

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. Frontend	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. Backend	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	19
4.1. Descrierea generală a sistemului informatic	19
4.2. Specificarea cerințelor sistemului informatic	20
4.3. Analiza sistemului informatic	21
4.3.1. Diagrame de activitate	21
4.3.2. Diagrame de procese și colaborare in BPMN	25

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

4.4. Proiectarea sistemului informatic.....	27
4.4.1. Diagrama de clase detaliată.....	27
4.4.2. Proiectarea bazei de date.....	28
4.4.3. Proiectarea interfețelor utilizator.....	29
4.5. Sistemul informatic implementat	30
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 top 10 cele mai vândute produse. De asemenea, pe această pagină se află și o listă a ofertelor culinare prezente pentru restaurantul respectiv, însoțită de un motor de căutare de tip text.
- Secțiunea de contact conține date relevante despre restaurant, cum ar fi numărul de telefon, 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 “Top 10” este pagina în care consumatorul poate vizualiza primele 10 cele mai vândute produse ale restaurantului, datorită transparenței putând face cu ușurință alegeri pe baza graficului prezentat.

Funcționalitatea aplicației este în continuare susținută de butoanele prezente. Apăsând pe un produs, se încarcă o secțiune pentru descrierea 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

2. Prezentarea domeniului

Aplicațiile de tip comerț pe internet au foarte multe avantaje. Acest fapt este 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 nobile. Asemănător, în Grecia antică, simpozioanele erau găzduite de oamenii bogați unde oaspeții mâncau și se distrau [10].

Înaintând pe axa temporală până în timpul Evului Mediu, în Europa erau adesea găzduite petreceri de către regi sau nobile. 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 [11].

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 [9].

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 [1].

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 [12].

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 [2].

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 [3].

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 [13].

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 până în prezent. Dezvoltarea tehnologiei și creșterea internetului au dat naștere unei

Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

afaceri în plină expansiune care se extinde anual. Comerțul electronic este aici pentru a rămâne și acesta 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.

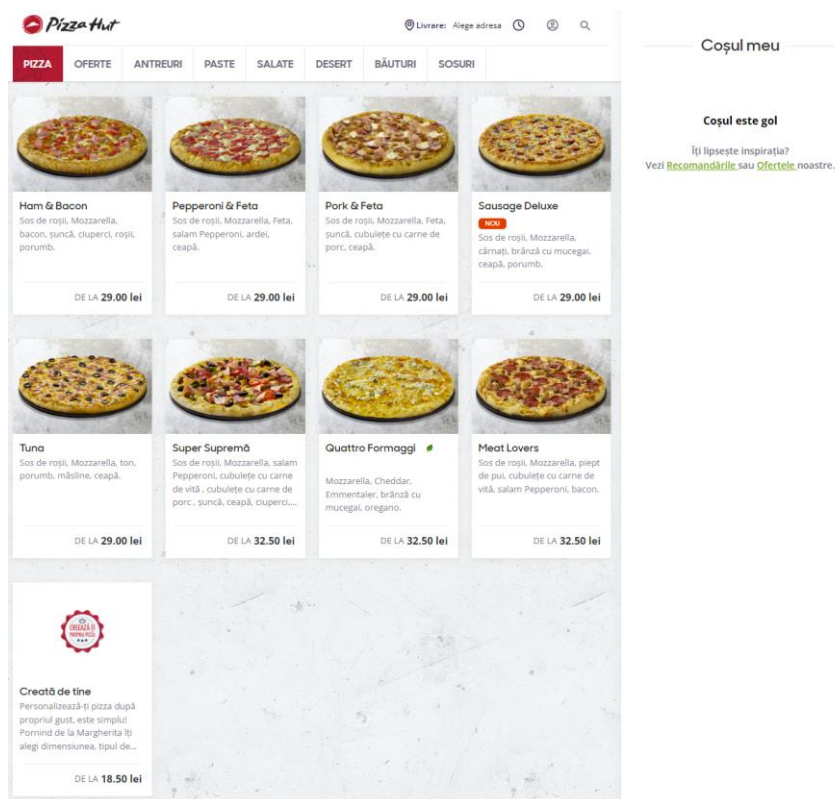
The screenshot displays the Pizza Hut website interface. At the top, there's a navigation bar with the Pizza Hut logo, a delivery address field, and social media icons. Below this is a menu bar with categories: PIZZA, OFERTE, ANTREURI, PASTE, SALATE, DESERT, BĂUTURI, and SOSURI. The main content area features a large promotional banner for 'AI PIZZA MARE LA PREȚ DE PIZZA MEDIE' (You get a large pizza at the price of a medium one) with a 'COMANDĂ ACUM' (Order Now) button. Below the banner, there's a section titled 'OFERTE' (Offers) with four promotional cards: 1. 'AI PIZZA MARE LA PREȚ DE PIZZA MEDIE' (You get a large pizza at the price of a medium one) with details about the offer being available from March to June 2023. 2. 'PENTRU ZILE LUNGI DE VARĂ, ÎNCEARCĂ FLATBREAD' (For long summer days, try Flatbread) with details about the offer being available from June 1st to June 30th. 3. 'GOOD DEALS DE LA 21.000 LEI' (Good Deals from 21,000 Lei) with details about the offer being available from June 1st to June 30th. 4. 'LUNEA DEVINE UN PIC MAI BUNA' (The month becomes a little better) with details about the offer being available from June 1st to June 30th. On the right side, there's a shopping cart icon labeled 'Coșul meu' (My Cart) which shows 'Coșul este gol' (Cart is empty) and a link to 'Vezi Recomandările sau Ofertele noastre' (See our Recommendations or Offers).

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, 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 a crea o pizza pe propriul plac. Franciza este renumită pentru rețetele și combinațiile 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 cu scopul de a include opțiuni mai sănătoase ale preparatelor sale. 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 KFC uniform in a kitchen, 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 RON", "62.00 RON", "AFLĂ MAI MULTE".
- % OFERTE #PEBUNE**: "REVENDICĂ-ȚI OFERTA PREFERATĂ!", "VEZI DETALII".

The footer contains social media links (Facebook, YouTube, Instagram, TikTok), app download links (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. Frontend

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 frontend 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 [6].

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++. Totodată, 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 [14].

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,

fiecare limbaj are rolul său, HTML fiind scheletul, CSS fiind aspectul clientului iar JS fiind modul în care funcționează aplicația [15].

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 evolueze.

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 [16].

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” aduce în discuție 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 rezultatul obținut.

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, acestea 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 [17].

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. Unealta aceasta 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 [18].

Diferențele dintre Angular și AngularJS

Deși au aproape același nume și ambele framework-uri sunt folosite pentru scrierea mai rapidă a aplicațiilor web, 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 sufixul 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. Backend

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, rulează oriunde” (“write once, run anywhere”), ceea ce înseamnă că odată 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++ oferind o mare flexibilitate când vine vorba de alocarea memoriei, pe când C# și Java folosesc un “garbage collector” care se ocupă cu dezalocarea memoriei atunci când se pierde referința unei zone de memorie [19].

Utilizări

Limbajul Java a fost folosit pentru foarte multe tipuri de aplicații de-a lungul timpului. În prezent este predominant folosit pentru scrierea aplicațiilor pentru Android și pentru aplicațiile de tip backend.

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 [4].

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 [8].

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, legându-le împreună. În trecut, se configurau aceste obiecte denumite “beans” prin fișiere de tip XML (Extensible Markup Language), dar în prezent se folosesc adnotări [20].

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 [5]. 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 dependențe deja inițializate pe care dezvoltatorul software nu mai trebuie să le configureze manual. Acest element ajută nu doar la grăbirea 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 [22].

Abordarea de opinie (“Opinionated approach”)

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

configureze tot proiectul în mod manual, Spring Boot selectează pachetele care trebuie instalate și totodată, ce valori implicite să folosească.

În timpul procesului de inițializare, dezvoltatorii software pot selecta toate dependențele de care are nevoie proiectul utilizând platforma Spring Boot Initializr. Platforma este una simplă care nu are nevoie de cod și crează un proiect de tip Spring Boot care vine împreună cu toate configurările necesare selectate de către programator.

Aplicații de sine stătătoare (“Standalone applications”)

Spring Boot le permite dezvoltatorilor software să creeze aplicații gata de a fi utilizate. Prin integrarea unui server web precum Tomcat sau Netty în aplicație, aceasta poate funcționa în mod independent, nefiind nevoie de server extern. Totodată, dacă este necesar, programatorii pot opta pentru excluderea serverului web implicit [21].

3.2.4. PostgreSQL

PostgreSQL este un sistem de gestionare de baza de date care se axează pe conformarea SQL și pe extensibilitate. PostgreSQL oferă funcționalități precum: trigger, view, chei externe și proceduri stocate. De asemenea, de la stații de lucru, la depozite de date sau servicii web, PostgreSQL este conceput pentru a gestiona o varietate largă de sarcini [23].

3.3. Alte tehnologii folosite în domeniul aplicațiilor de catering/E-commerce

3.3.1. React

React sau React.js este un framework pentru crearea aplicațiilor de tip website. Acesta are la bază limbajul JavaScript și este construit pe componente. Asemenea Angular, principalul scop al framework-ului React este de a ajuta dezvoltatorii de aplicații web în evoluția acestora într-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 [24].

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 “frontend”, putând astfel să pastreze același “backend”. De asemenea, există și opțiunea de a utiliza ambele opțiuni în paralel deoarece cuplarea de la client la server este una detașată, putând utiliza două aplicații de tip client care comunică cu același backend.

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 [26].

Chiar dacă acest framework este rescris cu totul de la fundație, 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 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 aplicației este de a prezenta produsele unei firme de catering locală într-o manieră cât mai simplistă pentru utilizator astfel facilitând o navigare ușoară printre ofertele culinare și mai ales ușurând procesul de comandă a produselor dorite.

