

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE



---

**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**CATEDRA CALCULATOARE**

## FUNDAMENTAL PROGRAMMING TECHNIQUES

### ASSIGNMENT 4

Food delivery management system

Student: Torzsa Gabriela

An studiu: II, semestrul II

Grupa: 30226

## CUPRINS

- I. Obiectivul temei
- II. Analiza problemei, modelare, scenarii, cazuri de utilizare
- III. Proiectare si Implementare
- IV. Rezultate
- V. Concluzii
- VI. Bibliografie

## I. Obiectivul temei

Obiectivul temei a fost sa proiectam o aplicatie care sa simuleze un restaurant cu un sistem de management a comenzilor. Aplicatia simuleaza activitatea administratorului restaurantului, a unui simplu client si a unui angajat. Administratorul se ocupa de crearea produselor in restaurant, clientul poate sa vizualizeze produsele disponibile, sa caute produse dupa anumite criterii si sa plaseze comenzi. Angajatul poate sa vizualizeze in timp real comenzile efectuate de client, in scopul inceperii prepararii acestora. Aplicatia implementeaza 3 tipuri de useri: administrator, client si angajat.

## II. Analiza problemei, modelare, scenarii, cazuri de utilizare

Aplicatia este structurata dupa diagrama prezentata in cerintele aplicatiei, cu clase noi adaugate pentru o structura a proiectului mai buna. Aplicatia este structurata in 5 pachete: business, interfaces, data, presentation si user.

1. Pachetul business contine clasele DeliveryService si UserService si pachetul interfaces
2. Pachetul data contine clasele ReportGenerator, SerializerDelivery si SerializerUser
3. Pachetul presentation contine toate clasele care se ocupa de interfata grafica cu utilizatorul si de efectele interactionarii utilizatorului cu aplicatia
4. Pachetul model contine clasele de tip model: clase pentru BaseProduct, CompositeProduct, MenuItem, Order si User

Clasa Main se afla in pachetul default.

### Scenarii / cazuri de utilizare

La momentul rularii aplicatiei, se va deschide o fereastră de Login in care utilizatorul se va putea loga sau in cazul in care acesta este un client care doreste sa realizeze o comanda si acesta nu este inregistrat in „baza de date” (intr-o lista de clienti) se va putea inregistra, iar mai apoi loga pentru a intra in aplicatie. Fereastra dedicata angajatului va fi in permanenta deschisa pentru a putea vedea tot timpul daca un client a plasat o comanda.

#### 1. Logarea ca administrator

Administratorul este unic. Acesta are un username si o parola cu care se logheaza in aplicatie. In cazul in care datele sunt introduse corect, la momentul apasarii butonului de Login, se va afisa o fereastră in care administratorul va fi instiintat ca a logarea a fost facuta cu succes si se va deschide o fereastră in care acesta va putea realiza modificari asupra starii restaurantului: adaugare si editare de produse existente, adaugarea de noi produse si stergerea de produse. La momentul logarii datele furnizate de fisierul .csv se vor importa in aplicatie.

## 2. Inregistrarea / logarea ca client

Clientii înainte de a se loga în aplicație trebuie să se înregistreze cu un username, un email și o parolă. Înregistrarea este posibilă prin apăsarea butonului de Register din fereastra de Login. În momentul apăsării butonului de Register, se va deschide o nouă fereastră, pe lângă cea de Login, în care i se va cere utilizatorului să se înregistreze. După introducerea datelor cerute, acesta trebuie să apese pe butonul de înregistrare în „baza de date”. După înregistrarea în sistem, va apărea o fereastră cu un mesaj de succes, iar utilizatorul va fi redirectionat la fereastra de login.

În momentul logării:

1. Dacă datele introduse sunt corecte, se va afișa un mesaj de succes și se va deschide o altă fereastră în care clientul poate să vizualizeze produsele din restaurant și să plaseze comenzi
2. Dacă datele introduse sunt greșite se va afișa un mesaj de eroare și i se va cere clientului să reintroducă datele corecte

Când clientul a reușit să se logheze, acesta va putea căuta produsele din restaurant prin apăsarea butonului de search. Când butonul de search este apăsător, se va deschide o nouă fereastră în care clientul va putea căuta produsele după mai multe criterii: nume, recenzie, număr de calorii, proteine, grăsimi, cantitate de sodiu sau preț. Prin apăsarea butonului de cancel, fereastra se va închide, iar clientul va fi din nou în fereastra principală. Prin apăsarea unui produs din listă, produsul se va adăuga într-o comandă, iar prin apăsarea butonului de order, comanda se va plasa. În cazul în care clientul dorește să plaseze o nouă comandă cu alte produse, va putea apăsa butonul de clear.

