**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

**DEPARTAMENTUL CALCULATOARE**

***Tehnici de programare fundamentale***

***Tema 4 – Food Delivery Management System***

Matioc Bogdan Darius, 30224

Profesor îndrumător: David Sera

Contents

[1. Obiectivul temei 3](#_Toc72481780)

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 3](#_Toc72481781)

[2.1 Analiza problemei 3](#_Toc72481782)

[2.2 Modelare 3](#_Toc72481783)

[2.3 Scenarii de utilizare 4](#_Toc72481784)

[3. Proiectare (decizii de proiectare, diagrame UML, structuri de date, proiectare clase, interfete, relatii, packages, algoritmi, interfata utilizator) 4](#_Toc72481785)

[4. Implementare 5](#_Toc72481786)

[4.1 Pachetul businessLogic 5](#_Toc72481787)

[4.1.1 Pachetul account 5](#_Toc72481788)

[4.1.2 Pachetul bill 5](#_Toc72481789)

[4.1.3 Pachetul menu 5](#_Toc72481790)

[4.1.4 Pachetul order 5](#_Toc72481791)

[4.1.5 Pachetul user 5](#_Toc72481792)

[4.2 Pachetul dataAccess 6](#_Toc72481793)

[4.2.1 Pachetul fileWork 6](#_Toc72481794)

[4.3 Pachetul Exceptions 6](#_Toc72481795)

[4.4 Pachetul Model 6](#_Toc72481796)

[4.4.1 Pachetul dto 6](#_Toc72481797)

[4.4.2 Pachetul entities 6](#_Toc72481798)

[4.5 Pachetul presentationLayer 7](#_Toc72481799)

[4.5.1. Pachetul controller 7](#_Toc72481800)

[4.5.2 Pachetul view 7](#_Toc72481801)

[5. Rezultate 22](#_Toc72481802)

[6. Concluzii 23](#_Toc72481803)

# Obiectivul temei

Obiectivul acestei teme este de a realiza o aplicație pentru a comanda mâncare. Aplicația trebuie să aibă multiple funcționalități în funcție de tipul utilizatorului care o folosește, acesta putând fii fie client, fie angajat, fie administrator.

Obiectivele secundare care fac posibilă realizarea unei astfel de aplicații sunt:

* Definirea claselor corespunzătoare
* Realizarea interfeței grafice
* Preluarea datelor necesare din interfața grafică
* Actualizarea tuturor modificărilor realizate supra datelor prin intermediul interfeței grafice
* Testarea și verificarea corectitudinii implementării realizate

# Analiza problemei, modelare, scenarii, cazuri de utilizare

## Analiza problemei

În primul rând, o să încep prin a descrie ce presupune și ce ar trebui să facă mai exact o astfel de aplicație de comandat mâncare. Această aplicație poate să fie folosită de către trei tipuri diferite de utilizatori. Unul dintre aceste trei tipuri este clientul. Prin intermediul acestei aplicații acesta are posibilitatea de a plasa o comandă. Poate să vizualizeze toate tipurile de produse existente și toate tipurile de meniuri existente. De asemenea, poate să caute un anume produs pe baza anumitor criterii cum ar fi numele, prețul, caloriile, grăsimile etc. Un alt tip de utilizator este administratorul. Acesta are posibilitatea de a adăuga noi produse și de a crea meniuri compuse din diverse produse. Poate, de asemenea, să editeze produsele sau să le șteargă. Acesta mai poate și să genereze un raport care conține detalii despre comenzile care au fost plasate, produse și clienți. Al treilea tip de utilizator posibil este angajatul. Acesta poate să vizualizeze o listă cu toate comenzile de până acum și, de asemenea, acesta este notificat de fiecare dată când apare o comandă nouă.

## Modelare

Din punct de vedere al modelării, se pot distinge două părți principale. Prima parte este reprezentată de interfața grafică. Utilizatorul interacționează în mod direct cu aceasta. Interfața este ușor de folosit și sugestivă, din punct de vedere al funcționalității pe care o are. Poate insera date noi, le poate șterge și, de asemenea, le poate modifica și actualiza. Cea de-a doua parte este reprezentată de partea de cod aflată în spatele interfeței, care face posibilă întreaga funcționalitate a programului și înglobează algoritmii necesari pentru toate operațiile.

## Scenarii de utilizare

Descriere caz de utilizare:

Dat fiind faptul că această aplicație permite a fi utilizată de către trei tipuri distincte de utilizatori, o să descriu câte un caz de utilizare pentru fiecare dintre cele trei tipuri: client, administrator și angajat.

# Proiectare (decizii de proiectare, diagrame UML, structuri de date, proiectare clase, interfete, relatii, packages, algoritmi, interfata utilizator)

Una dintre deciziile de proiectare pe care le-am luat vizavi de această aplicație constă în structurarea ei așa încât să respecte layered architecture. Prin prisma acestei structurări proiectul este bine delimitat din punct de vedere al claselor, delimitare care se bazează pe funcționalitatea pe care o au. Prin această structurare proiectul este mai lizibil și mai coerent din punct de vedere logic.

Algoritmii care stau la baza acestei aplicații de comandare de mâncare sunt aceia prin prisma cărora se implementează funcționalitățile de bază aplicației, cum ar fi algoritmii care realizează: adăugarea unui nou produs, ștergerea unui produs, editarea unui produs, căutarea unui produs după anumite criterii, plasarea unei comenzi, crearea de meniuri (produse compuse).

Interfața utilizator este ușor de folosit și intuitivă din punct de vedere al funcționalității. Aceasta permite personalizarea datelor de intrare și urmărirea rezultatului fără probleme.

# Implementare

Clasele cuprinse în acest proiect prin intermediul cărora se realizează funcționalitatea aplicatei sunt organizate (comform layered architecture) in urmatoarele pachete:

## Pachetul businessLogic

Pachetul contine intreaga logica a proiectului, de la logare, la crearea chitantei pana la geberarea de rapoarte, crearea de meniuri etc.

