten users with their respective details.

Identifier	Providers	Created	Signed in	User UID
ysf@mrwn.com	Email	7 Jan 2023	7 Jan 2023	HYEB0dHVVvDcy2AayCU1l2byIZGZ2
muad@gmail.com	Email	7 Jan 2023	7 Jan 2023	IN90tIiigKTDTac8sih41CGHLj2
bashir@hejij.com	Email	7 Jan 2023	7 Jan 2023	fz2Q2qjxKUjqQ6xk3yR586xKD0C3
habd@habd.com	Email	7 Jan 2023	7 Jan 2023	UlgvMyds1xeQzXMFcWaBD0EhE62
khalid@khalid.com	Email	6 Jan 2023	6 Jan 2023	9kvwASwIBwdw5sVQpMeBOEfVx7...
azwan@gmail.com	Email	6 Jan 2023	6 Jan 2023	Vix5karsFTWunrefzlmTXDvliBQ2
bashir@gmail.com	Email	5 Jan 2023	5 Jan 2023	iBM654kGgTewFUtzpbfWhdkdSgFg1
123@456.com	Email	5 Jan 2023	5 Jan 2023	wiTm0dA88UM2H7E15xMhq0E...
ali@ali.com	Email	27 Dec 2022	7 Jan 2023	EEGcAKCqyoV4vrJJmmmuUEGMx...
test@test.com	Email	27 Dec 2022	7 Jan 2023	WhK8UoCP7zezwTxu3eNjgxRP1

Rows per page: 50 | 1 - 10 of 10

## Firebase Database:

The screenshot shows the Cloud Firestore console. The left sidebar includes "Project Overview", "Build", "Release and monitor", "Analytics", "Engage", and "All products". It also features a "Customise your navigation" section and "Spark No cost \$0/month" information. The main area is titled "Cloud Firestore" and has tabs for "Data", "Rules", "Indexes", and "Usage". A "Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing" message with a "Configure App Check" button is displayed. The "Data" tab is selected, showing a "Panel view" of a "users" collection. The collection contains a single document named "EEGcAKCqyoV4...". This document has fields: "admin": true, "dob": "12 June 1992 at 00:00:00 UTC+", "email": "ali@ali.com", "gender": "female", and "name": "Ali Ali". There are also buttons for "Start collection" and "Add field".

## BodyMath Code:

### main.dart

```
import 'package:bmi_calculator/pages/admin_page.dart';
import 'package:bmi_calculator/pages/selection_page.dart';
import 'package:bmi_calculator/routes.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'pages/login_page.dart';
import 'package:firebase_auth/firebase_auth.dart';

//Testing branch stuff
Future<void> main() async {
    WidgetsFlutterBinding.ensureInitialized();
    await Firebase.initializeApp();
    runApp(AuthWidget());
}

final navigatorKey = GlobalKey<NavigatorState>();

class AuthWidget extends StatelessWidget {
    @override
    Widget build(BuildContext context) => MaterialApp(
        debugShowCheckedModeBanner: false,
        home: Scaffold(
            body: StreamBuilder<User?>(
                stream: FirebaseAuth.instance.authStateChanges(),
                builder: (context, snapshot) {
                    if (snapshot.hasData) {
                        // Get the user's document from the users collection
                        final uid = snapshot.data?.uid;
                        return StreamBuilder<DocumentSnapshot>(
                            stream: FirebaseFirestore.instance
                                .collection('users')
                                .doc(uid)
                                .snapshots(),
                            builder: (context, userSnapshot) {
                                if (userSnapshot.hasData) {
                                    // Check the value of the admin field
                                    if (userSnapshot.data!['admin']) {
                                        return AdminPage();
                                    } else {
                                        return Routes();
                                    }
                                } else {
                                    return CircularProgressIndicator();
                                }
                            },
                        );
                    } else {
                        return LoginPage();
                    }
                },
            ),
        ),
    );
}
```

```
) ;  
}
```

## routes.dart

```
import 'package:bmi_calculator/constants.dart';  
import 'package:bmi_calculator/main.dart';  
import 'package:bmi_calculator/pages/add_user_page.dart';  
import 'package:bmi_calculator/pages/admin_page.dart';  
import 'package:bmi_calculator/pages/ibw_calculator.dart';  
import 'package:bmi_calculator/pages/offline_mode.dart';  
import 'package:flutter/material.dart';  
import 'pages/selection_page.dart';  
import 'pages/bmi_calculator.dart';  
import 'pages/body_fat_calculator.dart';  
import 'pages/calorie_intake_calculator.dart';  
  
class Routes extends StatelessWidget {  
    @override  
    Widget build(BuildContext context) {  
        return MaterialApp(  
            debugShowCheckedModeBanner: false,  
            theme: kAppTheme,  
            routes: {  
                '/': (context) => SelectionPage(),  
                '/bmiCalculator': (context) => BMICalculator(),  
                '/bodyFatCalculator': (context) => BodyFatCalculator(),  
                '/ibwCalculator': (context) => IBWCalculator(),  
                '/CalorieIntakeCalculator': (context) => CalorieIntakeCalculator(),  
                '/offlineMode': (context) => OfflineMode(),  
                '/adminPage': (context) => AdminPage(),  
                '/authWidget': (context) => AuthWidget(),  
                '/addUserPage': (context) => AddUserPage(),  
            },  
            initialRoute: '/',  
        );  
    }  
}
```

## database\_brain.dart

```
import 'package:bmi_calculator/pages/selection_page.dart';  
import 'package:flutter/material.dart';  
import 'package:cloud_firestore/cloud_firestore.dart';  
import 'package:firebase_auth/firebase_auth.dart';  
  
class DatabaseBrain {  
    late final String name;  
    late final int age;  
    late final double previousBMI;  
  
    final PreferredSizeWidget? appBar;  
    final Widget? body;  
    final String? userID;
```

```
DatabaseBrain({this.body, this.appBar, required this.userID});  
  
final FirebaseFirestore _firestore = FirebaseFirestore.instance;  
  
Future<String> getName() async {  
    DocumentSnapshot ds =  
        await _firestore.collection("users").doc(userID).get();  
    Map<String, dynamic> data = ds.data() as Map<String, dynamic>;  
    String name = data["name"] as String; // check if it null or not  
    return name;  
}  
  
Future<bool> checkAdmin() async {  
    DocumentSnapshot ds =  
        await _firestore.collection("users").doc(userID).get();  
    Map<String, dynamic> data = ds.data() as Map<String, dynamic>;  
    bool admin = data["admin"] as bool; // check if it null or not  
    return admin;  
}  
  
Future<int> getAge() async {  
    DocumentSnapshot ds =  
        await _firestore.collection("users").doc(userID).get();  
    Map<String, dynamic> data = ds.data() as Map<String, dynamic>;  
    Timestamp timestamp = data["dob"] as Timestamp;  
    int age = DateTime.now()  
        .difference(DateTime.parse((timestamp.toDate().toString())))  
        .inDays ~/  
        365;  
    return age;  
}  
  
Future<double> getPreviousBMI() async {  
    DocumentSnapshot ds =  
        await _firestore.collection("users").doc(userID).get();  
    Map<String, dynamic> data = ds.data() as Map<String, dynamic>;  
    double previousBMI =  
        data["previousBMI"] as double; // check if it null or not  
    return previousBMI;  
}  
  
Future<String> getGender() async {  
    DocumentSnapshot ds =  
        await _firestore.collection("users").doc(userID).get();  
    Map<String, dynamic> data = ds.data() as Map<String, dynamic>;  
    String gender = data["gender"] as String; // check if it null or not  
    print(userID);  
    print(gender);  
    return gender;  
}  
}
```

## **calculator\_brain.dart**

```
import 'package:flutter/material.dart';
import 'dart:math';

import 'constants.dart';
import 'defines.dart';

class BMICalculatorBrain {
    final int height;
    final int weight;

    late final double _bmi;

    BMICalculatorBrain({required this.height, required this.weight});

    String calculateBMI() {
        _bmi = weight / pow(height / 100, 2);
        return _bmi.toStringAsFixed(1);
    }

    Text getResult() {
        if (_bmi <= 25 && _bmi > 18.5) {
            return Text(
                'NORMAL',
                style: TextStyle(
                    fontSize: 25.0,
                    color: Colors.green,
                    fontWeight: FontWeight.bold,
                ),
            );
        } else if (_bmi > 25 && _bmi <= 30) {
            return Text(
                'OVERWEIGHT',
                style: TextStyle(
                    fontSize: 25.0,
                    color: Colors.yellow[400],
                    fontWeight: FontWeight.bold,
                ),
            );
        } else if (_bmi > 30 && _bmi <= 35) {
            return Text(
                'SLIGHTLY OBESE',
                style: TextStyle(
                    fontSize: 25.0,
                    color: Colors.orange,
                    fontWeight: FontWeight.bold,
                ),
            );
        } else if (_bmi > 35 && _bmi <= 40) {
            return Text(
                'OBESE',
                style: TextStyle(
                    fontSize: 25.0,
                    color: Colors.red,
                    fontWeight: FontWeight.bold,
                ),
            );
        }
    }
}
```

```

    );
} else if (_bmi > 40) {
    return Text(
        'DANGEROUSLY OBESE',
        style: TextStyle(
            fontSize: 25.0,
            color: Colors.red[900],
            fontWeight: FontWeight.bold,
        ),
    );
} else {
    return Text(
        'UNDERWEIGHT',
        style: TextStyle(
            fontSize: 25.0,
            color: Colors.cyan,
            fontWeight: FontWeight.bold,
        ),
    );
}
}

Widget getAdvice() {
    if (_bmi <= 25 && _bmi > 18.5) {
        return Text(
            'You are doing well. Keep it up!',
            style: kAdviceTextStyle,
        );
    } else if (_bmi > 25 && _bmi <= 30) {
        return Text(
            'You are slightly above normal weight. Maybe you could decrease the amount of fat in your diet.',
            style: kAdviceTextStyle,
            textAlign: TextAlign.center,
        );
    } else if (_bmi > 30 && _bmi <= 35) {
        return Text(
            'You are starting to become Obese. Eat less and exercise more!',
            style: kAdviceTextStyle,
            textAlign: TextAlign.center,
        );
    } else if (_bmi > 35 && _bmi <= 40) {
        return Text(
            'You have a high body fat percentage. Go on extreme diets and exercise everyday!',
            style: kAdviceTextStyle,
            textAlign: TextAlign.center,
        );
    } else if (_bmi > 40) {
        return Expanded(
            child: Container(
                child: Image.asset('assets/images/toofat.gif'),
                padding: EdgeInsets.only(bottom: 10),
            ),
        );
    } else {
        return Text(

```

```

        'You have a lower than normal body weight. Try to eat more!',  

        style: kAdviceTextStyle,  

        textAlign: TextAlign.center,  

    );  

}  

}  

}  

}  

class BodyFatCalculatorBrain {  

    Gender? selectedGender = Gender.male;  

    int age = 18;  

    int height = 170;  

    int neck = 50;  

    int weight = 60;  

    int waist = 80;  

    int hip = 90;  

    late final double _bodyFat;  

    BodyFatCalculatorBrain(  

        required this.height,  

        required this.weight,  

        required this.neck,  

        required this.waist,  

        required this.hip,  

        required this.age,  

        required this.selectedGender);  

// https://1techplus.blogspot.com/2016/07/c-code-to-calculate-body-fat-of-person.html
  

    String calculateBodyFat() {  

        if (selectedGender == Gender.male) {  

            _bodyFat = ((495) /  

                (1.0324 -  

                    0.19077 * (log(waist - neck) / ln10) +  

                    0.15456 * (log(height) / ln10))) -  

                    450;  

            return _bodyFat.toStringAsFixed(1);  

        } else {  

            _bodyFat = ((495) /  

                (1.29579 -  

                    0.35004 * (log((waist + hip) - neck) / ln10) +  

                    0.22100 * (log(height) / ln10))) -  

                    450;  

            return _bodyFat.toStringAsFixed(1);  

        }  

    }
}
  

// https://tanita.eu/blog/healthy-body-fat-percentage
Text getResult() {
    if (selectedGender == Gender.male) {
        if (_bodyFat > 0 && _bodyFat < 7) {
            return Text(
                'UNDERFAT',
                style: TextStyle(
                    fontSize: 25.0,
                    color: Colors.green,

