TEHNICI DE PROGRAMARE

TEMA 4: FOOD DELIVERY MANAGEMENT SYSTEM

STUDENT: PELE CARMEN-IOANA

GRUPA 30222

Cuprins

1.OBIECTIVUL TEMEI

Obiectivul principal al acestei teme este de a implementa un sistem de food-delivery pentru o companie de catering. Clientul poate comanda produse din meniul companiei. Sistem ar trebui sa permita autentificarea a 3 tipuri de utilizatori, administratori, angajati si clienti.

Administratorul :

- Poate importa setul de produse initial
- Poate adauga, modifica si sterge produsele de baza dar poate introduce si produse compuse din mai multe produse de baza
- Poate genera rapoarte dupa anumite criterii

Clientul:

- Se poate loga si isi poate crea un cont
- Poate vedea produsele
- Poate cauta produse in functie de anumite criterii
- Poate crea comenzi

Angajatul :

- Este notificat atunci cand s-a introdus o comanda

Alte sub-obiective ar fi: analizarea problemei si identificarea necesitatilor, design-ul programului, implementarea si testarea functionalitatii lui.

2.ANALIZA PROBLEMEI, MODELARE, SCENARII, CAZURI DE UTILIZARE

Necesitatile functionale ale sistemului sunt:

- Programul ar trebui sa permita utilizatorilor sa se logeze intr-o anumita categorie
- Programul ar trebui sa permita administratorului sa importe setul de produse
- Programul ar trebui sa permita administratorului sa modifice setul de produse, sa adauge unele noi, sa stearga produse si sa creeze produse compuse
- Programul ar trebui sa permita administratorului sa genereze rapoarte in functie de anumite criterii
- Programul ar trebui sa ii permita clientului sa vada produsele oferite de companie
- Programul ar trebui sa ii permita clientului sa caute produse in functie de anumite criterii
- Programul ar trebui sa ii permita clientului sa adauge comenzi de produse
- Programul ar trebui sa ii permita angajatului sa vada in timp real comenziile adaugate

Necesitatile nefunctionale ale sistemului sunt:

- Simulatorul ar trebui sa fie intuitiv si usor de utilizat de catre utilizator

- Programul ar trebui sa atentioneze utilizatorii in cazul in care au introdus date gresite
- Programul ar trebui sa ilustreze tabelele cu produse atat pentru administratori cat si pentru clienti
- Programul ar trebui sa ilustreze evolutia comenzilor in timp real a comenzilor
  Cateva cazuri de utilizare sunt:

Use Case: importare date de catre administrator

Actor principal: administrator:

Scenariu principal de succes:

1. Administratorul se logheaza
2. Administratorul selecteaza din bara de meniu “import”, aceasta se afiseaza automat prima in momentul deschiderii ferestrei
3. Apasa butonul import
4. Datele sunt importate si se pot vedea in timp real in tabelul de pe fereastra

Un alt scenariu in cazul datelor incorect introduse:

1. Administratorul se logheaza
2. Acestuia nu i se permite accesul si este atentionat de faptul ca a introdus date gresite
3. Se reia pasul 1

Use Case: adaugare comanda

Actor principal: client

Scenariu principal de succes:

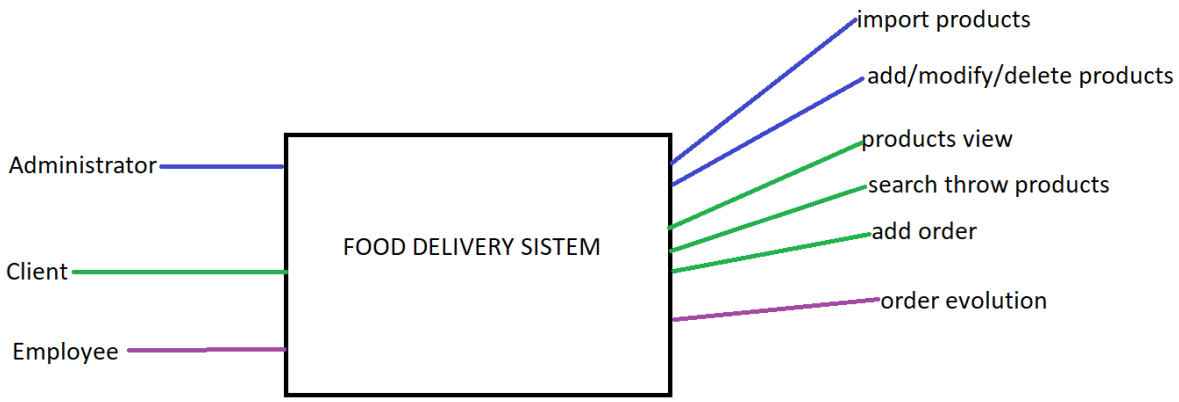
1. Clientul se logheza
2. Dupa ce I se deschide fereastra pentru comenzi acesta selecteaza din tabelul Base Products produsul dorit sau din tabel Composite Product
3. Apasa pe butonul cu denumirea tipului de produs dorit
4. Introduce id-ul sau de client
5. Apasa adauga comanda
6. Comanda este adaugata in lista si angajatul este notificat
7. Clientul poate vedea si el evolutia in timp real a comenzi, apasand “Take a look in the kitchen”

Un alt scenariu in cazul datelor incorect introduse:

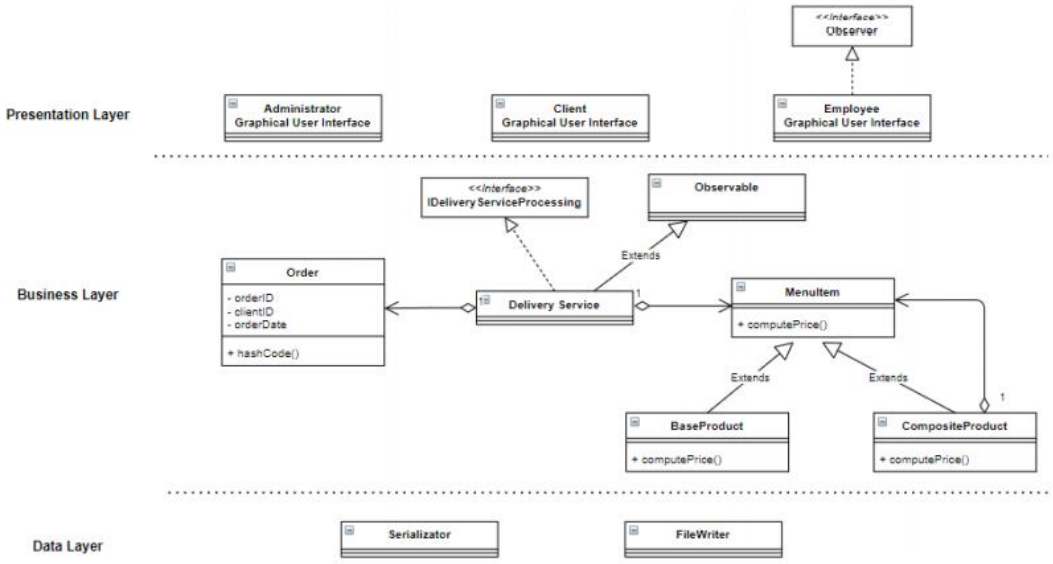
1. Clientul se logheaza
2. Acestuia nu i se permite accesul si este anuntat ca datele introduse pentru logare sunt incorecte
3. Se reia pasul 1

3.PROIECTARE (DECIZII DE PROIECTARE, DIAGRAME UML, STRUCTURI DE DATE, PROIECTARE CLASE, INTERFETE, RELATII, PACKAGES, ALGORITMI, INTERFATA UTILIZATOR)

3.1. Design-ul overall al sistemului:



3.2 Divizarea in sub-sisteme/pachete



3.3.Divizare in clase

Pachetul “businessLayer” contine:

- BaseProduct
- CompositeProduct
- DeliveryService
- IDeliveryServiceProcessing –interfata
- MenuItem
- Observable
- Order

Pachetul “connectionToDatabase” contine:

- AddClient
- ConnectionFactory
- CreateBill
- GenerareRaport
- OrderOperations
- SearchOperatons
- VerifyLogin

Pachetul “dataLayer” contine:

- Serializator
- WriteFile

Pachetul “importing” contine:

- ProductImport

Pachetul “mainData” contine:

- MainControl
- MainDataForDelivery

Pachetul “presentationLayer” contine:

- Administrator
- AdminLogin
- Client
- ClientLogin
- Employee
- EmployeeLogin

- FirstFrame
- NewClient
- Observer - interfata

3.4. Divizarea pe metode:

Clasa BaseProduct are metodele:

- getRating()
- setRating()
- getCalories()
- setCalories()
- getProteins()
- setProteins()
- getFats()
- setFats()
- getSodium()
- setSodium()

Clasa CompositeProduct are metodele:

- addProduse() –adauga un base product in lista de produse a composite product
- getProduse()
- setProduse()

Clase DeliveryService are metodele:

- addMenu()- adauga un produs de tip MenuItem
- getOrd()
- setOrd()
- getMenuitems()
- setMenuitems()
- toString()

Clasa MenuItem are metodele:

- getTitle()
- setTitle()
- getPrice()
- setPrice()
- getTip()
- setTip()

Clasa Observable are metode update().

Clasa Order are metodele:

- addBaseProduct() – adauga pretul predusului la totalul comenzii
- addCompositeProduct()– adauga pretul predusului la totalul comenzii
- getTotal()
- setTotal()
- getOrderID()
- setOrderID()
- getClientID()
- setClientID()
- getLivrat()
- setLivrat()
- getData()
- setData()

Clasa AddClient are metoda adaugareConClient().

