

DOCUMENTATIE

TEMA 4

NUME STUDENT: Rus Alexandru-Marian
GRUPA: 30227

CUPRINS

1. Obiectivul temei.....	3
2. Analiza problemei, modelare, scenarii, cazuri de utilizare	3
3. Proiectare.....	3
4. Implementare	4
5. Rezultate.....	6
6. Concluzii	7
7. Bibliografie.....	7

1. Obiectivul temei

Obiectivul principal al temei îl reprezintă designul și implementarea unei aplicații de tipul „Food delivery” având funcționalități, atât pentru clienți, cât și pentru manager. Aplicația este utilizată în scopul generării comenzilor de preparate și administrării meniului.

Obiectivele secundare care duc la îndeplinirea obiectivului principal sunt: analiza problemei și identificarea cerințelor, design-ul aplicației, precum și implementarea aplicației și totodată testarea acesteia.

1. Analiza problemei, modelare, scenarii, cazuri de utilizare

Cliantul își va utiliza credențialele (nume, parolă) pentru a se loga în aplicație. De asemenea, există și opțiunea înregistrării unui cont dacă este nou pe platformă. În cadrul interfeței grafice, acesta poate vizualiza meniul furnizat de către administrator (produse de bază și compuse) și își poate alege în funcție de preferințe, prin filtrarea elementelor preparatele dorite.

Principalele filtre de selecție sunt pe baza unor caracteristici specifice fiecărui produs: denumire, preț, valori nutriționale.

În plus, acțiunea de bază a clientului este cea de creare de comenzi. Astfel, acest proces se desfășoară intuitiv: se selectează din meniu produsul dorit, se adaugă automat într-o listă ce poate fi vizualizată în timp real (cosul clientului), alături de prețul calculat în prealabil.

În urma depunerii unei noi comenzi se vor reține data, ora, numele clientului, lista alimentelor selectate, precum și prețul. Aceste informații se vor folosi ulterior de către administrator în scopul generării unor rapoarte.

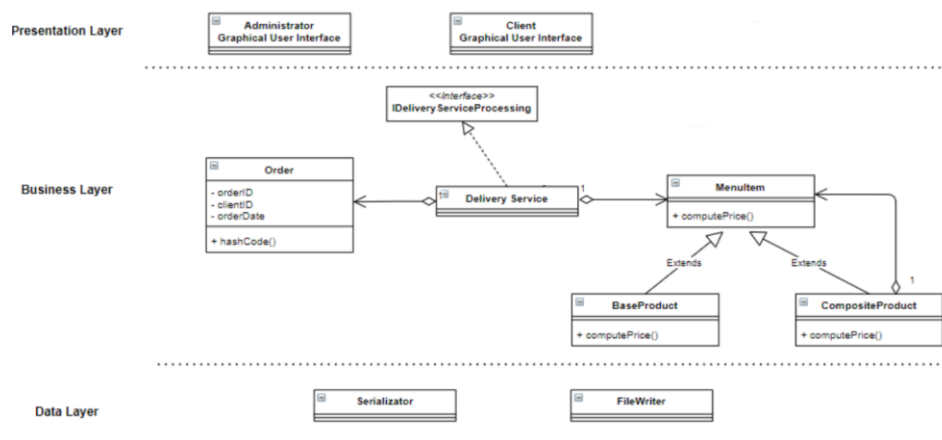
Interfața grafică pentru administrator permite atât introducerea produselor de bază, cât și crearea unor produse compuse din cele de bază. De exemplu: meniul zilei care poate conține un fel principal, o garnitură și un desert.

Administratorul poate modifica produsele existente indiferent de categoria lor. Rapoartele generate analizează performanța aplicației, astfel se monitorizează produsele comandate cel mai frecvent, orele de vârf, cele mai scumpe comenzi, alături de clienții care le-au executat.

2. Proiectare

Pentru proiectare am utilizat: JFrame- interfața grafică, ArrayList, HashSet- stocarea produselor și clienților înregistrați, HashMap- salvarea comenzilor, FileWriter- scrierea rapoartelor în fișier .txt, BufferedReader- generarea produselor din fișier, LambdaExpression și Stream- funcționalitatea de căutare a produselor și generarea rapoartelor.

Clienții, meniul și comenzile sunt salvate într-un fișier binar prin serializare, astfel se pot utiliza la viitoare execuții de program.



3. Implementare

Clasa ADMIN contine constructorul clasei cu 2 parametri- nume si parola si metoda de afisare toString

Clasa AdminGUI este alcatuita din toate elementele grafice ale interfetei pentru administrator (etichete, campuri text, butoane si tabele).

