

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI  
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

**Piece of Cake**

propusă de

***Camelia Băcăoanu***

**Sesiunea:** *iulie, 2019*

Coordonator științific  
**Prof. Colab. Florin Olariu**

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI  
FACULTATEA DE INFORMATICĂ

# **Piece of Cake**

***Camelia Băcăoanu***

**Sesiunea:** *iulie, 2019*

Coordonator științific  
***Prof. Colab. Florin Olariu***

Avizat,

Îndrumător Lucrare de Licență

Titlul,

Numele

și

prenumele

Data \_\_\_\_\_

Semnătura \_\_\_\_\_

### **DECLARAȚIE privind originalitatea conținutului lucrării de licență**

Subsemnatul(a) .....

domiciliul în .....

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

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

..... specializarea ....., promoția

....., declar pe propria răspundere, cunoscând consecințele falsului în

declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr.

1/2011 art.143 al. 4 și 5 referitoare la plagiat, că lucrarea de licență cu titlul:

\_\_\_\_\_

\_\_\_\_\_elaborată sub îndrumarea dl. / d-na

\_\_\_\_\_, pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

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

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

## DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Titlul complet al lucrării*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, *data*

Absolvent *Prenume Nume*

---

(semnătura în original)

# Cuprins

<b>1. Introducere .....</b>	<b>7</b>
1.1. Avantaje ale cumpărăturilor online .....	7
1.2. Dezavantaje ale cumpărăturilor online.....	7
1.3. Motivație .....	9
1.4. Structura lucrării.....	10
<b>2. Contribuții .....</b>	<b>11</b>
<b>3. CAPITOLUL I: Descrierea problemei .....</b>	<b>12</b>
<b>4. CAPITOLUL II: Aplicații similare .....</b>	<b>13</b>
<b>5. CAPITOLUL III: Tehnologii utilizate .....</b>	<b>14</b>
5.1. Node.js.....	14
5.2. Express.js.....	14
5.3. Typegoose .....	14
5.4. MongoDB.....	14
<b>6. CAPITOLUL IV: Structura aplicației.....</b>	<b>15</b>
6.1. Modele.....	15
6.2. Configurare.....	15
6.3. Arhive.....	16
6.4. Controllers .....	17
6.5. Static.....	17
6.6. Structura bazei de date .....	18

<b>7. CAPITOLUL V: Interacțiunea utilizatorului cu aplicația .....</b>	<b>21</b>
7.1. Utilizator de tip client.....	21
7.1.1. Neautentificat.....	21
7.1.2. Inregistrare.....	22
7.1.3. Autentificare .....	23
7.1.4. Adăugarea unui comentariu.....	24
7.1.5. Adăugarea la favorite.....	25
7.1.6. Adăugarea în coșul de cumpărături .....	26
7.1.7. Comandă personalizată.....	27
7.1.8. Trimiterea comenzii.....	28
7.2. Utilizator de tip administrator .....	29
7.2.1. Adăugare produs.....	29
7.2.2. Preluarea comenzii .....	30
<b>8. Concluzii .....</b>	<b>31</b>
<b>9. Bibliografie.....</b>	<b>32</b>

## Introducere:

Cumpărăturile online se referă la acțiunea de a căuta și achiziționa un bun sau serviciu prin intermediul Internetului.

Numărul persoanelor care preferă să își facă cumpărături online este în continuă creștere, în principal datorită faptului că oamenii consideră mai convenabil și mai ușor să caute și să achiziționeze ceea ce doresc din confortul casei lor.

## Avantaje ale cumpărăturilor online:

- **Flexibilitatea.** Majoritatea magazinelor obișnuite, așa cum le cunoaștem, au o oră de închidere și de deschidere și de asemenea sunt închise de sărbători. În magazinele online, însă, îți poți face cumpărăturile oricând, indiferent de ora sau zi. Alegi produsele dorite și plasezi comanda de oriunde ai fi: de la birou, de acasă sau din mașină, atât timp cât există Internet.
- **Accesibilitate.** În general același produs poate fi vândut de mai multe magazine dar cu preț diferit. Există posibilitatea consultării mai multor magazine înainte de a achiziționa produsul sau accesarea unui site care îți afișează toate magazinele care vând produsul dorit pentru al alege pe acela cu prețul cel mai bun.
- **Internațional.** Atunci când achiziționezi un produs prin intermediul Internetului, nu ești limitat de bariere spațiale. Poți cumpăra de oriunde în lume.

## Dezavantaje ale cumpărăturilor online:

- **Nu există posibilitatea de a proba.** Valabil în principal pentru articolele de vestimentație. Chiar dacă magazinele online oferă descrieri și tabele de mărimi există întotdeauna posibilitatea ca mărimile să difere sau ca produsul să arate altfel în realitate decât în poze.
- **Livrare cu întârziere.** Majoritatea magazinelor online pretind că oferă livrare în aproximativ 24h. Totuși, timpul de livrare depinde în primul rând de firma de curierat. În

special în perioada sărbătorilor sau a reducerilor timpul de livrare poate ajunge și la o săptămână, iar dacă comanda este din altă țară, poate ajunge la câteva luni.