Clasa ConnectionFactory are metodele:

- createConnection()
- getConnection()
- close()
- closes()- inchide un statement

Clasa CreateBill are metodele:

- scriereDateClient()
- scriereDateComanda()
- generareFactura()- creaza fisierul bonului si scrie datele clientului si comenzii
- scrie()
- inchidere()

Clasa GenerareRaport are metodele:

- timeIntervalOrders()- genereaza un raport(scrie raportul intr-un fisier text) cu comenzile dintr-un interval de timp introdus de administrator
- productsOrder()- genereaza un raport cu produsele comandate de mai multe ori decat o valoare data
- clientsRap()- genereaza un raport ce contine clientii care au comandat de mai multe ori decat o valoare introdusa, si valoarea comenzi a fost mai mare decat o valoare data

- productsDay()- genereaza un raport care contine produsele comandate intr-o zi introdusa de administrator si de cate ori au fost comandate

Clasa OrderOperations are metodele:

- addOrder()
- idOrder()
- verificareLivrareProdus() – se verifica daca data de finalizare a preparari comenzii e inaintea orei actuale
- verificarePlecareComanda()- se verifica daca data de finalizare a preparari unei comenzi e egala cu ora actuala

Clasa SearchOperations are metodele:

- searchByTitle()
- searchByRating()
- searchByCalories()
- searchByProteins()
- searchByFats()
- searchBySodium()
- searchByPrice()

Clasa VerifyLogin are metodele:

- verificareAdmin()
- verificareAngajat()
- verificareClient()

Clasa Serializator are metodele:

- write() -serializator
- read() -deserializator

Clasa WriteFile are metodele:

- scriere()
- inchidere()

Clasa ProductImport are metoda importpr() care importa produsele din products.csv .

Clasa MainDataForDelivery are metodele:

- addOrder()
- addBasePr()

- addCompositePr()
- removeBasePr()
- modifyBasePr()

Clasa Client reprezinta interfata accesata de client si are metoda notifyy() pentru a notifica angajatul ca s-a adaugat o noua comanda.

3.5. Decizii de proiectare*:*

Lambda expressions In Java

Lambda expresiile ne permit să creăm instanţe ale claselor cu o singură metodă într-un mod mult mai compact.

O *lambda expresie* constă:

- dintr-o listă de parametri formali, separaţi prin virgulă şi cuprinşi eventual între paranteze rotunde,
- săgeata direcţională `->`,
- un body ce constă dintr-o expresie sau un bloc de instrucţiuni.

Stream In Java

Introdus în Java 8, API Stream este folosit pentru a procesa colecții de obiecte. Un flux este o secvență de obiecte care suportă diferite metode care pot fi pipeline pentru a produce rezultatul dorit.
Caracteristicile fluxului Java sunt :

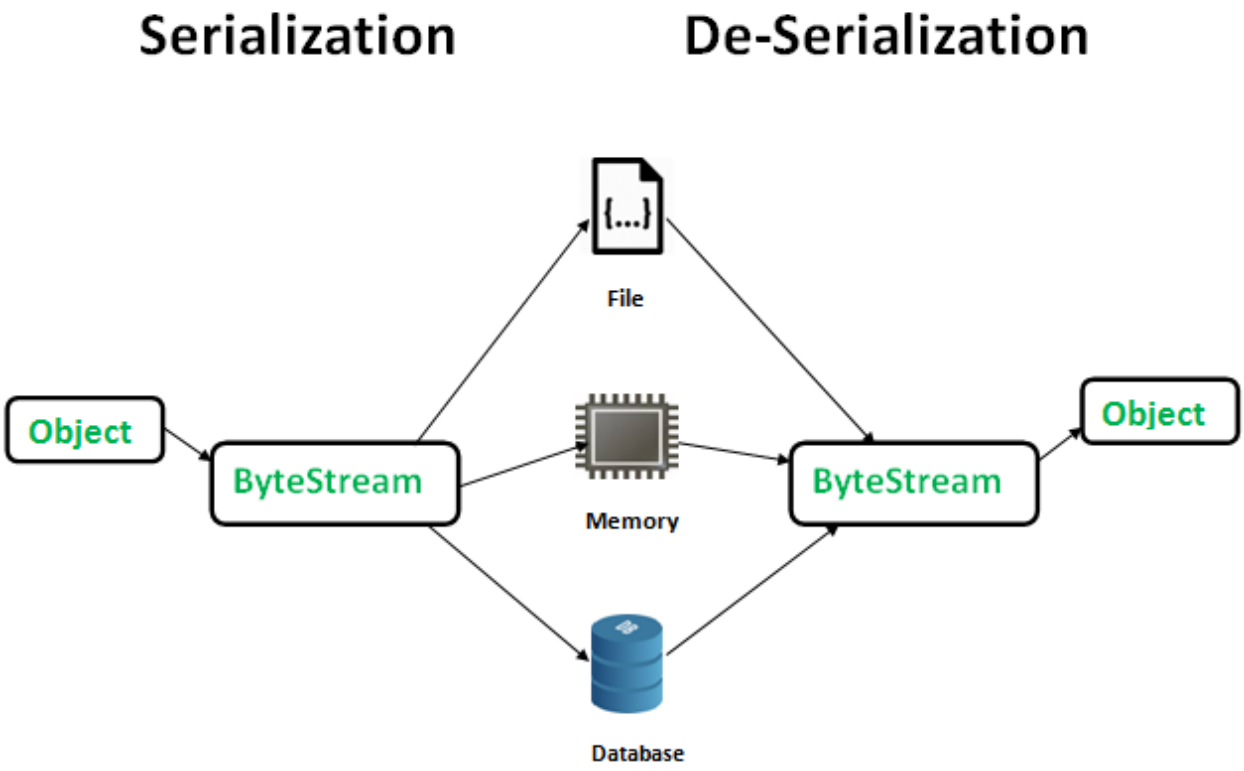
- Un flux nu este o structură de date, ci preia date din colecții, matrice sau canale I/O.
- Fluxurile nu schimbă structura datelor originale, ele oferă doar rezultatul conform metodelor pipeline.

- Fiecare operațiune intermediară este executată leneș și returnează un flux ca rezultat, prin urmare pot fi pipeline diferite operații intermediare. Operațiile terminalului marchează sfârșitul fluxului și returnează rezultatul.

Serialization and Deserialization in Java

Serializarea este un mecanism de conversie a stării unui obiect într-un flux de octeți. Dealerializarea este procesul invers în care fluxul de octeți este utilizat pentru a recrea obiectul Java real în memorie. Acest mecanism este utilizat pentru a persista obiectul.

Fluxul de bytes creat este independent de platformă. Deci, obiectul serializat pe o platformă poate fi dezertalizat pe o platformă diferită.



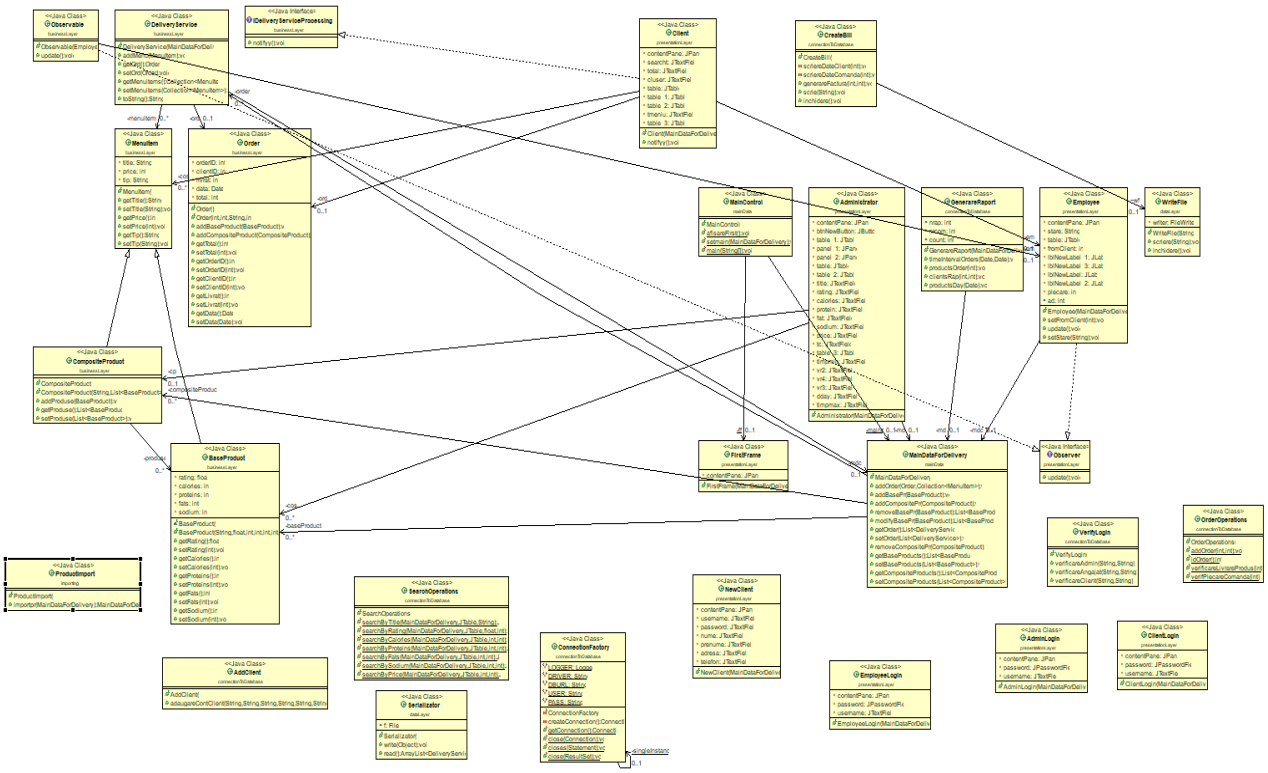
Assertions in Java

Un assert permite testarea corectitudinii oricăror ipoteze care au fost făcute în program.

Assertul este realizat folosind assert din Java. În timp ce execută afirmație, se crede că este adevărat. Dacă nu reușește, JVM aruncă o eroare numită AsserError. Acesta este utilizat în principal în scopuri de testare în timpul dezvoltării.

