

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică



Aplicație de tip RESTful API - Client

Profesor coordonator: Conf.univ.dr. Furtună Titus Felix

Student: Sicu Bogdan Andrei

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Cuprins

1) Introducere	4
2) Prezentarea domeniului	5
2.1) Istoria cateringului	5
2.2) Istoria internetului	7
2.3) Istoria comerțului electronic	8
2.4) Alte firme de catering	9
3) Prezentarea tehnologiilor	12
3.1) Front-end	12
3.1.1) JavaScript	12
3.1.2) HTML	12
3.1.3) CSS	13
3.1.4) TypeScript	14
3.1.5) Angular framework	14
3.2) Back-end	15
3.2.1) Java	15
3.2.2) Spring	16
3.2.3) Spring Boot	16
3.2.4) PostgreSQL	17
3.3) Alte tehnologii folosite în domeniul aplicațiilor de catering/E-commerce	17
3.3.1) React	17
3.3.2) Aplicații pentru telefon	18
3.3.3) ASP.NET Core	18
4) Prezentarea soluției informatice	18
4.1) Descrierea generală a sistemului informatic	18
4.2) Specificarea cerințelor sistemului informatic	19
4.3) Analiza sistemului informatic	20
4.3.1) Diagrame de activitate	20
4.3.2) Diagrame de procese și colaborare în BPMN	24
4.4) Proiectarea sistemului informatic	26
4.4.1) Diagrama de clase detaliată	26
4.4.2) Proiectarea bazei de date	27

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

4.4.3) Proiectarea interfetelor utilizator	28
4.5) Sistemul informatic implementat.....	29
4.5.1) Componenta backend (RESTful Api).....	30
4.5.2) Componenta frontend (client).....	36
Concluzie	41
Bibliografie	42

1) Introducere

Sistemul este reprezentat de două programe care comunică unul cu celălalt pentru a forma o aplicație de comerț pe internet îndreptată spre servirea de mâncăruri. Aplicația are ca scop expunerea ofertelor culinare ale unui restaurant și posibilitatea de comandare a acestora.

Secțiunea de start a aplicației conține antetul în care se află logo-ul și butoanele pentru accesarea paginilor de coș de cumpărături, contact și de conectare la contul utilizatorului. De asemenea, pe această pagină se află și o listă a ofertelor culinare prezente pentru restaurantul respectiv, însoțită de un motor de cautare de tip text.

Apăsând pe un produs, se încarcă o secțiune pentru prezentarea acestuia unde se pot observa diverse date, cum ar fi ingredientele folosite, prețul, poza produsului și opțiunea de a fi adăugat în coșul de cumpărături și/sau la secțiunea de mâncăruri favorite ale utilizatorului curent.

Secțiunea de contact conține date relevante despre restaurant, cum ar fi numărul de telefon pentru contact, adresa de mail, locația restaurantului și orele de funcționare.

Secțiunea căruciorului este locul în care utilizatorul poate să vizualizeze produsele pe care le are momentan în coșul de cumpărături, precum și prețul total al acestora. Tot aici există și posibilitatea ștergerii produselor pe care posibilul cumpărător nu le mai dorește.

Secțiunea profilului utilizatorului conține datele personale ale acestuia, precum și posibilitatea de a le modifica sau de a șterge contul cu totul. Printre datele sensibile se află: număr de telefon, nume, mail și adresa de livrare.

2) Prezentarea domeniului

Aplicațiile de tip comerț pe internet au foarte multe avantaje, acest fapt fiind dovedit atât de abundența lor în contextul actual, cât și de succesul răsunător pe care l-au avut în timpul pandemiei de COVID-19. Una dintre cele mai de succes industrii ale comerțului pe internet a fost și este în continuare industria de catering, acesta fiind și domeniul de activitate pentru soluția software prezentată. Atât avantajele cât și dezavantajele unei astfel de soluții se aplică pentru toate tipurile de business, indiferent dacă sunt îndreptate spre comerțul cu alimente sau spre comerțul cu bunuri nealimentare.

Printre avantaje se numără următoarele:

- posibilitatea extinderii afacerii și către utilizatorii care nu se află în proximitatea locației de desfășurare a activității economice;
- posibilitatea de expunere a reclamelor;
- accesibilitatea plății electronice

Cu toate acestea, există și dezavantaje, cel mai mare dezavantaj fiind costul de producție și de întreținere a unei astfel de soluții software, fiind greu accesibilă firmelor mici.

2.1) Istoria cateringului

Istoria cateringului datează încă din antichitate unde bogații vremii găzduiau banchete cu zeci, chiar sute de oameni. În Egiptul antic și Roma antică spre exemplu, erau pregătite sărbători care mai de care mai elaborate pentru regalitate și nobilime. Asemănător, în Grecia antică, simpozioanele erau găzduite de oamenii bogați unde oaspeții mâncau și se distrau.

Înaintând pe axa temporală până în timpul Evului Mediu, în Europa erau adesea găzduite petreceri de către regi sau nobilime. Asemeni zilelor noastre, petrecerile mari erau ținute în locuri speciale precum castele sau săli spațioase unde era nevoie de serviciile unui grup de servitori, bucătari și animatori.

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Odată cu evoluția rutelor comerciale s-au diversificat și influențele culinare, răspândind aromele din culturi și țări diferite. Totodată, creșterea nivelului de trai al clasei de mijloc a dus la o dezvoltare accentuată a cererii de servicii de catering pentru diverse evenimente precum banchete, nunți și alte petreceri.

Sectorul de catering a cunoscut schimbări suplimentare ca urmare a Revoluției Industriale. Utilizarea tehnicilor de producție în masă a permis firmelor de catering să se organizeze mai eficient și să deservească mai mulți oameni. Ca urmare, cateringul instituțional a devenit din ce în ce mai popular, școlile, spitalele și alte instituții apelând la aceste servicii pentru a servi mesele pentru mulțumirea oamenilor.

Cateringul s-a dezvoltat și mai mult în secolul al XX-lea, pe măsură ce au apărut noi tehnologii și moduri de transport. Întreprinderile de catering la scară largă care puteau furniza alimente pentru evenimente din întreaga țară și chiar din lume, au fost posibile prin progresele în refrigerare și transport.

Industria de catering este astăzi un sector de miliarde de dolari, având în compoziție de la mici firme de familie până la conglomerate globale masive. Nunțile, întâlnirile de afaceri, festivalurile și evenimentele sportive sunt doar câteva dintre evenimentele pentru care sunt oferite servicii de catering. Cu un accent din ce în ce mai mult pus pe alimente organice, pe bază de plante și produse la nivel local, afacerea s-a schimbat, de asemenea, pentru a se adapta cerințelor și gusturilor nutriționale în schimbare ale consumatorilor.

Pe scurt, dezvoltarea cateringului a fost un proces lung și intrigant, care a fost influențat de o varietate de factori culturali, progrese tehnologice și circumstanțe sociale și economice în schimbare. Cateringul a fost un factor cheie în aducerea oamenilor împreună pentru a sărbători și a se bucura de mâncare bună și de a se distra la ocazii, de la sărbători antice până la întâlniri corporatiste contemporane.

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

2.2) Istoria internetului

World Wide Web, cunoscut adesea sub numele de internet, a avut începuturile la sfârșitul anilor 1960 odată cu crearea rețelei ARPA (Advanced Research Projects Agency), un proiect al departamentului de apărare al SUA. Scopul acestui sistem de internet timpuriu, cunoscut sub numele de ARPANET, a fost de a oferi o rețea de comunicare sigură și eficientă între organizațiile guvernamentale, institutele militare de cercetare și instituțiile academice.

Primul sistem de e-mail a fost creat în anii 1970, iar ARPANET a permis transferul de fișiere între computerele conectate. Un protocol standard a fost necesar pentru a conecta rețelele proprii ale numeroaselor universități și institutii de cercetare până la începutul anilor 1980. Protocolul de control al transmisiei și protocolul Internet (TCP/IP), care sunt elementele de bază ale internetului așa cum îl cunoaștem astăzi, au fost create pentru a remedia acesta problema.

Popularitatea internetului a crescut într-un ritm de neegalat în anii 90. Creșterea site-urilor web comerciale, a platformelor de comerț electronic și a reclamelor online a avut loc în această perioadă de timp, cunoscută în mod obișnuit ca boom-ul dot-com. Apariția browserelor web precum Internet Explorer și Mosaic a făcut internetul foarte accesibil și ușor de utilizat, ceea ce a dus la acceptarea sa pe scară largă în întreaga lume. Primul motor de căutare, cunoscut sub numele de Archie, a fost lansat în 1996 și a simplificat cautarea de informații online pentru consumatori, fără a trebui să comute între numeroase site-uri web.

Apariția unor site-uri web precum Friendster, MySpace și în cele din urmă Facebook la începutul anilor 2000 a transformat internetul într-o platformă de interacțiune socială. Utilizarea pe scară largă a rețelelor sociale a avut un impact uriaș asupra modului în care oamenii interacționează unii cu ceilalți sau cu organizațiile și cu guvernul.

De atunci, internetul a continuat să crească și să evolueze chiar și mai mult odată cu apariția dispozitivelor mobile, cum ar fi smartphone-urile și tabletele, având ca rezultat dezvoltarea de site-uri și aplicații prietenoase cu acest fel de dispozitive. Astăzi, miliarde de oameni folosesc internetul ca instrument necesar pentru comunicare, educație, divertisment și afaceri. În următorii ani, noile tehnologii precum inteligența artificială, învățarea automată și altele asemenea promit să facă internetul și mai eficient și mai fascinant.

2.3) Istoria comerțului electronic

Termenul „e-commerce”, care înseamnă „comerț electronic”, descrie schimbul de produse și servicii pe internet. Cumpărăturile online, plățile electronice și comunicarea digitală fac parte din acest proces. În ciuda faptului că termenul de „e-commerce” este relativ nou, rădăcinile sale pot fi găsite în anii 60, când firmele au început să folosească schimbul electronic de date (EDI) pentru a comunica documente și informații comerciale cu alte întreprinderi.

Transferul electronic de fonduri (EFT), o variantă mai avansată a EDI, a fost implementat în anii 80. A facilitat companiilor posibilitatea de a transfera bani în mod digital între bănci, eliminând cerința pentru cecurile reale pe hârtie. Cu toate acestea, doar firmele mari își puteau permite să dezvolte și să utilizeze EDI și EFT din cauza complexității și costului acestora.

Câțiva retaileri majori au început să-și stabilească prezența online la mijlocul anilor 90, în special Amazon, care a fost înființat în 1994 și a ajuns rapid în fruntea clasamentului magazinelor online. O altă dezvoltare semnificativă a fost crearea unor metode de plată online sigure, cum ar fi PayPal, care a permis clienților să facă plăți electronice pentru bunurile lor. Acesta a fost un moment esențial pentru comerțul electronic, deoarece a abordat securitatea, una dintre principalele griji pentru potențialii cumpărători online.

Dezvoltarea tehnologiei mobile și a internetului a coincis cu creșterea cumpărăturilor online. De-a lungul timpului, acest lucru a făcut posibil ca mai multe persoane să acceseze internetul folosind o varietate de dispozitive, cum ar fi smartphone-uri și tablete. M-commerce, sau comerțul mobil, a fost creat ca urmare a creșterii utilizării internetului de pe dispozitivele mobile. Achizițiile directe de produse și servicii ar putea fi acum făcute de clienți folosind smartphone-urile lor sau alte dispozitive asemenea.

Cu un număr estimat de 2,14 miliarde de persoane care fac cumpărături online la nivel global în 2021, comerțul electronic s-a impus acum ferm în viața noastră de zi cu zi. Comerțul electronic a devenit mai vital ca niciodată ca urmare a epidemiei care a forțat multe firme să-și închidă locațiile fizice și a făcut posibil ca multe altele să își continue activitatea în mediul online.

Pe scurt, comerțul electronic a avansat semnificativ de la începuturile sale nefavorabile din anii 1960. Dezvoltarea tehnologiei și creșterea internetului au dat naștere unei afaceri în plină

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

expansiune care se extinde anual. Comerțul electronic este aici pentru a rămâne și va continua să influențeze modul în care achiziționăm bunuri și servicii în anii următori, în ciuda faptului că mai sunt dificultăți de depășit.

2.4) Alte firme de catering

Pizza Hut

Unul dintre cele mai recunoscute lanțuri de pizza din lume este Pizza Hut. Dan și Frank Carney, doi studenți la Universitatea de Stat din Wichita, au lansat afacerea în 1958. De atunci, Pizza Hut a crescut la peste 18.000 de magazine în peste 100 de țări.

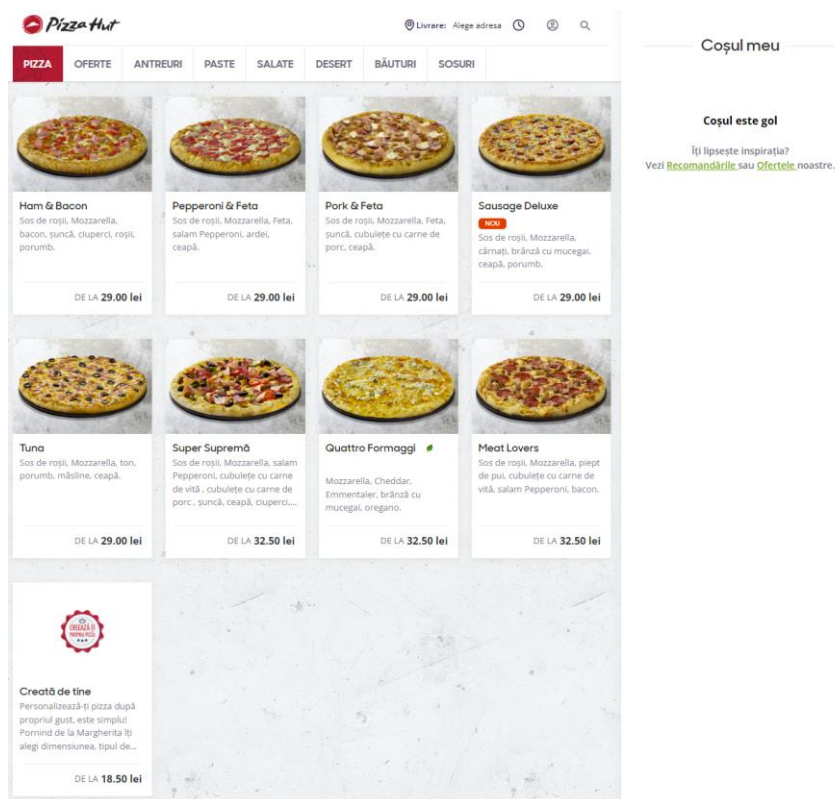
The screenshot displays the Pizza Hut website interface. At the top, there's a navigation bar with the Pizza Hut logo and links for 'Livrare: Alege adresa', 'PIZZA', 'OFERTE', 'ANTREURI', 'PASTE', 'SALATE', 'DESSERT', 'BĂUTURI', and 'SOSURI'. A large promotional banner for 'AI PIZZA MARE LA PREȚ DE PIZZA MEDIE' is featured, accompanied by an image of a pepperoni pizza. Below this, the 'OFERTE' section presents several deals: 'AI PIZZA MARE LA PREȚ DE PIZZA MEDIE' (repeated), 'FLATBREAD' (pizza crocantă, pe lung, de la 25 Lei), 'GOOD DEALS DE LA 21⁰⁰ Lei' (for families), and 'LUNEA DEVINE UN PIC MAI BUNA' (1+1 cadou pizza medie). Each offer includes a brief description and an 'ADAUGĂ' button. On the right side, a 'Coșul meu' (My Cart) section shows 'Coșul este gol' (Cart is empty) and suggests checking 'Recomandările' or 'Ofertele noastre'. At the bottom right, a Windows watermark reads 'Activate Windows Go to Settings to activate Windows.'

