

**Aplicatie de tip RESTful API - Client**

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

**Student:** Sicu Bogdan Andrei

Cuprins

[1) Introducere 3](#_Toc133327669)

[2) Prezentatrea domeniului 4](#_Toc133327670)

[2.1) Istoria firmelor de catering 5](#_Toc133327671)

[2.2) Istoria internetului 5](#_Toc133327672)

[2.3) Istoria comertului electronic 5](#_Toc133327673)

[2.4) Alte firme de catering 5](#_Toc133327674)

[3) Prezentartea tehnologiilor 6](#_Toc133327675)

[3.1) Front-end 6](#_Toc133327676)

[3.1.1) JavaScript 6](#_Toc133327677)

[3.1.2) HTML 6](#_Toc133327678)

[3.1.3) CSS 7](#_Toc133327679)

[3.1.4) TypeScript 8](#_Toc133327680)

[3.1.5) Angular framework 8](#_Toc133327681)

[3.2) Back-end 9](#_Toc133327682)

[3.2.1) Java 9](#_Toc133327683)

[3.2.2) Spring 10](#_Toc133327684)

[3.2.3) Spring Boot 10](#_Toc133327685)

[3.2.4) PostgreSQL 11](#_Toc133327686)

[3.3) Alte tehnologii folosite in domeniul aplicatiilor de catering/E-commerce 12](#_Toc133327687)

[3.3.1) React 12](#_Toc133327688)

[3.3.2) Aplicatii pentru telefon 12](#_Toc133327689)

[3.3.3) ASP.NET Core 12](#_Toc133327690)

[4) Prezentarea solutiei informatice 13](#_Toc133327691)

[4.1) Descrierea generala a sistemului informatic 13](#_Toc133327692)

[4.2) Specificarea cerintelor sistemului informatic 13](#_Toc133327693)

[4.3) Analiza sistemului informatic 14](#_Toc133327694)

[4.3.1) Diagrame de activitate 14](#_Toc133327695)

[4.4) Proiectarea sistemului informatic 15](#_Toc133327696)

[4.4.1) Diagrama de clase detaliata 15](#_Toc133327697)

[4.4.2) Proiectarea bazei de date 16](#_Toc133327698)

[Bibliografie 18](#_Toc133327699)

# Introducere

Sistemul este reprezentat de doua programe care comunica unul cu celalalt pentru a forma o aplicatie de comert pe internet indreptata spre servirea de mancaruri. Aplicatia are ca scop expunerea ofertelor culinare ale unui restaurant si posibilitatea de comandare a acestora.

Sectiunea de start a aplicatiei contine antetul in care se afla logo-ul si butoanele pentru accesarea paginilor de cos de cumparaturi, contact si de conectare la contul utilizatorului. De asemenea, pe aceasta pagina se afla si o lista a ofertelor culinare prezente pentru restaurantul respectiv, insotita de un motor de cautare de tip text.

Apasand pe un produs, se incarca o sectiune pentru prezentarea acestuia unde se pot observa diverse date, cum ar fi ingredientele folosite, pretul, poza produsului si optiunea de a fi adaugat in cosul de cumparaturi si/sau la sectiunea de mancaruri favorite ale utilizatorului curent.

Sectiunea de contact contine date relevante despre restaurant, cum ar fi numarul de telefon pentru contact, adresa de mail, locatia restaurantului si orele de functionare.

Sectiunea caruciorului este locul in care utilizatorul poate sa vizualizeze produsele pe care le are momentan in cosul de cumparaturi, precum si pretul total al produselor. De asemenea, exista posibilitatea stergerii produselor pe care posibilul cumparator nu le mai doreste.

Sectiunea profilului utilizatorului contine datele personale ale acestuia, precum si posibilitatea de a le modifica sau de a sterge contul cu totul. Printre datele sensibile se afla: numar de telefon, nume, mail si adresa de livrare.

# Prezentatrea domeniului

Aplicatiile de tip comert pe internet au foarte multe avantaje, acest fapt fiind dovedit atat de abundenta lor in contextul actual, cat si de succesul rasunator pe care l-au avut in timpul pandamiei de COVID-19. Una dintre cele mai de succes industrii ale comertului pe internet a fost si este in continuare industria de catering, acesta fiind si domeniul de activitate pentru solutia software prezentata. Atat avantaje cat si dezavantajele unei astfel de solutii se aplica pentru toate tipurile de business, indiferent daca sunt indreptate spre comertul cu alimente sau spre comertul cu bunuri nealimentare.