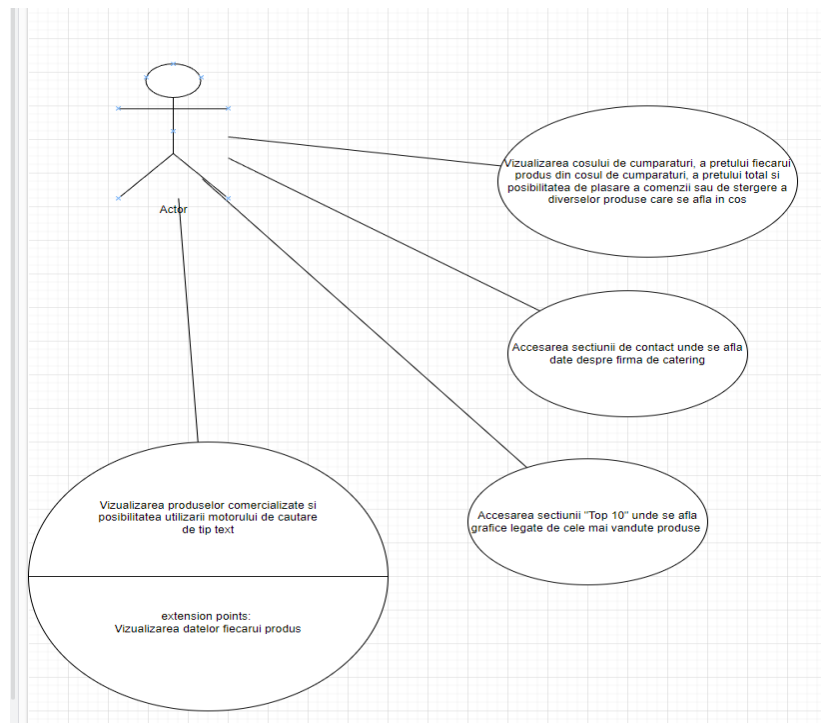
La deschiderea aplicației, utilizatorul este întâmpinat de pagina principală unde se află “headerul” care rămâne vizibil indiferent de ceea ce accesează consumatorul în pagină, acționând ca un meniu al aplicației. Opțiunea implicită este cea de prezentare a tuturor produselor împreună cu diverse categorii de produse și o bară de căutare de tip text unde utilizatorul poate introduce cuvinte cheie pentru a cauta produsul dorit. În plus, utilizatorul poate apăsa pe un produs pentru a-l vizualiza în mod detaliat și pentru a-l putea adăuga în coșul de cumpărături care ține minte produsele adăugate chiar și la închiderea website-ului. În acest mod clientul poate să-si continue cumpărăturile mai tarziu dacă dorește. Pentru a comanda produsele, cumpărătorul trebuie să completeze un formular care conține câteva date personale și 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 constau 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 vândute 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 diagramă faptul că utilizatorul poate să vizualizeze gama de produse comercializată de restaurantul “Pizza4you”, poate căuta produsele dorite prin motorul de căutare de tip text și de asemenea poate căuta produse și după categoriile acestora.

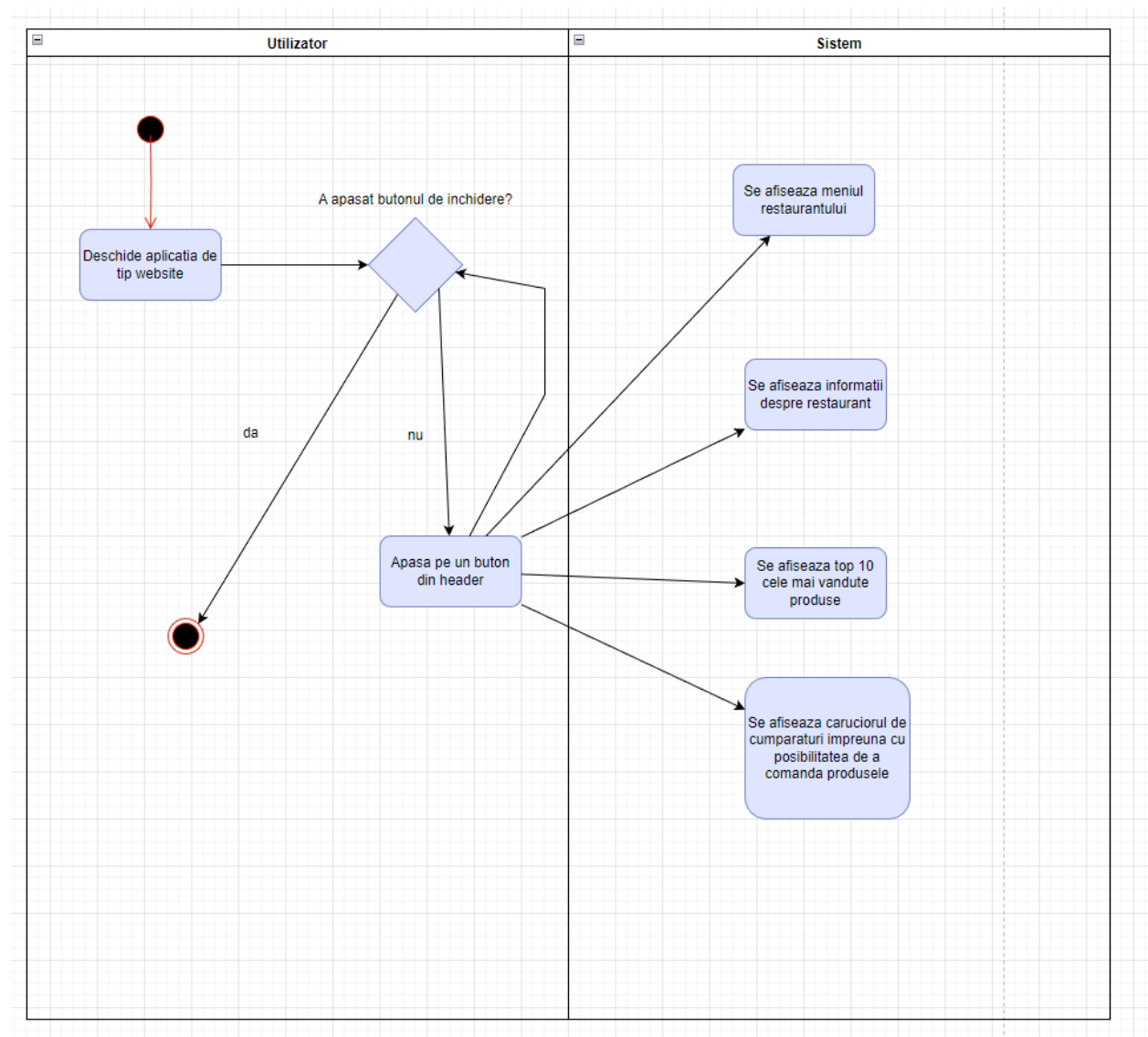
Din opțiunea de vizualizare a coșului de cumpărături, consumatorul poate să șteargă produse din acesta, să comande produsele respective și să vadă costul total al elementelor comandate. Pe lângă acestea, utilizatorul poate accesa și paginile “Contact” și “Top 10” pentru a vizualiza informații despre restaurant și totodata, pentru a vedea cele mai vândute produse ale restaurantului.

4.3. Analiza sistemului informatic

4.3.1. Diagrame de activitate

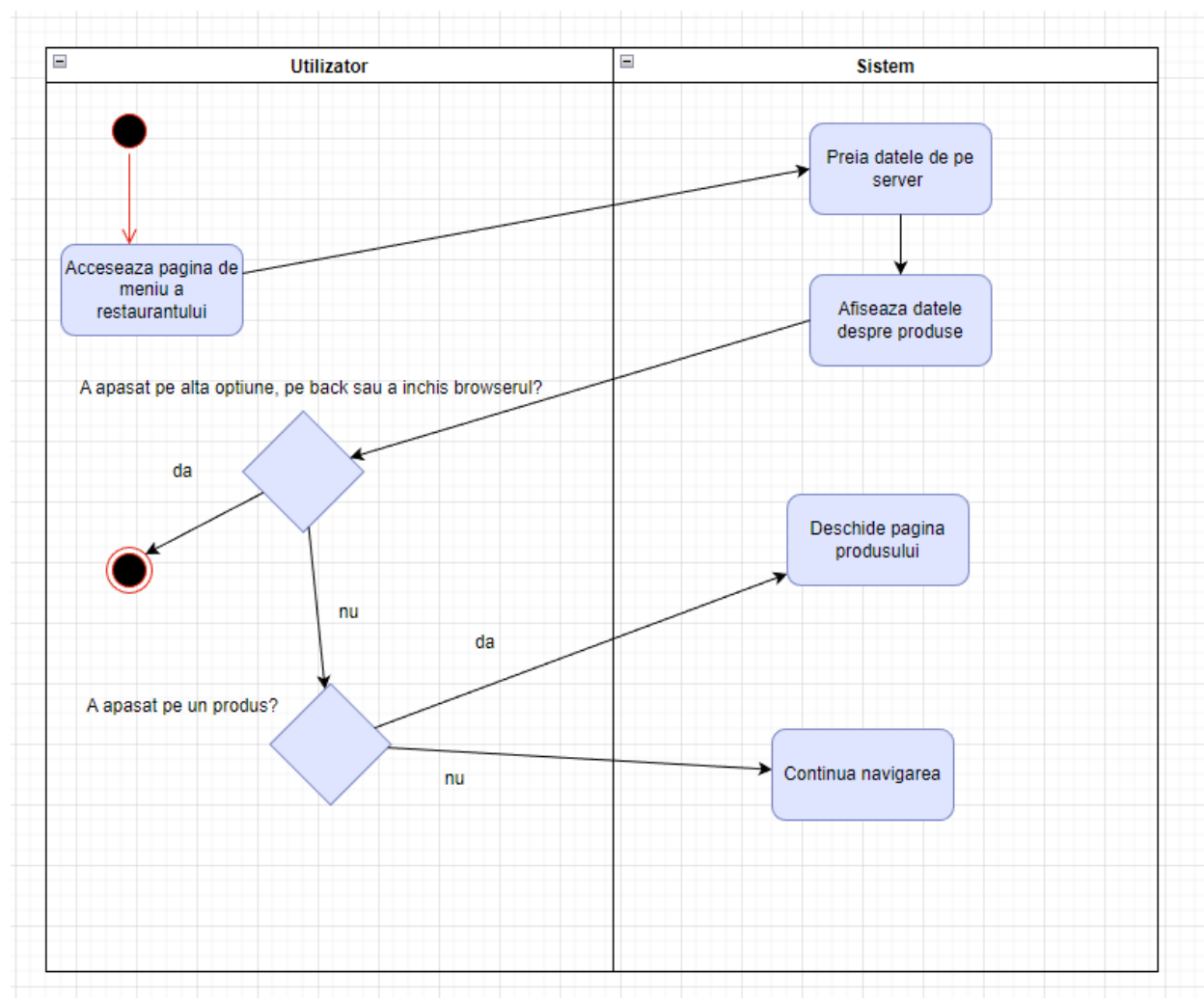
Diagramele de activitate sunt o unealtă esențială deoarece ne oferă un mijloc vizual prin care se poate înțelege mai bine funcționalitățile aplicației prezentate și cum se navighează prin aceasta. Diagramele sunt folosite în procesul de construire a unei aplicații pentru că prezintă un limbaj standard pe care-l pot înțelege programatorii, indiferent de specializarea acestora, astfel evitând problemele de comunicare dintre echipele care trebuie să lucreze împreună.

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 coș de cumpărături sau de meniu al restaurantului.

