

**UNIVERSITATEA „ALEXANDRU IOAN CUZA”
FACULTATEA DE INFORMATICĂ**



LUCRARE DE LICENȚĂ

QuickMenu

Sesiunea : Februarie 2020

Coordonator științific

Prof. Colab. Olariu Florin

Absolvent

Pester Leonard

**Iași
2020**

**UNIVERSITATEA „ALEXANDRU IOAN CUZA”
FACULTATEA DE INFORMATICĂ**

QuickMenu

Sesiunea: Februarie, 2020

Coordonator științific

Prof. Colab. Olariu Florin

Absolvent

Pester Leonard

**Iași
2020**

Avizat,

Îndrumător Lucrare

Titlul, Numele și prenumele _____

Data _____ Semnătura _____

DECLARAȚIE

privind originalitatea conținutului lucrării de licență/disertație

Subsemnatul(a)

domiciliul în

născut(ă) la data de, identificat prin CNP,

absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de

specializarea, promoția

declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul:

_____elab

orată sub îndrumarea _____, pe care urmează să
o susțin în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență/disertație să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data azi,

Semnătură student

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca lucrarea de licență cu titlul “**QuickMenu**”, codul sursă al programelor și celelalte conținuturi care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică. De asemenea, sunt de acord ca Facultatea de Informatică din cadrul Universității “Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, 12.02.2020

Absolvent, Pester Leonard

Cuprins

Introducere	5
Motivație	5
Obiective	5
Contribuții	6
Scurta descriere a soluției	7
Abordări tehnice	7
Instrumente software	7
Limbaje si medii de programare	8
1 Aplicații existente	9
1.1 Aplicația Fast Order	9
2 Descrierea problemei	10
3 Descrierea soluției	11
3.1 Funcționalități	11
3.1.1 Partea front-office	11
3.1.2 Partea back-office	15
4 Detalii de implementare	21
5 Diagrame	25
5.1 Arhitectura aplicației	25
5.2 Diagrame use-case	26
5.3 Relationarea bazei de date	28
Concluzii	29
Direcții viitoare	29
Bibliografie	30

Introducere

Motivație

În contextul industriei alimentare publice, observată dintr-un punct de vedere economico-social, reușește să-și păstreze rolul și importanța economică în viața oricărui individ al societății.

Serviciile turismului în ceea ce constă alimentația trebuie să fie prezente într-o manieră plăcută și utilă clienților, acesta fiind și un aspect important pentru orice spațiu comercial, conducerea acestuia oferind servicii de calitate conform nevoilor consumatorilor.

Deoarece această industrie este o prioritate a activităților turistice, se provoacă o presiune asupra administrării oricărui restaurant unde scopul principal este să obțină profit dar și o imagine cât mai bună. Majoritatea unităților de administrație publice se folosesc doar de personal pentru pentru administrarea activității spațiului comercial, uneori putând apărea dificultăți, probleme sau chiar neînțelegeri însoțite de învinuiri având un impact negativ asupra activității din cadrul unui restaurant din punct de vedere al imaginii acestuia, al timpului pierdut și al demersului comenzilor în timpul programului. Toate aceste aspecte negative ar putea fi ocolite prin implementarea unui sistem software pentru a putea face managementul unui restaurant mai accesibil fiecărei persoane din personal, acest sistem reușind să gestioneze comenzile plasate de clienți intensificându-se satisfacția prin care își aleg un meniu, dar în același timp facilitând și activitatea angajaților crescând și gradul de profitabilitate, ceea ce fiecare unitate de alimentație publică are ca prim plan.

Obiective

Ceea ce-mi propun să realizez este ca managementul unui restaurant să aibă un flow cât mai simplu și mai eficient astfel voi implementa o interfață front-office pentru clienții restaurantului și o interfață back-office pentru personalul acestuia.

Un obiectiv principal este să ne asigurăm că cei care ajung în incinta localului și ajung la un prim contact cu aplicația să nu întâmpine dificultăți în a folosi un dispozitiv. Ca prim pas clientul trebuie să selecteze modul în care să plaseze o comandă, fie selectând o servire clasică însoțită de un ospătar/chelner ce poate fi chemat la masă prin selectarea butonului 'Chelner', fie selectând butonul 'Aplicație', de unde își va putea adăuga în coș produsele din meniurile afișate, că într-un

final să finalizeze comandă fără a mai fi nevoie să apeleze la un ospătar/chelner, toate operațiunile desfășurându-se în front-office.

Odată ce comanda a fost finalizată, își întră-n rol panoul back-office ce poate fi accesat de personalul restaurantului. Aici ei pot vizualiza comenzile solicitate de clienți, în urmă pregătirii lor fiind adăugate într-o lista cu comenzi finalizate. Totodată în back-office pot fi accesate produsele ce se află în bucătărie dar și meniurile ce vor fi alese de clienți, acest lucru putând facilita managementul bucătărie putând fi observat foarte ușor stocul de produse obiectivul fiind să nu existe epuizare de stoc.

Această aplicație are ca scop final configurarea unei comenzi, iar finalizarea acesteia să se realizeze într-un timp cât mai scurt atât ca clientul să beneficieze de un timp scurt de așteptare dar și personalul să urmărească comandă/comenzile mult mai ușor.

Contribuții

Datorită faptului ca în grupul meu de prieteni exista o persoană a cărei familii deține un restaurant, au intervenit și discuții mai puțin fericite în grup referitoare la problemele pe care le întâmpină. Odată ce timpul alegerii temei pentru licență se apropia, problemele acelei persoane m-au inspirat să fac posibilă o soluție în a-i rezolva dificultățile menționate, acestea fiind de organizare a ospătarilor.

Odată cu alegerea temei am discutat cu coordonatorul de licență, iar după ce am primit acordul, împreună, am încercat să aducem un plus la ceea ce deja aveam în plan, mai exact un management complex cu o interacțiune între personal și consumatorii din restaurant, nu doar un management al personalului. După discuție am revenit la persoana menționată mai sus de unde am strâns mai multe informații de care aveam nevoie pentru a începe să creez aplicația.

Aici a fost momentul în care am ales tehnologiile și limbajele de programare pe care urma să le folosesc pentru a crea o aplicație de tip web pe care am reușit să o dezvolt în decursul mai multor luni.

În cadrul alcătuirii acestei aplicații am folosit cunoștințe obținute în timpul anilor petrecuți în facultate la materii cum ar fi:

- **Tehnologii web:** protocolul HTTP, limbajul PHP
- **Ingineria programării:** îmbunătățirea modului de lucru și realizarea diagramelor aferente aplicației
- **Programare Orientată pe Obiecte:** principiile programării orientate pe obiecte
- **Baze de date:** gestionarea bazei de date

Scurta descriere a soluției

Aplicația propusă este accesibilă prin intermediul unei conexiuni la Internet. Fiecare masă din incinta unității va fi însoțită de o tabletă cu aplicația instalată dar și de un număr, reprezentând numărul mesei unde comandă va fi servită. Această va avea următoarea componență:

- **Front-office:** interfață prielnică consumatorilor unde vor putea interacționa ori cu un ospatar ori direct cu aplicația configurându-și un meniu.
- **Back-office:** interfață pentru personal în scopul managementului restaurantului
- **Baza de date:** sursa stocării datelor

Abordări tehnice

Instrumente software

Visual Studio Code este un editor de cod sursă care poate fi utilizat cu o varietate de limbaje de programare precum PHP, C++, Java etc. Am ales acest editor datorită experienței pe care am acumulat-o realizând proiecte în cadrul facultății dar și din cauza simplității cu care se poate lucra cu el, putând fi accesibil pentru Windows, MacOS dar și Linux.

Xampp reprezintă una dintre cele mai populare unelte din mediul de dezvoltare PHP, fiind creat pentru a fi utilizat într-un mod foarte ușor și rapid.

Denumirea XAMPP provine de la următoarele:

- X - o denumire generică disponibilă pe mai multe platforme precum Windows, MacOS, Linux etc.
- A - Apache HTTP Server

- M - MySQL
- P - PHP
- P - Perl

Navicat este un program ce reușește să gestioneze ușor și plăcut o bază de date fiind soluția ideală pentru administrarea și dezvoltarea MySQL/MariaDB având un plus. fiind o aplicație ce permite conectarea simultană la o baza de date MySQL și una MariaDB. Acesta a fost ales datorită rapidității inserării sau modificării datelor care m-au ajutat la testarea aplicației.

Limbaje si medii de programare

În scopul realizării proiectului am ales să folosesc tehnologii noi ce-mi pot oferi oportunitatea de a construi o interfață modernă ușor de folosit adaptată anului 2020. Pentru această am folosit în partea de backend următoarele.

- **Codeigniter** ce este un framework puternic ce are la baza arhitectura de lucru model-view-controller (MVC). L-am ales datorită simplității prin care poți dezvoltă o aplicație web dar și prin faptul că m-a făcut mai productiv. Un avantaj al acestui framework php este că m-a ferit de erori precum SQLi și de mici detalii ce m-au ajutat să salvez timp în scopul că nu am stat să rescriu același cod.
- **PHP** fiind axat pe scripting este un limbaj de programare folosit pe scară largă în dezvoltarea aplicațiilor și paginilor web. Am ales acest limbaj deoarece se mulează perfect cu framework-ul menționat mai sus dar și din cauza plăcerii de a lucra cu el, plăcere dobândită în cadrul proiectelor din facultate.