Printre avantaje, se numara urmatoarele:

* posibilitatea extinderii afacerii si catre utilizatorii care nu se afla in proximitatea locatiei de desfasurare a activitatii economice;
* posibilitatea de expunere a reclamelor
* accesibilitatea platii electronice

Cu toate acestea, exista si dezavantaje, cel mai mare dezavantaj fiind costul de productie si de intretinere a unei astfel de solutii software, fiind greu accesibila firmelor mici.

## 2.1) Istoria firmelor de catering

## 2.2) Istoria internetului

## 2.3) Istoria comertului electronic

## 2.4) Alte firme de catering

# Prezentartea tehnologiilor

## 3.1) Front-end

### 3.1.1) JavaScript

JavaScript este un limbaj de programare care sta la baza tehnologiilor de tip World Wide Web (www), alaturi de HTML si CSS, urmand ca despre acestea din urma sa detaliem in subcapitolele urmatoare. Se estimeaza ca incepand cu anul 2022, in jur de 98% din site-urile web utilizeaza JavaScript ca prim limbaj de programare pentru partea de front-end sau cunoscut si sub forma denumirii de client. Toate browserele cunoscute de catre publicul larg au incorporat un motor special pentru interpretarea limbajului JavaScript pe dispozitivele consumatorilor.

JavaScript sau cunoscut si dupa abrevierea de JS, este un limbaj interpretat de nivel inalt care este conform standardului ECMAScript. Chiar daca acest limbaj are anumite caracteristici ale programarii orientata obiect, nu este considerat un limbaj orientat obiect. Acest lucru se datoreaza faptului ca nu respecta in totalitate paradigma programarii orientata obiect. Spre deosebire de Java si C# care sunt special orientate obiect, JavaScript poate fi folosit si pentru programare procedurala, asemenea limbajului C++. De asemenea, JS are o diferenta majora fata de limbajele mentionate anterior, prind faptul ca nu impune tipul variabilelor, putand astfel sa se modifice structura si tipul de data a unei variabile in timpul rularii programului. Acest lucru aduce atat beneficii cat si dezavantaje, fiind un limbaj mai maleabil, dar mai greu de depanat atunci cand apar erori.

Desi JavaScript este folosit in principal pentru aplicatii de tip client, acesta poate fi folosit si pentru aplicatii 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 afisarea in browser. Browserele web interpreteaza documentele de tip HTMLsi apoi le afiseaza, impreuna cu documentele de tip CSS si JS, daca acestea exista si sunt folosite in interiorul documentelor HTML. In aceasta combinatie, fiecare limbaj are rolul sau, HTML fiind scheletul, CSS fiind aspectul clientului, iar JS fiind modul in care functioneaza acesta.

Elementele de tip HTML reprezinta baza paginilor web, fara aceste elemente fiind imposibila alipirea componentelor de tip CSS si JavaScript la o interfata vizuala cu care sa poata interactiona utilizatorii.

HTML a aparut pe 24 Noiembrie, 1995, iar de atunci a primit multe imbunatatiri semnificative, trecand prin mai multe versiuni, versiunea curenta fiind HTML 5, aparuta pe 28 Octombrie, 2014 si continuand sa primeasca diverse imbunatatiri.

### 3.1.3) CSS

CSS sau Cascading Style Sheets nu este un limbaj de programare, ci un limbaj pentru stilizarea continutului unei pagini web. Acest limbaj este strans legat de HTML, legandu-se de tagurile specifice. Se poate lega de tagurile HTML prin mai multe tipuri de selectori: selectori de clasa, selectori de element si selectori de id. De asemenea, se pot insera stilurile CSS direct in tagurile HTML, sau pot fi importate dintr-un fisier extern.

CSS a fost construit atat pentru permiterea asezarii continutului in pagina web, cat si pentru stilizarea continutului. Datorita acestor specificatii, limbajul CSS ajuta la imbunatatirea vizibilitatii si accesibilitatii continutului unei pagini incarcate in browser.

Termenul de “Cascading” se refera la modul in care functioneaza prioritatea aplicarii stilurilor in pagina. Acest lucru este important deoarece un programator fara experienta poate aplica din greseala acelasi stil pentru mai multe taguri din continutul unui website fara sa doreasca acest lucru.