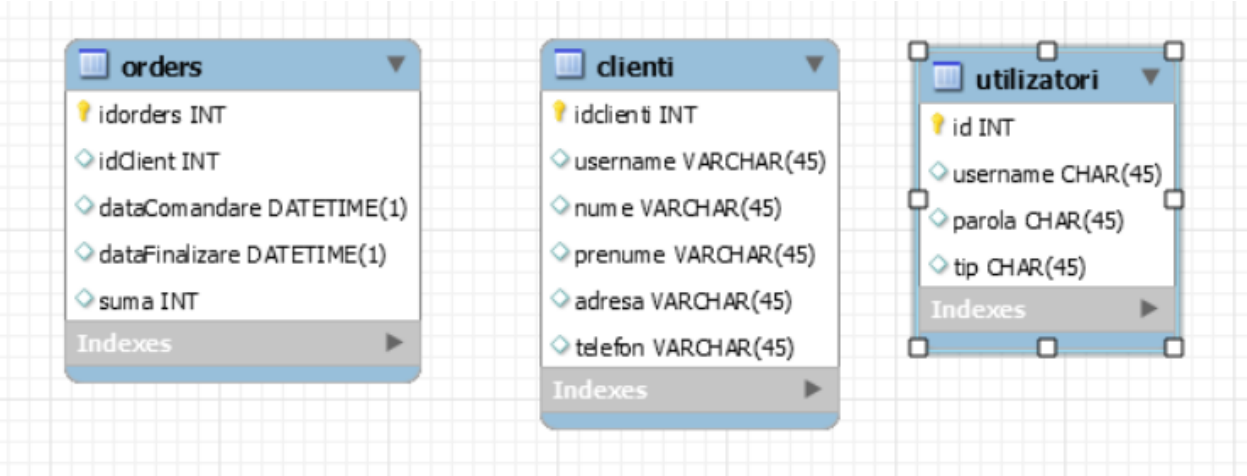
Declarația de assert este utilizată cu o expresie booleană și poate fi scrisă în două moduri diferite.

3.6 Diagrame UML



3.7. Structuri de date

In baza de date avem urmatoarea structura a datelor:



In sistemul din program avem urmatoarele structuri de date:

- MenuItem are ca campuri title, price, tip

- BaseProduct extinde MenuItem si are in plus campurile rating, calories, proteins, fats, sodium
- CompositeProduct extinde MenuItem si are campurile produse- care reprezinta lista de baseProducts pe care le contine meniul
- Order are campurile orderId, clientID, livrat (avem nevoie de acest camp pentru a verifica comenziile care au fost deja livrare si cele care sunt in curs de prelucrare), data
- Delivery Service are campurile ord- care reprezinta comanda, de tip Order, si menuitems- care reprezinta o colectie de produse pe care le doreste clientul
- MainDataForDelivery retine datele care sunt partajate intre fereste, astfel ca la datele principale ale sistemului sa fie disponibile in timp real in fiecare fereastra, o modificare a unei valori afectand toate celelalte ferestre. Aceasta are structura baseProducts (reprezinta lista de BaseProducts pe care o importa administratorul), compositeProducts (reprezinta lista de CompositeProducts pe care administratorul le creaza) si order(reprezinta o lista de tip DeliveryService care reprezinta defapt o structura ce pastreaza datele comenzii impartite pe anumite campuri din interiorul ei).

3.8.Algoritmi

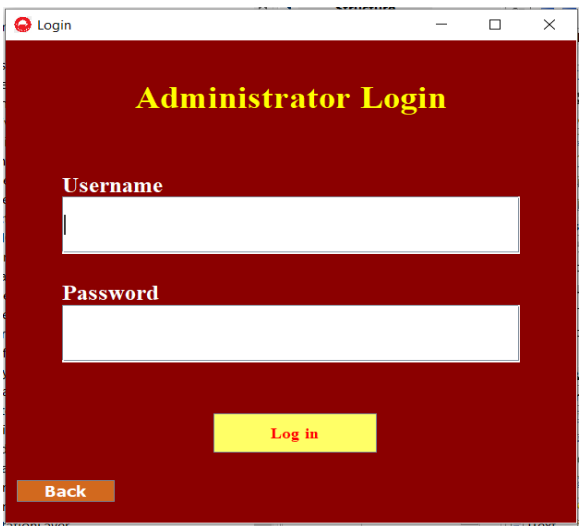
Algoritmul de generare a raportului cu comenzile dintr-un interval de timp: parcurgen lista de comenzi iar daca gasim una care e dupa de data 1 introdusa de administrator, si inainde de data 2, atunci se scrie in raport comanda.

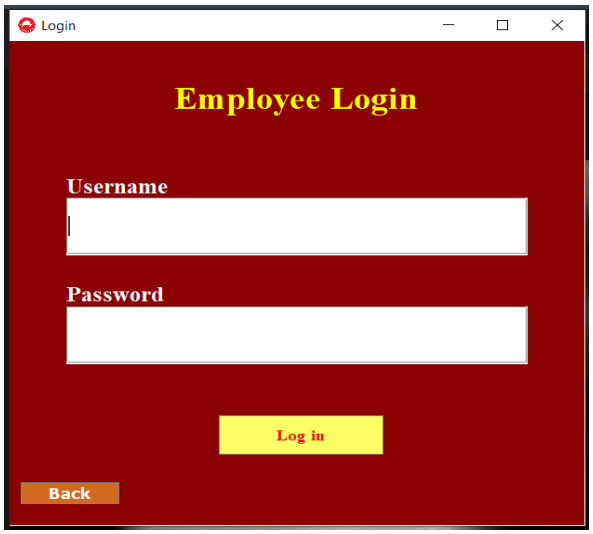
Algoritmul de generare a raportului cu produsele comandate de mai multe ori decat o valoare: Declaram un map in care pastram numele produsului si de cate ori a fost comandat, parcurgem fiecare comanda si actualizam map-ul. Pe urma parcurgem map-ul si daca numarul de ori de cate a fost comandat produsul este mai mare decat valoarea introdusa de administrator, atunci il scriem in raport.

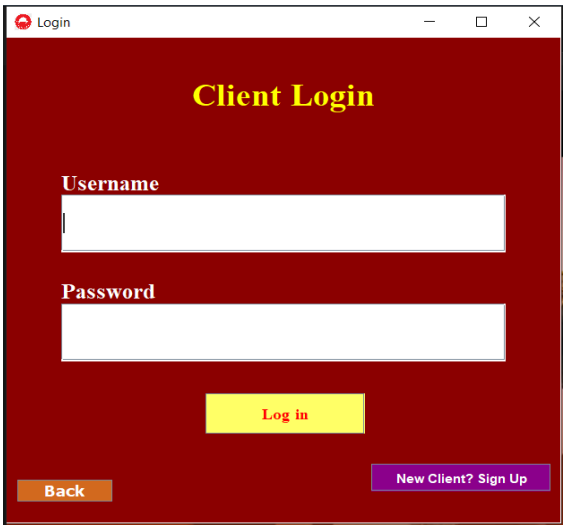
Algoritmul de generare a raportului ce contine clientii ce au comandat de un numar de ori mai mare decat o valoare introdusa si valoarea comenzii era mai mare decat un numar introdus: parcurgem lista de comenzi si punem toti clientii intr-o lista. Dupa luam fiecare client in parte, parcurgem lista de comenzi si daca indeplineste conditiile date il trecem in raport.

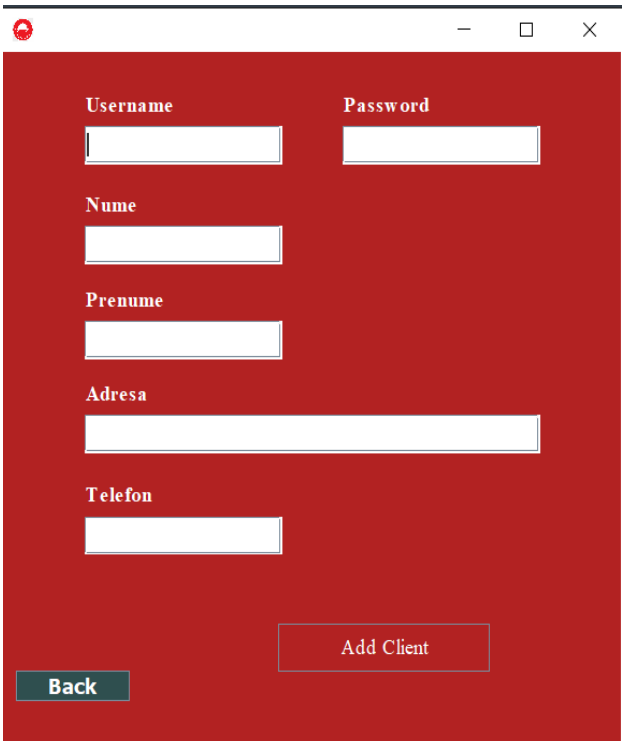
Algoritmul de generare a raportului ce contine produsele comandate intr-o anumita zi: Parcurgem lista de comenzi si populam un map de produse si numarul de comenzi ale lor, daca data comenzii este ziua introdusa de administrator atunci se adauga in raport produsul cu contorul.

