**Documentatie Tema 4**

Hitu Octavian

Grupa:30226

Profesor Laborator : Moldovan Dorin Vasile

**Cuprins**

1.Obietivul temei

2. Analiza problemei, modelared, scenario, cazzuri de utilizare

2.1. Analiza Problemei

2.2. Modelare

2.3. Cazuri de utilizare

3. Proiectare

3.1. Diagrama UML

3.2.Clasa Account

3.3.Clasa BaseProduct

3.4.Clasa CompositeProduct

3.5. Clasa Menu

3.6. Clasa Order

3.7.CLasa ReadFromFile

3.8. Clasa Serialization

3.9.Clasa DeliveryService

3.10.Clasa IDeliveryServiceProcessing

3.11.Clasa FirstFrame

3.12. Clasa UserPassFrame

3.13.Clasa ClientFrame

3.14Clasa AdministratorFrame

3.15.Clasa ControllerAdministratorFrame

3.16.Clasa controllerclientframe

3.17.Clasa ControllerFirstFrame

3.18.Clasa ControllerUserPassFrame

1.Obietivul temei

Proiectul pe care l-am implementat reda intr-o forma cat mai buna o aplicatie de tip “delivery management” pentru o companie de catering.Acest proiect are la baza o lista de produse simple pentru care am implementat anumite functonalitati. In aceasta aplicatie te poti loca ca si client sau ca si administrator.In acest proiect ca si client poti vedea lista de produse, poti adauga produse in cos so poti plasa comenzi. Ca si administrator poti face mai multe lucruri precum: editare de produse, adugare de produse, stergere , generare de rapoarte etc. In capitolele de urmeazava voi explica functionalitatile implementate in aceasta aplicatie. Pentru a intelege toate aspectele , noi trebuie sa intelegem mai intai ce inseamna conceptele de serializare si de deserializare, acesta fiind folosit in proiectul nostru.

Serializarea reprezinta transformarea unui obiect intr-o secventa de octeti , din care ulterior sa poate fi refacut intr-un obiect original.Procesul invers serializarii este deserializarea , care citeste un obiect serializat si ii reface starea originala. Totodata cu ajutor acestor doua noi putem sa retinem datele din lista de produse fara a o citi de fiecare data.

2. Analiza problemei, modelared, scenario, cazzuri de utilizare

2.1. Analiza Problemei

Daca este sa ne gandim intr-o situatie reala, problema proiectului nostru se regaseste peste tot in viata noastra , mai ales ca si clienti in momentul cand dorim sa comandam de mancare prin intermediul unei aplicatii de catering . fiecare persoana a folosit o aplicaatie de genul la un moment dat. Ca si client putem adauga produse in cos , putem plasa o comanda si sa cautam in lista de produse ce ne intereseaza. Din punctul de vedere al administratorului acesta poate sa creeze meniuri cum sunt la multe magazine de exemplu meniul zilei , poate sa modifice si sa adauge produse noi, dar poate si sa genereze rapoarte pe baza comenzilor.

2.2. Modelare

Modelarea aplicatiei se bazeaza pe conceptele programarii orientate pe obiecte, acestea fiind:

Clasele- prin clasa intelegem o forma a unui obiect deoarece aceasta reprezinta o descriere generalizata a acestuia. Clasa defineste toate atributele pe care un obiect le are ,iar metodele ce fac parte din acea clasa definesc functionalitatea acelui obiect;

Obiect- un obiect reprezinta o componenta software care incorporeaza atribute si operatii . Operatiile se pot efectua asupra atributelor .

Incapsularea- aceasta are rolul de a uni informatiile importante ale unui obiect. Ea totodata restrictioneza accesul la date si metode din exterior.

Mostenirea- din punct de vedere ierarhic aceasta permite claselor similare sa se suprapuna. Datorita ei clasele inferioare pot folosi metode din clasele superioare.

Polimorfismul- prin definitie aceasta reprezinta abilitate de a lua mai multe forme .Prin polimorfism aceeasi metoda ce este folosita intr-o superclasa poate fi suprascrisa fiind reutilizara in subclase ,ele avand o functionalitate diferita

2.3. Cazuri de utilizare

Aplicatia dispune de o inetrfata care permite utilizatorului sa isi selecteze pozitia de exmplu client sau administrator. Ca si client se deschide o alta interfata care iti permite sa te loghezi cu contul tau, dupa care intr-un final la ambele personae se vor deschide interfete principale de utilizare a aplicatiei.

In urmatoarele randuri vav oi prezenta parametrii de intararedin interfata grafica pentru fiecare tip de utilizator.