## 3. Înregistrare ca angajat

La începerea aplicației fereastra dedicată angajatului se va deschide și ea și va fi în permanentă deschisă pentru a putea vedea comenzile active, adică comenzile plasate de către fiecare client. Angajatul se poate loga și el în fereastra de login, dar acest lucru nu va fi necesar din cauza afișării ferestrei dedicate acestuia la începerea aplicației.

Toate ferestrele, mai puțin cea de login, au un buton de logout, pentru deconectarea userului actual, și posibilitatea logării unui alt tip de user.

## III. Proiectare și implementare

Proiectarea claselor a fost făcută după modelul dat în cerința realizării aplicației.

### Pachete și clase

#### Pachetul model

1. Pachetul conține clasele model, adică clasele pentru definirea unui produs de bază, unui produs compus, unui produs din meniu, a unei comenzi și a unui client.

- Produsul de baza, conform cerintei, contine attributele: titlu, recenzie, recenzie, numar de calorii, proteine, grasimi, cantitate de sodiu si pret.
  - Produsul compus din produse de baza are un titlu si o lista de produse de baza din care este format.
  - Produsul de tip produs meniu este reprezentat printr-o clasa abstracta care contine metoda de calcul al pretului si o metoda de obtinere a titlului produsului. Produsul de baza si cel compus extind clasa dedicata produsului de tip menuItem, adica, un produs din meniu poate fi ori un produs de baza, ori un produs compus din produse de baza. Astfel implementarile celor doua metode prezente in clasa abstracta, vor avea implementari diferite in functie de tipul de produs.
  - Comanda este reprezentata de un ID unic, un ID unic de client, de o data si ora a crearii comenzii, de o lista de produse si un pret total per comanda.
  - Userul (clientul) este caracterizat de un ID unic, de un username, un email si o parola.
- Toate clasele implementeaza metode mutatoare si accesoare pentru accesarea si modificarea atributelor.

#### Pachetul data

Pachetul contine clasele care se ocupa de serializarea si deserializarea claselor DeliveryService si UserService pentru salvarea starii aplicatiei.

1. Clasele SerializerDelivery si SerializerUser implementeaza metodele de serializare si deserializare a aplicatiei
2. Clasa ReportGenerator implementeaza metodele de creare a chitantei pentru fiecare comanda plasata de catre client si genereaza rapoartele pentru admin in functie de cerintele problemei:
  - Raport continand toate comenzile plasate intr-un interval de timp stabilit de administrator
  - Raport continand produsele care s-au comandat intr-un numar mai mare de o valoare aleasa de administrator
  - Raport continand detalii despre clientii care au plasat comenzi intr-un numar mai mare decat cel specificat de administrator si valoarea comenzii a fost mai mare decat o anumita cantitate, aleasa tot de administrator
  - Raport referitor la produsele care au fost comandate intr-o anumita zi, aleasa de administrator, si cantitatea acestora.

#### Pachetul business

Pachetul contine, pe langa clasele de management a restaurantului si a clientilor, pachetul interfaces in care am scris interfetele pentru DeliveryService si UserService. Interfetele IDeliveryServiceProcessing si IUserService contin antetul metodelor ce vor fi implementate ulterior in clasele care implementeaza interfetele.

## Clasa DeliveryService

Clasa are ca atribut o structura de tip `Map<Order, List<MenuItem>>` pentru stocarea informatiilor referitoare la comenzi si o structura de tip `ArrayList<MenuItem>` care va salva meniul (produsele) puse la dispozitie de catre restaurant.