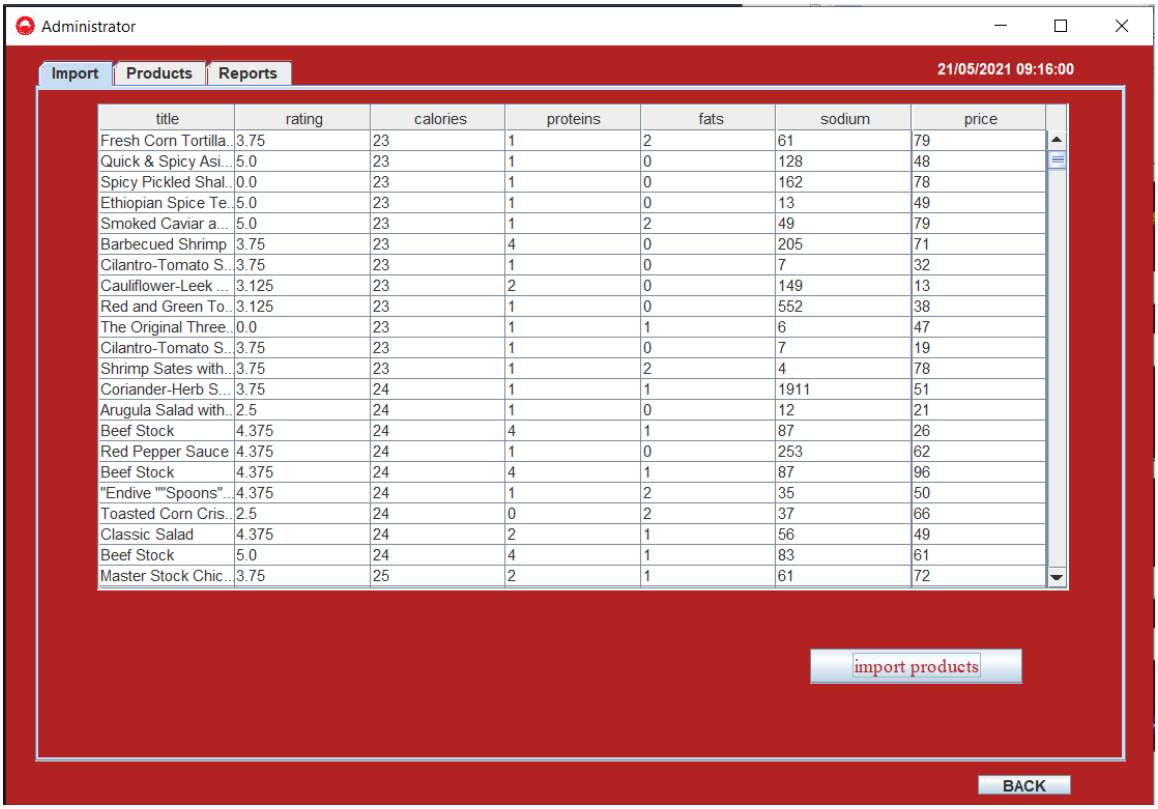
3.9.Interfata utilizatorului



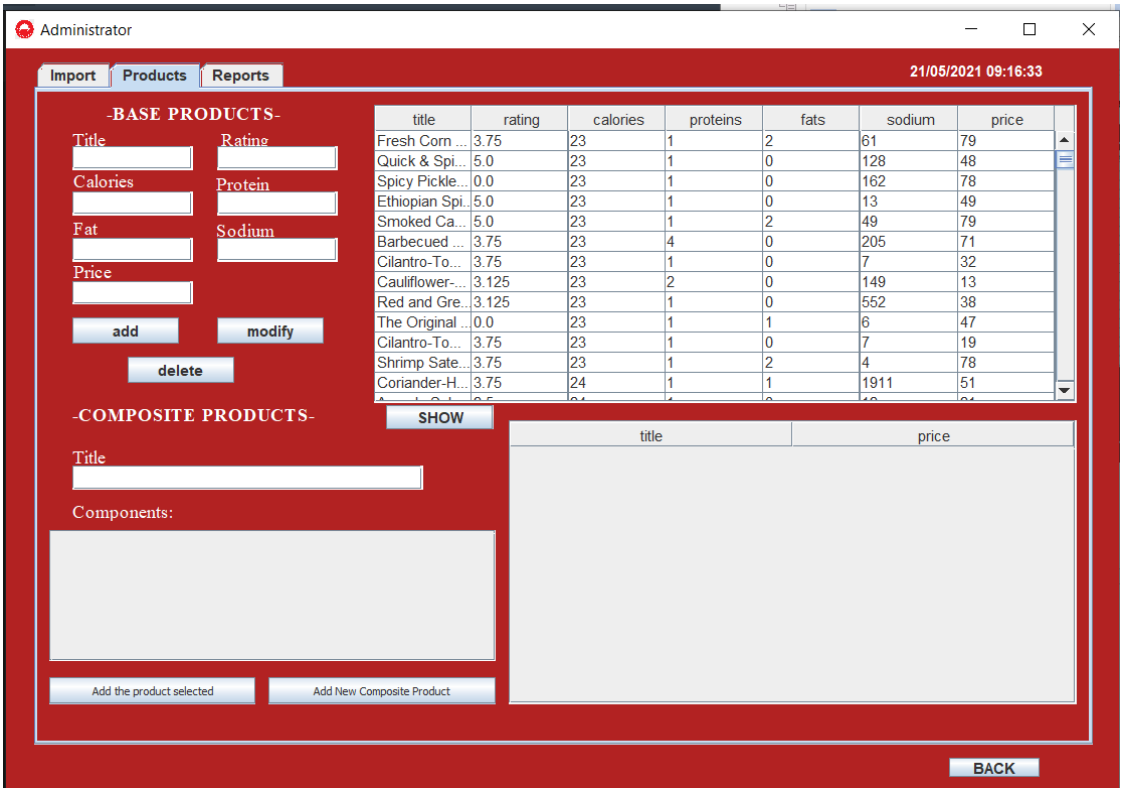




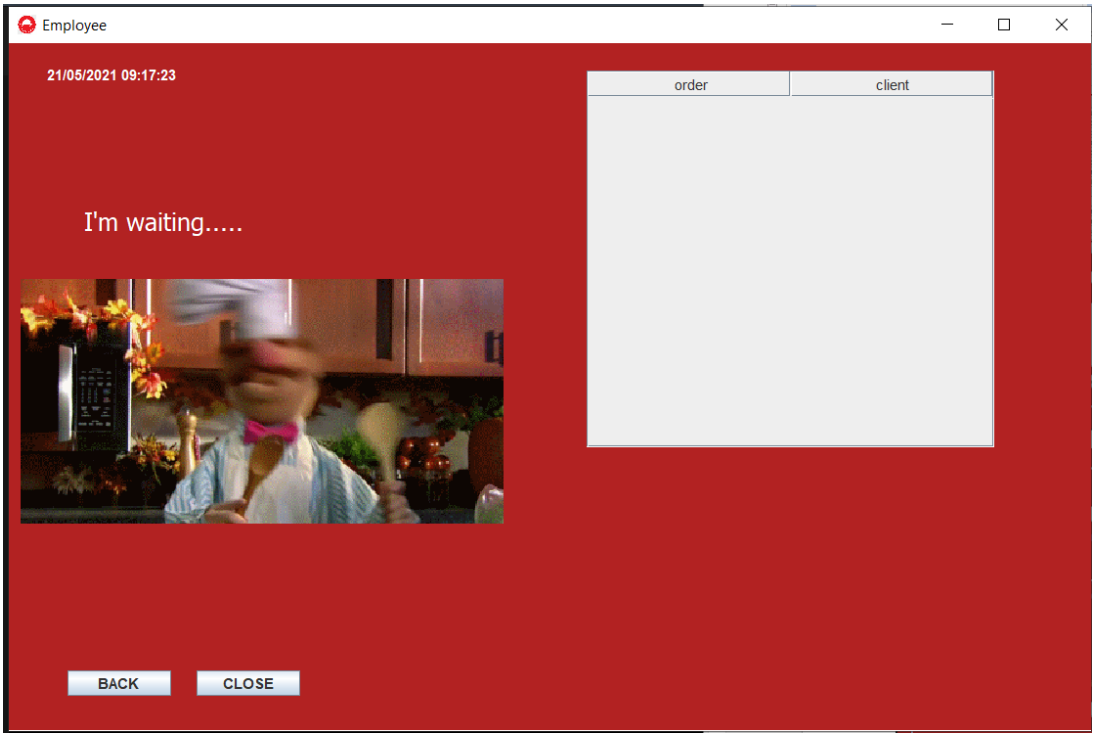


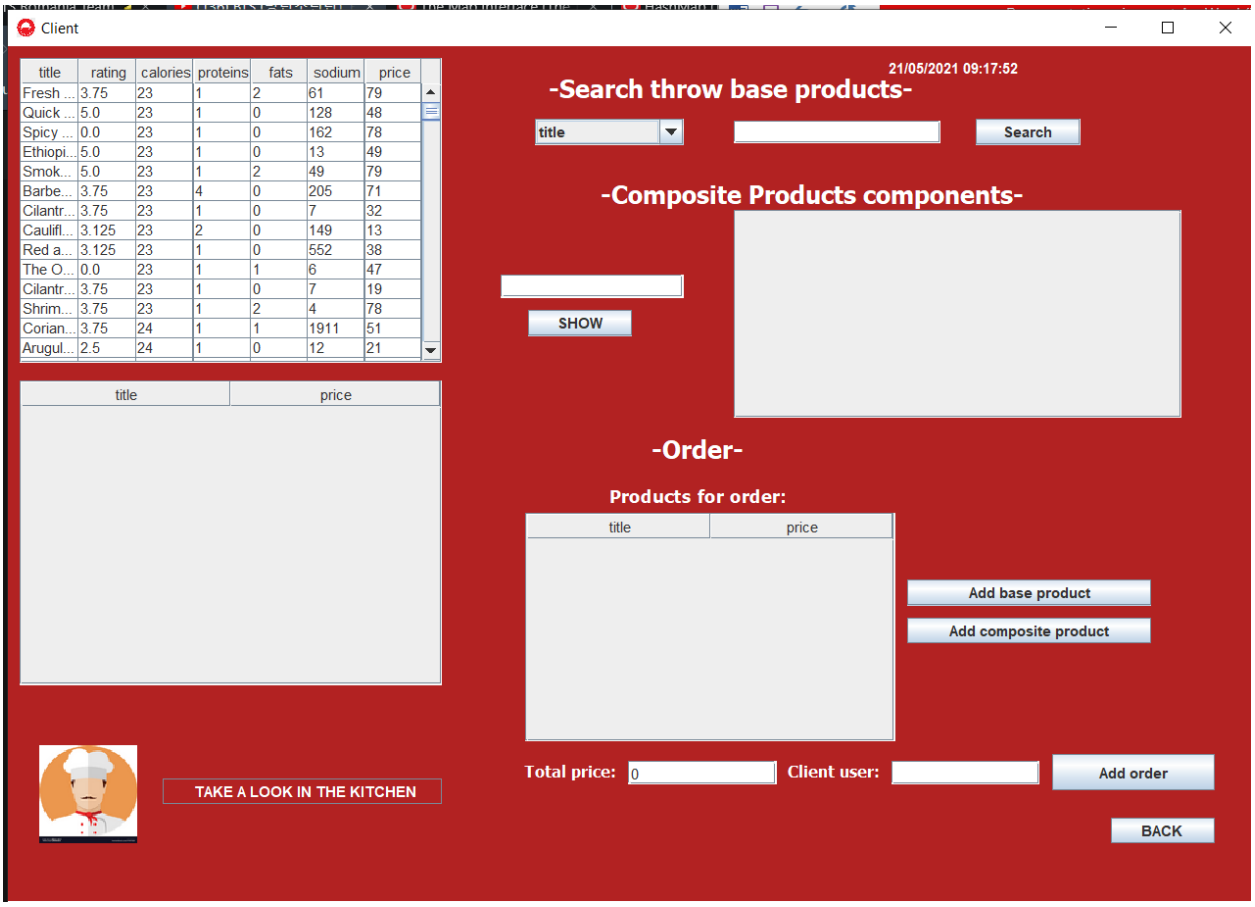


title	rating	calories	proteins	fats	sodium	price
Fresh Corn Tortilla...	3.75	23	1	2	61	79
Quick & Spicy Asi...	5.0	23	1	0	128	48
Spicy Pickled Shal...	0.0	23	1	0	162	78
Ethiopian Spice Te...	5.0	23	1	0	13	49
Smoked Caviar a...	5.0	23	1	2	49	79
Barbecued Shrimp	3.75	23	4	0	205	71
Cilantro-Tomato S...	3.75	23	1	0	7	32
Cauliflower-Leek ...	3.125	23	2	0	149	13
Red and Green To...	3.125	23	1	0	552	38
The Original Three...	0.0	23	1	1	6	47
Cilantro-Tomato S...	3.75	23	1	0	7	19
Shrimp Sates with...	3.75	23	1	2	4	78
Coriander-Herb S...	3.75	24	1	1	1911	51
Arugula Salad with...	2.5	24	1	0	12	21
Beef Stock	4.375	24	4	1	87	26
Red Pepper Sauce	4.375	24	1	0	253	62
Beef Stock	4.375	24	4	1	87	96
"Endive ""Spoons"...	4.375	24	1	2	35	50
Toasted Corn Cris...	2.5	24	0	2	37	66
Classic Salad	4.375	24	2	1	56	49
Beef Stock	5.0	24	4	1	83	61
Master Stock Chic...	3.75	25	2	1	61	72


Administrator
Import
Products
Reports
21/05/2021 09:16:33
-BASE PRODUCTS-
Title
Rating
Calories
Protein
Fat
Sodium
Price
add
modify
delete
title rating calories proteins fats sodium price
Fresh Corn ... 3.75 23 1 2 61 79
Quick & Spi... 5.0 23 1 0 128 48
Spicy Pickle... 0.0 23 1 0 162 78
Ethiopian Spi... 5.0 23 1 0 13 49
Smoked Ca... 5.0 23 1 2 49 79
Barbecued ... 3.75 23 4 0 205 71
Cilantro-To... 3.75 23 1 0 7 32
Cauliflower-... 3.125 23 2 0 149 13
Red and Gre... 3.125 23 1 0 552 38
The Original ... 0.0 23 1 1 6 47
Cilantro-To... 3.75 23 1 0 7 19
Shrimp Sate... 3.75 23 1 2 4 78
Coriander-H... 3.75 24 1 1 1911 51
-COMPOSITE PRODUCTS-
SHOW
title price
Title
Components:
Add the product selected
Add New Composite Product
BACK