### 3.1.4) TypeScript

Microsoft a creat TypeScript, un limbaj de programare de nivel inalt open source, cu scopul de a repara problemele des intampinate in JavaScript. Acest limbaj este un superset de JavaScript in care se pot impune tipul variabilelor, acesta neputand fi schimbat in timpul rularii programului. La momentul rularii programului, mai intai sunt verificate regulile impuse de TypeScript, iar apoi interpretatorul converteste codul in JavaScript.

### 3.1.5) Angular framework

Angular este un framework open-source baza pe TypeScript care este sustinut si intretinut de catre Google, impreuna cu o comunitate larga de oameni si companii. Acest framework este cunoscut si sub numele de Angular 2+ si a fost dezvoltat de catre o echipa de la Google, aceeasi echipa care a lucrat si la AngularJS. Aceasta unealta este utilizata pentru crearea de pagini web, permitand programatorilor sa dezvolte aplicatii intr-un mod mai rapid si mai curat, avand multe functionalitati deja implementate, fiind nevoie doar de adaptarea acestora in functie de nevoile aplicatiei.

***Diferentele dintre Angular si AngularJS***

Desi au aproape acelasi nume si ambele framework-uri sunt folosite pentru scrierea mai rapida a aplicatiilor, Angular si AngularJS sunt doua framework-uri scrise total diferit. AngularJS a aparut primul, pe data de 20 octombrie 2010, si a ramas fara suport tehnic din ianuarie 2022, iar Angular a aparut pe data de 14 septembrie 2016 si continua sa aibe suport tehnic pana in prezent, acesta din urma fiind si cel care continua sa fie utilizat de catre dezvoltatorii aplicatiilor web. De asemenea, pe langa diferentele semnificative de arhitectura, AngularJS este bazat pe JavaScript, de unde si prefixul JS, pe cand Angular este construit pe baza limbajului TypeScript.

In 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 inalt, orientat obiect care a fost proiectat avand cat mai putine dependente. Este un limbaj de programare de uz general care functioneaza dupa conceptul “scrie o data, ruleaza oriunde” (“write once, run anywhere”), ceea ce inseamna ca odata compilat codul Java, acesta poate rula pe orice masina care suporta platforma mentionata, fara a mai trebui recompilat codul. Aplicatiile Java sunt compilate in bytecode care poate rula pe orice sistem care detine o masina virtuala Java (Java virtual machine – JVM), indiferent de arhitectura sistemului. Asemenea limbajului C#, sintaxa Java seamana mult cu C/C++, avand mai putine facilitati low-level decat acestea, C/C++ avand oferind o mare flexibilitate cand vine vorba de alocarea memoriei, pe cand C# si Java folosesc un “garbage collector” care se ocupa cu dezalocarea memoriei atunci cand se pierde referinta unei zone de memorie.

***Utilizari***

Limbajul Java a fost folosit pentru foarte multe tipuri de aplicatii de-a lungul timpului, in prezent fiind predominant folosit pentru scrierea aplicatiilor pentru Android si pentru aplicatiile de tip back-end.

***Istoric***

Java a fost creat pe data de 19 mai 1995 de catre James Gosling, un programator al companiei Sun Microsystems. Atat librariile, cat si JVM si compilatorul limbajului Java au fost initial construite de catre Sun Microsystems. In prezent, Java 20 este ultima versiune, aparuta pe data de 21 martie 2023, Java 17 fiind ultima versiune stabila ce aduce schimbari semnificative limbajului, aparuta pe data de 14 septembrie 2021.

### 3.2.2) Spring

Spring este un framework si un container pentru “inversion of control” pentru platforma Java. Principalele caracteristici ale Spring framework pot fi folosite de catre orice aplicatie de tip Java, existand si anumite extensii ale acestui framework pentru scrierea de aplicatii web.

Una dintre cele mai importante functionalitati Spring este “inversion of control container”, care ofera o modalitate foarte eficienta a configurarii si coordonarii obiectelor Java. Containerul este responsabil de gestionarea obiectelor, mai exact de crearea, utilizarea si configurarea acestora, lenagadu-le impreuna. In trecut, se configurau aceste obiecte denumite “beans” prin fisiere de tip XML (Extensible Markup Language), dar in prezent se folosesc adnotari.

