DOCUMENTATIE.

FOOD DELIVERY MANAGEMENT SYSTEM.

Darlaczi Zoltan-Attila

Grupe 30229

**Cuprins**

1. Cerinta problemei ....……………………………….………….……
2. Obiectiv ..…………………………………………………………...
3. Implementare ………………………………………………………
   1. Diagrama UML ………………………………………………
   2. Clase si Metode ………………………………………………
   3. Interfata grafica (GUI)……………………..…………………
4. Testare ……………………………………………………………...
5. Concluzii si Dezvoltari ulterioare .…………………………………
6. Bibliografie ...………………………………………………………

Cerinta problemei

Cerinte temei de lavorator a fost proiectarea și implementarea unui sistem de management al livrării alimentelor pentru o companie de catering. Clientul poate comandați produse din meniul companiei. Sistemul ar trebui să aibă trei tipuri de utilizatori care se conectează folosind un nume de utilizator și o parolă: administrator, angajat obișnuit și client. Operatiile executate de administrator sunt: importarea setului initial de produse care este completat dintr-un fisier.csv, gestionare produserlor din meniu, actiuni prin care se poate gestiona produsels sunt: adaugare, stergere, modificare si creare meniu, ceea ce inseamna ca alcatuim un meniu sau in produs mai complex din produse mai base si din alte produse complexe. Generare rapoartelor de comenzi effectuate luan in considearea urmatoarele criterii: intervalul de timp al comenzilor, ar trebui generat raport cu comenzile effectuate intre o anumita ora de inceput si o ora sfarsita, indiferent de data. între o anumită oră de început și o dată de sfârșit dată, indiferent de dată.o produsele comandate de mai multe ori de câte ori până acum.o clienții care au comandat mai mult de un număr specificat de ori și valoareadin comanda a fost mai mare decât o sumă specificată.o produsele comandate într-o zi specificată cu numărul de ori pe care le au a fost comandat. Clientul poate: inregistrarea cu numele de utilizator si parola, apoi aceste date sunt folosite si la logare. Poate sa vizualizeze lista de produse, poate face search intre produsele si poate sa creeze o comanad, pentru fiecare comanda este generate nota, pe care trebuie afisata data, pretul si produsele comandate. Iar employee primeste doar notificare cand un client face o comanda.

Obiectiv

A fost proiectarea și implementarea unui sistem de management al livrării alimentelor pentru o companie de catering. Clientul poate comandați produse din meniul companiei. Sistemul ar trebui să aibă trei tipuri de utilizatori care se conectează folosind un nume de utilizator și o parolă: administrator, angajat obișnuit și client. Si a realiza operatiile sau actiunile effectuate de catre administrator client si employee. Administrator face import la un set de date, sau mai exact produse dintre care poate clientul sa alege ce vrea sa comanda. Adminul poate sa adauga, sterge, modifica un produs din acest set sau sa creeze un meniu nou. El poate sa genereze si rapoarte dupa anumite conditii, de exmplu toate comenzile effectuate intre doua ore, indifferent de dat. Clientul poate sa vizualizeze produsele, meniurile, poate sa cauta intre produse si poate sa aleaga mai multe produse si sa faca o comanda. Iar employee primeste doar notificare cand un client face o comanda.

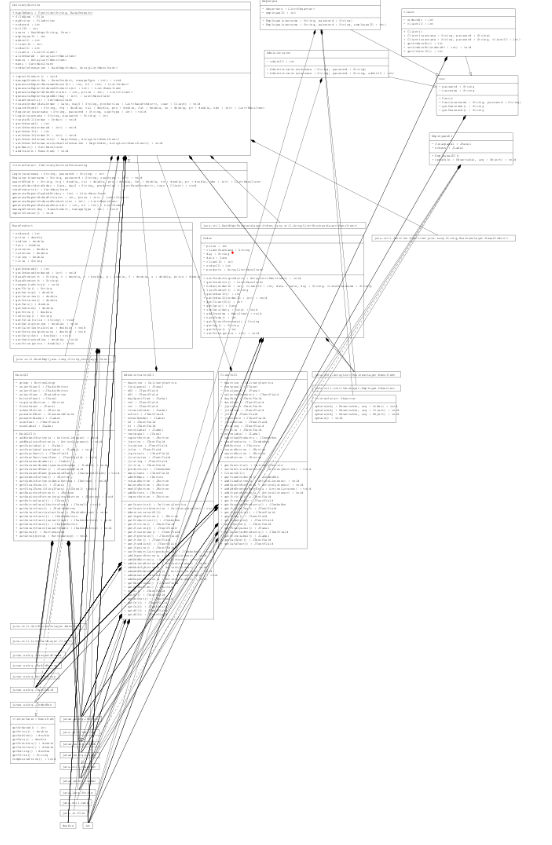
Implementare

Pentru implementare prima si prima data am inceput sa facem clasele dupa diagram care se afla in laborator. Sa creeam toate clasele si legaturile intre clase, sis a obtinem o diagrama asemenetoare. Apoi am implementat o interfata grafica pentru login, cu 2 butoane login si register, apoi pentru fiecare tip de use cate o interfata grafica si pentru fiecare butoane, si textfieldul sau comboxuri sau tabele pentru a putea efectua actiunile lor.

