Ministerul Educaţiei, Culturii și Cercetării

Universitatea Tehnică a Moldovei

Faculatea Calculatoare, informatică și microelectronică

Departamentul Ingineria Software și Automatică



**RAPORT**

Lucrarea de laborator nr.4

*la „Programarea aplicațiilor mobile”*

*Tema:”* *Utilizare pacakge state Management. Programarea asincrona - Future. ListView. Realizare  applicațiilor dupa design definit.”*

A efectuat:

st. gr. TI-204 Cecan Valerian

A verificat:

Asist.univ Alexandru Stamatin

Chişinău 2022

**Scopul Lucrării**: Realizarea unei aplicații mobile ce rulează pe un dispozitiv mobil sau emulator utilizand Flutter framework utilizand un package state management si programare asincrona.

**Sarcina Lucrării**: Realizarea Aplicatie mobila: Online Shop - care reprezinta un screen cu lista de produse disponibile intr-un magazin online. Datele se vor incarca dintr-un fisier .json stocat in aplicatie. Sa se utilizeze un package pentru State Management.

**Implementarea Sarcinii;**

Pentru inceput am realizat functia principala pentru a rula aplicația și am inițializat clasa Myapp, Clasa MyApp este, de asemenea, un Widget creat de aplicația noastră. La fel ca orice alte widget-uri încorporate, putem de asemenea să transmitem parametrii widget-ului nostru. Acest lucru se realizează furnizând constructorul în clasa MyApp, de tip stateless,

Dupa aceasta, am setat pagina Home.



Fig.1 clasa MyApp

Pagina home consta din urmatoarele componente:

Importurile necesare functionarii,printre care si GetX. GetX este o soluție foarte ușoară și puternică pentru Flutter. Combină gestionarea stării de înaltă performanță, injecția inteligentă a dependenței și gestionarea rutelor rapid și practic. GetX permite decuplarea totală a vizualizării, a logicii de prezentare, a logicii de afaceri, a injecției de dependență și a navigării

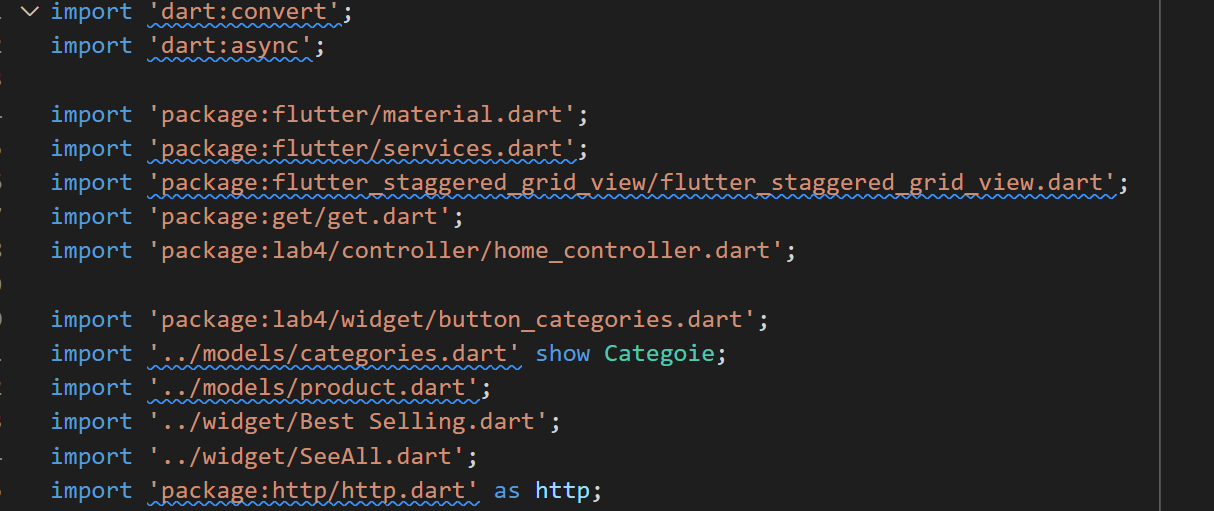


Fig.2 Importurile din fisierul home

Definim clasa Home si HomeState de tip statefull asta inseamna ca acest widget se poate schimba atunci când un utilizator interacționează cu el. Iar apoi definim Homecontroller-ul pentru aceasta pagina,el fiind intermediar intre View si Model.

