Universitatea Bucuresti

Facultatea de Matematica si Informatica

Specializarea Informatica

Proiect pentru metode de dezvoltare software:

**SPT FOOD- aplicatie de food delivery**

Studenti: Avian Silviu Gabriel

Ioan Tudor Alexandru

Manole Patricia Theodora

Seria 24 Grupa 241

1. **Introducere**

Aceasta documentatie sustine proiectul care poate fi gasit la adresa: <https://github.com/Silviu1409/MDS-Project>

Proiectul a fost creeat folosind limbajul Dart, si ofera o interfata grafica ce poate fi utilizata pentru gestionarea unei firme de livrare de mancare. Fiind o aplicatie mobila pentru Android, aceasta prezinta un design minimalist, implementeaza actiuni multiple si prezinta mai multe ecrane. De asemenea, fiind lipsita de ambiguitati, pastreaza interactiuniea cu utilizatorul simpla si eficienta, in timp ce ofera siguranta pastrarii datelor si usurarea prelucrarii acestora.

1. **Prezentarea aplicatiei**

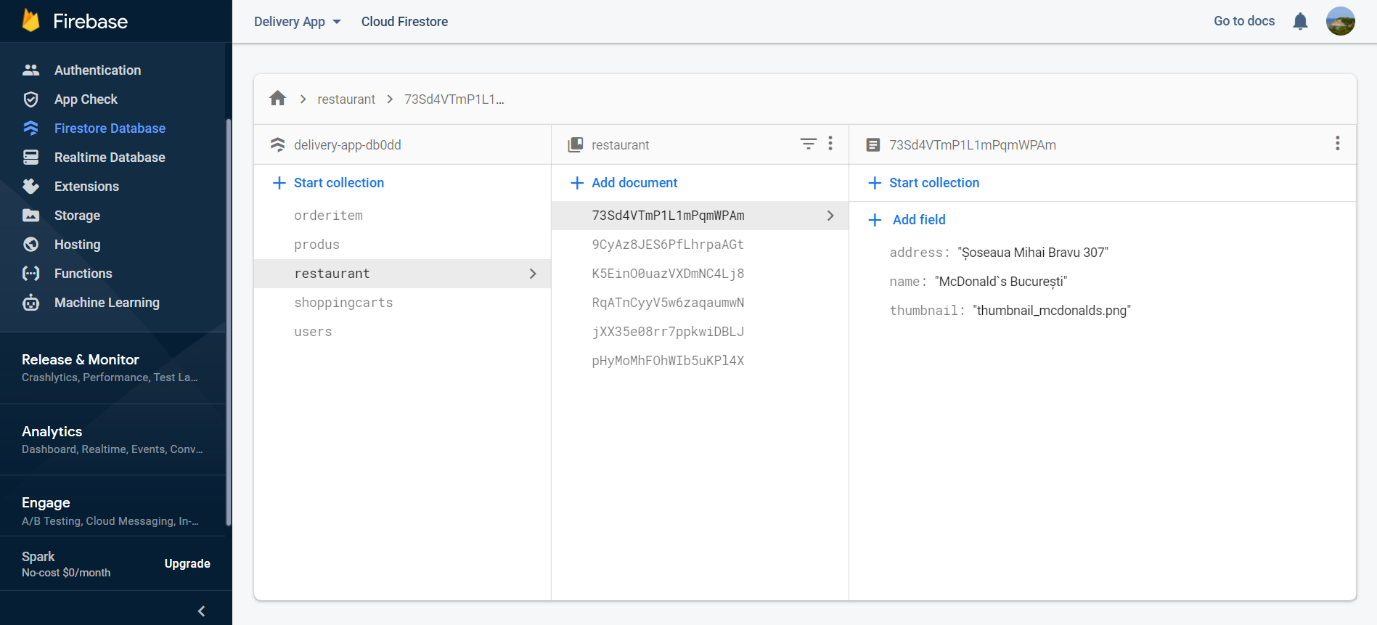
La deschiderea aplicatiei, te intalnesti cu valorile companiei noastre, urmand sa ajungi la pagina de Inregistrare/Autentificare. Odata creat contul, clientii au ocazia sa isi aleaga restaurantele preferate din lista cu acestea si sa isi comande produsele dorite. Comanda va fi gasita in cosul de cumparaturi care odata finalizata va fi transmisa catre curierii nostri si restaurantul ales. In plus, contul clientului ofera oportunitatea de a iti schimba datele alese initial la Inregistrare, impreuna cu o poza de avatar. De asemenea, interfata curierilor este una usor de folosit. Acestia gasesc pe prima pagina comenzile plasate de utilizatori, pe care le pot accepta sau refuza. Odata acceptata o comanda, aceasta apare in lista de comenzi care urmeaza sa fie livrate. Odata confirmata livrarea comenzii, aceasta dispare din lista curierului.

La deschiderea aplicatiei exista de asemenea un loading screen care tine user-ul ocupat cat timp se incarca datele din backend.



