Ministerul Educaţiei, Culturii și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Departamentul Ingineria Software și Automatică

**RAPORT**

Lucrare de laborator nr. 2, varianta II

Disciplina: PAM

Tema: Utilizare avansata de componente UI. Realizare app dupa design definit complex.

A efectuat: Russu Iulia TI-224

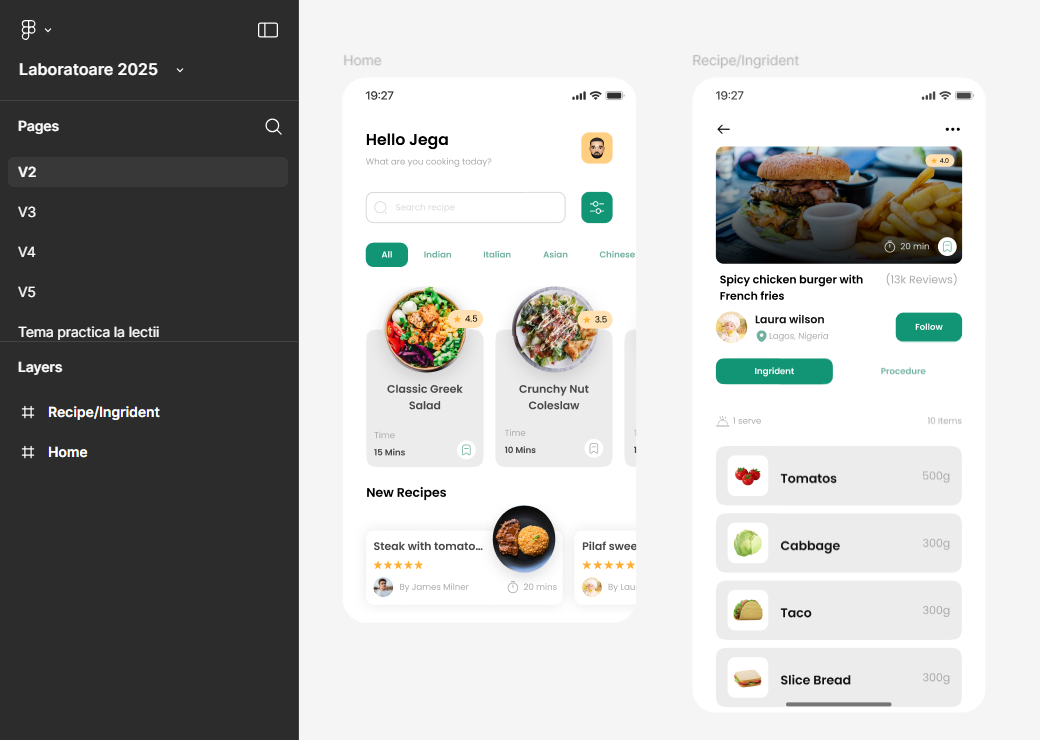
A verificat :

Buza D.

Chișinău 2025

**Sarcina de lucru**: Varianta 2.

Utilizare avansata de componente UI. Realizare app dupa design definit complex.  
Designul prezentat în Figma:



**Mersul lucrării:**

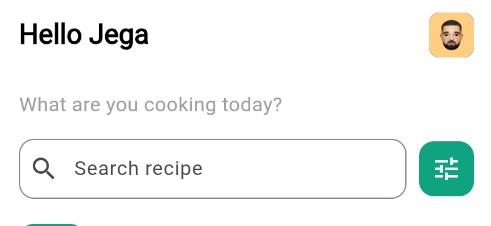


Figura 1 –Secțiunea cu search bar

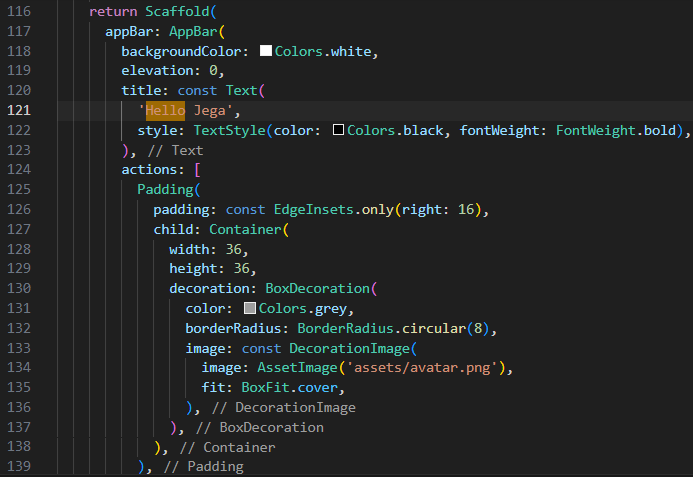


Figura 1.2 –Titlul + imaginea cu avatar-ul user-ului

În acest fragment de cod sunt utilizate elemente pentru crearea unei interfețe grafice. Scaffold este structura de bază a unei pagini în Flutter și oferă cadrul principal în care pot fi plasate componente precum bara de sus, conținutul sau butoanele de navigare. AppBar reprezintă bara de titlu aflată în partea superioară a ecranului; în acest caz, are un fundal alb, fără umbră (elevation: 0) și conține un titlu și o acțiune în partea dreaptă. Text este utilizat pentru afișarea textului „Hello Jega”, având un stil personalizat — culoare neagră și text îngroșat. În interiorul actions, este folosit un Padding pentru a adăuga spațiu la dreapta, iar un Container cu proprietăți de BoxDecoration afișează un avatar, aplicând o culoare gri, colțuri rotunjite și o imagine din fișierul local assets/avatar.png. Astfel, aceste elemente lucrează împreună pentru a construi o bară de aplicație estetică și funcțională.

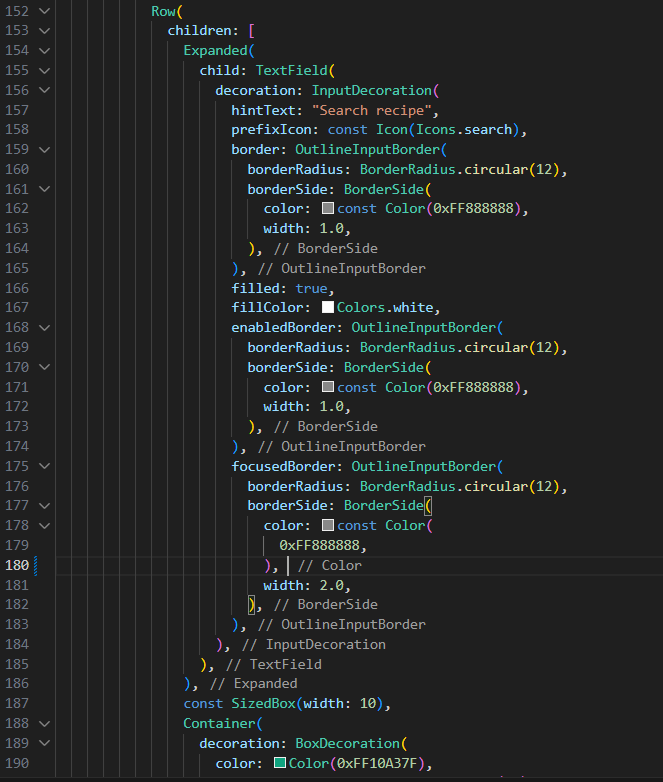


Figura 1.3 –SearchBar + butonul cu filtre

În acest fragment de cod sunt utilizate mai multe elemente UI pentru a construi secțiunea principală a paginii. SingleChildScrollView permite derularea conținutului pe ecran, astfel încât toate elementele din interiorul său pot fi vizualizate chiar și pe ecrane mici. Proprietatea padding adaugă un spațiu uniform în jurul conținutului. În interior se află un Column, care aliniază elementele vertical, iar crossAxisAlignment: CrossAxisAlignment.start asigură alinierea lor la stânga. Primul element este un Text care afișează întrebarea „What are you cooking today?” cu o dimensiune mică și culoare gri. SizedBox este folosit pentru a adăuga spațiu între elemente. Urmează un Row, care aliniază orizontal un TextField și un Container cu o pictogramă. TextField este câmpul de căutare, având un InputDecoration ce include un indiciu (hintText), o pictogramă de căutare (prefixIcon), un fundal alb și margini rotunjite. OutlineInputBorder definește conturul câmpului atât pentru starea activă (focusedBorder) cât și pentru cea normală (enabledBorder). Alături, Container are un fundal verde, colțuri rotunjite și conține o pictogramă de tip Icon(Icons.tune), reprezentând un buton de filtrare. Toate aceste elemente împreună creează o interfață clară și modernă pentru introducerea și filtrarea rețetelor.

