DOCUMENTATIE

FOOD DELIVERY MANAGEMENT SYSTEM

Studenta: Hirean Roxana-Maria

Grupa: 30224

Cuprins

[Obiective 2](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc2281_1065694278)

[Analiza problemei: scenarii și cazuri de utilizare 3](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc2283_1065694278)

[Cerinte functionale 3](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc2285_1065694278)

[Cazuri de utilizare 3](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc1334_269940582)

[Vizualizarea comenzilor plasate de clienti 3](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc2438_1065694278)

[Vizualizarea produselor 4](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc1336_269940582)

[Cautarea unui produs 4](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc1338_269940582)

[Plasarea unei comenzi 5](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc1340_269940582)

[Administrarea produselor 6](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc1342_269940582)

[Proiectare 7](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc2453_1065694278)

[Diagrama UML 7](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc2455_1065694278)

[Implementare 7](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc2465_1065694278)

[Clase 7](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc2467_1065694278)

[Interfata grafica 7](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc2469_1065694278)

[Clasa Delivery Service 8](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc2471_1065694278)

[Rezultate 8](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc1344_269940582)

[Dezvoltari ulterioare 8](file:///C:\Users\ROXY\Downloads\Documentatie%20(3).odt#__RefHeading___Toc2487_1065694278)

**OBIECTIVE**

Principalul obiectiv al acestei teme este implementarea unei aplicatii care simuleaza un sistem de livrari de mancare . Aceasta aplicatie are o parte care administreaza sistemul ( adauga produse in meniu , le editeaza sau le sterge ) , o parte care se ocupa cu comenzile ( angajatul ) si partea de client care pot face comenzi .

Ca obiective secundare avem :

* Utilizarea unui “ design patterns “ din Java (“ Observer “ ) ;
* Utilizarea unor colectii predefinite in implementarea sistemului de livrari ( “ HashMap “ ) ;
* Serializarea si deserializarea datelor .

Alte obiective ce trebuiesc indeplinite sunt urmatoarele:

1. Realizarea a patru interfete grafice , câte una pentru fiecare tip de utilizator. Interfetele au rolul de a realiza interacțiunea dintre utilizator si program, prin prelucrarea datelor introduse de catre utilizator, dar și de a prezenta rezultatul operatiilr de modificare a datelor efectuate de către utilizator.Fiecare interfata conține câte un Jbutton pentru fiecare operație pe care programul o poate realiza în funcție de interesele tipului de utilizator.
2. Salvarea datelor utilizate pentru conectarea utilizatorului dar si cele referitoare la comenzile plasate pana in prezent,. Salvarea se realizeaza prin serializare.
3. Afisarea intr un fisier .txt a facturilor tuturor comenzilor din fisier. Acestea cuprind id-ul comenzii, al clientului, datele fiecarui produs comandat, data la care a fost plasata comanda, cantitatea de produs comandata, dar si suma pe care trebuie sa o plateasca clientul, aceasta fiind produsul dintre numarul de produse si pretul unui produs.
4. Extragerea datelor din fisierele in care au fost salvate cand aplicatia este pornita, realizata prin deserializare.
5. Importarea produselor de baza prin citirea caracteristicilor acestora dintr un fisier .csv, utilizand stream si expresii lamda.
6. Actualizarea dupa plasarea fiecarei comenzi a numarului de produse din stoc, prin scaderea numarului de produse comandate de catre client.
7. Realizarea operatiilor de stergere, adaugare sau modificare a datelor produselor sau adaugarea unui nou produs compus din produse de baza sau din alte produse compuse, dar si afisarea tuturor produselor, in functie de comanda aleasa.
8. Implementarea functiilor pentru logarea clientilor sau inregistrarea unui nou utilizator.

