

**Lucrarea de laborator nr.1**

**Disciplina: Programarea Aplicațiilo Mobile**

**A efectuat student (ul/a): \_\_\_\_\_\_\_\_\_\_\_\_**(Nicolaescu Alina, gr.IA-221)

**A controlat:\_\_\_\_\_\_\_\_\_\_\_\_\_\_**(asist.univ. D.Buza )

**Chisinau 2024**

**Cuprins**

[**Sarcina:** 2](#_Toc178244988)

[**Descrierea Sarcinii:** 2](#_Toc178244989)

[**Descrierea aplicatiei:** 2](#_Toc178244990)

[**Codul:** 3](#_Toc178244991)

# **Sarcina:**

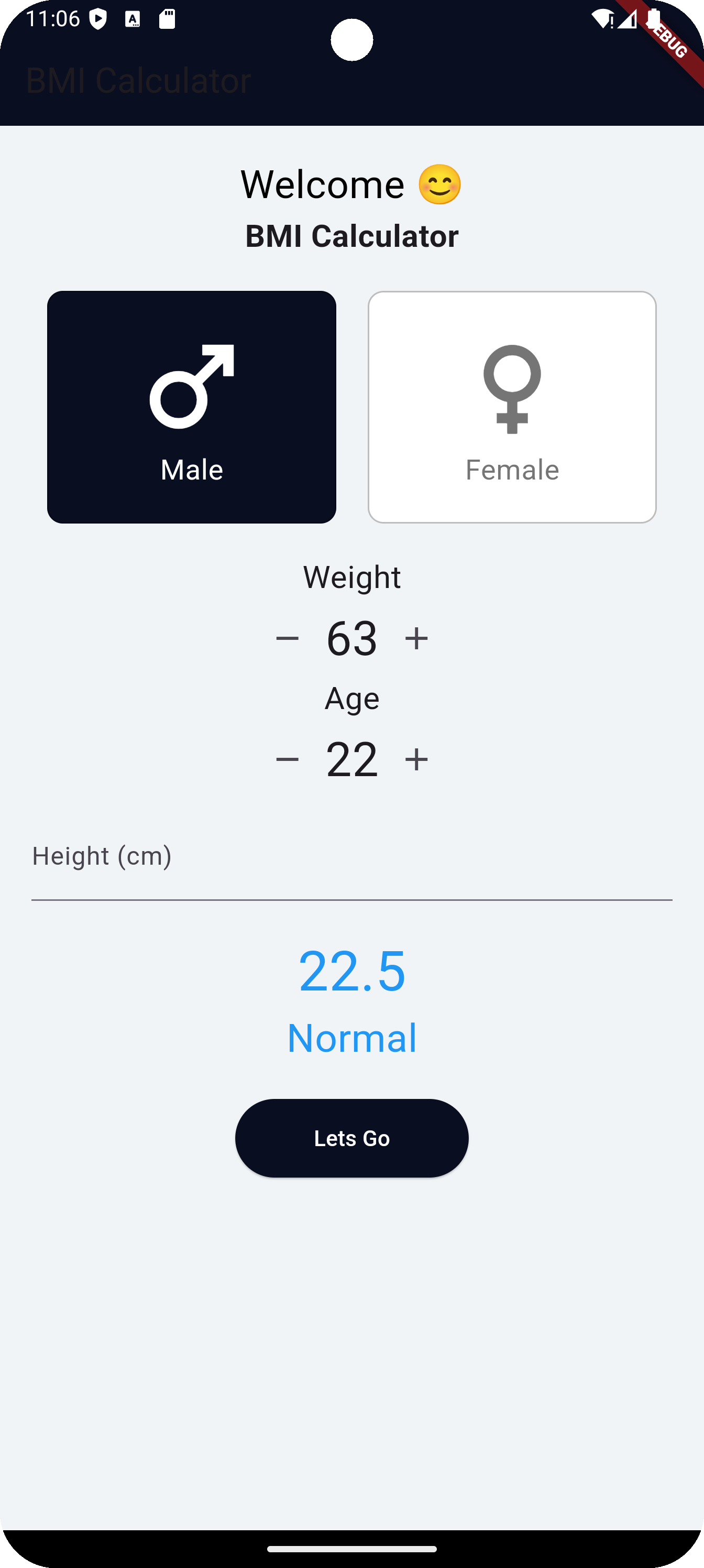
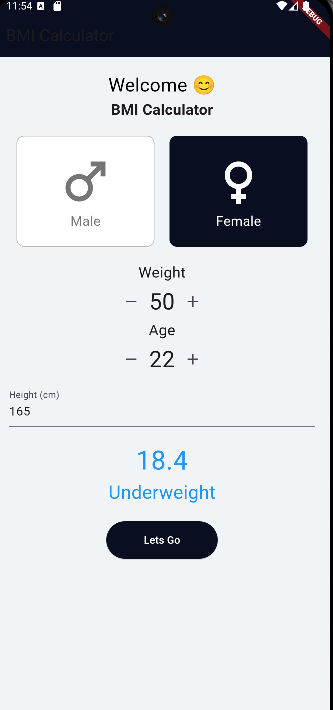
Dezvoltarea unei aplicații mobile pentru măsurarea normalității greutății unei persoane

# **Descrierea Sarcinii:**

Această sarcină implică dezvoltarea unei aplicații mobile care să permită utilizatorilor să evalueze normalitatea greutății lor în funcție de indicele de masă corporală (IMC). Aplicația trebuie să fie intuitivă, ușor de utilizat și să ofere rezultate precise pe baza unor parametri introduși, cum ar fi greutatea, înălțimea, sexul și vârsta utilizatorului.