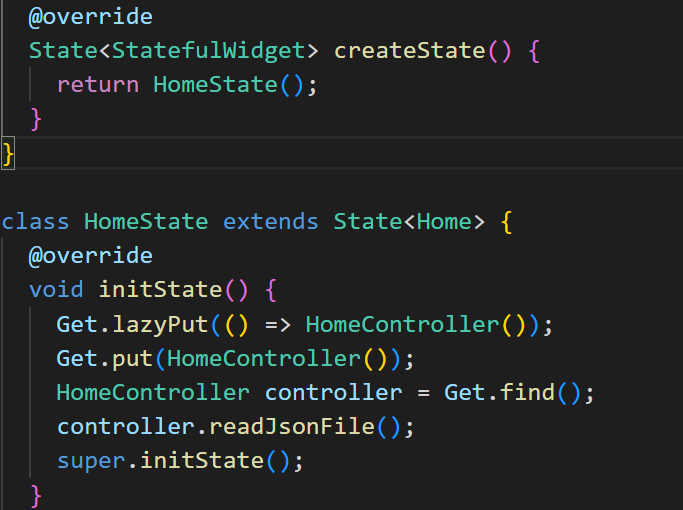


Fig.3 HomeState

Am definit widgetul build care ne returneaza clasa Scaffold

Scaffold este o clasă în flutter care oferă multe widget-uri sau putem spune API-uri precum Drawer, SnackBar, BottomNavigationBar, FloatingActionButton, AppBar etc. Scaffold se va extinde sau va ocupa întregul ecran al dispozitivului,si am definit logica pentru home controler,cum ar fi scroll-ul orizontal al categoriilor,si definirea produsel in pagina.



Fig.4 Home page build.

Definim functionalul home controllerului,indicand calea spre fisierul JSON cu informatia necesara aplicatiei pentru produse si categorii..

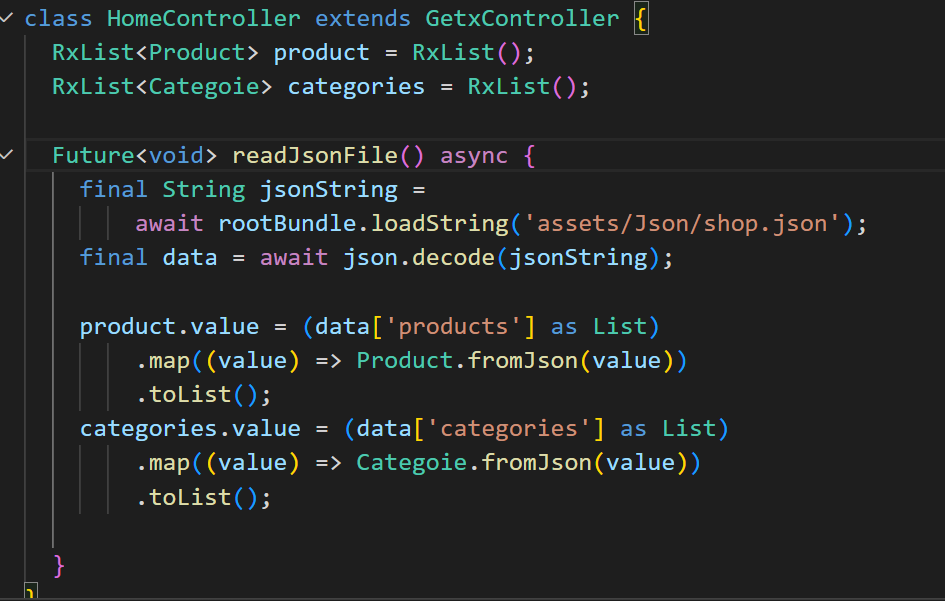
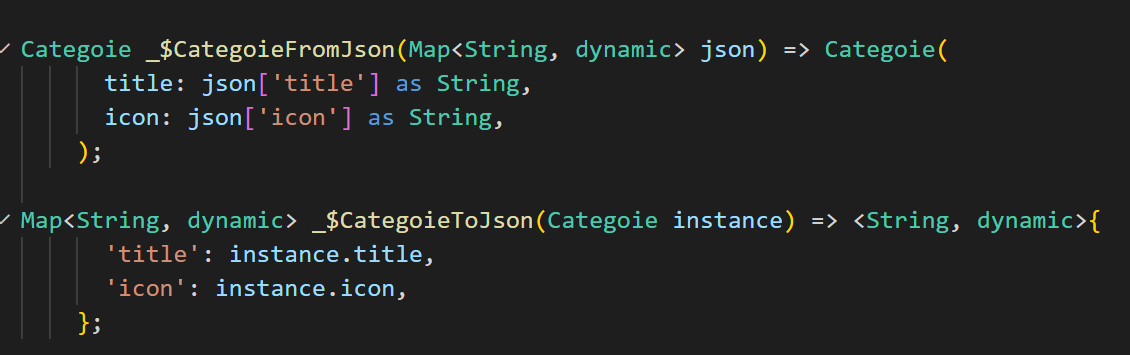