Pachetul contine Interfata BusinessLogic care este foarte utila deoarece prin implementarea ei se asigura prin contract implementarea metoda findAll(), acest aspect este foarte important in momenul descrieri clasei InitializitionTabel, o clasa abstracta si generica care contine doar functii generice si care are ca scop crearea unui format pentru tabele si popularea lor, eventual adaugarea de ascultatori pentru evenimente. Metoda findAll() este strict necesara pentru popularea tabelelor.

O alta clasa in pachet este clasa Download care se ocupa de crearea unui fisier pe dispozitivul propriu.

Pachetul contine la randul sau alte 5 subpachete pentru a se grupa cat mai clar logica.

### Pachetul account

Aici se afla logicica legata de conturile utilizatotilor, mai exact logica pentru logare sau pentru crearea unui nou cont.

### Pachetul bill

Contine clasa BillLogic care are ca scop reproducerea unui string cat mai apropiat de o chitanta reala.

### Pachetul menu

Concentreaza logica legata de meniuri si produse. In clase gasim implementarea metode de gestonare a cestor obiecte precum: adaugare, stergere, actualizare sau filtrare.

### Pachetul order

Este constituit din clasa OrderLogic care contine metode de generare de rapoarte si plasare de comenzi.

### Pachetul user

Concentreaza logica asupra utilizatorilor.

## Pachetul dataAccess

Se ocupa de stocarea si incarcarea datelor din / in fisiere, cat si de diferite operatii: adaugare, stergere, actualizare.

Pachetul se imparte in 2 subpachete.

### Pachetul fileWork

Se ocupa de stocarea si incarcarea datelor din / in fisiere.

#### Pachetul csvFile

Descrie operatii orientate pe fisiere „ .csv”.

##### onGenricObjects

Operatiile descrise in acest pachet sunt generice.

#### Pachetul operation

Descriu operatii un pic mai complexe decat o simpla scriere sau citire asupra continutului fisierelor.

##### onGenricObjects

Prezinta o abordare generica a operatiilor.

##### onCostumObjects

Restring operatiile la un anumit tip.

## Pachetul Exceptions

Pachetul contine clase care descriu diverse exceptii care pot apararea in rularea aplicatiei.

## Pachetul Model

Pachetul contine clase care descriu mai multe obecte sau tipuri de utilizatori din lumea reala.

Se imparte in 2 subchete (tipic Spring):

### 4.4.1 Pachetul dto

Grupeaza clase dto utile in transferul de date intre niivelul de logica si de nivelul de control

### 4.4.2 Pachetul entities

Contine clase folosite pentru stocarea datelor citite din fisiere.

## Pachetul presentationLayer

Aici vom gasi tot ce intra direct clasele si fxml-urile care intra direct cu utilizatorul.

### 4.5.1. Pachetul controller

Aici vom regasi descrierea pentru tot controlul aplicatiei: de la crearea de ascultatori la schibarea de scene sau initializarea de tabele.

#### Pachetul sceneControllers

Descrie controlul pentru scenele descrise in pachetul view.

Se stuctureaza in urmatoarele subpachete:

##### 4.5.1.1.1 Pachetul account

##### 4.5.1.1.2 Pachetul admin

##### 4.5.1.1.3 Pachetul client

##### 4.5.1.1.4 Pachetul employee

##### 4.5.1.1.5 Pachetul bill

#### 4.5.1.2 Pachetul switchScene

Avem logica pentru a schimba scenele intre ele. Aveam logica pentru meniul de control.

##### 4.5.1.2.1 Pachetul account

##### 4.5.1.2.2 Pachetul admin

##### 4.5.1.2.3 Pachetul client

##### 4.5.1.2.4 Pachetul employee

##### 4.5.1.2.5 Pachetul bill

#### 4.5.1.3 Pachetul tableInitialization

Contine descrierea logici pentru crearea de tabele si populaea lor. Regasim si logica de implementare a logici de adagare de ascultatori de evenimente.

### Pachetul view

Pachetul contine doar fisiere fxml prin care se costruieste interfața grafică cu care interactioneaza utilizatorii.

Prima fereastră de mai sus reprezintă fereastra de autentificare în aplicație. Utilizatorul poate să fie de trei tipuri, client, administrator sau angajat, așadar acesta trebuie să introducă numele de utilizator și parola corespunzătoare. În funcție de datele introduse o să se deschidă una din trei ferestrele: AdminHome ( daca utilizatorul este administretor ), ClientShop ( daca utilizatorul este client ), ProfileEmployee ( daca utilizatorul este angajat). In caz ca utilizatorul nu are un client acesta isi poate crea unul prin intermendiul interfetei „Create Account” un cont de client. La creare se solicita anumite date, iar prin apasarea unui buton se creaza un nou cont, in caz favorabil, iar in caz defavorabil se afiseaza un mesaj de eroare corespunzator.

Client:

Utilizatorul de tip Client are 3 optiuni odata logat in aplicatie.

Ca prima optiune este de a crea o comanda. Aceasta optiune este disponibila prin scena „Shop”. Interfata contine 2 tabele care reprezinta: cel din stanga este tabelul care contine optiunile din care clientul poate comanda, atat produse simple cat si meniuri compuse din mai multe multe produse, iar tabelul din dreapta este o reprezentare a cosului de cumparaturi. Remarcam prezenta a doua buroane „>>” si „<<” . Prin apasarea butonului „>>” se adauga elementul selectat in tabelul produselor valabile in cosul de cumparaturi, in caz ca exista un produs selectat, altfel nu se produce nicio schimbare. Butonul „<<” prin apasre elimina produsul selectat din cosul de cumparaturi, in caz ca este selectat un produs din cosul de cumparaturi, altfel nu se intampla nimic. Aceste 2 operatiuni actualizeaza constant pretul produselor din cosul de cumparaturi. De asemenea pentru a usura munca clientul remarcam ca avem textFieldul in care putem introduce un text, iar prin apasarea butonului search (reprezentat printr-o lupa) in tabelul produselor valabila vom regasi doar produsele care contin text scris anterior. La final mai ramane doar butonul „place order” pentru a finaliza comanda. In cazul in care cosul de cumparaturi este gol se va afisa un mesaj, altfel se va deschide o noua schena cu chitanta. Scena contine putonul „download prin care puten descarca chitanta pe propriu desktop.

