

**Aplicatie de tip RESTful API - Client**

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

**Student:** Sicu Bogdan Andrei

Cuprins

[1) Introducere 2](#_Toc132900228)

[2) Prezentatrea domeniului 3](#_Toc132900229)

[2.1) Istoria firmelor de catering 4](#_Toc132900230)

[2.2) Istoria internetului 4](#_Toc132900231)

[2.3) Istoria comertului electronic 4](#_Toc132900232)

[2.4) Alte firme de catering 4](#_Toc132900233)

[3) Prezentartea tehnologiilor 5](#_Toc132900234)

[3.1) Front-end 5](#_Toc132900235)

[3.1.1) JavaScript 5](#_Toc132900236)

[3.1.2) HTML 5](#_Toc132900237)

[3.1.3) CSS 6](#_Toc132900238)

[3.1.4) TypeScript 6](#_Toc132900239)

[3.1.5) Angular framework 6](#_Toc132900240)

[3.2) Back-end 6](#_Toc132900241)

[3.2.1) Java 6](#_Toc132900242)

[3.2.2) Spring framework 6](#_Toc132900243)

[3.2.3) Spring Boot framework 6](#_Toc132900244)

[3.3) Alte tehnologii folosite in domeniul aplicatiilor de catering/E-commerce 6](#_Toc132900245)

[4) Prezentarea solutiei informatice 6](#_Toc132900246)

[4.1) Descrierea generala a sistemului informatic 6](#_Toc132900247)

[4.2) Specificarea cerintelor sistemului informatic 6](#_Toc132900248)

[4.3) Analiza sistemului informatic 8](#_Toc132900249)

[4.3.1) Diagrame de activitate 8](#_Toc132900250)

[4.4) Proiectarea sistemului informatic 9](#_Toc132900251)

[4.4.1) Diagrama de clase detaliata 9](#_Toc132900252)

[4.4.2) Proiectarea bazei de date 10](#_Toc132900253)

[Bibliografie 12](#_Toc132900254)

# Introducere

Sistemul este reprezentat de doua programe care comunica unul cu celalalt pentru a forma o aplicatie de comert pe internet indreptate 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 in 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 un numar de telefon pentru contact, o 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 cu 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. Aceste 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, despre acestea din urma vom detalia 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 ce limbajele mentionate anterior, acesta fiind 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