```

```
        fontWeight: FontWeight.bold,
    ) ,
);
} else if (_bodyFat > 7 && _bodyFat <= 22) {
return Text(
'HEALTHY',
style: TextStyle(
fontSize: 25.0,
color: Colors.yellow[400],
fontWeight: FontWeight.bold,
),
);
} else if (_bodyFat > 22 && _bodyFat <= 28) {
return Text(
'OVERFAT',
style: TextStyle(
fontSize: 25.0,
color: Colors.orange,
fontWeight: FontWeight.bold,
),
);
} else {
return Text(
'OBESE',
style: TextStyle(
fontSize: 25.0,
color: Colors.red,
fontWeight: FontWeight.bold,
),
);
}
} else {
if (_bodyFat > 0 && _bodyFat < 23) {
return Text(
'UNDERFAT',
style: TextStyle(
fontSize: 25.0,
color: Colors.green,
fontWeight: FontWeight.bold,
),
);
} else if (_bodyFat > 23 && _bodyFat <= 34) {
return Text(
'HEALTHY',
style: TextStyle(
fontSize: 25.0,
color: Colors.yellow[400],
fontWeight: FontWeight.bold,
),
);
} else if (_bodyFat > 34 && _bodyFat <= 40) {
return Text(
'OVERFAT',
style: TextStyle(
fontSize: 25.0,
color: Colors.orange,
fontWeight: FontWeight.bold,
),
```

```

        ),
    );
} else {
    return Text(
        'OBESE',
        style: TextStyle(
            fontSize: 25.0,
            color: Colors.red,
            fontWeight: FontWeight.bold,
        ),
    );
}
}

Widget getAdvice() {
    if (selectedGender == Gender.male) {
        if (_bodyFat <= 0 && _bodyFat > 7) {
            return Text('Try to eat more.',
                style: kAdviceTextStyle, textAlign: TextAlign.center);
        } else if (_bodyFat > 7 && _bodyFat <= 22) {
            return Text('You are doing well. Keep it up!',
                style: kAdviceTextStyle, textAlign: TextAlign.center);
        } else if (_bodyFat > 22 && _bodyFat <= 28) {
            return Text(
                'You are starting to become Obese. Eat less and exercise more!',
                style: kAdviceTextStyle,
                textAlign: TextAlign.center);
        } else {
            return Text(
                'You have a high body fat percentage. Go on extreme diets and
exercise everyday!',
                style: kAdviceTextStyle,
                textAlign: TextAlign.center);
        }
    } else {
        if (_bodyFat <= 0 && _bodyFat > 23) {
            return Text('Try to eat more.',
                style: kAdviceTextStyle, textAlign: TextAlign.center);
        } else if (_bodyFat > 23 && _bodyFat <= 34) {
            return Text('You are doing well. Keep it up!',
                style: kAdviceTextStyle, textAlign: TextAlign.center);
        } else if (_bodyFat > 34 && _bodyFat <= 40) {
            return Text(
                'You are starting to become Obese. Eat less and exercise more!',
                style: kAdviceTextStyle,
                textAlign: TextAlign.center);
        } else {
            return Text(
                'You have a high body fat percentage. Go on extreme diets and
exercise everyday!',
                style: kAdviceTextStyle,
                textAlign: TextAlign.center);
        }
    }
}
}

```

```

class IBWCalculatorBrain {
    final int height;
    final int weight;
    final Gender? selectedGender;

    IBWCalculatorBrain(
        required this.height,
        required this.weight,
        required this.selectedGender);

    String calculateIBW() {
        if (selectedGender == Gender.male) {
            return (50 + 0.91 * (height - 152.4)).toStringAsFixed(1);
        } else {
            return (45.5 + 0.91 * (height - 152.4)).toStringAsFixed(1);
        }
    }

    Text getResult() {
        double result = weight - double.parse(calculateIBW());
        if (result > 1) {
            return Text((result.abs().toStringAsFixed(1)) + ' KG TO LOSE',
                style: kHeavyTextStyle, textAlign: TextAlign.center);
        }
        if (result < -1) {
            return Text((result.abs().toStringAsFixed(1)) + ' KG TO GAIN',
                style: kHeavyTextStyle, textAlign: TextAlign.center);
        } else {
            return Text('IDEAL WEIGHT ACHIEVED',
                style: TextStyle(
                    fontSize: 50.0, fontWeight: FontWeight.w900, color:
Colors.green),
                textAlign: TextAlign.center);
        }
    }
}

class CalorieIntakeCalculatorBrain {
    final int height;
    final int weight;
    final int age;
    final Gender? selectedGender;

    CalorieIntakeCalculatorBrain(
        required this.height,
        required this.weight,
        required this.age,
        required this.selectedGender);

    String calculateCalorieIntake() {
        if (selectedGender == Gender.male) {
            return (((66.5 + 13.8 * weight + 5 * height) - (6.8 * age)) * 1.2)
                .toStringAsFixed(1);
        } else {
            return (((655.1 + 9.6 * weight + 1.9 * height) - (4.7 * age)) * 1.2)
                .toStringAsFixed(1);
    }
}

```

```

        }

    Text getResult() {
        double result = double.parse(calculateCalorieIntake());
        return Text('DAILY CALORIE INTAKE\n ${result.toStringAsFixed(0)} 
CALORIES',
                    style: kHeavyTextStyle, textAlign: TextAlign.center);
    }
}

```

## defines.dart

```

enum Gender {
    male,
    female,
}

```

## constants.dart

```

import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';

const kBottomContainerHeight = 80.0;
const kActiveCardColor = Color(0xFF592E83);
const kInactiveCardColor = Color(0xFF33114A);
const kBottomContainerColor = Color(0xFF60AB9A);
const kActiveSliderColor = Color(0xFF60AB9A);
const kInactiveSliderColor = Color(0xFFbdb0e3);
const kCardTextStyle = TextStyle(
    fontSize: 18.0,
    color: Color(0xFFbdb0e3),
);
const kHeavyTextStyle = TextStyle(
    fontSize: 50.0,
    fontWeight: FontWeight.w900,
);
const kLargeButtonTextStyle = TextStyle(
    fontSize: 20.0,
    fontWeight: FontWeight.bold,
);
final kResultTextStyle = GoogleFontsanton(
    textStyle: TextStyle(
        color: Colors.white,
        fontSize: 40,
    ),
);
const kCalculatedBMITextStyle = TextStyle(
    fontSize: 70.0,
    fontWeight: FontWeight.w600,
);
const kAdviceTextStyle = TextStyle(
    fontSize: 25.0,
);
const kInfoTextStyle = TextStyle(

```

```

        fontSize: 20.0,
        color: Color(0xFFbdb0e3),
    );
const kLabelTextStyle = TextStyle(
    fontSize: 16.0,
    color: Color(0xFFFFFFFF),
);
const kSelectionTextStyle = TextStyle(
    fontSize: 30.0,
    color: Color(0xFFFFFFFF),
    fontWeight: FontWeight.w600,
);

final kAppTheme = ThemeData.dark().copyWith(
    appBarTheme: AppBarTheme(color: Color(0xFF230C33)),
    scaffoldBackgroundColor: Color(0xFF230C33),
    textTheme: TextTheme(bodyText2: TextStyle(color: Colors.white)),
);

final kAdminTheme = ThemeData.dark().copyWith(
    appBarTheme: AppBarTheme(color: Colors.blueGrey[800]),
    scaffoldBackgroundColor: Colors.blueGrey[800],
    textTheme: TextTheme(bodyText2: TextStyle(color: Colors.white)),
);

final kAdminTextStyle = TextStyle(
    fontSize: 20,
    fontWeight: FontWeight.bold,
    color: Colors.blueGrey[850],
);

```

## login\_page.dart

```

import 'package:bmi_calculator/pages/bmi_calculator.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:quickalert/models/quickalert_type.dart';
import 'package:quickalert/widgets/quickalert_dialog.dart';
import '../constants.dart';
import '../widgets/reusable_card.dart';
import 'package:bmi_calculator/database_brain.dart';

class LoginPage extends StatefulWidget {
    @override
    State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
    @override
    Widget build(BuildContext context) {
        final _emailController = TextEditingController();
        final _passwordController = TextEditingController();

        Future signIn() async {
            showDialog(

```

```
context: context,
barrierDismissible: false,
builder: (context) {
    return const Center(
        child: CircularProgressIndicator(),
    );
});
try {
    await FirebaseAuth.instance.signInWithEmailAndPassword(
        email: _emailController.text.trim(),
        password: _passwordController.text.trim(),
    );
    // Check the user's role and navigate to the appropriate page
    Navigator.pop(context);
} on FirebaseAuthException catch (e) {
    print(e);
    if (e.code == 'user-not-found' || e.code == 'wrong-password') {
        Navigator.pop(context);
        return QuickAlert.show(
            context: context,
            type: QuickAlertType.error,
            title: 'Incorrect Email/Password',
            text: 'Please enter your correct user details',
            backgroundColor: kInactiveCardColor,
            titleColor: Colors.white,
            textColor: Colors.white,
            confirmBtnColor: kBottomContainerColor,
            confirmBtnText: 'OK',
        );
    } else if (e.code == 'invalid-email' || e.code == 'unknown') {
        Navigator.pop(context);
        return QuickAlert.show(
            context: context,
            type: QuickAlertType.error,
            title: 'Invalid Email',
            text: 'Please enter a valid email address',
            backgroundColor: kInactiveCardColor,
            titleColor: Colors.white,
            textColor: Colors.white,
            confirmBtnColor: kBottomContainerColor,
            confirmBtnText: 'OK',
        );
    }
}
}

return MaterialApp(
    debugShowCheckedModeBanner: false,
    theme: kAppTheme,
    home: Scaffold(
        resizeToAvoidBottomInset: false,
        body: SingleChildScrollView(
            child: Padding(
                padding: const EdgeInsets.all(16.0),
                child: Center(
                    child: Column(
                        mainAxisAlignment: MainAxisAlignment.center,
```

```
crossAxisAlignment: CrossAxisAlignment.start,
children: [
    ReusableCard(
        margin: EdgeInsets.all(0),
        padding: EdgeInsets.all(5),
        color: Colors.transparent,
        cardChild: Column(
            children: [
                Container(
                    child: Image.asset(
                        'assets/images/logo_foreground.png',
                        scale: 0.5,
                    ),
                ),
                Text(
                    'Welcome back',
                    style: kResultTextStyle,
                ),
                SizedBox(
                    height: 5,
                ),
                Text(
                    'Please enter your credentials',
                ),
                SizedBox(
                    height: 30,
                ),
                Container(
                    child: TextField(
                        controller: _emailController,
                        cursorColor: Colors.white,
                        decoration: const InputDecoration(
                            focusedBorder: OutlineInputBorder(
                                borderSide: BorderSide(color: Colors.white),
                            ),
                            border: OutlineInputBorder(),
                            labelText: 'Email',
                            labelStyle: kLabelTextStyle,
                        ),
                    ),
                ),
                SizedBox(
                    height: 15,
                ),
                Container(
                    child: TextField(
                        obscureText: true,
                        controller: _passwordController,
                        cursorColor: Colors.white,
                        decoration: const InputDecoration(
                            focusedBorder: OutlineInputBorder(
                                borderSide: BorderSide(color:
Colors.white)),
                            border: OutlineInputBorder(),
                            labelText: 'Password',
                            labelStyle: kLabelTextStyle,
                        ),
                    ),
                ),
            ],
        ),
    ),
]
```

```

        ),
    ],
),
),
SizedBox(
    height: 30,
),
Padding(
    padding: const EdgeInsets.symmetric(horizontal: 1.0),
    child: GestureDetector(
        onTap: signIn,
        child: Container(
            padding: EdgeInsets.all(20),
            child: Center(
                child: Text(
                    'SIGN IN',
                    textAlign: TextAlign.center,
                    style: TextStyle(fontWeight: FontWeight.w700),
                ),
            ),
            decoration: BoxDecoration(
                color: kBottomContainerColor,
                borderRadius: BorderRadius.circular(12.0),
            ),
        ),
    ),
),
],
),
),
),
),
),
);
}
}
}
}