Apoi utilizatorul va trece prin 3 slide-uri care ofera informatii despre serviciile oferite de aplicatia noastra.

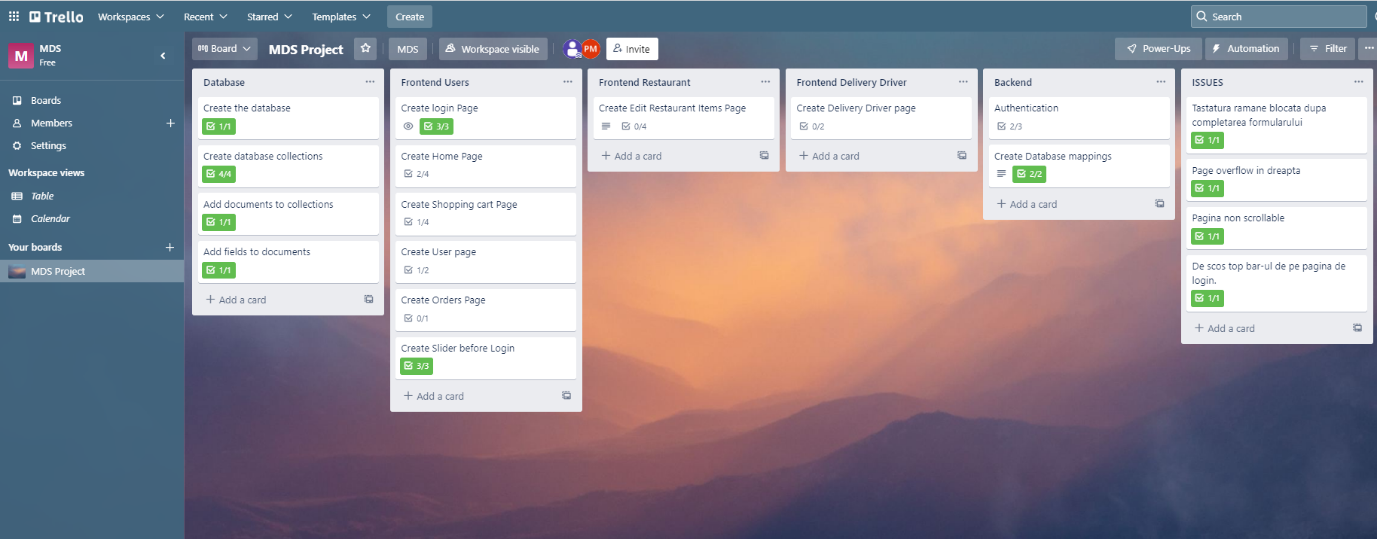


Prima interactiune propriu-zisa cu aplicatia va fi pagina de autentificare. In cazul unui utilizator nou, acesta isi va crea un cont in sectiunea de inregistrare, completand campurile necesare : email , parola, username si numar de telefon. Aceste date sunt salvate intr-o baza de date nonrelationala. Dupa crearea contului, la autentificarea utilizatorului, verificam datele cu cele salvate in baza de date si oferim accesul catre urmatoarea pagina, unde utilizatorul se va intalni interfata corespunzatoare tipului de cont: client sau curier. 

Interfata clientilor ofera o pagina cu restaurantele partenere si meniurile oferite de acestea, un cos de cumparaturi unde sunt adaugate produsele selectate si pagina contului utilizatorului. Interfata curierului ofera o pagina de start unde in coltul din stanga sus isi poate regasi avatarul, care il redirectioneaza spre pagina contului, iar pe restul paginii poate regasi comenzile care sunt in asteptare si cele pe care le-a acceptat, dar trebuie livrate.

1. **Backlog**

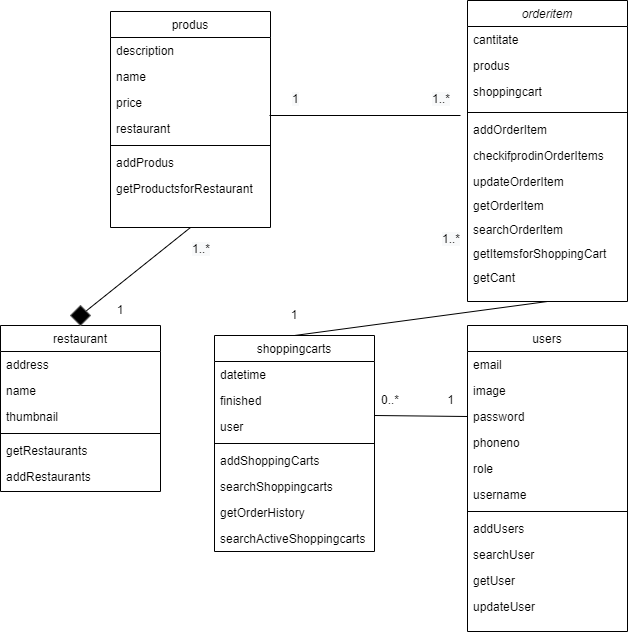
Am folosit Trello pentru a monitoriza si manageria sarcinile avute in timpul procesului de dezvoltare. Acesta poate fi gasit la <https://trello.com/b/9RWTZwVI/mds-project>.