Administrator
Import
Products
Reports
21/05/2021 09:16:51
time interval of the orders
first order:
last order:
Generare
the products ordered more than a specified number of times so far
value:
Generare
the clients that have ordered more than a specified number of times
and the value of the order was higher than a specified amount
value for orders:
value for price:
Generare
the products ordered within a specified day with the number of times they have been ordered
date :
Generare
BACK

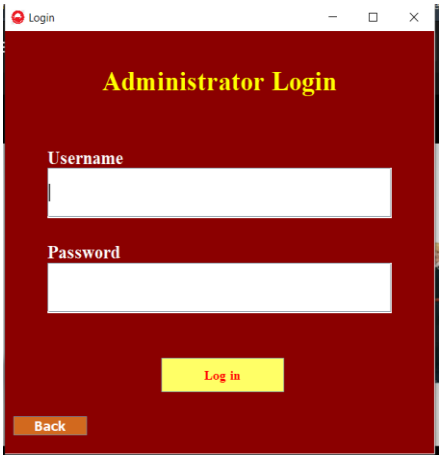

Employee
21/05/2021 09:17:23
order
client
I'm waiting.....
BACK
CLOSE

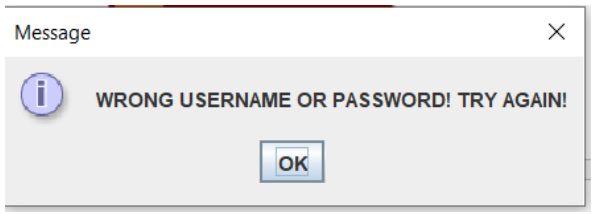

Client
title rating calories proteins fats sodium price
Fresh ... 3.75 23 1 2 61 79
Quick ... 5.0 23 1 0 128 48
Spicy ... 0.0 23 1 0 162 78
Ethiopi... 5.0 23 1 0 13 49
Smok... 5.0 23 1 2 49 79
Barbe... 3.75 23 4 0 205 71
Cilantr... 3.75 23 1 0 7 32
Caulifl... 3.125 23 2 0 149 13
Red a... 3.125 23 1 0 552 38
The O... 0.0 23 1 1 6 47
Cilantr... 3.75 23 1 0 7 19
Shrim... 3.75 23 1 2 4 78
Corian... 3.75 24 1 1 1911 51
Arugul... 2.5 24 1 0 12 21
title price
TAKE A LOOK IN THE KITCHEN
21/05/2021 09:17:52
-Search throw base products-
title
Search
-Composite Products components-
SHOW
-Order-
Products for order:
title price
Add base product
Add composite product
Total price: 0
Client user:
Add order
BACK

4.IMPLEMENTARE + REZULTATE

Cand utilizatorul deschide aplicatia, se importa automat in memorie comenzile anterioare si i se va deschide fereastra:

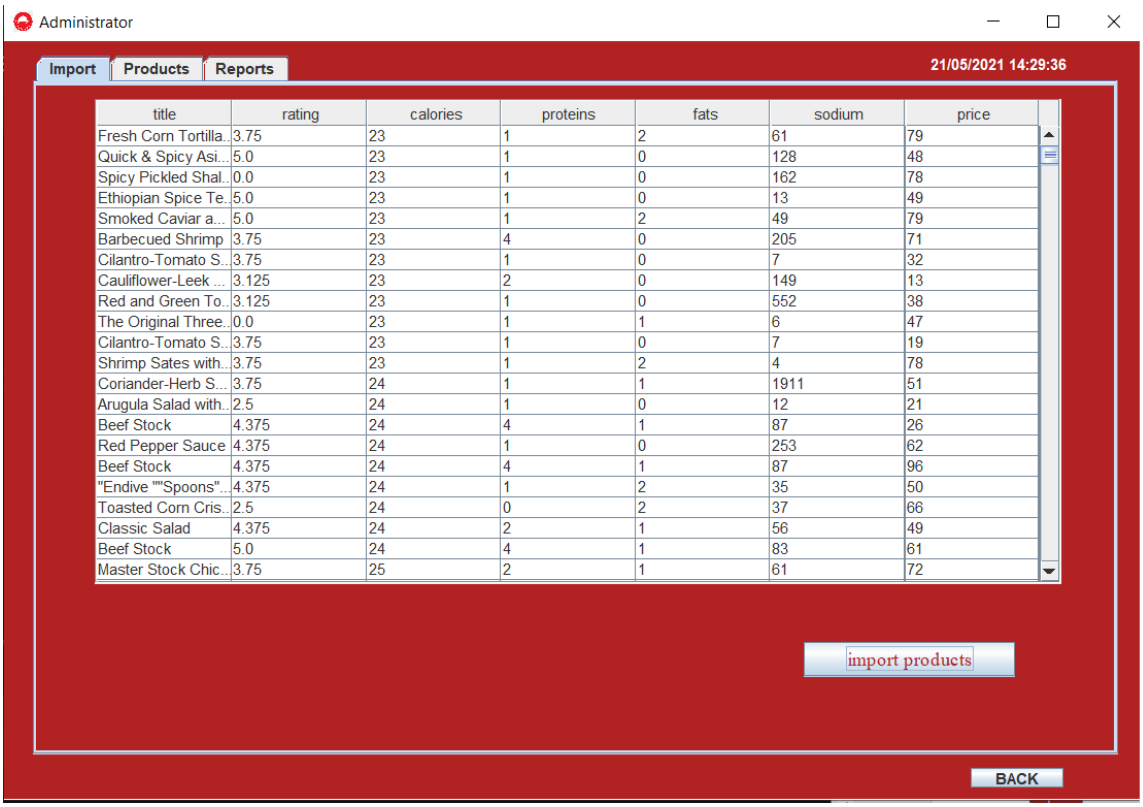
Daca acesta va apasa butonul administrator/employee/client i se va deschide o alta fereastra unde trebuie sa se logheze.





Daca datele introduse de acesta nu sunt corecte ii va aparea o fereastra care il atentineaza si nu i se permite accesul mai departe.