Incepem cu a realiza inregistrarea pentru fiecare tip de client folosint un hashmap cu key-ul username is value password. Apoi luam prima interfata grafica de exemplu admin. Pentru admin facem butoane, cate un buton pentru fiecare actiune, combo box pentru a importa, a vizualiza si a selecte produsele din fisierul .csv. Apoi implementam actiunile pe produse. Adaugare unui produs, stergerea unui produs, modificare unui produs si crearea unui meniu cu mai multe produse. Si dupa aia implementam si metodele, clasele pentru a realiza rapoartele dupa anumite conditii. Pentru a crea un nou meniu, selectem din combo box un porudus, apasam pe un buton si il aduaga in lista de “Asteptare”, iar asa putem sa adauagam mia multe produse. Apoi can apasam pe un buten care realizeaza crearea meniului.

Apoi facem interfata grafica pentru client. Pentru Client facem 2 combo boxuri pentru a vizualiza toate produsele si pentru a putea face search intre produse. Pentru a realiza o comanda din combo box putem sa selectem produsele sis a adaugam asa, este mult mai simplu decat sa scriem in casute de text numele fiecarui produs sau date despre produse. Apoi realizam si interfata grafica pentru Employee. Ne Folosim de Observable Design pattern pentru a notifica employee cand un client face comanda.

Diagrama UML



Unified Modeling Language sau UML pe scurt este un limbaj standard pentru descrierea de modele si specificatii pentru software. UML a fost la bază dezvoltat pentru reprezentarea complexității programelor orientate pe obiect, al căror fundament este structurarea programelor pe clase, și instanțele acestora Cu toate acestea, datorită eficienței și clarității în reprezentarea unor elemente abstracte, UML este utilizat dincolo de domeniul IT.

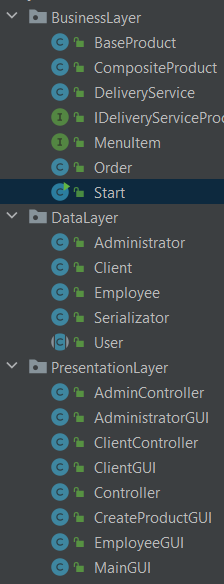
Din pacate nu se vede foarte clar pe diagrama UML legaturile intre clasele pentru ca avem foarte multe clase, iar clasele fac foarte multe import-uri, dar dupa rularea programului si daca ne uitam peste cod ne asiguram ca programul face cee ace trebe si se vede si din cod legaturile intre clase.

Diagrama UML a acestui proiect este destul de complicate. Din moment ce avem foarte multe clase diagrama devine compplicata din ce in ce maim ult. Dar se vede foarte bine ca exista legatura intre toate clasele.

Mai jos o sa descriu clasele, metodele si legaturile intre clase si atunci diagram UML o sa fie clar, si legaturile intre tabele o sa fie explicate si clare

Clase si Metode

In acest proiect am folosit foarte multe clase. Mai jos o sa prezint doar clasele si metodele principale din clase, nu si toate clasele care de exmplu extinda o clasa sau implementeaza o clasa si in rest au constructorul default si nu ai metode.



Deci dupa cum se vede si in imaginea de mai sus, avem 3 package-uri si foarte multe clase. Din cauza asta in sectiunea asta a documentatiei o sa descriu doar clasele cele mai importante, precum DeliveryService, interfata MenuItem si cele doua clase BaseProduct, Composite product, aceste 2 clase impreuna cu interfata folosesc Composite Design Pattern, iar mai sunt clasele: Order is User.

Deci clasa DeliveryService implementeaza interfata IDerliveryServiceProcessing. Practic aceasta clasa este brain-ul, adica mintea acestiu proiect. In aceasta clasa sunt descries si implementate toate metodele care realizeaza actiunile facute de catre admin si client. Deci aceasta clasa este motorul proiectului. Metodele importante descries in aceasta clasa sunt: import products, adaugare product, midficare product, delete product, create menu si 4 metode pentru generare rapoarte, iar alte metode importante pentru client sunt view product, search product si create order, adica creare comanda.