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Brandul Pizza Hut este binecunoscut pentru selecția sa extinsă de pizza, care variază de la opțiuni tradiționale, cum ar fi pepperoni și cârnați, până la selecții mai neobișnuite, cum ar fi pizza cu vită și porc. Sunt disponibile, de asemenea, o tot felul de garnituri, salate și deserturi. Pizza Hut se mândrește cu utilizarea ingredientelor proaspete de top în toate creațiile lor. Clienții pot selecta crusta, sosul și toppingurile pe care le doresc pentru fiecare pizza personalizată. Franciza este renumită pentru preparatele sale inventive de pizza, cum ar fi pizza cu crustă umplută și pizza Cheesy Bites care are cuburi mici de brânză coapte chiar în aluat. Pizza Hut a lucrat pentru a-și spori inițiativele de mediu în ultimii ani. Gigantul lanț de restaurante a promis că va folosi proceduri mai ecologice pe parcursul operațiunilor sale, va procura ingrediente mai durabile și va reduce amprenta de carbon.

În general, Pizza Hut rămâne un brand de pizza respectat și de succes care oferă pizza și garnituri gustoase, capacități de livrare rapidă și de comandă online, precum și dedicare pentru practicile durabile.



Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

KFC

KFC sau Kentucky Fried Chicken este un lanț de restaurante fast-food cunoscut pentru puiul său prăjit. Colonelul Harland Sanders a început afacerea în 1952 în Louisville, Kentucky, iar de atunci a devenit una dintre cele mai mari francize de fast-food din lume. Rețeta secretă pentru puiul prăjit KFC este una dintre caracteristicile sale distinctive. 11 ierburii și condimente sunt combinate în rețeta care este încă păstrată secret fiind unul dintre cele mai bine pazite secrete comerciale din industria culinară.

Brandul KFC este prezent în peste 140 de țări și are peste 23.000 de magazine. Cu schema sa de culori roșu și alb și imaginea binecunoscută a colonelului Sanders purtând un costum alb și o cravată neagră, corporația și-a stabilit un sentiment puternic de identificare a mărcii. KFC a lucrat pentru a-și diversifica meniul în ultimii ani pentru a include opțiuni mai sănătoase. Compania a redus conținutul de sare în câteva dintre elementele actuale ale meniului și a adăugat produse precum pui la grătar și salate.

The screenshot displays the KFC Romania website. At the top is a red navigation bar with the KFC logo and links for MENU, DELIVERY, RESTAURANTE, DESPRE NOI, MEDIA, and CARIERE. Below this is a large banner featuring a man in a red shirt and grey apron, with the text "Îți place să gătești pui?" and "Noi te plătim pentru asta." and a button "APLICĂ ACUM".

Below the banner are three promotional tiles:

- BAGĂ-TE-N FAȚĂ**: "Fără să vezi fete-fete", "CLICK & COLLECT", "COMANDĂ PRIN CLICK & COLLECT", "VEZI DETALII".
- Doar la KFC Drive**: "2x MENIU ZINGER/FILLET MOZZARELLA BURGER", "68.00 lei", "62.00 lei", "AFLĂ MAI MULTE".
- % OFERTE**: "#PEBUNE", "REVENDICĂ-ȚI OFERTA PREFERATĂ!", "VEZI DETALII".

The footer contains social media links (Facebook, YouTube, Instagram, TikTok), app download buttons (App Store, Google Play), and a link to "SPUNE-NE CUM TE-AI SIMȚIT LA NOI" with a "COMPLETEAZĂ CHESTIONARUL" button. At the very bottom are links for CONTACT, ABONARE NEWSLETTER, COLABORARE, INFORMAȚII PRODUSE, TERMENI ȘI CONDIȚII, POLITICA DE CONFIDENȚIALITATE ȘI POLITICA PRIVIND PRELUCRAREA DATELOR, DECLARAȚIA COOKIE, and REGULAMENT KFC POP A GUESS.

3) Prezentarea tehnologiilor

3.1) Front-end

3.1.1) JavaScript

JavaScript este un limbaj de programare care stă la baza tehnologiilor de tip World Wide Web (www), alături de HTML și CSS, urmând ca despre acestea din urmă să detaliez în subcapitolele următoare. Se estimează că din anul 2022 în jur de 98% din site-urile web au început să utilizeze JavaScript ca prim limbaj de programare pentru partea de front-end sau cunoscut și sub forma denumirii de client. Toate browserele cunoscute de către publicul larg au încorporat un motor special pentru interpretarea limbajului JavaScript pe dispozitivele consumatorilor.

JavaScript sau cunoscut și după abrevierea de JS este un limbaj interpretat de nivel înalt care este conform standardului ECMAScript. Chiar dacă acest limbaj are anumite caracteristici ale programării orientată obiect, nu este considerat un limbaj orientat obiect. Acest lucru se datorează faptului că nu respectă în totalitate paradigma programării orientată obiect. Spre deosebire de Java și C# care sunt special orientate obiect, JavaScript poate fi folosit și pentru programare procedurală, asemenea limbajului C++. De asemenea, JS are o diferență majoră față de limbajele menționate anterior, prin faptul că nu impune tipul variabilelor, putând astfel să se modifice structura și tipul de dată a unei variabile în timpul rulării programului. Acest lucru aduce atât beneficii cât și dezavantaje, fiind un limbaj mai maleabil, dar mai greu de depanat atunci când apar erori.

Deși JavaScript este folosit în principal pentru aplicații de tip client, acesta poate fi folosit și pentru aplicații de tip Back-end sau server, cu ajutorul sistemului Node.js.

3.1.2) HTML

HTML sau HyperText Markup Language nu este un limbaj de programare, ci un limbaj de marcare pentru documente care sunt menite pentru afișarea în browser. Browserele web interpretează documentele de tip HTML și apoi le afișează, împreună cu documentele de tip CSS și JS, dacă acestea există și sunt folosite în interiorul documentelor HTML. În această combinație,

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

fiecare limbaj are rolul său, HTML fiind scheletul, CSS fiind aspectul clientului iar JS fiind modul în care funcționează acesta.

Elementele de tip HTML reprezintă baza paginilor web, fără aceste elemente fiind imposibilă alipirea componentelor de tip CSS și JavaScript la o interfață vizuală cu care să poată interacționa utilizatorii.

HTML a apărut pe 24 Noiembrie 1995 iar de atunci a primit multe îmbunătățiri semnificative, trecând prin mai multe versiuni, versiunea curentă fiind HTML 5, apărută pe 28 Octombrie 2014 și continuând să primească diverse îmbunătățiri.

3.1.3) CSS

CSS sau Cascading Style Sheets nu este un limbaj de programare, ci un limbaj pentru stilizarea conținutului unei pagini web. Acest limbaj este strâns legat de HTML, legându-se de tagurile specifice. Se poate lega de tagurile HTML prin mai multe tipuri de selectori: selectori de clasă, selectori de element și selectori de id. În același timp, se pot insera stilurile CSS direct în tagurile HTML sau pot fi importate dintr-un fișier extern.

CSS a fost construit atât pentru permiterea așezării conținutului în pagina web cât și pentru stilizarea conținutului. Datorită acestor specificații, limbajul CSS ajută la îmbunătățirea vizibilității și accesibilității conținutului unei pagini încărcate în browser.

Termenul de “Cascading” se referă la modul în care funcționează prioritatea aplicării stilurilor în pagină. Acest lucru este important deoarece un programator fără experiență poate aplica din greșeală același stil pentru mai multe taguri din conținutul unui website fără să dorească acest lucru.

3.1.4) TypeScript

Microsoft a creat TypeScript, un limbaj de programare de nivel înalt open source, cu scopul de a repara problemele des întâmpinate în JavaScript. Acest limbaj este un superset de JavaScript în care se pot impune tipul variabilelor, acesta neputând fi schimbat în timpul rulării programului. La momentul rulării programului, mai întâi sunt verificate regulile impuse de TypeScript, iar apoi interpretatorul convertește codul în JavaScript.

3.1.5) Angular framework

Angular este un framework open-source bazat pe TypeScript care este susținut și întreținut de către Google, împreună cu o comunitate largă de oameni și companii. Acest framework este cunoscut și sub numele de Angular 2+ și a fost dezvoltat de către o echipă de la Google, aceeași echipă care a lucrat și la AngularJS. Această unealtă este utilizată pentru crearea de pagini web, permițând programatorilor să dezvolte aplicații într-un mod mai rapid și mai curat, având multe funcționalități deja implementate, fiind nevoie doar de adaptarea acestora în funcție de nevoile aplicației.

Diferențele dintre Angular și AngularJS

Deși au aproape același nume și ambele framework-uri sunt folosite pentru scrierea mai rapidă a aplicațiilor, Angular și AngularJS sunt două framework-uri scrise total diferit. AngularJS a apărut primul, pe data de 20 octombrie 2010, și a rămas fără suport tehnic din ianuarie 2022, iar Angular a apărut pe data de 14 septembrie 2016 și continuă să aibă suport tehnic până în prezent, acesta din urmă fiind și cel care continuă să fie utilizat de către dezvoltatorii aplicațiilor web. De asemenea, pe lângă diferențele semnificative de arhitectură, AngularJS este bazat pe JavaScript, de unde și prefixul JS, pe când Angular este construit pe baza limbajului TypeScript.

În prezent, Angular a ajuns la versiunea a 15-a, pe data de 16 noiembrie 2022.

3.2) Back-end

3.2.1) Java

Java este un limbaj de programare de nivel înalt, orientat obiect care a fost proiectat având cât mai puține dependențe. Este un limbaj de programare de uz general care funcționează după conceptul “scrie o dată, ruleaza oriunde” (“write once, run anywhere”), ceea ce înseamnă că o dată compilat codul Java, acesta poate rula pe orice mașină care suportă platforma menționată, fără a mai trebui recompilat codul. Aplicațiile Java sunt compilate în bytecode care poate rula pe orice sistem care deține o mașină virtuală Java (Java virtual machine – JVM), indiferent de arhitectura sistemului. Asemenea limbajului C#, sintaxa Java seamănă mult cu C/C++, având mai puține facilități low-level decât acestea, C/C++ având oferind o mare flexibilitate când vine vorba de alocarea memoriei, pe când C# și Java folosesc un “garbage collector” care se ocupă cu dezallocarea memoriei atunci când se pierde referința unei zone de memorie.

Utilizari

Limbajul Java a fost folosit pentru foarte multe tipuri de aplicații de-a lungul timpului, în prezent fiind predominant folosit pentru scrierea aplicațiilor pentru Android și pentru aplicațiile de tip back-end.

Istoric

Java a fost creat pe data de 19 mai 1995 de către James Gosling, un programator al companiei Sun Microsystems. Atât librăriile, cât și JVM și compilatorul limbajului Java au fost inițial construite de către Sun Microsystems. În prezent, Java 20 este ultima versiune apărută pe data de 21 martie 2023, Java 17 fiind ultima versiune stabilă ce aduce schimbări semnificative limbajului, apărută pe data de 14 septembrie 2021.

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

3.2.2) Spring

Spring este un framework și un container pentru “inversion of control” pentru platforma Java. Principalele caracteristici ale Spring framework pot fi folosite de către orice aplicație de tip Java, existând și anumite extensii ale acestui framework pentru scrierea de aplicații web.

Una dintre cele mai importante funcționalități Spring este “inversion of control container”, care oferă o modalitate foarte eficientă a configurării și coordonării obiectelor Java. Containerul este responsabil de gestionarea obiectelor, mai exact de crearea, utilizarea și configurarea acestora, lenagadu-le împreună. În trecut, se configurau aceste obiecte denumite “beans” prin fișiere de tip XML (Extensible Markup Language), dar în prezent se folosesc adnotări.

Spring a apărut în anul 2003, prima versiune fiind 0.9. În prezent ultima versiune este 6.0, apărută pe data de 16 noiembrie 2022.

3.2.3) Spring Boot

Spring Boot framework este o extensie a framework-ului Spring, aducând îmbunătățiri semnificative atunci când vine vorba de complexitatea necesară pentru configurarea unei aplicații de tip Spring. Acesta este folosit în mod special pentru crearea aplicațiilor de tip web precum: MVC, RESTful API sau microservicii. Există 3 capabilități importante prin care Spring Boot ușurează munca programatorului:

Autoconfigurare

Autoconfigurarea se referă la faptul că aplicațiile de tip Spring Boot vin cu dependente deja inițializate pe care dezvoltatorul software nu mai trebuie să le configureze manual. Acest fapt ajută nu doar la grabirea procesului de dezvoltare a aplicațiilor, ci și la aplicarea bunelor practici care duc la construirea unui cod mai ușor de citit și de întreținut. De asemenea, aceste setări implicite pot fi schimbate la nevoie.

Abordarea de opinie (“Opinionated approach”)

Pe baza cerințelor proiectului, Spring Boot adoptă o abordare de opinie pentru adăugarea și setarea dependentelor de pornire a aplicației. În loc ca programatorul să ia toate deciziile și să

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

configureze tot proiectul in mod manual, Spring Boot selecteaza pachetele care trebuiesc instalate si de asemenea, ce valori implicite sa foloseasca.

In timpul procesului de initializare, dezvoltatorii software pot selecta toate dependentele de care are nevoie proiectul utilizand platforma Spring Boot Initializr. Aceasta platforma este una simpla, care nu are nevoie de cod si creaza un proiect de tip Spring Boot care vine impreuna cu toate configurarile necesare selectate de catre programator.

Aplicatii de sine statatoare (“Standalone applications”)

Spring Boot le permite dezvoltatorilor software sa creeze aplicatii gata de a fi utilizate. Prin integrarea unui server web precum Tomcat sau Netty in aplicatie, aceasta poate functiona in mod independent, nefiind nevoie de server extern. Totodată, daca este necesar, programatorii pot opta pentru excluderea serverului web implicit.