**ANALIZA PROBLEMEI: SCENARII SI CAZURI DE UTILIZARE**

Principalul scop este de a dezvolta un program ce implementeaza toate cerintele cerute , si , in acelasi timp , foloseste paradigmenele Programarii Orientate pe Obiect . Pentru a face asta , avem nevoie sa gasim un mod sa ne stocam informatiile , si pentru asta , alegem “ HashSet “ pentru a retine meniului sistemului , o lista cu toate comenzile effectuate , si un “ HashMap “ , care are ca si cheie comanda , si ca valoare , o lista de “ MenuItems “ . Oricum , adevarata provocare este sa implementam un anumit “ desing pattern “ , si sa serializem si sa deserializam obiectele noastre , pentru a nu pierde datele cand aplicatia este inchisa.

Cerinte functionale

* Aplicatia trebuie sa prelucreze în mod corect datele introduse de utilizator.
* Aplicatia ar trebui sa i permita utilizatorului sa selecteze orice operație din interfata grafica corespunzatoare categoriei de utilizatori în care se incadreaza.
* Aplicația ar trebui sa efectueze cu succes orice operație selectata de catre utilizator.
* Aplicatia ar trebui să facă posibila vizualizarea datelor și rezultatelor după fiecare operație în interfata grafica sau fisierele txt sau ser.
* Aplicatia ar trebui sa afiseze într-un fisier .txt câte o factura pentru fiecare comanda efectuata.
* Aplicația ar trebui să aibă o interfata grafica cu un aspect plăcut dar care să fie și organizata și ușor de utilizat.

Cazuri de utilizare:

**Vizualizarea comenzilor plasate de clienti**

**Actor principal**: angajatul

**Scenariul principal de succes:**  
1. Utilizatorul selecteaza prima fereastra (“ANGAJAT”),

2. Utilizatorul apasa butonul cu textul“Vizualizare comenzi”.

3. Apare un nou panou cu un scroll pane in care apar comenzile

4. Cand un client plaseaza o noua comanda, aceasta este afisata in fereasta.

5. Comanda este procesata de catre angajat.

**Secvență alternativă**:

1. Niciun client nu este conectat

2. Interfata va ramane goala pana cand un client se conecteaza si plaseaza o comanda, cand se revine in cazul de success.

**Vizualizarea produselor**

**Actor principal**: clientul

**Scenariul principal de succes:**  
1. Utilizatorul selecteaza prima fereastra (“CLIENT”),

2. Clientul introduce un nume de utilizator si o parola si apasa pe butonul “Autentificare”,  
3. Utilizatorul selecteaza optiunea de afisare a listei de produse prin apasarea primului buton.

4. Se deschide un nou panou care contine un scroll pane in care sunt afisate toate produsele si un buton pentru revenire,

5. Utilizatorul apasa pe butonul cu sageata si revine la meniul prinipal.

**Secvență alternativă**: Date de conectare invalide

Utilizatorul introduce datele pentru autentificare.

a) Datele nu corespund niciunei valori din lista de clienti.

1. Se apasa pe butonul pentru inregistrare .

2. Datele noului client sunt savare in lista de clienti din fisierul .ser.

3. Clientul se poate autentifica cu succes cu datele salvate anterior.

4. Se revine la scenariul de succes

b) Datele corespund contului unui alt utilizator.

1. Se apasa pe butonul pentru revenire (“<=”).

2. Se reintroduc datele corecte sau se parcurg pasii de la punctul a) pentru clientii noi.

3. Se revine in scenariul de succes.

**Cautarea unui produs**

**Actorul principal: clientul.**

**Scenariul principal de succes:**

1. Utilizatorul selecteaza prima fereastra (“CLIENT”),

2. Clientul introduce un nume de utilizator si o parola si apasa pe butonul “Autentificare”,  
3. Utilizatorul selecteaza optiunea de cautare a unui produs din listei de produse prin apasarea celui de al doilea buton.