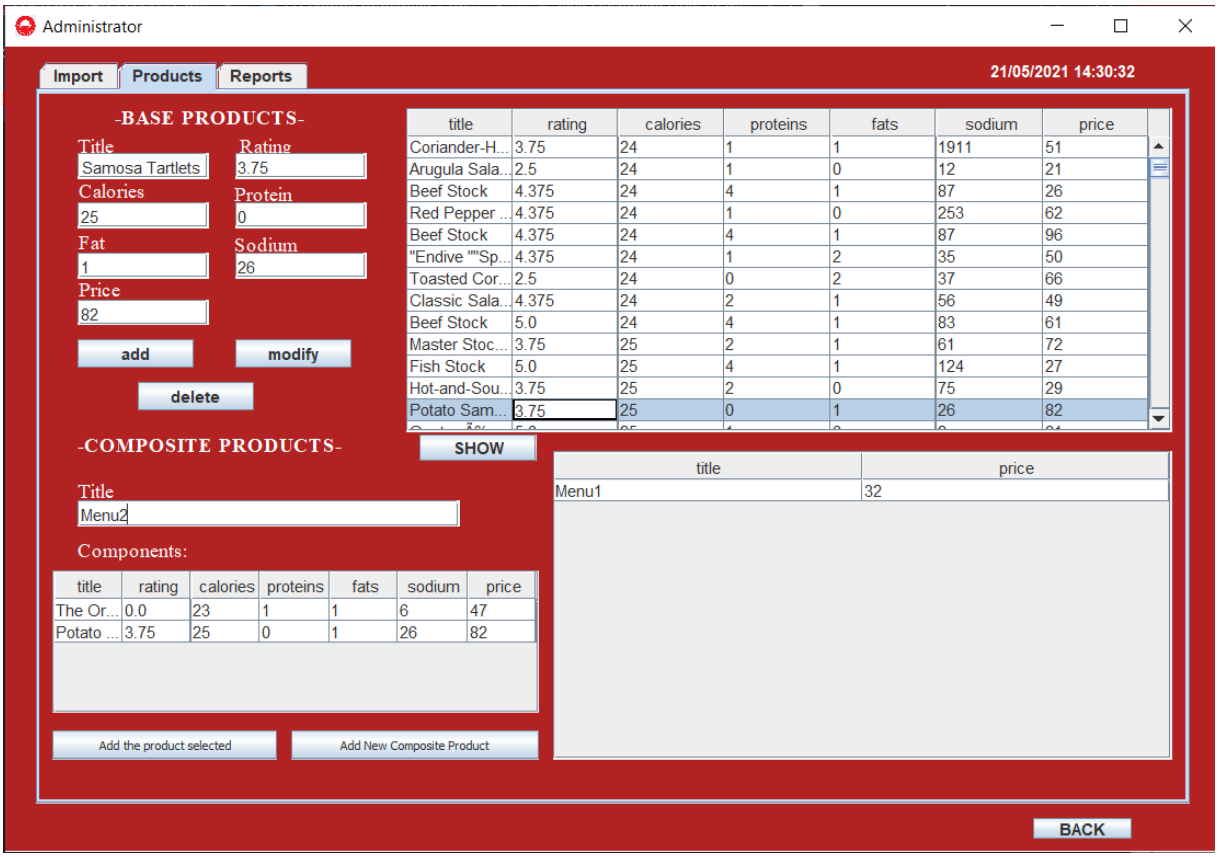
Daca s-a logat ca administrator i se va deschide fereastra. Primul panel este cel de import, acesta va trebui sa apese butonul de import pentru a importa datele din products.csv in sistem.



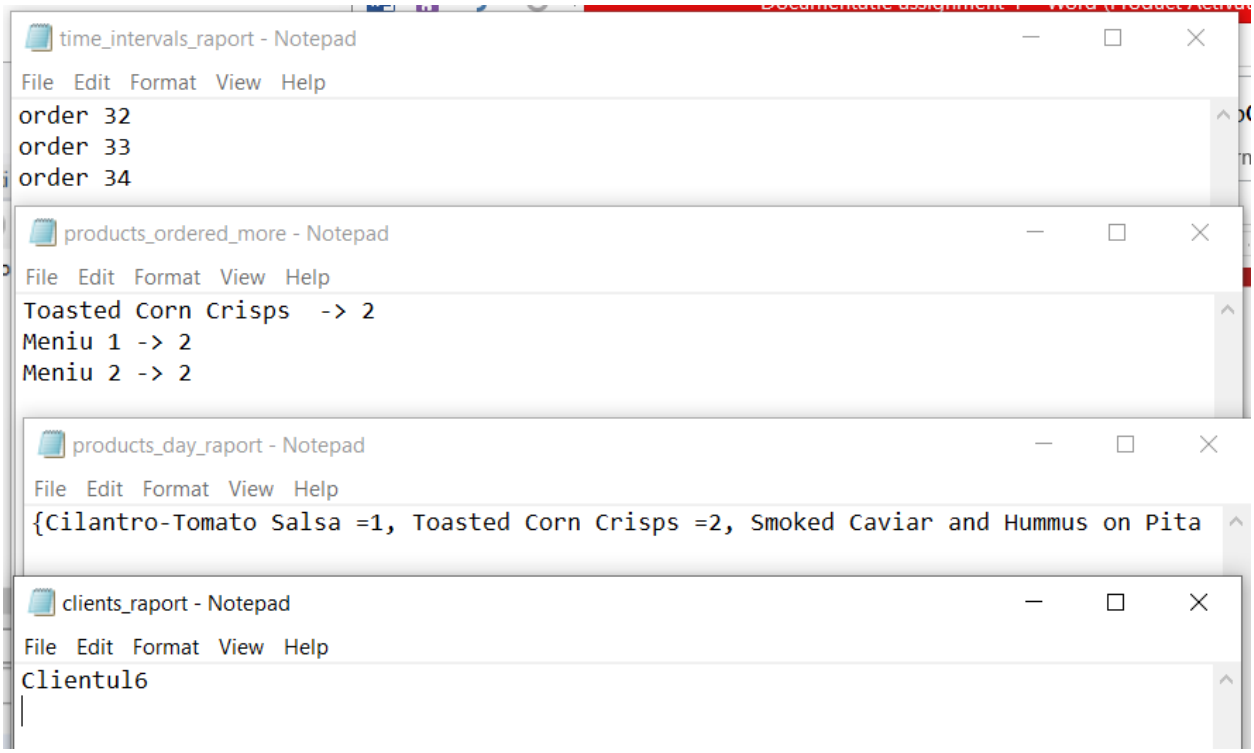
title	rating	calories	proteins	fats	sodium	price
Fresh Corn Tortilla...	3.75	23	1	2	61	79
Quick & Spicy Asi...	5.0	23	1	0	128	48
Spicy Pickled Shal...	0.0	23	1	0	162	78
Ethiopian Spice Te...	5.0	23	1	0	13	49
Smoked Caviar a...	5.0	23	1	2	49	79
Barbecued Shrimp	3.75	23	4	0	205	71
Cilantro-Tomato S...	3.75	23	1	0	7	32
Cauliflower-Leek ...	3.125	23	2	0	149	13
Red and Green To...	3.125	23	1	0	552	38
The Original Three...	0.0	23	1	1	6	47
Cilantro-Tomato S...	3.75	23	1	0	7	19
Shrimp Sates with...	3.75	23	1	2	4	78
Coriander-Herb S...	3.75	24	1	1	1911	51
Arugula Salad with...	2.5	24	1	0	12	21
Beef Stock	4.375	24	4	1	87	26
Red Pepper Sauce	4.375	24	1	0	253	62
Beef Stock	4.375	24	4	1	87	96
"Endive ""Spoons"...	4.375	24	1	2	35	50
Toasted Corn Cris...	2.5	24	0	2	37	66
Classic Salad	4.375	24	2	1	56	49
Beef Stock	5.0	24	4	1	83	61
Master Stock Chic...	3.75	25	2	1	61	72

Daca intra pe panel products acesta poate sa introduca/modifice si sterge date produse din categoria Base-Products, nu poate modifica insa titlul si pretul unui produs, deoarece acestea reprezinta date de diferentiere intre produse (asa am ales). Acesta poate introduce si noi produse din categoria Composite Products selectand din tabelul cu BaseProducts un produs, apasand butonul „add the product selected” , adaugand astfel in lista meniului respectiv produs, la final cand dupa ce a ales produsele componente ale Composite Products, apasa butonul „add new composite product” si acesta e adaugat in sistem.

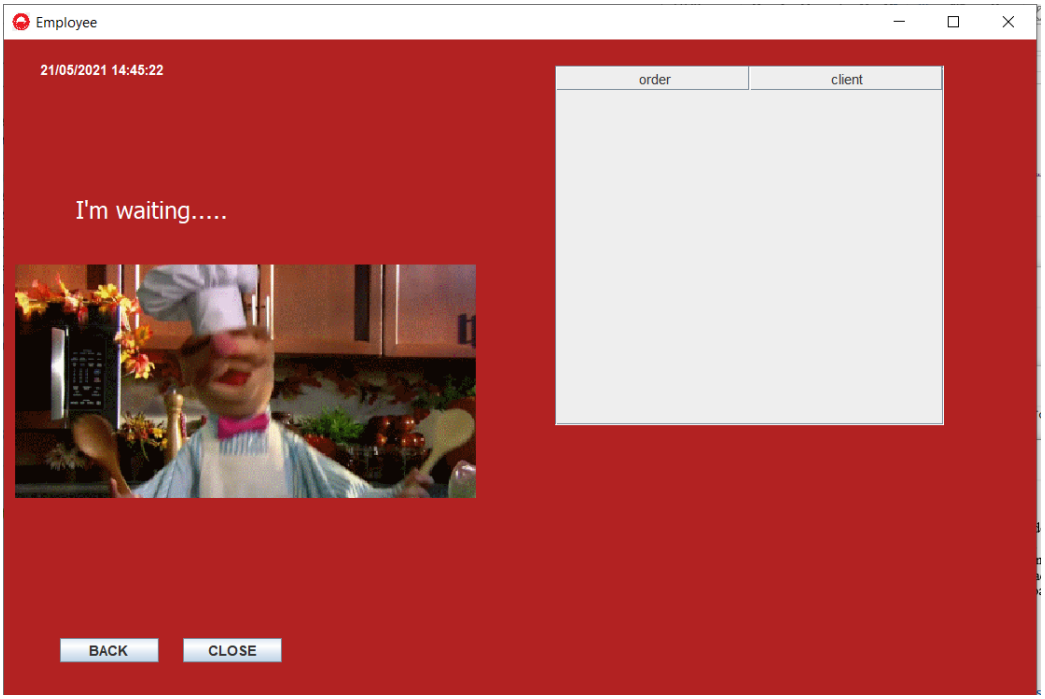
La panel-ul reports acesta poate genera rapoartele , introduce datele in casute si apasa pe butoanele de generare din dreptul casutei unde a introdus datele. Rapoartele se vor genera in fisiere text , `time_intervals_raport.txt, products_ordered_more.txt, clients_raport.txt` si `products_day_raport.txt.`