3.2.4) PostgreSQL

PostgreSQL este un sistem de gestionare de baza de date care se axeaza pe conformarea SQL si pe extensibilitate. PostgreSQL ofera functionalitati precum: trigger, view, chei externe si proceduri stocate. De asemenea, de la statii de lucru, la depozite de date sau servicii web, PostgreSQL este conceput pentru a gestiona o varietate larga de sarcini.

3.3) Alte tehnologii folosite in domeniul aplicatiilor de catering/E-commerce

3.3.1) React

React, sau React.js este un framework pentru crearea aplicatiilor de tip website. Acesta are la baza limbajul JavaScript si este construit pe componente. Asemenea Angular, principalul scop al framework-ului React este de a ajuta dezvoltatorii de aplicatii web in dezvoltarea acestora intr-

un mod mai rapid, mai ușor de citit și mai ușor de depanat. Acesta este competitorul principal al framework-ului Angular, fiind cel mai utilizat framework până în prezent.

3.3.2) Aplicații pentru telefon

În ciuda faptului că foarte multe aplicații de E-commerce sunt construite sub forma unui website, există tot atâtea soluții software care în loc de a folosi un website pentru interfața cu care interacționează utilizatorul, folosesc o aplicație nativă de telefon, înlocuind doar partea de “front-end”, putând astfel să păstreze același “back-end”. De asemenea, există și opțiunea de a utiliza ambele opțiuni în paralel deoarece cuplarea de la client la server este una detasată, putând utiliza două aplicații de tip client care comunică cu același back-end.

3.3.3) ASP.NET Core

Microsoft a creat ASP.NET Core, un framework open-source care înlocuiește ASP.NET. Este un framework modular care rulează atât pe .NET Framework pentru Windows, cât și pe arhitectura multiplatformă .NET. De la versiunea ASP.NET Core 3, framework-ul nu mai suportă .NET Framework, continuându-se suportul doar pentru .NET Core.

Chiar dacă acest framework este rescris cu totul de la fundatie, deoarece este proiectat pentru a îmbina arhitecturile vechi ASP.NET Web API și ASP.NET MVC, prezintă un nivel ridicat de compatibilitate cu ASP.NET.

ASP.NET Core a apărut pe 27 mai 2016, Microsoft continuând să îmbunătățească această tehnologie, în prezent ajungând la versiunea 7.0, apărută pe data de 8 noiembrie 2022.

4) Prezentarea soluției informatice

4.1) Descrierea generală a sistemului informatic

Sistemul este reprezentat de o aplicație E-commerce de tip REST Api – client cu specific de catering. Arhitectura aleasă este crossplatform ceea ce înseamnă că indiferent de sistemul de

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

operare sau de dispozitivul de pe care este accesat site-ul, acesta va rula deoarece orice website este scris într-un limbaj interpretat în browserul de pe care este accesat. Scopul acestei aplicații este de a prezenta produsele unei firme de catering locală într-o manieră cât mai simplă pentru utilizator astfel încât să faciliteze o navigare ușoară printre ofertele culinare și de asemenea să ușureze procesul de comandă a produselor dorite.

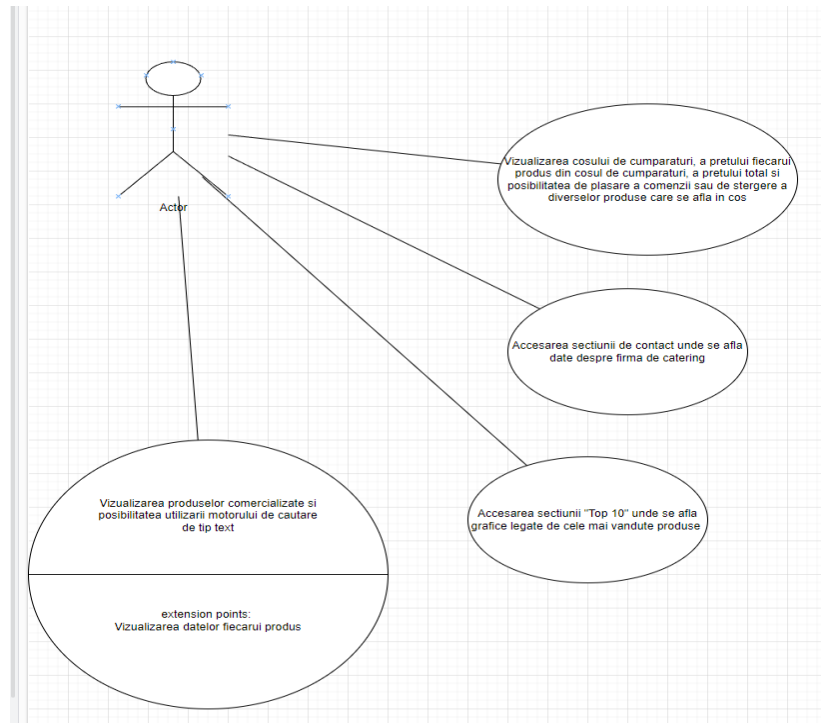
La deschiderea aplicației, utilizatorul este întâmpinat de pagina principală unde se află “Headerul” care rămâne vizibil indiferent de ceea ce accesează utilizatorul în pagină, acționând ca un meniu al aplicației. Opțiunea implicită este cea de prezentare a tuturor produselor împreună cu diverse categorii de produse și o bară de căutare de tip text unde utilizatorul poate introduce cuvinte cheie pentru a căuta un produs dorit. De asemenea, utilizatorul poate apăsa pe un produs pentru a-l vizualiza în mod detaliat și pentru a-l putea adăuga în cosul de cumpărături care ține minte produsele adăugate chiar și la închiderea website-ului, astfel clientul putând să-și continue cumpăraturile mai târziu dacă dorește. Pentru a comanda produsele, acesta trebuie să completeze un formular care conține câteva date personale și de asemenea, adresa la care dorește să ajungă comanda. Odată plasată comanda, utilizatorul primește un mail de confirmare.

Ultimele două funcționalități ale aplicației au rol de prezentare, acestea constând în secțiunea de “Contact” unde utilizatorul poate vizualiza date despre restaurantul care deține site-ul, și secțiunea “Top 10” unde poate vizualiza un grafic cu top 10 cele mai vandute produse.

4.2) Specificarea cerințelor sistemului informatic

Diagrama cazurilor de utilizare

În diagrama prezentată sunt expuse, în mod grafic, actorul (consumatorul), principalele funcționalități ale aplicației care vor fi implementate în decursul etapei de proiectare și relațiile dintre aceștia. În cadrul aplicației “Pizza4you” se află un actor reprezentat de consumatorul final.



Putem observa din diagrama faptul ca utilizatorul poate sa vizualizeze gama de produse comercializata de restaurantul “Pizza4you”, poate cauta produsele dorite prin motorul de cautare de tip text si de asemenea poate cauta produse si dupa categoriile acestora.

Din optiunea de vizualizare a cosului de cumparaturi, consumatorul poate sa stearga produse din cosul de cumparaturi, sa comande produsele respective si sa vada costul total al produselor comandate. De asemenea, utilizatorul poate accesa si paginile “Contact” si “Top 10” pentru a vizualiza informatii despre restaurant si totodata, pentru a vedea cele mai vandute produse ale restaurantului.

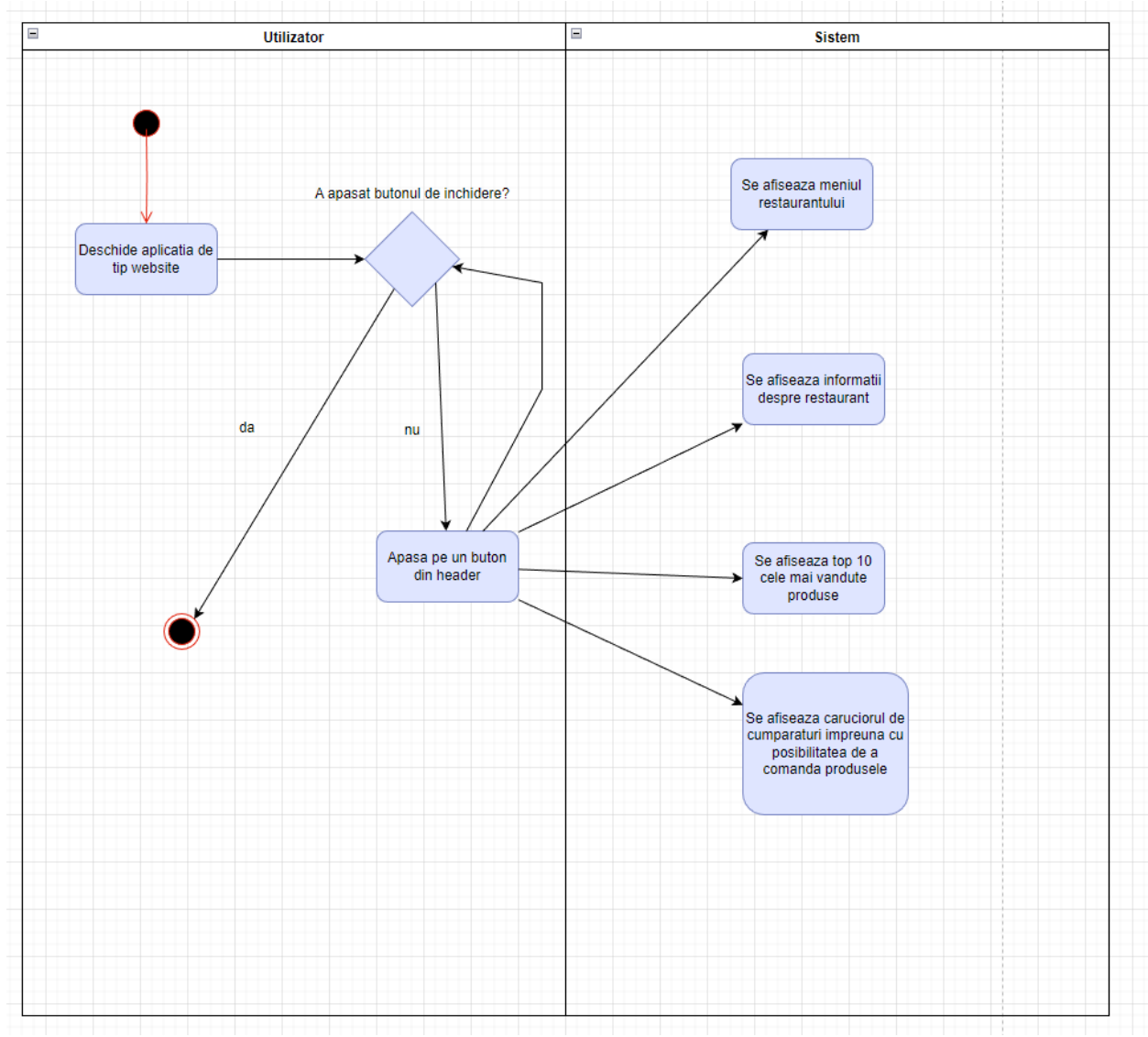
4.3) Analiza sistemului informatic

4.3.1) Diagrame de activitate

Diagramele de activitate sunt utile pentru că ne oferă o un mijloc vizual prin care putem înțelege mai bine funcționalitățile aplicației prezentate și cum putem naviga prin aceasta. Aceste diagrame sunt folosite în procesul de creare a unei aplicații deoarece prezintă un limbaj

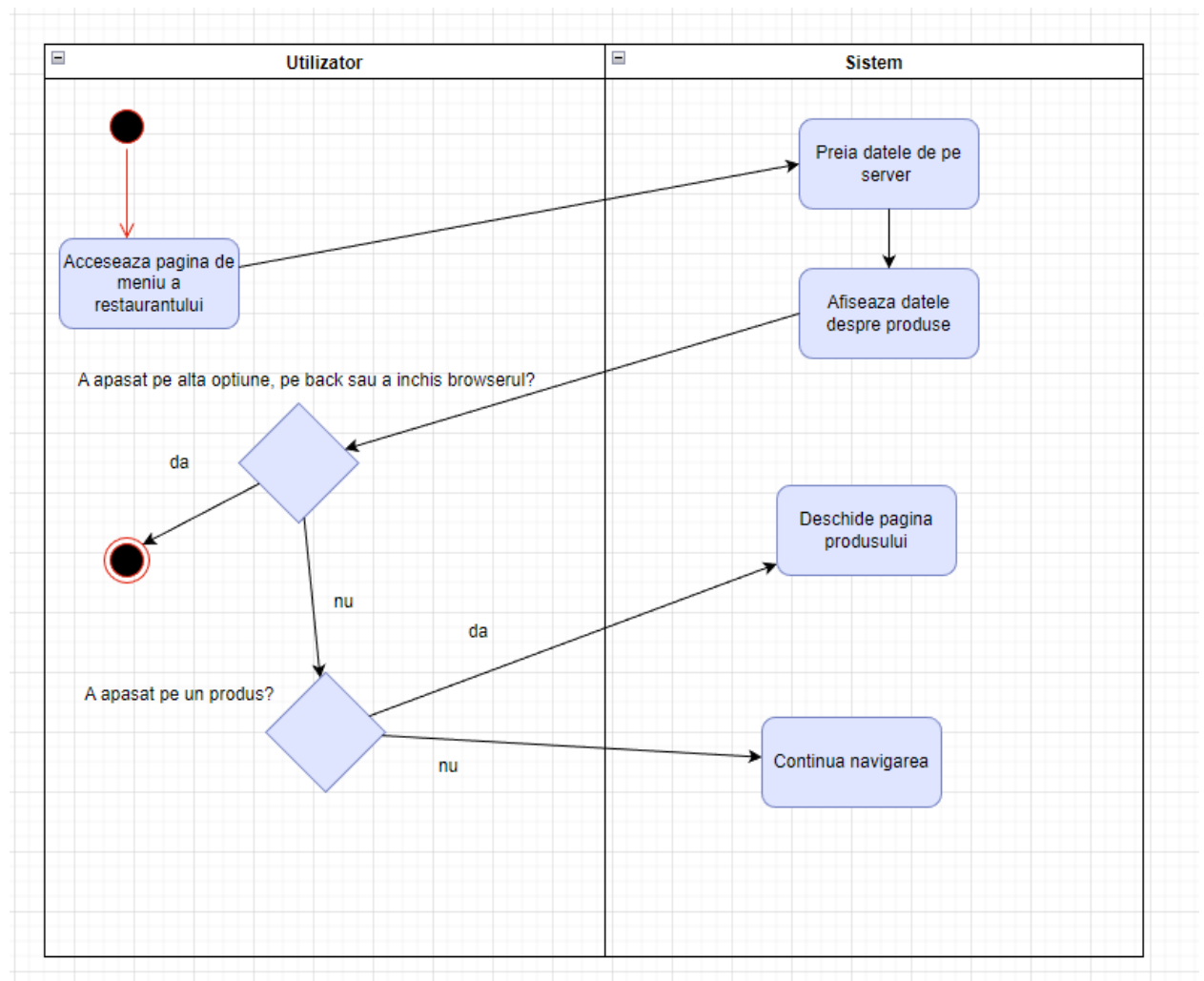
standardizat pe care-l poate înțelege orice programator, astfel evitând problemele de comunicare dintre echipele care ar putea lucra la un proiect mai mare și complex.