Spring a aparut in anul 2003, prima versiune fiind 0.9. In prezent ultima versiune este 6.0, aparuta pe data de 16 noiembrie 2022.

### 3.2.3) Spring Boot

Spring Boot framework este o extensie a framework-ului Spring, aducand imbunatatiri semnificative atunci cand vine vorba de complexitatea necesara pentru configurarea unei aplicatii de tip Spring. Acesta este folosit in mod special pentru crearea aplicatiilor de tip web precum: MVC, RESTful API sau microservicii. Exista 3 capabilitati importante prin care Spring Boot usureaza munca programatorului:

***Autoconfigurare***

Autoconfigurarea se refera la faptul ca aplicatiile de tip Spring Boot vin cu dependente deja initializate pe care dezvoltatorul software nu mai trebuie sa le configureze manual. Acest fapt ajuta nu doar la grabirea procesului de dezvoltare a aplicatiilor, ci si la aplicarea bunelor practici care duc la construirea unui cod mai usor de citit si de intretinut. De asemenea, aceste setari implicite pot fi schimbate la nevoie.

***Abordarea de opinie (“Opinionated approach”)***

Pe baza cerintelor proiectului, Spring Boot adopta o abordare de opinie pentru adaugarea si setarea dependentelor de pornire a aplicatiei. In loc ca programatorul sa ia toate deciziile si sa 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. De asemenea, 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 usor de citit si mai usor de depanat. Acesta este competitorul principal al framework-ului Angular, fiind cel mai utilizat framework pana in prezent.

### 3.3.2) Aplicatii pentru telefon

In ciuda faptului ca foarte multe aplicatii de E-commerce sunt construite sub forma unui website, exista tot atatea solutii software care in loc de a folosi un website pentru interfata cu care interactioneaza utilizatorul, folosesc o aplicatie nativa de telefon, inlocuind doar partea de “front-end”, putand astfel sa pastreze acelasi “back-end”. De asemenea, exista si optiunea de a utiliza ambele optiuni in paralel deoarece cuplarea de la client la server este una detasata, putand utiliza doua aplicatii de tip client care comunica cu acelasi back-end.

### 3.3.3) ASP.NET Core

Microsoft a creat ASP.NET Core, un framework open-source care inlocuieste ASP.NET. Este un framework modular care ruleaza atat pe .NET Framework pentru Windows, cat si pe arhitectura multiplatforma .NET. De la versiunea ASP.NET Core 3, framework-ul nu mai suporta .NET Framework, continuandu-se suportul doar pentru .NET Core.

Chiar daca acest framework este rescris cu totul de la fundatie, deoarece este proiectat pentru a imbina arhitecturile vechi ASP.NET Web API si ASP.NET MVC, prezinta un nivel ridicat de compatibilitate cu ASP.NET.

ASP.NET Core a aparut pe 27 mai 2016, Microsoft continuand sa imbunatateasca aceasta tehnologie, in prezent ajungand la versiunea 7.0, aparuta pe data de 8 noiembrie 2022.