În realizarea interfeței, pe partea de frontend am folosit:

- **Bootstrap** ce este framework-ul ce m-a ajutat să realizez un aspect plăcut aplicației. Este un framework puternic și intuitiv ce te ajută să creezi un site foarte repede și ușor cu ajutorul limbajelor de editare și stilizare precum HTML, CSS și JavaScript.

1 Aplicații existente

1.1 Aplicația Fast Order

Fast Order este o aplicație ce rulează pe dispozitivele mobile având ca sisteme de operare Android și iOS, fiind creată și dezvoltată de doi români ce au urmărit să eficientizeze desfășurarea activității ospătarilor.

Consumatorilor, pe baza locației, le sunt prezentate cele mai apropiate restaurante din zona în care se află, acest lucru putând fi posibil doar cu o conexiune la Internet și aprobarea locației din aplicație. Urmează momentul în care le este afișat meniul unității selectate de unde își pot adauga produse în coș.

Aplicația necesită autentificare, dar este realizată în așa fel încât clienții restaurantului pot urmări meniul, însă în cazul în care doresc să plaseze o comandă sunt obligați să se autentifice

.Dacă vorbim despre prețul acestei aplicații, restaurantele ce doresc să utilizeze aplicația au nevoie de un abonament în valoare de 50-150 € lunar, acest preț stabilindu-se cu dimensiunea restaurantului.

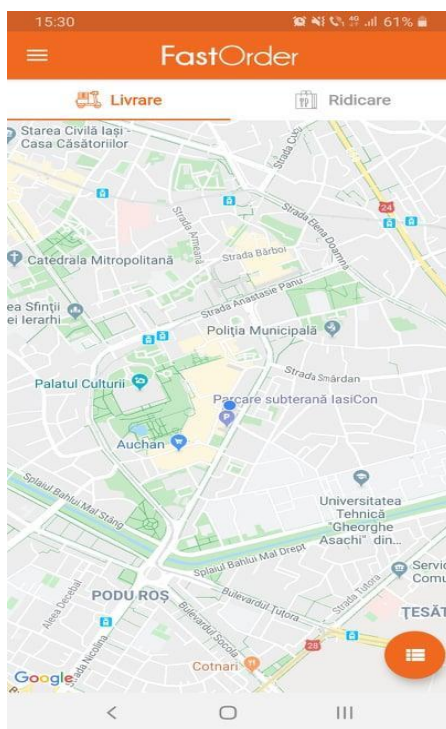


Figura 1 : Aplicatia Fast Order

Dezvoltatorii aplicației, Lucian Cramba și Alexandru Coman, de origine română, au dezvoltat și o aplicație separată pentru personalul unui restaurant. Aceasta conține comenzile ce au fost trimise către aplicație, comenzile confirmate și comenzile gata de servire. Ea mai conține și posibilitatea de a urmări istoricul comenzilor și câteva statistici conform comenzilor.

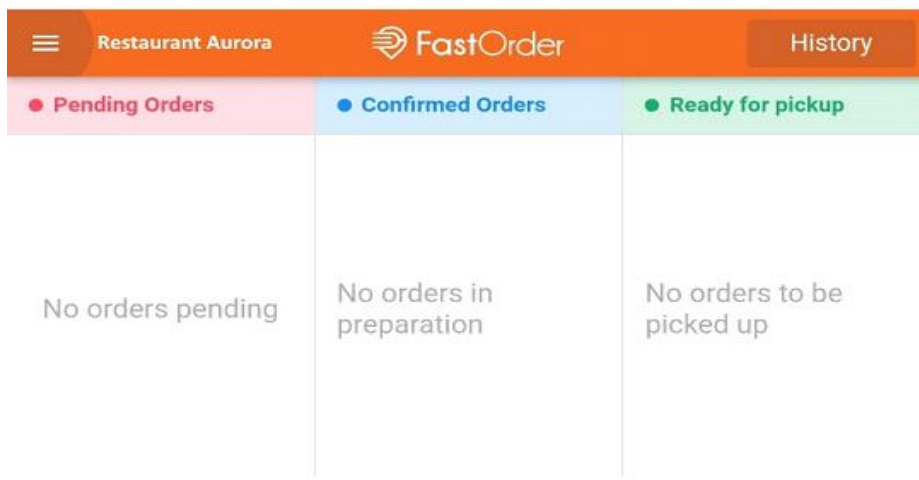


Figura 2 : Fast Order Back-Office

Sursă: <https://www.expressoft.eu/vanzare-cloud/restaurant>

Elementul de noutate este constituit de faptul că soluția prezentată reușește să asigure clienților plasarea unei comenzi direct din aplicația QuickMenu, cu gestionarea stocului direct în mediul intern al incintei prin intermediul modulului back-office. Așadar, produsul realizat nu este destinat unui singur segment de utilizatori, ci acoperă concomitent interacțiunile dintre clienții și personalul unui restaurant.

2 Descrierea problemei

Economia și piața restaurantelor a crescut în ultima perioada având un impact mare în creșterea cerințelor consumatorilor, principalele criterii fiind ambianța spațiului, servirea, ce duc la o consecință în care românii vor opta pentru o unitate unde servirea este făcută cu profesionalism, iar timpul de așteptare să fie cât mai mic.

În general o activitate ce necesită forță de muncă este procesarea comenzilor deoarece necesită multă atenție atât la primirea acesteia dar și la distribuirea ei, fiind necesară o foarte bună coordonare a personalului. Am menționat acest lucru din simplu fapt că în procesul de preluare și servire pot apărea neînțelegeri între angajați, aglomerație printre mese și de multe ori

neconcordanțe între ceea ce clientul a comandat și ceea ce i-a fost servit provocând o reflecție negativă față de ceilalți clienți prezenți.

Prin simplul fapt că trăim într-o generație unde tehnologia este unealtă ce ne scapă de toate probleme, consider că problemele prezentate mai sus vor dispărea, ușurând munca ospătarilor cât și a bucătarilor, dar nu numai munca, timpul fiind la rândul lui un aspect foarte important.

3 Descrierea soluției

3.1 Funcționalități

3.1.1 Partea front-office

Primul contact al unui client cu aplicația este atunci când ajunge la masă unde este însoțit de o tabletă și un număr aferent mesei la care s-a așezat. Prima pagină este alcătuită din câteva informații referitoare la cum se folosește aplicația dar și care sunt opțiunile pe care le oferă această.

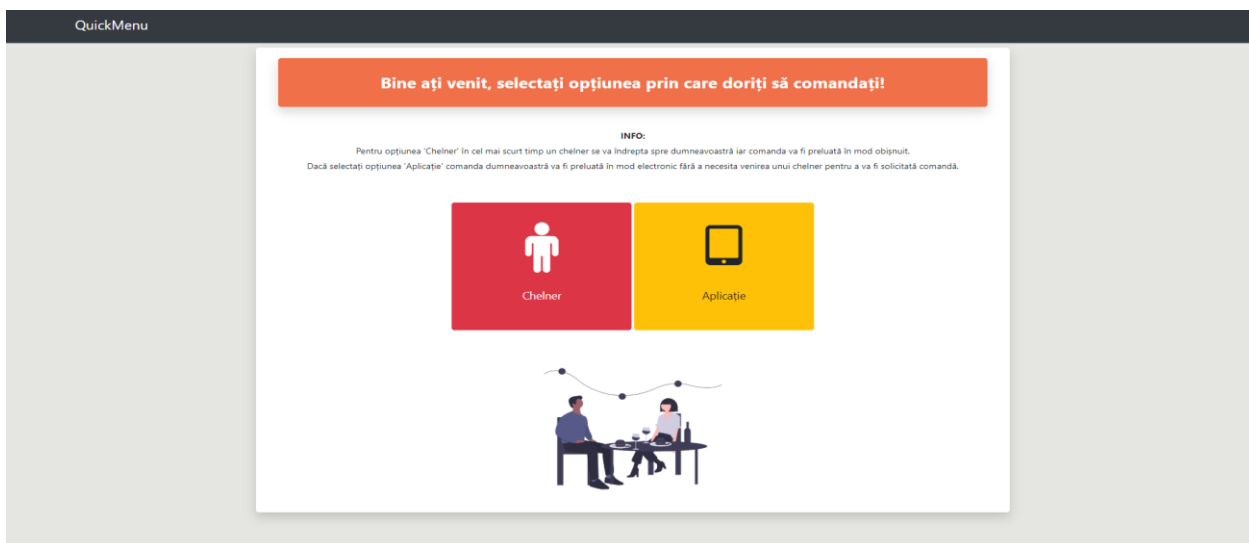


Figura 3 : Pagina de interacțiune cu aplicația

Clientul poate opta între două opțiuni, acestea fiind prezentate astfel:

- **Chelner**
- **Aplicație**

Odată selectata opțiunea ‘Chelner’, clientul este informat ca un ospatar/chelner a fost atenționat ca ați ajuns la masa și doriți să vi se ia comanda. Aceasta optiune a fost creata pentru persoanele ce nu doresc sa foloseasca aplicatia, iar comanda să aibă loc ca într-un restaurant obișnuit oferindu-le un meniu propriu zis, iar in cazul în care după un scurt timp doriți sa adaugati la comanda și alte produse, de asemenea butonul poate fi apăsat pentru a chema din nou o persoana din personal.