```

### selection\_page.dart

```

import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import '../constants.dart';
import './widgets/reusable_card.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'calorie_intake_calculator.dart';

class SelectionPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return WillPopScope(
        onWillPop: () async {
            FirebaseAuth.instance.signOut();
            return false;
        }
    );
  }
}

```

```
},
child: Scaffold(
  backgroundColor: Color(0xFF230C33),
  body: SafeArea(
    child: Column(
      children: [
        Expanded(
          child: ReusableCard(
            color: kActiveCardColor,
            cardChild: Padding(
              padding: const EdgeInsets.symmetric(horizontal: 40),
              child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                  Expanded(
                    child: Text(
                      'BMI CALCULATOR',
                      style: kSelectionTextStyle,
                      textAlign: TextAlign.center,
                    ),
                  ),
                  Icon(
                    FontAwesomeIcons.dumbbell,
                    size: 35.0,
                    color: Colors.white,
                  ),
                ],
              ),
            ),
            onPressed: () {
              Navigator.pushNamed(context, '/bmiCalculator');
            },
          ),
        ),
        Expanded(
          child: ReusableCard(
            color: kActiveCardColor,
            cardChild: Padding(
              padding: const EdgeInsets.symmetric(horizontal: 40),
              child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                  Expanded(
                    child: Text(
                      'BODY FAT',
                      style: kSelectionTextStyle,
                      textAlign: TextAlign.center,
                    ),
                  ),
                  Icon(
                    FontAwesomeIcons.percent,
                    size: 35.0,
                    color: Colors.white,
                  ),
                ],
              ),
            ),
          ),
        ),
      ],
    ),
  ),
)
```

```
        onPressed: () {
            Navigator.pushNamed(context, '/bodyFatCalculator');
        },
    ),
),
Expanded(
    child: ReusableCard(
        color: kActiveCardColor,
        cardChild: Padding(
            padding: const EdgeInsets.symmetric(horizontal: 40),
            child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                    Expanded(
                        child: Text(
                            'IDEAL WEIGHT',
                            style: kSelectionTextStyle,
                            textAlign: TextAlign.center,
                        ),
                    ),
                    Icon(
                        FontAwesomeIcons.weightScale,
                        size: 35.0,
                        color: Colors.white,
                    ),
                ],
            ),
        ),
        onPressed: () {
            Navigator.pushNamed(context, '/ibwCalculator');
        },
    ),
),
Expanded(
    child: ReusableCard(
        color: kActiveCardColor,
        cardChild: Padding(
            padding: const EdgeInsets.symmetric(horizontal: 40),
            child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                children: [
                    Expanded(
                        child: Text(
                            'CALORIE INTAKE',
                            style: kSelectionTextStyle,
                            textAlign: TextAlign.center,
                        ),
                    ),
                    Icon(
                        FontAwesomeIcons.pizzaSlice,
                        size: 35.0,
                        color: Colors.white,
                    ),
                ],
            ),
        ),
        onPressed: ()
```

```

        {
            Navigator.pushNamed(context, '/CalorieIntakeCalculator');
        },
    ),
),
],
),
),
),
);
}
}

```

## bmi\_calculator.dart

```

import 'package:bmi_calculator/database_brain.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import '../widgets/reusable_card.dart';
import '../constants.dart';
import '../calculator_brain.dart';
import 'results_page.dart';

class BMICalculator extends StatefulWidget {
    @override
    _BMICalculatorState createState() => _BMICalculatorState();
}

class _BMICalculatorState extends State<BMICalculator> {
    DatabaseBrain databaseBrain =
        DatabaseBrain(userID:
    FirebaseAuth.instance.currentUser?.uid.toString());
    int height = 150;
    int weight = 60;
    String bmi = '25';

    @override
    Widget build(BuildContext context) {
        return WillPopScope(
            onWillPop: () async {
                return true;
            },
            child: Scaffold(
                appBar: AppBar(
                    toolbarHeight: 60,
                    title: Text('BMI CALCULATOR'),
                    centerTitle: true,
                ),
                body: Column(
                    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                    children: [
                        Expanded(
                            child: ReusableCard(
                                color: kActiveCardColor,
                                cardChild: Column(
                                    mainAxisAlignment: MainAxisAlignment.center,

```

```
        children: [
          Text(
            'HEIGHT',
            style: kCardTextStyle,
          ),
          Row(
            mainAxisAlignment: MainAxisAlignment.center,
            crossAxisAlignment: CrossAxisAlignment.baseline,
            textBaseline: TextBaseline.alphabetic,
            children: [
              Text(
                height.toString(),
                style: kHeavyTextStyle,
              ),
              Text(
                'cm',
                style: kCardTextStyle,
              )
            ],
          ),
          SliderTheme(
            data: SliderTheme.of(context).copyWith(
              trackHeight: 1.5,
              thumbShape:
                  RoundSliderThumbShape(enabledThumbRadius: 10.0),
              overlayShape:
                  RoundSliderOverlayShape(overlayRadius: 30.0),
            ),
            child: Slider(
              value: height.toDouble(),
              min: 60.0,
              max: 250.0,
              activeColor: kActiveSliderColor,
              inactiveColor: kInactiveSliderColor,
              onChanged: (double newValue) {
                setState(() {
                  height = newValue.round();
                });
              },
            ),
          ),
        ],
      ),
    ),
  ),
  Expanded(
    child: ReusableCard(
      color: kActiveCardColor,
      cardChild: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text(
            'WEIGHT',
            style: kCardTextStyle,
          ),
          Row(
            mainAxisAlignment: MainAxisAlignment.center,
```



```

        ),
    );
}

class BottomButton extends StatelessWidget {
    final Function()? onTap;
    final String? text;

    BottomButton({required this.onTap, required this.text});

    @override
    Widget build(BuildContext context) {
        return GestureDetector(
            onTap: onTap,
            child: Container(
                child: Center(
                    child: Text(
                        text!,
                        style: kLargeButtonTextStyle,
                    ),
                ),
                padding: EdgeInsets.only(bottom: 5.0),
                margin: EdgeInsets.only(top: 10),
                width: double.infinity,
                height: kBottomContainerHeight,
                decoration: BoxDecoration(
                    color: kBottomContainerColor,
                    borderRadius: BorderRadius.circular(2.5),
                ),
            ),
        );
    }
}

```

## body\_fat\_calculator.dart

```

import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
import '../widgets/reusable_card.dart';
import '../constants.dart';
import '../widgets/rounded_button.dart';
import '../calculator_brain.dart';
import 'results_page.dart';
import '../defines.dart';
import 'package:bmi_calculator/database_brain.dart';

class BodyFatCalculator extends StatefulWidget {
    @override
    _BodyFatCalculatorState createState() => _BodyFatCalculatorState();
}

class _BodyFatCalculatorState extends State<BodyFatCalculator> {
    DatabaseBrain databaseBrain =
        DatabaseBrain(userID:

```

```
FirebaseAuth.instance.currentUser?.uid.toString());
```

```
@override
void initState() {
    databaseBrain.getGender().then((value) {
        if (value == 'male') {
            selectedGender = Gender.male;
        } else if (value == 'female') {
            selectedGender = Gender.female;
        }
    });
    databaseBrain.getAge().then((value) {
        age = value;
    });
    super.initState();
}
```

```
Gender? selectedGender;
int age = 18;
int height = 170;
int neck = 50;
int weight = 60;
int waist = 90;
int hip = 110;
int bodyFat = 15;
```

```
@override
Widget build(BuildContext context) {
    return WillPopScope(
        onWillPop: () async {
            return true;
        },
        child: Scaffold(
            appBar: AppBar(
                toolbarHeight: 60,
                title: Text('Body Fat Calculator'),
                centerTitle: true,
            ),
            body: Column(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: [
                    Expanded(
                        child: Row(
                            children: [
                                Expanded(
                                    child: ReusableCard(
                                        color: kActiveCardColor,
                                        cardChild: Column(
                                            mainAxisAlignment: MainAxisAlignment.center,
                                            children: [
                                                Text(
                                                    "HEIGHT",
                                                    style: kCardTextStyle,
                                                ),
                                                Text(
                                                    height.toString(),
                                                    style: kHeavyTextStyle,
                                                ),
                                            ],
                                        ),
                                    ),
                                ),
                            ],
                        ),
                    ),
                ],
            ),
        ),
    );
}
```

```
) ,
Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
        RoundIconButton(
            IconData: FontAwesomeIcons.minus,
            onPressed: () {
                setState(() {
                    height--;
                });
            }
        ),
        SizedBox(width: 10),
        RoundIconButton(
            IconData: FontAwesomeIcons.plus,
            onPressed: () {
                setState(() {
                    height++;
                });
            }
        ),
    ],
),
),
),
),
),
),
),
),
),
Expanded(
    child: ReusableCard(
        color: kActiveCardColor,
        cardChild: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
                Text(
                    "WEIGHT",
                    style: kCardTextStyle,
                ),
                Text(
                    weight.toString(),
                    style: kHeavyTextStyle,
                ),
            ],
            mainAxisSize: MainAxisSize.center,
            Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                    RoundIconButton(
                        IconData: FontAwesomeIcons.minus,
                        onPressed: () {
                            setState(() {
                                weight--;
                            });
                        }
                    ),
                    SizedBox(width: 10),
                    RoundIconButton(
                        IconData: FontAwesomeIcons.plus,
                        onPressed: () {
                            setState(() {
                                weight++;
                            });
                        }
                    ),
                ],
            ),
        ],
    ),
),
```

```
        } ) ;
    ] ,
) ,
) ,
) ,
)
] ,
),
)
Expanded(
child: Row(
children: [
Expanded(
child: ReusableCard(
color: kActiveCardColor,
cardChild: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: [
Text(
"NECK",
style: kCardTextStyle,
),
Text(
neck.toString(),
style: kHeavyTextStyle,
),
Row(
mainAxisAlignment: MainAxisAlignment.center,
children: [
RoundIconButton(
iconData: FontAwesomeIcons.minus,
onPressed: () {
setState(() {
neck--;
}) ;
}),
SizedBox(width: 10),
RoundIconButton(
iconData: FontAwesomeIcons.plus,
onPressed: () {
setState(() {
neck++;
}) ;
}),
],
),
),
),
),
Expanded(
child: ReusableCard(
color: kActiveCardColor,
cardChild: Column(
```

```
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
            Text(
                "WAIST",
                style: kCardTextStyle,
            ),
            Text(
                waist.toString(),
                style: kHeavyTextStyle,
            ),
            Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                    RoundIconButton(
                        iconData: FontAwesomeIcons.minus,
                        onPressed: () {
                            setState(() {
                                waist--;
                            });
                        },
                    ),
                    SizedBox(width: 10),
                    RoundIconButton(
                        iconData: FontAwesomeIcons.plus,
                        onPressed: () {
                            setState(() {
                                waist++;
                            });
                        },
                    ),
                ],
            ),
            ],
        ),
    ),
),
Expanded(
    child: ReusableCard(
        color: kActiveCardColor,
        cardChild: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
                Text(
                    "HIP",
                    style: kCardTextStyle,
                ),
                Text(
                    hip.toString(),
                    style: kHeavyTextStyle,
                ),
            ],
        ),
        mainAxisSize: MainAxisSize.center,
        children: [
            RoundIconButton(
                iconData: FontAwesomeIcons.minus,
```



```

@Override
Widget build(BuildContext context) {
    return GestureDetector(
        onTap: onTap,
        child: Container(
            child: Center(
                child: Text(
                    text!,
                    style: kLargeButtonTextStyle,
                ),
            ),
        ),
        padding: EdgeInsets.only(bottom: 5.0),
        margin: EdgeInsets.only(top: 10),
        width: double.infinity,
        height: kBottomContainerHeight,
        decoration: BoxDecoration(
            color: kBottomContainerColor,
            borderRadius: BorderRadius.circular(2.5),
        ),
    ),
);
}
}