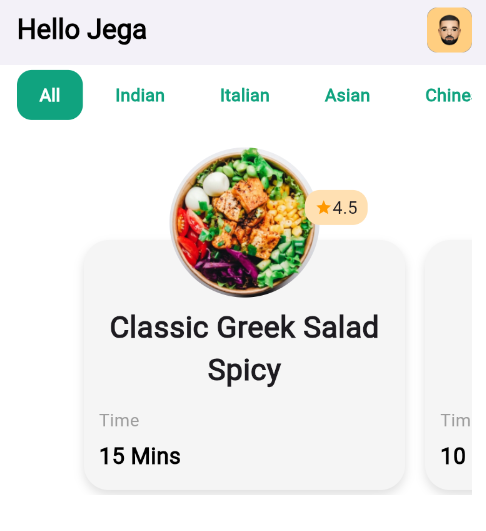


Figura 2 – Secțiunea cu meniul principal, cu butoane pentru categorii

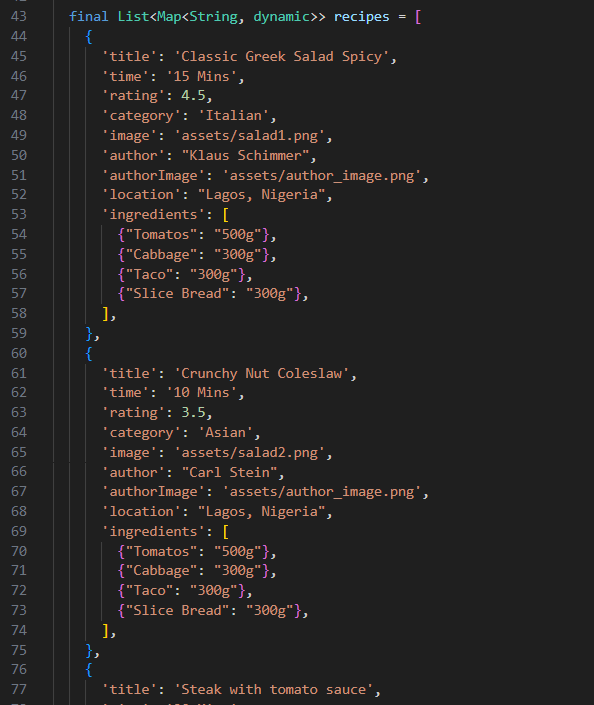


Figura 2.1 – Lista declarată cu toate elementele meniului (recete)