- **Suprasolicitate.** În perioada reducerilor magazinele online pot fi accesate de prea mulți oameni și aceasta să ducă la picarea serverelor.
- **Produsele pot ajunge degradate.** Acest dezavantaj nu mai este o responsabilitate a magazinul online de la care s-a comandat. După predarea produselor, acest aspect depinde de firma de curierat.

Comerțul online în România a avut un start timid și întârziat față de celelalte țări europene. Aceasta început greoi se datorează în mare parte și faptului că românii au avut acces la internet greu. Chiar dacă multe firme au încercat o abordare online încă înainte de 2010, populația țintă era încă foarte mică în România. În țări precum Germania, Danemarca sau Suedia, în anul 2018, peste 70% din persoanele cu acces la internet făceau cumpărături online în timp ce în România doar 20% din utilizatorii de internet au apelat cel puțin o dată la comerțul online pe parcursul a 12 luni.

Însă, de la an la an, comerțul online în România câștigă din ce în ce mai multă popularitate printre români.

În anul 2018 s-a înregistrat o valoare a cumpărăturilor online făcute de români de aproximativ 3,5 miliarde de euro, cu 30% mai mult decât în anul 2017, când valoarea a fost de 2,8 miliarde de euro. De asemenea, din punct de vedere al numărului de magazine online, acesta s-a dublat de la aproximativ 7.000 de magazine în anul 2017 la peste 15.000 de comercianți în 2018. Creșterea este spectaculoasă având în vedere faptul că în 2016 erau aproximativ 5.000 de magazine online.

O ramură a comerțului online care este la început de drum în România este cea alimentară. Fie că este vorba de restaurante, patiserii sau fast-food, atât comercianții cât și o mare parte dintre clienți evită să facă pasul spre mediul virtual.

În orașele mari, numărul oamenilor care preferă să apeleze la un restaurant online sau o platformă care le permite să comande de la mai multe restaurante este într-o continuă creștere. Utilizatorii au la dispoziție meniul online și își pot alocă oricât timp de gândire înainte de a plasa comanda. Clienții preferă să apeleze la spațiul online deoarece aglomerația și locurile limitate dintr-un restaurant obișnuit reprezintă adesea o problemă pentru persoanele care preferă



un mediu mai retras și liniștit. De asemenea, timpul economisit și faptul că un client nu este nevoit să se oprească din activitățile curente este un mare plus.

Platformele care permit comenzile de mâncare de la restaurante multiple, în special cele mobile, sunt foarte apreciate, de atât de clienții care sunt mulțumiți că au la dispoziție, într-o singură aplicație, mai multe meniuri, cât și de restaurantele partenere, deoarece vânzările lor cresc considerabil.

### **Motivație:**

„Piece of Cake” este o aplicație web de tip magazin online, care permite unui utilizator care îndeplinește funcția de administrator să pună la dispoziția unui utilizator de tip client produse variate, iar acesta din urmă, la rândul său, le poate comanda sau poate plasa o comandă personalizată.

Alegerea ca temă a lucrării de licență a realizării unei aplicații web, care are rol de patiserie online, a fost o alegere bazată pe pasiunea pentru gătit, pe care am dezvoltat-o pe parcursul anilor.

Având o pasiune pentru gătit încă din tinerețe și urmărind diverse canale de gătit, am putut realiza importanța rețelelor media la începutul unei afaceri și mai ales cât de mult poate ajuta propria aplicație pentru dezvoltarea acesteia.

Am putut observa totuși că în România acest lucru este trecut cu vederea, că foarte puțini oameni își duc micile afaceri din domeniul culinar în mediu virtual. Foarte puțini dintre cei ce decid să își promoveze afacerea online nu trec mai departe de rețelele de socializare precum „Facebook” sau „Instagram”. Această abordare este una care împiedică creșterea afacerii, întrucât oamenii au nevoie de un meniu ușor accesibil, concret și, de asemenea, au nevoie de posibilitatea realizării unei comenzi rapide.

**Structura lucrării:**

Pentru o înțelegere mai bună a proiectului realizat și a structurii acestuia, lucrarea în cauză va fi descrisă pe parcursul a cinci capitole care au ca scop surprinderea cât mai exactă a elementelor aplicației și a folosirii acesteia.

**CAPITOLUL I: Descrierea problemei** – o descriere sumară a problemei ce se caută a fi rezolvată prin aplicația „Piece of Cake”.

**CAPITOLUL II: Aplicații similare** – trecerea în revistă a unor aplicații similare celei realizate, diferențele în comparație cu aplicația „Piece of Cake” și cine deține aceste aplicații.

