VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANA SANGAMA, BELAGAVI – 590 018

An Internship Project Report
on

BMI Calculator App

Submitted in partial fulfillment of the requirements for the VII Semester of degree of **Bachelor of Engineering in Information Science and Engineering** of Visvesvaraya Technological University, Belagavi

By

Anushri Malli

1RN18IS022

Under the Guidance of

Dr. Suresh L
Professor & Head
Department of ISE

ESTD:2001
An Institute with a Difference

Department of Information Science and Engineering
RNS Institute of Technology

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,
Channasandra, Bengaluru-560098

2021-2022

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar post,
Channasandra, Bengaluru - 560098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Internship work entitled ***BMI Calculator App*** has been successfully completed by **Anushri Malli (1RN18IS022)** bonafide student of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements of 7th semester for the award of degree in **Bachelor of Engineering in Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Mrs. Shreedevi Suresh
Internship Guide
Associate Professor
Department of ISE

Dr. Suresh L
Professor and HoD
Department of ISE
RNSIT

Dr. M K Venkatesha
Principal
RNSIT

External Viva

Name of the Examiners **Signature with Date**

1. ____________________ **1.** ____________________

2. ____________________ **2.** ____________________

DECLARATION

I, **Anushri Malli [USN: 1RN18IS022]** student of VII Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Internship work entitled ***BMI Calculator App*** has been carried out by us and submitted in partial fulfillment of the requirements for the *VII Semester degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2021-2022.

Place : Bengaluru

Date : 12/01/2022

Anushri Malli

(1RN18IS022)

ABSTRACT

BMI is a measurement of a person's leanness or corpulence based on their height and weight, and is intended to quantify tissue mass. It is widely used as a general indicator of whether a person has a healthy body weight for their height. Specifically, the value obtained from the calculation of BMI is used to categorize whether a person is underweight, normal weight, overweight, or obese depending on what range the value falls between. These ranges of BMI vary based on factors such as region and age, and are sometimes further divided into subcategories such as severely underweight or very severely obese. Being overweight or underweight can have significant health effects, so while BMI is an imperfect measure of healthy body weight, it is a useful indicator of whether any additional testing or action is required.

With this BMI Calculator, Body Mass Index (BMI) can be calculated and evaluated based on the relevant information on body weight, height and gender. The potential of BMI calculator apps in facilitating weight loss/gain lies in their ability to increase treatment adherence through strategies such as self-monitoring. This help will help to monitor the body mass index for maintaining healthy weight.

ACKNOWLEDGMENT

At the very onset I would like to place our gratefulness to all those people who helped me in making the Internship a successful one.

Coming up, this internship to be a success was not easy. Apart from the sheer effort, the enlightenment of the very experienced teachers also plays a paramount role because it is they who guided me in the right direction.

First, I would like to thank the **Management of RNS Institute of Technology** for providing such a healthy environment for the successful completion of internship work.

In this regard, I express sincere gratitude to my beloved Principal **Dr. M K Venkatesha,** for providing us all the facilities.

I am extremely grateful to my own and beloved Professor and Head of Department of Information science and Engineering, **Dr. Suresh L**, for having accepted to patronize me in the right direction with all her wisdom.

I place my heartfelt thanks to **Dr. Suresh L** Professor and HOD, Department of Information Science and Engineering for having guided internship and all the staff members of the department of Information Science and Engineering for always helping.

I thank **Mr. Akshay D R, ENMAZ**, for providing the opportunity to be a part of the Internship program and having guided me to complete the same successfully.

I also thank our internship coordinator **Dr. R Rajkumar,** Associate Professor, Department of Information Science and Engineering. I would thank my friends for having supported me with all their strength and might. Last but not the least, I thank my parents for supporting and encouraging me throughout. I have made an honest effort in this assignment.

Anushri Malli
1RN18IS022

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATION

UI	User Interface
APK	Android Application Packet
SDK	Software Development Kit
JSON	JavaScript Object Notation

Chapter 1

INTRODUCTION

1.1 Introduction to Flutter

Flutter is Google’s Mobile SDK to build native iOS and Android, Desktop (Windows, Linux, macOS), Web apps from a single codebase. When building applications with Flutter everything towards Widgets – the blocks with which the flutter apps are built. They are structural elements that ship with a bunch of material design-specific functionalities and new widgets can be composed out of existing ones too. The process of composing widgets together is called composition. The User Interface of the app is composed of many simple widgets, each of them handling one particular job. That is the reason why Flutter developers tend to think of their flutter app as a tree of widgets.

1.1.1 History

Flutter launched as a project called Sky which at the beginning worked only on Android. Flutter’s goal is enabling developers to compile for every platform using its own graphic layer rendered by the Skia engine. Here’s a brief presentation of Flutter’s relatively short history.

Flutter is a free and open-source mobile UI framework created by Google and released in May 2017. In a few words, this allows you to create a native mobile application with only one code. It means that you can use one programming language and one codebase to create two different apps (IOS and Android).