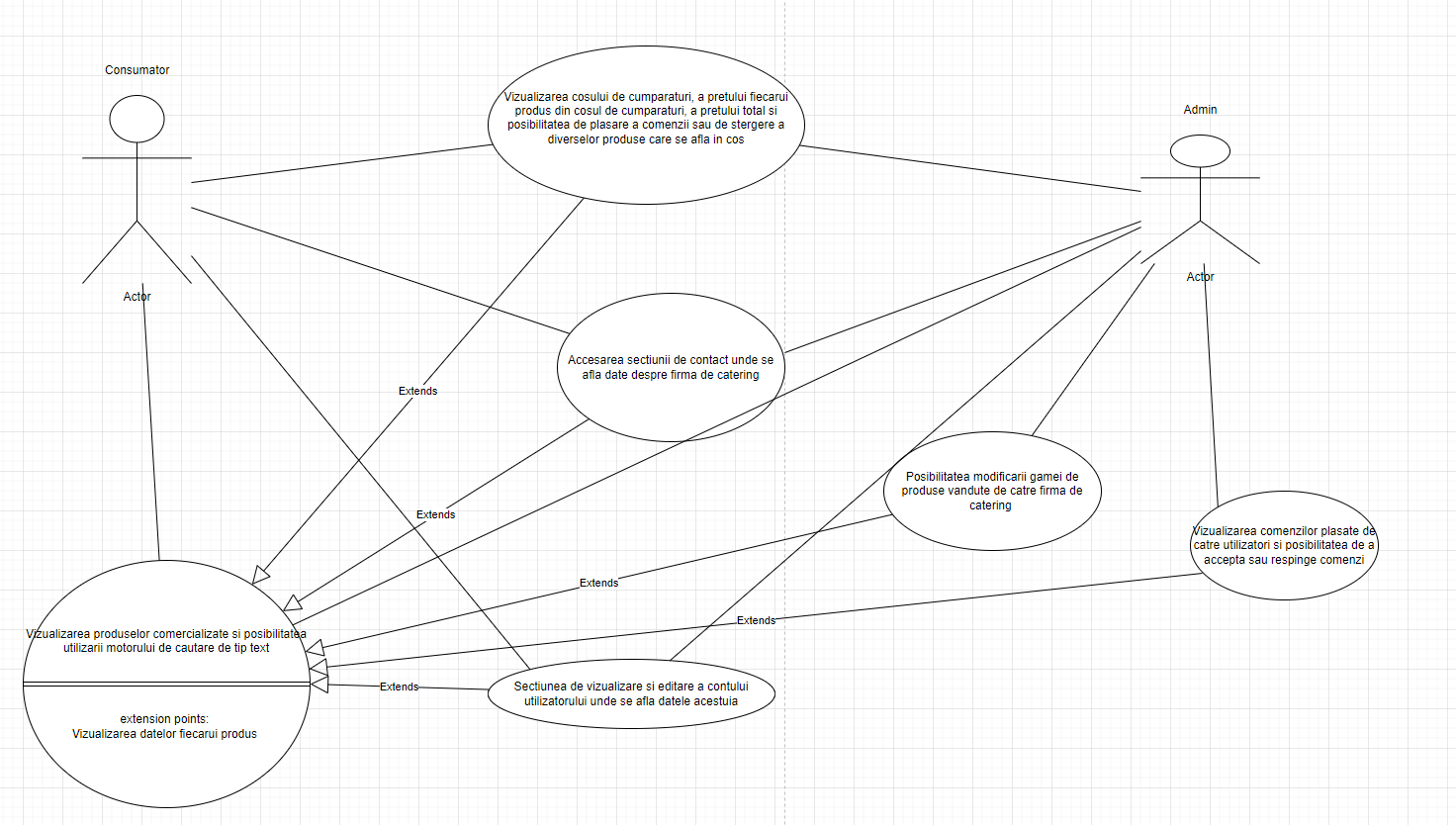
# Prezentarea solutiei informatice

## 4.1) Descrierea generala a sistemului informatic

## 4.2) Specificarea cerintelor sistemului informatic

**Diagrama cazurilor de utilizare**

In diagrama prezentata sunt expuse, in mod grafic, actorii, principalele functionalitati ale aplicatiei care vor fi implementate in decursul etapei de proiectare si relatiile dintre acestia. In cadrul aplicatiei “Pizza4you” se afla doi actori, unul fiind cel din categoria de admin, iar celalalt fiind din categoria de consumator final, deoarece aceasta se doreste a fii o aplicatie de comert electronic.



Putem observa din diagrama ca adminul, pe langa atributiile sale speciale, are de asemenea si toate drepturile pe care le are un consumator, Se poate observa faptul ca amandoi pot sa vizualizeze gama de produse comercializata de restaurantul “Pizza4you”, pot cauta produsele dorite prin motorul de cautare de tip text.

Din optiunea de vizualizare a cosului de cumparaturi, atat adminul cat si consumatorul pot sa stearga produse din cosul de cumparaturi, sa comande produsele respective si sa vada costul total al produselor comandate. De asemenea, ambele tipuri de utilizatori au acces la sectiunea de editare a contului personal unde se afla datele sensibile salvate pe cont, cum ar fi mail, nume si adresa personala.