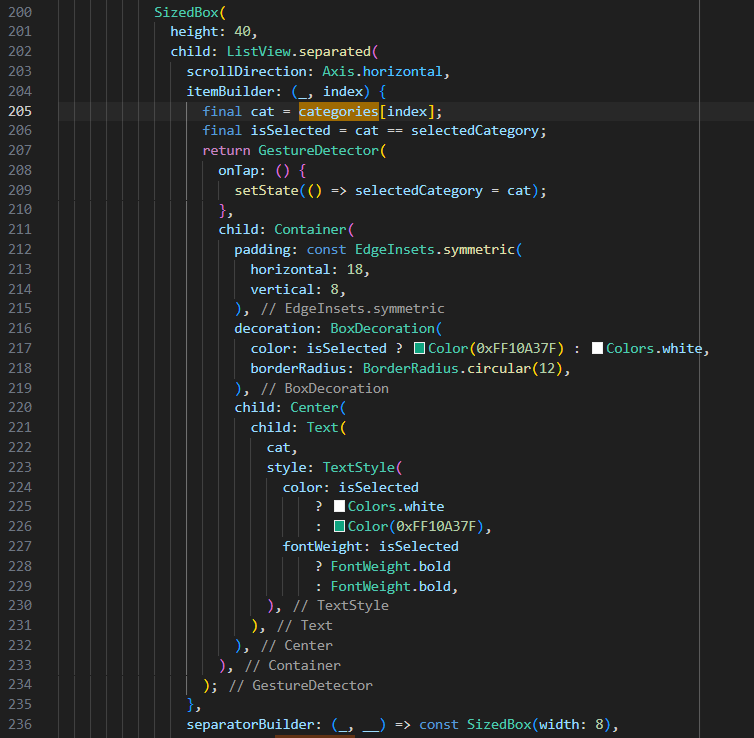


Figura 2.2 – Butoanele cu categorii

În acest fragment de cod este creat un rând de butoane orizontale care reprezintă categoriile de rețete, utilizând un ListView.separated. Acesta afișează o listă derulabilă orizontal, unde fiecare element este separat de un spațiu de 8 pixeli, adăugat prin separatorBuilder. Containerul este inclus într-un SizedBox cu înălțimea de 40 de pixeli, pentru a controla dimensiunea întregului rând. În interiorul funcției itemBuilder, se obține fiecare categorie din lista categories, iar prin variabila isSelected se verifică dacă categoria curentă este cea selectată. Fiecare element este înfășurat într-un GestureDetector, care detectează atingerea utilizatorului și, prin metoda setState, actualizează categoria selectată. Vizual, fiecare categorie este reprezentată printr-un Container cu margini rotunjite și culori diferite în funcție de selecție: verde pentru cea activă și alb pentru celelalte. Textul din interiorul butonului își schimbă, de asemenea, culoarea în alb când este selectat și în verde când nu este.



Figura 2.3 – Containerul cu scroll pentru elementele meniului

În acest fragment de cod este creat un carusel interactiv de rețete folosind PageView.builder, care permite derularea orizontală a cardurilor. Acesta este plasat într-un SizedBox cu înălțimea de 280 pixeli pentru a controla spațiul ocupat pe ecran. Proprietatea viewportFraction: 0.75 face ca fiecare pagină să ocupe 75% din lățimea ecranului, oferind un efect vizual modern cu următoarea și precedenta rețetă parțial vizibile. Fiecare card este creat dinamic din lista filtered, care conține rețetele filtrate.

Fiecare element este înconjurat de un GestureDetector, care detectează atingerea utilizatorului și, la apăsare, deschide o pagină nouă — RecipeDetailScreen — cu detaliile rețetei selectate, prin intermediul Navigator.push.

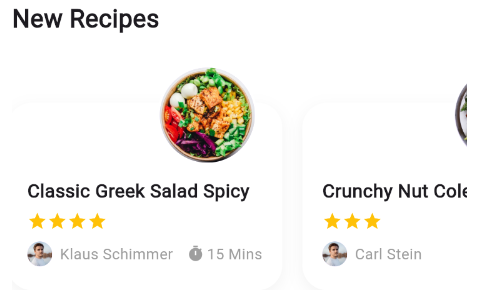


Figura 3 – Secțiunea “New Recipes”