The first version of Flutter was known by the codename "Sky" and ran on the Android operating system. It was unveiled at the 2015 Dart developer summit[6] with the stated intent of being able to render consistently at 120 frames per second.[7] During the keynote of Google Developer Days in Shanghai in September 2018, Google announced Flutter Release Preview 2, which is the last big release before Flutter 1.0. On December 4th of that year, Flutter 1.0 was released at the Flutter Live event, denoting the first "stable" version of the Framework. On December 11, 2019, Flutter 1.12 was released at the Flutter Interactive event.[8]

On May 6, 2020, the Dart software development kit (SDK) in version 2.8 and the Flutter in version 1.17.0 were released, where support was added to the Metal API, improving performance on iOS devices (approximately 50%), new Material widgets, and new network tracking.

On March 3, 2021, Google released Flutter 2 during an online Flutter Engage event. This major update brought official support for web-based applications with new CanvasKit renderer and web specific widgets, early-access desktop application support for Windows, macOS, and Linux and improved Add-to-App APIs.[9] This release included sound null-safety, which caused many breaking changes and issues with many external packages, but the Flutter team included instructions to mitigate these changes as well.

On September 8th, 2021, the Dart SDK in version 2.14 and Flutter version 2.5 were released by Google. The update brought improvements to the Android Full-Screen mode and the latest version of Google's Material Design called Material You. Dart received two new updates, the newest lint conditions have been standardized and preset as the default conditions as well Dart for Apple Silicon is now stable.

1.1.2 Framework-Architecture

The major components of Flutter include:

- Dart platform
- Flutter engine
- Foundation library
- Design-specific widgets
- Flutter Development Tools (DevTools)

Dart platform

Flutter apps are written in the Dart language and make use of many of the language's more advanced features.

On Windows, macOS, and Linux[11] Flutter runs in the Dart virtual machine, which features a just-in-time execution engine. While writing and debugging an app, Flutter uses Just In Time compilation, allowing for "hot reload", with which modifications to source files can be injected into a running application. Flutter extends this with support for stateful hot reload, where in most cases changes to source code are reflected immediately in the running app without requiring a restart or any loss of state.

For better performance, release versions of Flutter apps targeting Android and iOS are compiled with ahead-of-time (AOT) compilation.

Flutter engine

Flutter's engine, written primarily in C++, provides low-level rendering support using Google's Skia graphics library. Additionally, it interfaces with platform-specific SDKs such as those provided by Android and iOS.[10] The Flutter Engine is a portable runtime for hosting Flutter applications. It implements Flutter's core libraries, including animation and graphics, file and network I/O, accessibility support, plugin architecture, and a Dart runtime and compile toolchain. Most developers interact with Flutter via the Flutter Framework, which provides a reactive framework and a set of platform, layout, and foundation widgets.

Foundation library

The Foundation library, written in Dart, provides basic classes and functions that are used to construct applications using Flutter, such as APIs to communicate with the engine.

Design-specific widgets

The Flutter framework contains two sets of widgets that conform to specific design languages: Material Design widgets implement Google's design language of the same name, and Cupertino widgets implement Apple's iOS Human interface guidelines.

Chapter 2

LITERATURE SURVEY

2.1 BMI Calculator App

This module calculates the BMI based on the height in meter and weight in kilogram of a user using the formula in.

$$BMI = Weight\ (kg)\ /\ (Height\ (m))^2$$

BMI can be defined as a heuristic proxy for estimating human body fat based on an individual's weight and height. According to World Health Organization (WHO), Asian BMI classifications can be categorized to 4 groups as tabulated. This classification is intended for both men and women.

Category	**Weight**	**Meaning**
1	< 18.5	Underweight
2	18.2 to < 23	Ideal Weight
3	23 to < 27.5	Pre-obese
4	>= 27.5	Obese

Fig: WHO Asian BMI classifications.

