

Proiect Învățare automată Alergare

Mihai Crisan

May 2024

Contents

1	Introducere	2
1.1	Domeniul de aplicare	2
2	Descriere Generală	2
2.1	Perspectiva de dezvoltare	2
2.2	Customizabilitate	2
2.3	Caracteristicile Produsului	2
2.3.1	Must	2
2.3.2	Should	3
2.3.3	Could	3
2.3.4	Won't	3
2.4	Utilizatorii aplicației	3
2.4.1	Client	3
2.4.2	Ospătar	3
2.4.3	Manager	4
2.5	Mediul de operare	4
2.6	Constrângeri de proiectare și implementare	4
3	Diagrame	5
4	Alte Cerințe non-funcționale	7
4.1	Performanță	7
4.2	Atribute de calitate a software-ului	7
5	Probleme și testare	7
6	Tehnologii	8
7	Metodologia de dezvoltare	8
8	Cerințe finale obligatorii proiectului	8
9	Concluzii și dezvoltare viitoare	9

1 Introducere

Aceasta aplicație de management al unui bar este o platformă web dedicată staff-ului cât și clienților, proiectată pentru a oferi o legătură între angajați și client, netezind procesul de preluare a comenzilor.

Proiectul are ca scop dezvoltarea unei aplicații web, destinată afacerilor de tip bar/restaurant pentru ca un client să poată comanda fără nevoie directă de o interacțiune cu ospătarul până la momentul servirii.

1.1 Domeniul de aplicare

Aplicația este proiectată pentru a fi folosită în incinta unui bar, oferind capacitatea punerii unei comenzi, actualizarea comenzii și îndeplinirea comenzii de fiecare utilizator într-un mod fluent.

2 Descriere Generală

2.1 Perspectiva de dezvoltare

Aplicația este dezvoltată în modul AGILE. Feature-urile aplicației vor fi implementate într-un mod inițial iar după stabilirea valabilității lor ele vor fi implementate în varianta finală

2.2 Customizabilitate

Aplicația nu este gândită pentru a fi un produs întreg, ci ca un produs funcțional ce poate fi adaptat pentru nevoile fiecărui client. Partea de user interface se va implementa fără costuri adiționale în colaborare cu clientul iar la nevoie se pot implementa funcționalități noi contra cost.

2.3 Caracteristicile Produsului

Aplicația a fost gândită cu o listă MoSCoW (Must, Should, Could, Won't) înainte de a se începe dezvoltarea

2.3.1 Must

- Bază de date: Aplicația trebuie să permită stocarea produselor, comenzilor, credențialelor într-un mod securizat.
- Sistem Înregistrare: Trebuie implementat un sistem de înregistrare a clienților format din username și parolă, parola fiind criptată.
- Sistem Autentificare: Trebuie implementat un sistem pentru a permite clientului, managerului și ospătarilor de a se loga și a accesa doar paginile destinate lor.

- Comandă: Trebuie implementată o modalitate de a se pune o comandă și a de se actualiza în sistem în timp real pentru a putea fi pregătită.
- Atribuire Status: Comenzile trebuie să poată fi categorizate ca în pregătire sau finalizate pentru a fi eficientizat procesul de livrare la masă.
- Introducere Stoc: Managerul trebuie să aibă opțiunea de a adăuga inventar și noi lucruri în baza de date.
- Modificare Stoc: Stocul trebuie să fie modificabil, însă nu neapărat să permită ștergerea inventarului.