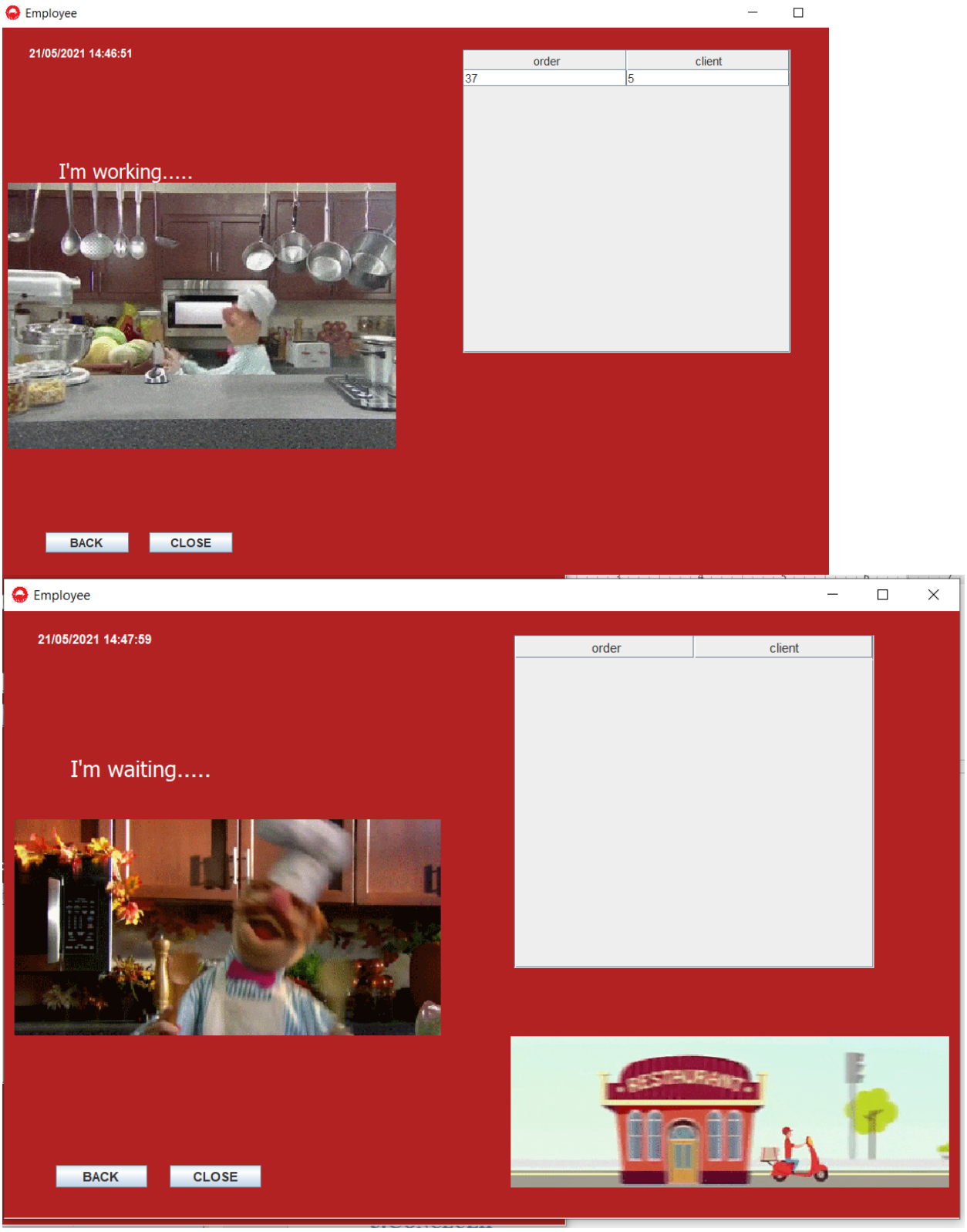

Administrator
Import
Products
Reports
21/05/2021 14:30:32
-BASE PRODUCTS-
Title
Samosa Tartlets
Rating
3.75
Calories
25
Protein
0
Fat
1
Sodium
26
Price
82
add
modify
delete
title rating calories proteins fats sodium price
Coriander-H... 3.75 24 1 1 1911 51
Arugula Sala... 2.5 24 1 0 12 21
Beef Stock 4.375 24 4 1 87 26
Red Pepper ... 4.375 24 1 0 253 62
Beef Stock 4.375 24 4 1 87 96
"Endive ""Sp... 4.375 24 1 2 35 50
Toasted Cor... 2.5 24 0 2 37 66
Classic Sala... 4.375 24 2 1 56 49
Beef Stock 5.0 24 4 1 83 61
Master Stoc... 3.75 25 2 1 61 72
Fish Stock 5.0 25 4 1 124 27
Hot-and-Sou... 3.75 25 2 0 75 29
Potato Sam... 3.75 25 0 1 26 82
-COMPOSITE PRODUCTS-
SHOW
Title
Menu2
Components:
title rating calories proteins fats sodium price
The Or... 0.0 23 1 1 6 47
Potato ... 3.75 25 0 1 26 82
Add the product selected
Add New Composite Product
title price
Menu1 32
BACK


Administrator
Import
Products
Reports
21/05/2021 14:35:38
time interval of the orders
first order:
last order:
Generare
the products ordered more than a specified number of times so far
value:
Generare
the clients that have ordered more than a specified number of times
and the value of the order was higher than a specified amount
value for orders:
value for price:
Generare
the products ordered within a specified day with the number of times they have been ordered
date :
Generare
BACK

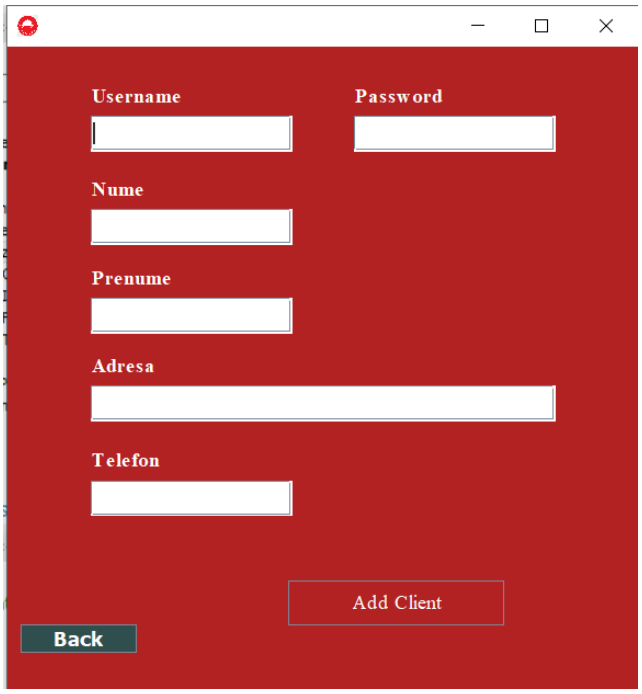


Daca se logeaza ca angajat, i se va deschide o fereastra ce ilustreaza evolutia comenzilor, avem un tabel unde se adauga comenzile in coada. Cat timp nu sunt comenzi se va afisa un textbox „I'm waiting" si o imagine sugestiva, iar daca sunt comenzi atunci se va afisa „I'm working" si alta imagine. In momentul cand o comanda e eleminata din coada (prepararea ei s-a incheiat) atunci va aparea o imagine care ilustreaza livrarea unei comenzi. Coada de comezi este actualizata continuu.

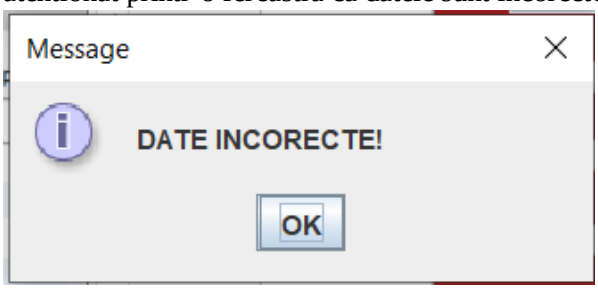


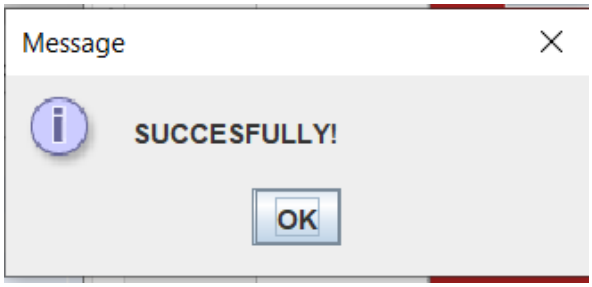


Pe fereastra de logare a Clientului apare si un buton de creare de cont pentru un client.

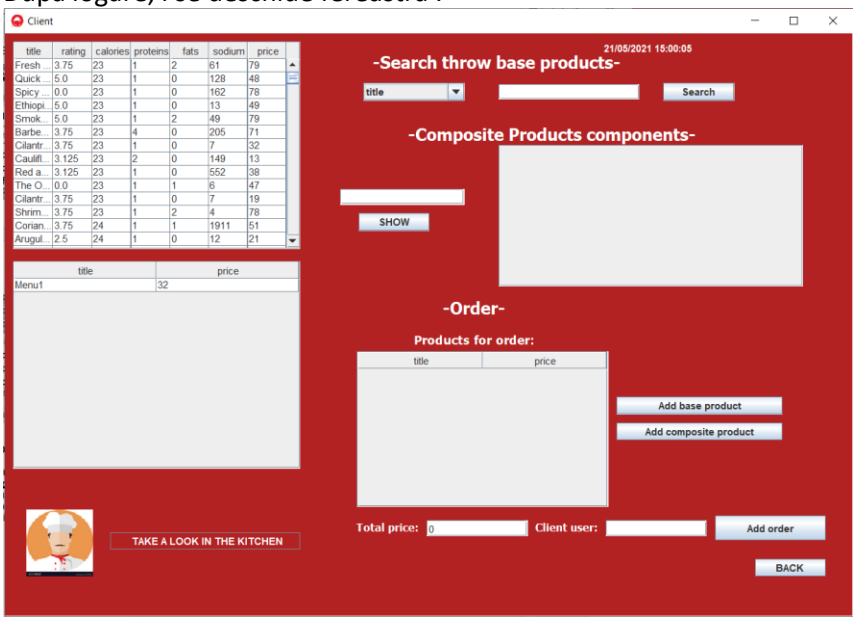


Dupa ce isi adauga datele si apasa Add Client acestea se vor adauga automat in tabele din baza de date si i se afiseaza o fereastra cu mesajul „Succesfully!”. Daca nu a introdus datele si apasa butonul este atentionat printr-o fereastra ca datele sunt incorecte.