1. **UML**

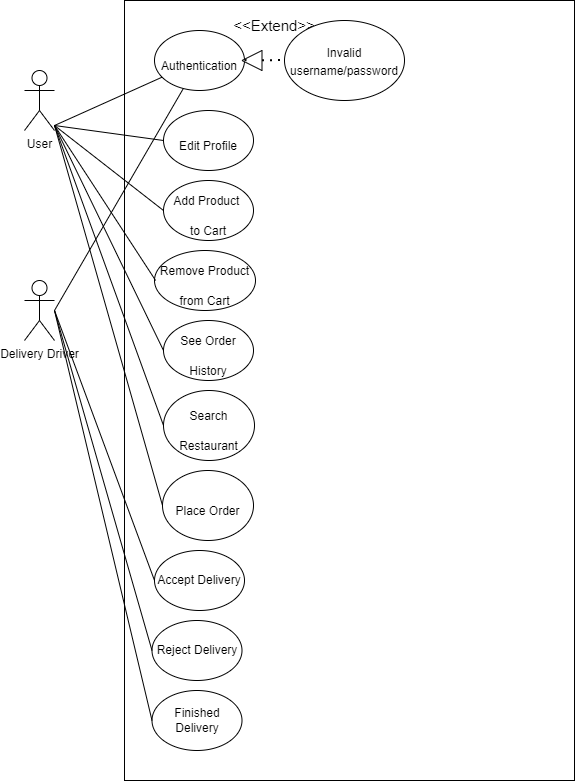
Pentru o intelegere mai buna a functionalitatii aplicatiei am creat o diagrama a claselor folosite, astfel fiind usor de vizualizat cum functioneaza in spate aplicatia noastra.