```

## ibw\_calculator.dart

```

import 'package:bmi_calculator/database_brain.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import '../widgets/reusable_card.dart';
import '../constants.dart';
import '../calculator_brain.dart';
import 'results_page.dart';
import 'package:bmi_calculator/defines.dart';

class IBWCalculator extends StatefulWidget {
    @override
    _IBWCalculatorState createState() => _IBWCalculatorState();
}

class _IBWCalculatorState extends State<IBWCalculator> {
    DatabaseBrain databaseBrain =
        DatabaseBrain(userID:
    FirebaseAuth.instance.currentUser?.uid.toString());
    Gender selectedGender = Gender.male;

    @override
    void initState() {
        databaseBrain.getGender().then((value) {
            if (value == 'male') {
                selectedGender = Gender.male;
            } else if (value == 'female') {
                selectedGender = Gender.female;
            }
        });
        super.initState();
    }
}

```

```
}

Gender? selectedGender;
int height = 150;
int weight = 60;
int age = 20;
String? gender;

@Override
Widget build(BuildContext context) {
    return WillPopScope(
        onWillPop: () async {
            return true;
        },
        child: Scaffold(
            appBar: AppBar(
                toolbarHeight: 60,
                title: Text('IDEAL WEIGHT'),
                centerTitle: true,
            ),
            body: Column(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: [
                    Expanded(
                        child: ReusableCard(
                            color: kActiveCardColor,
                            cardChild: Column(
                                mainAxisAlignment: MainAxisAlignment.center,
                                children: [
                                    Text(
                                        'HEIGHT',
                                        style: kCardTextStyle,
                                    ),
                                    Row(
                                        mainAxisAlignment: MainAxisAlignment.center,
                                        crossAxisAlignment: CrossAxisAlignment.baseline,
                                        textBaseline: TextBaseline.alphabetic,
                                        children: [
                                            Text(
                                                height.toString(),
                                                style: kHeavyTextStyle,
                                            ),
                                            Text(
                                                'cm',
                                                style: kCardTextStyle,
                                            )
                                        ],
                                    ),
                                ],
                            ),
                            sliderTheme: SliderTheme(
                                data: SliderTheme.of(context).copyWith(
                                    trackHeight: 1.5,
                                    thumbShape:
                                        RoundSliderThumbShape(enabledThumbRadius: 10.0),
                                    overlayShape:
                                        RoundSliderOverlayShape(overlayRadius: 30.0),
                                ),
                                child: Slider(

```





```

        borderRadius: BorderRadius.circular(2.5),
    ),
),
);
}
}

```

## calorie\_intake\_calculator.dart

```

import 'package:bmi_calculator/database_brain.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import '../widgets/reusable_card.dart';
import '../constants.dart';
import '../calculator_brain.dart';
import 'results_page.dart';
import 'package:bmi_calculator/defines.dart';

class CalorieIntakeCalculator extends StatefulWidget {
    @override
    _CalorieIntakeCalculator createState() => _CalorieIntakeCalculator();
}

class _CalorieIntakeCalculator extends State<CalorieIntakeCalculator> {
    DatabaseBrain databaseBrain =
        DatabaseBrain(userID:
    FirebaseAuth.instance.currentUser?.uid.toString());

    @override
    void initState() {
        databaseBrain.getGender().then((value) {
            if (value == 'male') {
                setState(() {
                    selectedGender = Gender.male;
                });
            } else if (value == 'female') {
                setState(() {
                    selectedGender = Gender.female;
                });
            }
            print(selectedGender);
        });
        databaseBrain.getAge().then((value) {
            age = value;
        });
        super.initState();
    }

    Gender? selectedGender;
    int height = 150;
    int weight = 60;
    int age = 20;
    String? gender;

    @override
    Widget build(BuildContext context) {

```

```
return WillPopScope(
  onWillPop: () async {
    return true;
},
child: Scaffold(
  appBar: AppBar(
    toolbarHeight: 60,
    title: Text('CALORIE INTAKE'),
    centerTitle: true,
),
body: Column(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    Expanded(
      child: ReusableCard(
        color: kActiveCardColor,
        cardChild: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Text(
              'HEIGHT',
              style: kCardTextStyle,
            ),
            Row(
              mainAxisAlignment: MainAxisAlignment.center,
              crossAxisAlignment: CrossAxisAlignment.baseline,
              textBaseline: TextBaseline.alphabetic,
              children: [
                Text(
                  height.toString(),
                  style: kHeavyTextStyle,
                ),
                Text(
                  'cm',
                  style: kCardTextStyle,
                )
              ],
            )
          ],
        ),
SliderTheme(
  data: SliderTheme.of(context).copyWith(
    trackHeight: 1.5,
    thumbShape:
      RoundSliderThumbShape(enabledThumbRadius: 10.0),
    overlayShape:
      RoundSliderOverlayShape(overlayRadius: 30.0),
),
child: Slider(
  value: height.toDouble(),
  min: 60.0,
  max: 250.0,
  activeColor: kActiveSliderColor,
  inactiveColor: kInactiveSliderColor,
  onChanged: (double newValue) {
    setState(() {
      height = newValue.round();
    });
}
),

```

```
        ),
        ],
        ),
        ),
        ),
        Expanded(
        child: ReusableCard(
        color: kActiveCardColor,
        cardChild: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
        Text(
        'WEIGHT',
        style: kCardTextStyle,
        ),
        Row(
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.baseline,
        textBaseline: TextBaseline.alphabetic,
        children: [
        Text(
        weight.toString(),
        style: kHeavyTextStyle,
        ),
        Text(
        'KG',
        style: kCardTextStyle,
        )
        ],
        ),
        SliderTheme(
        data: SliderTheme.of(context).copyWith(
        trackHeight: 1.5,
        thumbShape:
        RoundSliderThumbShape(enabledThumbRadius: 10.0),
        overlayShape:
        RoundSliderOverlayShape(overlayRadius: 30.0),
        ),
        child: Slider(
        value: weight.toDouble(),
        min: 30.0,
        max: 200.0,
        activeColor: kActiveSliderColor,
        inactiveColor: kInactiveSliderColor,
        onChanged: (double newValue) {
        setState(() {
        weight = newValue.round();
        });
        },
        ),
        ),
        ],
        ),
        ),
        ),
        BottomButton(
```

```
        onTap: () {
            CalorieIntakeCalculatorBrain calculatorBrain =
                CalorieIntakeCalculatorBrain(
                    height: height,
                    weight: weight,
                    age: age,
                    selectedGender: selectedGender);
            print(selectedGender);
            Navigator.push(
                context,
                MaterialPageRoute(
                    builder: (context) => CalorieIntakeResultsPage(
                        resultText: calculatorBrain.getResult(),
                    ),
                ),
            );
        },
        text: 'CALCULATE',
    )
),
),
),
);
}
}

class BottomButton extends StatelessWidget {
final Function()? onTap;
final String? text;

BottomButton({required this.onTap, required this.text});

@Override
Widget build(BuildContext context) {
    return GestureDetector(
        onTap: onTap,
        child: Container(
            child: Center(
                child: Text(
                    text!,
                    style: kLargeButtonTextStyle,
                ),
            ),
            padding: EdgeInsets.only(bottom: 5.0),
            margin: EdgeInsets.only(top: 10),
            width: double.infinity,
            height: kBottomContainerHeight,
            decoration: BoxDecoration(
                color: kBottomContainerColor,
                borderRadius: BorderRadius.circular(2.5),
            ),
        ),
    );
}
}
```

## results\_page.dart

```
import 'package:bmi_calculator/pages/offline_mode.dart';
import 'package:flutter/material.dart';
import '../constants.dart';
import '../widgets/reusable_card.dart';

class BMIResultsPage extends StatelessWidget {
    final String bmi;
    final Text resultText;
    final Widget advice;

    BMIResultsPage(
        required this.bmi, required this.resultText, required this.advice);

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                centerTitle: true,
                title: Text('BMI CALCULATOR'),
            ),
            body: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                crossAxisAlignment: CrossAxisAlignment.stretch,
                children: [
                    Expanded(
                        child: Center(
                            child: Container(
                                padding: EdgeInsets.only(top: 30.0),
                                child: Text(
                                    'Your Result',
                                    style: kResultTextStyle,
                                ),
                            ),
                        ),
                    ),
                    Expanded(
                        flex: 5,
                        child: ReusableCard(
                            color: kActiveCardColor,
                            cardChild: Column(
                                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                                crossAxisAlignment: CrossAxisAlignment.center,
                                children: [
                                    Container(
                                        child: resultText,
                                    ),
                                    Container(
                                        child: Text(
                                            bmi,
                                            style: kCalculatedBMITextStyle,
                                        ),
                                    ),
                                    Container(
                                        child: advice,
                                    ),
                                ],
                            ),
                        ),
                    ),
                ],
            ),
        );
    }
}
```

```
        ],
    ),
),
),
),
BottomButton(
    onTap: () {
        Navigator.pop(context);
    },
    text: 'CALCULATE AGAIN'
),
),
);
}
}

class BodyFatResultsPage extends StatelessWidget {
    final String bodyFat;
    final Text resultText;
    final Widget advice;

    BodyFatResultsPage(
        required this.bodyFat, required this.resultText, required
this.advice);

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                centerTitle: true,
                title: Text('BODY FAT CALCULATOR'),
            ),
            body: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                crossAxisAlignment: CrossAxisAlignment.stretch,
                children: [
                    Expanded(
                        child: Center(
                            child: Container(
                                padding: EdgeInsets.only(top: 30.0),
                                child: Text(
                                    'Your Result',
                                    style: kResultTextStyle,
                                ),
                            ),
                        ),
                    ),
                    Expanded(
                        flex: 5,
                        child: ReusableCard(
                            color: kActiveCardColor,
                            cardChild: Column(
                                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                                crossAxisAlignment: CrossAxisAlignment.center,
                                children: [
                                    Container(
                                        child: resultText,
                                    ),
                                ],
                            ),
                        ),
                    ),
                ],
            ),
        );
    }
}
```

```

        Container(
            child: Text(
                bodyFat,
                style: kCalculatedBMITextStyle,
            ) ,
        ) ,
        Container(
            child: advice,
        ) ,
    ] ,
) ,
),
BottomButton(
    onTap: () {
        Navigator.pop(context);
    },
    text: 'CALCULATE AGAIN'
),
),
);
}
}

class IBWResultsPage extends StatelessWidget {
final String ibw;
final Text resultText;

IBWResultsPage({required this.ibw, required this.resultText});

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            centerTitle: true,
            title: Text('IDEAL WEIGHT'),
        ),
        body: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: [
                Expanded(
                    child: Center(
                        child: Container(
                            padding: EdgeInsets.only(top: 30.0),
                            child: Text(
                                'Your Result',
                                style: kResultTextStyle,
                            ) ,
                        ) ,
                    ) ,
                ) ,
                Expanded(
                    flex: 5,
                    child: ReusableCard(
                        color: kActiveCardColor,
                        cardChild: Column(