Diagrama de Activitate – “Header”



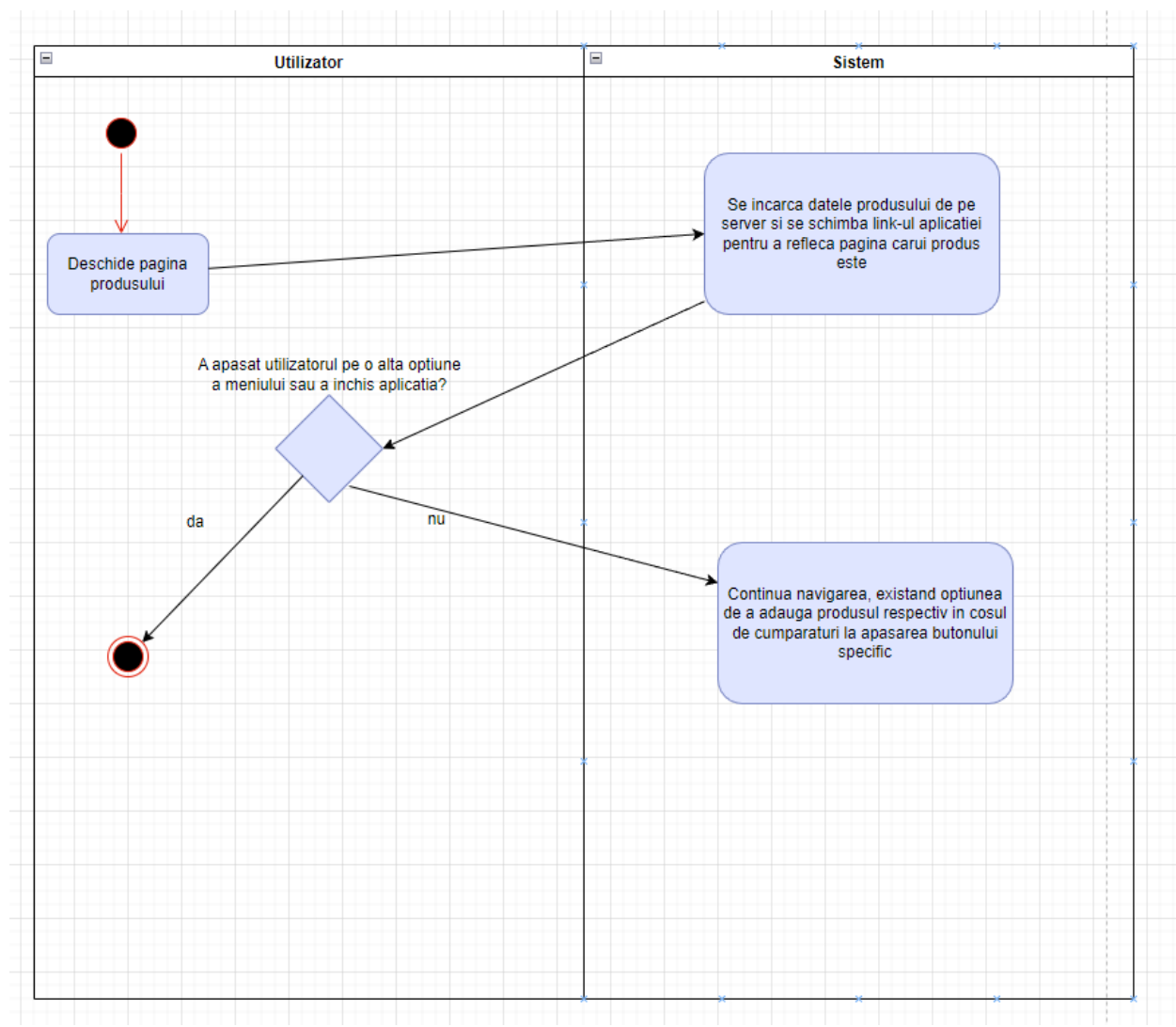
În diagrama de activitate prezentată ca fiind cea a “Headerului” se poate observa faptul că utilizatorul poate accesa setul de opțiuni așa cum dorește odată ce a deschis pagina website-ului. Acesta poate naviga prin paginile de contact, de “Top 10”, de cos de cumpărături sau de meniu al restaurantului.

Diagrama de activitate – Pagina cu produse



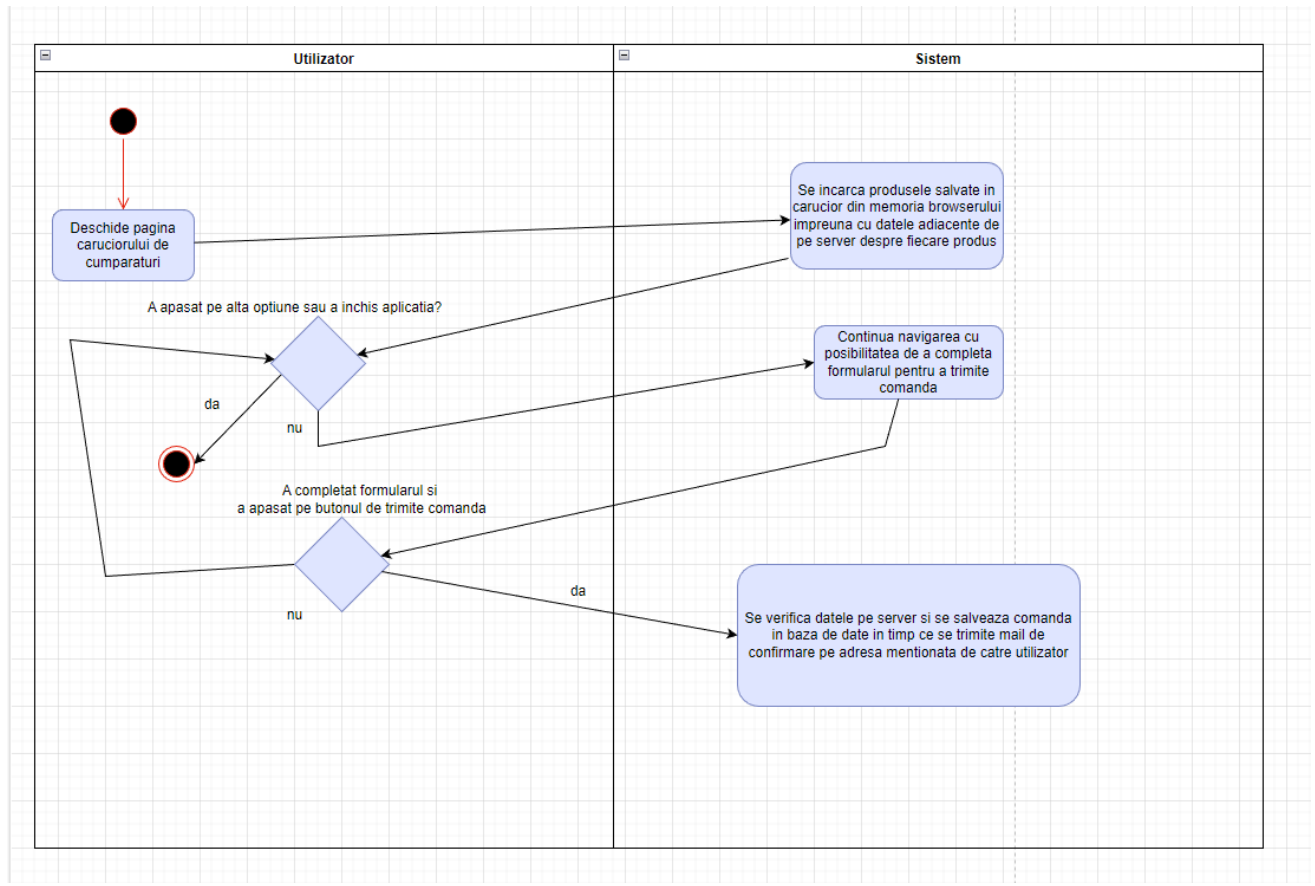
În această diagramă se poate observa faptul că toate datele sunt salvate pe un server, astfel orice utilizator, indiferent de moment, locație sau device-ul de pe care accesează site-ul va avea datele sincronizate, astfel neputând să comande ceva ce nu se află în meniul restaurantului. De asemenea, utilizatorul poate trece la alta pagină oricând dorește, permitând o experiență fluidă în utilizarea aplicației. Totodată, din această pagină se poate accesa și pagina individuală a fiecărui produs.

Diagrama de activitate – Pagina produsului



In figura de mai sus se poate observa faptul ca datele fiecarui produs sunt luate de pe server, aceste date continand poza, numele, ingredientele si asa mai departe. Se poate vedea faptul ca utilizatorul poate sa adauge produsul respectiv in cosul de cumparaturi pentru a-l comanda. O caracteristica importanta care trebuie mentionata este faptul ca flexibilitatea pe care o aveam datorita “Headerului” nu dispare, consumatorul putand oricand sa navigheze la alta pagina.

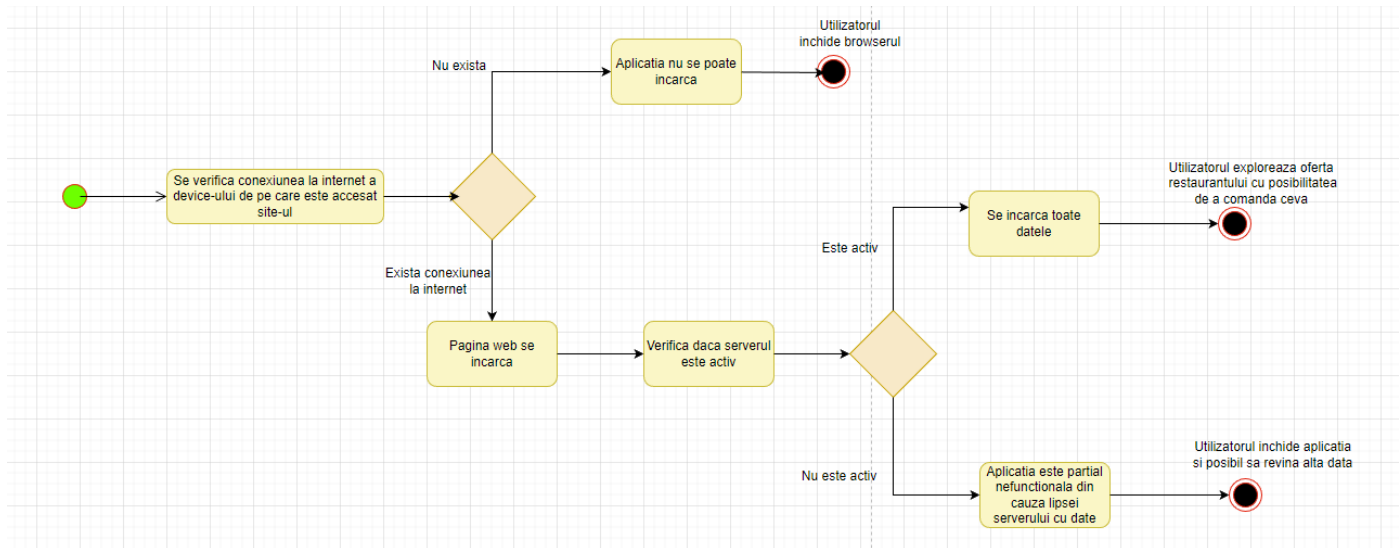
Diagrama de activitate – Caruciorul de cumparaturi



În poza de mai sus se poate remarca faptul că funcționalitatea caruciorului de cumparaturi este singura care interacționează cu memoria locală a browserului de pe care este accesat site-ul de către utilizator. În această memorie sunt salvate produsele pe care utilizatorul le-a adăugat la orice moment în cos, astfel chiar dacă consumatorul închide aplicația, la următoarea redeschidere acesta nu va pierde produsele pe care deja le salvase. De asemenea, se poate remarca faptul că utilizatorul poate completa un formular cu datele necesare pentru a comanda, iar la apăsarea de buton pentru trimiterea acesteia primește și un mail de confirmare pe adresa personală pe care a menționat-o în formular.

4.3.2) Diagrame de procese și colaborare în BPMN

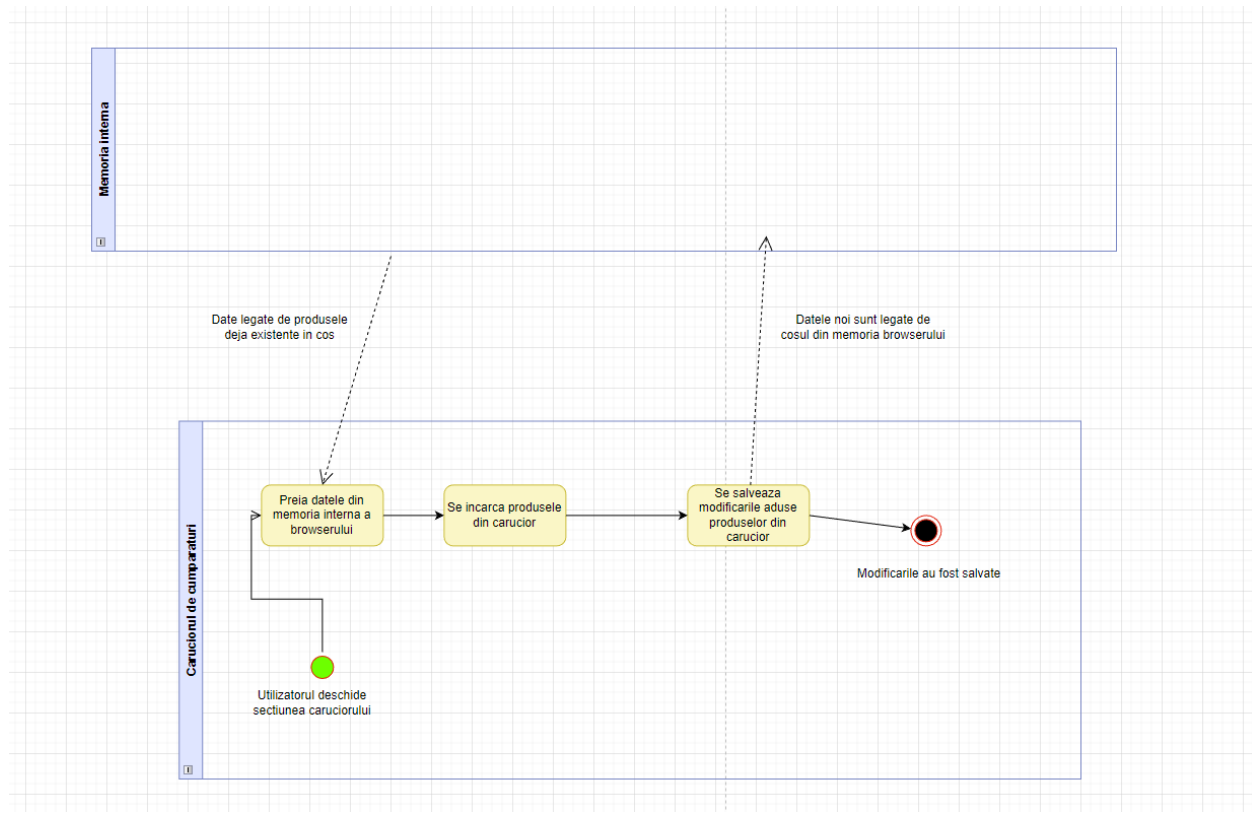
Diagrama de Procese - Deschiderea Interfetei cu Utilizatorul (website-ul)



În diagrama prezentată se observă faptul că pentru funcționarea corespunzătoare a aplicației, aceasta depinde de mai mulți factori. Primul factor este conexiunea la internet, utilizatorului neputând să acceseze website-ul fără aceasta. De asemenea, chiar dacă consumatorul are o conexiune stabilă la internet pe tot parcursul utilizării aplicației, aceasta nu va rula corespunzător dacă serverul nu este activ. Chiar dacă la prima observație arhitectura aplicației pare foarte complexă și fragilă, acestea sunt niște consecințe asumate imposibil de evitat atunci când este vorba de o aplicație distribuită care trebuie să deservească în mod egal un număr mare de utilizatori indiferent de unde se află aceștia.