Client:

Username- este numele pentru logarea in cont

Password-parola de la contul acestuia

Title, price, rating, sodium, calories, protein fat – acestea sunt atributiile unui produs. Acestea se pot introduce in interfata pentru a-l putea ajuta pe client sa introduca sau sa caute acel produs.

Administrator:

In interfata administratorului nu vom mai avea logare deoarece nu este nevoie acesta fiind doar unul.

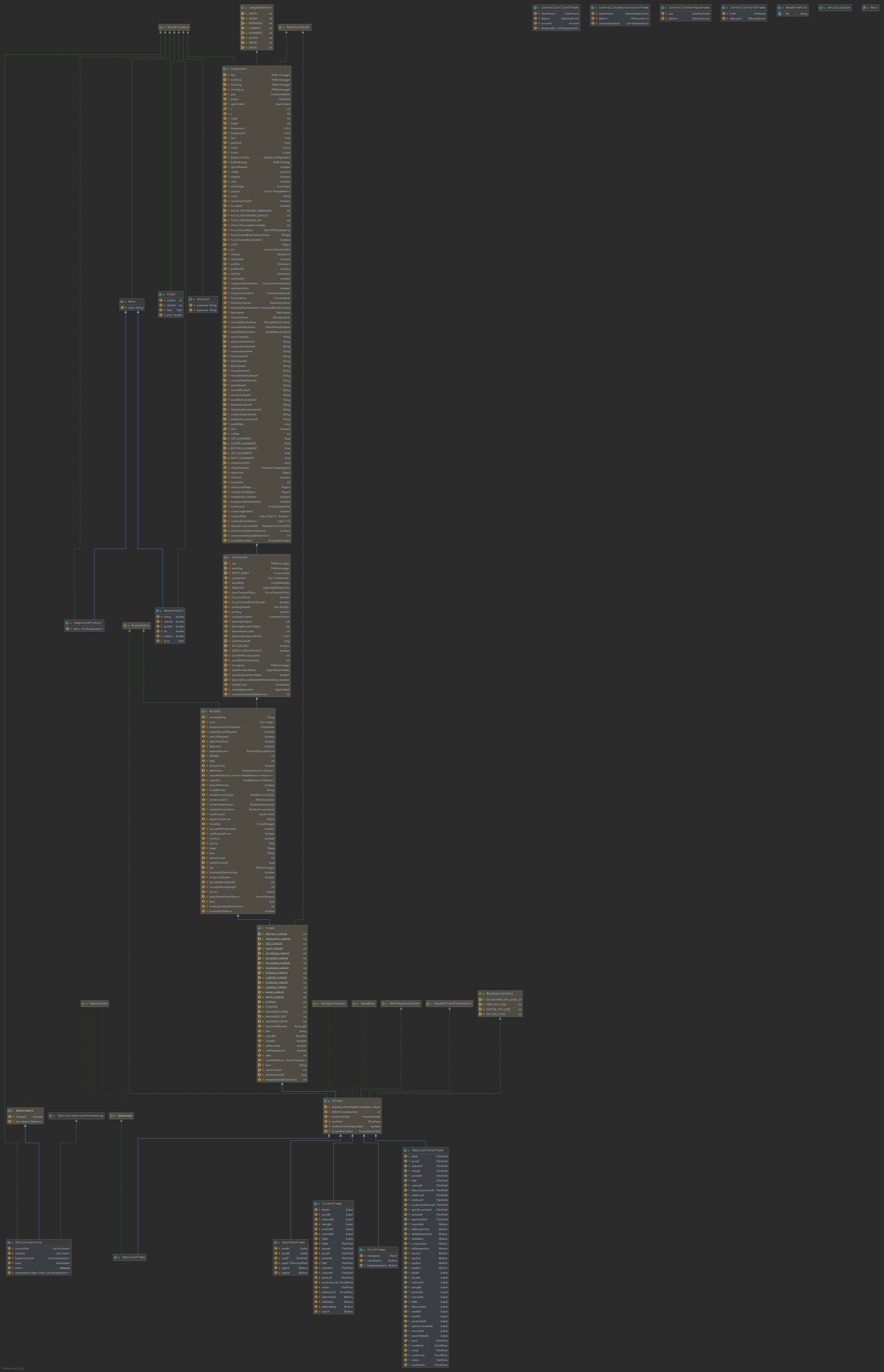
Title, price, rating, sodium, calories, protein fat – acestea sunt atributiile unui produs. Cu ajutorul lor administratorul poate sa adauge sa editeze sau sa stearga un produs din lista cu produse.