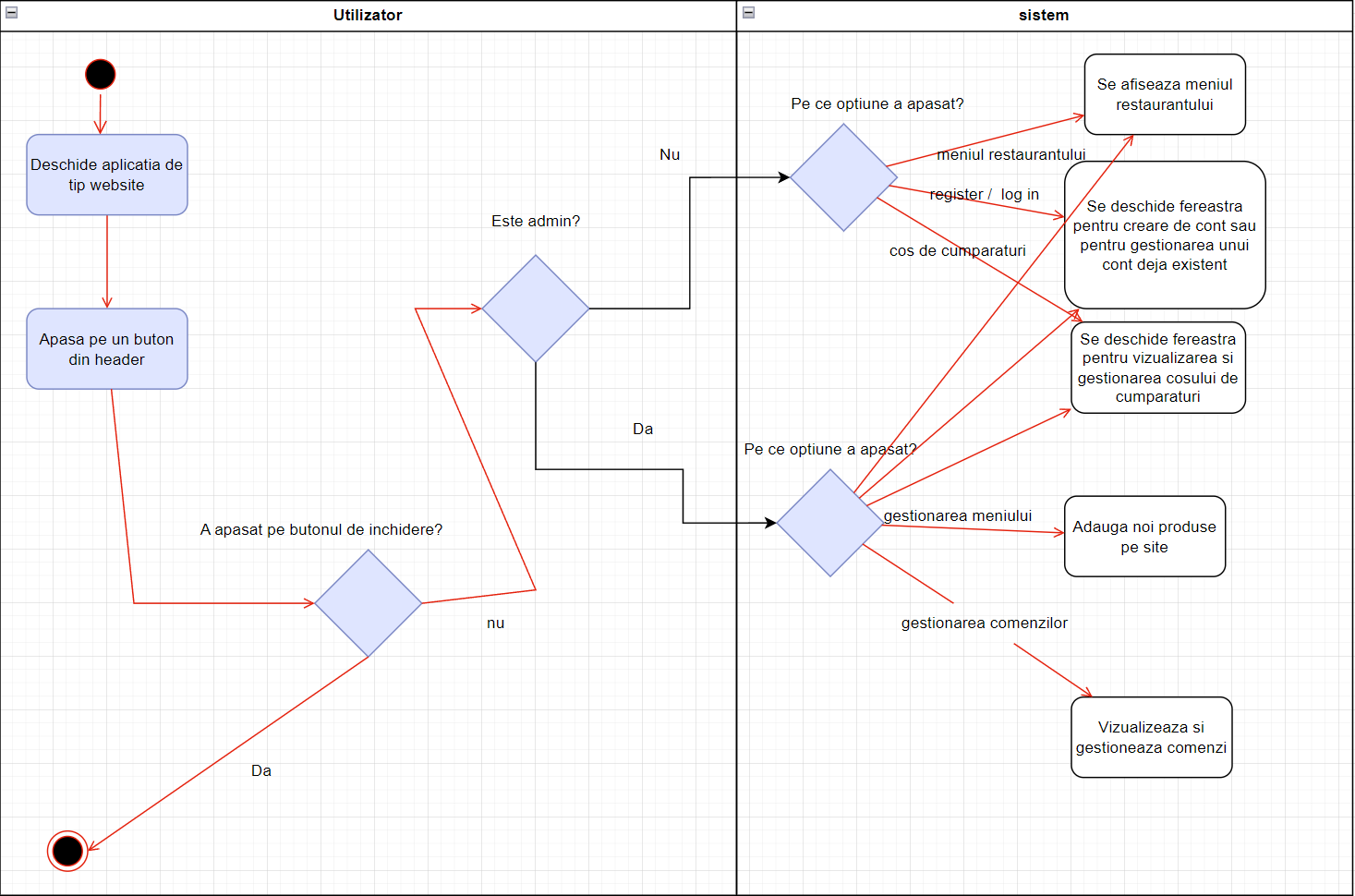
Spre deosebire de un consumator obisnuit, adminul are posibilitatea de a adauga sau sterge produse din gama restaurantului. De asemenea, acesta poate sa vizualizeze toate comenzile plasate de catre consumatori si are posibilitatea de a le verifica si confirma.