## 3.2) Back-end

### 3.2.1) Java

### 3.2.2) Spring framework

### 3.2.3) Spring Boot framework

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

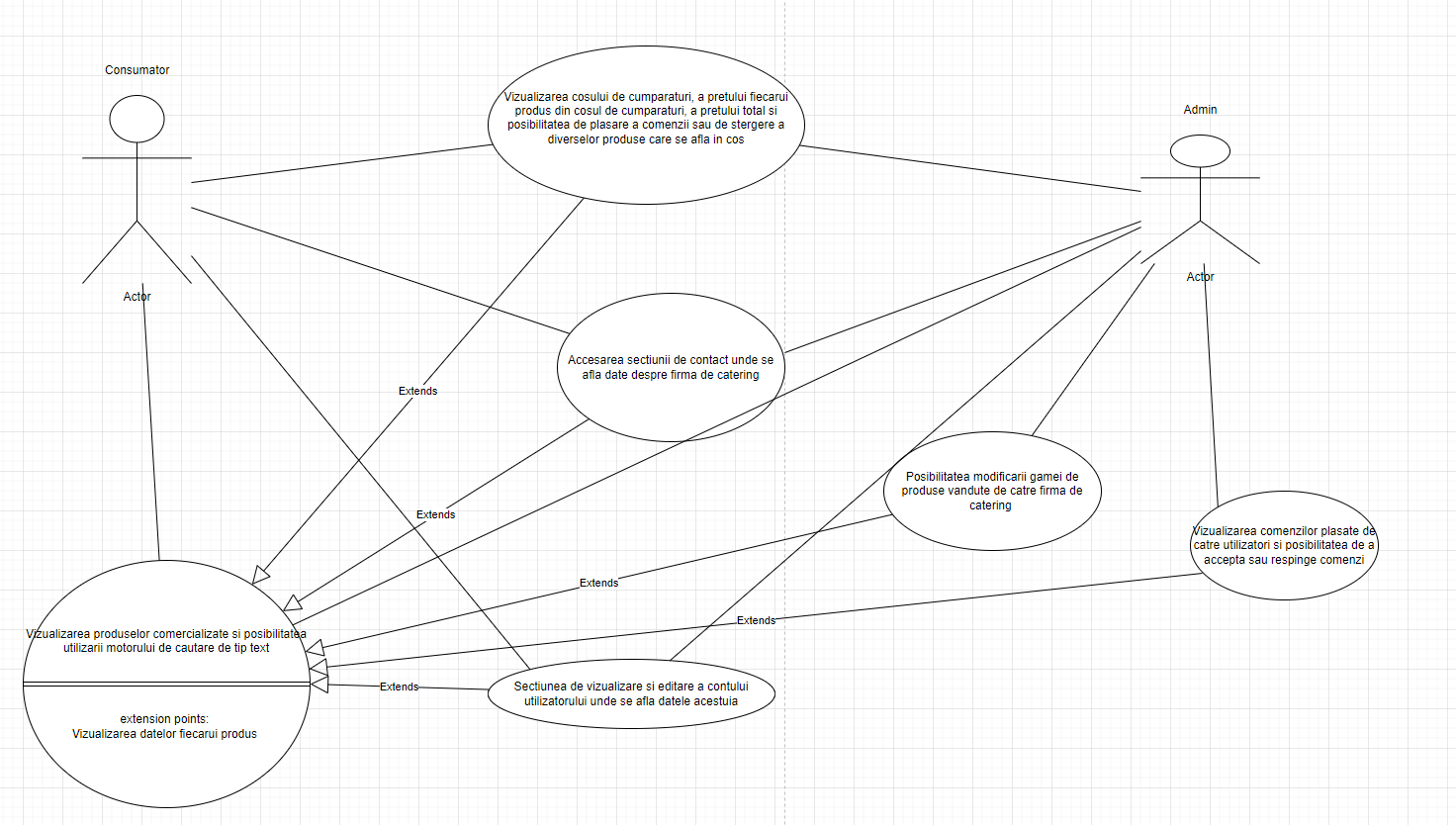
# Prezentarea solutiei informatice

## 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, acrotii, 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.