Start h, end h, number times, date specific, more products, amount – aceste elemente vor ajuta administratorul sa genereze anumite rapoarte care se bazeaza pe aceste elemente pe care el le va introduce.

3. Proiectare

Aplicatia are la baza 18 clase dintre care una fiind main-ul ,acestea fiind grupate in package-uri: BusinesLayer, controller, datalayer, model si presentation. . Acesta este organizata confrom sablonului Model-View-Controller (MVC). Acest tip de organizare reprezinta separarea interfetei utilizator într-o vedere (View) care interactioneaza cu Modelul după nevoi, si un Controller care raspunde la cererile utilizatorului, interactionand atat cu View-ul, cat si cu Modelul.

3.1. Diagrama UML



Package-ul model

3.2.Clasa Account

Aceasta clasa are ca si variabile username si password. A fost create pentru a putea face posibila logarea in aplicatia noastra. Fiecare client are specific lui un username si o parola cu care se va loga pentru a putea intra in interfata de utilizare a aplicatiei.

3.3.Clasa BaseProduct

Aceasta clasa sta la baza aplicatiei noastre deoarece lista pe care o folosim este formata din baseproduct-uri. Aceasta clasa are urmatoarele: rating, calories, protein, fat, sodium, price. Acestea sunt atributele produselor pe care un utilizator le poate cumpara sau un administrator le poate modifica. In aceasta clasa am generat pe langa contructor metodele de get si set, toString , equals, hashcode.

3.4.Clasa CompositeProduct

In aceasta clasa se da o lista de baseproduct-uri si am implementat ca si metode si am implementat metodele de insumare a grasimi, proteinelor a pretului, a ratingului, a caloriilor si a sodiumului.

3.5. Clasa Menu

In clasa menu am am dat variabila name pe care o vom folosi la composite product-uri . tot aici avem si metodele de get si setname.

3.6. Clasa Order

In clasa Order am dat caracteristicile pentru o comanda si acestea sunt : order id , client id, date si price. Pe langa contructor am implmentat si metodele de get si set , toString , equals si hashcode.

Package dataLayer

3.7.CLasa ReadFromFile

Aceasta clasa are rolul de a citi din fisier toate produsele din fisier prin metoda read file implementata in interior. Am declarata un separator si metoda citeste cu ajutorul separatorului toate produsele si le introduce in baseproduct.

3.8. Clasa Serialization

In aceasta clasa este implementat serializarea si deserializarea. Folosind un fileinput si un object input vom scries au citi acel obiect in functie de metoda pe care o folosim.

Package-ul BusinessLayer

3.9.Clasa DeliveryService

In aceasta clasa sunt implementate functionalitatile propriuzise ale acestei aplicatii. Ca si prima moetoda o avem pe cea de readproduct care are ca argumente un file si un hashset de produse simple. Citeste din file produsele dupa care le adauga in lista de baseproduct-uri declrata la inceputul clasei. La final de aplica serializarea intr-un bloc try-catch. Totodata avem metoda de addaccount care adauga un cont in lista de conturi dupa care serializeaza toata lista. Mai avem si metoda de search account care verifica daca un cont se afla deja in lista de conturi return true in cazul in care contul exista sau false in cazul in caer contul nu exista in lista de conturi. Continuam cu metoda add base product care adauga in lista de produse un nou produs , dupa care serializeaza toata lista de produse. Inca o metoda este aceea de set product care ne ajuta la modificarea unui produs avand ca argument dat un produs p , modificandul pe cel existent cu cel nou. O alta functie pe care o gasim este cea de delete base product care sterge un anumit produs din lista la final ca toate celelalte serializand lista de produse. Functia add simple order ne adauga o comanda noua in lista de comenzi dupa care serializeaza lista. Urmatoare functie add map order list face acelasi lucru ca cea dinaintea ei doar ca aceasta adauga comezi compuse din mai multe elemente. Totodata avem si metoda de add employee obs care adauga observatorul in momentul cand am plasat o comanda si suntem informati ca un angajat ne a preluat-o.

3.10.Clasa IDeliveryServiceProcessing

In clasa IDeliveryServiceProcessing apar operatiile care pot fi executate de administrator si de client , ele fiind implementate mai tarziu in DeliveryService.

Package-ul presentation

3.11.Clasa FirstFrame

Aici avem implementat interfaa pentru primul frame al aplicatiei. Aici utilizatorul are doua butoane pe care le poate apasa. Cel de administrator care-l duce in interfata pentru administrator si cel de client care-l va duce in interfata pentru login sau register.