Un aspect important de menționat odată cu crearea acestei opțiuni este că trăim într-o lume unde nu toată persoanele se descurcă să folosească un device sau un obiect ce reprezintă tehnologie, de aceea scopul este ca publicul țintă să fie format din orice categorie de oameni.

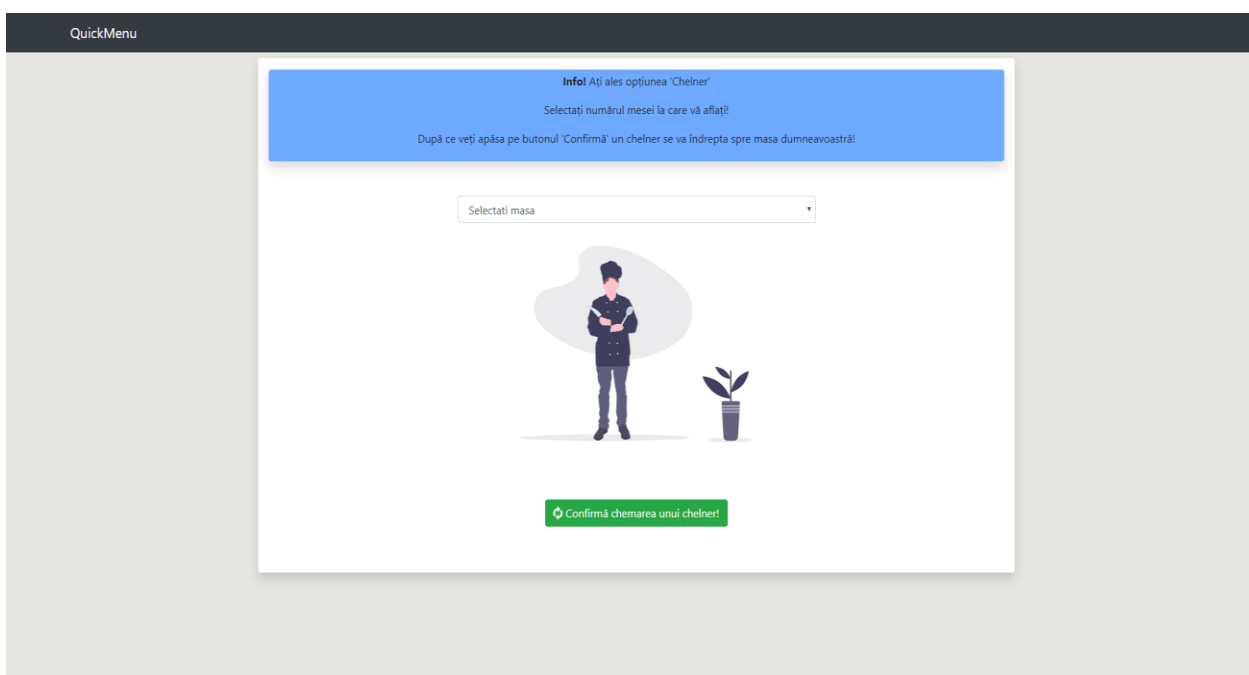


Figura 4: Pagina opțiunii chelner

După ce masa a fost selectată, odată cu apăsarea butonului „Confirmă chemarea unui chelner”, pagină principală va reveni pe ecran pentru a putea selecta dacă dorim o altă operațiune, iar pe pagina comenzilor va fi afișată solicitarea alături de numărul mesei selectate.

După ce ospătarul a luat comanda manual, pentru a o putea trimite spre bucătar va folosi aplicația unde va configura meniul cu cerințele clientului

În cazul în care dorim să folosim aplicația pentru a configura un meniu, vom selecta opțiunea ‘Aplicație’ ce ne va direcționa într-o pagină unde ne vor fi afișate toate produsele pe categorii însoțite de informații aferente produselor cum ar fi nume, preț, descriere dar și o imagine.

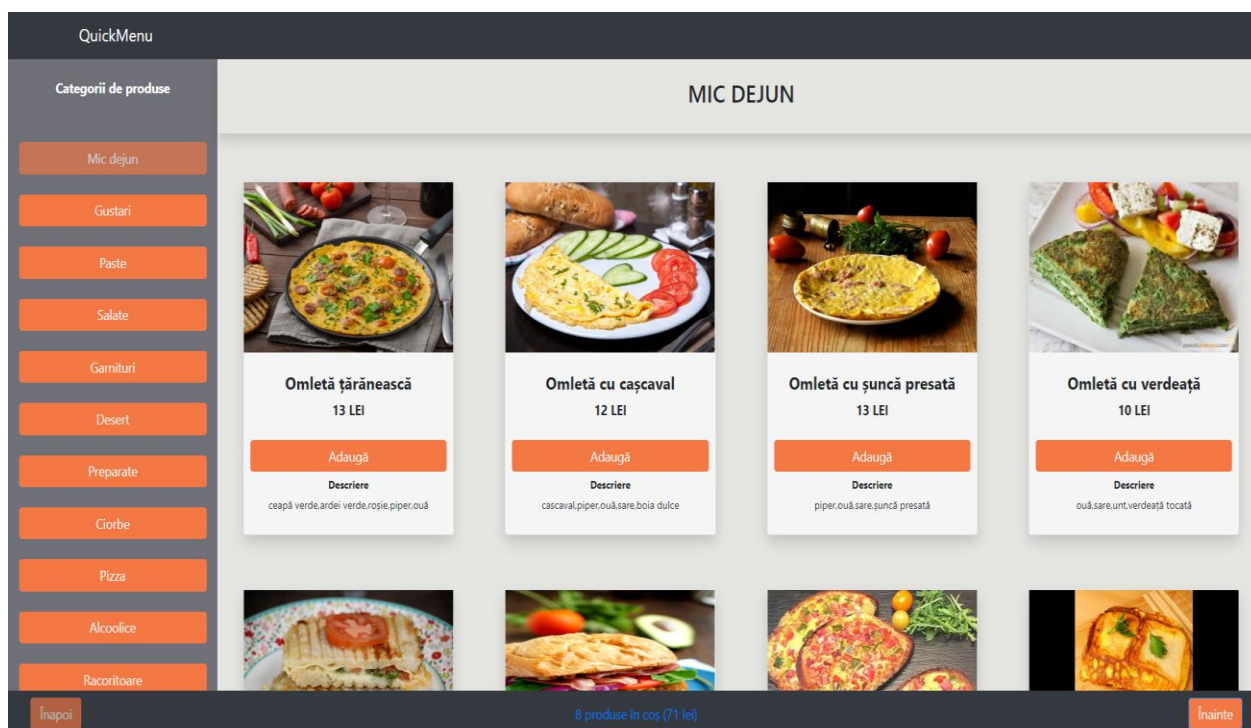


Figura 5: Pagina de configurare a comenzii

După cum se poate observa și în “Figura 5”, pe pagină de configurare a unei comenzi întâmpinăm în partea stânga categoriile de produse disponibile în restaurant, însoțite în partea dreapta cu produsele aferente categoriilor.

Acțiunile pe care le putem efectua în această pagină sunt următoarele: putem naviga printre categorii unde vom găsi numeroase produse conform categoriei alese, navigarea se poate realiza fie prin apăsarea butonului aferent categoriei dorite fie selectând unul din butoanele “Înainte” respectiv “Înapoi”, o altă acțiune este adăugarea produselor în coș, ce se realizează prin apăsarea butonului “Adaugă”. În același timp putem urmări în timp real în partea de jos numărul de produse selectate dar și suma totală a produselor ce au fost adăugate în coș.

Odată ce au fost selectate produsele din categoriile preferate tot ce mai rămâne este să finalizăm comanda, aceasta trebuind să ajungă la bucătar.