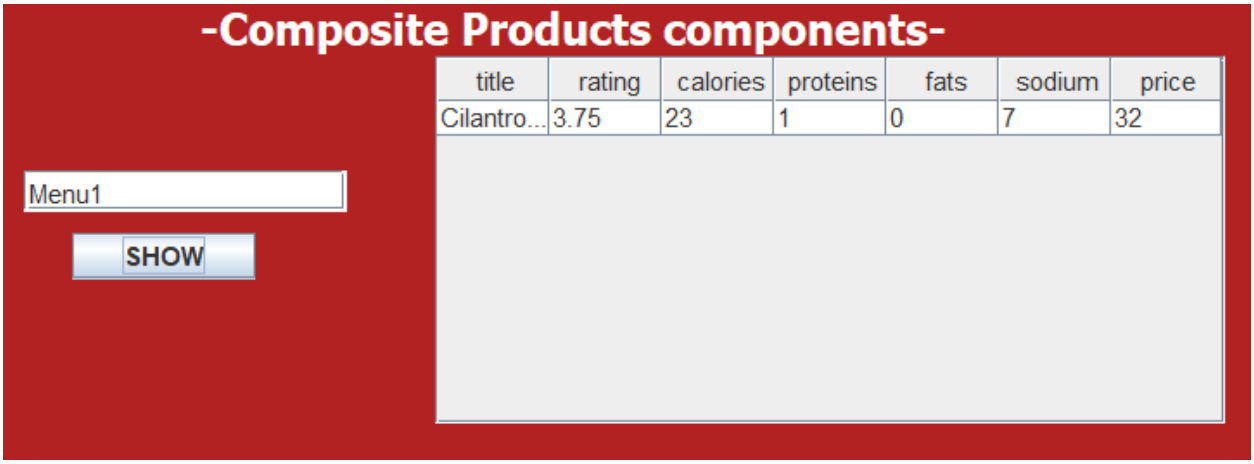


Dupa logare, I se deschide fereastra :

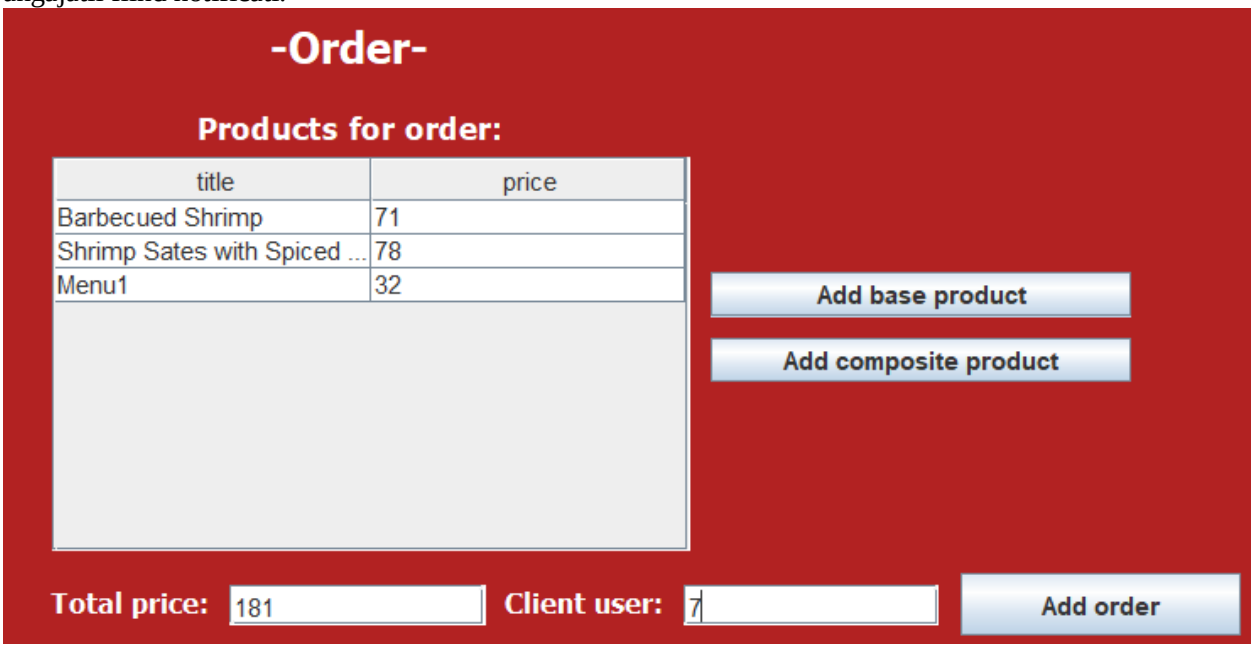


Acesta poate vizualiza produsele, grupate in doua tabele, in functie de cele 2 tipuri. Poate cauta BaseProducts dupa anumite criterii, alegand din categoria din comboBox, introducand o valoare in casuta si apasand Search.

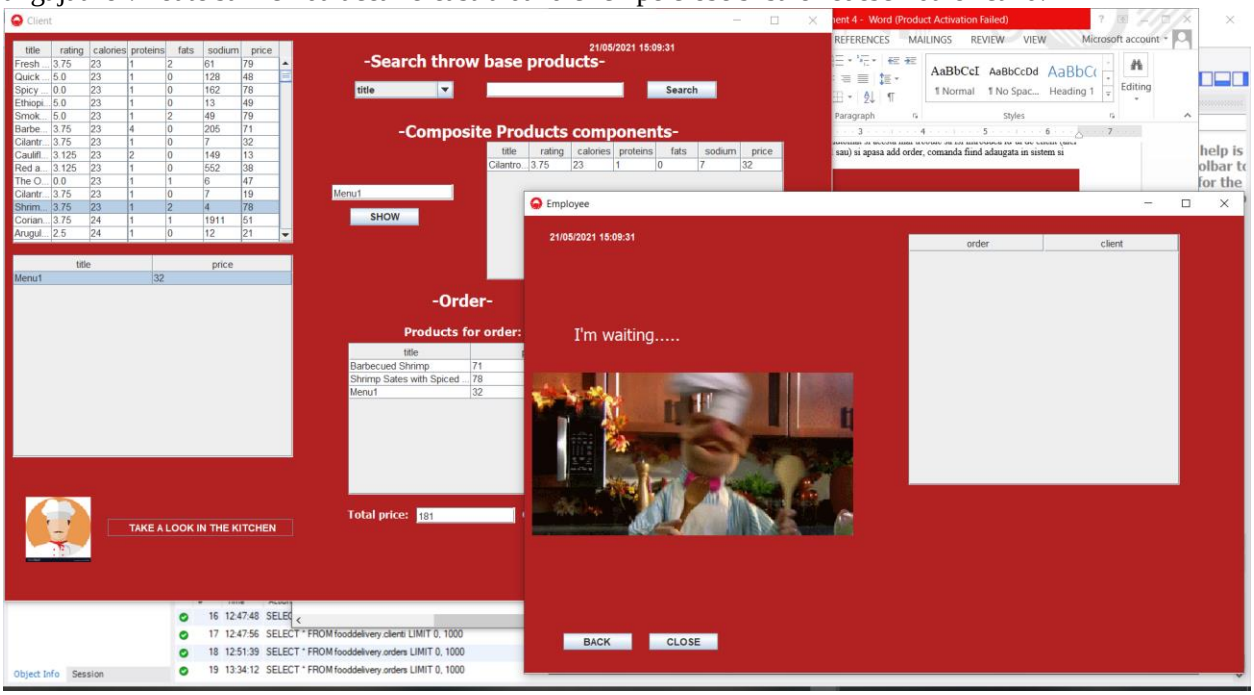
Acesta poate vedea ce produse contine un meniu, selectand un CompositeProduct din tabel si apasand SHOW.



La partea de adaugare comanda, acesta isi selecteaza din fiecare tabel cate un produs pe rand si il introduce in „cos”, pretul final e calculat automat si acesta mai trebuie sa isi introduca id-ul de client (aici se presupune ca fiecare client isi stie id-ul sau) si apasa add order, comanda fiind adaugata in sistem si angajatii fiind notificati.



Pentru a vedea evolutia comenzii in timp real evolutia prepararii comenzi, clientul poate apasa pe TAKE A LOOK IN THE KITCHEN si i se va deschide fereastra ce ilustreaza evolutia comenziilor, adica cea a angajatilor. Poate sa inchida acea fereastra dand click pe close si sa o redeschida oricand.



La adaugarea unei comenzi, se genereaza automat si un bon, cate unul pentru fiecare comanda.
Un exemplu de bon e:

5.CONCLUZII

Prin urmatorii pasi:

- Stabilirea problemei
- Implementare clase si metode cat mai divizat
- Trasare interfata grafica
- Schitarea diagramei UML
- Testare

Am reusit sa implementez cerinta temei.

Ca o dezvoltare ulterioara a aplicatiei, ar fi o modifcare a serializatorului, deoarece cand am mai multe comenzi in coada update() se face din ce in ce mai greu, adica dureza mai mult.

S-ar putea optimiza si interfata angajatului, avand mai multe optiuni si sa i se ofere posibilitatea de a efectua si el diferite operatii. S-ar putea modifica ca fiecare produs sa contina si un camp ce ar reprezenta valoarea timpului de preparare. In programul meu fiecare produs are un timp de preparare de 30 de secunde. Ar putea fi pastrate si Composite Products in fisiere diferite, ca sa nu fim nevoiti ca de fiecare data sa cream Composite Products daca vrem sa aceste tipuri de produse in lista pentru comenzi.

6.BIBLIOGRAFIE

- ASSIGNMENT_4_SUPPORT_PRESENTATION
- Tema_4
  https://www.geeksforgeeks.org/stream-in-java/
  https://www.geeksforgeeks.org/serialization-in-java/
  https://www.geeksforgeeks.org/assertions-in-java/