**CAPITOLUL III: Tehnologii utilizate** – conține descrierea sumară a tehnologiilor folosite în vederea realizării aplicației „Piece of Cake” aceste tehnologii fiind Node.js, framework-ul Express.js, librăria Typegoose și baza de date MongoDB.

**CAPITOLUL IV: Structura aplicației** – conține descrierea detaliată a arhitecturii aplicației, cum este structurată partea de back-end și ce conține baza de date.

**CAPITOLUL V: Interacțiunea utilizatorului cu aplicația** – în acest capitol va fi prezentat cum poate un utilizator să interacționeze cu aplicația, atât pentru un administrator al aplicației cât și pentru un utilizator de tip client.

**Contribuții:**

Unul dintre cele mai importante aspecte ale unei aplicații web este impactul vizual. Este mai probabil ca un client să renunțe la folosirea unei aplicații deoarece nu este mulțumit de aspectul acesteia, indiferent dacă este mulțumit sau nu de serviciile oferite. În realizarea aplicației „Piece of Cake” am încercat să implementez o interfață prietenoasă, cât mai atractivă și ușor de folosit.

Secțiunea de produse favorite și posibilitatea de a căuta în aplicație sunt cele ce îi permit unui client accesul rapid la produse, acesta reprezentând economie de timp pentru un client care știe ce vrea.

Existența unui coș de cumpărături este benefică atât pentru comerciant cât și pentru client. Comanda este clară și plasată în siguranță, posibilitatea ca aceasta să nu fie onorată după cerințele scrise ale clientului fiind foarte mică.

Prin sistemul de monitorizare al comenzilor, un client poate verifica stadiul în care se află comanda sa și poate semnala comerciantul dacă identifică o problemă.

Personalizarea unei comenzi permite unui client să dețină controlul atât asupra bugetului alocat pentru un tort, cât și asupra elementelor pe care acesta le conține. Clienții pot alege cum arată tortul atât la interior cât și la exterior, lăsându-și imaginația să creeze un tort pe placul lor.

## **Capitolul I. Descrierea problemei:**

În zonele rurale și în orașele mici, piața alimentară în mediul online este foarte slabă. Deși există restaurante, patiserii sau mici comercianți care au propria lor afacere, acestea rămân la nivel local din cauza faptului că afacerea lor nu este promovată într-un mod eficient.

Mulți comercianți preferă să își promoveze produsele și afacerile prin intermediul rețelelor de socializare, așteptând să fie contactați telefonic, fără a avea o aplicație care să prezinte meniul și să le permită clienților să comande online.

Consider că o aplicația web, care ar pune meniul la dispoziția clientului într-un mod atractiv și ar permite comenzile online, ar ajuta la creșterea și popularitatea acestui domeniu.

Folosită în zonele rurale sau a orașelor mici, permițându-i comerciantului să se dezvolte și să acumuleze un câștig mai mare de pe urma afacerii sale, o aplicație web ar fi soluția pentru problema absenței restaurantelor online din zona respectivă.

## Capitolul II. Aplicații similare:

Există o multitudine de aplicații web, de tip restaurant online, care permit clienților să consulte meniul și să plaseze comenzi online. Însă foarte puține dintre acestea permit utilizatorilor să își personalizeze comanda.

Un exemplu de o astfel de aplicație web este cea deținută de „Pizza Hut”. Clienții au posibilitatea să aleagă dintr-un meniu variat sau opta pentru crearea propriei pizza.

În cazul variantei din urmă, utilizatorul trebuie să aleagă dimensiunea blatului, tipul acestuia, topping-ul dorit și la sfârșit ce cantitate din produsul anterior creat se dorește a fi comandată.

Diferențele principale dintre aplicația celor de la „Pizza Hut” și aplicația „Piece of Cake” sunt constituite de domeniu și de sistemul de monitorizare. În timp ce aplicația celor de la „Pizza Hut” este destinată domeniului restaurantelor de tip fast-food, „Piece of Cake” este destinată patiseriilor. De asemenea, aplicația restaurantului nu deține un sistem de monitorizare al comenzilor.