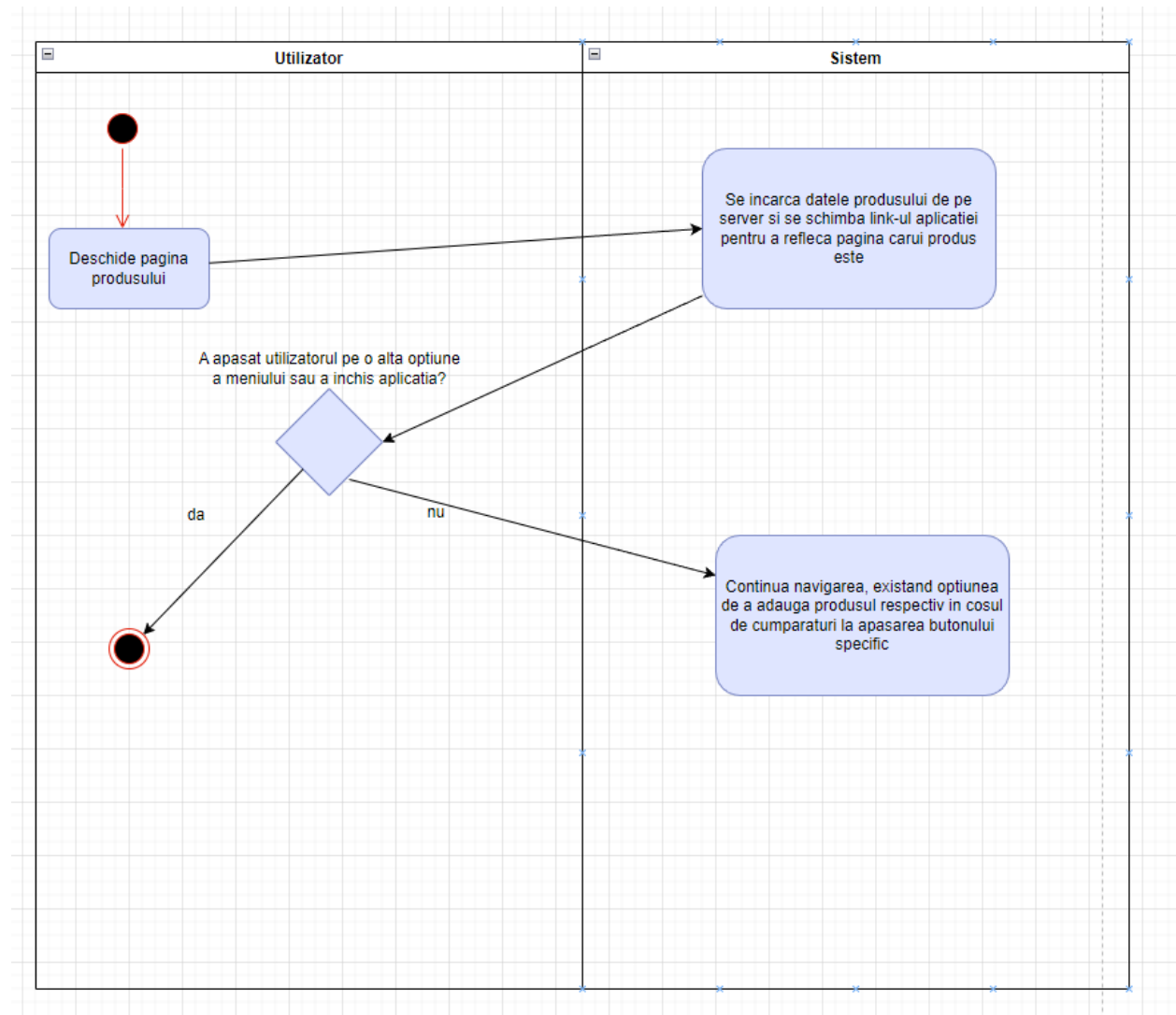
Diagrama de activitate – Pagina cu produse



În această diagramă se observă 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 comanda ceva ce nu se află în meniul restaurantului. De asemenea, utilizatorul poate trece la altă pagină oricând dorește, permițând o experiență fluidă în

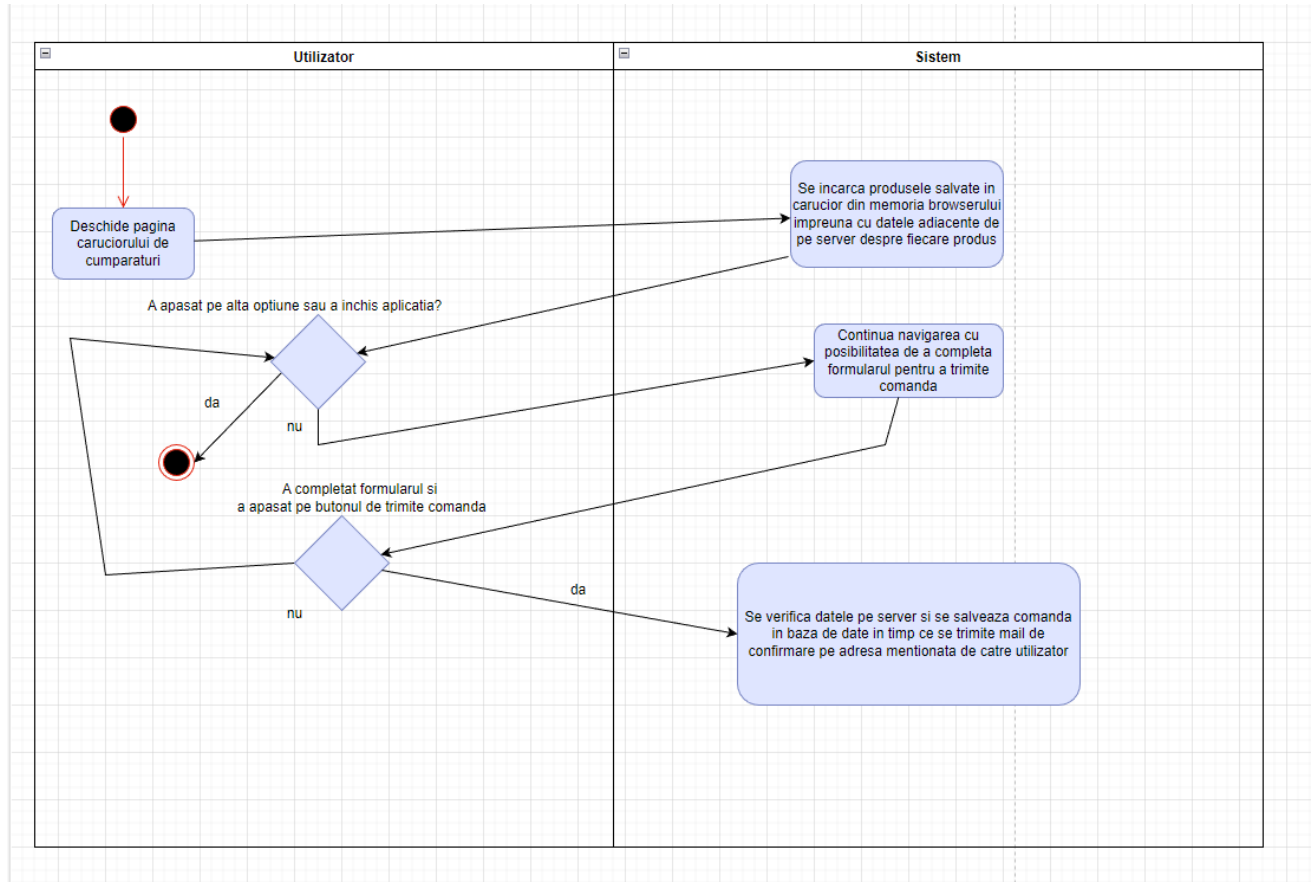
utilizarea aplicației. Totodată, din această pagină se poate accesa și pagina individuală a fiecărui produs.

Diagrama de activitate – Pagina produsului



În figura de mai sus se poate vedea faptul că datele fiecărui produs sunt luate de pe server, aceste date conținând poza, numele, ingredientele și așa mai departe. Se poate remarca faptul că utilizatorul poate să adauge produsul respectiv în coșul de cumpărături pentru a-l comanda. O caracteristică importantă care trebuie menționată este faptul că flexibilitatea pe care o aveam datorită “headerului” nu dispăre, consumatorul putând oricând să navigheze la altă pagină.

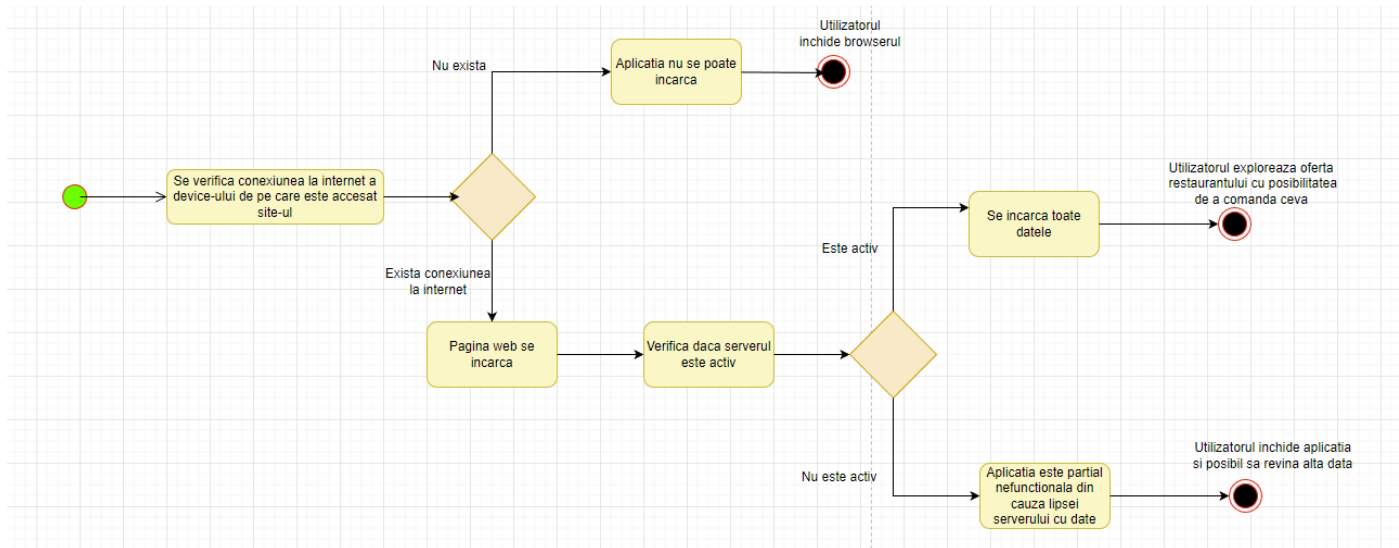
Diagrama de activitate – Căruciorul de cumpărături



În poza prezentată anterior se poate observa faptul că funcționalitatea căruciorului de cumpărături 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 coș, 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 in BPMN