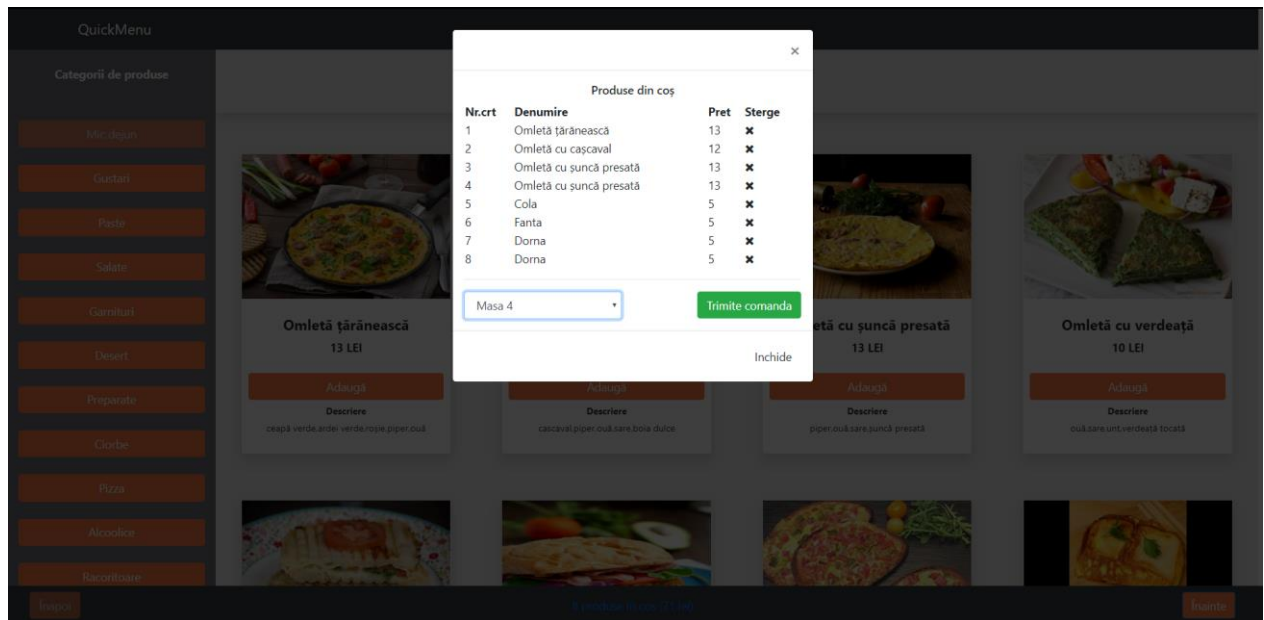


Figura 6: Coșul cu produsele selectate

În partea de jos unde vedem ce produse am adăugat în coș și suma totală a produselor din coș putem finaliza comanda simplu apăsând pe numărul de produse, moment în care se va deschide un modal că în “Figura 6” pentru a ne putea vizualiza produsele selectate.

Odată deschis modalul tot ce mai este de făcut este să ne verificăm comandă, în cazul în care ceva este un plus sau ne răzgândim, foarte ușor putem renunța la un produs prin apăsarea pe butonul “X”. Fără doar și poate dacă dorim să mai adăugăm și alte produse pe lângă cele deja selectate închidem modalul și continuăm căutarea de alte produse pentru a le adăuga. Țin să menționez că este foarte important să selectăm și masă, după cum am menționat mai sus fiecare masă este însoțită pe lângă tableta cu aplicație și de un număr aferent mesei la care se situează clientul, astfel în stânga butonului “Trimite comanda” printr-o simplă apăsare se va deschide o lista cu toate mesele din restaurant, tot ce mai e de făcut fiind să selectați numărul mesei la care vă aflați.

În cazul în care apăsați pe “Trimite comanda” fără a selecta masă operațiunea se va opri și va atenționa că nu ați selectat masă printr-un mesaj că în “Figura 7”.



Figura 7: Mesaj de atentionare pentru neselectare masa

Am ajuns în momentul în care totul este selectat și tot ce mai ramane e să finalizăm comanda și să așteptăm servirea, astfel printr-o simplă apăsare a butonului “Trimite comanda” bucătarului o să-i apară comanda dumneavoastră și o va efectua.

Aceasta este partea “Front-office”, parte la care clientul are acces dar și ospătarul, în cazul selectării opțiunii “Chelner”. Am încercat să construiesc un mers al lucrurilor simple, care se pot deduce foarte ușor, rapid și nu în ultimul rând eficient. Culoarele alese au fost în funcție de tema aplicației care este una bazată pe mâncăruri, portocaliu fiind predominant deoarece este o culoare ce provoacă confort și este asociat mâncării sănătoase.

3.1.2 Partea back-office

Back-office este pagina de administrare a personalului. Clienții nu au acces la ea, fiind folosită pentru a realiza managementul ingredientelor, produselor, comenzilor primite și a celor finalizate.

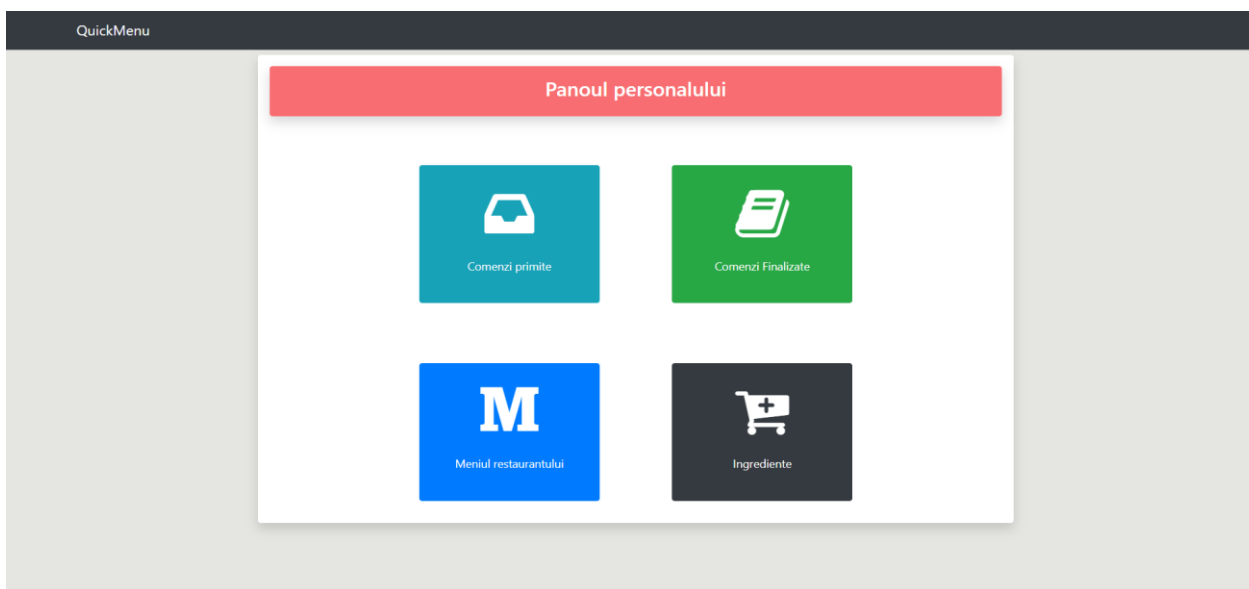


Figura 8: Panoul personalului

După ce clientul și-a finalizat comanda, aceasta a fost salvată și urmează să fie procesată de bucătar. Panoul personalului din “Figura 8” este descris prin patru butoane cu caracteristici ce au legătură între ele.

Voi vorbi mai întâi despre secțiunea “Ingrediente” ce conține o listă cu toate ingredientele din bucătărie din care urmează să se realizeze fiecare dintre produsele de pe pagină din “Figura 5”.

Nr. Crt.	Denumire	Cantitate	
25	verdeată tocată	543534g	
24	unt	54353g	
23	șuncă presată	543534g	
22	boia dulce	235435g	
21	carne tocată	54354353g	
20	piper	54353g	
19	roșie	5435435g	
18	cimbru	5435345g	
17	mărar	5435435g	
16	ardei verde	543543g	
15	ceapă verde	543534g	
14	salam	535636g	
13	paste	44545g	

Figura 9: Inventarul/Stocul de produse

Fiecare ingredient va fi însoțit de o denumire și cantitatea cântărită în grame care se afla în bucătărie. Dacă un ingredient este nou în bucătărie, utilizatorul îl poate adăuga la secțiunea “Adaugă ingredient”.

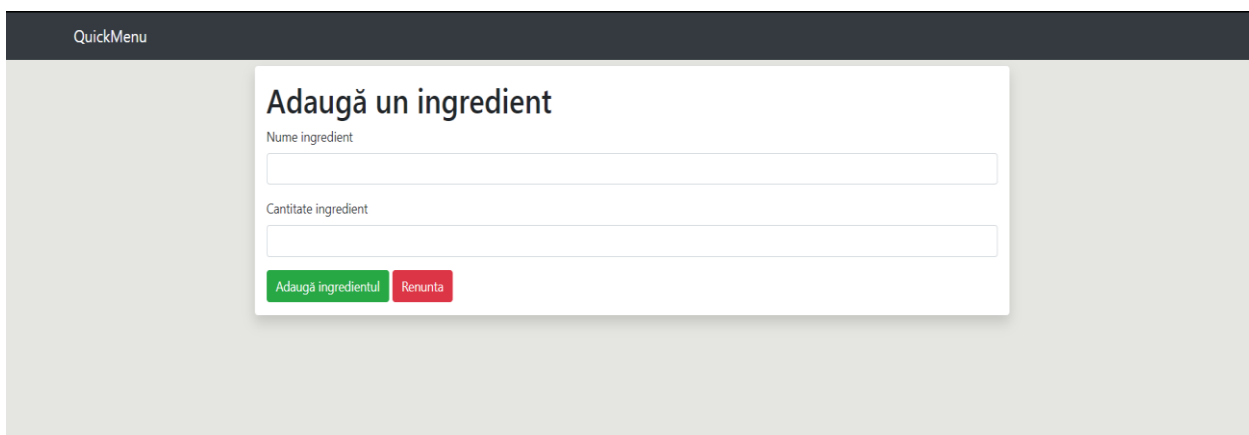


Figura 10: Adaugă un ingredient

Există cazul în care numele unui ingredient a fost introdus greșit, cantitatea a fost introdusă greșit sau urmează să se mai adauge pe stoc un anumit ingredient, de aceea există posibilitatea de a edita fiecare produs prin butonul de la “Figura 11”. Iar dacă produsul nu se mai folosește în bucătărie acesta poate fi șters prin butonul de la “Figura 12”.