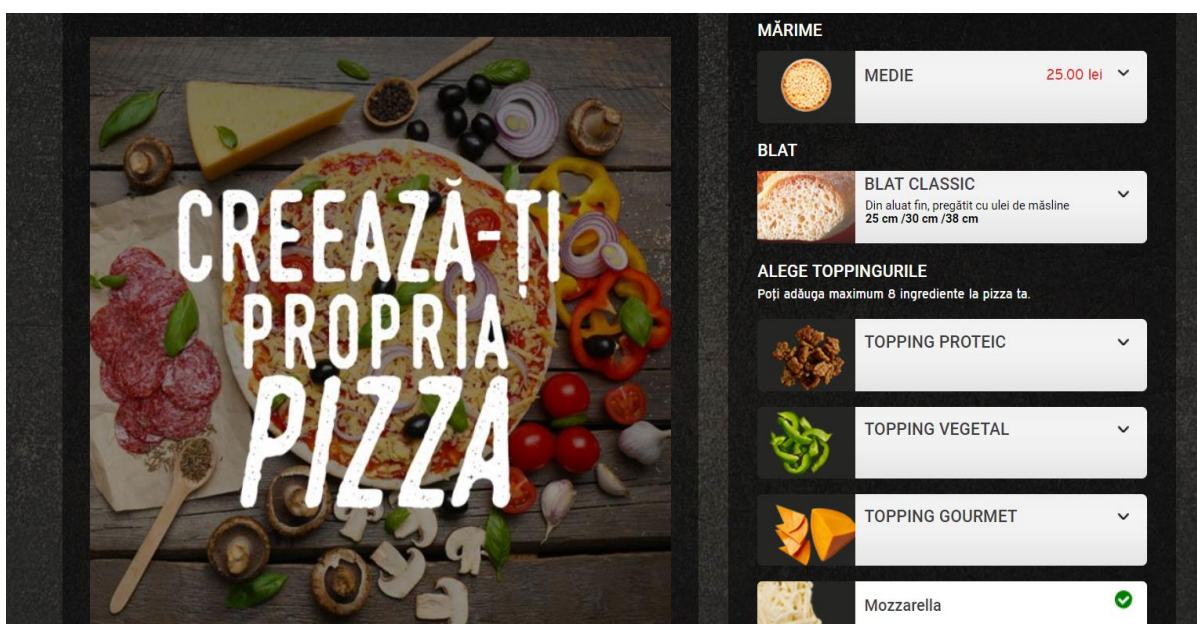


Fig. 1: „Pizza Hut”, creează-ți propria pizza

## **Capitolul III. Tehnologii utilizate:**

### **1. Node.js**

Node.js este un mediu gratuit pentru server, care folosește JavaScript și poate fi rulat pe platforme variate precum Windows, Linux, Unix, Mac OS X, etc.

Acest mediu permite generare de conținut dinamic într-o pagină web, manipularea fișierelor, preluarea datelor dintr-un form și adăugarea, ștergerea și modificarea datelor din baza de date.

### **2. Express.js**

Express.js este un framework minimal pentru Node.js, proiectat cu scopul de a crea aplicații web sau API-uri.

### **3. Typegoose:**

Mongoose este o bibliotecă de modelare a datelor obiect (ODM) pentru MongoDB și Node.js.

O problemă care apare atunci când este utilizat Mongoose cu TypeScript este că trebuie definite atât modelul Mongoose, cât și interfața TypeScript. Dacă modelul se modifică, trebuie să păstrăm sincronizarea fișierului de interfață TypeScript, sau interfața TypeScript nu ar reprezenta structura reală de date a modelului.

Această problemă poate fi rezolvată cu ajutorul bibliotecii Typegoose care ne permite să definim modelele Mongoose folosind clasele din Typescript.

### **4. MongoDB**

MongoDB este o bază de date NoSQL de tip document. Acest lucru înseamnă că în aceasta pot fi stocate documentele JSON, iar structura acestor documente poate varia. Acesta este unul dintre avantajele utilizării NoSQL, deoarece accelerează dezvoltarea aplicațiilor și reduce complexitatea implementărilor.

## Capitolul IV. Structura aplicației

### 1. Modelele

Modelele ne permit să definim modelele Mongoose folosind clase Typescript. Mai exact, aceasta este secțiunea unde putem defini numele unei tabele din baza de date și de asemenea numele câmpurilor din aceasta cât și tipurile lor.

```
import { prop, Typegoose } from 'typegoose';
import mongoose = require('mongoose');
import Schema = mongoose.Schema;
import Types = mongoose.Types;
import ObjectId = Types.ObjectId;

export class Recipe extends Typegoose {

    @prop({required: true})
    _id: ObjectId;

    @prop({required: true, unique: true})
    name: string;

    @prop({required: true})
    kg: string;

    @prop({required: true})
    price: string;

    @prop({required: true})
    ingredients: string;

    @prop({required: true})
    type: string;

    @prop({required: true})
    photo: string;

}
```

Fig. 2: Modelul unui produs

### 2. Configurarea (config)

În această secțiune se realizează conexiunea la baza de date și asigurarea unui port disponibil pentru Express Web Server.

De asemenea aici se creează containerul IoC care conține toate arhivele (repositories) folosite în aplicație.