### Metodele clasei

- `isWellFormed()` - verifica daca lista de comenzi si lista de produse din meniu sunt nule sau nu
- `findProduct(...)` - se va cauta in lista de `menuItems` produsul cu titlul egal cu cel dat ca parametru si il va returna

#### Aministrator:

- `importProducts()` (override din interfata) – contine o lista de produse de tipul `MenuItem` in care se importa din fisierul `products.csv` datele (produsele cu toate informatiile acestora). Fisierul contine duplicate, dar aplicatia va importa primul produs gasit cu acelasi nume.
- `addProduct(...)` – metoda care se ocupa de adaugarea / editarea unui produs. Cand administratorul apasa pe titlul unui produs din lista de produse, fieldurile care caracterizeaza produsul se vor auto-completa cu detaliile produsului. In cazul in care titlul produsului ramane neschimbat si administratorul modifica oricare din celelalte fielduri, la apasarea butonului de add/edit se va edita produsul existent in meniu, iar in cazul in care administratorul modifica si titlul produsului, la apasarea butonului se va adauga un produs nou cu noile caracteristici.
- `deleteProduct(...)` – cand administratorul alege un produs din lista de produse, iar apoi apasa butonul de delete, produsul respectiv se va sterge.

#### Client:

- `addOrder(...)` – metoda calculeaza pretul total al produselor adaugate de client in comanda si va seta pentru comanda respectiva un ID unic (care va creste cu unu la fiecare adaugare de comanda noua), ID-ul clientului care a realizat comanda si data, respectiv ora plasarii comenzii. Comanda se va adauga in lista de comenzi declarata ca atribut al clasei.
- `generateBill(...)` – in cazul in care comanda s-a plasat, se va apela metoda de generare chitanta din clasa `ReportGenerator` si se va crea o chitanta in format `.txt` continand ID-ul comenzii, data si ora plasarii comenzii, lista de produse comandate si pretul total.

```
bill2.txt x
1 Order 2
2 Date/time: Thu May 27 13:50:42 EEST 2021
3 Products:
4 BaseProduct title = 'Ethiopian Spice Tea ', rating = 5.0, calories = 23, protein = 1, fat = 0, sodium = 13, price = 49
5 BaseProduct title = 'Spicy Pickled Shallots ', rating = 0.0, calories = 23, protein = 1, fat = 0, sodium = 162, price = 78
6 Total price: 127
```

## Clasa UserService

Clasa are ca atribut o lista de useri (clienti) in care se vor stoca clientii noi inregistrati si o variabila in care se va retine ID-ul ultimului client adaugat, pentru unicitatea ID-urilor clientilor. Clasa contine un constructor in care se creaza lista de clienti.

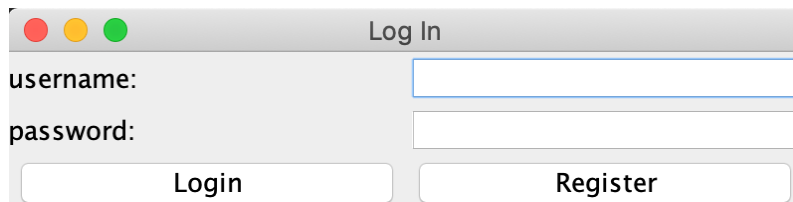
### Metodele clasei

- addUser(...) – seteaza ID-ul unic al noului client inregistrat si il adauga in lista de clienti
- findByUsername(...) – cauta in lista de clienti daca exista clientul cu username-ul egal cu cel dat ca parametru la metoda si in cazul in care exista in returneaza
- deleteUser(...) – sterge clientul din lista de clienti

## Pachetul presentation

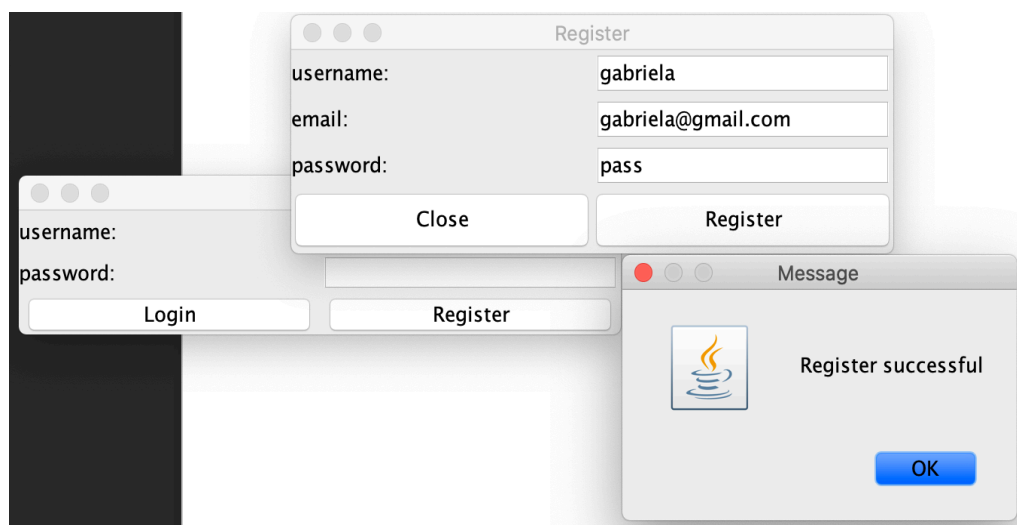
Pachetul contine clasele care implementeaza toate ferestrele din aplicatie.