1. Metoda setUpProductTable initiaza un nou tabel cu produsele extrase din fisierul .csv. Am utilizat variabila „pushed” pentru ca butonul ce actioneaza metoda sa execute importarea produselor o singura data.
2. Metoda deleteRow se ocupa cu stergerea randului selectat din tabel, alaturi de alimentul regasit la acea pozitie.
3. Metoda addBaseElement are rolul de a adauga un nou produs de baza. Aceasta foloseste informatia introdusa de administrator legata de specificatiile produsului (proteine, grasimi. Sodiu etc.)
4. Metoda baseProductEdit editeaza produsul selectat din tabel prin completarea directa in coloanele acestuia.
5. Cele doua metode refreshTable si refreshComposedTable au rolul de a actualiza informatia din tabel in urma modificarilor anterioare.
6. Metodele pentru manipularea produselor de baza folosesc aceleasi principii ca si pentru produsele compuse.
7. Metoda hourReport extrage din interfata ora de start si de sfarsit si afiseaza in fisier toate comenzile executate in acel interval orar.
8. Metoda productReport afiseaza in fisiera toate produsele comandate de mai mult de un numar specificat de ori.
9. Metoda clientsReport prezinta toti clientii care au comandat de un anumit numar de ori, cu un pret mai mare sau egal cu cel prezentat in interfata.

10. Metoda `dateReport` afiseaza produsele comandate intr-o anumita zi de un anumit numar de ori.

Clasa `AdminControl` contine un obiect de tipul `AdminGUI`, un obiect de tipul „`DeliveryService`”. Aceasta are rolul de a deschide interfata Administrator si de a implementa toti ascultatorii pentru butoane

Clasa `BaseProduct` contine produsele de baza. Aceasta mosteneste clasa `MenuItem` si implementeaza `Serializable`. Campurile acestei clase sunt: titlul(mostenit din `MenuItem`), rating, calorii, grasimi, proteine, sodiu si pret. Clasa este alcatuita din metode de tipul: setter si getter si metoda `toString`.

Clasa `Client` evidentiaza campurile fiecarui client, nume si parola.. Ea implementeaza `Serializable`.

Clasa `ClientGUI` contine toate elementele pentru interfata grafica a clientului. Principalele metode sunt:

1. Metoda `viewTables` actualizeaza atat tabelul cu produsele de baza, cat si el cu produse compuse.
2. Metoda `searchTables` aplica functionalitatea de filtrare a produselor utilizand informatiile introduse de client pentru a afisa meniul dorit. Astfel, tabelele se reinitializeaza dupa o apasare de buton.
3. Metoda `addProductOrder` genereaza lista produselor pentru comanda accesate prin selectia directa din tabel si actualizeaza dupa fiecare selectie pretul comenzii.
4. Metoda `makeOrder` preia informatia generata de `addProductOrder` si executa noua comanda a clientului.

Clasa `ClientControl` deschide interfata pentru client si implementeaza toti ascultatorii pentru butoane.

Clasa `CompositeProduct` mosteneste `MenuItem` si implementeaza `Serializable`. Aceasta contine pe langa titlu, o lista de produse de baza.

1. Metoda `getComposedTitle` concateneaza denumirile fiecarui produs din lista. Se va folosi pentru afisare in tabele.
2. Metoda `computePrice` insumeaza pretul total accesand fiecare pret al produselor din lista.

Clasa `DeliveryService` contribuie la actiunea fiecarei componente a programului. Aceasta implementeaza `Serializable` si interfata `IDeliveryService` in care sunt prezente metodele specifice actiunii administratorului. Ea contine o lista de clienti, meniul(`HashSet` de `MenuItem`) si comenzile (`Map<Order, List<MenuItem>>`). De asemenea, contine matricea de tip `Object` care se va transmite pentru scrierea in tabel (informatiile despre produse). La sfaritul apelului de constructor, dupa ce se initializeaza fiecare set, se apeleaza metoda `deserialize` care extrage informatiile despre clienti, produse si clienti din fisierul binar.

1. Metoda `serialize` salveaza in fisierul binar clientii, produsele, comenzile.

2. Metoda `importProducts` foloseste un obiect al clasei `FileOperator` pentru a face legatura cu fisierul `.csv` in care se afla produsele de baza pe care le adauga in meniu.

Fiecare functionalitate de stergere, cautare, editare si adaugare din interfete folosesc metodele implementate in clasa `DeliveryService`. Astfel, `LoginControl`, `AdminControl`, `ClientControl` au acelasi obiect al clasei `DeliveryService` care apeleaza metodele: `deleteProduct`, `addbaseProduct`, `editBaseProduct` etc.

3. Metoda `getData` furnizeaza matrita de tip `Object` care se va folosi in tabele. Metodele pentru generarea rapoartelor si filtrarea continutului pentru client folosesc `LambdaExpression` si `StreamProcessing`.

Clasa `FileOperator` se ocupa de citirea produselor din fisierul `.csv`. Se declara calea directa catre fisier si se actioneaza citirea folosind un `bufferedReader`.

In fisierul de rapoarte se scrie utilizand campul de tipul `fileWriter`.

Interfata de autentificare este descrisa in clasa `LoginGUI` care e controlata de `LoginControl`. Astfel, pentru logare se verifica credentialele si se acceseaza interfata de client sau administrator, pentru noii clienti este posibila inregistrarea tot din `LoginGUI` actionata de `LoginControl`.