3.12. Clasa UserPassFrame

In aceasta clasa am implementatinterfata destinate logari clientului. Aici vor fi doua textfiled-uri in care urilizatorul isi va putea scrie username-ul si parola dupa care va avea doua butoane, cel de login daca contul este deja inregistrat sau cel de register daca este utilizator nou al aplicatiei.Daca acesta apasa contul de login daca nu este inregistrat ii va aparea un popout cu inregistrare.

3.13.Clasa ClientFrame

Aici am create interfata destinata clientului prin care poate sa importe toate produsele interfata pentru a le vedea sa adauge un produs in lista de orders , sa plaseze o comanad si sa caute un produs dupa anumite criterii de cautare . Avem TextFiled uri pentru toate atributele de la produse pentru a le introduce si doua liste una pentru toate produsele si una pentru produsele adaugate care vor putea fi comandate.

3.14Clasa AdministratorFrame

In aceasta clasa am implemantat inetrfata pentru administrator. Avem textfield-uri unde putem introduce atributele unui produs, dupa care vom avea urmatoarele butoane: import – listeaza produselein interfata; delete base prod – care va sterge un prous din lista noastra de produse; edit base prod – care ca edita un produs dupa atributele pe care I le vom da; add base prod – care va adauga un produs nou in lista noastra de produse; add list- care va adauga in lista un element; compose – care va compune un meniu folosnd ca nume ce scriem noi in textfield; raport\_1, raport\_2,raport\_3 , raport\_4 care vor face rapoartele in functie de termenii pe care in dam in textfield urile alaturate precum : start h, end h , number times, date specified, more products si amount.tot aici putem vedea lista cu orders-urile pe care le-au facut clientii.

Package-ul controller

3.15.Clasa ControllerAdministratorFrame

Aici am implementat action listenerii pentru butoanele din View. Avem butonul import care va ajuta la afisarea in interfata a elementelor din lista de produse . Butonul de edi base product va edita produsele dupa argumentele datele date de noi; add product va adauga un element creat de noi; delete va sterge un elementdat de noi ; add list va adauga in lista un element pe care l-am ales pentru a putea crea un meniu; compose va compune un meniu dupa elementele date de noi; rap1, rap2, rap3, rap4 vor crea rapoartele pentru elementele pe care le- am inserat de la tastatura.

3.16.Clasa controllerclientframe

Aici am implementat action listenerii pentru butoanele din View. Avem butonul import care va ajuta la afisarea in interfata a elementelor din lista de produse. Avem butonul de add list care dupa un nume ne va pune in lista acel produs. Totodata mai avem butonul de add order care ne va plasa comanda dupa produsele pe care le avem deja in lista. Butonul de search ne va cauta un anumit element in lista dupa proprietatile date de noi. Avem si un scroll panel care ne ajuta sa vedem lista de produse.

3.17.Clasa ControllerFirstFrame

Aici sunt implementati action listenerii pentru cele doua butoane de administrator si de client; daca apesi pe cel de administrator se va deschide frame-ul pentru administrator, iar daca apesi pe cel de client se va deschide interfata d logare.

3.18.Clasa ControllerUserPassFrame

Aici sunt implementate butoanele de login si de register. Daca contul este deja create vom apasa login si intram in interfata client, dar daca nu este inregitrat trebuie apasat per register.

4.Rezultate

Rezultatele acestui proiect pot fi vazute prin pornirea aplicatiei si folosirea ei propriuzisa. Am reusit sa creez o interfata sugestibila si usor de utilizat pentru orice utilizator. Aplicatia reuseste sa surprinda cu usurinta o aplicatie pentru catering. De la prima interactiune trebuie sa selectezi daca esti admin sau client dupa care daca esti client trebuie sa te loghezi. Dupa logare trebuie sa accesezi interfata de utilizare sa sa faci comenzi. Ca si admin dupa apasarea butonului vei putea face modificari in lista sau genera rapoarte.

5.Concluzii

Prin acest proiect am putu sa vad concret si sa inteleg avantajul reflexiei inca de la primele faze ale implementarii aplicatiei. Pot spune ca am invatat importanta organizarii timpului acordat unui proiect, structurarea sa si analizarea cu atentie a problemei inainte de inceperea implementarii propriuzisa a proiectului.

Din punctul meu de vedere , posibilele dezvoltari care pot fi aduse aplicatiei sunt : scrierea unei chitante, imbunatatirea interfetei din punct de vedere estetic pentru a fi mai primitoarea si mai usor de inteles de catre utilizator.

In concluzie , cred ca am am respectat conditiile prestabilite pentru aceasta aplicatie si ca acest proiect reuseste cu succes sa fie o aplicatie pentru o firma de catering.