4. Se deschide un nou panou care contine patru text fielduri in care se introduc datele pentru cautare corespunzatoare fiecarui criteriu si un scroll pane in care sunt afisate toate produsele care corespund datelor introduse si un buton pentru revenire,

5. Clientul introduce datele pentru fiecare criteriu dupa care face cautarea.

6. Utilizatorul apasa butonul din dreapta criteriului selectat.

7. Rezultatele cautarii sunt afisate in interfata grafica.

5. Utilizatorul apasa pe butonul cu sageata si revine la meniul prinipal.

**Secvență alternativă**: Datele introduse sunt invalide

Utilizatorul introduce datele pentru autentificare.

Datele pentru autentificare nu corespund niciunei valori din lista de clienti.

- Se apasa pe butonul pentru inregistrare .

- Datele noului client sunt savare in lista de clienti din fisierul .ser.

- Clientul se poate autentifica cu succes cu datele salvate anterior.

- Se revine la scenariul de succes

Datele introduse nu corespund niciunei valori din lista de produse

- Scroll pane-ul din interfata grafica ramane gol pana la introducerea unor date din lista.

**Plasarea unei comenzi**

**Actor principal**: clientul

**Scenariul principal de succes:**

1. Utilizatorul selecteaza prima fereastra (“CLIENT”),

2. Clientul introduce un nume de utilizator si o parola si apasa pe butonul “Autentificare”,  
3. Utilizatorul selecteaza optiunea de plasare a unei comenzi.

4. Se deschide un nou panou care contine un combo box care contine denumirile tuturor produselor din lista si trei butoane: unul pentru a adauga un produs in lista de cumparaturi; unul pentru a finaliza o comanda si un buton pentru revenirea la meniul principal.

5. Clientul selecteaza pe rand produsele pe care doreste sa le comande.

6. Dupa fiecare produs selectat se apasa butonul pentru adaugare.

7. Utilizatorul apasa butonul pentru finalizarea comeni dupa ce a introdus toate produsele dorite.

8. Comanda plasata poate sa fie vizualizata in interfata pentru client, in fisierul ser sau in factura generata in fisierul .txt.

9. Utilizatorul apasa pe butonul cu sageata si revine la meniul prinipal.

10. In fisierul .txt sunt afisate facturile comenzilor plasate pana in preznt.

**Secvență alternativă**: Datele introduse sunt invalide

Utilizatorul introduce datele pentru autentificare.

Datele pentru autentificare nu corespund niciunei valori din lista de clienti.

- Se apasa pe butonul pentru inregistrare .

- Datele noului client sunt savare in lista de clienti din fisierul .ser.

- Clientul se poate autentifica cu succes cu datele salvate anterior.

- Se revine la scenariul de succes

Clientul nu finalizeaza comanda

- Daca o comanda nu este finalizata inainte de a iesi din fereastra, comanda nu este luata in considerare si trebuie plasata o comanda noua.

**Administrarea produselor**

**Actor principal**: administratorul

**Scenariul principal de succes:**  
1.Utilizatorul selecteaza prima fereastra (“ADMINISTRATOR”),

2.Utilizatorul apasa al doilea buton.

3. Se deschise un nou panou in care se introduc datele pentru produse si se selecteaza operatia dorita.

4. Utilizatorul introduce denumirea produsului adaugat si completeaza restul informatiilor cu valorile dorite.  
5. Utilizatorul apasa pe butonul “Adaugare”.

6. Este adaugat un nou produs cu caracteristicile introduse in interfata grafica.

7. Se introduce un titlu al unui produs existent in meniu.

8. Administratorul apasa butonul pentru stergere.

9. Produsele cu acea denumire sunt sterse din lista de produse.

10. Este introdusa denumirea unui produs existent. Se seteaza valoarea pe care administratorul doreste sa o modifice .

11. Utilizatorul apasa butonul pestru modificare. Produsul modificat poate sa fie vizualizat in lista de produse din interfata pentru clienti.

12. Utilizatorul apasa butonul pentru revcenire si se intoarce la meniul principal pentru administrator.

**Secvență alternativă**: Date invalide

1. Utilizatorul introduce in casutele pentru numere caractere invalide

- Apare o exceptie de tipul format number exception .