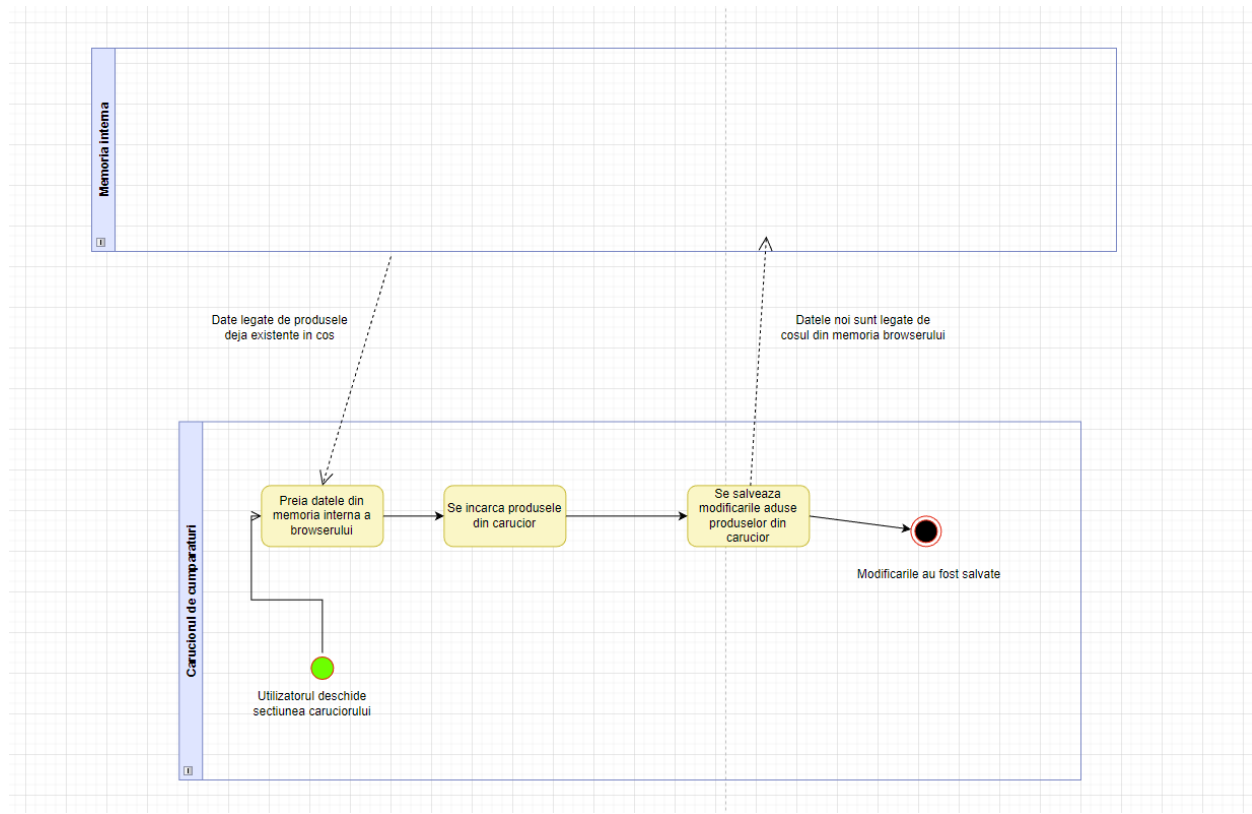
Diagrama de Procese - Deschiderea Interfeței cu Utilizatorul (website-ul)



În diagrama prezentată se remarcă faptul că funcționarea corespunzătoare a aplicației depinde de mai mulți factori. Primul factor este conexiunea la internet, utilizatorul neputând să acceseze website-ul fără aceasta. Totodată, chiar dacă consumatorul are o conexiune stabilă la internet pe tot parcursul utilizării aplicației, aplicația 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 soluție software 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 o multitudine de soluții pentru a minimiza problemele menționate mai sus, dar și aceste soluții au limitări, proprietarul site-ului web neputând influența 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 dă greși și se închide, alta poate să îi ia locul.

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

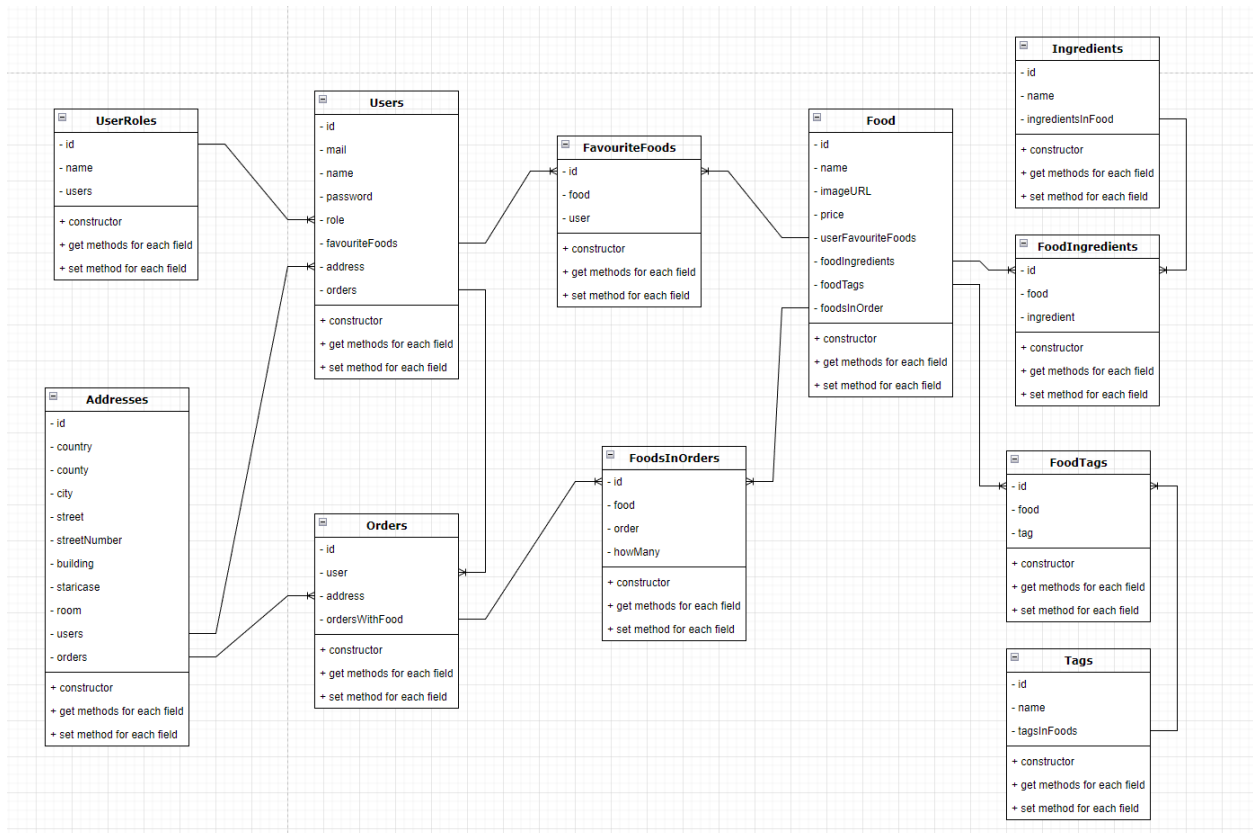


În poza de mai sus se poate observa procesul de comunicare dintre interfața cu care interacționează utilizatorul și memoria internă a browserului pe care rulează website-ul. Atunci când consumatorul deschide pagina coșului de cumpărături, aplicația încarcă datele din memoria internă și afișează produsele, aducând de pe server și datele adiacente de care este nevoie. În altă ordine de idei, interacțiunea cu memoria internă apare la orice schimbare adusă coșului de cumpărături, indiferent dacă este vorba de adăugarea sau ștergerea de produse din acesta.

4.4. Proiectarea sistemului informatic

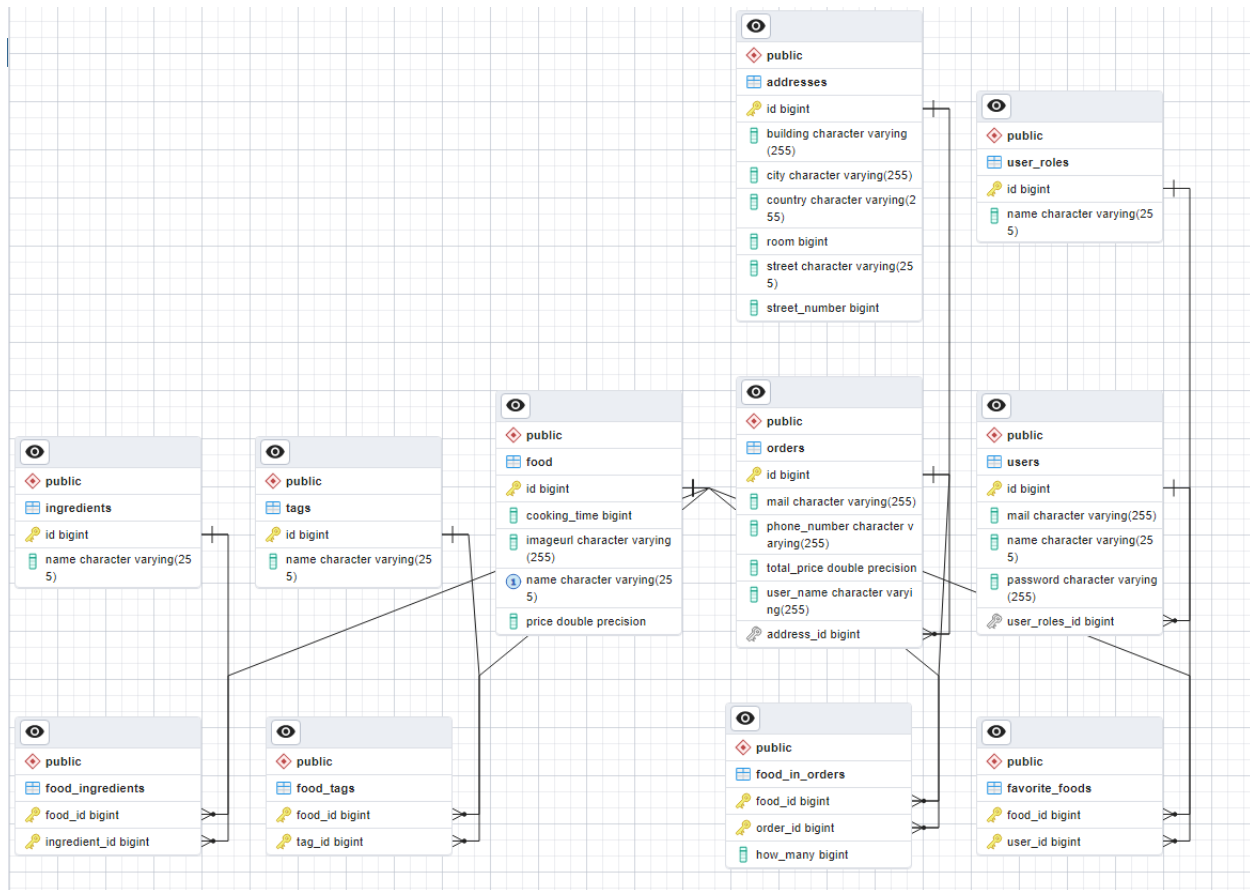
4.4.1. Diagrama de clase detaliată

În diagrama prezentată mai jos se poate vizualiza într-un mod simplificat clasele principale ale aplicației “Pizza4you”. Totodată se observă conținutul claselor împreună cu relațiile dintre acestea. După cum se poate observa, clasele sunt construite în așa fel încât să fie cât mai ușor și eficient de lucrat cu ele, având date cât mai puține și o redundanță cât mai mică. De asemenea, clasele respective conțin atât metode pentru modificarea și accesarea fiecărui atribut, cât și constructorii pentru crearea de obiecte.