Class Diagram

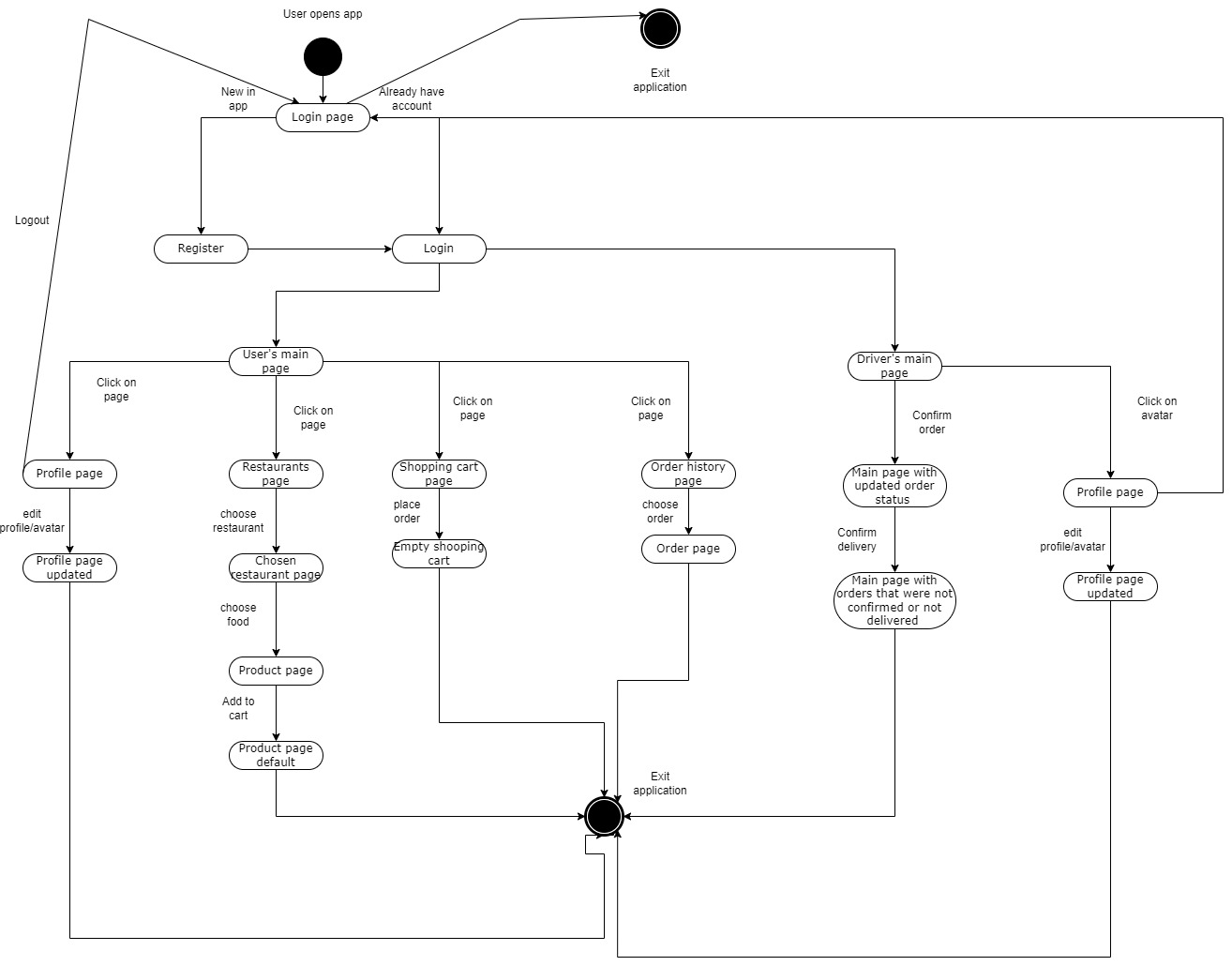


Apoi, pentru a putea prezenta modul interactiunii clientilor cu aplicatia noastra am realizat Use Case diagram, iar pentru o explicare intuitiva a folosirii acesteia State Machine Diagram in care se poate observa fiecare stare in care se afla aplicatia in timpul folosirii.

Use Case Diagram



State Machine Diagram

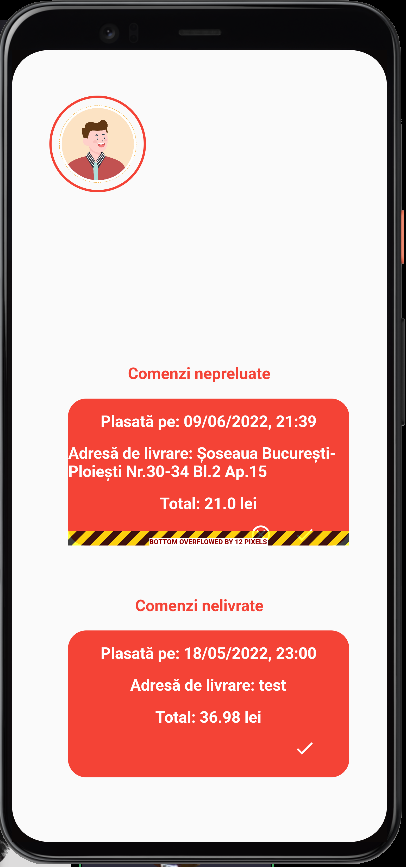


1. **Bug Report**

In timpul dezvoltarii aplicatiei ne-am confruntat cu diverse probleme. Pentru raportarea acestora am creat o sectiune separata in backlog, de „BUGS”. Acolo au fost notat diversele bug-uri aparute in timpul incercarii aplicatiei, fiind marcate ca „done” dupa rezolvarea lor. Aceasta modalitate de notare a bug-urilor ne-a ajutat sa pastram o evidenta a problemelor aparute, facilitand astfel evitarea reaparitiei acestora in urmatorii pasi de dezvoltare.

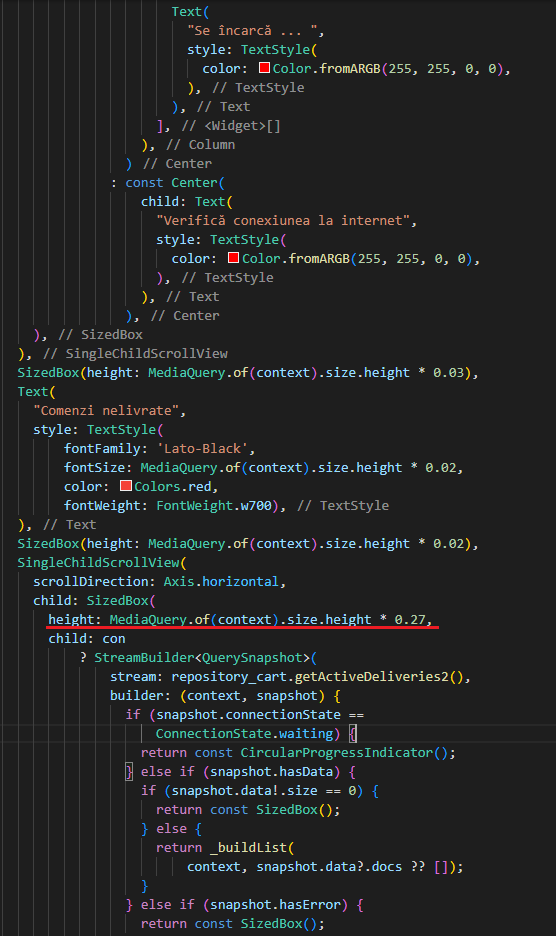
Cateva exemple de probleme gasite sunt:

1. Pe pagina principala a curierului,la comenzile nepreluate in momentul in care adresa era prea lunga aparea overflow in partea de jos casetei in care este afisata comanda pe ecran. Pentru a rezolva aceasta problema, am modificat dimensiunile casetei in care este pus textul astfel incat sa poata incadra orice adresa.