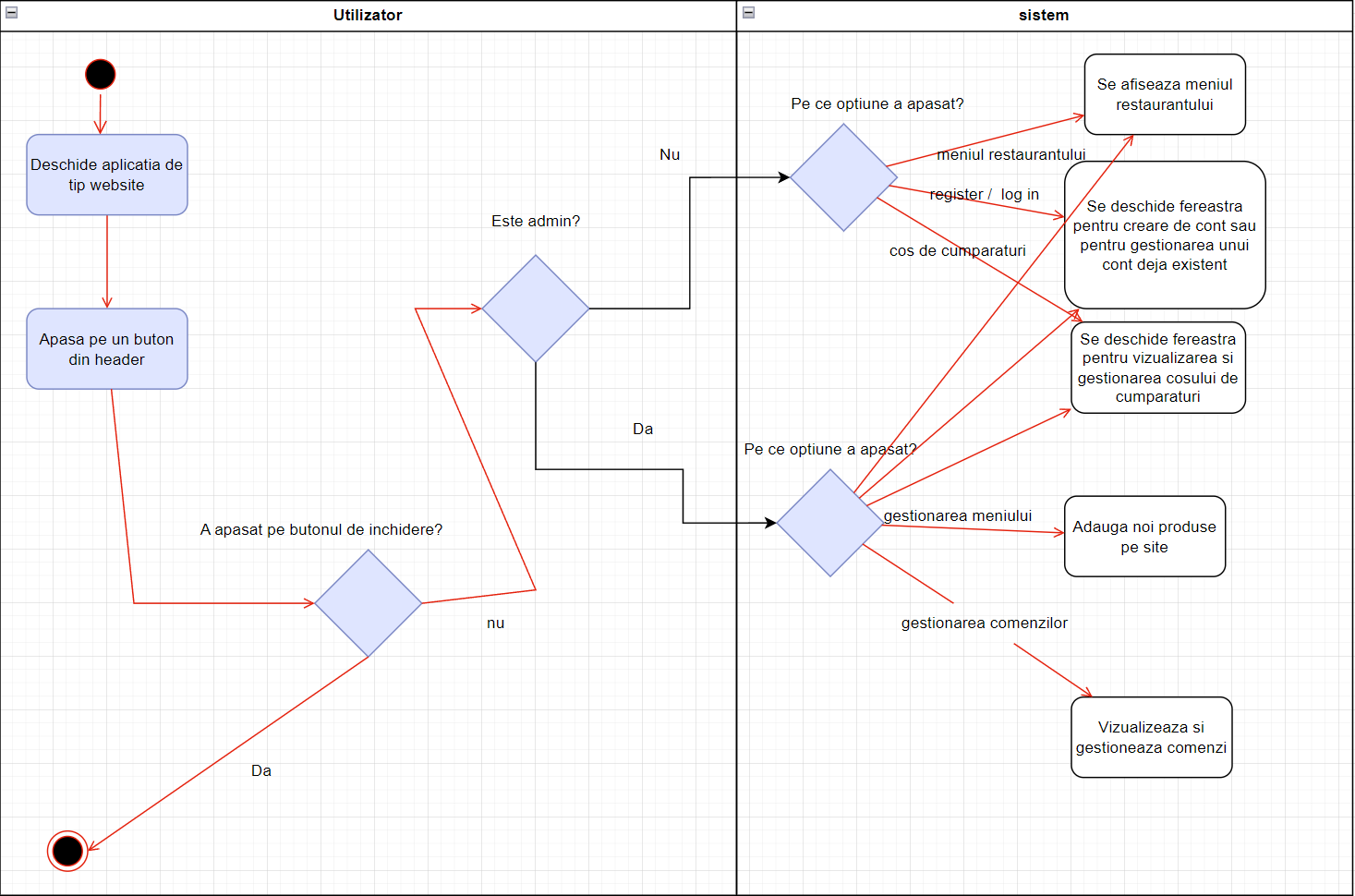
Spre deosebire de un consumator obisnuit, adminul are posibilitatea de a adauga sau sterge produse din gama restaurantului. De asemenea, acesta poate sa vizualizeze toate comenzile plasate de catre consumatori si are posibilitatea de a le verifica si confirma.