Clientul are posibilitate de a-si vedea informatile contului in interfata „Profile”.

De asemenea clientul poate vedea un istoric al cumparaturilor efectuate. (Intr-o versiune actualizata aici va fi posibil si vizualizarea statusului: plasata, in lucru, in livare, ridicata).

Administrator:

Intefata „Home” ( AdminHome ) prezinta prin intermediul a patru butoane optiunile: „UpdateStock”, „UpdateMenu”, „CreateMenu”, „Create Reports”.

Interfata „UpdateMenu” implementeaza optiunile de a sterge si de a actualiza meniuri. Actualizarea se face prin modificarea datelor in tabel si apasarea tastei enter. Stergerea se face prin selectarea uneui produs si apasarea butonului „Delete”.

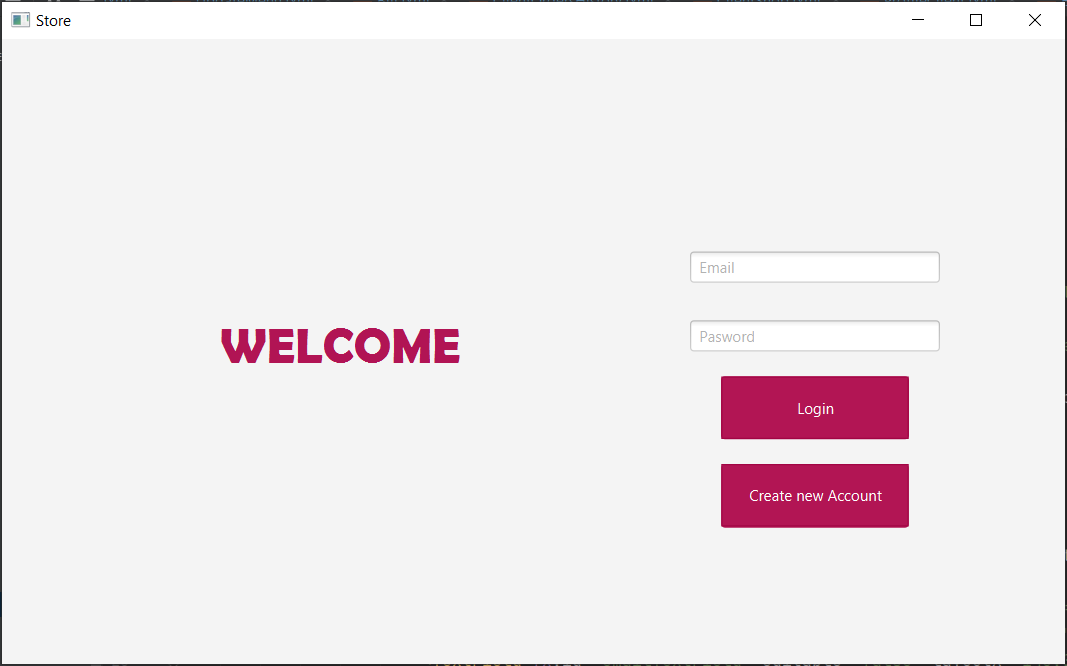
Interfata „UpdateStock” implementeaza optiunile de a sterge, de a actualiza si de a adauga produse din/in stock. Actualizarea si stergerea se efectueaza asemanator cu cea a meniurilor. Adaugarea unui produs se face prin completarea TextFieldurilor din partea inferioara a ferestrei si apasarea butonului „Add”.

Interfata „CreateMenu” ofera posibilitatea administratorului de a crea un nou meneu. Aceasta interfata este asemanatoare cu cea a clientului pentru a plasa comenzi.

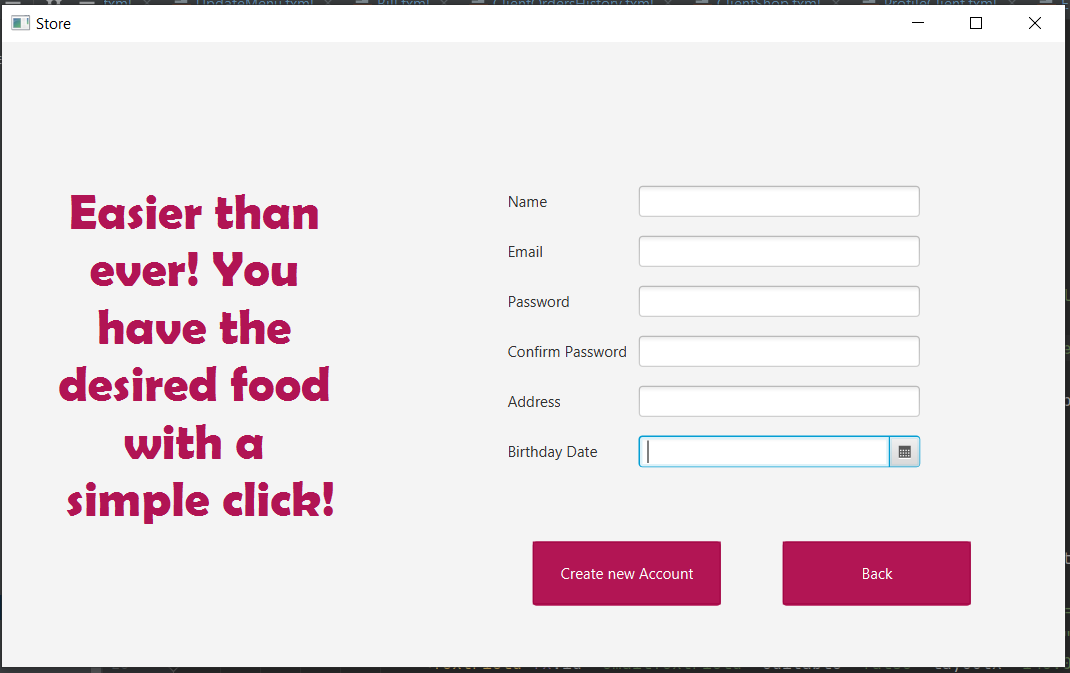
Interfata „Create Report” ofera possibilitatea administratorului de a crea diverse rapoarte. Aceste rapoarte vor fi vizualizate in interfata „Report”. Administratorul are optiunea de a descarca aceste rapoarte.