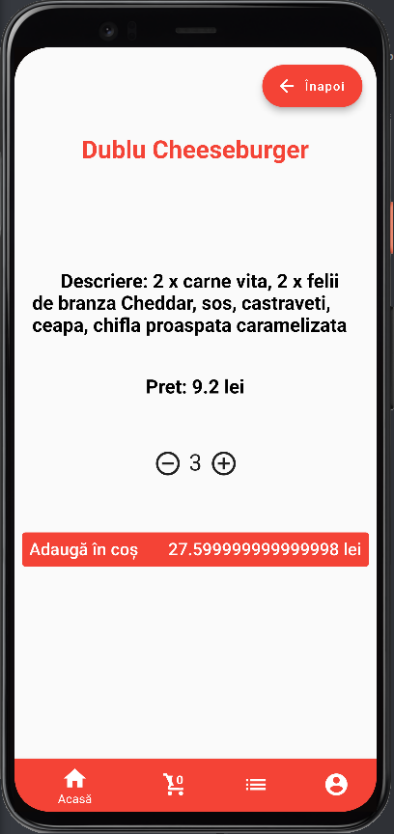
* *

Codul inainte si dupa rezolvare: A fost modificata dimensiunea casetei in raport cu dimensiunea ecranului dispozitivului.

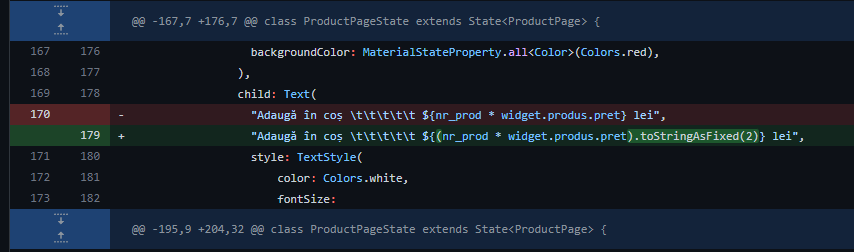
 

1. Cand se calcula valoarea cosului, cand costul produselor nu era fix, suma totala nu era cea asteptata. Astfel am aproximat rezultatul folosind o anumita precizie (in acest caz 2 zecimale).



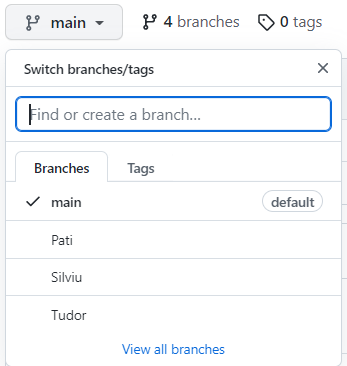
Rezolvarea se poate vedea din acest commit in care a fost adaugata functia de aproximare la 2 zecimale.



1. **Source Control**

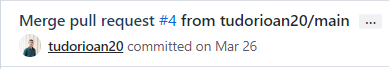
Branch Creation

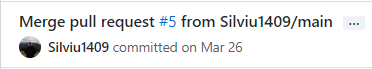
Pentru a optimiza modul de lucru la aplicatie, ne-am creat fiecare un branch separat pe care am lucrat. Ulterior, cand un user story era terminat se crea un pull request pe main. Acesta urma sa fie aprobat de cineva din echipa, codul nou fiind adaugat celui principal. Am lucrat pe tot parcursul semestrului la aceasta aplicatie, ceea ce reiese si din numarul de commit-uri.

.

Merge/Rebase

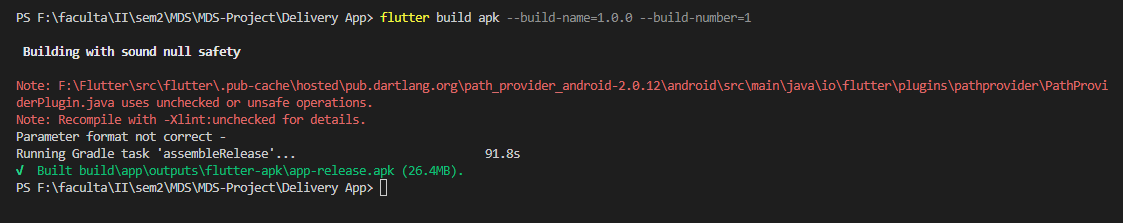


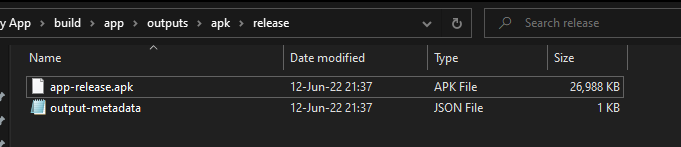




1. **Build tools**

Proiectul a fost dezvoltat folosind Visual Studio Code, iar ca build tool am folosit Flutter, transformand astfel source code-ul intr-un program executabil de tip apk, executand comanda „flutter build apk” in terminal.