Fig.5 HomeController

Definim modelurile pentru categorii si produse,in mapa models.



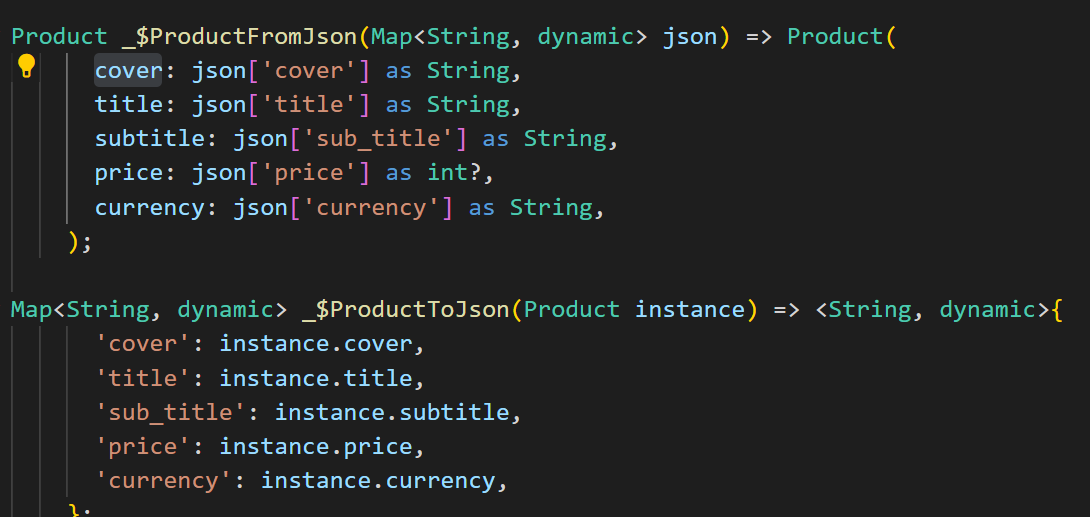


Fig 6 Definirea modelelor

Iar acum dupa definirea modelelor si controllerului a venit randul view-ului care este reprezentat de widgeturi,pentru inceput definim widgetul pentru categorii