2.3.2 Should

- Alegere Masa pe baza mesei : Clientul transmite în câmpul de detalii masa la care este, plus cerințele speciale dacă ele există
- Extragere Sold : Managerul are opțiunea de a selecta toate vânzările din o perioadă determinată și de a vedea suma lor

2.3.3 Could

-
- Alegere pe baza de cod QR: Clientul poate selecta masa automat pe baza unui cod QR (MENȚIUNE: în varianta prezentată la proiect acest feature nu a fost implementat încă)

2.3.4 Won't

- Hipercomplexitate: Aplicația nu este una ce trebuie să conțină toate featureurile de care un client are nevoie în producție precum o interfață completă. Aceste lucruri vor fi implementate la nevoia clientului pentru a oferi un look și funcționalități aparte în funcție de cerințele clientului

2.4 Utilizatorii aplicației

2.4.1 Client

Clientul este cel mai important utilizator al acestei aplicații. Clientul este cel ce începe fluxul unei comenzi, de aceea focusul aplicației este în a oferi o experiență bună acestor utilizatori. Atribuțiile clientului constă în crearea unui cont și plasarea comenzilor.

2.4.2 Ospătar

Utilizatorii cu rol de ospătar au responsabilitatea de a verifica constant fluxul de comenzi și de a atribui starea comenzilor. Acest rol oferă și posibilitatea de a executa fiecare funcție a rolului de utilizator.

2.4.3 Manager

Utilizatorii cu rolul de manager au responsabilitatea de a updata stock-ul constant pentru a se asigura că se pot finaliza operațiunile fără probleme.

Acest utilizator mai are posibilitatea de a înregistra produse și de a extrage vânzările pe o perioadă determinată. Acest utilizator poate de asemenea să execute toate operațiunile ce pot fi îndeplinite de ceilalți utilizatori.

2.5 Mediul de operare

Dacă vorbim de platformă hardware, această aplicație nu necesită specificații complexe pentru rulare. Aceasta este destinată sistemului de operare Windows, construită în Java 21.0.2, necesitând o conexiune la o bază de date MySQL și o conexiune la rețea.

2.6 Constrângeri de proiectare și implementare

Aplicația trebuie menținută constant pentru a se asigura stabilitatea ei. Baza de date nu trebuie modificată de persoane neautorizate deoarece există posibilitatea unui conflict ce poate duce la oprirea prematură a aplicației.