##### Pachetul account

* Login

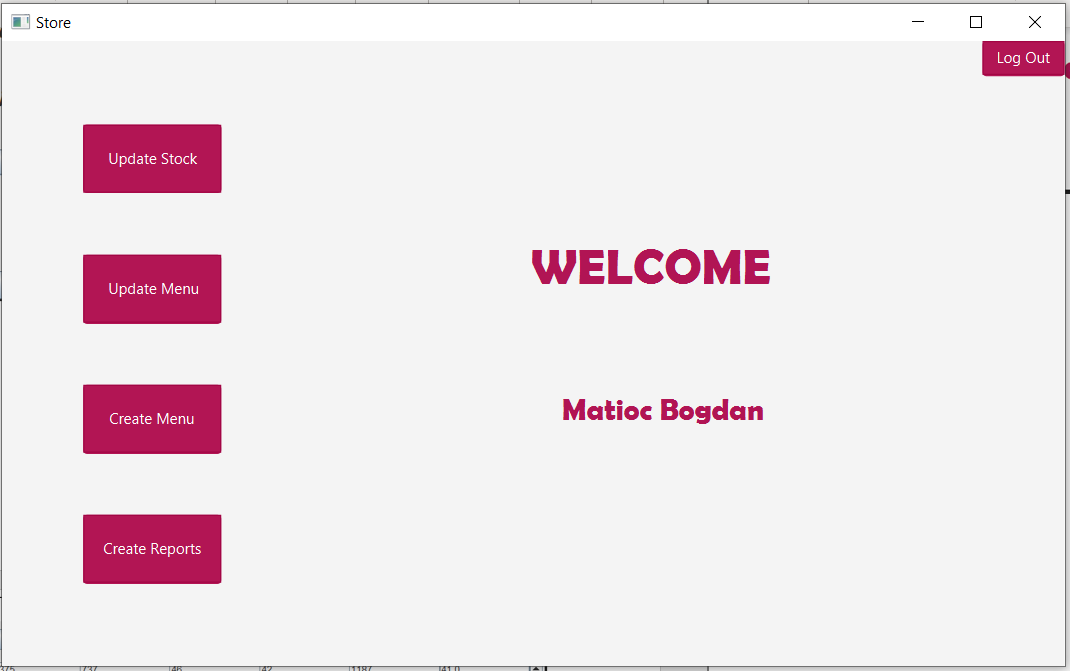


* CreateAccount

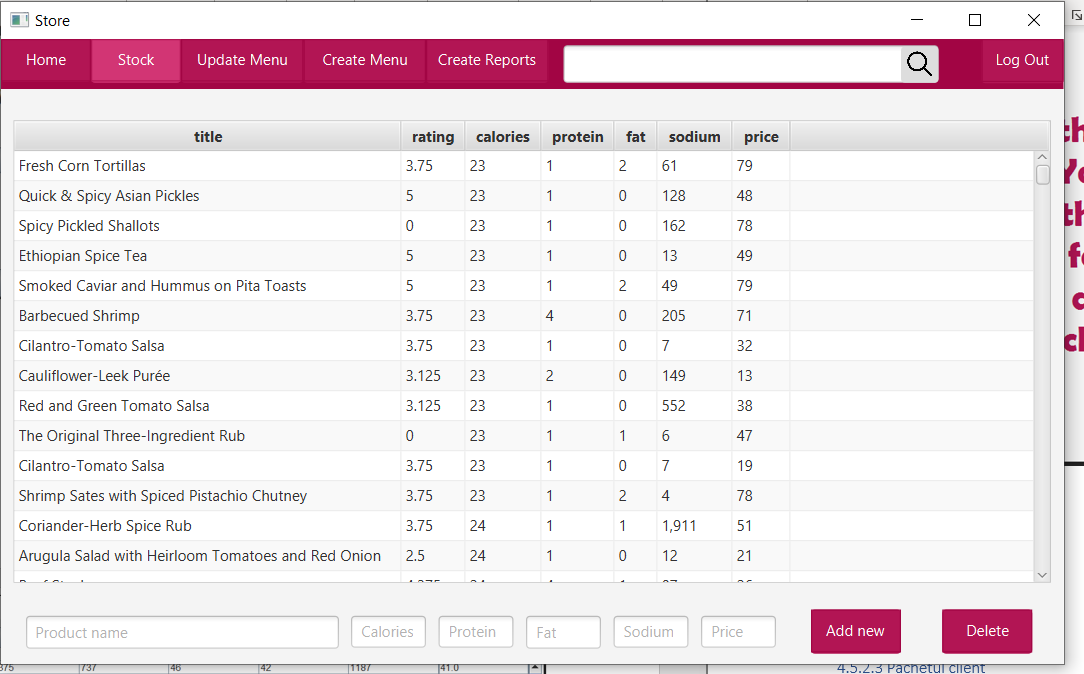


##### Pachetul admin

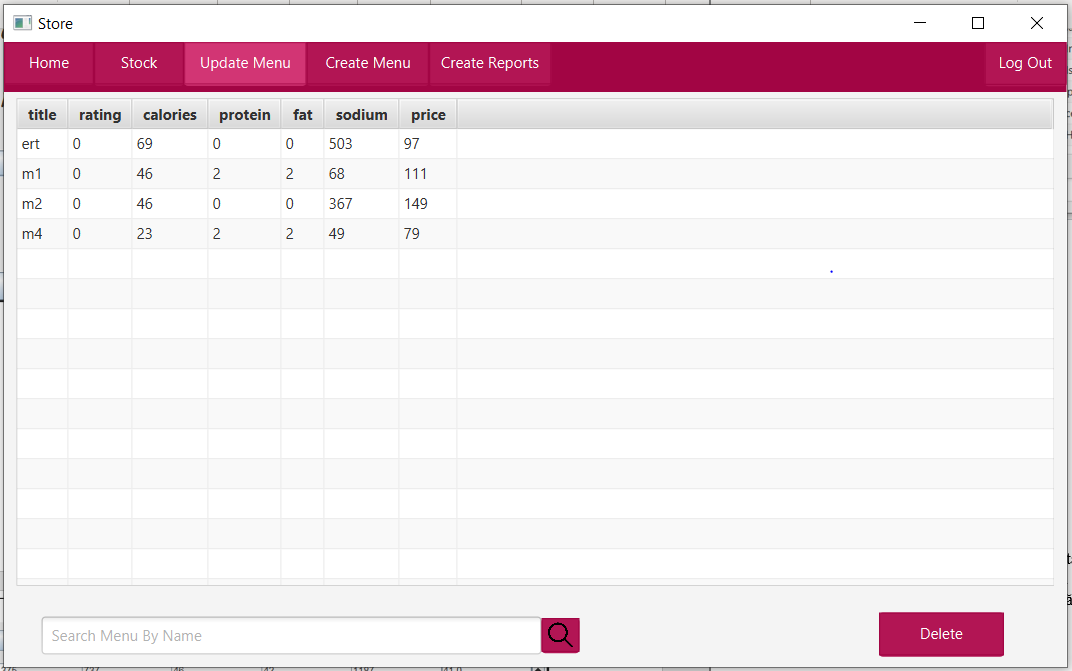
* AdminHome



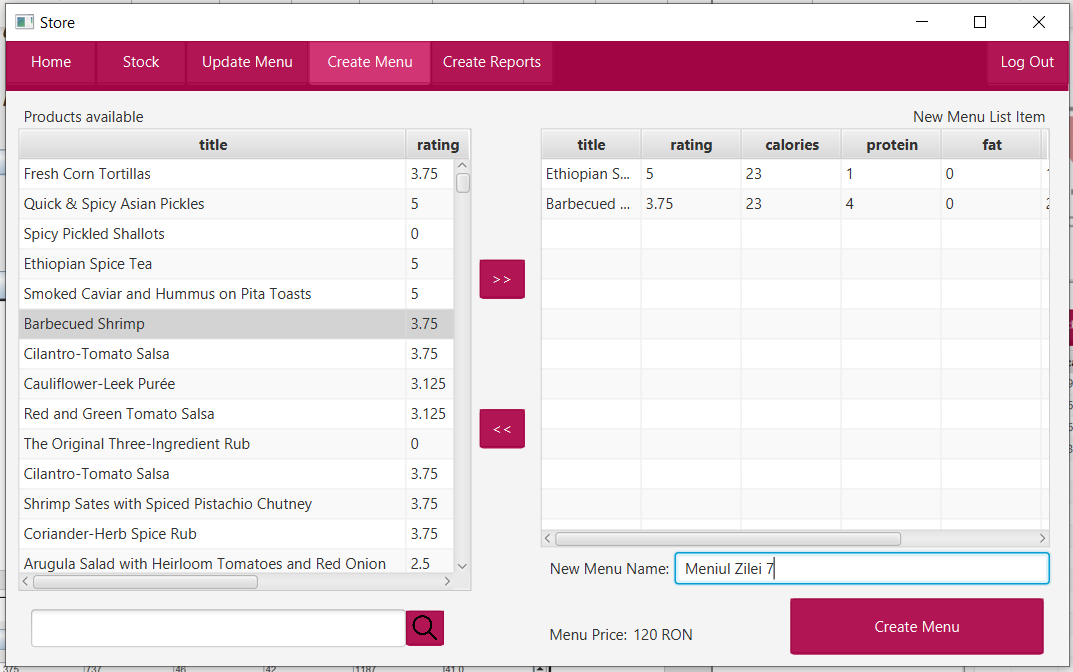
* Stock



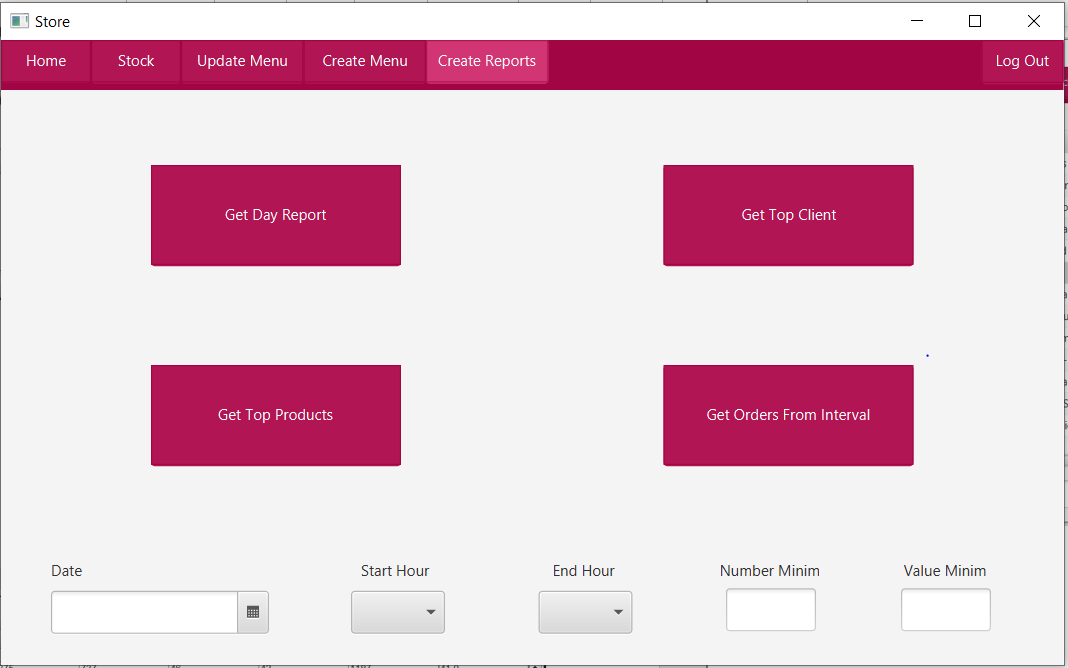
* UpdateMenu



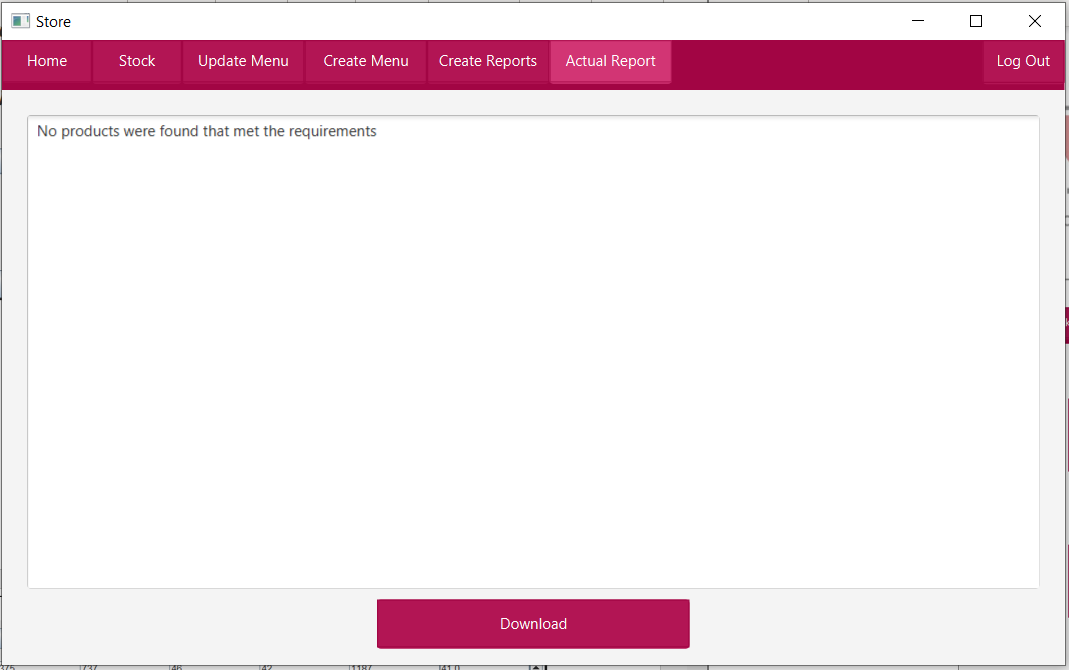
* CreateMenu



* CreateReports