2.1.1 Introduction

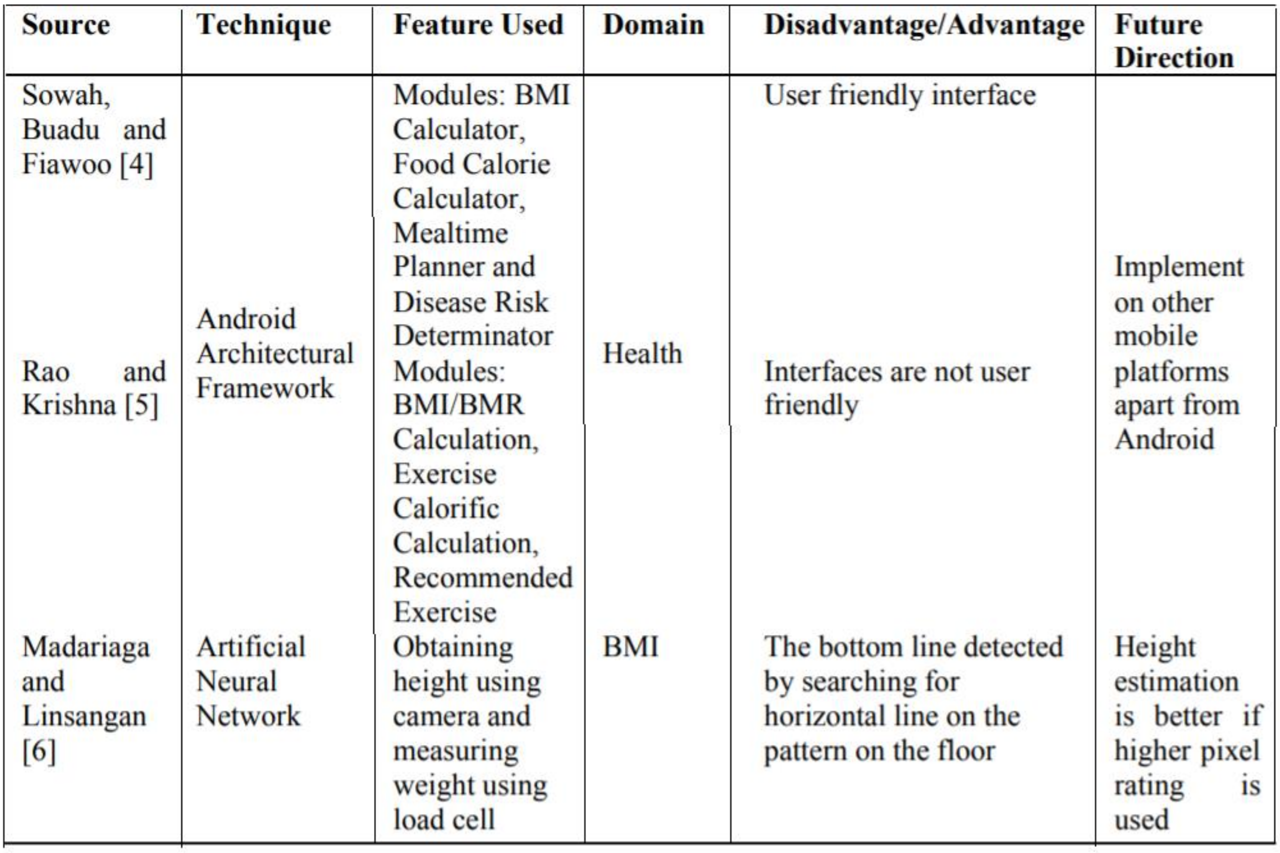
BMI is a measurement of a person's leanness or corpulence based on their height and weight, and is intended to quantify tissue mass. It is widely used as a general indicator of whether a person has a healthy body weight for their height. Specifically, the value obtained from the calculation of BMI is used to categorize whether a person is underweight, normal weight, overweight, or obese depending on what range the value falls between. Being overweight or underweight can have significant health effects.

With this BMI Calculator, Body Mass Index (BMI) can be calculated and evaluated based on the relevant information on body weight, height and gender. The potential of BMI calculator apps in facilitating weight loss/gain lies in their ability to increase treatment adherence through strategies such as self-monitoring. This help will help to monitor the body mass index for maintaining healthy weight.

2.1.2 Related work

2.1.2.1 BMI Calculator application

In order to design and develop BMI Calculator Application, few related works have been reviewed. The related works are summarized below, starting from the earlier published work to the most recent ones.



Source	Technique	Feature Used	Domain	Disadvantage/Advantage	Future Direction
Sowah, Buadu and Fiawoo [4]	Android Architectural Framework	Modules: BMI Calculator, Food Calorie Calculator, Mealtime Planner and Disease Risk Determinator	Health	User friendly interface	Implement on other mobile platforms apart from Android
Rao and Krishna [5]		Modules: BMI/BMR Calculation, Exercise Calorific Calculation, Recommended Exercise		Interfaces are not user friendly	
Madariaga and Linsangan [6]	Artificial Neural Network	Obtaining height using camera and measuring weight using load cell	BMI	The bottom line detected by searching for horizontal line on the pattern on the floor	Height estimation is better if higher pixel rating is used

Fig: Past researches on BMI Calculator

Chapter 3

ANALYSIS

3.1 Hardware and Software Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.

Processor	:	Pentium 4 Processor
Processor Speed	:	2.4 GHz
RAM	:	2 GB
Storage Space	:	40 GB

The software requirements are very minimal and the program can be run on the machines with these requirements satisfied:

Editor	:	Visual Studio Code
Operating System	:	Windows/Mac OS
IDE	:	VS Code
Backend Tool	:	JSON

3.2 Tools/ Languages/ Platform

Various tool used in making this project is given below:

Editor/IDE	:	Visual Studio Code
Operating System	:	Windows/Mac OS
Languages	:	Dart
Backend Tool	:	JSON

3.3 Functional Requirements

Flutter

Flutter is Google's Mobile SDK to build native iOS and Android apps from a single codebase. When building applications with Flutter everything towards Widgets – the blocks with which the flutter apps are built. The User Interface of the app is composed of many simple widgets, each of them handling one particular job. That is the reason why Flutter developers tend to think of their flutter app as a tree of widgets.