Apoi avem interfata interfata MenuItem si cele doua clase BaseProduct, Composite product, aceste 2 clase impreuna cu interfata folosesc Composite Design Pattern. Aceste clase si interfata sunt clase pentru produs, pentru produse mai complex care este alcatuit din produse mai simple.

Clasa User este o clasa abstracta si este extensa de Administrator, Client si Employee. Iar clasa Order este clasa pentru comanda.

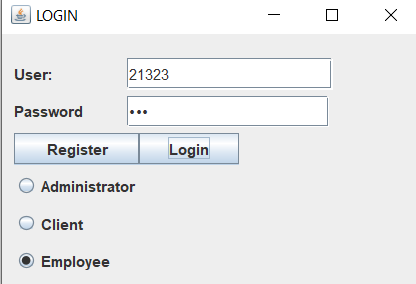
Am realizat si Observable Desgibn Pattern cu EmployeeGui si ClientController, cee ace inseamnca ca de fiecare data cand un client face o comanda, muncitorii sau angajtii sunt anuntate.

Avem 3 package-uri: Business Layer, Data Layet si Presentation Layer. In package-ul Business Layer sunt implementate clasele care realizeaza practice toate actiunile in aces proiect. In Data Layer sunt clasele de utilizator, iar in presentation layer sunt clasele pentru interfete grafice.

Interfata grafica(GUI)

Deci cum am mentionat si mai sus avem 4 interfete grafice:una pentru login, una pentru clienti, una pentru produse, si una pentru employee. Clasele in care sunt realizate interfetele grafice apartin la presentation pacckage.

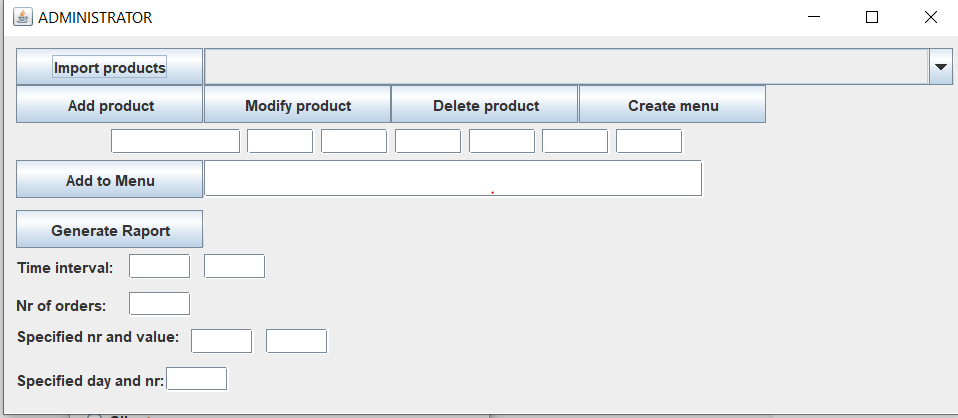
In pozele de mai jos sunt prezentate interfetele grafice asa cum apar ele cand proiectul este rulat, adica la inceputul programului. Cand rulam programul prima data apare interfata de login. Aici trebuie sa selectem tipul de user, apoi sa dam un nume de utilizator si o parola si apoi sa apasam pe register. Ne-am inregistrat, Dupa asta putem sa ne logam fara sa mai selectam tipul, deoarece programul stie dupa date introduse ce fel de utilizator este.



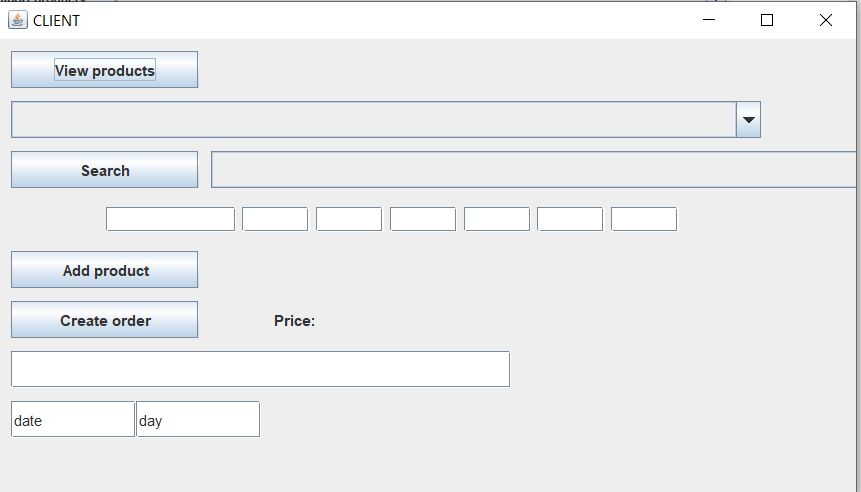
Daca am create un utilizator de admin atunci interfata de mai jos este cea care va aparea. Este o interfata bine aranjata si cred ca dupa butoane deja ne dam seama din prima vedere fiecare buton ce face. Import product importa toate produsele din fisierul .csv. Produsele sunt afisate in combo boxul de langa buton. Apoi avem butonul add product. Pentru a putea folosi acest buton trebuie sa umplam casutele de text sub buton. Fiecare casuta reprezinta o trasatura a unui produs. Titu, rating, calorii, protein, sodiu, fat si price. Fix asa. Deci umplam castutele de text, apasam pe add product si adaugam un nou produs in lista de produse. Daca vrem sa modificam vreun produs atunc il cautam pe lista, il selectem si in casuta conform trasaturi pe care vrem sa o modificam introducem noua valoare, apasam pe buton si se modifica produsul. Daca vrem sa stergem vreun produs, pur si simplu il selectam, apasam pe buton si se sterge produsul.