- Se asteapta introducerea unor date valide

-Se revine la scenariul de succes.

2. Datele introduse nu corespund niciunei valori din lista de produse

- Scroll pane-ul din interfata grafica ramane gol pana la introducerea unor date din lista.

**PROIECTARE : DIAGRAMA UML**

**Diagram

Description automatically generated**

**IMPLEMENTARE**

**Clase:**

Clasele sunt impartite in diferite pachete, in functie de rolul fiecareia in cadrul acestui proiect:

1. Pachetul “Presentation” – acesta contine clasele care implementeaza interfetele grafice
2. Pachetul “Business” – acesta contine clasele cu functiile realizate de program
3. Pachetul “Data Layer” – acesta contine clase folosite pentru scrierea datelor in fisier, pentru serializare si clasele Client, Administrator, Employee si User

Interfata grafica

Proiectul contine patru interfete grafice mai importante, cate una pentru fiecare categorie de utilizatori.

Interfata pentru REGISTER si LOG IN: Pagina pentru autentificare contine doua text fielduri in care se introduc numele de utiliztor si parola si doua butoane: unul pemtru autentificare pentru clientii ale caror date au fost deja salvate, si un buton pentru inregistrarea unui nou client.

Interfata pentru CLIENT GUI care poarta acelasi nume, contine o fereastra principala pentru autentificare, o fereastra pentru meniul principal de unde se poate selecta operatia dorita si cate o fereastra pentru fiecare actiune efectuata. Pagina pentru cautarea unui produs contine text fielduri pentru fiecare criteriu in functie de care se poate face cautarea, cate un buton pentru fiecare categorie care este apasat pentru fiecare cautare, un buton pentru revenire si un panou pentru afisarea rezultatelor. Fereastra pentru plasarea unei comenzi care contine un combo box pentru adaugarea produselor, un buton pentru adaugare si unul pentru finalizarea comenzii.

Interfata pentru EMPLOYEE GUI care contine doar un buton pentru afisarea comenzilor si un panou cu un scroll pane in care sunt afisate comenzile plasate si butoane pentru finalizarea acestora.

Interfata pentru ADMINISTRATOR GUI contine operatiile pentru citirea produselor, administrarea acestora, adaugarea unui produs compus si generarea rapoartelor. Fereastra pentru administrarea produselor contine text fielduri, si butoane pentru adaugarea, stergerea sau modificarea unui produs si pentru revemire. Fereastra pentru crearea unui produs compus contine un text field, un buton pentru adaugarea produselor si unul pentru adaugarea produsului finalizat. Fereastra pentru generarea rapoartelor cintine text fieldurile pentru datele necesare pentru fiecare raport si cate un buton pentru fiecare rapport si un scrol pane in care sunt afisate rezultatele raoartelor.

**Clasa Delivery Service**

Clasa Delivery Service este clasa de baza a proiectului si cuprinde metodele ce implementeaza operatiile realizare de catre client si manager. Clasa implementeaza interfata iDelivery Service Processing , utilizand Design by Contract Pattern . Metoda de tipul “Well formed” verifica conditiile care trebuie sa fie indeplinite pentru apelarea fiecarei metode. Pe langa aceste metode; in aceasta clasa mai sunt implementate si metode care seteaza si returneaza valoarea unei variabile a clasei.

Clasele **Base Product** si **Composite product** extind clasa abstracta **Menu Item**, implementeaza Composite Design Pattern.

**REZULTATE**

Au fost efectuate mai multe testari cu rezultat pozitiv, al caror rezultat a fost vizibil in interfata grafica, dat si in fisierul file.txt, continand descrierea evenimentelor din fiecare moment, dar si rezultatul simularii sub forma unei statistici.

**DEZVOLTARI ULTERIOARE**

Acest proiect poate sa fie dezvoltat prin marirea intervalului de valori pe care il pot lua atributele, sau chiar pein punerea in practica intr-un magazin adevarat.