Compared to its contemporary technologies like React Native, Kotlin, and Java, Flutter is much better in regard to having a Single Codebase for Android and iOS, Reusable UI and Business Logic, high compatibility, performance, and productivity.

Dart

Dart is an open-source general-purpose programming language developed by Google. It supports application development in both client and server-side. But it is widely used for the development of android apps, iOS apps, IoT(Internet of Things), and web applications using the Flutter Framework.

Syntactically, Dart bears a strong resemblance to Java, C, and JavaScript. It is a dynamic object-oriented language with closure and lexical scope. The Dart language was released in 2011 but came into popularity after 2015 with Dart 2.0.

JSON

JSON (JavaScript Object Notation) is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays (or other serializable values). It is a common data format with diverse uses in electronic data interchange, including that of web applications with servers.

JSON is a language-independent data format. It was derived from JavaScript, but many modern programming languages include code to generate and parse JSON-format data. JSON filenames use the extension .json .

Douglas Crockford originally specified the JSON format in the early 2000s. He and Chip Morningstar sent the first JSON message in April 2001.

Chapter 4

SYSTEM DESIGN

4.1 Input Page widget-tree:

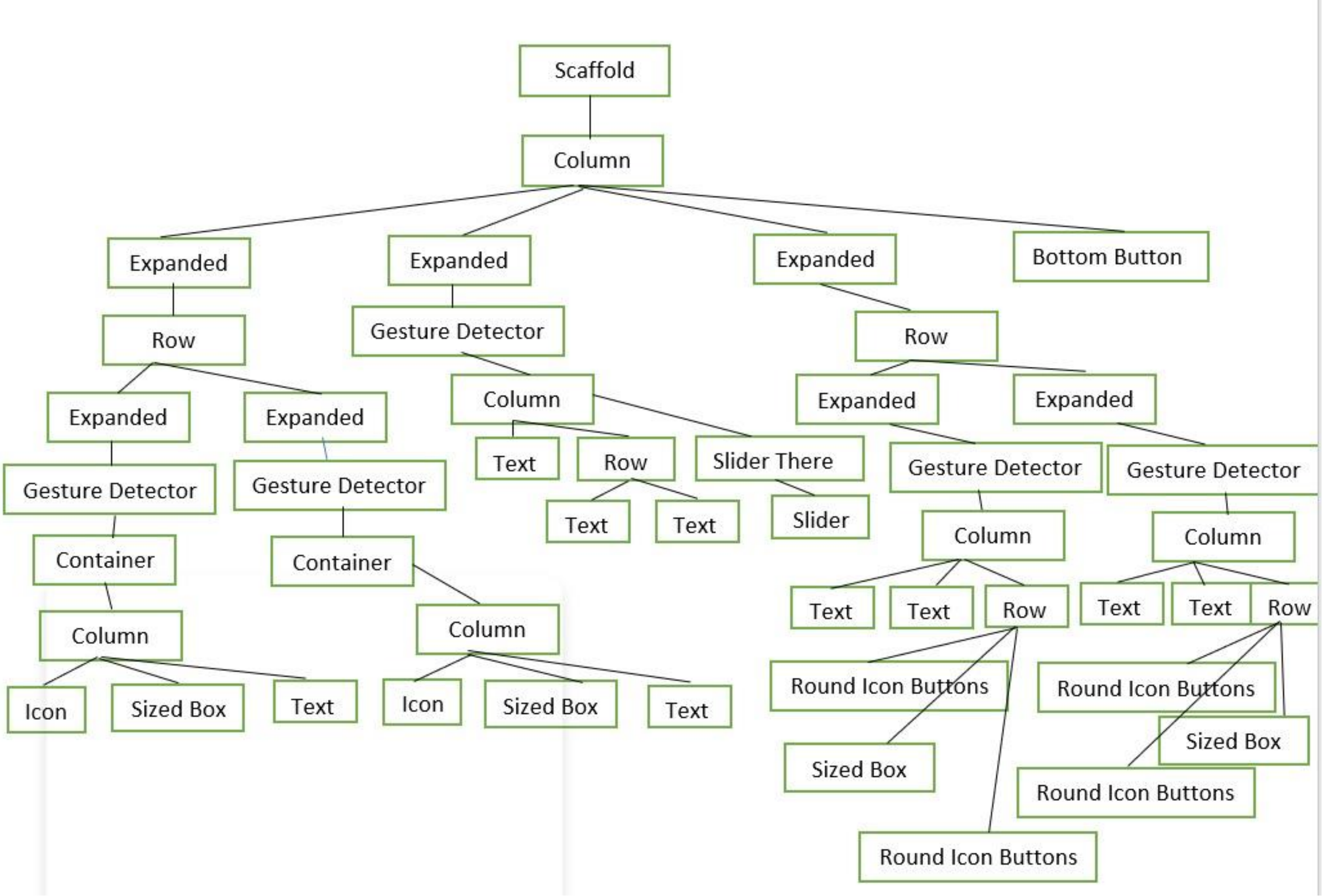


Figure 4.1

4.2 Results Page widget-tree:

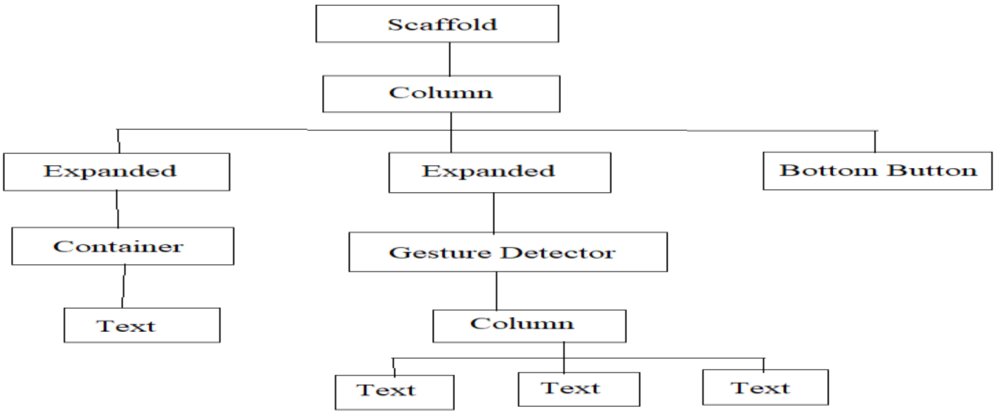


Figure 4.2

4.5 Schema design:

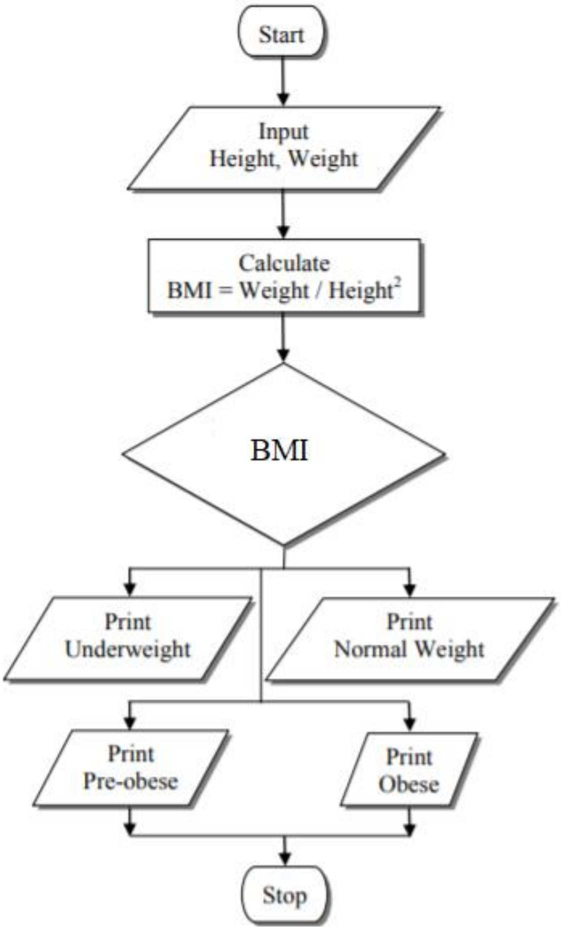
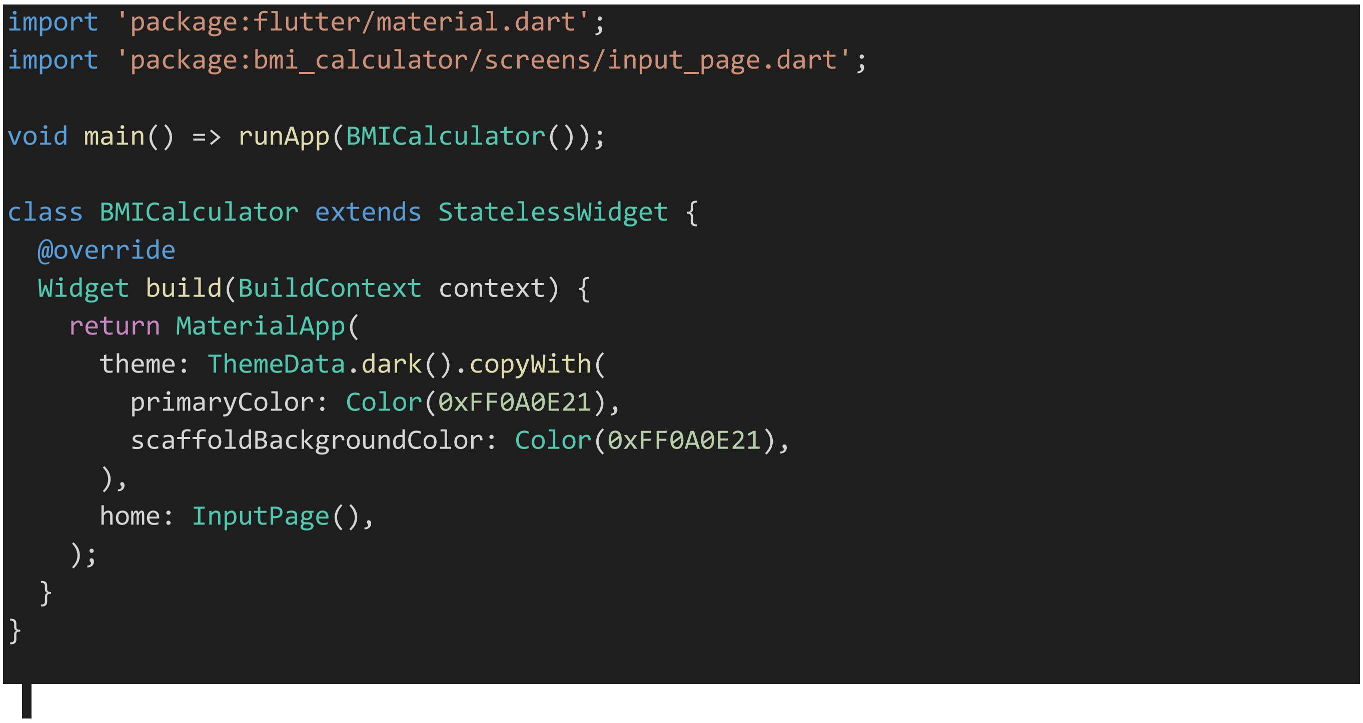