Figura 11 : Buton pentru editare



Figura12 : Buton pentru ștergere

Pentru ca ștergerea unui ingredient să aibă loc trebuie confirmată acțiunea, în cazul în care butonul s-a apăsă din greșeală.

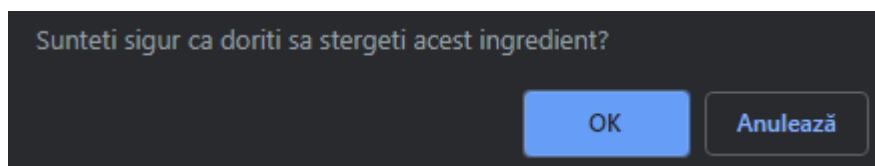


Figura 13 : Confirmare pentru ștergerea unui ingredient

Bucătarul vrea să prepare un produs și nu știe dacă are suficient din ingredientul “X”, astfel el este scutit printr-o simplă căutare precum în “Figura 14” pentru a verifica.

Inventar

Afișează câte 50 randuri pe pagina

Cautare: Ciuperci

Nr. Crt.	Denumire	Cantitate	Adaugă ingredient
8	ciuperci	65654g	 

Pagina 1 din 1 (rezultate filtrate din 25 totale) Pagina precedenta 1 Pagina urmatoare

Figura 14 : Caută ingredient

Pe panoul personalului avem și secțiunea “Meniul restaurantului” ce reprezintă locul unde se pot viziona toate produsele pe care bucătarul le poate prepara cu ajutorul ingredientelor din secțiunea “Ingrediente”.





























QuickMenu					
Produse					
Afișează câte 50 randuri pe pagina		Cautare:			
Nr. Crt.	Denumire	Pret	Ingrediente	Categorie	Adaugă produs
51	Omletă țărănească	13 lei	ceapă verde,ardei verde,roșie,piper	Mic dejun	 
50	Omletă cu cașcaval	12 lei	cascaval,piper,ouă,sare,boia dulce	Mic dejun	 
49	Omletă cu șuncă presată	13 lei	piper,ouă,sare,șuncă presată	Mic dejun	 
48	Omletă cu verdeată	10 lei	ouă,sare,unt,verdeată tocată	Mic dejun	 
47	Sandwich de pui	11 lei	pui,chifla,ceapa,salata,ceapa,cascaval	Mic dejun	 
46	Cola	5 lei		Băuturi răcoritoare	 
45	Fanta	5 lei		Băuturi răcoritoare	 
44	Dorna	5 lei		Băuturi răcoritoare	 
43	Pepsi	5 lei		Băuturi răcoritoare	 
42	Cappy portocale	7 lei		Băuturi răcoritoare	 
41	Cappy piersică	7 lei		Băuturi răcoritoare	 
40	Borsec	5 lei		Băuturi răcoritoare	 
39	Paste cu ton	18 lei	paste,piper,ton,usturoi,masline,sare	Paste	 
38	Paste carbonara	18 lei	paste,piper,masline,sare,bacon	Paste	 

Figura 15 : Pagina cu produsele din meniul

Produsele conțin denumire, preț, ingredientele din care este format produsul respectiv, categoria din care face parte dar și o imagine. Asemănător paginii “Ingrediente”, putem edita sau șterge un produs dar și adaugă.

Adaugă produs

Nume produs

Paste carbonara

Pret produs

18

Ingrediente

x paste x piper x sare

Categorie produs

Mic dejun

Mic dejun

Gustări

Paste

Salate

Garnituri

Desert

Preparate din pui

Preparate din porc

Băuturi alcoolice

Băuturi răcoritoare

Băuturi calde

Ciorbe

Pizza

Preparate din paste

Avatar

Alege fișierele Nu ai ales niciun fișier

Adauga meniu Renunta

Figura 16 : Pagina de adăugare a unui produs

Fiecare comanda ce a fost finalizată de un client va apărea în secțiunea “Comenzi primite” ce este accesata de bucătar.

QuickMenu

Comenzi primite

Comanda numărul: 4 Comanda numărul: 5 Comanda numărul: 6 Comanda numărul: 7 Comanda numărul: 8

Comanda numărul: 9 Comanda numărul: 10 Comanda numărul: 11 Comanda numărul: 12

Inapoi

Figura 17 : Pagina comenzilor finalizate de clienți

Bucătarul va executa comenzile în ordinea primirii lor, prima comandă fiind ultima primită, ele adăugându-se la coada.

QuickMenu

Comanda cu numarul: 13

Inapoi

Masa cu numarul: 5
Total comanda: 37 lei

Afiseaza cate 50 randuri pe pagina

Cautare:

Nr. Crt.	Denumire	Portii/Bucati
2	Omletă cu cașcaval	2
1	Omletă țărănească	1

Pagina 1 din 1

Pagina precedenta

1

Pagina urmatoare

(*) Adăugați numărul de grame din fiecare ingredient pe care le-ai folosit pentru produsele de mai sus.

ceapă verde	ardei verde	roșie	pipere
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
cașcaval	ouă	sare	
<input type="text"/>	<input type="text"/>	<input type="text"/>	

Finalizare

Figura 18 : Pagina unei comenzi

În pagină unei comenzi vom găsi numărul comenzii, masa de la care s-a făcut comanda, totalul de plată al comenzii, denumirea produselor și numărul de porții/bucăți aferente produselor.

În partea de jos bucătarul trebuie sa introducă numărul de grame pe care l-a folosit pentru fiecare produs în parte pentru a reuși sa mențină stocul ingredientelor. După ce a completat spațiile urmează sa finalizeze comanda clientului, astfel apăsând pe butonul “Finalizează”, comanda va fi plasată în secțiunea cu “Comenzi finalizate”.

Comenzi finalizate

Comanda numarul: 1

Comanda numarul: 2

Comanda numarul: 3

Comanda numarul: 13

Inapoi

Figura 19 : Pagina comenzilor finalizate

Această pagină este urmărită de ospătari pentru a urmări când bucătarul termină de făcut o comandă. Dacă apare o comandă precum una din cele din “Figura 19” ospătarul o poate vizualiza

asemănător cu ceea ce a vizualizat bucătarul, și în funcție de masa acesta se va îndrepta spre ea pentru a servi comanda.

Comanda cu numarul: 13 este gata de servire!

Inapoi

Masa cu numarul: 5
Total comanda: 37 lei

Afiseaza cate 50 randuri pe paginaCautare:

Nr. Crt.	Denumire	Portii/Bucati
2	Omletă cu cașcaval	2
1	Omletă țărănească	1

Nr. Crt.

Denumire

Portii/Bucati

Pagina 1 din 1

Pagina precedenta1Pagina urmatoare

Figura 20 : Pagina unei comenzi finalizate

4 Detalii de implementare

Pentru afișarea produselor, am creat un array cu numele “meniu” ce are ca elemente fiecare dintre categoriile de produse. Fiecare element din “meniu” reprezintă la rândul lui un alt array conținând toate produsele din baza de date conform numelui categoriei. Astfel am reușit să împărțim produsele pe categorii pentru a fi mai ușor de afișat.

După cum ați putut vedea în “Figura 5”, în funcție de ce categorie selectăm din partea stânga, vor fi afișate produsele categoriei selectate însoțite de numele ei. Acest lucru a fost posibil realizând mai multe pagini pentru fiecare categorie, afișată fiind pagină cu categoria selectată iar restul fiind ascunse.