### 3. Arhivele (repositories)

Folosim arhivele pentru a defini funcțiile de manipulare a bazei de date: adăugarea datelor, ștergerea acestora, preluarea datelor și modificarea lor.

```
import { Recipe } from "../models/Recipe";
import { Provides } from "typescript-ioc";
import { IRecipeRepository } from "../IRecipeRepository";

@Provides(IRecipeRepository)
export class RecipeRepository implements IRecipeRepository {

    private RecipeModel;

    constructor() {
        this.RecipeModel = new Recipe().getModelForClass(Recipe);
    }

    public getAll(): Promise<Recipe[]> {
        return this.RecipeModel.find().exec();
    }

    public getById(id: string): Promise<Recipe> {
        return this.RecipeModel.findById(id).exec();
    }

    public add(document: Recipe): Promise<Recipe> {
        let newRecipes = new this.RecipeModel(document);
        return newRecipes.save();
    }

    public update(id: string, document: any): Promise<Recipe> {
        return this.RecipeModel.findByIdAndUpdate(id, document, { new: true }).exec();
    }

    public delete(id: string): Promise<Recipe> {
        return this.RecipeModel.findByIdAndRemove(id).exec();
    }

    public getRecipeByRegex(RecipeName: string): Promise<Recipe[]> {
        return this.RecipeModel.find({name: new RegExp(RecipeName, 'i' )}).exec();
    }

    public getRecipeByName(RecipeName: string): Promise<Recipe[]> {
        return this.RecipeModel.find({name: RecipeName }).exec();
    }

    public getRecipesByType(type: string): Promise<Recipe[]> {
        return this.RecipeModel.find({type: type}).exec();
    }
}
```

Fig. 3: Arhivă pentru manipularea modelului Recipe



## 4. Controllers

Un controller este legătura dintre pagina web și arhivele sus menționate. Din pagina web se trimite o cerere, aceasta ajunge la controller, iar controller-ul folosește o funcție importată din arhivă pentru a obține rezultatul la cerere. Rezultatul este trimis în final înapoi la pagina inițială.

```

constructor() {
    this.router = Router();
    this.init();
    this.recipeModel = new Recipe().getModelForClass(Recipe);
    this.selectedObjectModel = new SelectedObject().getModelForClass(SelectedObject);
}

public getMainPage (req: Request, res: Response): void{

    var path = require('path');
    res.sendFile(path.resolve(__dirname + ".././lib/static/Cauta.html"));
}

public getAllRecipesByName(req:Request, res:Response): void {

    this.recipeRepository.getRecipeByRegex(req.body.reteta)
    .then(Cauta => {

        if(Cauta.length == 0) {
            res.status(this.HttpStatus_NoContent).send();
        } else {
            res.status(this.HttpStatus_OK).json(Cauta)
        }
    })
    .catch(err => res.status(this.HttpStatus_BadRequest).send(err));
}

public getNumarProduce(req: Request, res: Response): void{
    this.selectedObjectRepository.getSelectedObjectByUserId(req.body.id_user)
    .then(sel =>{
        console.log(sel)
        res.status(this.HttpStatus_Created).json(sel.length)
    })
    .catch(err => {console.log(err);res.status(this.HttpStatus_BadRequest).send(err)});
}

private init(): any {
    this.router.get('/:NumeReteta', this.getMainPage.bind(this));
    this.router.post('/allRetete', this.getAllRecipesByName.bind(this));
    this.router.post('/getNrProd', this.getNumarProduce.bind(this));
}

public getRoutes(): Router {
    return this.router;
}

```

Fig. 4: Controller pentru pagina unui produs

## 5. Static

Această secțiune conține totalitatea elementelor statice ale aplicației, precum pagini html, stilizarea acestora (css), JavaScript (listeners, renders, librării adiacente) și poze.