Figure 4.5

Chapter 5

IMPLEMENTATION DETAILS

5.1 main.dart

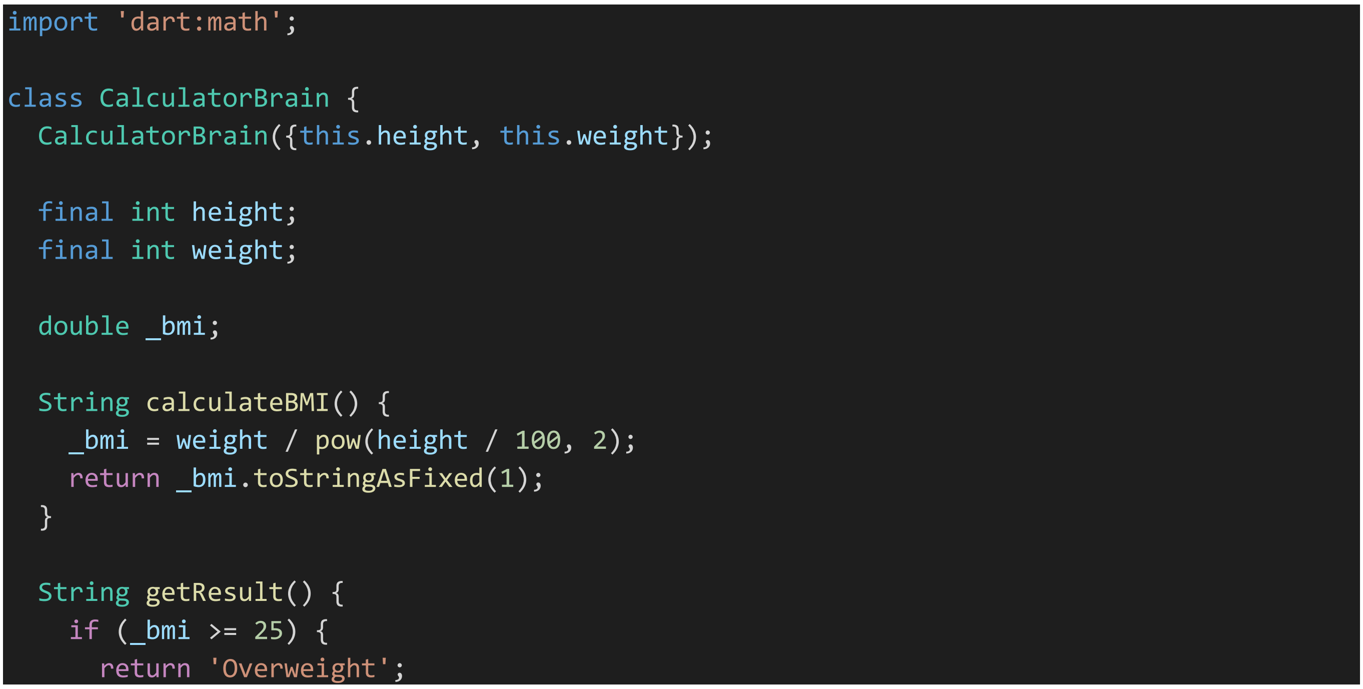


```dart
import 'package:flutter/material.dart';
import 'package:bmi_calculator/screens/input_page.dart';

void main() => runApp(BMICalculator());

class BMICalculator extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData.dark().copyWith(
        primaryColor: Color(0xFF0A0E21),
        scaffoldBackgroundColor: Color(0xFF0A0E21),
      ),
      home: InputPage(),
    );
  }
}
```