Din clasa `MainClass` se incepe rularea programului prin declararea unui nou obiect de tipul `LoginControl` si actionarea metodei `start`.

Clasa `MenuItem` implementeaza `Serializable` si este clasa „tata” pentru `BaseProduct` si `CompositeProduct`. Aceasta contine doar titlul produsului. In functie de acest titlu se suprascrisce metoda `equals` si `hashCode`, astfel in meniu nu vor aparea duplicate (produse cu denumiri identice).

Clasa `Order` contine informatiile unei comenzi: `id`, numele clientului, data si ora si pretul. Aceasta implementeaza `Serializable` si reprezinta tipul cheii pentru `HashMap`-ul de comenzi. ‘
Si aici am suprascris metoda `hashCode` si `equals` (in functie de `id` si numele clientului).

5. Rezultate

Generarea produselor de baza(administrator)

The screenshot displays the Administrator interface. On the left, there is a sidebar with buttons: **base**, **IMPORT**, **SEARCH**, **DELETE**, **REFRESH**, **EDIT**, **compose**, **ADD**, **REFRESH**, **SAVE**, **DELETE**, and **EDIT**. The main area is titled **ADMINISTRATOR** and contains a table with the following columns: `title`, `rating`, `calories`, `protein`, `fats`, `sodium`, and `price`. The table lists various products such as Apple Sorbet, Chocolate Or., Vanilla Custard, Crab Cakes, Festival, Pear-Hazelnut, Chocolate w., Veal Scallops, Baked Salmon, Thai Curry, Mustard-Crout, Lemon-Nutm., Cheddar Hors., Braised Lamb, Roasted Quail, Fried Onion R., Pear and Gin., Lychee Martini, Braised Lamb, Thai Spice S., Rice with Soy, Bubby's Gran., Arctic Char w., Saffron Shrim., Spicy Hoisin, Sideswalker, Frangipane, Michael Lewis, White Chocol., Lemon Lime, Camoli Chee, Sweet Potato, Crème Fraîche, Pistachio-Cru., Curried Chick., Lemon Pudd., and Masa Stuffing. On the right side, there is a **REPORTS** section with buttons for **Star Hour**, **End Hour**, **Nr prod**, **Clients**, **Amount**, **Products**, and **Date**. Below this is a **NEW BASE PRODUCT** form with fields for **TITLE**, **RATING**, **CALORIES**, **PROTEIN**, **FATS**, **SODIUM**, and **PRICE**, and an **ADD** button.

Crearea unei comenzi (client)

MENU

FOOD DELIVERY						
title	rating	calories	protein	fats	sodium	price
Apple Sorbet	3.125	217	1	1	7	73
Chocolate Cr.	3.75	410	6	29	80	99
Vanilla Custard	5.0	306	8	20	57	86
Crab Cakes	4.375	462	19	25	642	27
Festival	0.0	219	0	8	12	16
Pear-Hazelnut	3.125	537	14	25	128	93
Chocroute w.	4.375	720	36	33	787	27
Veal Scallops	3.75	301	23	17	773	72
Baked Salmon	4.375	287	13	17	180	11
Thai Curry	3.75	358	14	24	81	48
Mustard-Citrus	4.375	556	10	3	53	27
Lemon-Mulm.	3.75	80	1	5	17	17
Cheddar Hots	4.375	252	9	23	371	26
Roasted Quail	4.375	634	21	24	1158	93
Fried Onion R.	3.75	607	15	30	49	28
Pear and Gin.	4.375	316	4	13	264	75
Lychee Marini	7.0	514	2	1	5	17
Braised Lamb	3.75	942	20	45	430	100
Thai Spices S.	3.75	644	8	28	953	67
Rice with Soy	4.375	386	8	3	336	78
Bubby's Gran.	2.5	407	10	11	23	48
Arctic Char w.	4.375	473	39	31	162	27
Saffron Shrim	4.375	80	6	4	310	55
Onion-Mulm.	6.0	6678	212	603	2064	58

CLIENT ale

FILTER

TITLE:

RATING: FATS:

CALORIES: SODIUM:

PROTEIN: PRICE:

SEARCH CLEAR

order 83

ADD ORDER

Generarea raportului "startHour-endHour" pentru intervalul 11-13

Hour report:

ORDER 1 CLIENT ale 54\$ 2022-05-10 12:21

ORDER 2 CLIENT ale 54\$ 2022-05-10 12:23

ORDER 3 CLIENT ale 59\$ 2022-05-10 12:24

4. Concluzii

Prin intermediul acestei teme am invatat sa folosesc stream-uri ,expresii lambda si set-uri.

Ca posibilitate de dezvoltare ulterioara ,s-ar putea introduce o interfata angajat care sa execute comenzile clientilor.

5. Bibliografie

<https://docs.oracle.com/en/>

<https://www.geeksforgeeks.org/>

<https://stackoverflow.com/>