Aceeași acțiune se poate realiza și prin apăsarea butoanelor “Înainte” respectiv “Înapoi”, schimbarea paginilor făcându-se în ordinea în care sunt afișate categoriile.

```

function go_to_category(cat)
{
    $('#catContainer').hide();
    $('#btn').prop('disabled',false);
    currentCategory=cat;
    $('#categorie_'+categories[currentCategory-1]).hide();
    $('#categorie_'+categories[currentCategory]).show();

    if(maxCat==(currentCategory+1)){
        $('#btn_next').html("Finalizeaza comanda").removeClass().addClass("btn btn-success").attr('onclick','checkout()');
        $('#btn_prev').html("Înapoi").prop('disabled',false);
    }else if(currentCategory==0){
        $('#btn_prev').html("Înapoi ").prop('disabled',true);
        $('#btn_next').html("Înainte").removeClass().addClass("btn btn-primary").attr('onclick','next_cat()');
    }else{
        $('#btn_prev').html("Înapoi").prop('disabled',false);
        $('#btn_next').html("Înainte").removeClass().addClass("btn btn-primary").attr('onclick','next_cat()');
    }
    $('#btn_'+categories[currentCategory]).prop('disabled',true);
}

```

Figura 21 : Asignarea categoriei pentru afișarea produselor

În scopul realizării coșului de produse, am construit un array ce are ca elemente fiecare produs din lista la care s-a realizat acțiunea “Adaugă produs”, în același timp calculând-se și suma totală a comenzii, adăugându-se prețul fiecărui produs .

```

function add_to_cart(productInfo){
    cart.push(productInfo);
    productInfo=JSON.parse(productInfo);
    cartTotal+=parseFloat(productInfo['menu_pret']);
    $('#cart_details').html(cart.length +' produse în coș ('+ cartTotal +' lei)');
    $("#cartProducts tbody").append("<tr><td>"+cart.length+"</td><td>"+productInfo['menu_nume']+"</td><td>"+productInfo['menu_pret']+"</td><td><i class='fa fa-remove' onclick='remove_product(\"+(cart.length-1)+\")'></i></td></tr>");
}

```

Figura 22 : Adăugarea produselor în coș

Odată ce produsele au fost adăugate în coș, urmează finalizarea comenzii unde se apelează funcția “sendOrder()” ce pregătește datele de care sunt nevoie pentru a fi adăugate în baza de date.

Comanda a fost plasată, iar aceasta îi este afișată bucătarului după cum ați văzut în “Figura 18”. Datele sunt modelate astfel încât fiecărui produs să îi fie asignat numărul de porții/bucăți, dar și fiecare ingredient ce compune produsele comenzii însoțite de o input unde bucătarul la sfârșit va adăuga numărul în grame de ingrediente folosit pentru comanda respectivă. Am ales că stocul să

fie modificat de bucătar, deoarece în momentul în care o comandă a fost realizată este momentul final în care s-au folosit ingredientele, astfel bucătarul după ce finalizează comanda, în baza de date se va scădea numărul de grame introdus aferent fiecărui ingredient acesta apărând pe pagină o singură dată, chiar dacă s-a folosit la două produse diferite același ingredient.

```
public function comanda_finalizata($comanda_id){
    $comanda = $this->comenzi_model->getComandaById($comanda_id);
    $i = count(explode(",",$comanda->comanda_ingrediente));

    for($key=0;$key <$i ;$key++){
        $grame[$key] = $this->input->post("ingredient_". $key);
    }

    foreach(explode(",",$comanda->comanda_ingrediente) as $key=>$ing){
        $grame_din_ingrediente = $this->produse_model->getProductByName($ing);
        $grame_pentru_update = (int)$grame_din_ingrediente[0]->produs_cantitate - (int)$grame[$key];
        $produse_cantitate= ['produs_cantitate' => (int)$grame_pentru_update];
        $update_cantitate = $this->produse_model->updateCantitate($ing,$produse_cantitate);
    }

    $data = [
        'comanda_id'      => $comanda_id,
        'comanda_remove' => 1
    ];
    $this->comenzi_model->changeComandaRemove1($data,$comanda_id);
    header('Location: '.$this->config->item('base_url'). 'index.php/comenzi/comenzi_prime');
}
```

Figura 23 : Finalizarea unei comenzi și modificarea stocului

Bucătarul și ospătarul dețin pagini diferite pentru observarea statusului comenzilor, astfel am făcut posibilă o modalitatea de a vedea comenzile primite de bucătar și comenzile finalizate de ospătar, adăugând în baza de date o coloană numită “comenzi_remove” ce primește două valori, “0” respectiv “1”, “0” reprezentând că o comandă este în curs de pregătire, iar “1” că o comandă este finalizată și gata de servire.

Aplicația dispune și de validate de date atunci când sunt introduse de utilizatori, acest lucru fiind verificat și din backend și din frontend. Din backend a fost realizată cu ajutorul unei clase pe care o oferă framework-ul Codeigniter, și anume o clasă completă de validare a formularelor pentru pregătirea datelor care ajută la reducerea cantității de cod.


```

public function editAction()
{
    $this->form_validation->set_rules('menu_nume', 'Nume produs', 'required');
    $this->form_validation->set_rules('menu_pret', 'Pret produs', 'required');
    $this->form_validation->set_rules('menu_categorie', 'Categorie produs', 'required');
    $this->form_validation->set_message('required', 'Campul "{field}" este obligatoriu.');
```

//validare date

```

    if ($this->form_validation->run() == false) {
        //incarca un produs
        $this->load->view(
            'menu/edit.php',
            ['menu' => $this->menu_model->getProductById($menu_id), 'post' => 1]
        );
    } else {

        $data = array(
            'menu_nume'          =>$this->input->post('menu_nume'),
            'menu_pret'          =>$this->input->post('menu_pret'),
            'menu_categorie'     =>$this->input->post('menu_categorie'),
            'menu_ingredient'    =>implode(",",$this->input->post('menu_ingredient')),
        );

        // try to update source
        if ($this->menu_model->updateMenu($menu_id, $data)) {
            header('Location: '.$this->config->item('base_url').'index.php/menu/menu');
        } else {
            header('Location: '.$this->config->item('base_url').'index.php/menu/menu');
        }
    }
}

```

Figura 24 : Validarea datelor pe partea backend înainte de a fi adăugate în baza de date

Pe partea de frontend a fost realizată cu ajutorul funcției “required”.

```

<input type="text" class="form-control" id="produs_nume" name="produs_nume" value="<?php echo set_value('produs_nume'); ?>" required>

```

Figura 24 : Validarea datelor pe partea frontend înainte de a fi adăugate în baza de date

5 Diagrame

5.1 Arhitectura aplicației

Este împărțită pe două module separate, dar cu caracteristici comune, ele comunicând cu ajutorul bazei de date. Primul modul, front-office, este modulul utilizat de clienții restaurantului și are următoarea arhitectura:

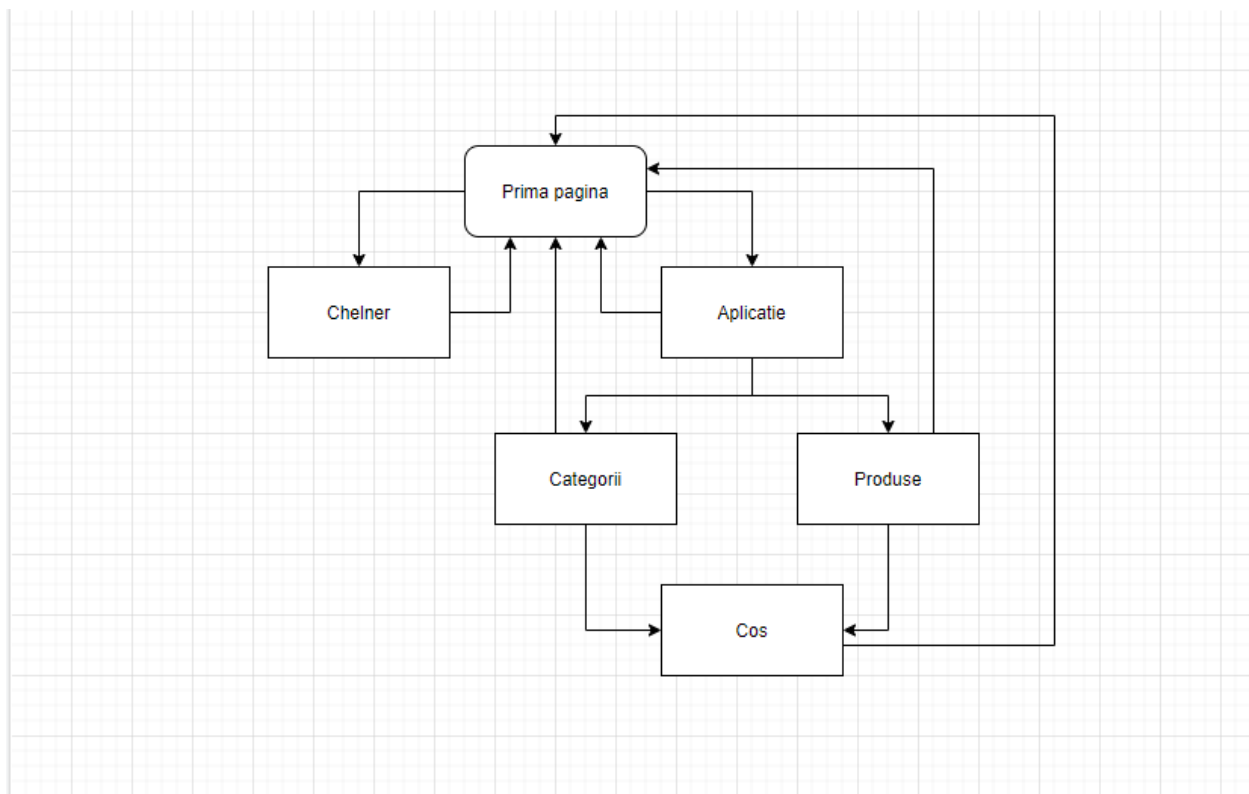


Figura 25 : Diagrama arhitecturală a modulului front-office

Clientul aflându-se în oricare din paginile din front-office, se poate întoarce la pagina principală în cazul schimbării opțiunii modului de comandare.