## 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**

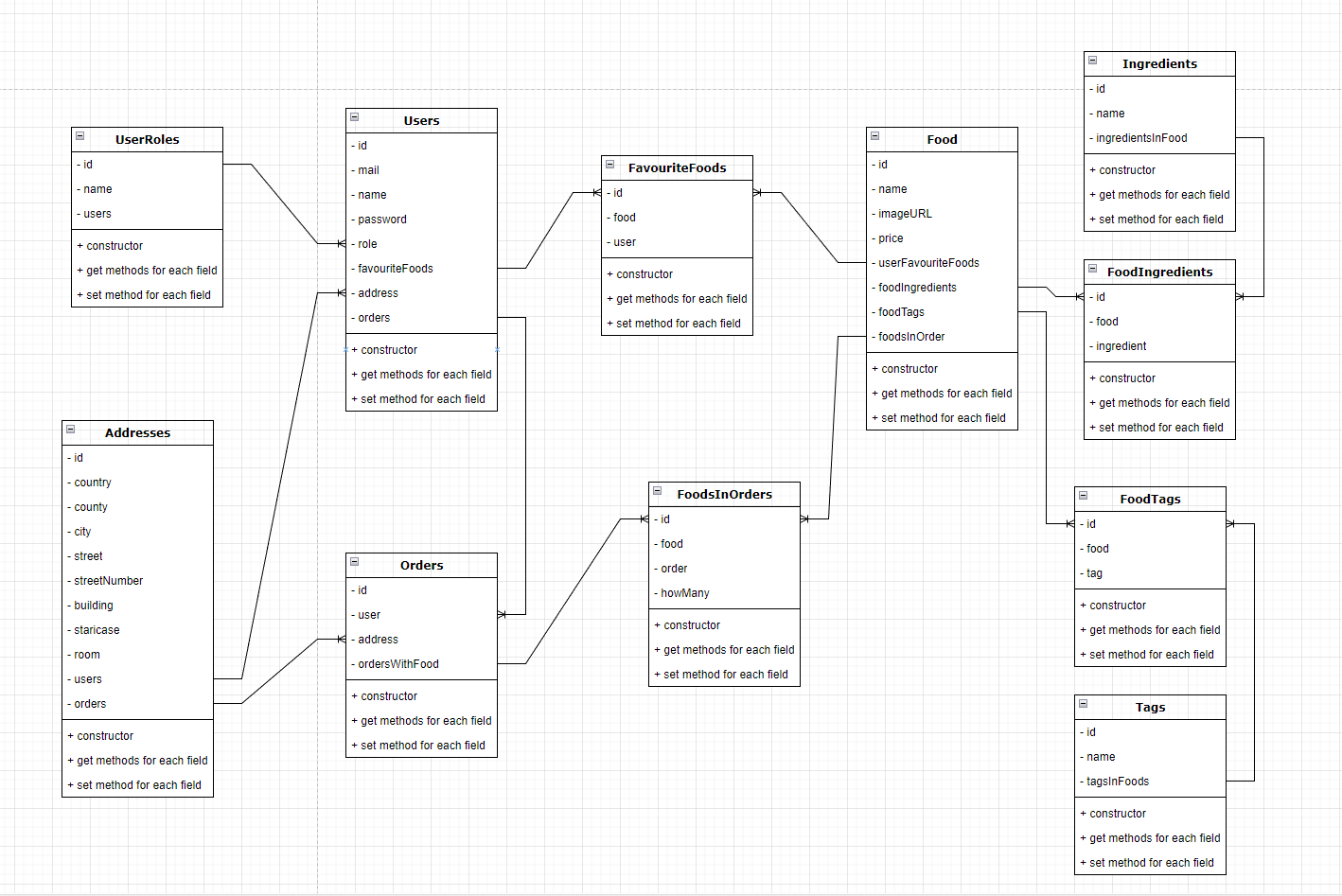
****

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

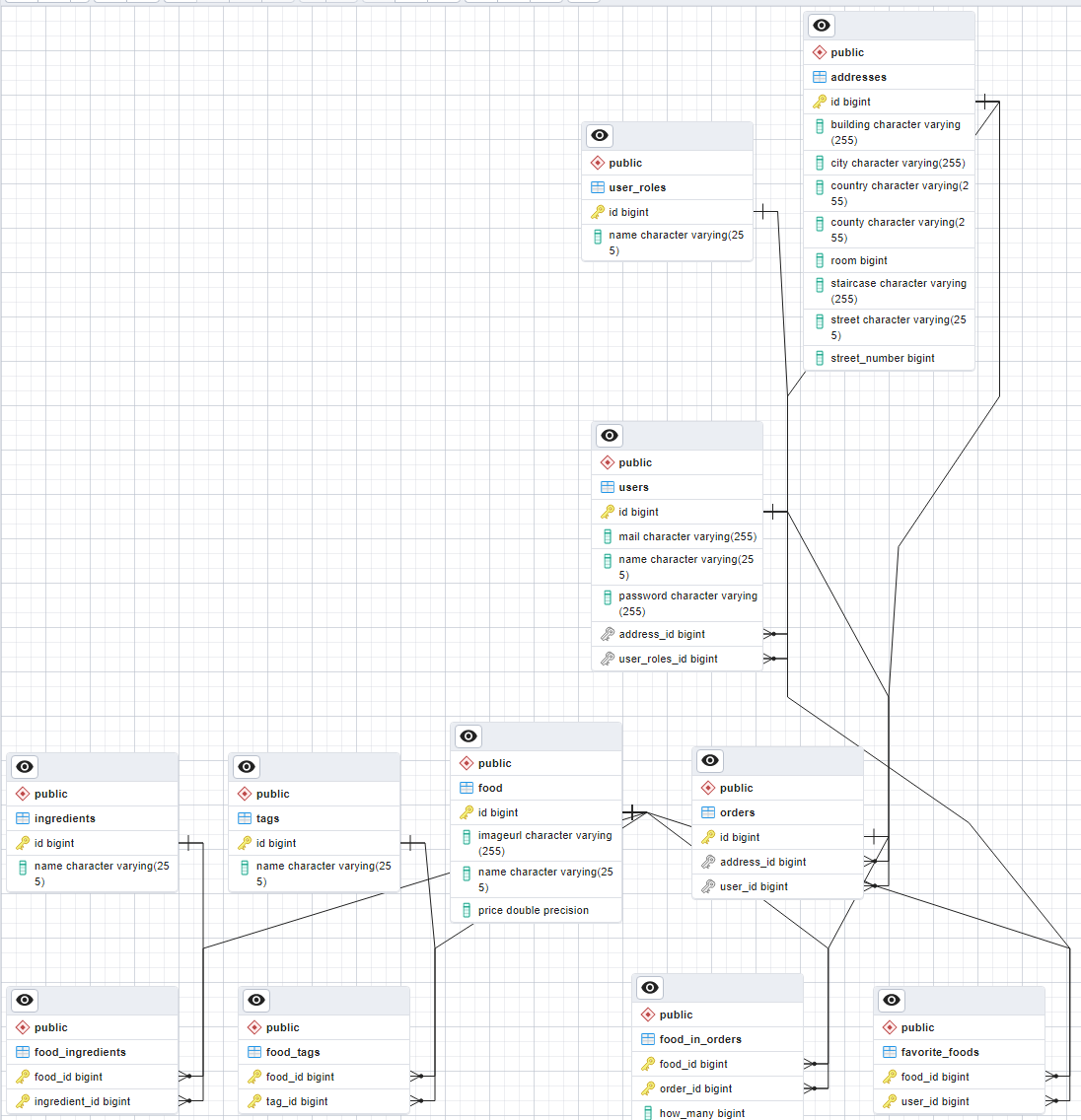
### 4.4.1) Diagrama de clase detaliata

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>