```

```
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
            Container(
                child: Text(
                    ibw + ' KG',
                    style: kCalculatedBMITextStyle,
                ),
            ),
            Container(
                child: Center(child: resultText),
            ),
        ],
    ),
),
),
),
),
),
),
BottomButton(
    onTap: () {
        Navigator.pop(context);
    },
    text: 'CALCULATE AGAIN'
),
),
),
),
),
),
),
),
},
),

class CalorieIntakeResultsPage extends StatelessWidget {
final Text resultText;

CalorieIntakeResultsPage({required this.resultText});

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            centerTitle: true,
            title: Text('IDEAL WEIGHT'),
        ),
        body: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: [
                Expanded(
                    child: Center(
                        child: Container(
                            padding: EdgeInsets.only(top: 30.0),
                            child: Text(
                                'Your Result',
                                style: kResultTextStyle,
                            ),
                        ),
                    ),
                ),
                Expanded(
                    flex: 5,
                    child: ReusableCard(

```

```
        color: kActiveCardColor,
        cardChild: Column(
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            crossAxisAlignment: CrossAxisAlignment.center,
            children: [
                Container(
                    child: Center(child: resultText),
                ),
            ],
        ),
    ),
),
BottomButton(
    onTap: () {
        Navigator.pop(context);
    },
    text: 'CALCULATE AGAIN'
],
);
}
}
```

admin page.dart

```

        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) => AddUserPage(),
            ),
          );
        },
      cardChild: Center(
        child: Text(
          'ADD USER',
          style: kSelectionTextStyle,
        ),
      ),
    ),
  ),
),
Expanded(
  child: ReusableCard(
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
        builder: (context) => DeleteUserPage(),
      ),
    );
  },
  color: Colors.blueGrey,
  cardChild: Center(
    child: Text(
      'DELETE USER',
      style: kSelectionTextStyle,
    ),
  ),
),
],
),
),
),
));
}
}

```

## add\_user\_page.dart

```

import 'package:bmi_calculator/widgets/reusable_card.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

import '../constants.dart';
import 'package:intl/intl.dart';

class AddUserPage extends StatefulWidget {
  @override
  _AddUserPageState createState() => _AddUserPageState();
}

```

```
}
```

```
class _AddUserPageState extends State<AddUserPage> {
  final _formKey = GlobalKey<FormState>();

  // Add text controllers for each of the form fields
  final _nameController = TextEditingController();
  final _emailController = TextEditingController();
  final _passwordController = TextEditingController();
  final _dobController = TextEditingController();

  String? _selectedGender;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blueGrey[700],
      appBar: AppBar(
        title: Text('Create New User'),
        backgroundColor: Colors.blueGrey[700],
        centerTitle: true,
      ),
      body: Form(
        key: _formKey,
        child: ReusableCard(
          color: Colors.blueGrey,
          margin: EdgeInsets.all(8),
          padding: EdgeInsets.all(20),
          cardChild: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              TextFormField(
                controller: _nameController,
                decoration: InputDecoration(labelText: 'Name'),
                validator: (value) {
                  if (value?.isEmpty ?? true) {
                    return 'Please enter a name';
                  }
                  return null;
                },
              ),
              TextFormField(
                controller: _emailController,
                decoration: InputDecoration(labelText: 'Email'),
                validator: (value) {
                  if (value?.isEmpty ?? true) {
                    return 'Please enter an email';
                  }
                  return null;
                },
              ),
              TextFormField(
                controller: _passwordController,
                decoration: InputDecoration(labelText: 'Password'),
                validator: (value) {
                  if (value?.isEmpty ?? true) {
                    return 'Please enter a password';
                  }
                  return null;
                },
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

```

        }
        return null;
    },
),
TextField(
    controller: _dobController,
    decoration:
        InputDecoration(labelText: 'Date of Birth (YYYY-MM-DD)'),
    validator: (value) {
        if (value?.isEmpty ?? true) {
            return 'Please enter a date of birth';
        }
        return null;
    },
),
DropdownButtonFormField(
    value: _selectedGender,
    items: [
        DropdownMenuItem(
            child: Text('Male'),
            value: 'male',
        ),
        DropdownMenuItem(
            child: Text('Female'),
            value: 'female',
        ),
    ],
    onChanged: (value) {
        setState(() {
            _selectedGender = value as String?;
        });
    },
    decoration: InputDecoration(labelText: 'Gender'),
),
ReusableCard(
    color: Color(0xFFB0BEC5),
    padding: EdgeInsets.all(20),
    onPressed: () async {
        if (_formKey.currentState!.validate()) {
            // Parse the date of birth string into a DateTime object
            var dob =
                DateFormat("yyyy-MM-dd").parse(_dobController.text);

            // Create a new user with FirebaseAuth
            final FirebaseAuth _auth = FirebaseAuth.instance;
            final authResult =
                await _auth.createUserWithEmailAndPassword(
                    email: _emailController.text,
                    password: _passwordController.text,
                );

            // Get the user's UID
            final uid = authResult.user?.uid;

            // Add a new document with the user's UID as the document
            await FirebaseFirestore.instance

```

ID

```

        .collection('users')
        .doc(uid)
        .set({
            'name': _nameController.text,
            'email': _emailController.text,
            'password': _passwordController.text,
            'admin': false,
            'dob': Timestamp.fromDate(dob),
            'gender': _selectedGender,
        });
        Navigator.pop(context);
    }
},
cardChild: Text(
    'Create User',
    style: kAdminTextStyle,
),
),
),
],
),
),
),
);
}
}

```

## delete\_user\_page.dart

```

import 'package:bmi_calculator/constants.dart';
import 'package:bmi_calculator/widgets/reusable_card.dart';
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

class DeleteUserPage extends StatefulWidget {
    @override
    _DeleteUserPageState createState() => _DeleteUserPageState();
}

class _DeleteUserPageState extends State<DeleteUserPage> {
    final _formKey = GlobalKey<FormState>();

    // Add a text controller for the email field
    final _emailController = TextEditingController();

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            backgroundColor: Colors.blueGrey[700],
            appBar: AppBar(
                title: Text('Delete User'),
                centerTitle: true,
                backgroundColor: Colors.blueGrey[700],
            ),
            body: Form(
                key: _formKey,
                child: ReusableCard(

```

```

        color: Colors.blueGrey,
        margin: EdgeInsets.all(8),
        padding: EdgeInsets.all(20),
        cardChild: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
                TextFormField(
                    controller: _emailController,
                    decoration: InputDecoration(labelText: 'Email'),
                    validator: (value) {
                        if (value?.isEmpty ?? true) {
                            return 'Please enter an email';
                        }
                        return null;
                    },
                ),
                ReusableCard(
                    color: Color(0xFFB0BEC5),
                    padding: EdgeInsets.all(20),
                    onPressed: () async {
                        if (_formKey.currentState!.validate()) {
                            // Query the users collection for a document with the
                            matching email
                            QuerySnapshot snapshot = await
                            FirebaseFirestore.instance.collection('users').where('email', isEqualTo:
                            _emailController.text).get();
                            if (snapshot.docs.length == 0) {
                                // If no documents are found, display an error message
                                ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text('No user
                                was found with the specified email.')));
                            } else {
                                // If a document is found, delete it
                                DocumentSnapshot user = snapshot.docs[0];
                                await user.reference.delete();

                                ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text('The user
                                was deleted successfully.')));
                            }
                        }
                    },
                    cardChild: Text('Delete User',
                        style: kAdminTextStyle,
                    ),
                ),
            ],
        );
    }
}

```

### **card\_content.dart**

```
import 'package:flutter/material.dart';
import '../constants.dart';

class CardContent extends StatelessWidget {
  IconData cardIcon;
  String cardText;

  CardContent({required this.cardIcon, required this.cardText});

  @override
  Widget build(BuildContext context) {
    return Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Icon(
          cardIcon,
          size: 80.0,
        ),
        SizedBox(
          height: 15.0,
        ),
        Text(
          cardText,
          style: kCardTextStyle,
        )
      ],
    );
  }
}
```

### **reusable\_card.dart**

```
import 'package:flutter/material.dart';

class ReusableCard extends StatelessWidget {
  final Color color;
  final Widget? cardChild;
  final Function()? onPress;
  final EdgeInsetsGeometry? margin;
  final EdgeInsetsGeometry? padding;

  ReusableCard({required this.color, this.cardChild, this.onPress,
  this.margin = const EdgeInsets.all(15.0), this.padding = const
  EdgeInsets.all(0)});

  @override
  Widget build(BuildContext context) {
    return GestureDetector(
      onTap: onPress,
      child: Container(
        margin: margin,
        padding: padding,
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(10),

```

```
        color: color,
    ),
    child: cardChild,
),
);
}
}
```

### rounded\_button.dart

```
import 'package:flutter/material.dart';
import '../constants.dart';

class RoundIconButton extends StatelessWidget {
    final IconData iconData;
    final Function() onPressed;

    RoundIconButton({required this.iconData, required this.onPressed});

    @override
    Widget build(BuildContext context) {
        return RawMaterialButton(
            child: Icon(iconData),
            onPressed: this.onPressed,
            shape: CircleBorder(),
            fillColor: kInactiveSliderColor,
            constraints: BoxConstraints.tightFor(
                width: 56.0,
                height: 56.0,
            ),
        );
    }
}
```

### pubspec.yaml

```
name: bmi_calculator
description: A new Flutter application.

version: 1.0.0+1

environment:
  sdk: ">=2.12.0 <3.0.0"

dependencies:
  flutter:
    sdk: flutter

  cupertino_icons: ^0.1.2
  font_awesome_flutter: ^10.3.0
  flutter_login: ^4.1.1
  flutter_gif: ^0.0.4
  firebase_core: ^2.4.0
  firebase_auth: ^4.2.3
  quickalert: ^1.0.1
```

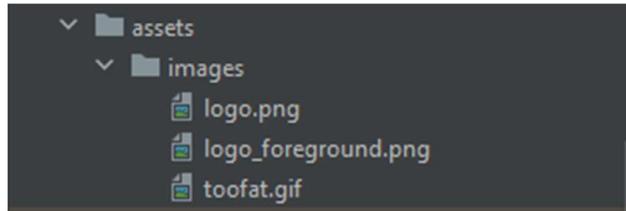
```
google_fonts: ^3.0.1
cloud_firestore: ^4.3.1
timeago: ^3.3.0
age_calculator: ^1.0.0
flutter_launcher_icons: ^0.11.0

flutter_icons:
  android: true
  ios: true
  image_path: "assets/images/logo.png"
  adaptive_icon_background: "#FFFF00"
  adaptive_icon_foreground: "assets/images/logo_foreground.png"