4.4.2. Proiectarea bazei de date

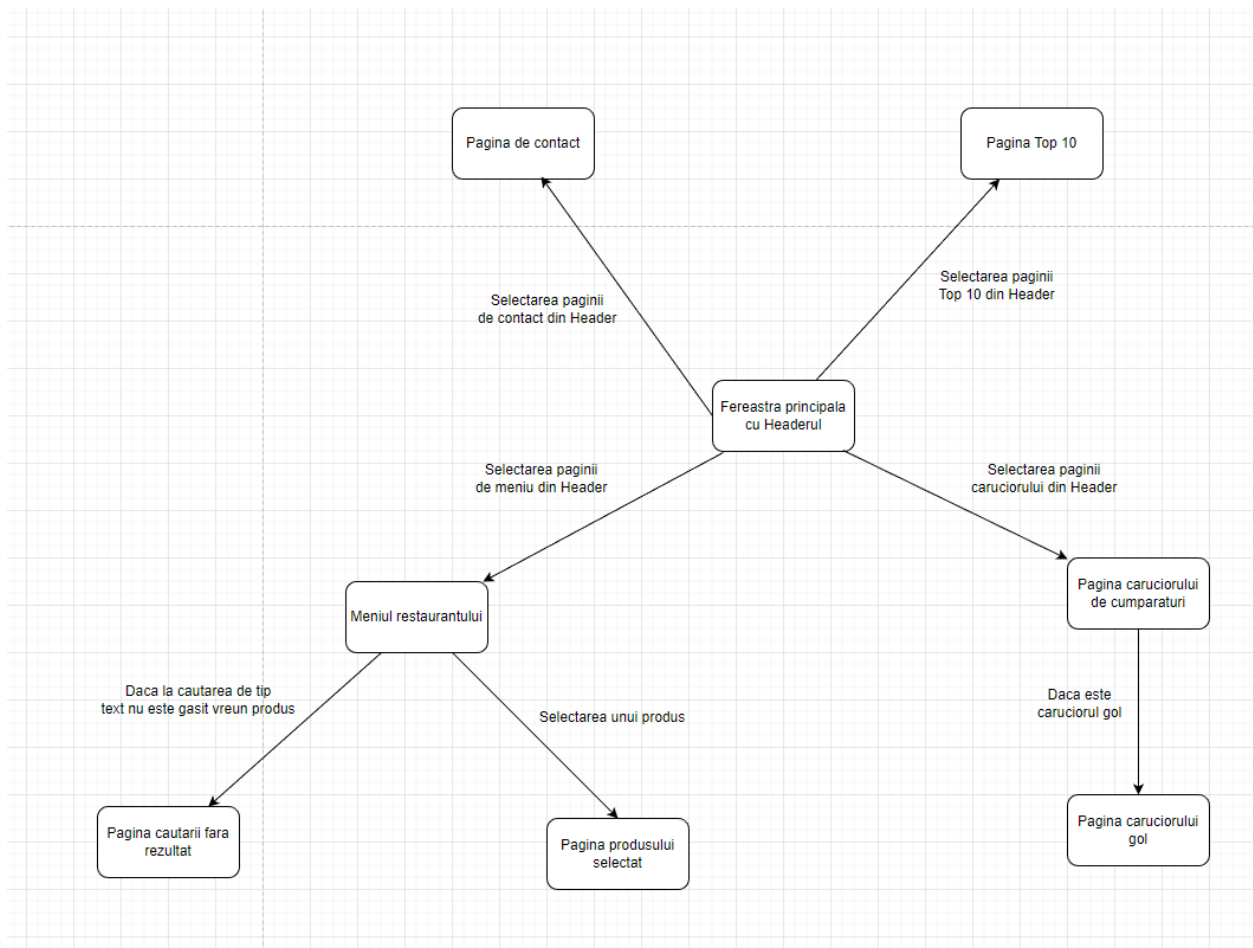
O diagramă a bazei de date este o componentă esențială pentru o mai bună înțelegere a modului în care sunt stocate informațiile, arătând legăturile, datele și tipurile acestora într-un mod ușor de înțeles vizual. Mulțumită avantajelor sale, acest tip de diagramă este utilizată pe scară largă pentru a crea aplicații complexe, ajutând programatorii să comunice și să lucreze mai eficient pentru crearea sau întreținerea de sisteme software.



Mulțumită modului prin care este realizată aplicația prezentată, diagrama de clase este un pas important și premergător în proiectarea bazei de date, tabelele și clasele fiind corelate. După cum se poate observa mai sus, legăturile 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 legătură la cele două tabele. Totodată, se pot observa tipurile de date din fiecare tabelă.

4.4.3. Proiectarea interfețelor utilizator

Diagrama de mapare a interfețelor este necesară pentru o înțelegere mai bună a procesului de navigare prin aplicație, astfel putând știind cum se ajunge dintr-un punct al aplicației în altul fără apariția posibilelor confuzii.



În imaginea prezentată la acest punct observăm o hartă a interfețelor aplicației. Aici putem vizualiza modul prin care sunt conectate interfețele unele cu celelalte. După cum putem remarca, odată ce deschidem aplicația suntem întâmpinați de pagina principală unde se află “headerul” aplicației care funcționează ca un meniu, astfel reușindu-se comutarea 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 dispare niciodată putând merge la altă pagină direct din “header” chiar dacă nu există opțiunea în mod explicit în fereastra 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. Aplicația are funcționalități asemănătoare cu alte website-uri de catering precum cele deținute de Pizza Hut sau Jerry's Pizza.

În continuare vor fi prezentate componentele din care este construită soluția software.

4.5.1. Componenta backend (RESTful Api)

Această parte este “creierul” soluției informatice. Toate datele sunt procesate prin intermediul codului Java și a framework-ului Spring Boot din care este compus serverul și de asemenea stocate în baza de date PostgreSQL cu care interacționează backendul. Chiar dacă serverul a fost conceput pentru a rula cu website-ul aplicației, acesta poate oricând să ruleze împreună cu alte aplicații în cazul în care se dorește schimbarea interfeței cu clientul sau adăugarea unor noi aplicații pentru a îmbunătăți experiența utilizatorilor (ex: aplicație de telefon care să interacționeze cu serverul).

Baza de date

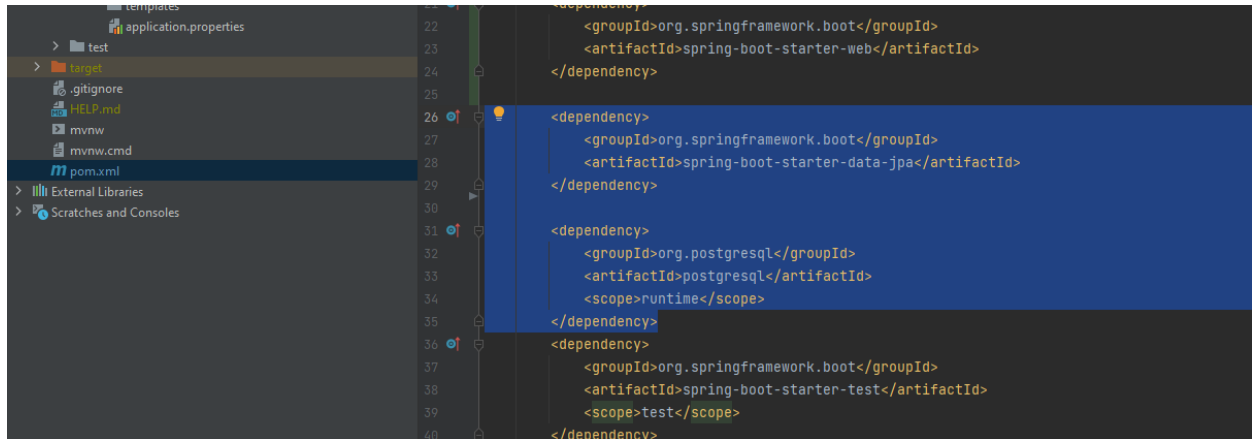
În trecut pentru fiecare aplicație care necesita o bază de date era nevoie de o echipă de analiști care să fie specializați în limbajul SQL pentru conceperea, construirea și menținerea bazei de date. În zilele noastre nu mai este esențial acest lucru datorită framework-urilor de mapare de date care ajută programatorii ca din diferite limbaje să creeze baze de date complexe direct din codul sursă al aplicației, fără a mai fi nevoie de o echipă specializată în SQL. Într-adevar, este important ca programatorii, indiferent dacă lucrează cu Java, C# sau alte limbaje de programare folosite pentru backend, să cunoască elemente reprezentative despre bazele de date de tip SQL și NoSQL și diferențele dintre acestea, astfel putând înțelege cu adevărat capabilitățile bazei de date pe care o implementează.

În aplicația prezentată, maparea bazei de date se face cu ajutorul framework-urilor “Hibernate” & “JPA”. Aceste framework-uri trebuiesc specificate în fisierul “pom.xml” care

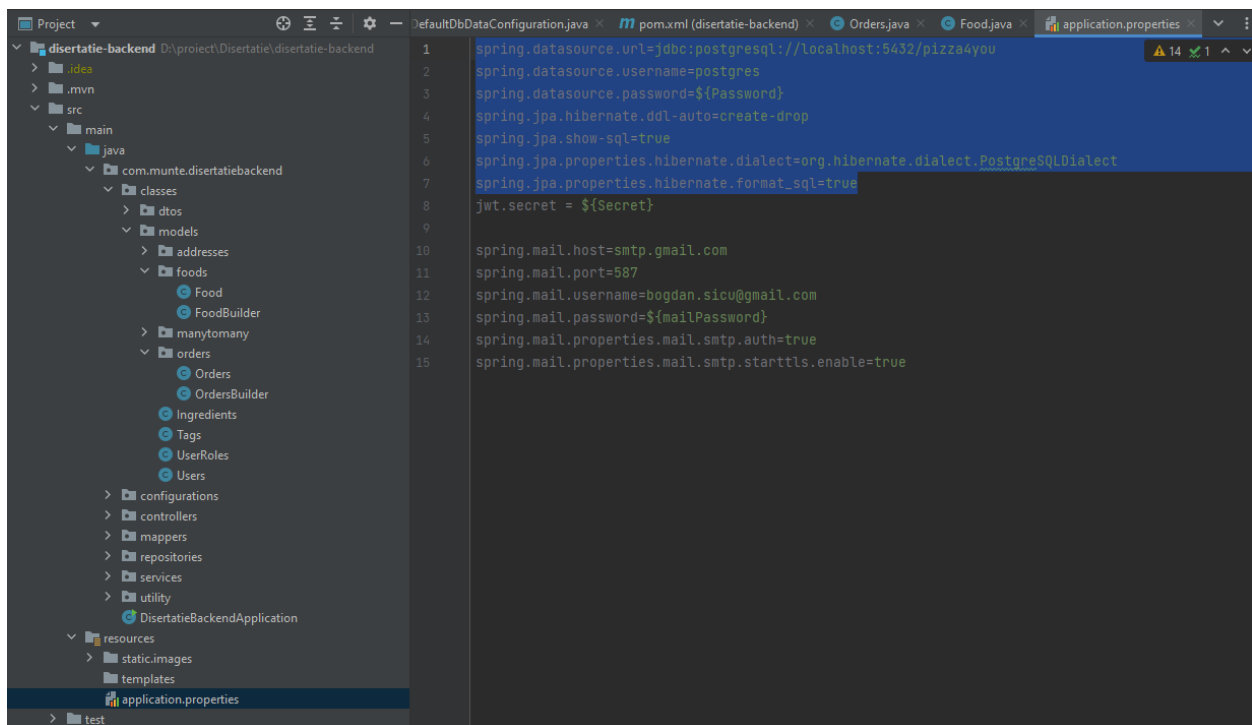
Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

conține toate dependențele proiectului față de alte librării și framework-uri. Pentru a putea conecta baza de date la aplicația noastră trebuie să specificăm și framework-ul pentru tipul bazei de date, indiferent dacă este SQL sau NoSQL și indiferent de furnizorul acesteia.