În figura de mai jos se poate urmări arhitectură back-office a aplicației:

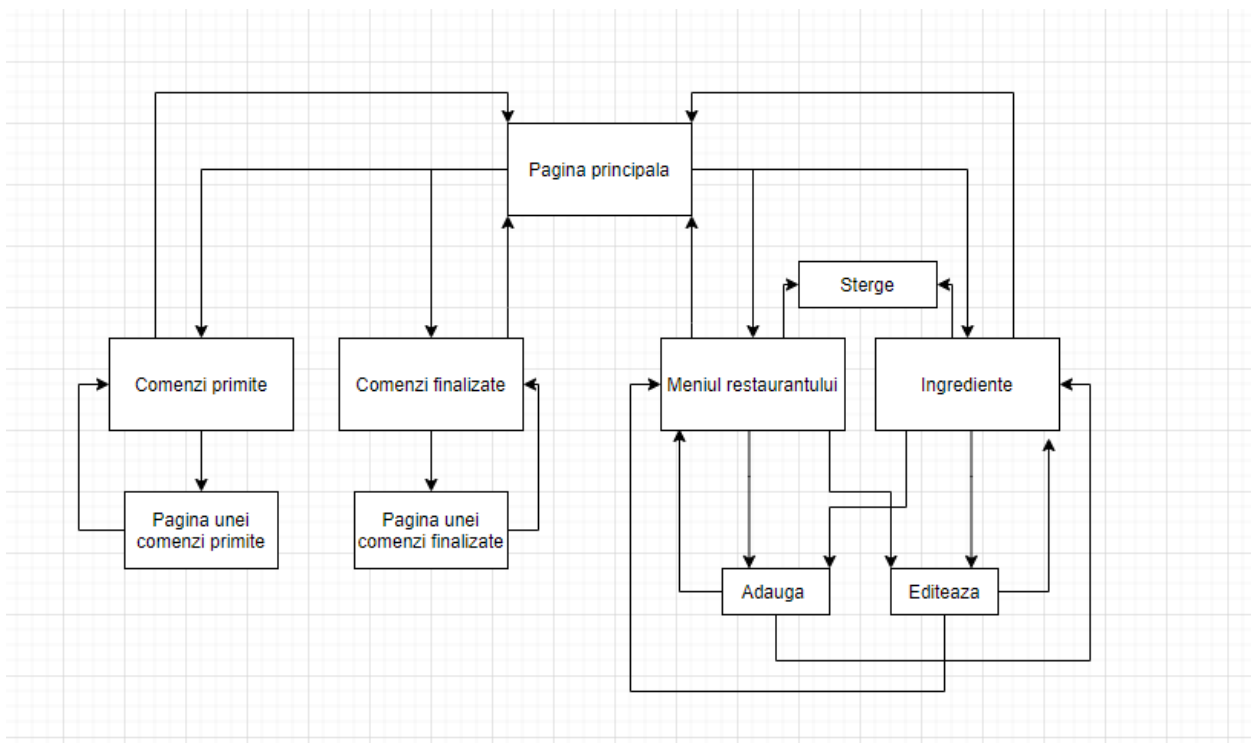


Figura 26 : Diagrama arhitecturală a modului back-office

5.2 Diagrame use-case

Aplicația nu necesită autentificare, utilizatorii împărțindu-se în două categorii. Astfel acțiunile clienților sunt limitate, pe când angajații pot folosi modulul clienților pentru a plasa o comandă în cazul selectării secțiunii “Chelner” de către client.

În figura de mai jos se pot urmări acțiunile pe care un client le poate săvârși:

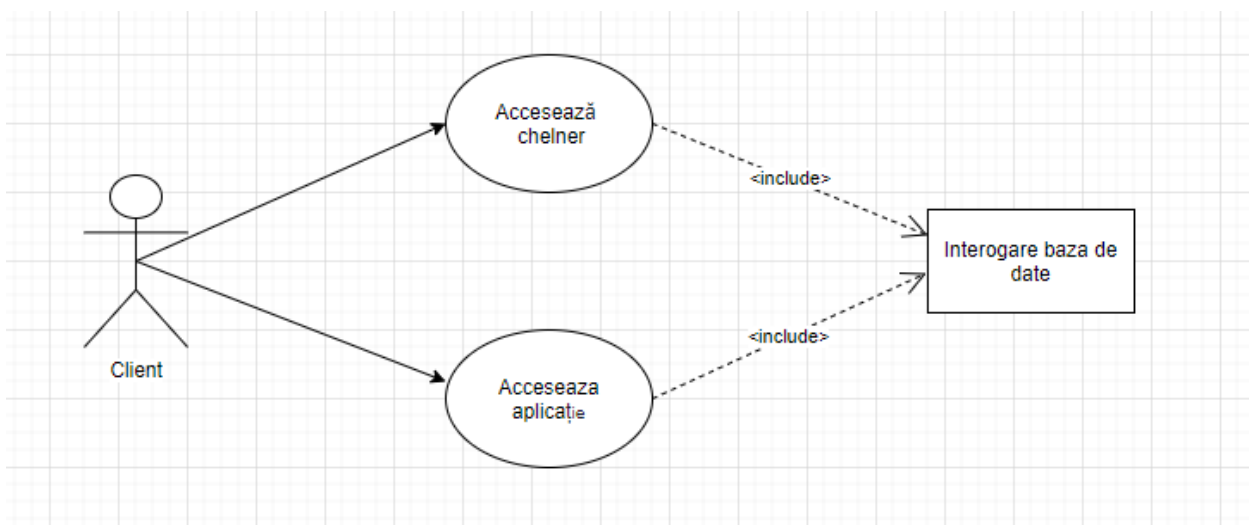


Figura 27 : Diagrama use-case ce generează interogări la baza de date

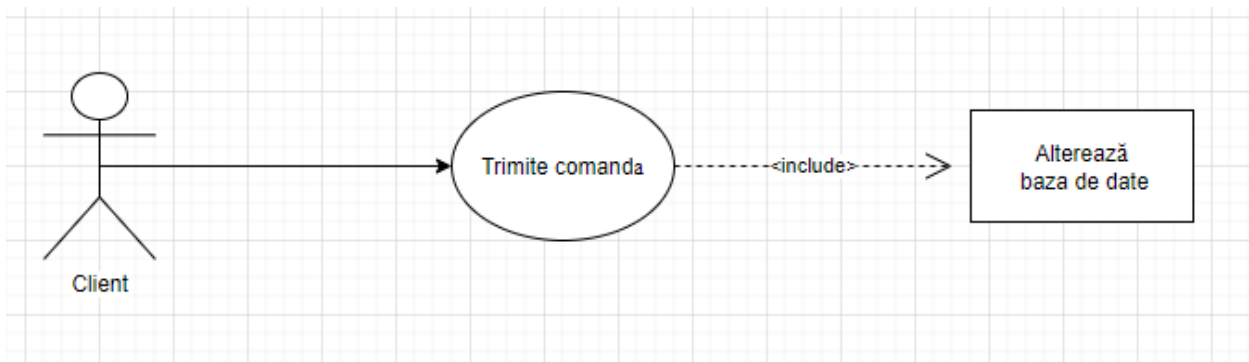


Figura 28 : Diagrama use-case ce alterează baza de date

Acțiunile pe care angajații le pot săvârși sunt următoarele:

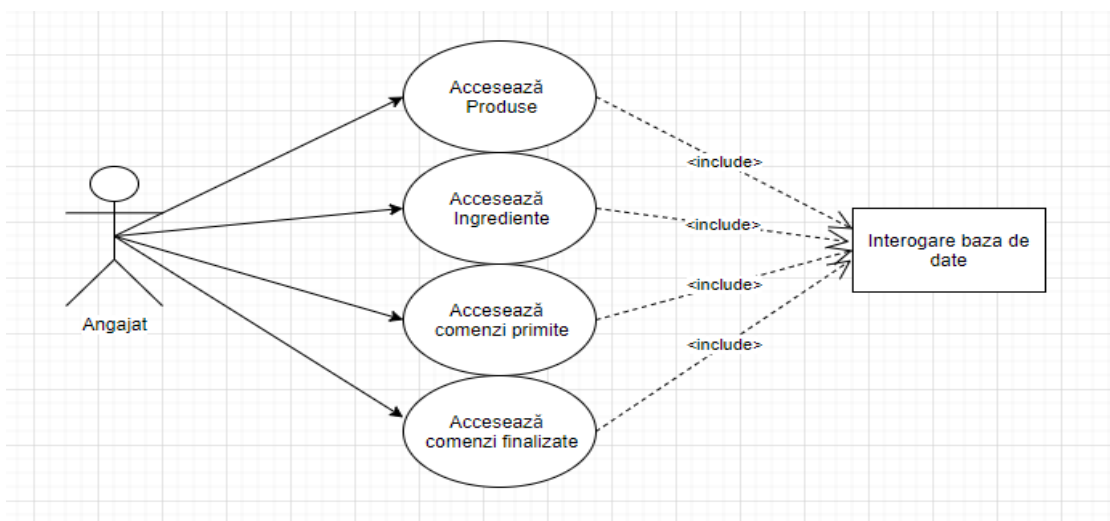


Figura 29 : Diagrama use-case ce gestionează interogări în baza de date

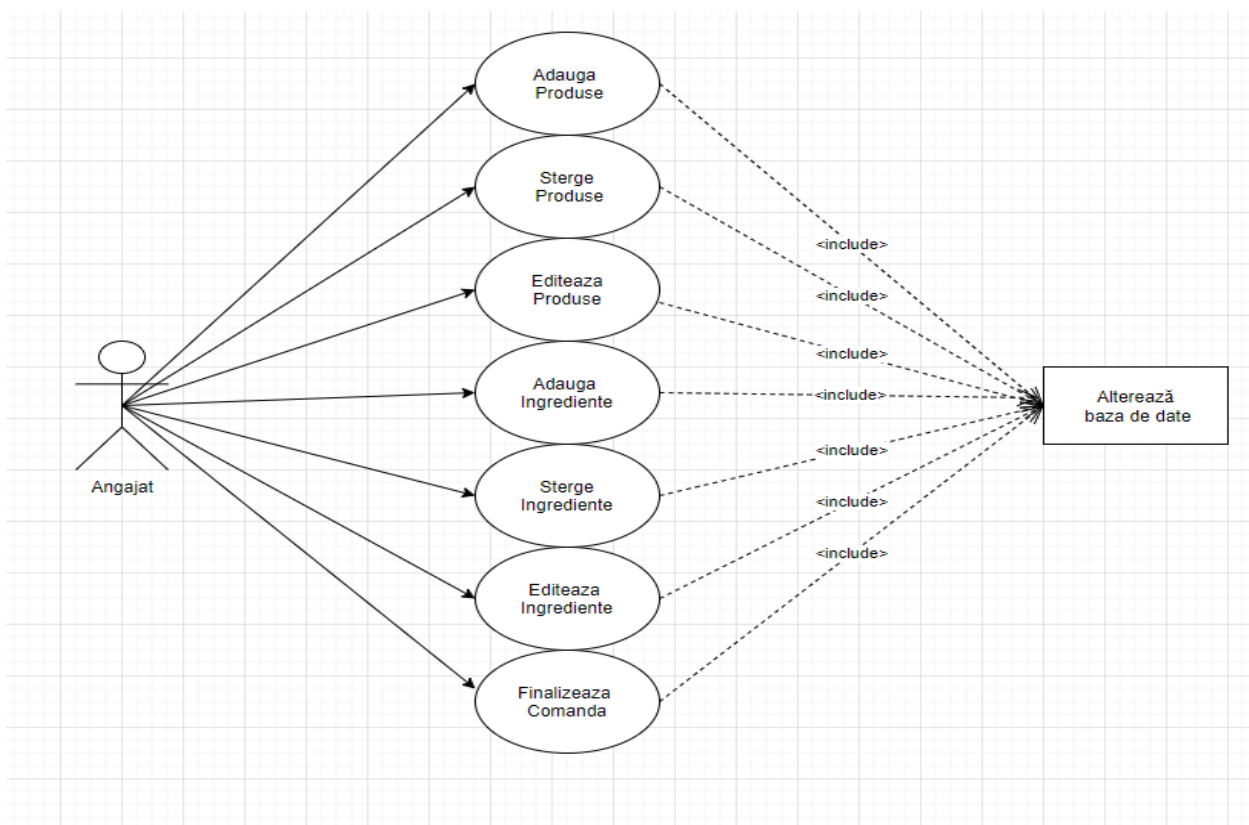


Figura 30 : Diagrama use-case ce alterează baza de date

5.3 Relaționarea bazei de date

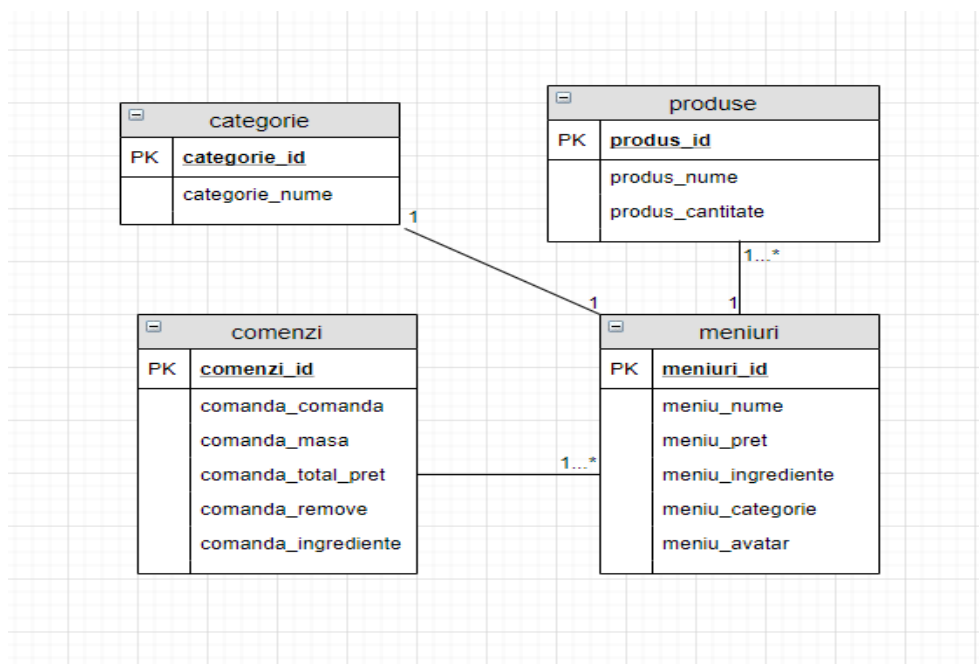


Figura 31 :Relaționarea bazei de date

Concluzii

Conform aspectelor prezentate în capitolele de mai sus, pot afirma că obiectivul lucrării a fost atins, soluția propusă răspunzând unei nevoi reale adaptat tehnologiei din zilele noastre, pe scurt gestionarea comenzilor în cadrul unei unități de alimentație publică.

- Analiza pe care am avut-o asupra unui caz real, și ascultând nevoile unui proprietar al unui restaurant, m-au motivat să fac posibilă realizarea unei aplicații ce a avut ca scop principal să contribuie la optimizarea timpului de așteptare în procesul unei comenzi, acest aspect reprezentând cel mai important factor ce contribuie la îmbunătățirea unui business al unei unități de alimentație publică.
- Un alt factor important este faptul că meniul digital oferă o perspectivă detaliată asupra produselor, mai ales dacă aceasta este și una prietenoasă. Consider că am atins acest aspect folosind unele dintre cele mai noi și performanțe tehnologii reușind să ofer o interacțiune plăcută a clientului cu aplicația, dar nu numai, chiar și angajaților, acest fapt crescându-le interesul de a o folosi și munci cu drag.
- Prin urmare, aplicația “ **Quick Menu** ” este unealtă ce vine în ajutorul oricărui restaurant care dorește să-și crească productivitatea sau să elimine problemele interioare, oferind o alternativă rapidă și profitabilă în cadrul unui proces în care timpul și resursele sunt foarte importante.

Direcții viitoare

Aplicația poate fi îmbunătățită adăugând următoarele:

- Un sistem de gestionare a clădirii împreună cu mesele din unitate, totodată afișându-se mesele care sunt ocupate, respectiv libere.
- Posibilitatea unei comenzi online, acest lucru fiind posibil cu un modul separat după selectarea produselor.
- Gestionarea unor conturi pentru angajați împreună cu un sistem de rating unde pot fi vizualizați angajații care au avut o activitate mai sporită.
- Prezicerea unui stoc săptămânal sau lunar ușurând inventarul ingredientelor.
- Aplicația să poată fi descărcată pe dispozitive personală fiind accesată pe baza unei scanări a unui cod de bare prezent la masă.

Bibliografie

1. Codeigniter Documentation - <https://codeigniter4.github.io/userguide/>
2. PHP Documentation- <https://www.php.net/docs.php>
3. Visual Studio Code Documentation - <https://code.visualstudio.com/docs>
4. Draw.io - <https://www.draw.io/>
5. Fast Order Application - <https://www.expressoft.eu/vanzare-cloud/restaurant>
6. Bootstrap Documentation - <https://getbootstrap.com/docs/4.4/getting-started/introduction/>
7. Navicat Documentation - <https://www.navicat.com/en/support/online-manual>
8. jQuery Documentation - <https://api.jquery.com/>
9. JavaScript Documentation - <https://devdocs.io/javascript/>