## 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 ințelege 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**

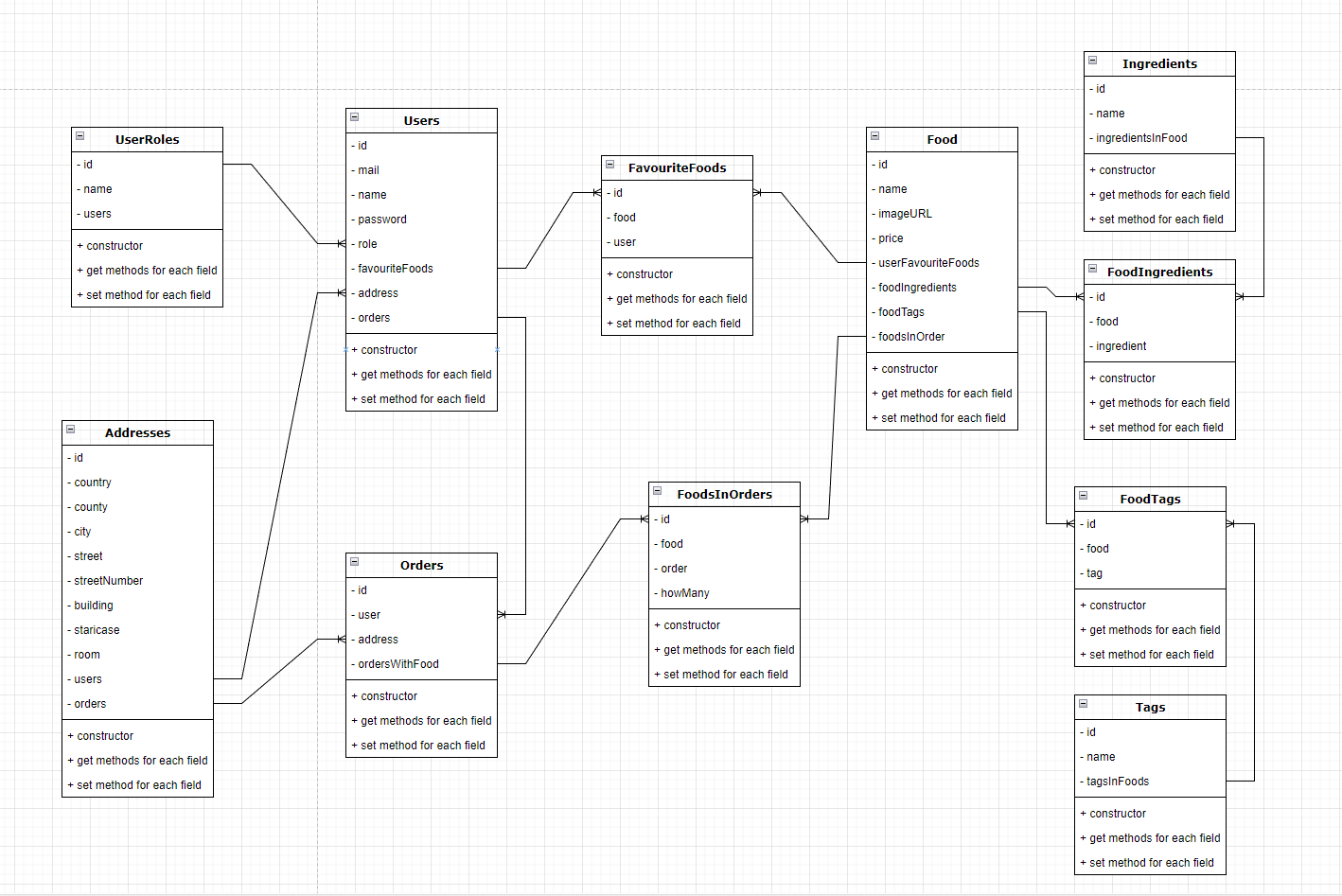
****

In diagrama de activitate prezentata ca fiind cea a headerului se poate observa faptul ca utilizatorul poate accesa setul de optiuni asa cum doreste odata ce a deschis pagina website-ului. Utilizatorul, are diverse posibilitati in functie de rolul acestuia. Daca este un utilizator obisnuit, acesta poate naviga prin paginile de resigster / login, de cos de cumparaturi sau de meniu al restaurantului. Utilizatorul de tip admin are in plus 2 optiuni de administrare.