Important de menționat este faptul că pentru a putea crea o bază de date, trebuie să existe deja instalat software-ul pentru baza de date dorită și de asemenea, trebuie menționată conexiunea serverului de baze de date în fișierul “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")
```

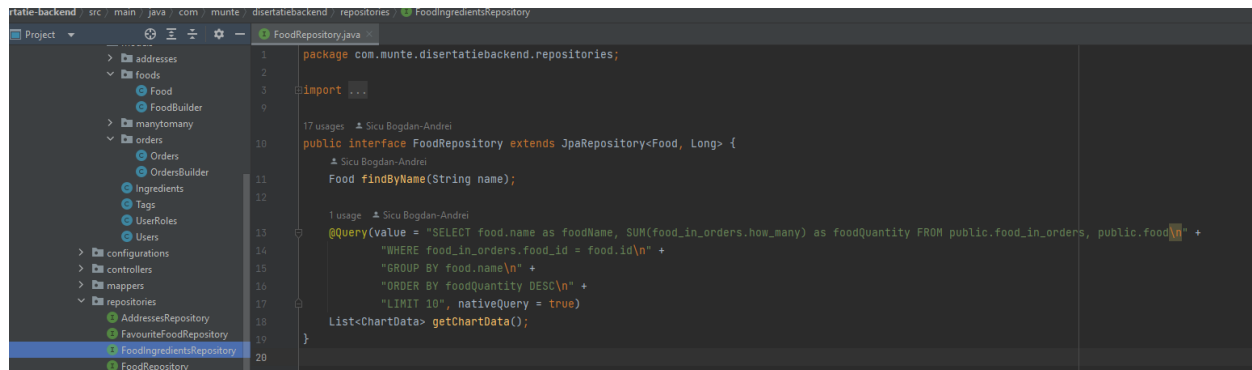
Așa cum am menționat mai sus, în prezent, în cele mai multe cazuri, baza de date este construită direct din codul sursă al aplicației, acest lucru fiind implementat ușor diferit de la limbaj la limbaj. În cazul nostru, având Java & Hibernate & JPA, există adnotări speciale care specifică ce tip de clase din aplicație pot fi considerate tabele și ce attribute din aceste clase pot fi considerate câmpuri în tabele.

Se poate observa în poza alăturată faptul că la începutul clasei avem adnotarea “@Entity” care marchează clasa ca fiind o tabelă pentru baza de date. Adnotarea “@Column”

desemnează attributele clasei ca fiind coloane în tabela care trebuie să fie creată, având și alte adnotări complementare pentru a specifica dacă un câmp este id, autogenerat, dacă este cheie externă și așa mai departe. Se remarcă faptul că există diverse valori care pot fi date în interiorul parantezelor de lângă aceste cuvinte cheie pentru a specifica diverse opțiuni semnificative precum: numele tabelii, numele coloanei și așa mai departe. Lipsa acestor opțiuni duce la aplicarea opțiunilor implicite (ex: coloana va fi denumită exact după denumirea atributului din clasa scrisă în Java).

Preluarea datelor din baza de date

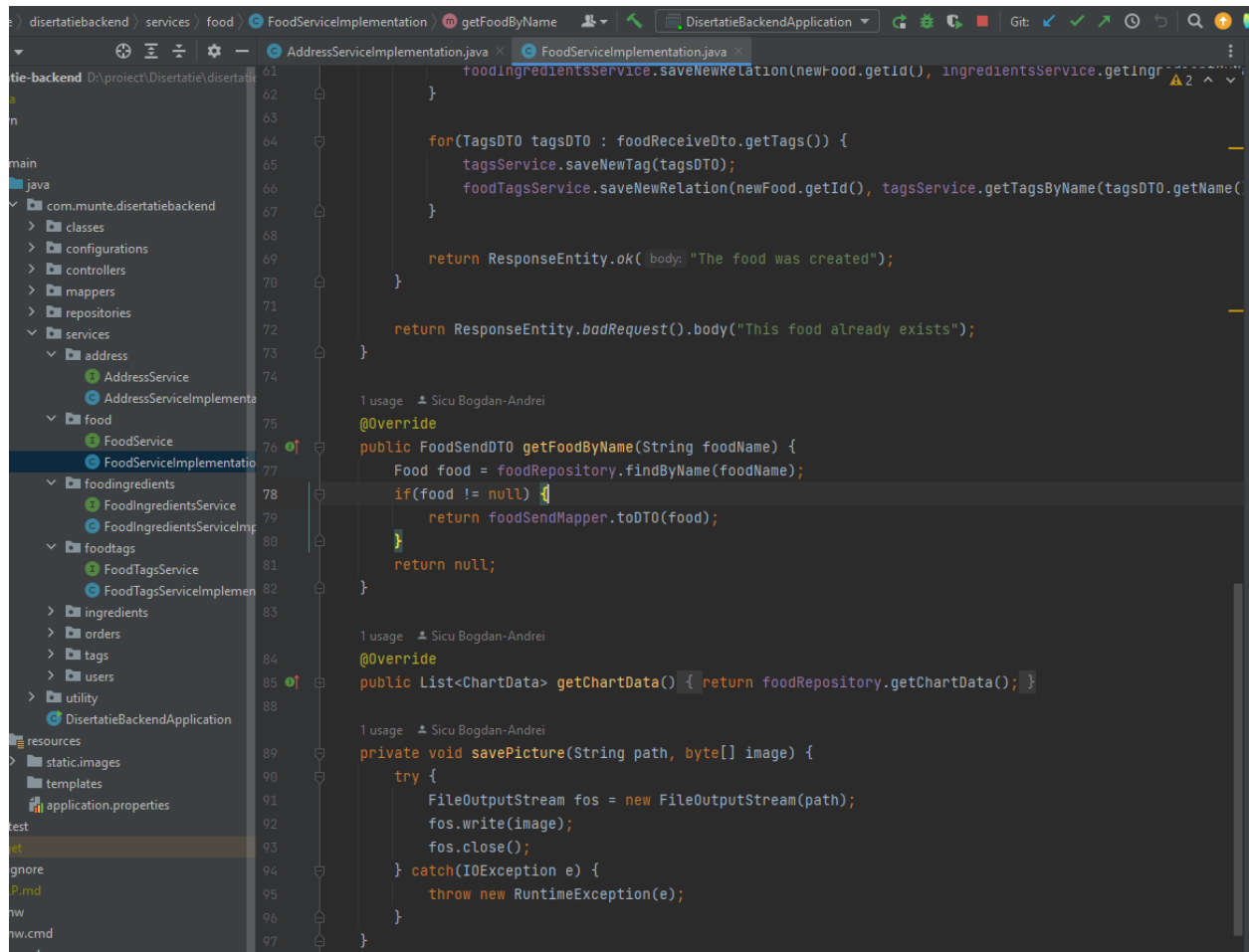
Pentru tranzacțiile asupra bazei de date este necesară o clasă separată de tip “interface” care extinde funcționalitățile JPA prin “JpaRepository<Clasă, Cheie_Primară>”. Chiar dacă există multe funcționalități generice de tip CRUD deja implementate, uneori este necesară o implementare mai avansată care poate fi scrisă direct în limbaj nativ SQL. Acest lucru poate fi observat în fotografia de mai jos.



Procesarea datelor

O bună practică a unui backend de tip REST este segmentarea acestuia în “Repository” (legatura cu baza de date) -> “Service” (procesarea datelor) -> “Controller” (preluarea de cereri de la aplicațiile de tip client). Acestea fiind menționate, secțiunea prezentă se axează pe partea de “Service” care spre deosebire de părțile de “Repository” și “Controller” care se ocupă de elemente foarte specifice, acesta este foarte volatil putând procesa datele printr-o multitudine de moduri.

Spre exemplu, în service-ul “FoodServiceImpl.java” are loc o diversificare a acțiunilor realizate fiindcă la adăugarea de mâncăruri noi este necesară o procesare adițională din cauza pozei cu care vine la pachet. Această poză poate fi transmisă doar ca vector de bytes și trebuie convertită în fișier atunci când ajunge pe server și salvată în memoria acestuia.



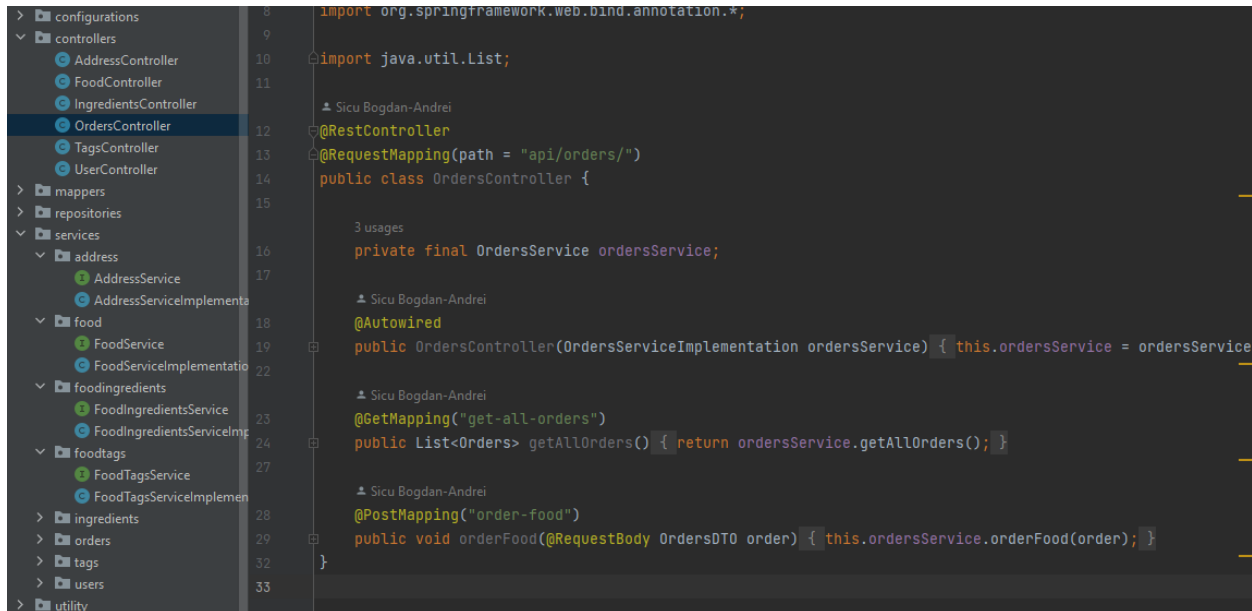
Primirea cererilor de la client

Așa cum a fost menționat mai sus, există o parte a serverului care se ocupă în mod special doar de preluarea cererilor de la aplicațiile de tip client și de trimiterea de răspuns atunci când se termină de procesat datele.

Se remarcă în poza de mai jos faptul că pentru ca o componentă a aplicației să se poată ocupa de primirea de cereri, trebuie să fie desemnată ca fiind o clasă de tip “Controller”. Acest lucru este făcut prin adăugarea adnotării “@RestController”, este adesea folosită împreună cu adnotarea “@RequestMapping()” în care se pot adăuga diverși parametri precum cel de “path” care specifică adăugarea din link care este alipită pentru a face o cerere către 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;