3 Diagramme

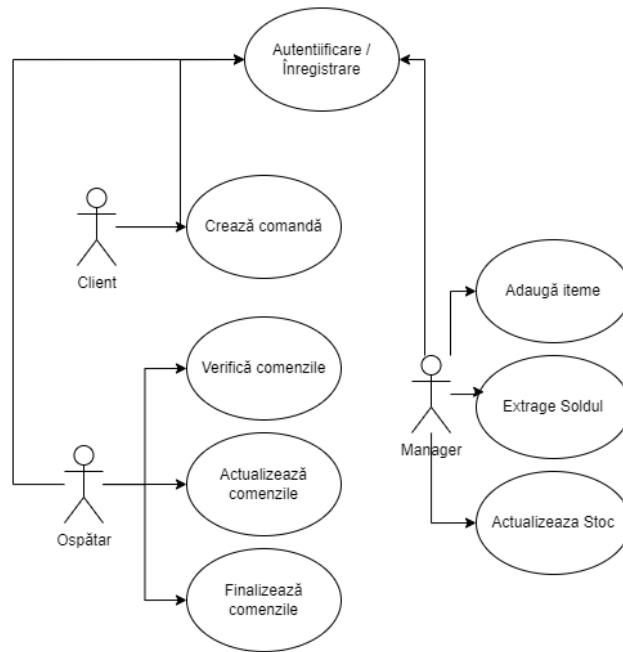


Figure 1: Diagramă use case

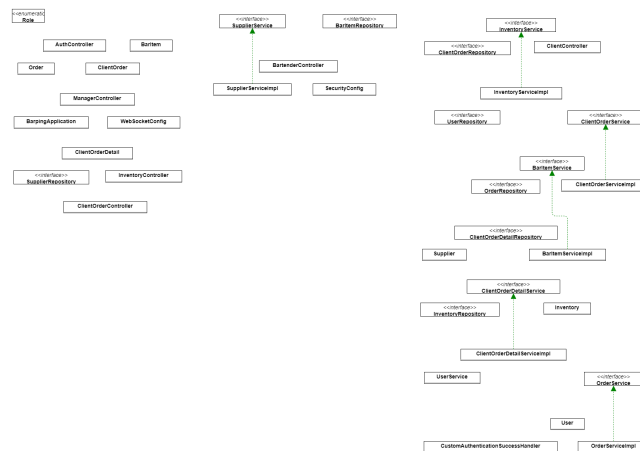


Figure 2: Diagrama aplicației în Java

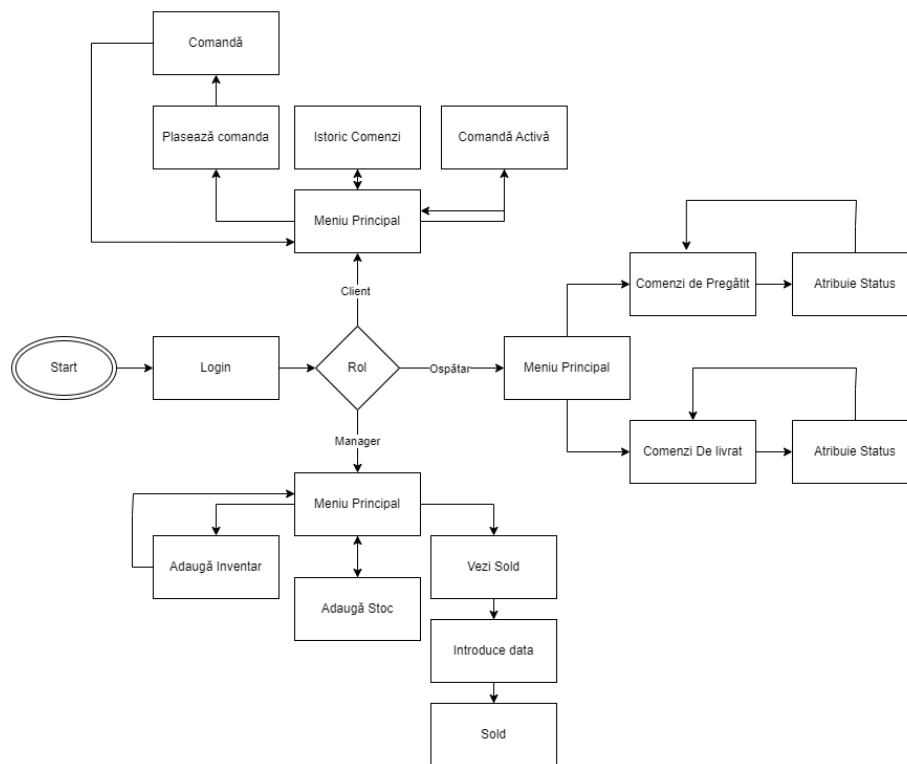


Figure 3: Diagramă flow

4 Alte Cerințe non-funcționale

4.1 Perfomanță

Pentru o bună folosire a aplicației este necesar ca aceasta să fie foarte responsivă și să funcționeze pe orice device.

Acest lucru este obținut prin folosirea unui browser de către utilizatori, presupunând că orice device modern poate rula un browser și poate acces pagini web fără probleme.

De asemenea, este important ca toate operațiunile să se întâmple în timp real pentru a asigura că toate comenzile sunt livrate la timpul lor. Beneficiază de acest lucru și baza de date, permitând actualizarea stocului pe loc pentru a rezolva eventualele lipsuri sau probleme întâmpinate pe parcurs

4.2 Atribute de calitate a software-ului

Aplicația destinată pentru preluarea comenzilor dintr-un bar este proiectată pentru a asigura o platformă fiabilă pentru administrarea și gestionarea eficientă a operațiunilor în cadrul unui bar.