Pentru a crea un nou meniu, selectem din combo box un porudus, apasam pe Add To Menu si il aduaga in lista de “Asteptare”, iar asa putem sa adauagam mia multe produse. Apoi can apasam pe Create Menu cream meniul.

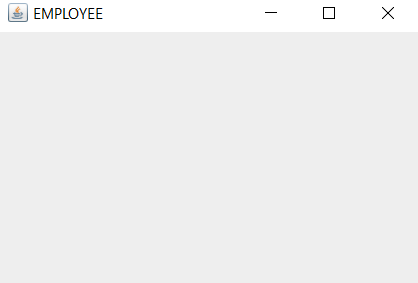
General Raport este butonul pentru a crea rapoarte. Se poate crea 4 tipuri de rapoarte, fiecare are ceva conditie. Raportul se genereaza conform pozitia casutelor de text si conform textului care este introdus sau nu in casuta de text.



Daca am create un utilizator de tip Client atunci aceasta casuta va aparea. Primul buton este pentru a vizualiza produsele. Produsele apar in combo boxul de sub buton. Cu ajutorul butonului Search si a casutelor de text buten realiza cautare in lista de produse. Introducem ceva in casutele de text, practice introducem o conditei si apoi apasam butonul si asa raelizam cautarea. Crearea unei comanda se face asemenator cum se face si creare unui nou meniu la administrator. Pentru a crea un noua comanda, selectem din combo box un porudus, apasam pe Add Product si il aduaga in lista de “Asteptare”, iar asa putem sa adauagam mia multe produse.Introducem data in casuta de text de sub casuta de text in care apare numele produsele adugate in lista. Apoi apasam pe create order si asa cream o comanda. Cand plasam o comanda se trimite o notificare la Employee pentru ca sa fie anuntat ca s-a facut comanda. Acest lucru este realizat folosin Observable Design Pattern.



Daca intram ca si Employee prima data vedem o interfata grafica goala pe car enu apare absolut nimic. Doar atunci apare un text care anunta pe angajatii ca s-a facut o comanda cand un client chiar comanda ceva.



Concluzii si Dezvoltari ulterioare

Toate Comenzile facute sunt afisate intr-un fisier .txt, numit orders.txt. Proiectul realizat este un restaurant , folosind un fisier .csv, puteam sa spunem ca am creat un simulator de restaurant. Administratorul isi face treaba cu produsele, vin clientii, se uita, comanda, apoi angajatii sun anuntati ca s-a facut o comanda si tot asa.

In concluzei in acest laborator a fost proiectat și implementat un sistem de management al livrării alimentelor pentru o companie de catering. Sistemul ar trebui să aibă trei tipuri de utilizatori care se conectează folosind un nume de utilizator și o parolă: administrator, angajat obișnuit și client. Fiecare utilizator poate sa executa operatii/actiuni unice, dar importante pentru realizarea sistemului de managemene al livrarii alimentelor pentru o companie de catering.

Dezvoltari ultarioare: in loc de employee sa fie chelner si bucatar. Iar chelnerul sa fie observer si sa-l anunte pe bucatar observable cand s-a facut o comanda.Se poate implementa si o metoda in care comanda este livrata iar clientul o plateste. Fiecare client are un buget initial.

Inca o dezvoltare ulterioare poate fi adaugarea si data in care comanda a fost plasata dar si o data estimata cand comanda va fi livrata.

Bibliografie

* Youtube
* Wikipedia
* Toate linkurile din pdf-ul proiectului