## 6. Structura bazei de date

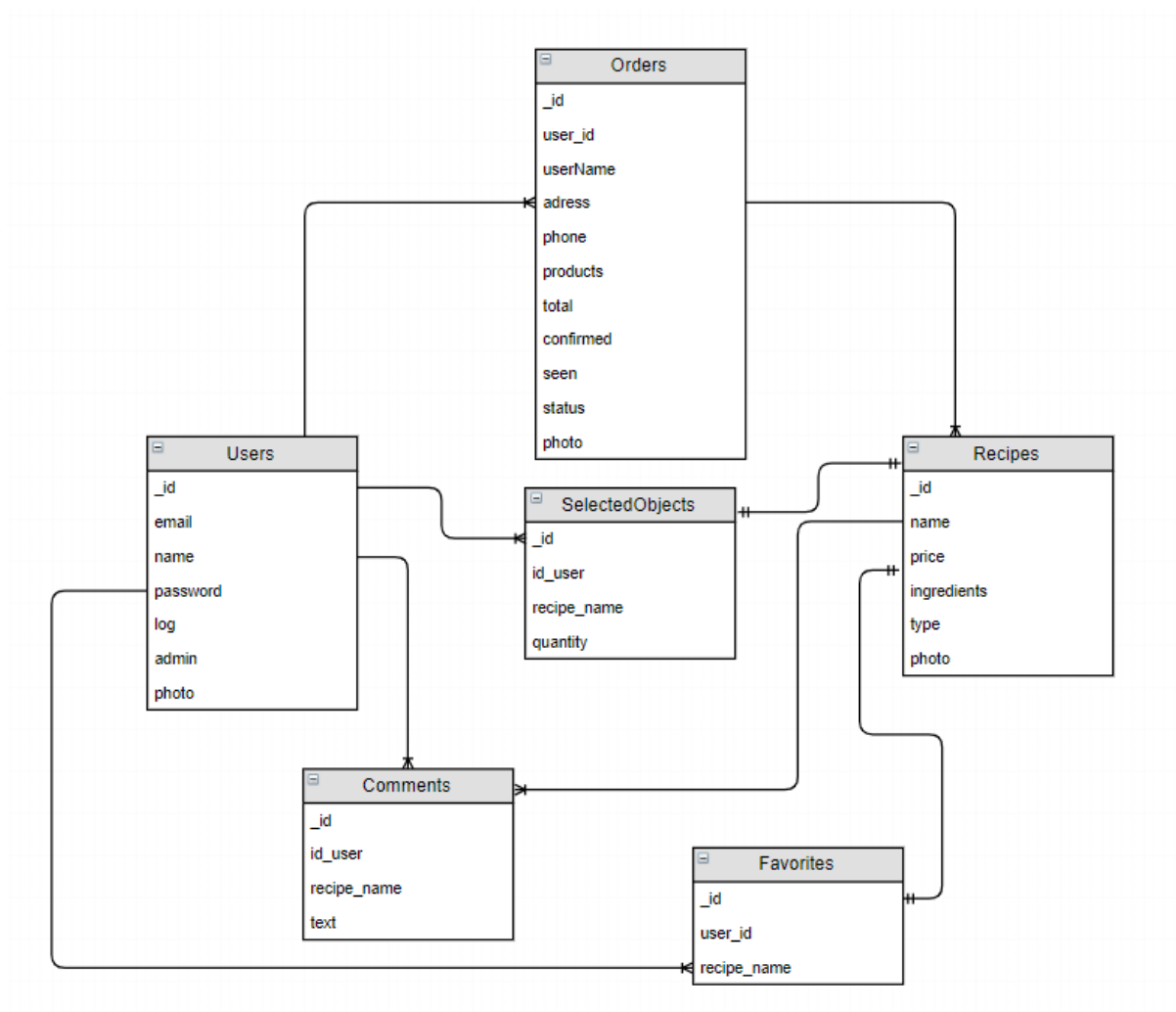


Fig. 5: Structura bazei de date

Tabela **users** conține:

- **id**: un id de tipul ObjectId, unic;
- **email**: o variabilă de tip string reprezentând email-ul folosit la crearea unui cont;
- **password**: variabilă de tip string, parola aleasă de utilizator când își creează un cont;
- **admin**: variabilă de tip bool, are valoarea true doar pentru utilizatorul ce are rol de administrator, pentru oricare alt utilizator este false implicit;
- **name**: variabilă de tip string, nume de utilizator, ales la crearea contului;
- **photo**: variabilă de tip string reprezentând calea unei poze;

Tabela **recipes** conține:

- **id**: un id de tipul ObjectId, unic;
- **name**: variabilă de tip string, numele produsului;
- **price**: variabilă de tip string, prețul unui produs;
- **ingredients**: variabilă de tip string, ingredientele unui produs;
- **type**: variabilă de tip string, tipul unui produs (tort, brioșă sau altele);
- **photo**: variabilă de tip string reprezentând calea unei poze;

Tabela **favorites** conține:

- **id**: un id de tipul ObjectId, unic;
- **userId**: variabilă de tip string, id-ul utilizatorului care adaugă produsul la favorite;
- **recipeName**: variabilă de tip string, numele produsului adăugat la favorite;

Tabela **comments** conține:

- **id**: un id de tipul ObjectId, unic;
- **userId**: variabilă de tip string, id-ul utilizatorului care adaugă un comentariu;
- **recipeName**: variabilă de tip string, numele produsului la a cărui pagină s-a comentat;
- **text**: variabilă de tip string, conținutul comentariului;

Tabela **selectedObjects** conține:

- **id**: un id de tipul ObjectId, unic;
- **userId**: variabilă de tip string, id-ul utilizatorului care adaugă produsul în coșul de cumpărături;
- **recipeName**: variabilă de tip string, numele produsului adăugat în coșul de cumpărături;
- **quantity**: variabilă de tip number, câte produse de același fel se doresc a fi comandate;

Tabela **orders** conține:

- **id**: un id de tipul ObjectId, unic;
- **userId**: variabilă de tip string, id-ul utilizatorului care a plasat comanda;
- **userName**: variabilă de tip string, conține numele și prenumele utilizatorului;

- **address:** variabilă de tip string, adresa de livrare;
- **phone:** variabilă de tip string, numărul la care poate fi contactat destinatarul;
- **products:** variabilă de tip string, produsele comandate, cantitatea lor și prețurile acestora;
- **total:** variabilă de tip string, valoarea totală a bunurilor comandate;
- **confirmed:** variabilă de tip boolean, true dacă este confirmată comanda, false altfel;
- **seen:** variabilă de tip boolean, true dacă adminul a văzut comanda, false altfel;
- **status:** variabilă de tip string, statutul comenzii (în procesare, preluat, în lucru, finalizat);
- **photo:** variabilă de tip string, calea spre poza unui tort, dacă trimiterea comenzii se face din „Personalizare”, altfel string gol.