### Clasa LoginGUI

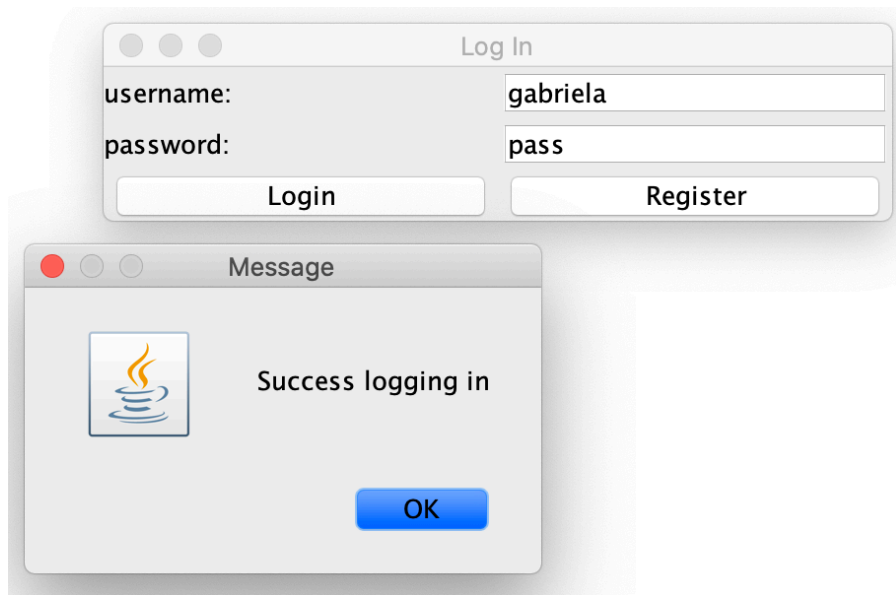


Reprezinta interfata grafica destinata logarii in aplicatia restaurantului. Fereastra se va deschide odata cu rularea aplicatiei. In functie de tipul de utilizatori, se vor completa casetele de text si se vor deschide ferestrele aferente. Cand se apasa butonul de exit al ferestrei, toate datele care au fost introduse si modificarile facute vor fi serializate.

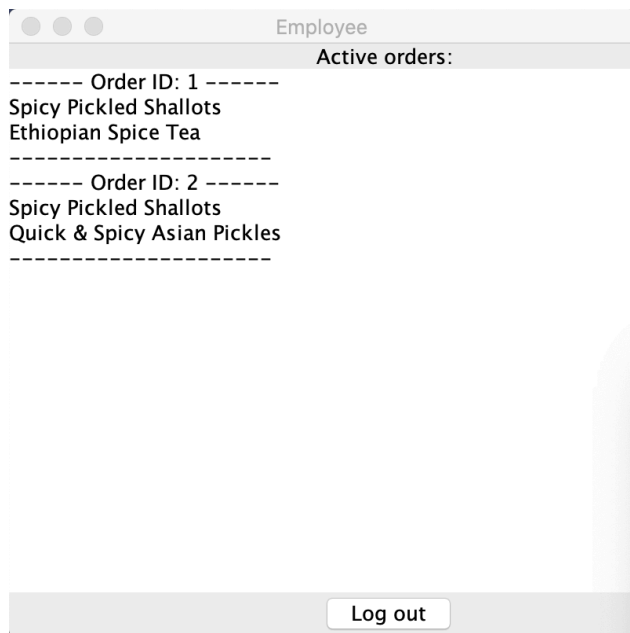
### Clasa RegisterGUI



Reprezinta interfata grafica pentru inregistrarea unui nou client in aplicatie. Dupa apasarea butonului de register din fereastra Register, noul client va fi adaugat in lista de clienti si se va putea loga ulterior in aplicatie.