De-a lungul timpului s-au inventat mai multe soluții pentru a minimiza problemele menționate mai sus, dar și aceste soluții au limitări, proprietarul site-ului web neputând să influențeze conexiunea la internet a utilizatorilor care vor să îi acceseze produsul software. Una din soluțiile care rezolvă în mare măsură problemele de downtime ale serverului este Docker & Kubernetes. Acestea sunt două produse software care pot ambala aplicația (serverul) într-un container și să-l multiplice ori de câte ori este nevoie astfel încât chiar dacă o instanță a serverului da greș și se închide, alta poate să îi ia locul.

Diagrama de Comunicare – Memoria Internă a Browserului și Cosul de Cumpărături



În poza de mai sus se poate vedea procesul de comunicare dintre interfata cu care interacționează utilizatorul și memoria internă a browserului pe care rulează website-ul. Atunci când consumatorul deschide pagina cosului de cumpărături, aplicația încarcă datele din memoria internă și afișează produsele, aducând de pe server și datele adiacente de care este nevoie. De asemenea, interacțiunea cu memoria internă apare la orice schimbare adusă cosului de cumpărături, indiferent dacă este vorba de adăugare sau ștergere de produse din acesta.

4.4) Proiectarea sistemului informatic

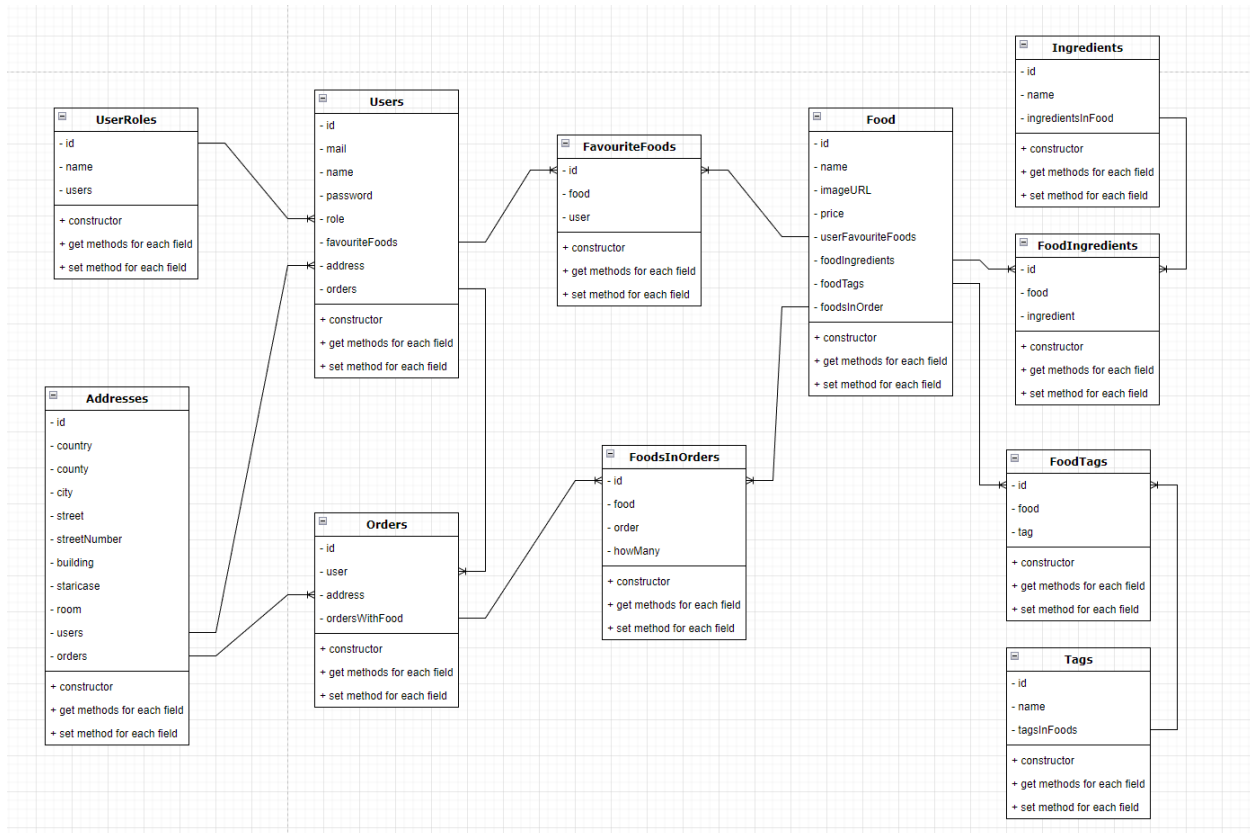
4.4.1) Diagrama de clase detaliată

În diagrama de mai jos putem vizualiza într-un mod foarte ușor și clar clasele principale ale aplicației "Pizza4you". Totodată putem observa întregul conținut al claselor împreună cu

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

legăturile dintre acestea. După cum se poate remarca, clasele sunt făcute în așa fel încât să fie cât mai ușor și rapid de lucrat cu acestea, având date cât mai puține și cu o redundanță cât mai mică. De asemenea, clasele respective conțin constructori și funcții de acces și de modificare a fiecărui atribut.



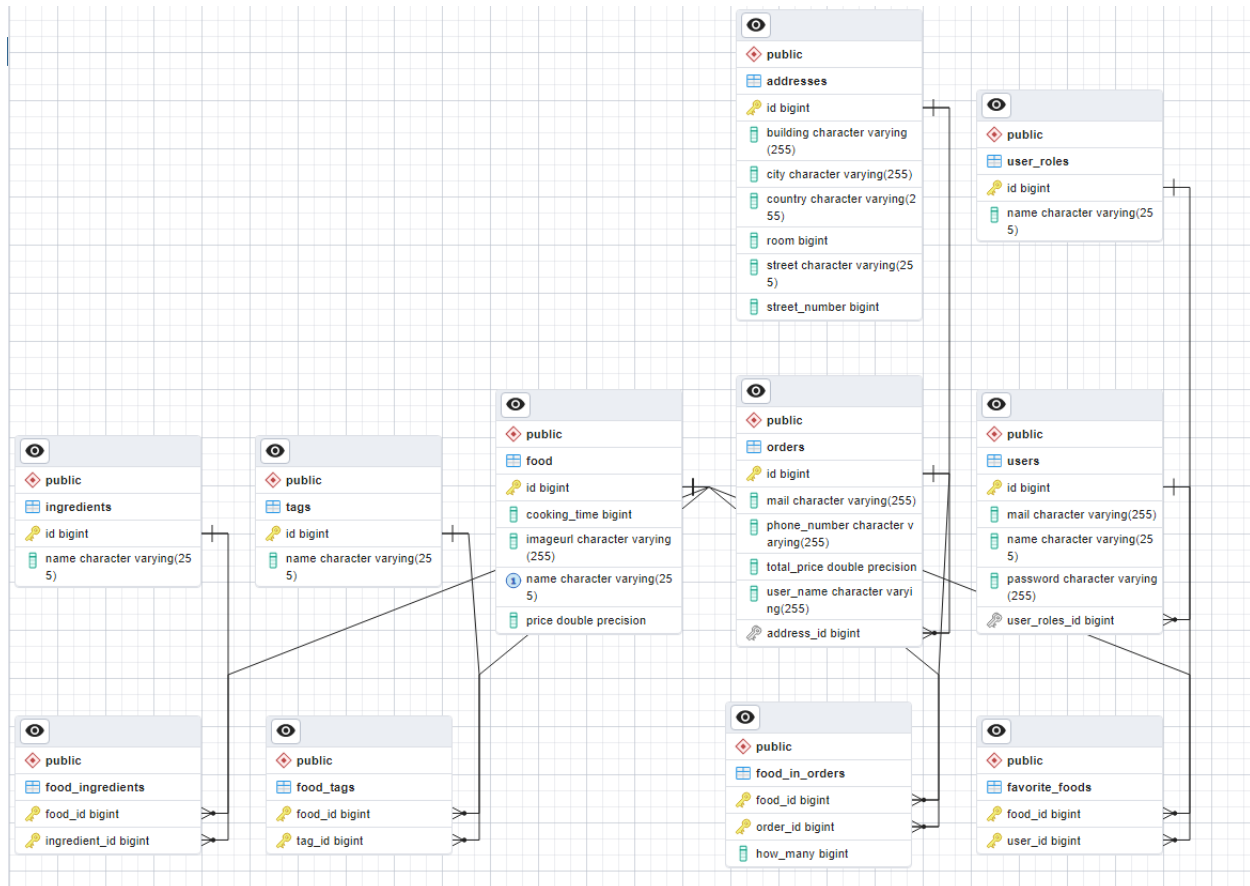
4.4.2) Proiectarea bazei de date

Diagrama bazei de date este o componentă foarte importantă pentru o înțelegere mai bună a modului de stocare a informației, afișând legăturile, datele și tipurile acestora într-un mod vizual, cât mai ușor de înțeles. Datorită acestor avantaje, acest tip de diagrame este foarte folosit la scară

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

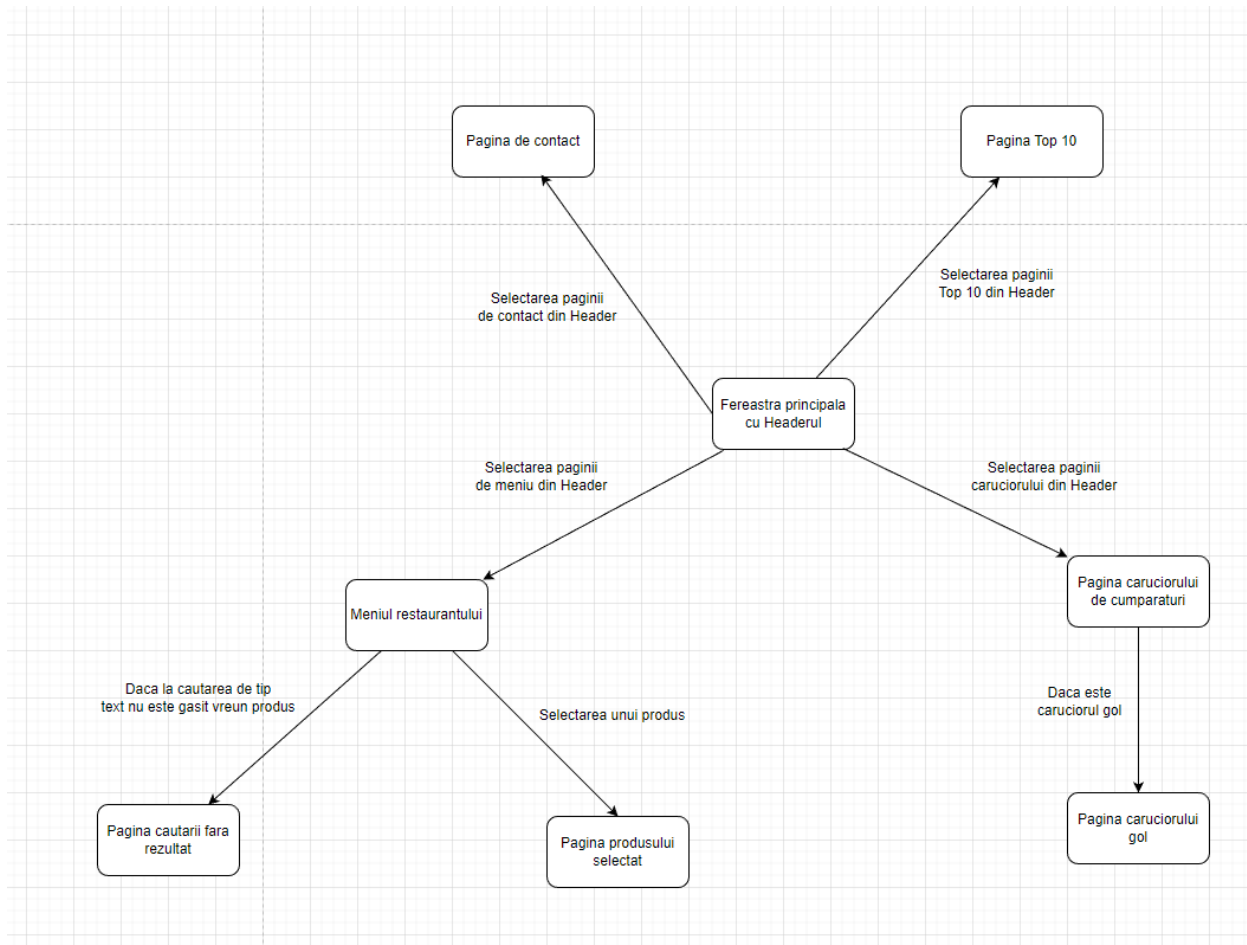
largă pentru crearea de aplicații complexe, ajutând programatorii să comunice și să lucreze mai eficient pentru crearea sau menținerea sistemelor software.