dev_dependencies:
  flutter_test:
    sdk: flutter

flutter:
  assets:
    - assets/images/
  uses-material-design: true
```

**assets folder:**

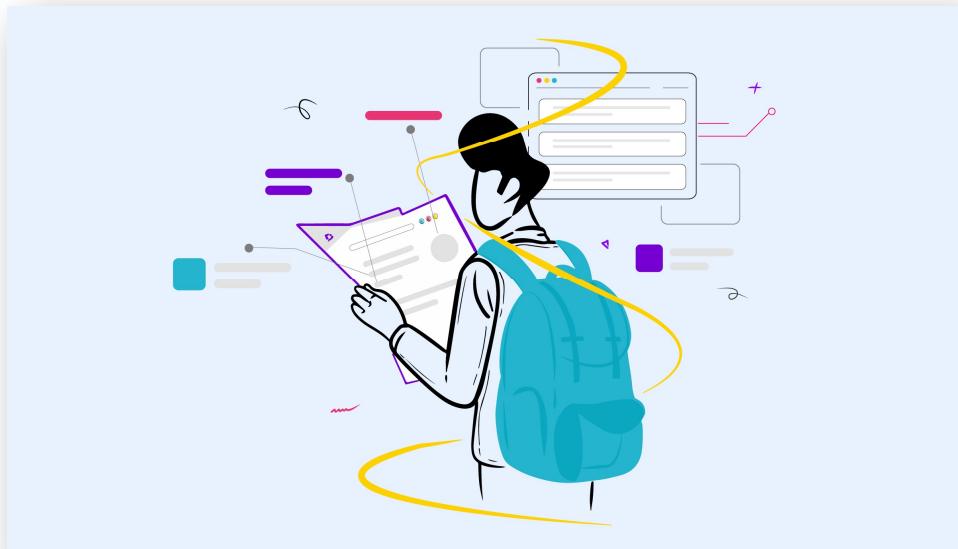


# **BODYMATH APP**

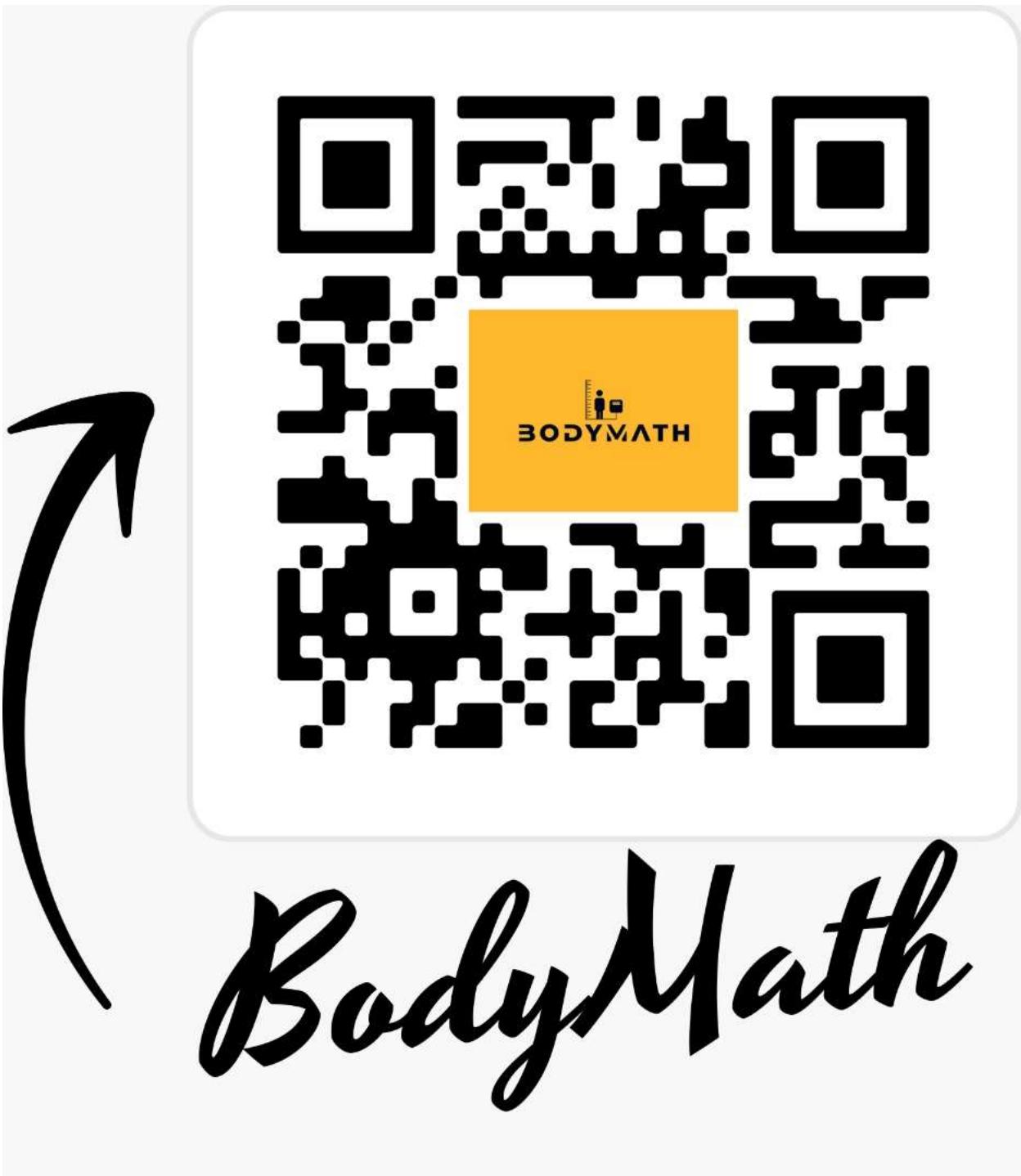
## **BY SOG CO.**



## **RC9: Software User Manual**

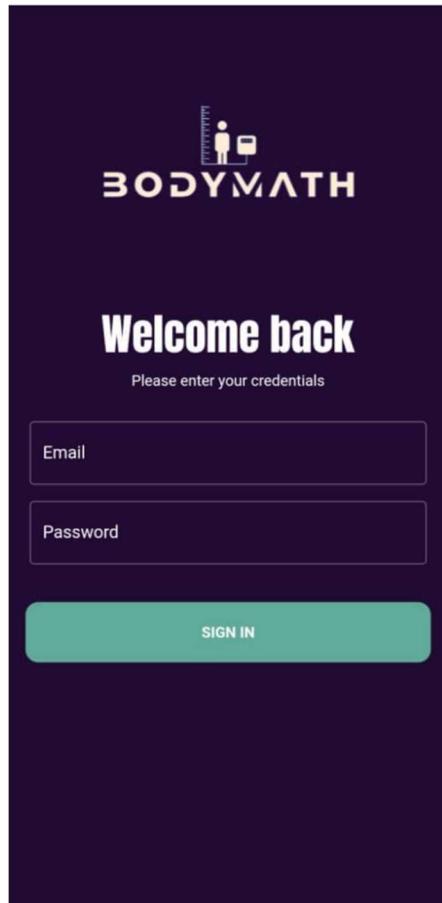


Scan QR to Download App



**Member Login:**

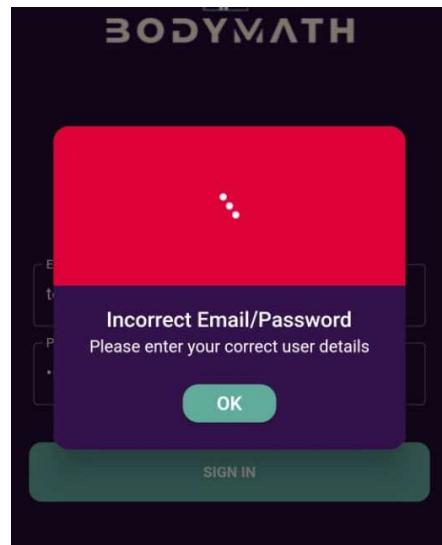
**Step1:** Enter your email and password.



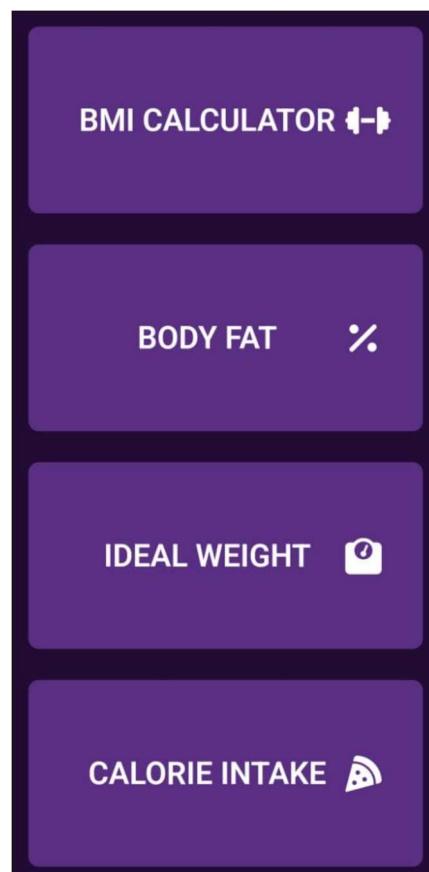
Step2: Press “SIGN IN” button.



In the case of incorrect user details, pop up error message will appear.

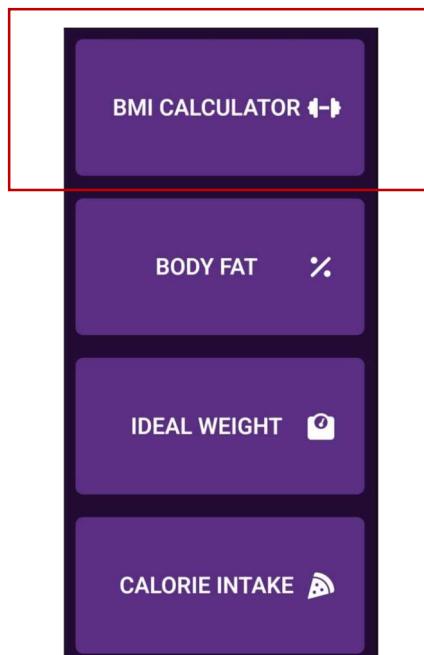


If the user details are correct, the application main page will appear.

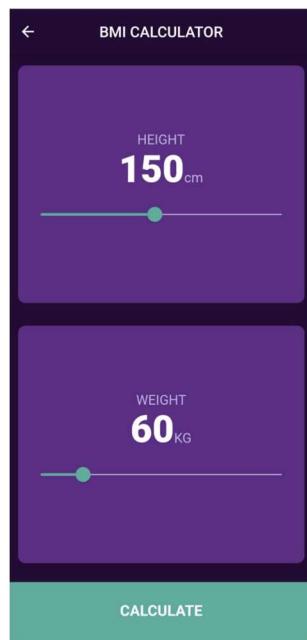


## BMI Calculator:

**Step 1:** Press on “BMI CALCULATOR” button.

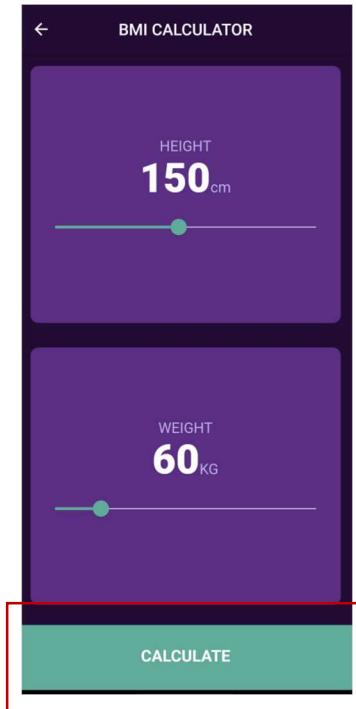


BMI calculator page will appear.

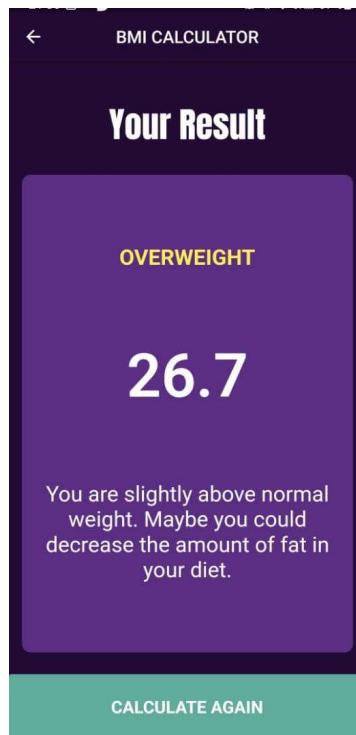


**Step 2:** Set your height (in cm) and weight (in kg).

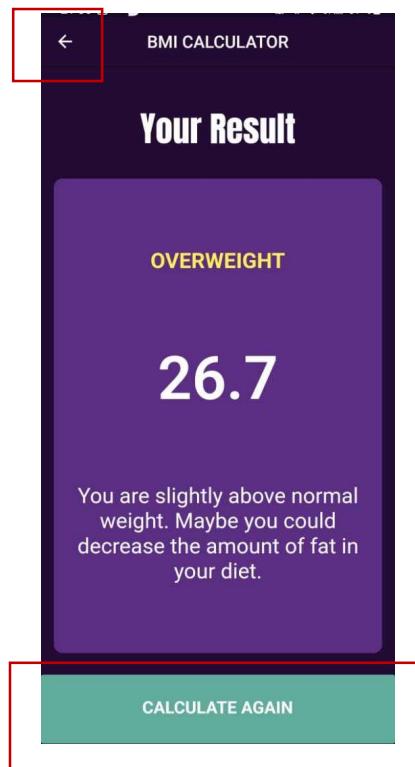
**Step 3:** Press “CALCULATE” button.



**Step 4:** Result page will appear.

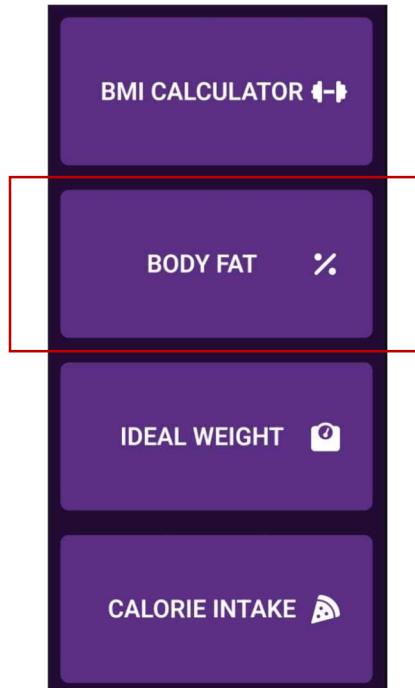


Step 5: You can press “CALCULATE AGAIN” button, or back button to go back to BMI calculator page.



## Body Fat Calculator:

**Step 1:** Press on “BODY FAT” button.

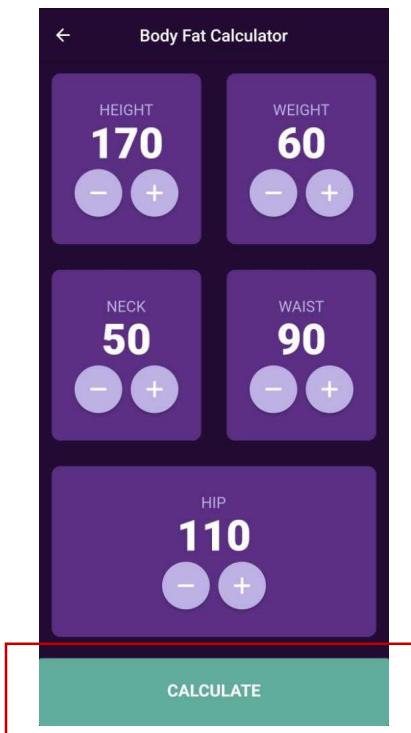


Body Fat calculator page will appear.

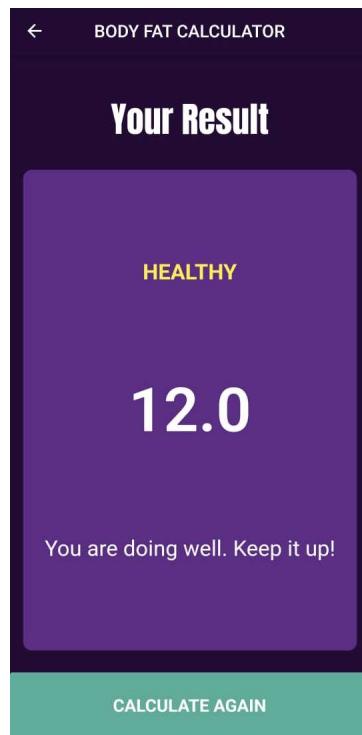


**Step 2:** Set your height (in cm), weight (in kg), neck (in cm), waist (in cm), hip (in cm) values.

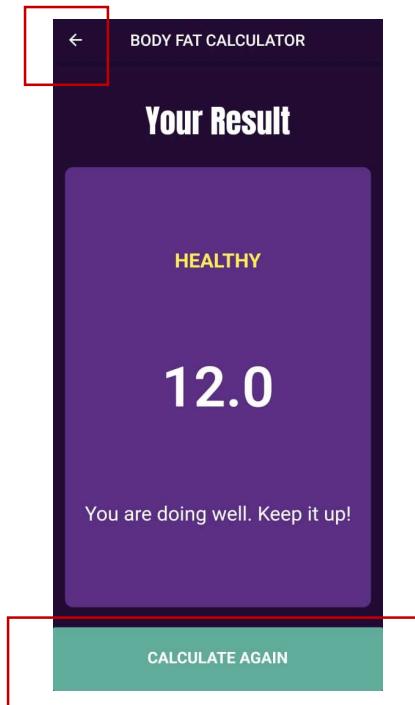
**Step 3:** Press “CALCULATE” button.



**Step 4:** Result page will appear.

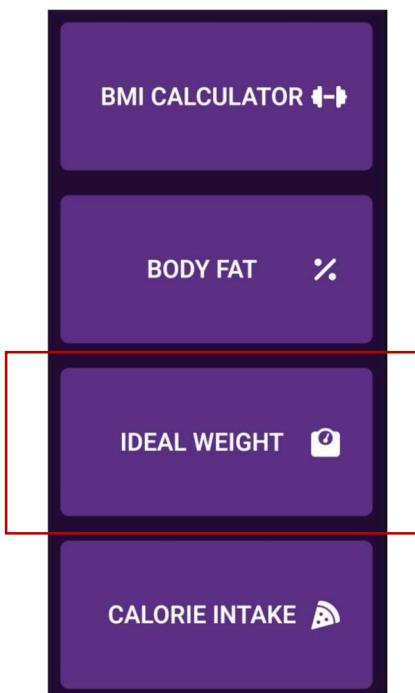


Step 5: You can press “CALCULATE AGAIN” button, or back button to go back to Body Fat calculator page.

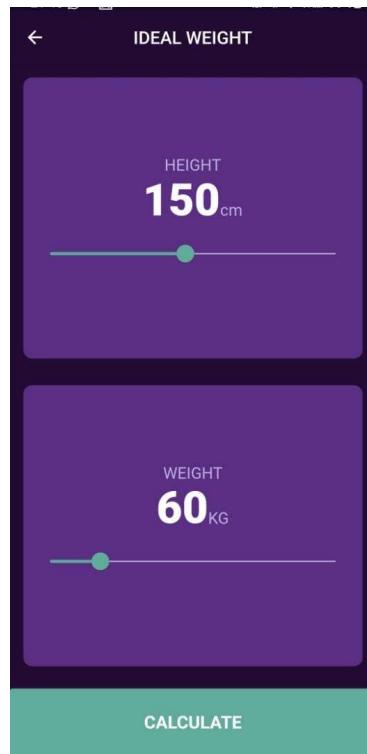


### Ideal Weight Calculator:

**Step 1:** Press on “IDEAL WEIGHT” button.

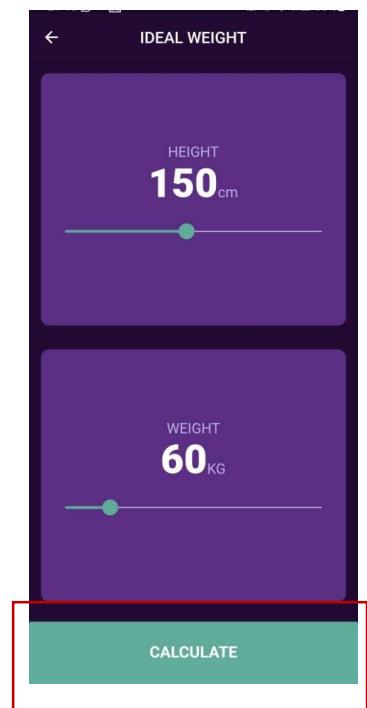


Ideal Weight calculator page will appear.

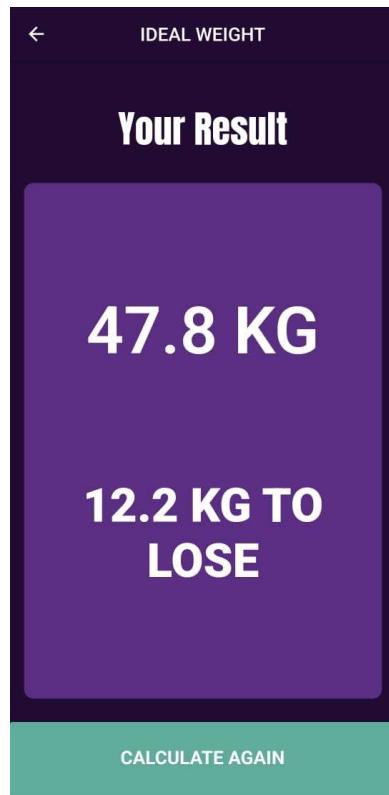


**Step 2:** Set your height (in cm) and weight (in kg).

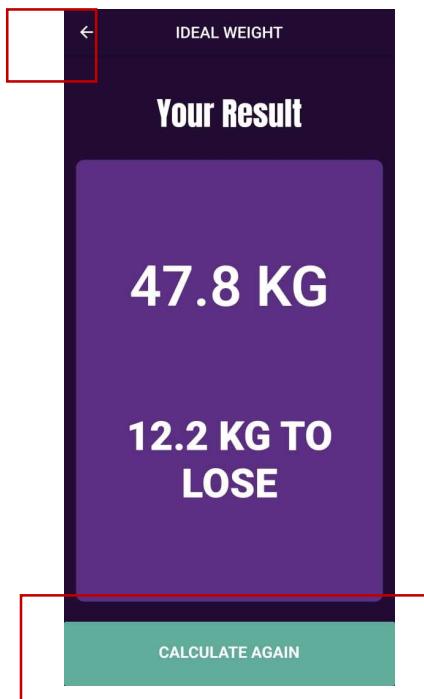