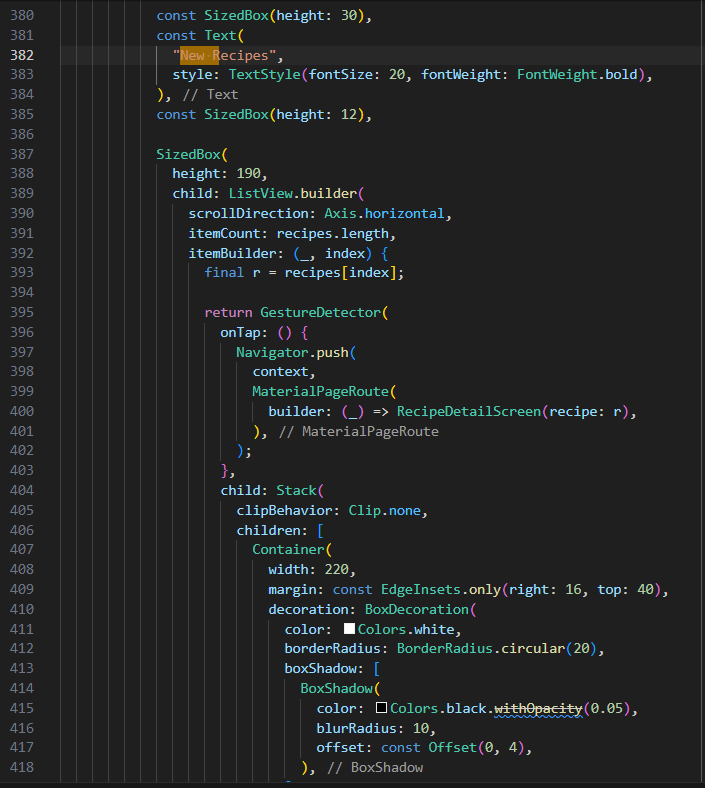


Figura 3.1 – Secțiunea “New Recipes” cu scroll container

În acest fragment de cod este construită o secțiune intitulată „New Recipes”, care afișează o listă orizontală de carduri cu rețete noi. Secțiunea începe cu un Text evidențiat printr-un stil îngroșat și o dimensiune a fontului de 20, urmat de un spațiu vertical (SizedBox(height: 12)) pentru separare vizuală. Lista de rețete este creată cu ajutorul unui ListView.builder cu derulare orizontală, ce generează dinamic fiecare element din lista recipes.

Fiecare card de rețetă este un GestureDetector, care permite navigarea către o pagină de detaliu (RecipeDetailScreen) atunci când utilizatorul atinge cardul. Cardul propriu-zis este realizat cu un Container alb, rotunjit la colțuri și cu umbră subtilă, oferind un aspect elegant.