# **Descrierea aplicatiei:**

Aplicația creată este una simplă și intuitivă pentru calcularea indicelui de masă corporală (IMC), dezvoltată folosind Flutter. Aplicația oferă utilizatorilor o interfață prietenoasă și eficientă, care le permite să introducă rapid și ușor parametri personali precum genul, greutatea, vârsta și înălțimea. Aceasta este construită într-un stil modern, folosind o temă luminoasă și personalizată, care îmbunătățește experiența utilizatorului.



***Figura 1. Prezentarea interfeței aplicației***

Interfața principală include două butoane clare pentru selecția genului, masculin și feminin, fiecare reprezentat prin iconuri intuitive și text. Selecția este marcată prin schimbarea vizuală a butonului ales, controlată de o variabilă booleană. Aceste elemente vizuale sunt combinate pentru a asigura o experiență de utilizare fluidă, fără ambiguități.

Utilizatorii pot ajusta parametrii precum greutatea și vârsta folosind butoane de tip increment/decrement, amplasate strategic pentru a ușura interacțiunea. Greutatea și vârsta sunt afișate în centrul acestor controale, iar valorile pot fi ajustate fără introducerea manuală. Înălțimea este introdusă printr-un câmp text, unde utilizatorul poate introduce valoarea dorită în centimetri, ceea ce este ulterior convertit și utilizat în calculul IMC.

Atunci când toți parametrii sunt introduși, aplicația calculează IMC-ul utilizatorului folosind formula clasică – greutatea împărțită la pătratul înălțimii în metri. Valoarea calculată este afișată într-un format mare și colorat, iar utilizatorul primește o interpretare clară a rezultatului său, clasificată în funcție de standardele internaționale (subponderal, normoponderal, supraponderal sau obezitate).

Codul include, de asemenea, componente reutilizabile, precum GenderButton pentru selecția genului și InputField pentru ajustarea valorilor de greutate și vârstă, asigurând astfel o arhitectură modulară și ușor de întreținut. În concluzie, aplicația oferă o metodă rapidă și eficientă pentru evaluarea IMC-ului, cu o interfață plăcută și feedback instant asupra categoriei de greutate în care se încadrează utilizatorul.

# **Codul:**

import 'package:flutter/material.dart';  
  
void main() => runApp(BMICalculator());  
  
class BMICalculator extends StatelessWidget {  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 theme: ThemeData.light().copyWith(  
 primaryColor: Color(0xFF0A0E21),  
 scaffoldBackgroundColor: Color(0xFFF1F4F6),  
 ),  
 home: BMICalculatorPage(),  
 );  
 }  
}  
  
class BMICalculatorPage extends StatefulWidget {  
 @override  
 \_BMICalculatorPageState createState() => \_BMICalculatorPageState();  
}  
  
class \_BMICalculatorPageState extends State<BMICalculatorPage> {  
 bool isMale = true;  
 double weight = 70;  
 int age = 23;  
 double height = 170;  
 double bmi = 0;  
 String bmiCategory = '';  
  