5.2 calculator_brain.dart



```dart
import 'dart:math';

class CalculatorBrain {
  CalculatorBrain({this.height, this.weight});

  final int height;
  final int weight;

  double _bmi;

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

  String getResult() {
    if (_bmi >= 25) {
      return 'Overweight';
} else if (_bmi > 18.5) {
      return 'Normal';
    } else {
      return 'Underweight';
    }
  }

  String getInterpretation() {
    if (_bmi >= 25) {
      return 'You have a higher than normal body weight. Try to exercise more.';
    } else if (_bmi >= 18.5) {
      return 'You have a normal body weight. Good job!';
    } else {
      return 'You have a lower than normal body weight. You can eat a bit more.';
    }
  }
}
```

5.3 constants.dart

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

const kBottomContainerHeight = 80.0;
const kActiveCardColour = Color(0xFF1D1E33);
const kInactiveCardColour = Color(0xFF111328);
const kBottomContainerColour = Color(0xFFEB1555);

const kLabelTextStyle = TextStyle(
  fontSize: 18.0,
  color: Color(0xFF8D8E98),
);

const kNumberTextStyle = TextStyle(
  fontSize: 50.0,
  fontWeight: FontWeight.w900,
);

const kLargeButtonTextStyle = TextStyle(
  fontSize: 25.0,
  fontWeight: FontWeight.bold,
);

const kTitleTextStyle = TextStyle(
  fontSize: 50.0,
  fontWeight: FontWeight.bold,
);

const kResultTextStyle = TextStyle(
  color: Color(0xFF24D876),
  fontSize: 22.0,
  fontWeight: FontWeight.bold,
);

const kBMITextStyle = TextStyle(
  fontSize: 100.0,
  fontWeight: FontWeight.bold,
);