## Capitolul V. Interacțiunea unui utilizator cu aplicația

### 1. Utilizator de tip client

Neautentificat:

Vizionarea produselor disponibile pe pagină poate fi făcută de orice utilizator, fie că acesta este autentificat sau nu.

Astfel un client poate naviga prin aplicație (să consulte meniul sau să deschidă pagina unui produs specific) și de asemenea poate căuta un anumit produs.

Pentru a căuta un produs, utilizatorul trebuie să introducă un cuvânt în bara de căutare și să apese butonul ce conține iconița în formă de lupă, pentru a începe căutarea. Dacă numărul produselor care conțin în denumirea lor acel cuvânt este diferit de 0, utilizatorului i se vor afișa toate acele produse.

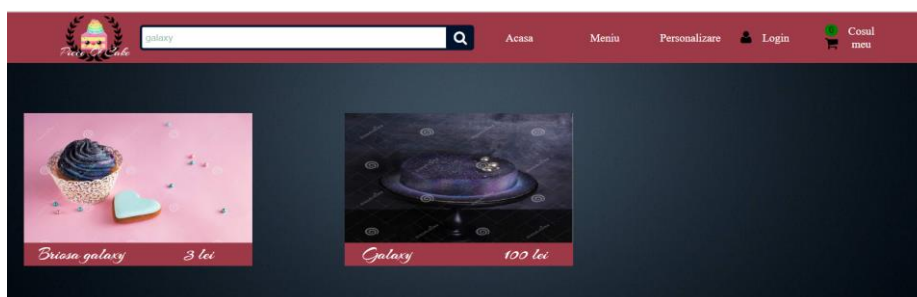


Fig. 6: Căutarea cuvântului „galaxy” prezent în două produse

În caz contrar, utilizatorului i se va afișa mesajul „Niciun produs găsit”.



Fig. 7: Căutarea cuvântului „ciocolata”, inexistent pe pagină

Pentru a putea adăuga în coș și comanda, utilizatorii au nevoie de un cont și să fie autentificați. Înregistrarea sau autentificarea utilizatorului se realizează prin apăsarea butonului Login din bara de navigare, urmată de introducerea datelor.

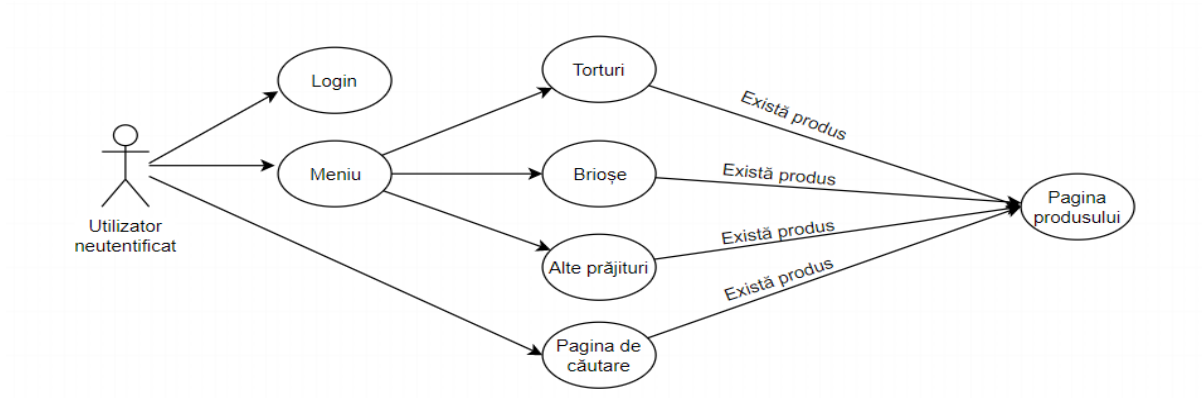


Fig. 8: Acțiunile pe care le poate face un utilizator neautentificat

### Înregistrare:

Clientul introduce un email valid și unic, un nume de utilizator și o parolă. Dacă email-ul nu este valid, datele nu vor fi introduse în baza de date, utilizatorul va fi anunțat că trebuie să introducă un email valid și întreg procesul va fi luat de la capăt.