**Step 3:** Press “CALCULATE” button.



**Step 4:** Result page will appear.

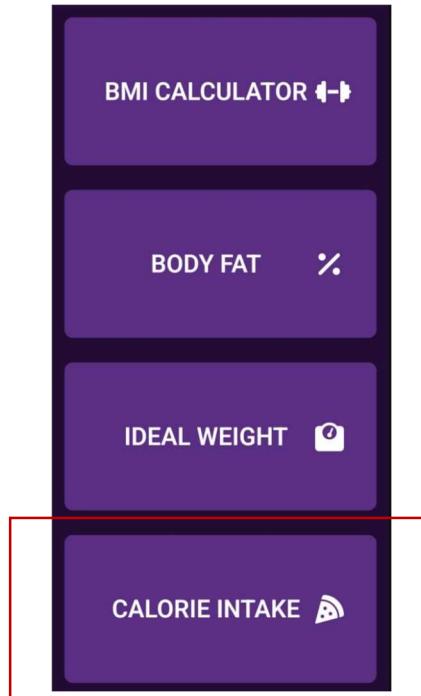


**Step 5:** You can press “CALCULATE AGAIN” button, or back button to go back to BMI calculator page.

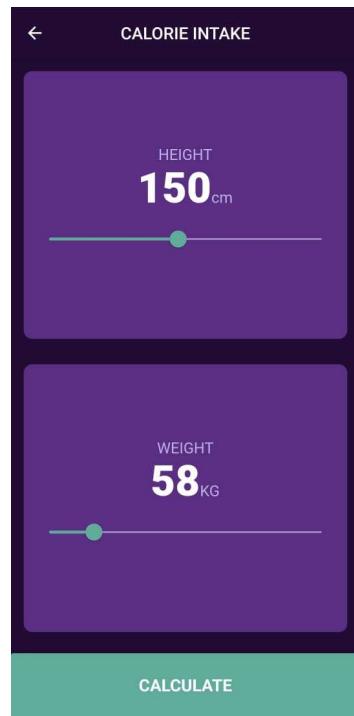


## Calorie Intake Calculator:

**Step 1:** Press on “CALORIE INTAKE” button.

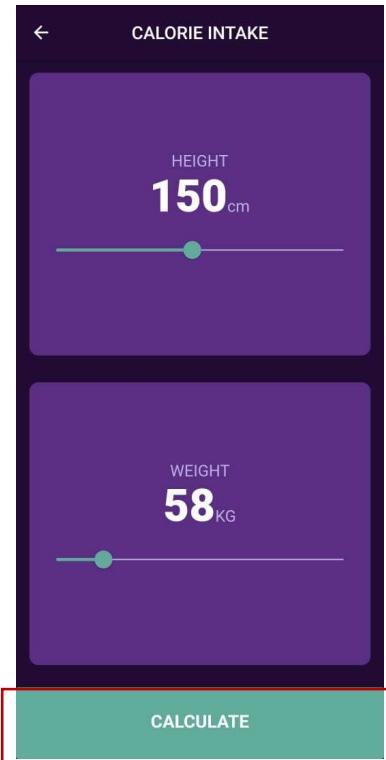


Calorie Intake calculator page will appear.



**Step 2:** Set your height (in cm) and weight (in kg).

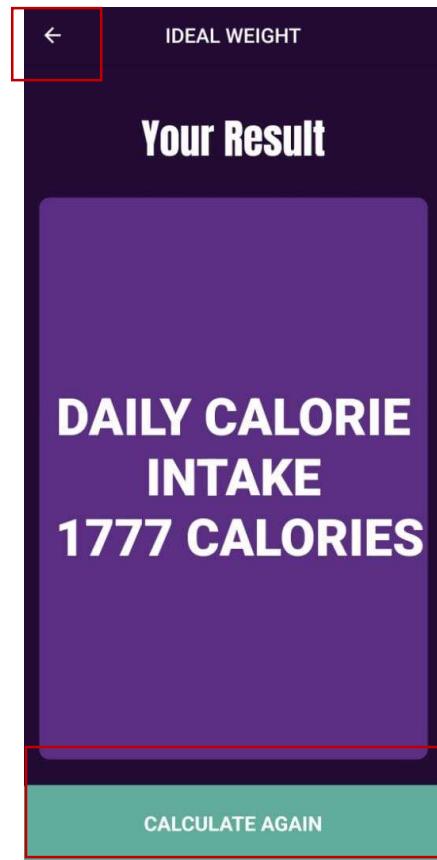
**Step 3:** Press “CALCULATE” button.



**Step 4:** Result page will appear.

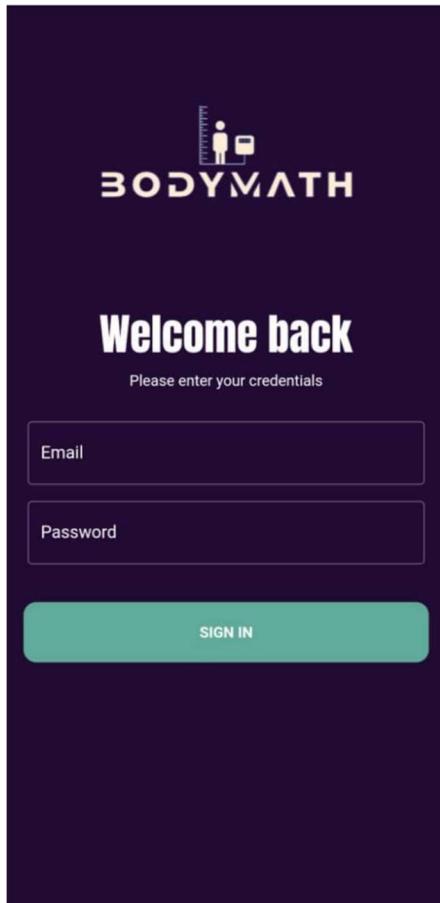


Step 5: You can press “CALCULATE AGAIN” button, or back button to go back to BMI calculator page.



**Admin Login:**

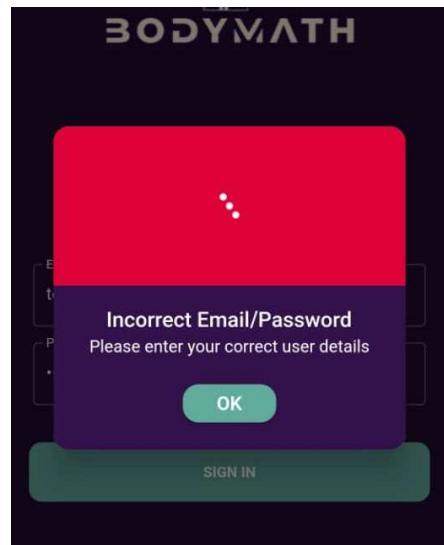
**Step1:** Enter your email and password.



Step2: Press “SIGN IN” button.



In the case of incorrect admin details, pop up error message will appear.

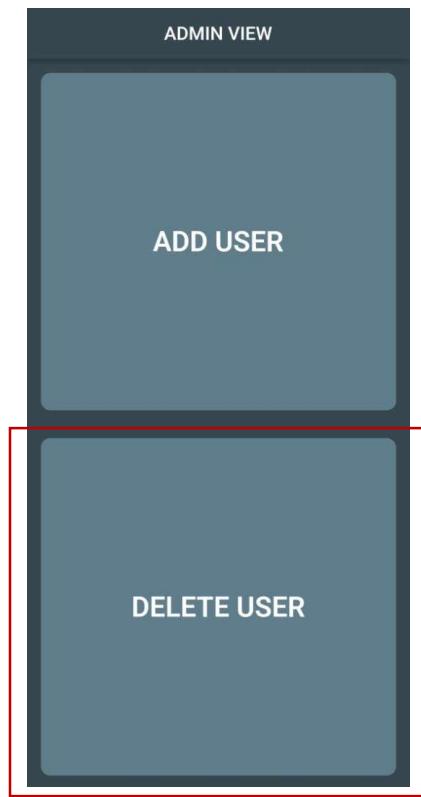


If the admin details are correct, the admin page will appear.

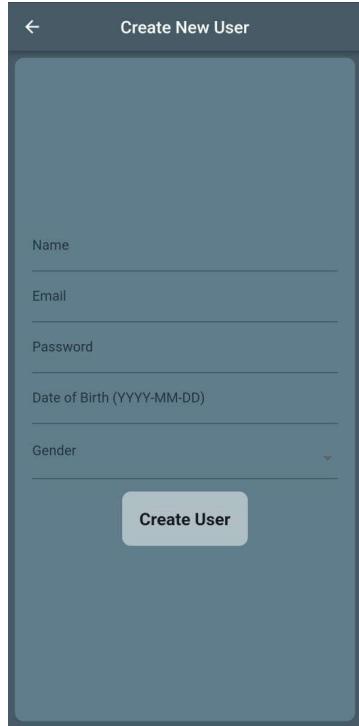


**Add Delete:**

**Step 1:** Press “DELETE USER” button.

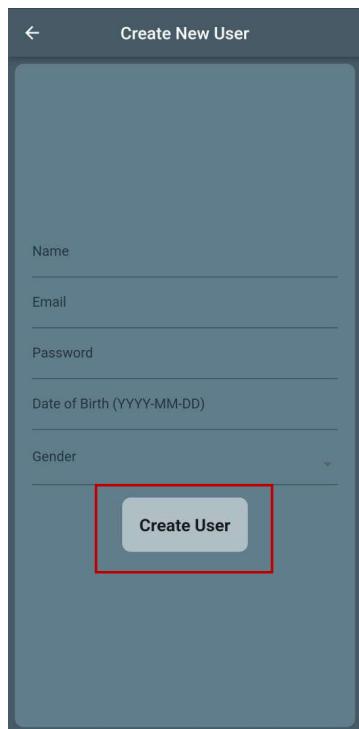


Delete user page will appear.



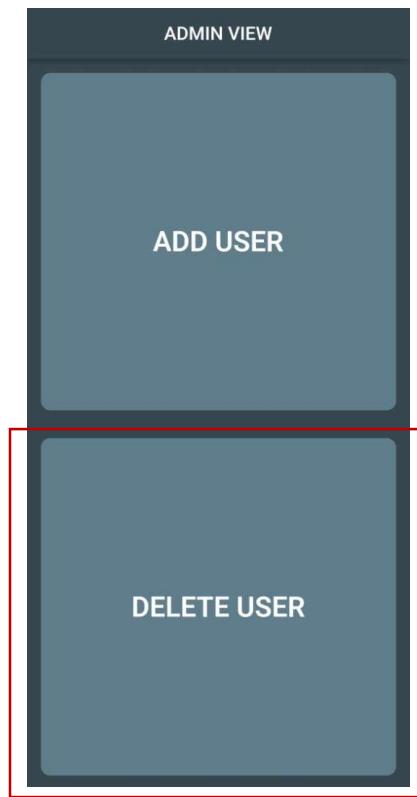
Step 2: Enter user's name, email, password, date of birth, and gender.

Step 3: Press “Create User” button.



## Add User:

**Step 1:** Press “ADD USER” button.

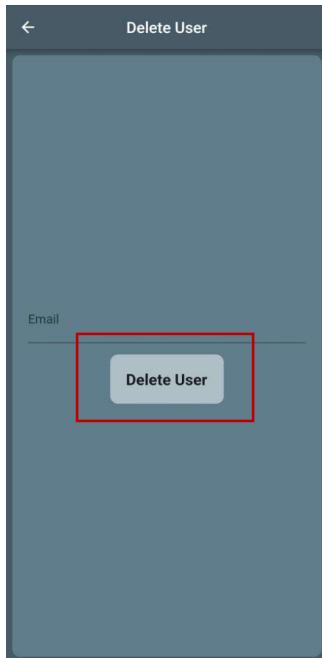


Add user page will appear.



Step 2: Enter user's email.

Step 3: Press “Delete User” button.



If the user's email does not exist, error message will appear.



# Project Post-mortem Report

**Project Title:** BodyMath

**Team Members:**

- 1) Marawan Ashraf
- 2) Ali Karimeh
- 3) Bashir Tawfig
- 4) Yousef Yasser

## Project Overview

What is the main objective of this project?

The main objective of this project is to help users improve their physical health and well-being by providing them with tools and resources for tracking and managing their fitness activities.

What were the original criteria for project success?