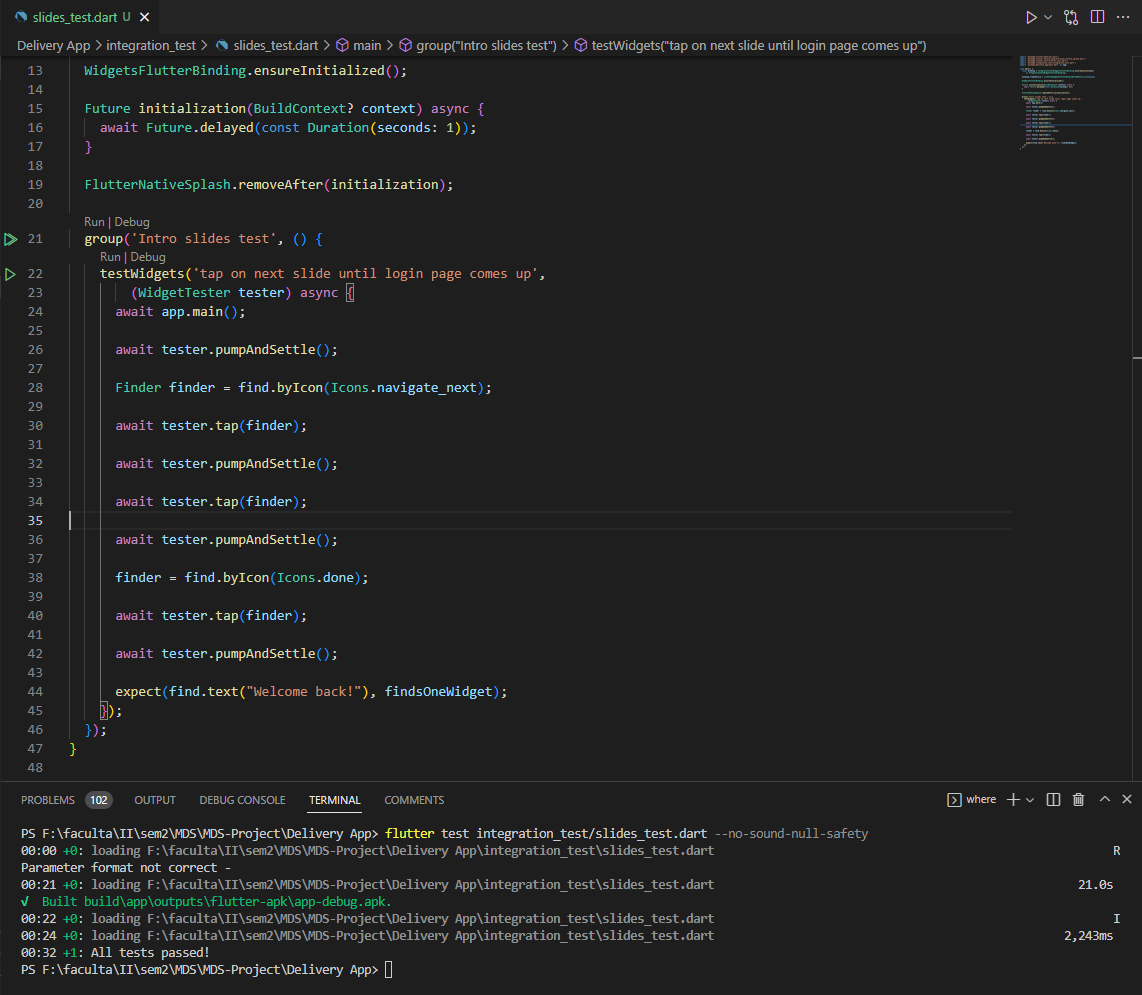


In folderul in care se afla proiectul vor fi generate 2 fisiere. Primul este fisierul apk destinat instalarii aplicatiei, iar cel de-al doilea este un mesaj care spune daca apk-ul a fost generat cu succes sau nu.