 void calculateBMI() {  
 if (height > 0) {  
 setState(() {  
 bmi = weight / ((height / 100) \* (height / 100));  
 if (bmi < 18.5) {  
 bmiCategory = "Underweight";  
 } else if (bmi >= 18.5 && bmi <= 24.9) {  
 bmiCategory = "Normal";  
 } else if (bmi >= 25 && bmi <= 29.9) {  
 bmiCategory = "Overweight";  
 } else {  
 bmiCategory = "Obese";  
 }  
 });  
 }  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 title: Text('BMI Calculator'),  
 backgroundColor: Color(0xFF0A0E21),  
 ),  
 body: Padding(  
 padding: const EdgeInsets.all(20.0),  
 child: Column(  
 children: [  
 Text('Welcome 😊', style: TextStyle(fontSize: 25, color: Colors.*black*)),  
 Text('BMI Calculator', style: TextStyle(fontSize: 20, fontWeight: FontWeight.*bold*)),  
 SizedBox(height: 20),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 GenderButton(  
 gender: "Male",  
 selected: isMale,  
 onTap: () {  
 setState(() {  
 isMale = true;  
 });  
 },  
 ),  
 GenderButton(  
 gender: "Female",  
 selected: !isMale,  
 onTap: () {  
 setState(() {  
 isMale = false;  
 });  
 },  
 ),  
 ],  
 ),  
 SizedBox(height: 20),  
 InputField(  
 label: 'Weight',  
 value: weight.toStringAsFixed(0),  
 onIncrement: () {  
 setState(() {  
 weight++;  
 });  
 },  
 onDecrement: () {  
 setState(() {  
 if (weight > 0) weight--;  
 });  
 },  
 ),  
 InputField(  
 label: 'Age',  
 value: age.toString(),  
 onIncrement: () {  
 setState(() {  
 age++;  
 });  
 },  
 onDecrement: () {  
 setState(() {  
 if (age > 0) age--;  
 });  
 },  
 ),  
 SizedBox(height: 10),  
 TextField(  
 keyboardType: TextInputType.*number*,  
 decoration: InputDecoration(labelText: 'Height (cm)'),  
 onChanged: (value) {  
 setState(() {  
 height = double.*tryParse*(value) ?? 0;  
 });  
 },  
 ),  
 SizedBox(height: 20),  
 Text(  
 bmi == 0 ? 'BMI' : bmi.toStringAsFixed(1),  
 style: TextStyle(fontSize: 35, color: Colors.*blue*),  
 ),  
 Text(  
 bmiCategory,  
 style: TextStyle(fontSize: 25, color: Colors.*blue*),  
 ),  
 SizedBox(height: 20),  
 ElevatedButton(  
 onPressed: calculateBMI,  
 style: ElevatedButton.*styleFrom*(  
 backgroundColor: Color(0xFF0A0E21),  
 padding: EdgeInsets.symmetric(horizontal: 50, vertical: 15),  
 ),  
 child: Text('Lets Go', style: TextStyle(color: Colors.*white*)),  
 ),  
 ],  
 ),  
 ),  
 );  
 }  
}  
  
class GenderButton extends StatelessWidget {  
 final String gender;  
 final bool selected;  
 final VoidCallback onTap;  
  
 GenderButton({required this.gender, required this.selected, required this.onTap});  
  
 @override  
 Widget build(BuildContext context) {  
 return Expanded(  
 child: GestureDetector(  
 onTap: onTap,  
 child: Container(  
 margin: EdgeInsets.symmetric(horizontal: 10),  
 decoration: BoxDecoration(  
 color: selected ? Color(0xFF0A0E21) : Colors.*white*,  
 borderRadius: BorderRadius.circular(10),  
 border: Border.all(color: Colors.*black26*),  
 ),  
 padding: EdgeInsets.symmetric(vertical: 20),  
 child: Column(  
 children: [  
 Icon(gender == "Male" ? Icons.*male* : Icons.*female*, size: 80, color: selected ? Colors.*white* : Colors.*black54*),  
 Text(gender, style: TextStyle(fontSize: 18, color: selected ? Colors.*white* : Colors.*black54*)),  
 ],  
 ),  
 ),  
 ),  
 );  
 }  
}  
  
class InputField extends StatelessWidget {  
 final String label;  
 final String value;  
 final VoidCallback onIncrement;  
 final VoidCallback onDecrement;  
  
 InputField({required this.label, required this.value, required this.onIncrement, required this.onDecrement});  
  
 @override  
 Widget build(BuildContext context) {  
 return Column(  
 children: [  
 Text(label, style: TextStyle(fontSize: 20)),  
 Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 IconButton(  
 icon: Icon(Icons.*remove*),  
 onPressed: onDecrement,  
 ),  
 Text(value, style: TextStyle(fontSize: 30)),  
 IconButton(  
 icon: Icon(Icons.*add*),  
 onPressed: onIncrement,  
 ),  
 ],  
 ),  
 ],  
 );  
 }  
}

**Concluzii**

În acest laborator, s-a dezvoltat o aplicație mobilă pentru calculul indicelui de masă corporală (IMC), utilizând Flutter, care oferă o interfață intuitivă și compatibilitate multiplatformă. S-a realizat o implementare eficientă și modulară, cu widgeturi personalizate pentru selecția genului și ajustarea parametrilor de greutate și vârstă. Interfața simplifică introducerea datelor, iar calculul IMC este rapid și afișat într-un mod clar, cu interpretarea rezultatelor conform standardelor internaționale.

Aplicația permite utilizatorilor să interacționeze facil cu elementele vizuale, oferind un feedback imediat prin ajustarea dinamică a valorilor și afișarea rezultatelor în funcție de datele introduse. Utilizarea butoanelor incrementale și a câmpurilor de introducere manuală a asigurat o experiență fluidă, contribuind la o utilizare practică și fără complicații.

Astfel, aplicația demonstrează că Flutter este ideal pentru dezvoltarea rapidă de aplicații mobile cu o experiență prietenoasă pentru utilizatori, adresându-se unui public divers, interesat de monitorizarea sănătății prin evaluarea greutății.