BodyMath should contain basic functionality for users, including performing correct and precise calculations of their physique based on their inputs. As for the admins, the application should provide a simple but effective user management system.

Was the project completed according to the original expectation?

The BodyMath project was completed according to the original expectation

## Project Highlights

What elements of the project went wrong?

The application does not account for unrealistic user inputs, which can lead to unrealistic results.

What specific process need improvement?

Due to a lack of proficient graphic designers on our team, the application contained a minimal number of graphic elements.

How can these processes be improved in the future?
By taking graphic design courses, our team could implement what was learned on our application.
What were the key problems areas?
Lack of artistic vision.
List any technical challenges.
An iOS application could not be built due to the lack of Apple MacBooks in our team.

#### Planning Phase

Lesson learned	Achieved	Comments
The project plans and scheduling were well documented, complete with adequate structure and detail.	Yes	
The project schedule contained all elements of the project	Yes	
The tasks were clearly defined	Yes	
The requirements were gathered and clearly documented.	Yes	
The criteria were clear for all phases of the project	Yes	

#### Execution

Lesson learned	Achieved	Comments
The project reached its original goals.	Yes	
Unexpected changes that occurred were of manageable frequency and intensity.	Yes	

Project baselines were thoughtfully managed.	Yes	
Fundamental project management processes were efficient.	Yes	
Project progress was tracked and reported in an accurate, organized manner.	Yes	

### **Human Factors**

Lesson Learned	Achieved	Comments
The project manager reported to the appropriate parties	Yes	
Project management was effective.	Yes	
There was efficient communication among project team members.	Yes	
Functional areas collaborated effectively.	Mixed	Minor issue: no accountability for improbable user inputs. (To be fixed immediately).

### **Overall**

Lesson Learned	Achieved	Comments
Original cost projections were accurate	Yes	
Gym administration needs were met.	Yes	
Objectives of the application were met.	Yes	