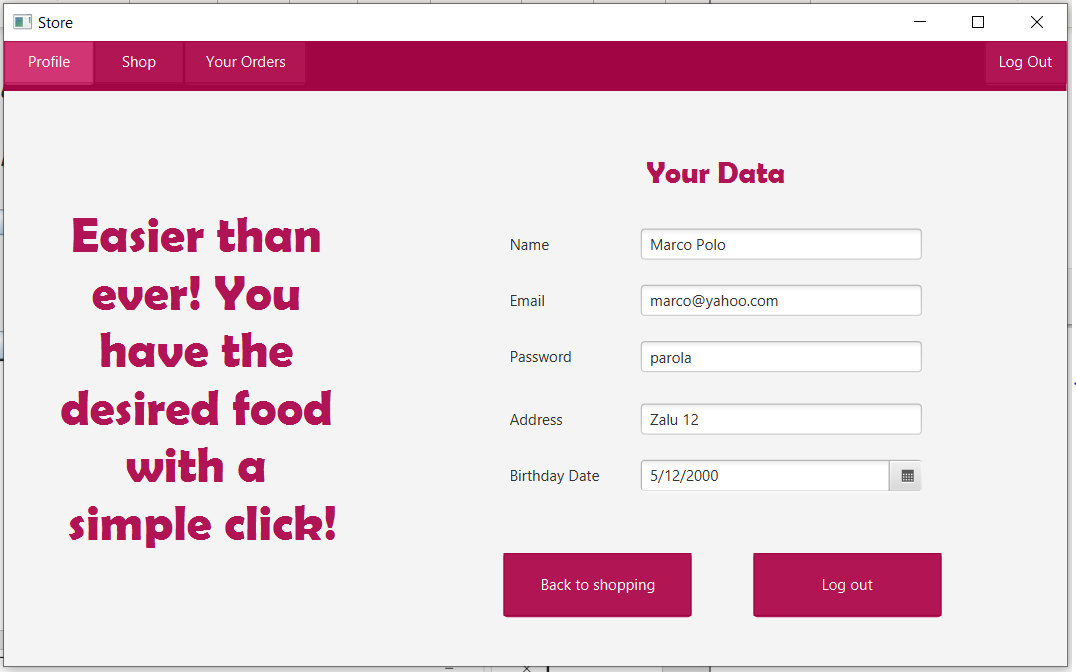


* Report

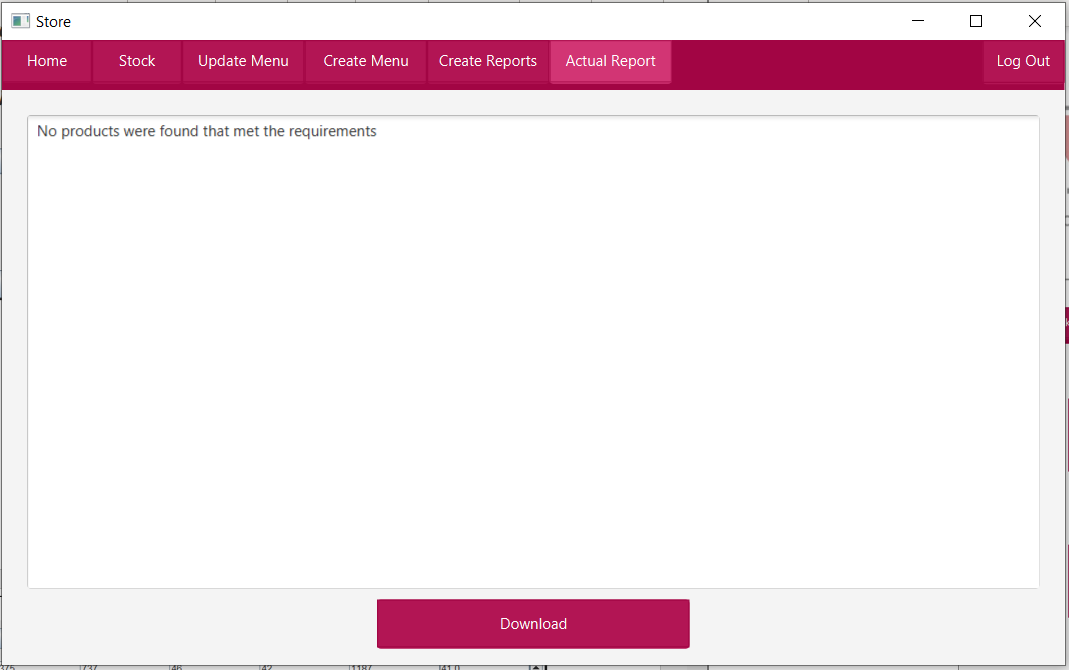


##### Pachetul client

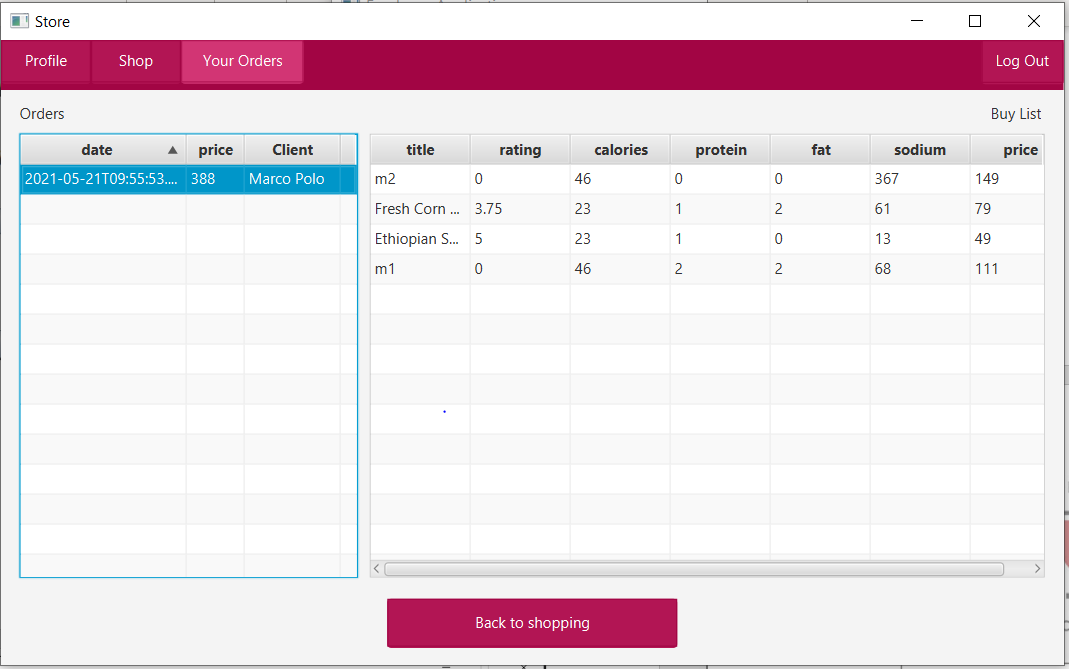
* ProfileClient



* ClientShop

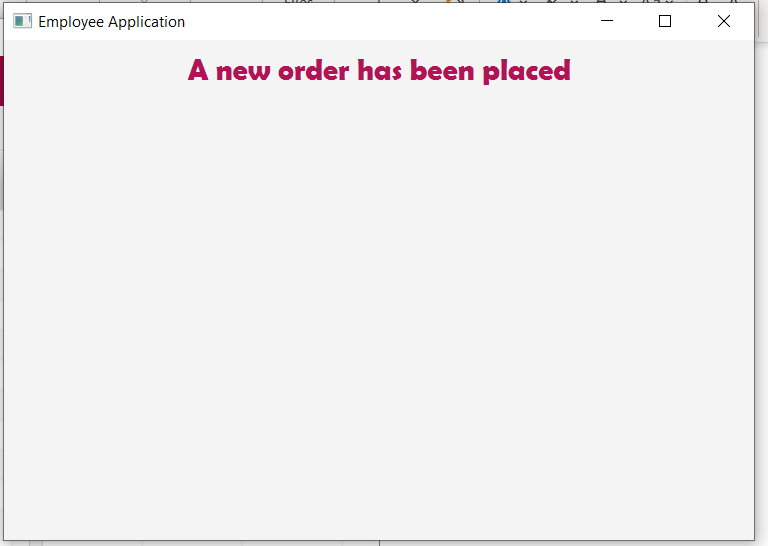


* ClientOrdersHistory

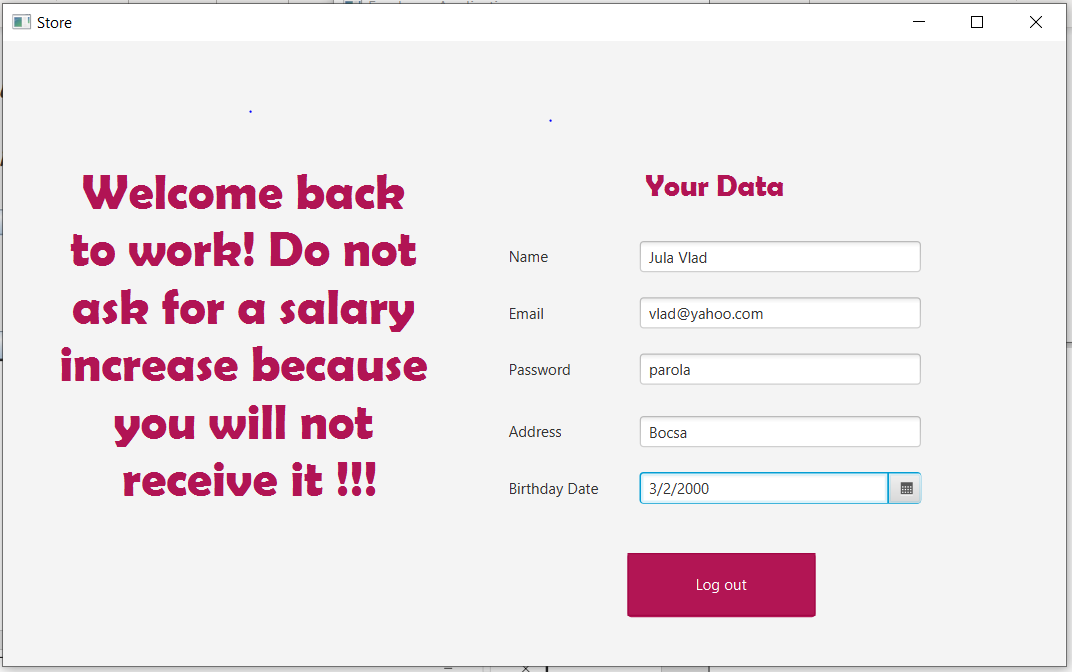


##### Pachetul employee

* EmployeeNotify

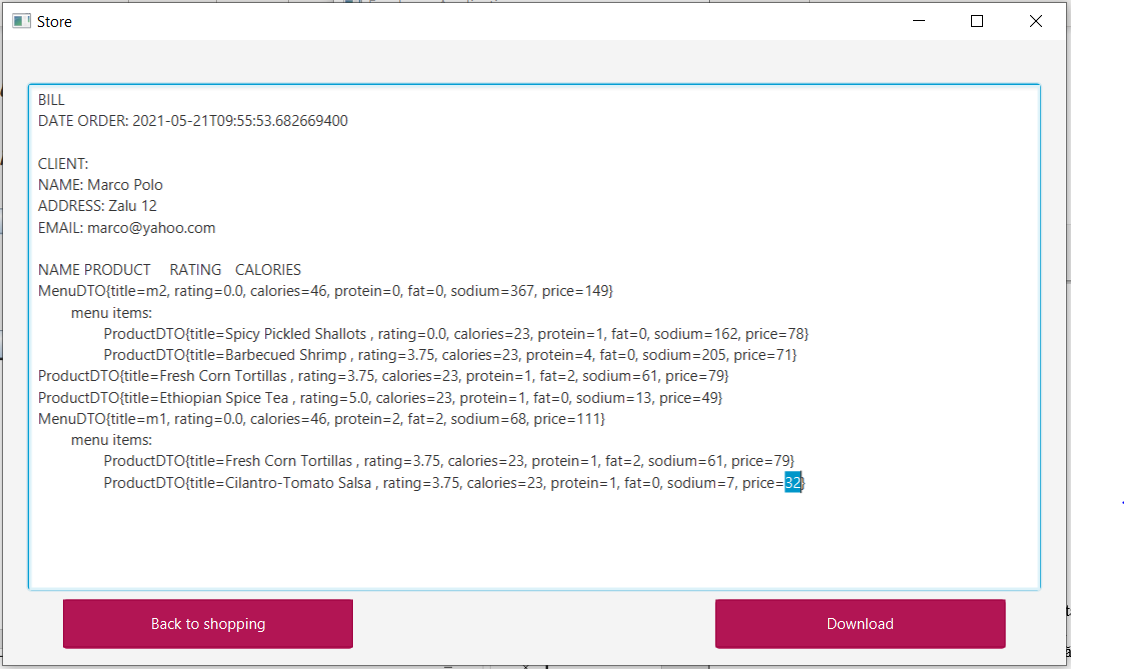


* ProfileEmployee



