

Project Report: BMI Calculator Mobile Application

Unit: Mobile Applications Design and Development

Development Environment: Android Studio (Java)

• Introduction

The objective of this project was to develop a mobile application that calculates the Body Mass Index (BMI) of a user based on their weight and height. The app serves as a health monitoring tool, providing immediate feedback on whether a user's BMI falls within a healthy range.

• User Interface (UI) Design

The UI was designed using **XML (Extensible Markup Language)** with a focus on simplicity and user efficiency.

• Layout Architecture

The app utilizes a **LinearLayout** with a vertical orientation. This choice ensures a linear flow of data entry, which is ideal for utility applications where the user follows a step-by-step process.

• UI Components

- **EditText (Weight & Height):** Configured with `inputType="numberDecimal"`. This is a critical design choice that restricts user input to numbers and decimals, improving the overall User Experience (UX) by automatically launching the numeric keypad.
- **Buttons (Compute & Reset):** Large, accessible buttons are used to trigger the logic or clear the state of the app.
- **TextView (Result):** Positioned at the bottom to display the final calculation and health category.

• Application Logic and Implementation

The logic of the application is handled in **MainActivity.java**, which bridges the gap between the user interface and the mathematical calculations.

- **Event Handling**

The app uses **OnClickListeners** for the buttons. When the "Compute" button is pressed, the app executes a multi-step logic flow:

- **Capture:** It retrieves the string input from the `EditText` fields.
- **Validation:** It checks if the fields are empty. If they are, it uses `setError()` to provide visual feedback directly on the input box—a standard Android design pattern.
- **Error Handling:** A `try-catch` block is implemented to catch `NumberFormatException`. This prevents the app from crashing if a user enters invalid characters (like a single decimal point).

- **The Mathematical Algorithm**

The core of the app is the BMI formula:

$$BMI = \frac{\text{weight(kg)}}{\text{height(m)}^2}$$

In Java, this is implemented as:

```
double bmi = weight / (height * height);
```

- **Classification Logic**

The app uses a private helper method, `getBMICategory()`, to interpret the numerical result based on World Health Organization (WHO) standards:

BMI Range	Category
< 18.5	Underweight
18.5 – 24.9	Normal
25.0 – 29.9	Overweight
> 30.0	Obese

- **Key Features & Functionality**

- **Reset Functionality:** The `btnReset` listener clears all inputs and result text, allowing for a fresh calculation without manually deleting text.
- **Input Focus:** The use of `requestFocus()` during validation ensures that the cursor automatically moves to the field that needs correction, streamlining the user interaction.
- **Data Formatting:** The result is formatted to **two decimal places** using `String.format()`, ensuring the output is professional and readable.

- APP IN WORK:

79

2

Compute BMI

Reset

BMI: 19.75 (Normal)

60

2

Compute BMI

Reset

BMI: 15.00 (Underweight)

Medium Phone API 36.1 x +

0

0

Compute BMI

Reset

Error: Values must be greater than 0

+

-

1:1

Medium Phone API 36.1 x +

50

1.3

Compute BMI

Reset

BMI: 29.59 (Overweight)

+

-

1:1

6. Conclusion

The BMI Calculator successfully demonstrates the core principles of Android development: UI design via XML, event-driven programming in Java, and robust error handling. The app is not only functional but also resilient against common user input errors.