## 4.4) Proiectarea sistemului informatic

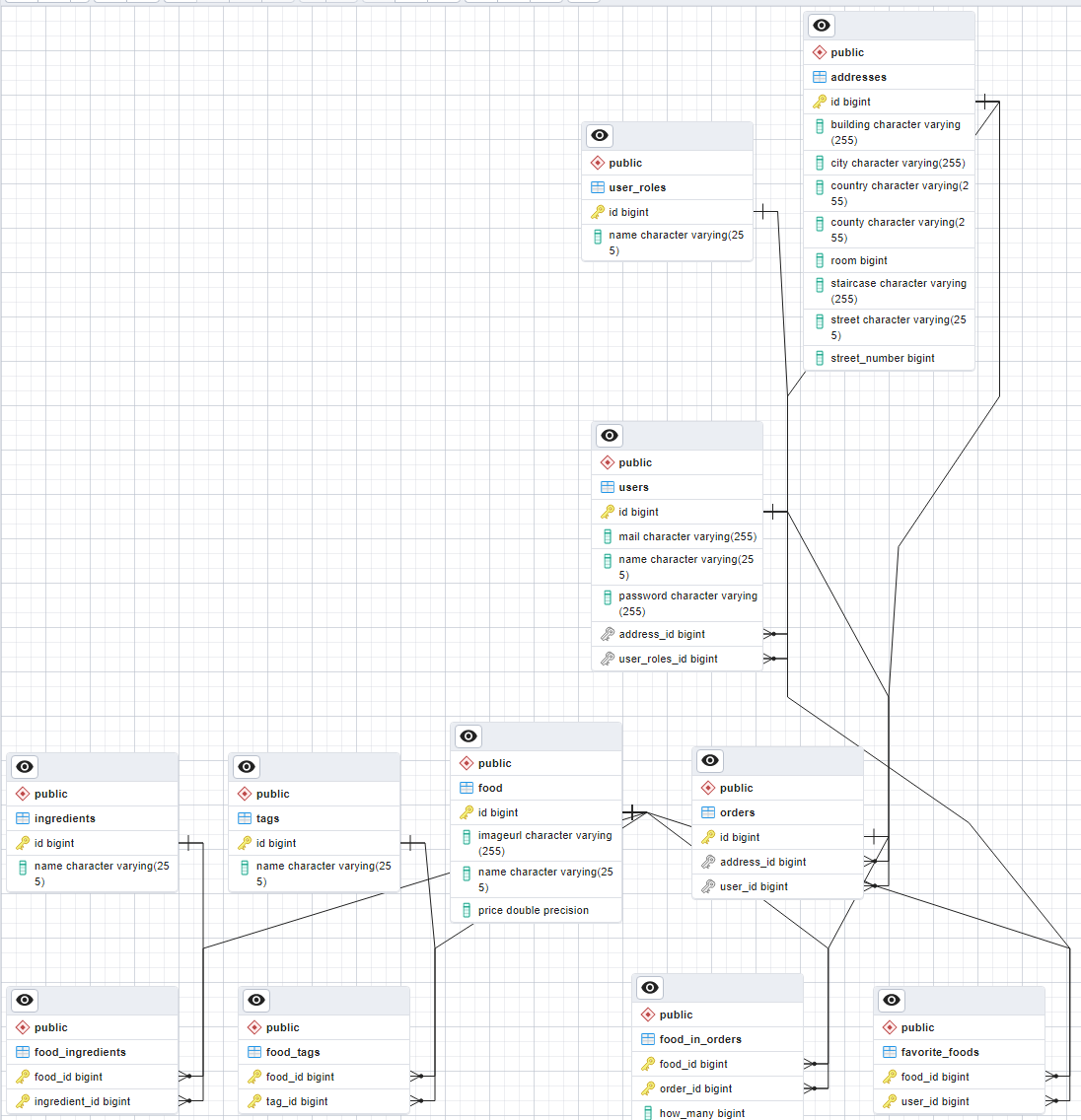
### 4.4.1) Diagrama de clase detaliata

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 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 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 contin defapt legaturi de one to many de la tabela de lagatura la cele doua tabele. De asemenea, se pot observa tipurile de date din fiecare tabela.

# Bibliografie

<https://en.wikipedia.org/wiki/JavaScript>

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

<https://en.wikipedia.org/wiki/HTML>

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

<https://en.wikipedia.org/wiki/CSS>

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

<https://en.wikipedia.org/wiki/TypeScript>

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

<https://en.wikipedia.org/wiki/Angular_(web_framework)>

<https://angular.io/docs>

<https://en.wikipedia.org/wiki/Java_(programming_language)>

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

<https://en.wikipedia.org/wiki/Spring_Framework>

<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://en.wikipedia.org/wiki/PostgreSQL

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

<https://react.dev/>

<https://en.wikipedia.org/wiki/React_(software)>

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

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

<https://en.wikipedia.org/wiki/ASP.NET_Core>