Datorită modului prin care este construită aplicația prezentată, diagrama de clase este un pas premergător în proiectarea bazei de date, clasele fiind corelate cu tabelele bazei de date create. După cum se poate observa mai sus, legaturile de “many to many” dintre tabele sunt create cu ajutorul unor tabele adiacente care conțin defapt legături de tip “one to many” de la tabela de lagatura la cele doua tabele. De asemenea, se pot observa tipurile de date din fiecare tabela.

4.4.3) Proiectarea interfetelor utilizator

Diagrama de mapare a interfetelor este utila pentru a ne ajuta la intelegerea mai buna a procesului de navigare din interiorul aplicatiei, astfel stiind cum putem ajunge dintr-un punct al aplicatiei in altul fara aparitia posibilelor confuzii.



În imaginea prezentată la acest punct observăm o mapă a interfețelor aplicației. Aici putem vizualiza ce interfețe sunt conectate și în ce mod. După cum putem vedea, odată ce deschidem aplicația suntem întâmpinați de pagina principală unde se află "Headerul" aplicației care funcționează ca un meniu, astfel putând comuta de la o fereastră la alta în funcție de opțiunea selectată din acesta. Un lucru important de menționat este faptul că deși nu se poate ajunge la unele ferestre prin "header" (ex: pagina unui produs anume), acesta nu dispăre niciodată, astfel putând merge la alta pagină direct din "header" chiar dacă nu există opțiunea în mod explicit în fereastră actuală.

4.5) Sistemul informatic implementat

Sistemul informatic implementat este reprezentat de două programe care comunică unul cu celălalt pentru a forma o aplicație de comerț pe internet îndreptată spre servirea de mâncăruri

pentru un restaurant local. Aplicatia are functionalitati asemanatoare cu alte website-uri de catering precum cele detinute de Pizza Hut sau Jerry's Pizza.

In continuare vor fi prezentate componentele din care este construita solutia software.

4.5.1) Componenta backend (RESTful Api)

Aceasta parte este “creierul” solutiei informatice, toate datele fiind procesate prin intermediul codului Java si a framework-ului Spring Boot din care este compus serverul si de asemenea stocate in baza de date PostgreSQL cu care interactioneaza backendul. Chiar daca severul a fost conceput pentru a rula cu website-ul aplicatiei, acesta poate oricand sa ruleze impreuna cu alte aplicatii in cazul in care se doreste schimbarea interfetei cu clientul sau adaugarea unor noi aplicatii pentru a imbunatati experienta utilizatorilor (ex: aplicatie de telefon care sa interactioneze cu serverul).

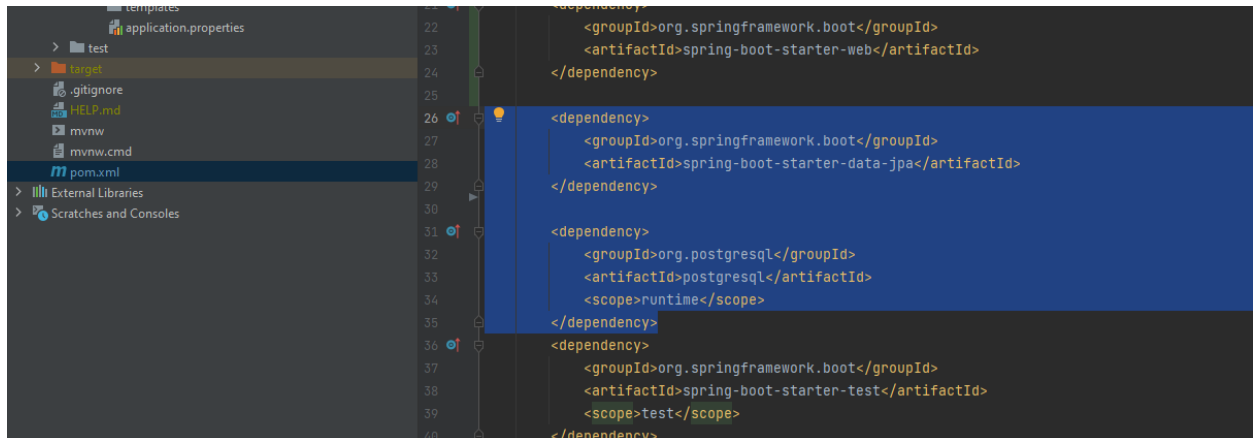
Baza de date

In trecut pentru fiecare aplicatie care necesita o baza de date era nevoie de o echipa de analisti care sa fie specializati in limbajul SQL pentru conceperea, construirea si mentinerea bazei de date. In zilele noastre nu mai este atat de necesar acest lucru datorita framework-urilor de mapare de date care ajuta programatorii din diferite limbaje sa creeze baze de date complexe direct din codul sursa al aplicatiei, fara a mai fi nevoie de o echipa specializata in SQL. Intr-adevar, este important ca programatorii, indiferent daca lucreaza cu Java, C# sau alte limbaje de programare folosite pentru backend, sa cunoasca elemente esentiale despre bazele de date de tip SQL si NoSQL si diferentele dintre acestea, astfel putand intelege cu adevarat functionalitatile bazei de date pe care o implementeaza.

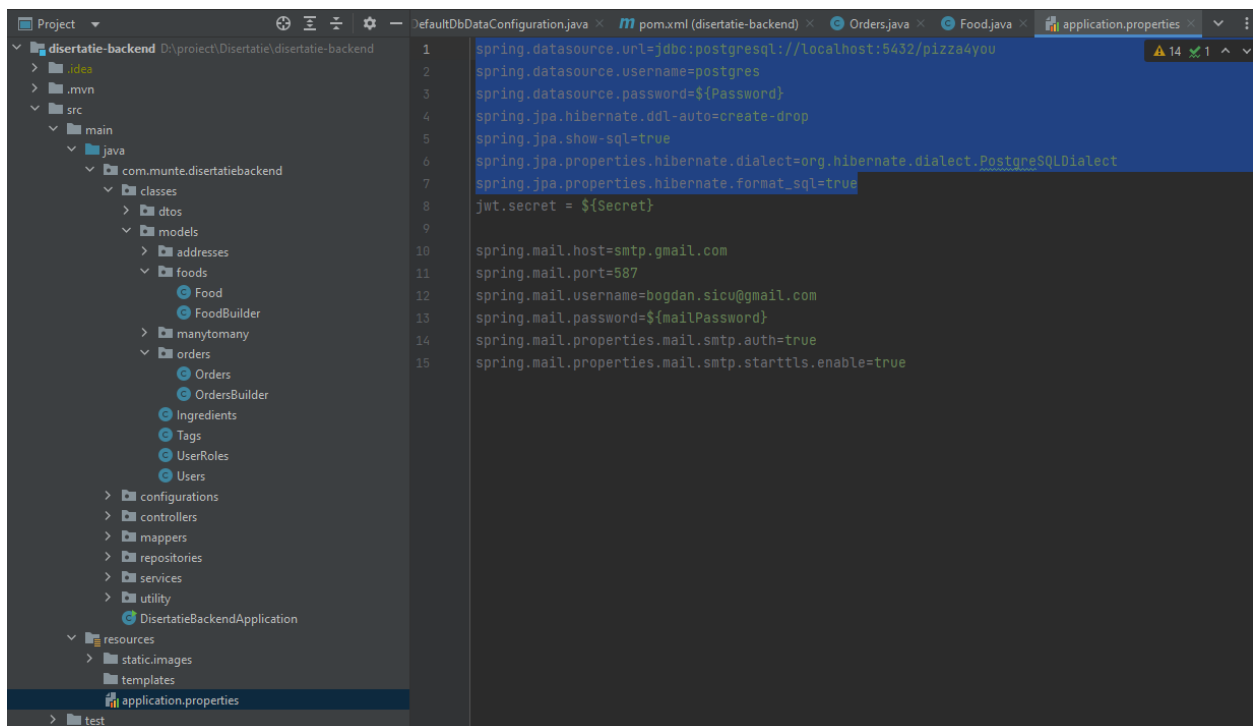
In aplicatia prezentata, maparea bazei de date se face cu ajutorul framework-urilor Hibernate & JPA. Aceste framework-uri trebuiesc specificate in fisierul “pom.xml” care contine toate dependentele proiectului fata de alte librarii si framework-uri. Pentru a putea conecta baza de date la aplicatia noastra trebuie sa specificam si framework-ul pentru tipul de bazei de date, indiferent daca este SQL sau NoSQL si indiferent de furnizorul acesteia.

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică



Important de mentionat este faptul ca pentru a putea crea o baza de date, trebuie sa existe deja instalat software-ul pentru baza de date dorita si de asemenea, trebuie mentionata conexiunea serverului de baze de date in fisierul “application.properties” de la nivelul folderului de resurse.



Maparea bazei de date

```
@Entity
@Table(name = "FOOD")
public class Food {

    2 usages
    @Id
    @Column(name = "id")
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    2 usages
    @Column(name = "name", unique = true)
    private String name;

    2 usages
    @Column(name = "imageUrl")
    private String imageUrl;

    2 usages
    @Column(name = "price")
    private Double price;

    2 usages
    @Column(name = "cooking_time")
    private Long cookTime;

    1 usage  ▲ Sicu Bogdan-Andrei
    public Long getCookTime() { return cookTime; }

    2 usages  ▲ Sicu Bogdan-Andrei
    public void setCookTime(Long timeToCook) { this.cookTime = timeToCook; }

    2 usages
    @JsonIgnore
    @OneToMany(mappedBy = "food")
    private Set<FavouriteFoods> userFavouriteFoods = new HashSet<>();

    2 usages
    @OneToMany(mappedBy = "food")
    private Set<FoodIngredients> foodIngredients = new HashSet<>();

    2 usages
    @OneToMany(mappedBy = "food")
```

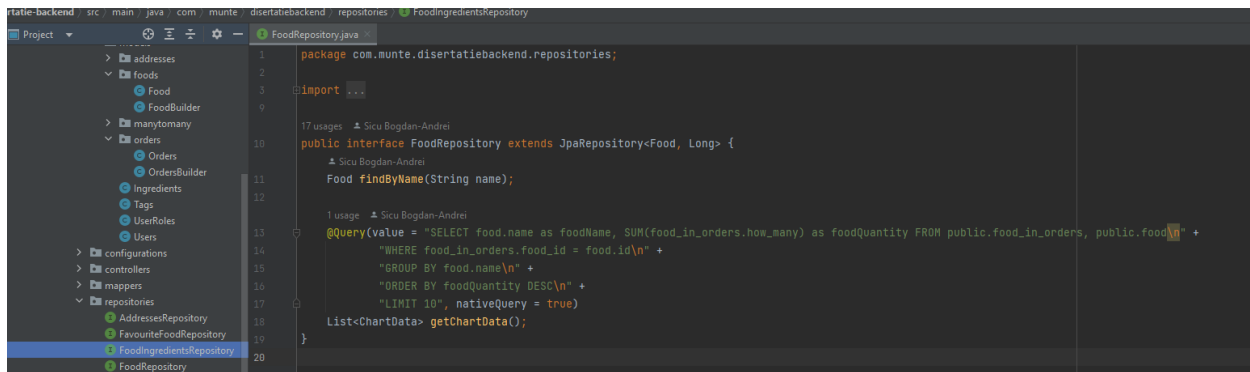
Asa cum am mentionat mai sus, in zilele noastre, in cele mai multe cazuri, baza de date este construita direct din codul sursa al aplicatiei, acest lucru fiind implementat usor diferit de la limbaj la limbaj. In cazul nostru, avand Java & Hibernate & JPA, exista adnotari speciale care specifica ce tip de clase din aplicatie pot fi considerate tabele si ce attribute din aceste clase pot fi considerate campuri in baza de date.

Se poate observa in poza alaturata faptul ca la inceputul clasei avem adnotarea “@Entity” care marcheaza clasa ca fiind o tabela pentru baza de date. Adnotarea “@Column”

desemneaza attributele clasei ca fiind coloana in tabela care trebuie sa fie creata, avand si alte adnotari complementare pentru a specifica daca un camp este id, autogenerat, daca este cheie externa si asa mai departe. De asemenea, se remarca faptul ca exista diverse valori care pot fi date in interiorul parantezelor de langa aceste cuvinte cheie pentru a specifica diverse optiuni semnificative precum: numele tabelului, numele coloanei si asa mai departe. Lipsa acestor optiuni duce la aplicarea optiunilor implicite (ex: coloana va fi denumita exact dupa denumirea atributului din clasa scrisa in Java).

Preluarea datelor din baza de date

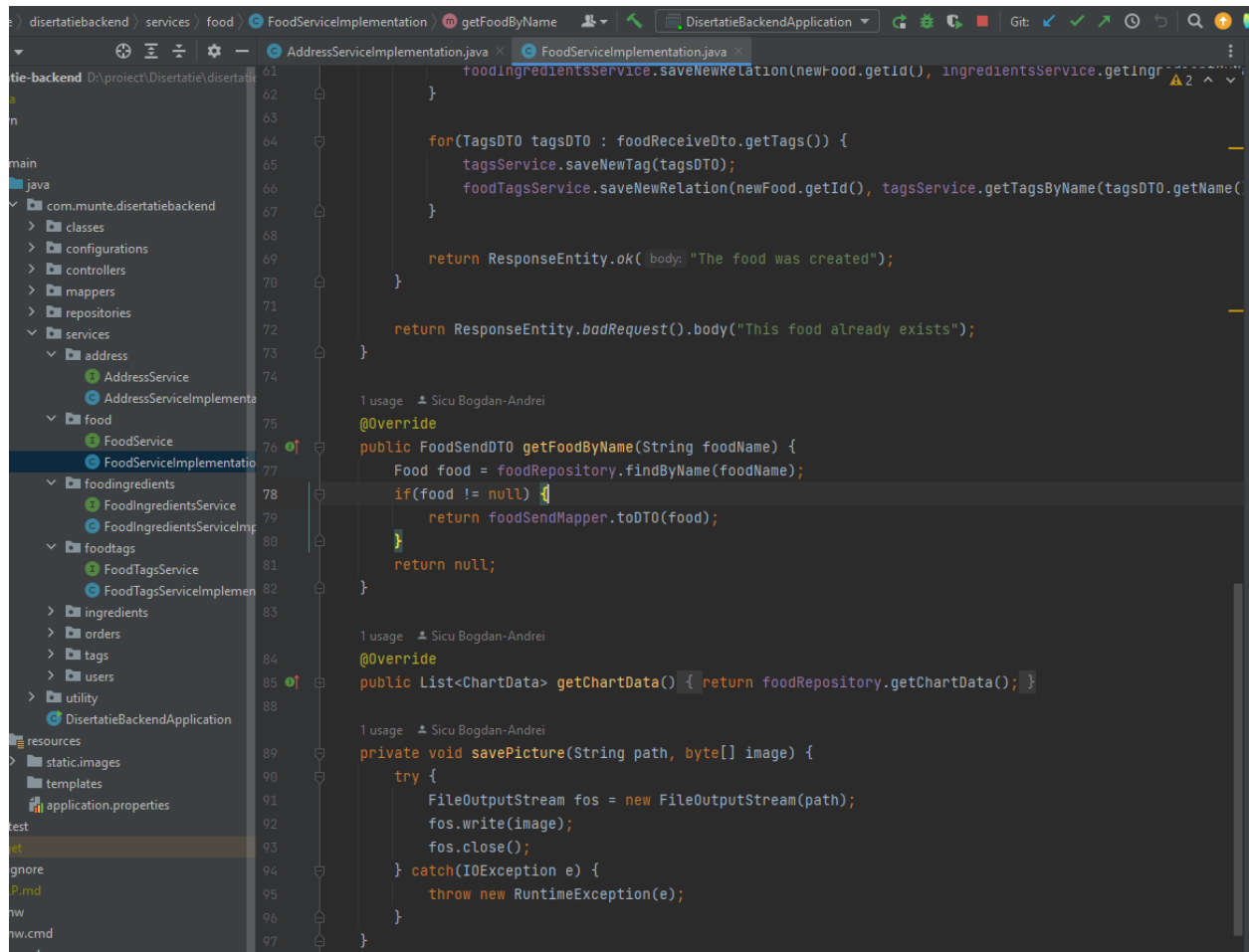
Pentru tranzactiile asupra bazei de date este necesara o clasa separata de tip “interface” care extinde functionalitatile JPA prin “JpaRepository<Clasa, Cheie_Primary>”. Chiar daca exista multe functionalitati generice de tip CRUD deja implementate, uneori este necesara o implementare mai avansata care poate fi scrisa direct in limbaj nativ SQL. Acest lucru poate fi observat in poza de mai jos.