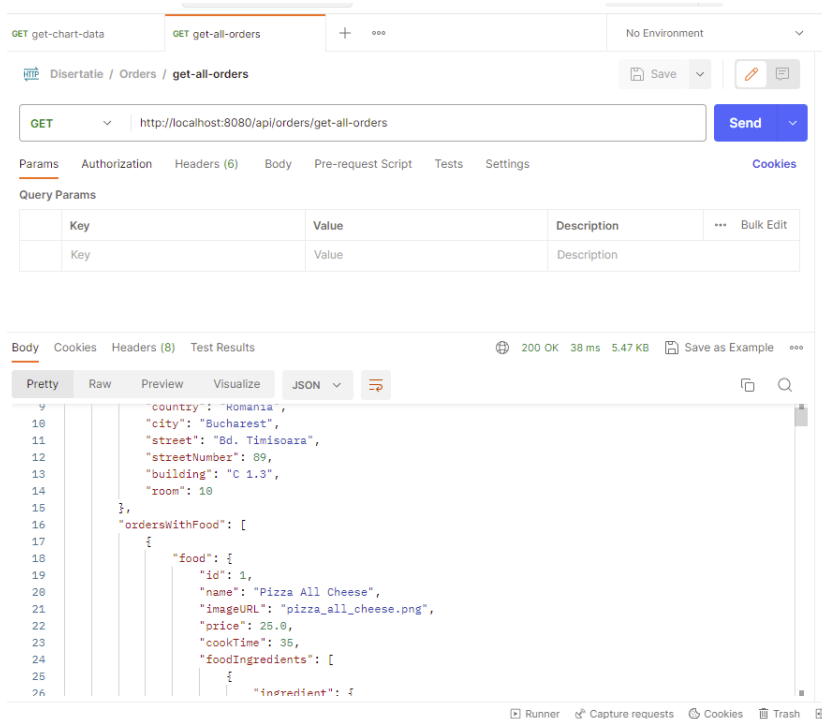
@SicBogdanAndrei
@RestController
@RequestMapping(path = "api/orders/")
public class OrdersController {

    3 usages
    private final OrdersService ordersService;

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

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

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



Pentru că o metodă să fie considerată “request” sau cerere trebuie să aibă adnotarea “@RequestMapping”, verbul fiind oricare dintre cele acceptate, cel mai des întâlnite fiind: “GET” (ex: @GetMapping), “PUT”, “POST” și “DELETE”. De asemenea, pentru fiecare cerere se poate adăuga o extensie a link-ului care trebuie accesat. Spre exemplu, pentru apelarea metodei “getAllOrders()”

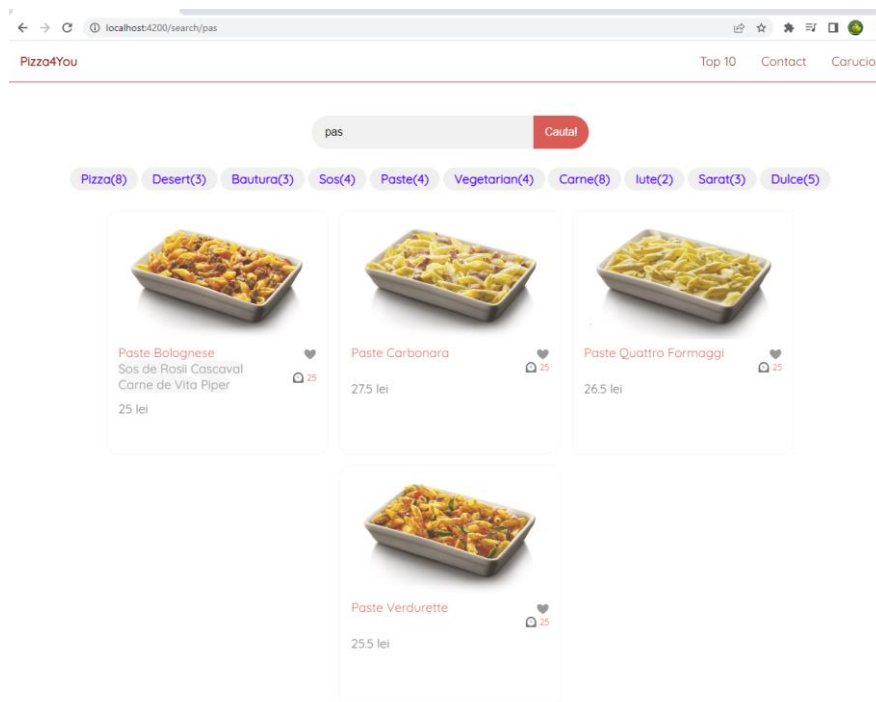
trebuie trimisă o cerere la “domeniu/api/orders/get-all-orders” având verbul “GET”, așa cum se poate vedea în poza adiacentă.

4.5.2. Componenta frontend (client)

Componenta frontend are o importanță deosebită deoarece cu aceasta interacționează consumatorul, astfel are puterea primei impresii. Indiferent de nivelul de complexitate sau de cât de puternic ar fi serverul, dacă interfața cu care interacționează utilizatorul nu este prietenoasă, acesta poate renunța la utilizarea sa, astfel restaurantul pierzând un potențial client.

În cazul interfeței proiectate pentru aplicația prezentată au fost folosite Angular 2+, TypeScript, HTML și CSS, iar punctele tari constau în simplitate și ușurința de utilizare, așadar consumatorului fiindu-i comod să vizualizeze și să comande produsele dorite.

Așa cum a fost menționat anterior în faza de proiectare a aplicației, la deschiderea site-ului utilizatorul este întâmpinat de pagina principală unde se află “headerul” și lista de mâncăruri. Totodată se remarcă în poza de mai jos faptul că dacă utilizatorul dorește să caute vreun produs, poate să o facă cu ușurință și fără a depune prea mult efort prin intermediul căutării de tip text. La apăsarea butonului “Caută!” se schimbă și link-ul aplicației. În acest mod se poate accesa căutarea chiar dintr-un link presalvat. Un lucru foarte asemănător se întâmplă și la căutarea făcută cu ajutorul selectării tagurilor de deasupra listei de produse.




Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Dacă la căutare nu este găsit niciun produs se afișează un anunț pentru a specifica acest detaliu.

Nu am gasit ceea ce cautai!

La selectarea unui produs utilizatorul este mutat pe pagina produsului unde are opțiunea de adăugare în coș a acestuia. Odată ce se adaugă în coș, produsul este salvat în memoria locală a browserului de pe care a fost folosită funcționalitatea, astfel consumatorul poate să comande mai târziu produsele salvate chiar și la o revenire pe site la alta dată.



Paste Bolognese

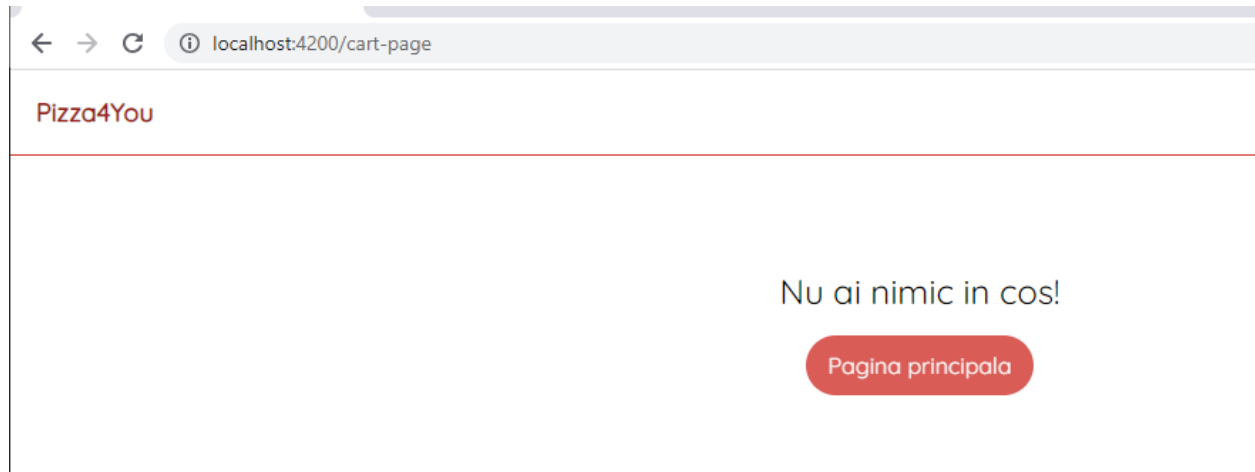
Timp de preparare in jur de 25 de minute

Pret: 25 lei

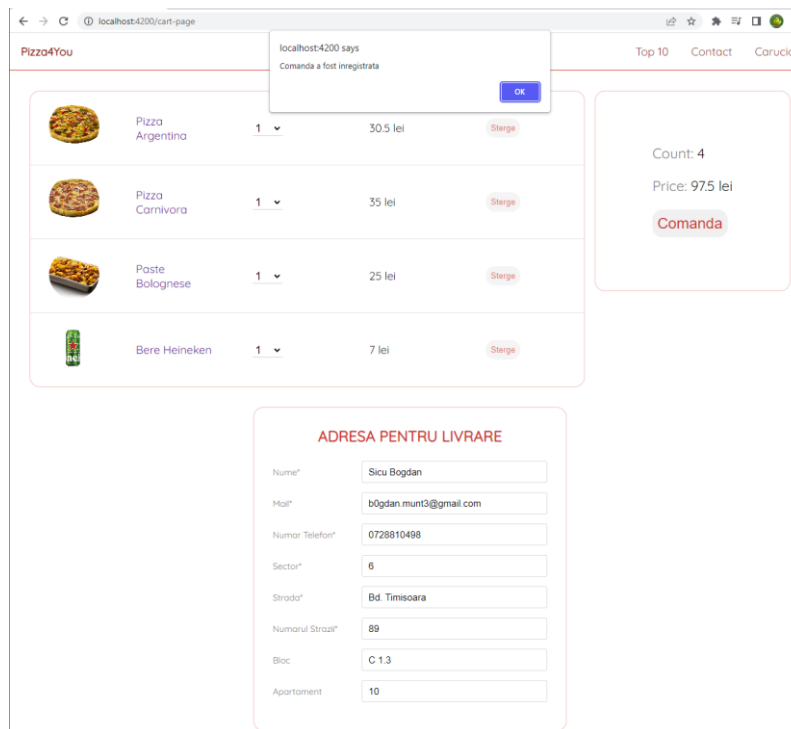
Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Dacă clientul intră pe pagina de cărucior fără a avea nimic salvat este întâmpinat de un mesaj împreună cu un buton care îl redirecționează la meniul restaurantului.

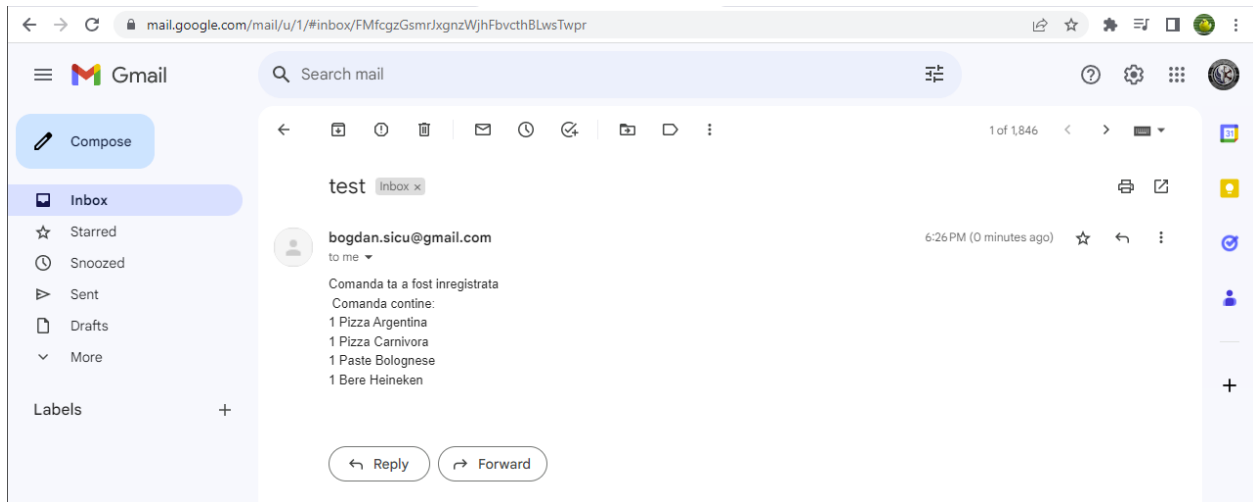