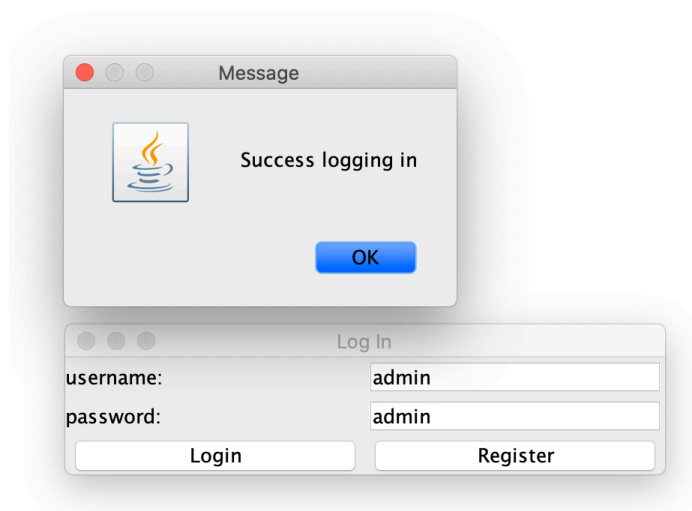
### Clasa EmployeeGUI



Reprezinta interfata grafica destinata angajatului, care la momentul plasarii unor noi comenzi de catre clienti, va putea vedea comenziile si continutul acestora. Fereastra va fi vizibila de la inceputul rularii aplicatiei.



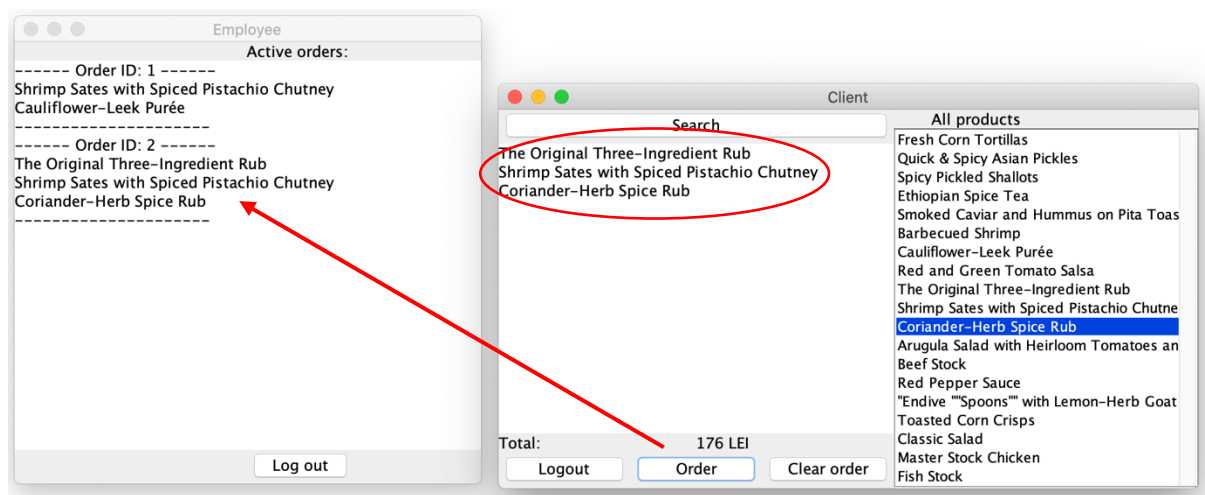
## Clasa AdministratorGUI



Reprezinta interfata grafica dedicata administratorului. Dupa logarea in aplicatie, daca administratorul apasa pe un produs din lista de produse, toate caracteristicile produsului selectat se vor scrie in casetele text. Administratorul va putea modifica datele din casute, astfel adaugand sau editand un produs existent sau sterge produsul selectat.