Procesarea datelor

O buna practica a unui backend de tip REST este segmentarea acestuia in “Repository” (legatura cu baza de date) -> “Service” (procesarea datelor) -> “Controller” (preluarea de cereri de la aplicatiile de tip client). Acestea fiind mentionate, sectiunea prezenta se axeaza pe partea de “Service” care spre deosebire de partile de “Repository” si “Controller” care se ocupa de chestii foarte specifice, acesta este foarte volatil putand procesa datele printr-o multitudine de moduri.

Spre exemplu, in service-ul “FoodServiceImplmentation.java” se intampla lucruri destul de diverse fata de celelalte deoarece procesarea si adaugarea de mancaruri noi care au si imagini, astfel trebuind ca vectorul de bytes sa-l converteasca intr-un fisier si sa-l salveze in memoria masinii pe care ruleaza serverul.



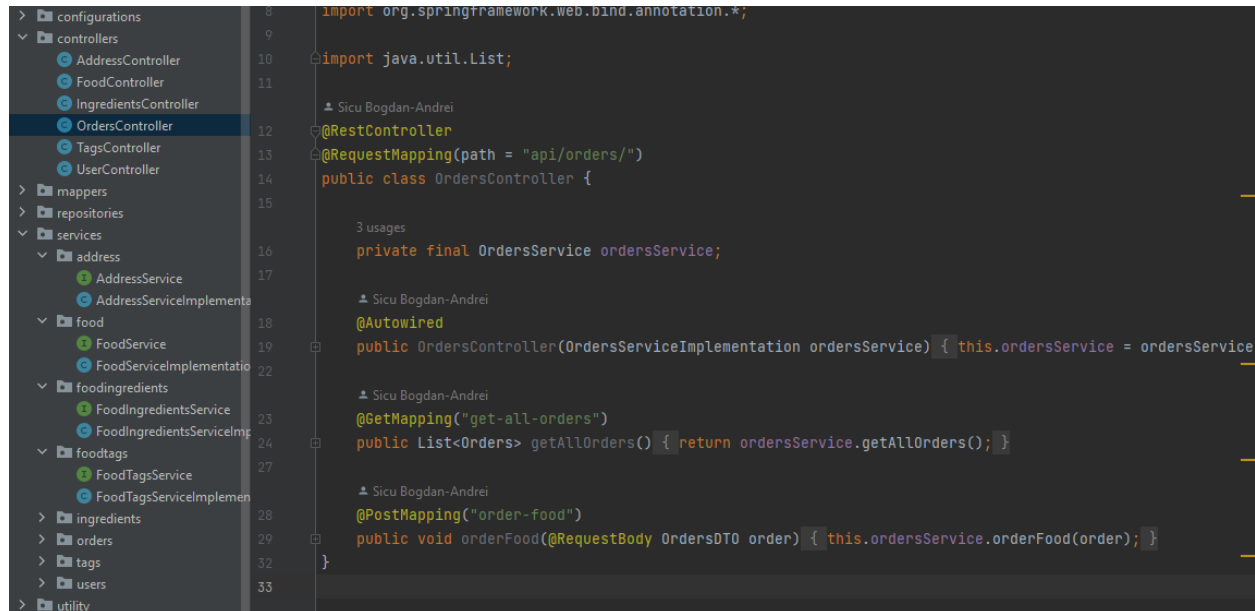
Primirea de cereri de la client

Asa cum a fost mentionat mai sus, exista o parte a serverului care se ocupa in mod special doar de preluarea cererilor de la aplicatiile de tip client si de trimiterea de raspuns atunci cand se termina de procesat datele.

Se remarca in poza de mai jos faptul ca pentru ca o componenta a aplicatiei sa se poata ocupa de primirea de cereri, trebuie sa fie desemnata ca fiind o clasa de tip “Controller”, acest lucru fiind facut prin adaugarea adnotarii “@RestController”. Aceasta este adesea folosita impreuna cu adnotarea “@RequestMapping()” in care se pot adauga diversi parametri precum cel de “path” care specifica adaugarea din link care este adaugata pentru a face o cerere catre clasa respectiva.

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică



```
import org.springframework.web.bind.annotation.*;
import java.util.List;

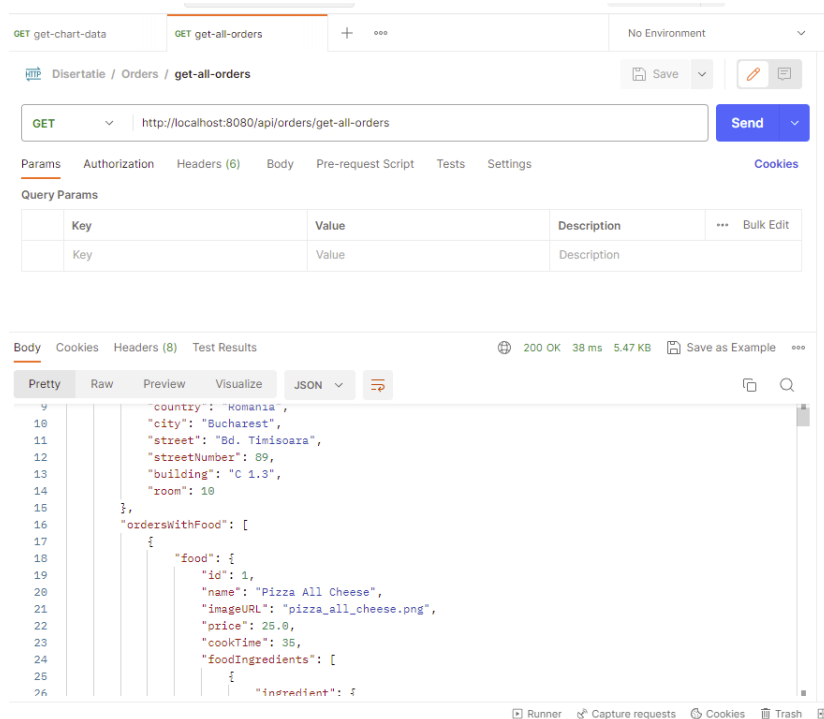
@RestController
@RequestMapping(path = "api/orders/")
public class OrdersController {

    3 usages
    private final OrdersService ordersService;

    @Autowired
    public OrdersController(OrdersServiceImplementation ordersService) { this.ordersService = ordersService; }

    @GetMapping("get-all-orders")
    public List<Orders> getAllOrders() { return ordersService.getAllOrders(); }

    @PostMapping("order-food")
    public void orderFood(@RequestBody OrdersDTO order) { this.ordersService.orderFood(order); }
}
```



Pentru ca o metoda sa fie considerata “request” sau cerere trebuie sa aiba adnotarea “@RequestMapping”, verbul fiind oricare dintre cele acceptate, cel mai des intalnite fiind: “GET” (ex: @GetMapping), “PUT”, “POST” si “DELETE”. De asemenea, pentru fiecare cerere se poate adauga o extensie a link-ului care trebuie accesat. Spre exemplu, pentru apelarea metodei “getAllOrders()”

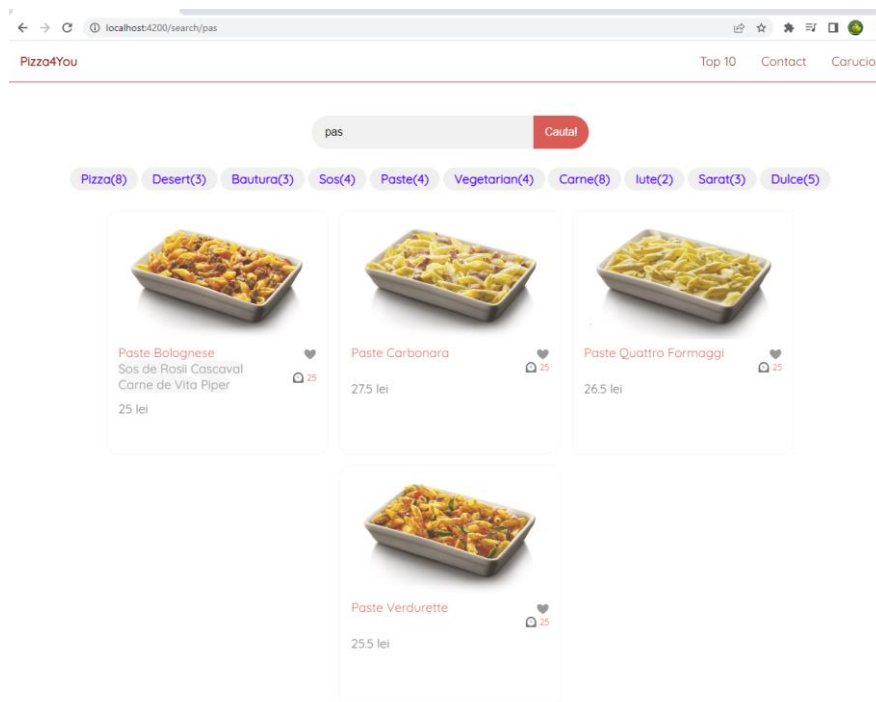
trebuie trimisa o cerere la [“http://localhost:8080/api/orders/get-all-orders”](http://localhost:8080/api/orders/get-all-orders) avand verbul “GET”, asa cum se poate vedea in poza adiacenta.

4.5.2) Componenta frontend (client)

Componenta frontend are o importanta deosebita deoarece cu aceasta interactioneaza consumatorul, astfel avand puterea primei impresii. Indiferent de cat de sofisticat sau de puternic ar fi serverul, daca interfata cu care interactioneaza utilizatorul nu este prietenoasa, acesta poate pleca, astfel restaurantul pierzand un potential client.

In cazul interfetei proiectate pentru aplicatia prezentata au fost folosite Angular 2+, JavaScript, HTML si CSS, iar punctele tari ale acestora constau in simplitate si usurinta de utilizare, asadar consumatorului fiindu-i usor sa vizualizeze si sa comande produsele dorite.

Asa cum a fost mentionat mai sus in faza de proiectare a aplicatiei, la deschiderea site-ului utilizatorul este intampinat de pagina principala unde se afla “headerul” si lista de mancaruri. Totodata se remarca in poza de mai jos faptul ca daca utilizatorul doreste sa caute vreun produs, poate sa o faca prin intermediul cautarii de tip text. La apasarea butonului “Cauta!” se schimba si link-ul aplicatiei, astfel putand accesa cautarea chiar dintr-un link presalvat. Un lucru foarte asemanator se intampla si la cautarea facuta prin intermediul selectarii tagurilor de deasupra listei de produse.



Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Daca la cautare nu este gasit niciun produs se afiseaza un anunt pentru a specifica acest lucru.

Nu am gasit ceea ce cautai!

La selectarea unui produs utilizatorul este mutat pe pagina produsului unde are optiunea de a adaugare in cos a acestuia. Odata adauga in cos, produsul este salvat in memoria locala a browserului de pe care a fost folosita aceasta functionalitate, astfel consumatorul putand sa comanda mai tarziu produsele salvate chiar si la o revenire pe site la alta data.

localhost:4200/food/Paste%20Bolognese

Pizza4You

Top 10ContactCarucior



Paste Bolognese

Carne de Vita

Piper

Sos de Rosii

Cascaval

Paste

Carne

Timp de preparare in jur de 25 de minute

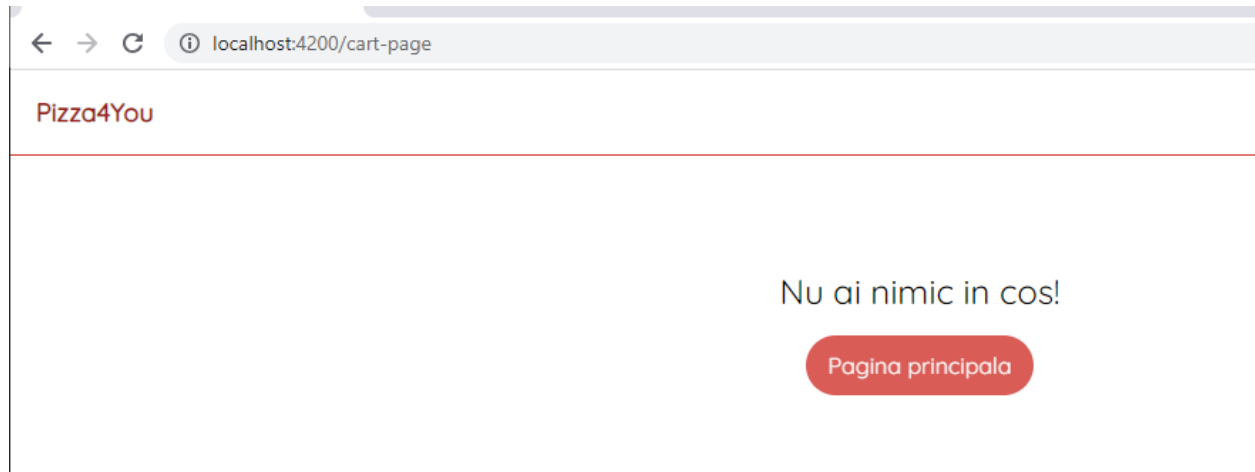
Pret: 25 lei

Adauga in Cos

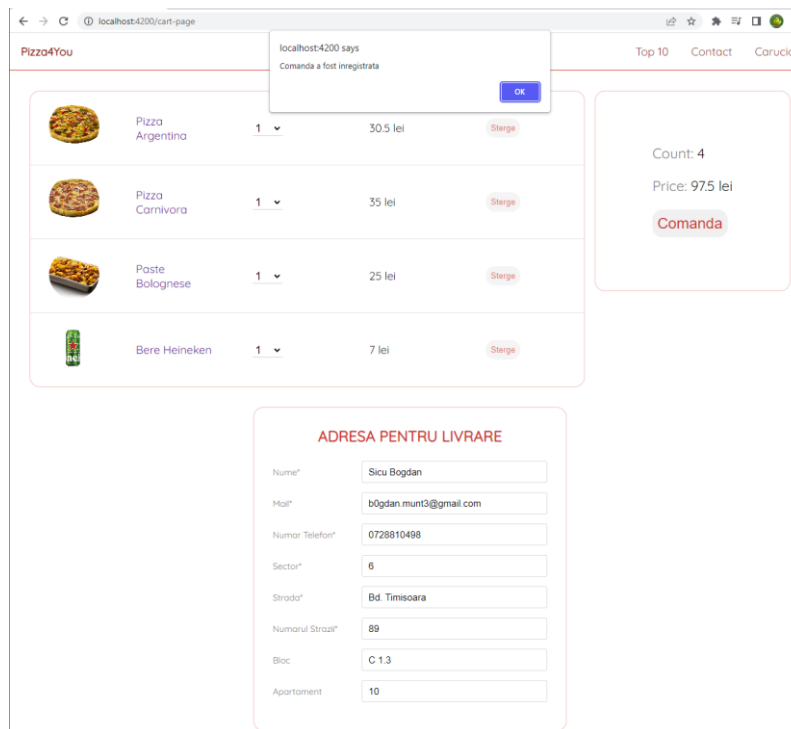
Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Daca clientul intra pe pagina de carucior fara a avea nimic salvat este intampinat de un mesaj impreuna cu un buton care il redirectioneaza la meniul restaurantului.