Odată ce sunt adăugate câteva produse în coș sunt expuse împreună cu prețul fiecărui produs, cantitatea și opțiunea de ștergere a produsului, formularul pentru adresa de livrare și opțiunea de comandă. La plasarea comenzii, dacă totul este în regulă utilizatorul primește notificare atât pe site cât și pe mailul personal așa cum se observă în pozele inserate mai jos.

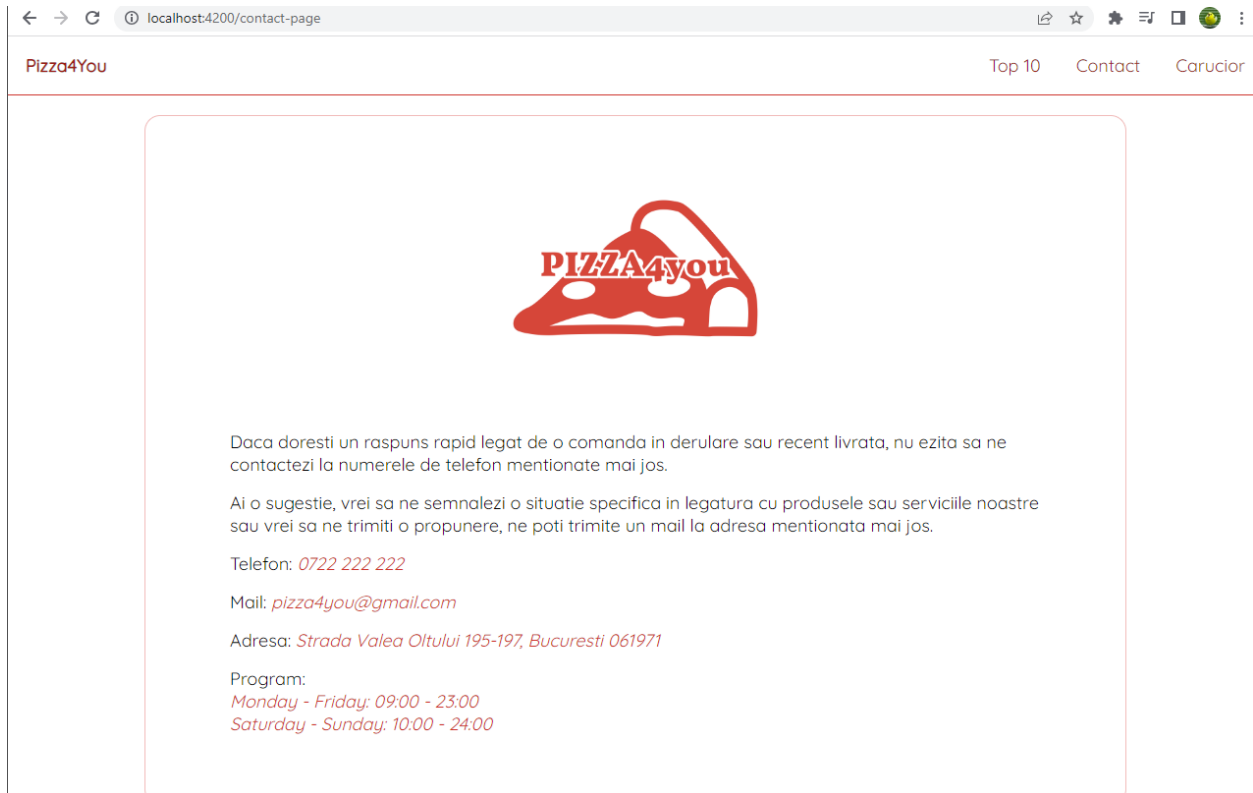


Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică



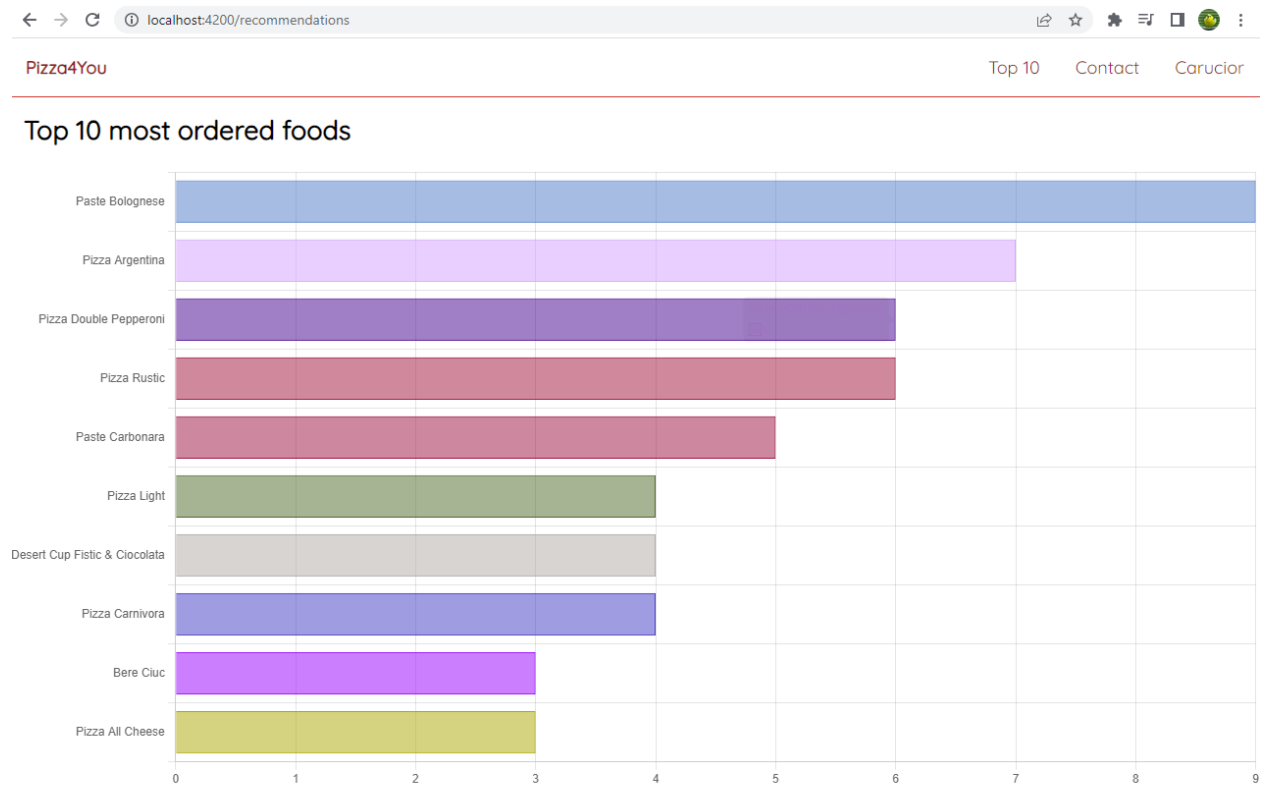
Ultimele două ferestre au rol informativ, fereastra de contact a restaurantului având date statice care nu se modifică indiferent de interacțiunea utilizatorului cu site-ul.



Academia de Studii Economice din București

Facultatea de Cibernetică, Statistică și Informatică Economică

Ultima pagină constituie un grafic de tip “bar chart” care se actualizează la fiecare comandă plasată, astfel utilizatorii putând vedea în timp real cele mai vândute și apreciate produse de către alți consumatori.



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 răsunător 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 atât în mod fizic cât și 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 giganții industriei, fiind atât de cunoscuți, obligă utilizatorii să își creeze conturi atunci când doresc să dea o comandă direct pe site, problemă care nu este prezentă în soluția enunțată mai sus.

Posibile îmbunătățiri care ar ajuta ar fi posibilitatea de creare de cont neobligând utilizatorii de a avea neapărat nevoie de unul. Prin prisma acestei situații se poate păstra ideea beneficiului 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 experimentați în privința bazelor de date.

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

Bibliografie

Carti:

- [1] “The Theory of Catering” – Autori: R. Kinton, V. Ceserani și D. Foskett
- [2] “The History of the Internet” – Autori: Christos J.P. Moschovitis, Hilary Poole, Tami Schuyler și Theresa M. Senft
- [3] “E-commerce – History of online shopping” – Autor: Simaira Mou
- [4] “Data Structures and Algorithms in Java 2nd Edition” – Autor: Robert Lafore
- [5] “REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces” – Autor: Mark Masse
- [6] “JavaScript: The Definitive Guide” – Autor: David Flanagan
- [7] “Learning SQL: Generate, Manipulate and Retrieve Data” – Autor: Alan Beaulieu
- [8] “Mastering Spring 5” – Autor: Ranga Rao Kanaram

Link-uri:

- [9] <https://medium.com/made-from-scratch-fine-catering/the-history-of-catering-in-the-united-states-b7a3c64184f4>
- [10] <https://www.lefrog.co.uk/the-history-of-catering/>
- [11] <https://viaverdimiami.com/history-of-catering/>
- [12] <https://edcora.ro/istoria-internetului-pe-scurt/>
- [13] <https://business.com.ro/istoria-si-evolutia-comertului-electronic/>
- [14] <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [15] <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [16] <https://developer.mozilla.org/en-US/docs/Web/CSS>

- [17] <https://www.typescriptlang.org/docs/>
- [18] <https://angular.io/docs>
- [19] <https://docs.oracle.com/en/java/>
- [20] <https://docs.spring.io/spring-framework/docs/current/reference/html/>
- [21] <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
- [22] <https://www.ibm.com/topics/java-spring-boot>
- [23] <https://www.postgresql.org/docs/>
- [24] <https://react.dev/>
- [25] <https://stackdiary.com/front-end-frameworks/>
- [26] <https://learn.microsoft.com/en-us/dotnet/>