const kBodyTextStyle = TextStyle(
  fontSize: 22.0,
);
```

Chapter 6

TESTING

6.1 Introduction

Testing is a process of executing a program with the interest of finding an error. A good test is one that has high probability of finding the yet undiscovered error. Testing should systematically uncover different classes of errors in a minimum amount of time with a minimum number of efforts. Two classes of inputs are provided to test the process

A software configuration that includes a software requirement specification, a design specification and source code.

A software configuration that includes a test plan and procedure, any testing tool and test cases and their expected results.

6.2 Levels of Testing

6.2.1 Unit Testing

Unit testing is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. Unit testing is commonly automated, but may still be performed manually. The objective in unit testing is to isolate a unit and validate its correctness. A manual approachto unit testing may employ a step-by-step instructional document. The unit testing is the process of testing the part of the program to verify whether the program is working corrector not. In this part the main intention is to check the each and every input which we are inserting to our file. Here the validation concepts are used to check whether the program istaking the inputs in the correct format ornot.

Unit testing may reduce uncertainty in the units themselves and can be used in a bottom-up testing style approach. By testing the parts of a program first and then testing the sum of its parts, integration testing becomes much easier. Unit test cases embody characteristics that are critical to the success of the unit.

6.2.2 Integration Testing

Integration testing is also taken as integration and testing this is the major testing process where the units are combined and tested. Its main objective is to verify whether the major parts of the program is working fine or not. This testing can be done by choosing the optionsin the program and by giving suitable inputs.

6.2.3 System Testing

System testing is defined as testing of a complete and fully integrated software product. This testing falls in black-box testing wherein knowledge of the inner design of the code isnot a pre-requisite and is done by the testing team. System testing is done after integrationtesting is complete. System testing should test functional and non-functional requirementsof the software.

6.2.4 Validation Testing

In this, requirements established as part of software requirements analysis are validated against the software that has been constructed. Validation testing provides final assurance that software meets all functional, behavioral and performance requirements. Validation can be defined in many ways but a simple definition is that validation succeeds when software Function in a manner that can be reasonably by the customer.

1. Validation test criteria
2. Configuration review
3. Alpha and Beta testing (conducted by end user)

6.2.5 Output Testing

After preparing test data, the system under study is tested using the test data. While testing the system using test data, errors are again uncovered and corrected by using above testing and corrections are also noted for future use.

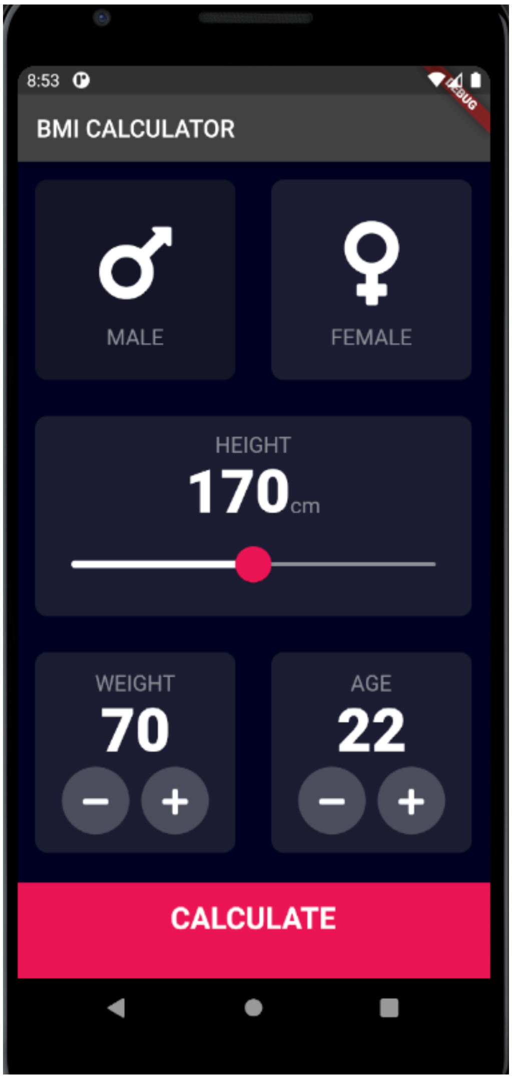
6.2.6 User Acceptance Testing

User acceptance testing is a type of testing performed by the end user or the client to verify/accept the software application to the production environment.

User Acceptance Testing is done in the final phase of testing.

Chapter 7

DISCUSSION OF RESULTS



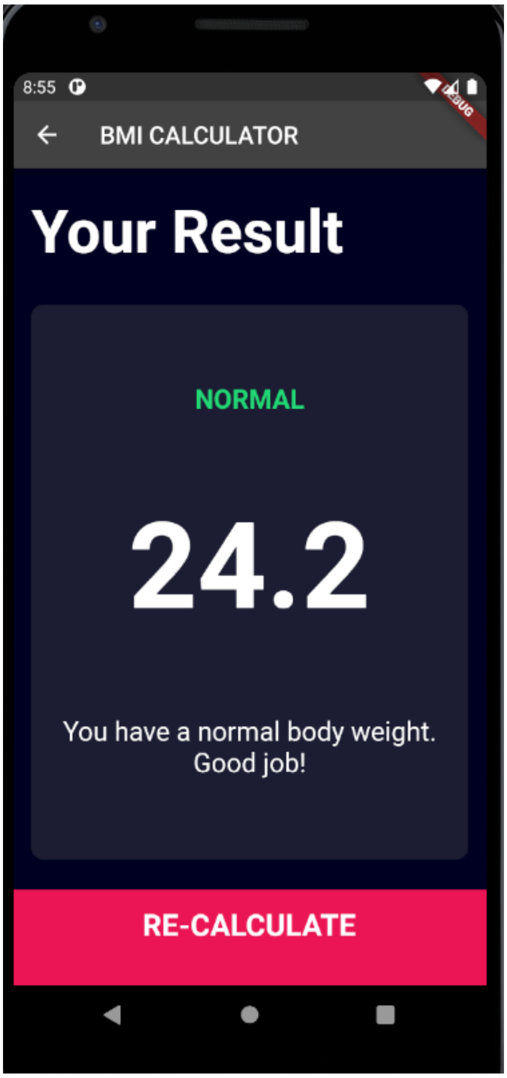
HEIGHT 170 cm

WEIGHT 70

AGE 22

Figure 7.1

This screen asks the user to enter their Gender, height, weight and age to calculate their BMI.



Your Result

NORMAL

24.2

You have a normal body weight. Good job!

Figure 7.2

This screen shows if the user has a proper Body Mass Index and shows if it is normal, underweight or overweight.

Chapter 8

CONCLUSION AND FUTURE REFERENCES

8.1 Conclusion

An application that calculates body mass index on different categories including underweight, normal weight, pre-obese and obese is needed by the health-conscious community. Being overweight or underweight can have significant health effects, so while BMI is an imperfect measure of healthy body weight, it is a useful indicator of whether any additional testing or action is required.

Although BMI is a widely used and useful indicator of healthy body weight, it does have its limitations. BMI is only an estimate that cannot take body composition into account. Due to a wide variety of body types as well as distribution of muscle, bone mass, and fat, BMI should be considered along with other measurements rather than being used as the sole method for determining a person's healthy body weight.

8.2 Future Reference

In the future, this application could be greatly improved and expanded to include new features. Some of them are mentioned below-

- The ideal weight after the BMI index calculation can be hown. So that the user knows how many kgs need to be gained or lost to be ideal weight.
- A meal planning feature can be added to help people get an idea of what they should eat to gain/lose weight to get to their ideal weight.
- A chart can be added to keep a track of changes in BMI over the months.
- A login feature can be added for the app to be used by multiple users and save their progress under their own accounts.

Chapter 9

REFERENCES

1. **https://rubygarage.org**

2. **www.hackernoon.com**

3. **www.ncbi.nlm.nih.gov**

4. **www.ispo.com**

5. **www.github.com**

6. **www.stackoverflow.com**

7. Adria Muntaner-Mas, Antonio Martinez-Nicolas, Carl J. Lavie, Steven N. Blair, Robert Ross, Ross Arena, and Francisco B. Ortega (2019). A Systematic Review of Fitness Apps and Their Potential Clinical and Sports Utility for Objective and Remote Assessment of Cardiorespiratory Fitness. Sports Medicine 2019, 49(4), 587-600. doi:10.1007/s40279-019-01084-y