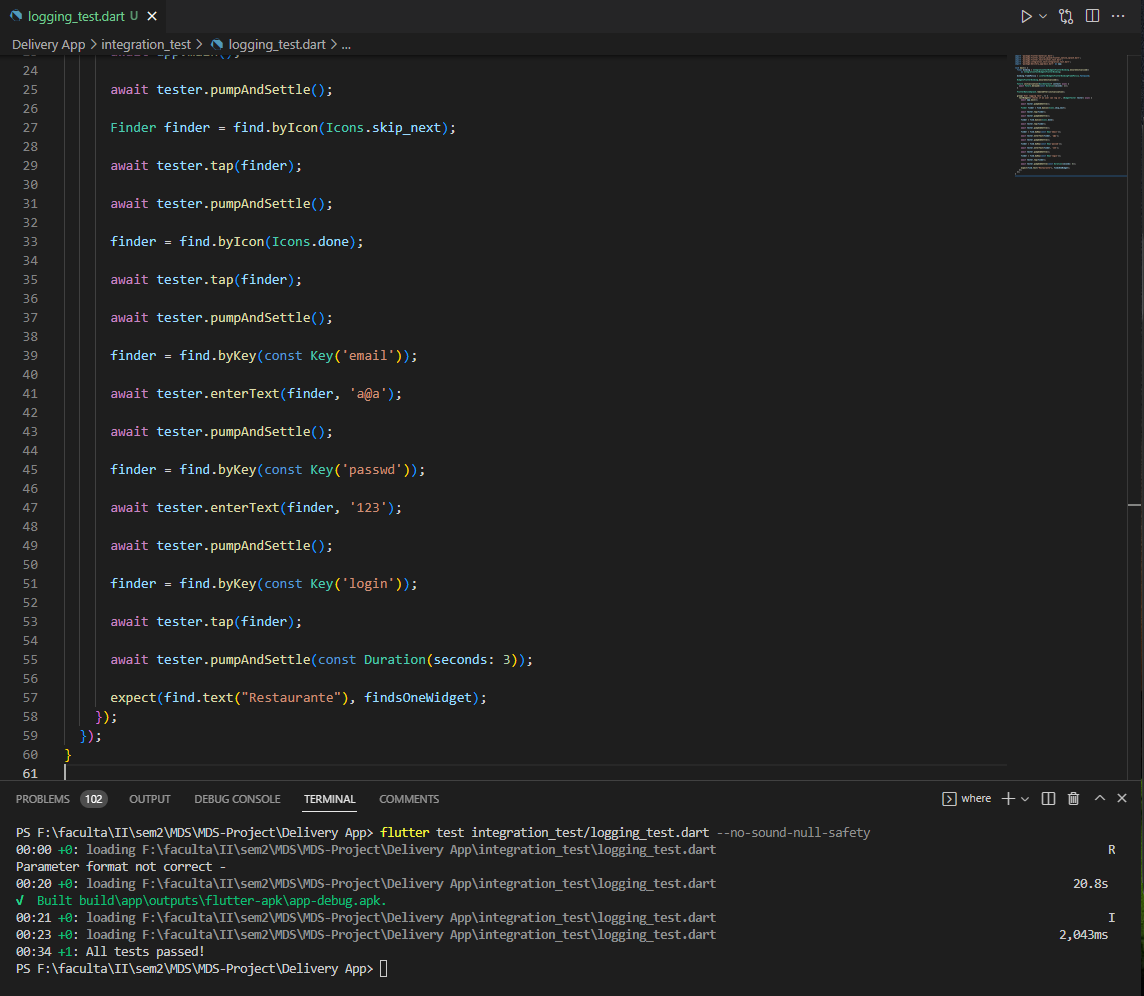
**8.Teste automate**

Pentru partea de testare automata am folosit pachetele „test” si „integration\_test”. Am facut 3 integration tests care verifica functionarea intro slide-urilor, a login-ului si a adaugarii de produse in cos.Testarea se face scriind o serie de comenzi care sa fie executate de catre un serviciu pe un device. Am utilizat 2 unit tests pentru verificarea functionalitatilor de crestere si scadere al numarul de produse si calcularea totalului din cos.