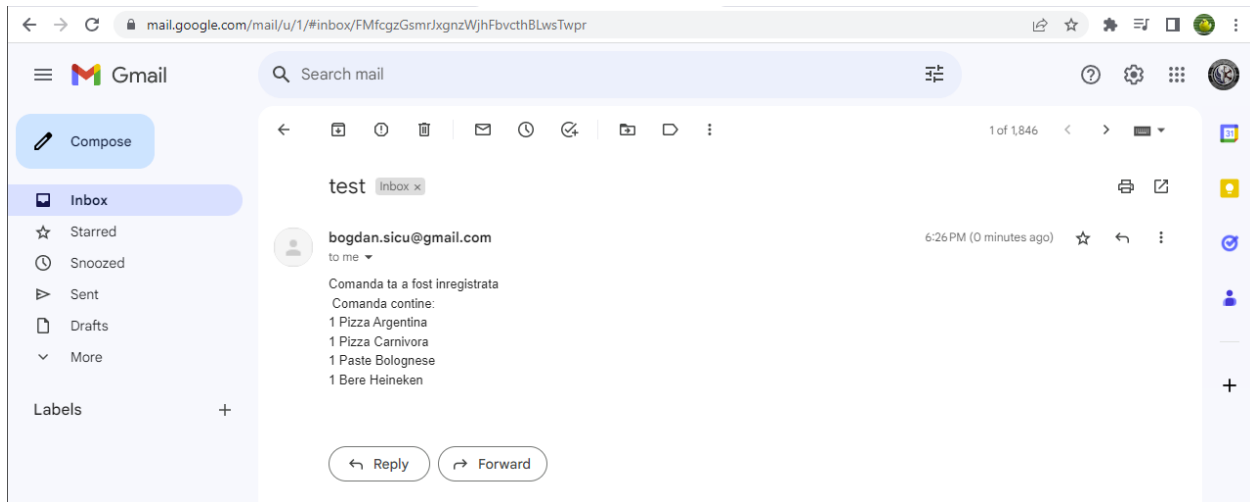


Odata ce sunt adaugate cateva produse in cos sunt expuse impreuna cu pretul fiecarui produs, cantitatea, optiunile de stergere a produsului, formularul pentru adresa de livrare si optiunea de comanda. La plasarea comenzii, daca totul este in regula utilizatorul primeste notificare atat pe site cat si pe mailul personal asa cum se observa in pozele de mai jos.

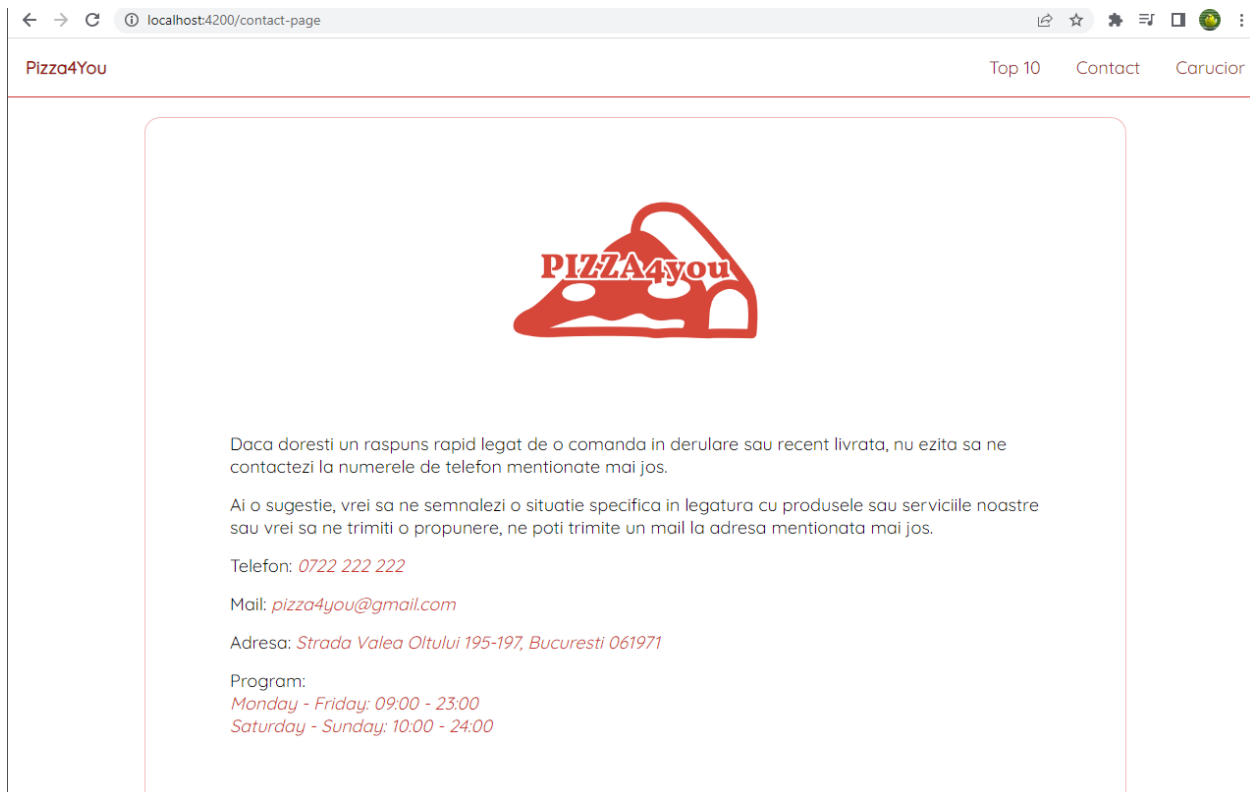


Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică



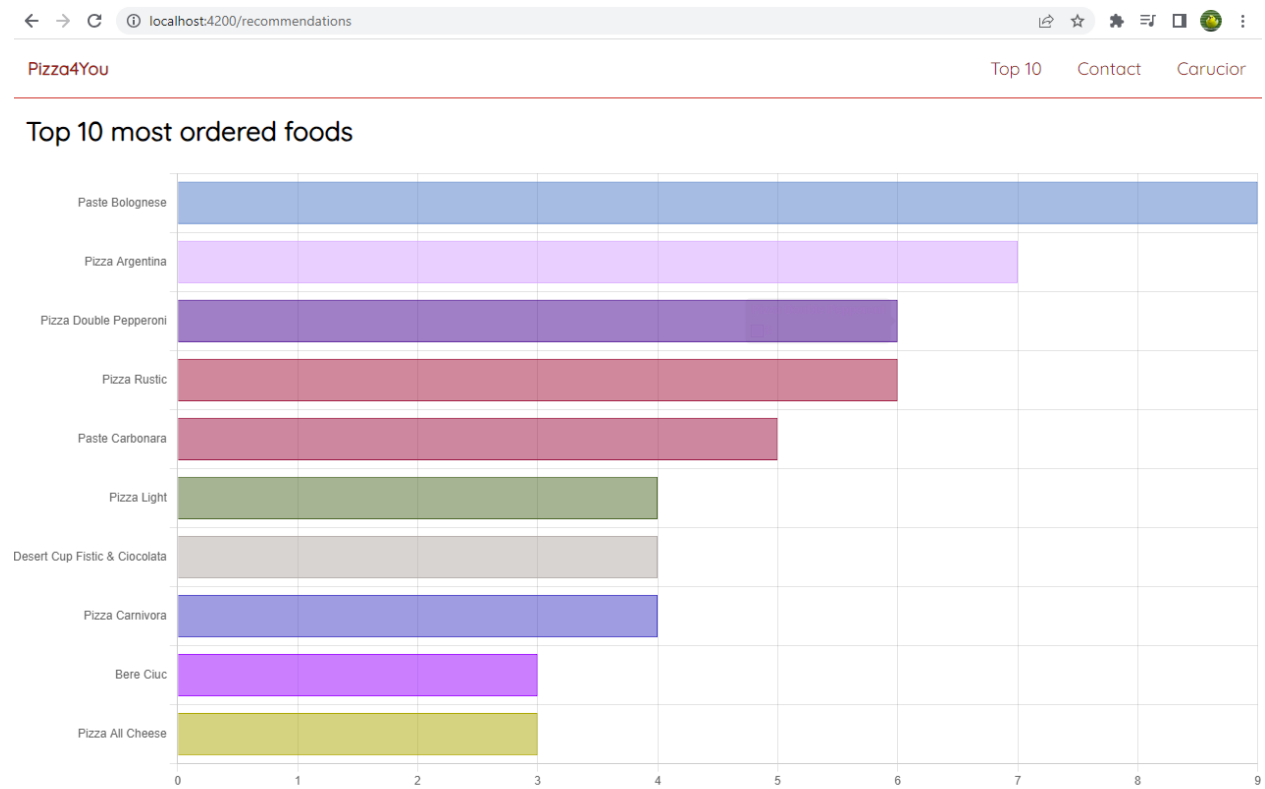
Ultimele doua ferestre au rol informational, fereastra de contact a restaurantului avand date statice care nu se modifica indiferent de interactiunea utilizatorului cu site-ul.



Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Ultima pagina constituie un grafic de tip “bar chart” care se actualizeaza la fiecare comanda plasata, astfel utilizatorii putand vedea in timp real cele mai vandute produse.



Concluzie

Alegerea de a dezvolta o soluție software de tip E-commerce pentru prezentarea și vânzarea produselor unui restaurant local a fost determinată de succesul rasunător pe care l-au avut atât comerțul electronic indiferent de domeniul de activitate cât și succesul firmelor de catering din ultimii ani.

Unul din avantajele semnificative pe care le are o aplicație de acest tip este accesibilitatea, astfel restaurantul putând fi vizitat, chiar și doar prin intermediul online, de către mai mulți posibili consumatori. Fiind un website conceput pentru o firmă locală și nu pentru un lanț mare de restaurante lipsesc diverse funcționalități precum crearea de cont, dar acest lucru poate fi și un bonus deoarece gigantul industriei, fiind atât de cunoscut, obligă utilizatorii să își creeze conturi atunci când doresc să dea o comandă direct pe site, problema care nu este prezentă în soluția prezentată mai sus.

Posibile îmbunătățiri care ar ajuta ar fi posibilitatea de creare de cont neobligând utilizatorii de a avea neapărat nevoie de unul, astfel putând păstra și beneficiul menționat anterior. Un cont de utilizator aduce diverse beneficii, cum ar fi: istoricul comenzilor, urmărirea unei comenzi, posibilitatea plății online, păstrarea adresei nemaifiind necesară introducerea acesteia la fiecare comandă și așa mai departe. O altă funcționalitate avantajoasă ar fi introducerea unei secțiuni accesibile doar conturilor de tip admin pentru facilitarea adăugării de noi produse într-un mod mai ușor pentru angajații care nu sunt foarte descurcați cu bazele de date.

În concluzie, aplicația prezentată are potențialul de a crește vânzarile restaurantului care se folosește de aceasta, astfel asigurându-și un loc mai solid pe piața industriei de catering care este într-o continuă creștere.

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Bibliografie

Carti:

“The Theory of Catering” – Autori: R. Kinton, V. Ceserani si D. Foskett

“The History of the Internet” – Autori: Christos J.P. Moschovitis, Hilary Poole, Tami Schuyler si Theresa M. Senft

“E-commerce – History of online shopping” – Autor: Simaira Mou

“Data Structures and Algorithms in Java 2nd Edition” – Autor: Robert Lafore

“REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces” – Autor: Mark Masse

“JavaScript: The Definitive Guide” – Autor: David Flanagan

“Learning SQL: Generate, Manipulate and Retrieve Data” – Autor: Alan Beaulieu

“Mastering Spring 5” – Autor: Ranga Rao Kanaram

Link-uri:

<https://medium.com/made-from-scratch-fine-catering/the-history-of-catering-in-the-united-states-b7a3c64184f4>

<https://www.lefrog.co.uk/the-history-of-catering/>

<https://viaverdimiami.com/history-of-catering/>

<https://edcora.ro/istoria-internetului-pe-scurt/>

<https://business.com.ro/istoria-si-evolutia-comertului-electronic/>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

<https://developer.mozilla.org/en-US/docs/Web/HTML>

<https://developer.mozilla.org/en-US/docs/Web/CSS>

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

<https://www.typescriptlang.org/docs/>

<https://angular.io/docs>

<https://docs.oracle.com/en/java/>

<https://docs.spring.io/spring-framework/docs/current/reference/html/>

<https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>

<https://www.ibm.com/topics/java-spring-boot>

<https://www.postgresql.org/docs/>

<https://react.dev/>

<https://stackdiary.com/front-end-frameworks/>

<https://learn.microsoft.com/en-us/dotnet/>