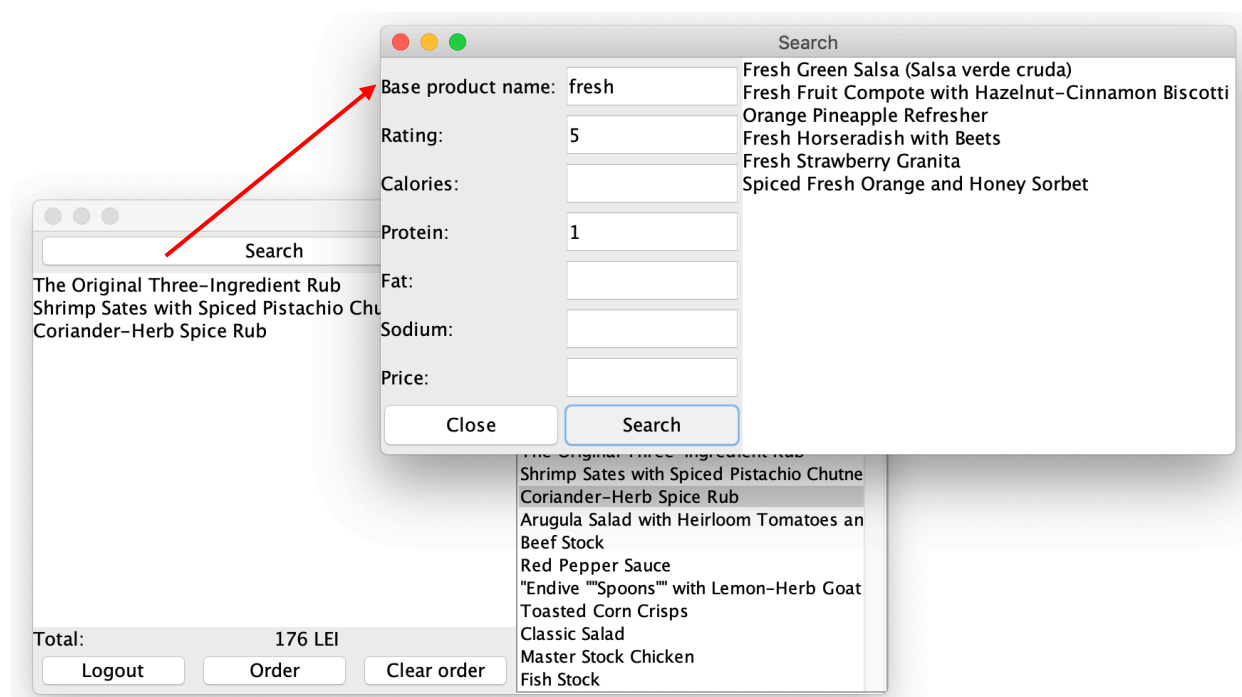
## Clasa ClientGUI



Reprezinta interfata grafica cu care va interactiona clientul. Clientul va putea selecta un produs din lista de produse pentru a-l adauga in comanda. La apasarea butonului de Order, comanda se va plasa si chitanta aferenta comenzii se va genera. Pentru stergerea listei de produse care

urmeaza sa fie comandate, trebuie apasat butonul de Clear order, astfel clientul logat va putea plasa o noua comanda, cu alte produse selectate.

Clientul poate sa caute produse dupa anumite criterii, apasand butonul de Search. In acest caz, se va deschide o noua fereastră, implementata in clasa SearchGUI.



Clientul poate cauta anumite produse din meniu, in functie de diverse criterii: titlu, recenzii, calorii, proteine, grasimi, sodiu, si pret. In functie de valorile introduse, se vor filtra produsele din lista totala de produse si se vor afisa doar cele care indeplinesc criteriile cerute de catre client. In momentul in care clientul doreste sa inchida fereastra destinata cautarii produselor, va apasa pe Close, iar fereastra se va inchide, acesta revenind la fereastra principala.

Prin apasarea butonului de Logout din toate ferestrele care il contin, userul logat la momentul respectiv va fi delogat, iar fereastra de Login va aparea. Astfel, un alt utilizator se va putea loga, respectiv inregistra.