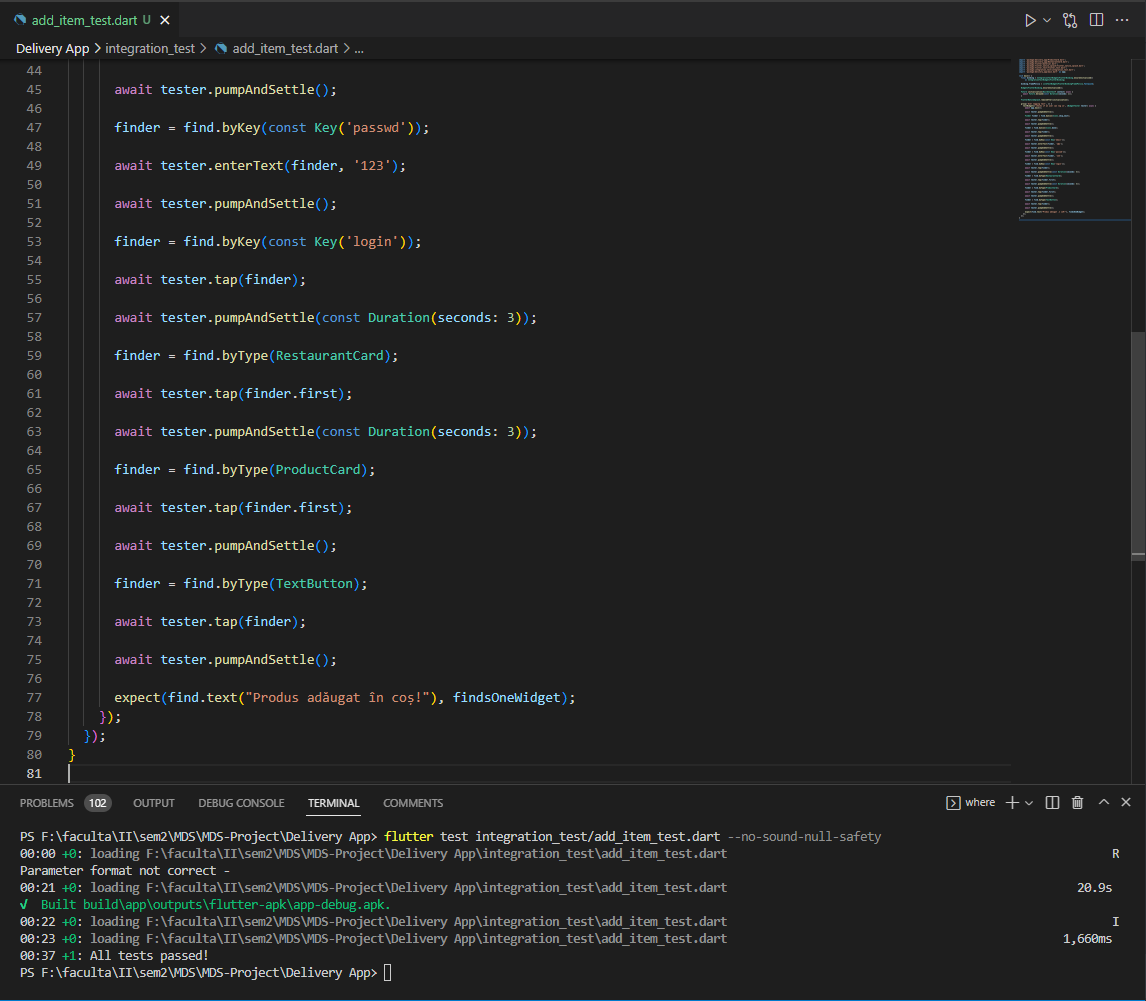
Integration test : Intro Slides



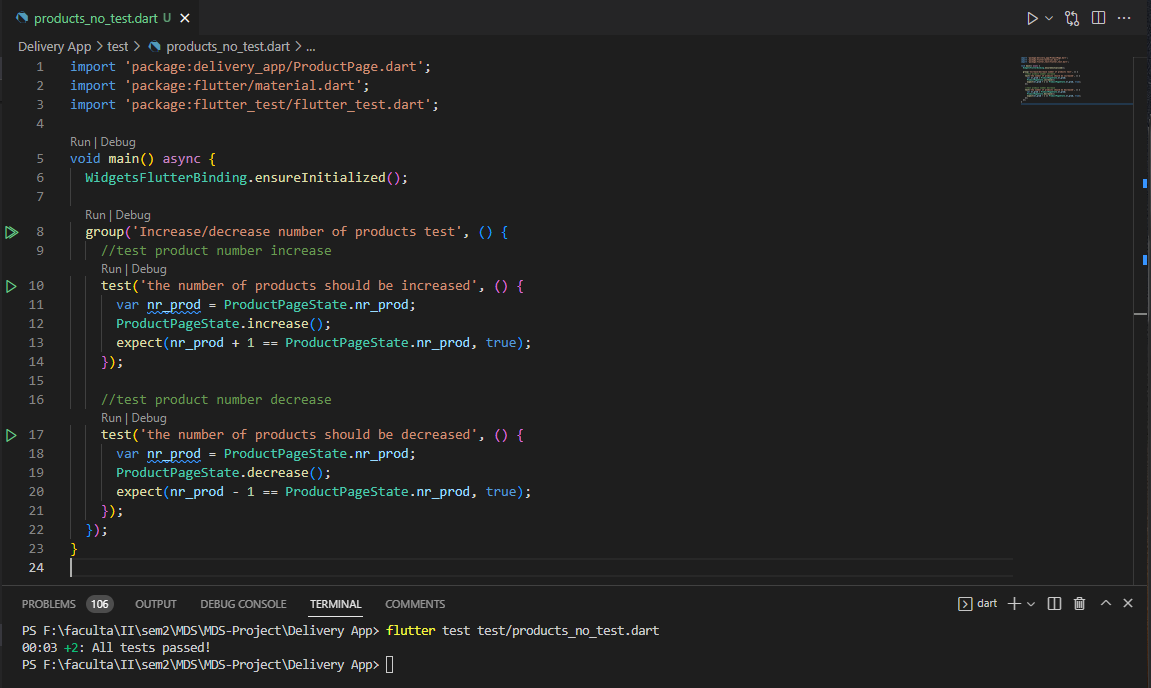
Integration test : Login



Integration test: Adauga produs in cos



Unit test: Creste/Scade numarul de produse



Unit test: Calculare total cos