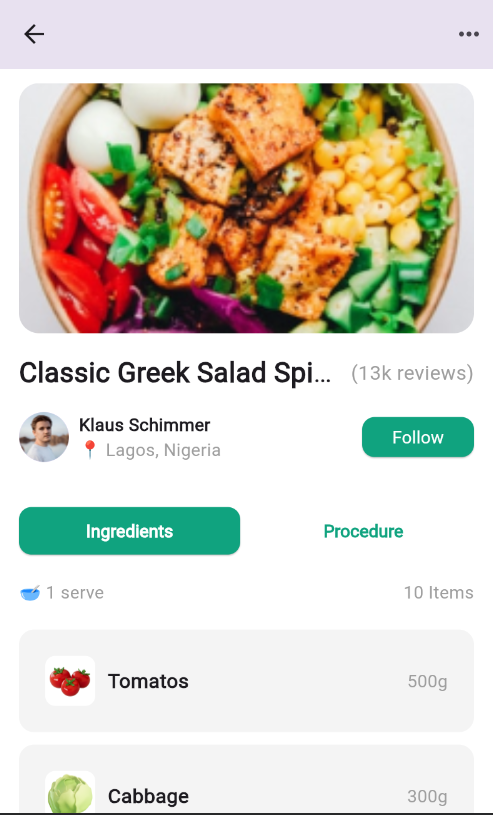


Figura 4 – Pagina cu detalii

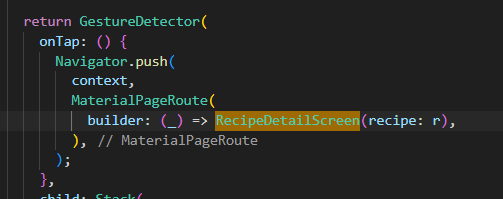
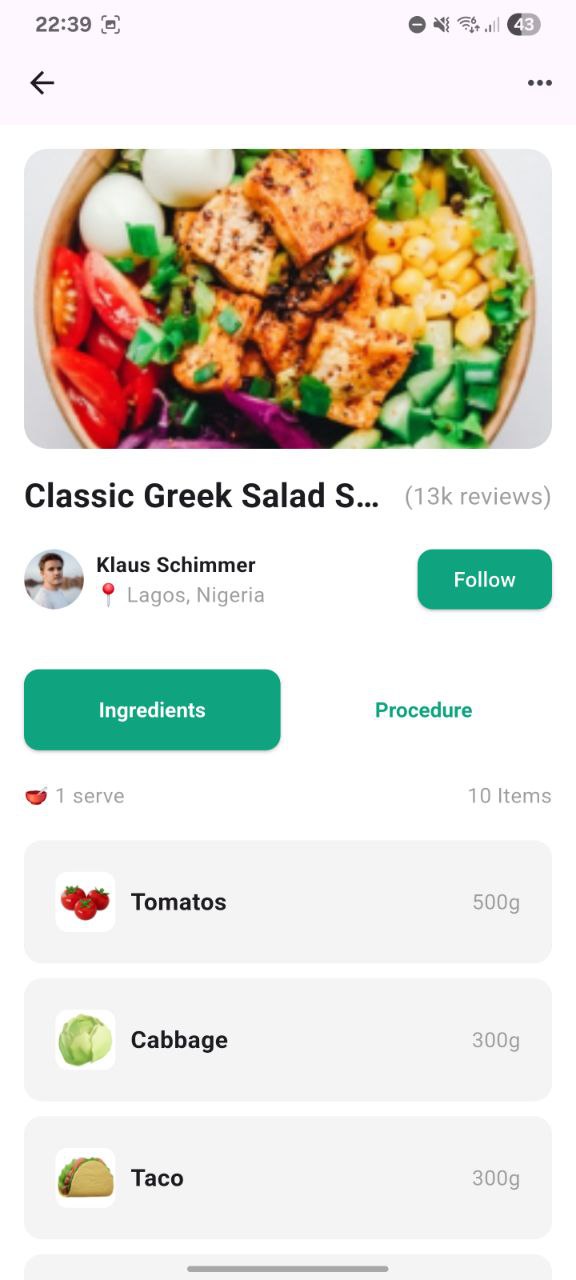
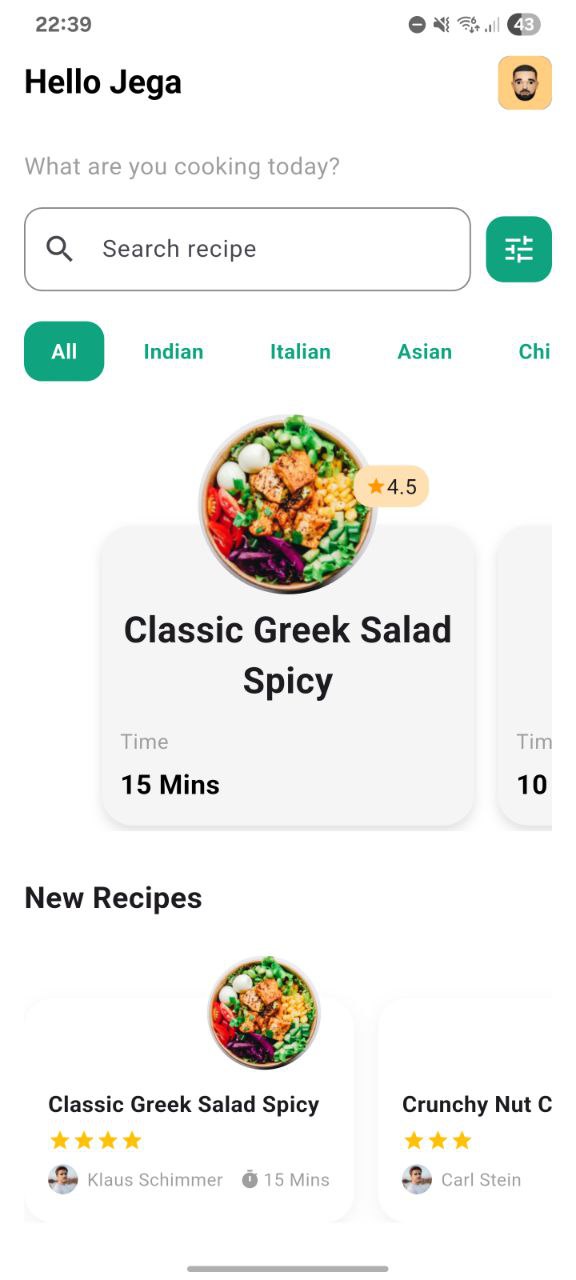


Figura 4.1 – Pagina cu detalii și acțiunea OnTap

Acest fragment de cod Flutter definește ecranul detaliat al unei rețete în aplicație, prin intermediul clasei RecipeDetailScreen, care este un StatelessWidget și primește un Map<String, dynamic> cu datele rețetei. Structura principală este oferită de Scaffold, care conține AppBar cu un buton de acțiune (more\_horiz) și un SingleChildScrollView pentru a permite derularea verticală a întregului conținut.

  
Figura 5 – Rezultatul final

**Concluzie**

În concluzie, acest laborator a demonstrat modul practic de realizare a unei aplicații mobile simple în Flutter, capabilă să efectueze conversia temperaturii între diferite unități. Prin utilizarea widgeturilor de bază, a controlului stării și a elementelor de interfață precum **TextField, RadioButton** și **ElevatedButton**, am înțeles mai bine structura unei aplicații Flutter și modul în care logica poate fi integrată cu designul pentru a oferi o experiență interactivă și intuitivă utilizatorului.