Fiabilitatea este un aspect cheie al sistemului, acesta garantând funcționarea corectă și continuă a tuturor funcționalităților. Aplicația trebuie să fie disponibilă constant pentru a asigura o experiență fluidă pentru utilizatori, eliminând loteria erorilor.

Interoperabilitatea este o altă caracteristică cheie, permițând sistemului să se integreze eficient cu alte platforme sau aplicații relevante. Capacitatea de a lucra în mod eficient cu anumite servicii și sisteme este esențială pentru a ne asigura de o experiență completă și fără probleme.

Usurinta de intretinere si actualizare reprezinta ce-a de-a 3-a caracteristică cheie, care permite clienților să nu piardă timpul cu diferite probleme sau erori software, astfel sistemul fiind mai eficient și mai econom pe termen lung.

5 Probleme și testare

În lipsa timpului și a cunoștințelor de scriere a testelor pentru a se testa automat aplicația, testarea s-a realizat manual, fiecare feature fiind testat în diferite ipostaze, chiar și în edge-cases.

O problemă inițială a fost versionarea corectă a diferitelor librării pentru a se asigura că ele sunt compatibile una cu alta. Un exemplu foarte bun în această privință reprezintă librăria Spring Security, ce oferă documentația contradictorie în funcție de resursele abordate.

O altă problemă mare a fost standardizarea comenzilor pentru ca ele să poată fi introduse și preluate din baza de date. Au fost folosite diferite variante până ce s-a ajuns la o soluție adecvată acestei probleme

6 Tehnologii

Aplicația a fost dezvoltată folosind Spring Boot, lista completă de tehnologii fiind:

- JAVA 21.0.2 - limbajul principal de dezvoltare a aplicației
- Springboot - Framework-ul principal folosit în dezvoltare, standard în dezvoltarea aplicațiilor web folosind java
- Thymleaf - O tehnologie de frontend folosită pentru a servi pagini HTML și pentru stilizarea paginilor
- MYSQL - principala bază de date a aplicației având o implementare ușoară și fiind foarte rezilientă.

7 Metodologia de dezvoltare

Aplicația a fost dezvoltată de o singură persoană ce a dus ca dezvoltatorul să îndeplinească toate rolurile precum Manager, Developer și Tester.

8 Cerințe finale obligatorii proiectului

Acest proiect a venit cu diferite cerințe din punct de vedere tehnologic. Mai jos se regăsesc cerințele și cum au fost implementate.

- Integrare SpringBoot- Springboot a fost folosit drept framework-ul de dezvoltare principal
- FrontEnd la alegere - A fost folosit thymeleaf deoarece pentru o integrare ușoară și directă a frontend-ului
- Integrare bază de date - A fost folosit MYSQL pentru o integrare directă cu aplicația
- Integrare JPA - A fost folosit pentru o comunicare directă între aplicație și baza de date și a mapat obiectele relațional
- Dependency Injection - a fost folosit pentru a clarifica comunicarea între clase și a menține totul ușor de înțeles
- Websockets - Au fost folosite pentru actualizare comenzilor odată ce un ospătar a schimbat valoarea lor
- WebSecurity - Websecurity-ul a fost implementat în metoda standard oferită de Springboot pentru a asigura siguranța și criptarea elementelor sensibile

9 Concluzii și dezvoltare viitoare

Această aplicație este un bun punct de plecare pentru implementarea unui sistem digital de comenzi într-un bar.

Oferă un schelet bun, fiind ușor de modificat și cusomizat pentru cerințele fiecărui client

În viitor se poate realiza feature-ul neîmplinit în această versiune (scanarea pe bază de coduri QR) dar și alte funcționalități generale de care ar beneficia orice client.

Se pot îmbunătăți toate aspectele ale aplicației, însă un lucru cu prioritate mai ridicată ar fi sistemul de ales al lucrurilor ce utilizatorul de tip client, cel actual nefiind complet intuitiv. Acest lucru se poate realiza prin o testare în masă și colectând user feedback.