Fig.7 widget categorii

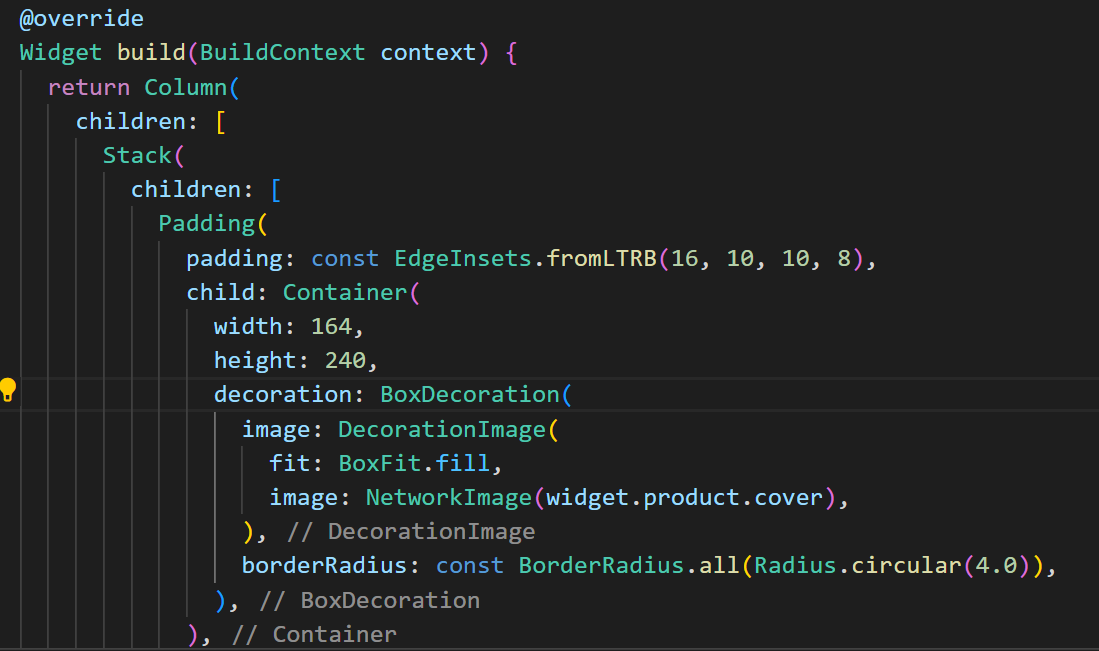


Fig.8 widget produse

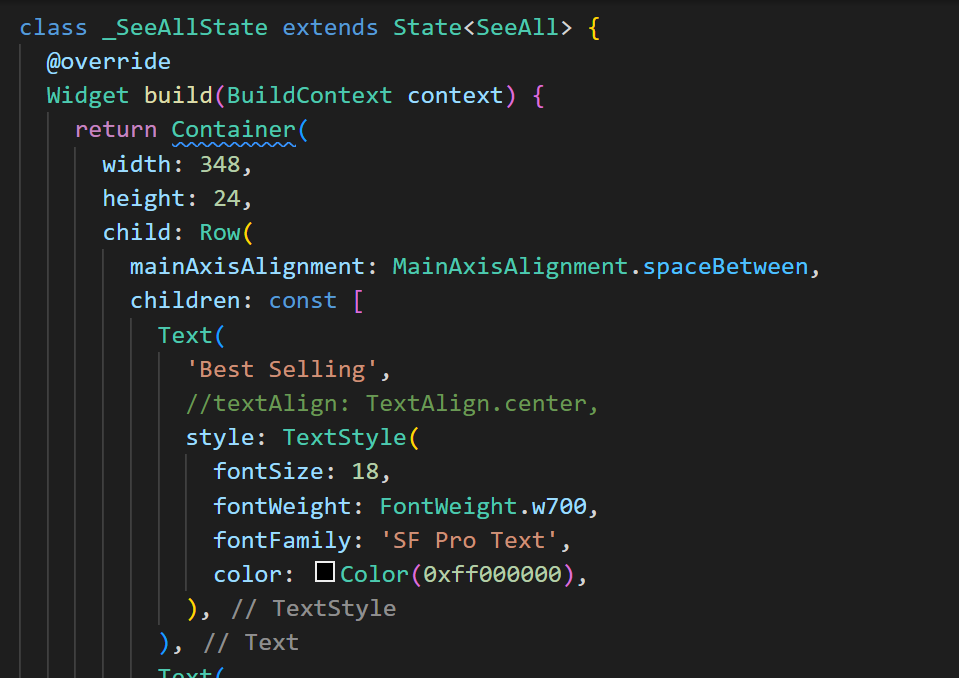


Fig.9 Widget pentru Best Selling.

**Concluzie.**

In acest laborator am realizat o aplicație cross platform, utilizânt Flutter framework, m-am facut cunoscut cu package-uri pentru state managemen, si anume GetX, rezolvările includ îmbunătățirea lizibilității codului, gestionarea modificărilor complexe și actualizări ale aplicațiilor și reducerea codului cu câteva linii utilizate pentru a executa funcționalități complexe Am îndeplinit sarcina propusa, utilizand pe langa elementele grafice ale Dart, si cele de functional, cum ar fi split și contain. Iar în final, am obținut o aplicație funcționala