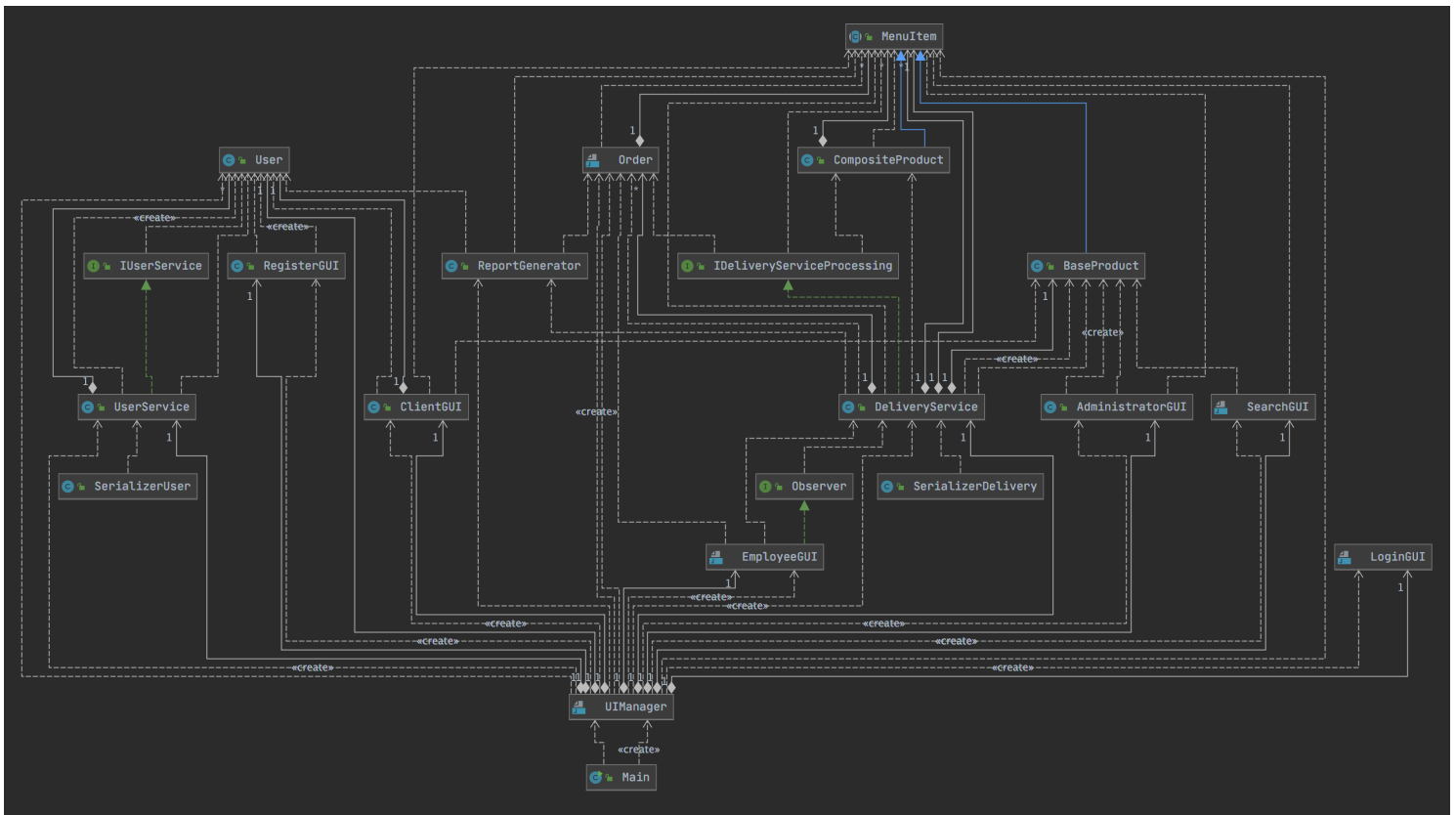
### Clasa UIManger

Clasa contine un constructor in care se creaza o instanta de DeliveryService si una de UserService, se seteaza ascultatorii tuturor butoanelor din aplicatie si se afiseaza ferestrele de Login si EmployeeScreen.

### Pachetul Default

Clasa Main – se creaza o instanta de UIManager

## Diagrama UML



## IV. Rezultate

Pentru testarea aplicatiei m-am folosit de *System.out.println()*. In final, am testat toate functionalitatile interactionand cu interfețele grafice ale aplicatiei, introducand clienti, plasand comenzi, cautat produse dupa anumite criterii, adaugand, editand si stergand produse ca administrator.

## V. Concluzii

A fost nevoie de foarte mult search pe google pentru a rezolva erorile intalnite pe parcurs, dar am invatat cum se folosesc streamurile si expresiile lambda.

## VI. Bibliografie

Tot internetul.