##### 4.5.2.5 Pachetul bill

* Bill



# Rezultate

Corectitudinea aplicație a fost testată prin utilizarea ei în toate cazurile și prin testarea tuturor funcționalităților descrise mai sus. Pentru a mă asigura că toate rezultatele sunt corecte, am verificat constant ca datele din interfață să corespundă cu datele regăsite în fișierele care stochează toate informațiile.

# Concluzii

Prin intermediul acestei teme mi-am consolidat cunoștințele de programare orientată pe obiect. Mai mult decât atât, am învățat mai multe despre cum să lucrez cu Map și cum să îmi extrag datele dintr-o astfel de corecție. Tot cu ajutorul acestei teme mi-am aprofundat noțiunile legate de serializare și deserializare.

Această aplicație ar putea să fie dezvoltată prin adăugarea unei opțiuni prin care clienții să poată lăsa un feedback vizavi de experiența lor utilizând această aplicație. Mai mult decât atât, aplicația ar putea să fie dezvoltată și prin adăugarea unui mecanism care oferă reduceri sau diverse beneficii clienților care sunt cei mai fideli aplicației și care au comenzi care depășesc o anumită sumă. O altă îmbunătățire ar fii aceea ca toți clienții să fie notificați atunci când apare un nou produs de bază sau un nou produs compus.