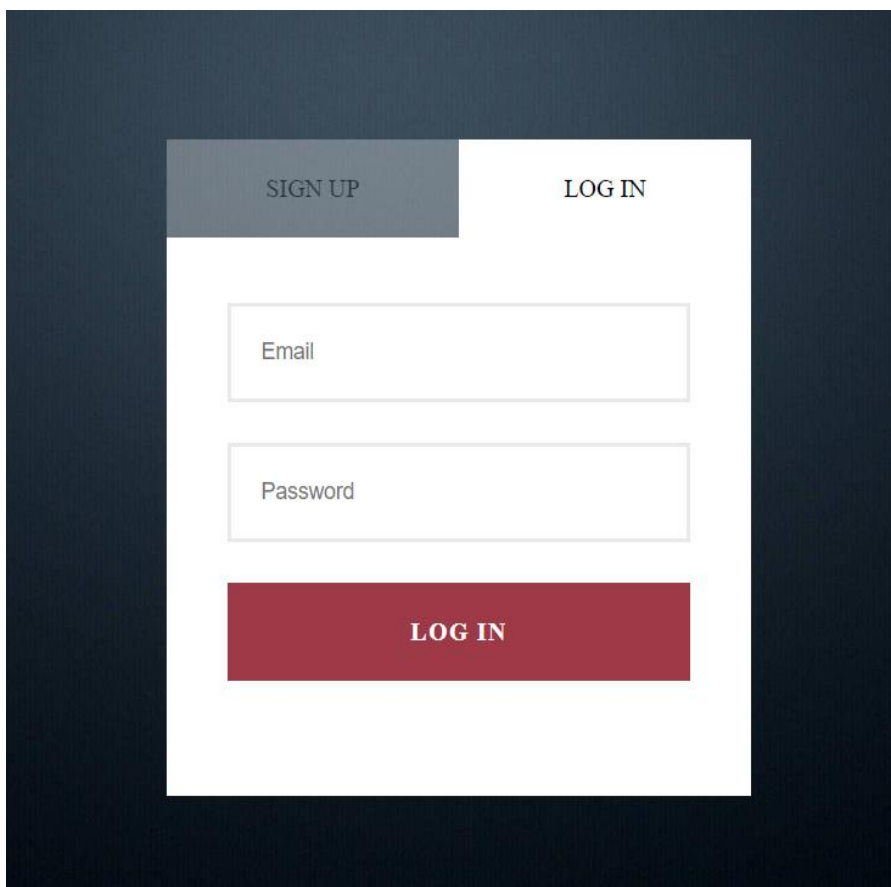
În cazul în care email-ul este valid și unic, un nou cont va fi creat cu succes și se va putea autentifica.

Fig. 9: Formular de înregistrare

### Autentificare:

Clientul introduce email-ul și parola. Datele sunt trimise către server și se verifică dacă exista email-ul.

Dacă email-ul există, se verifică dacă parola introdusă și cea din baza de date coincid. În caz afirmativ, pe fiecare pagină butonul de Login din bara de navigare este înlocuit cu butonul Contul meu și, de asemenea, utilizatorul va putea face mai multe acțiuni.



The image shows a login form centered on a dark blue background. The form itself is white. At the top of the form, there are two buttons: 'SIGN UP' on the left and 'LOG IN' on the right. Below these buttons are two input fields: 'Email' and 'Password'. At the bottom of the form is a large red button labeled 'LOG IN'.

Fig. 10: Formular de autentificare

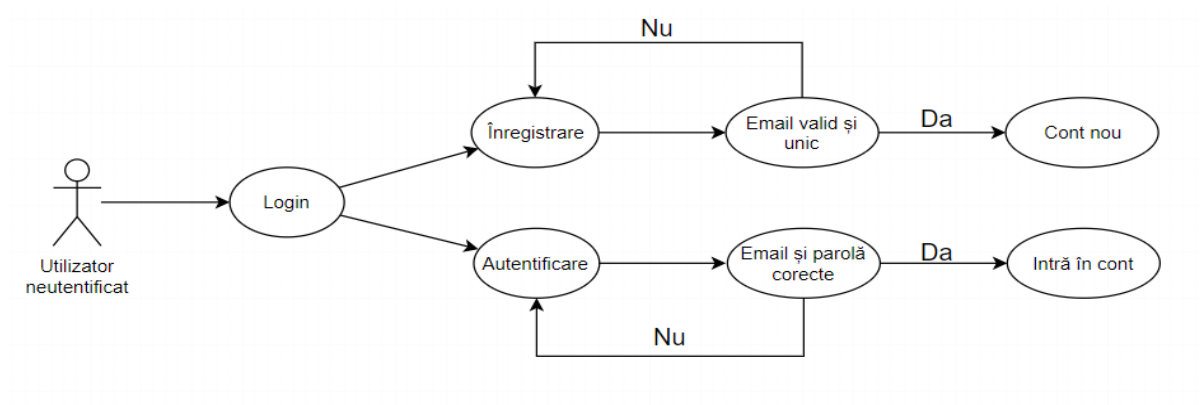


Fig. 11: Înregistrarea și autentificarea unui utilizator

După autentificare utilizatorul are acces la mai multe acțiuni, precum adăugarea produselor în coșul de cumpărături și plasarea unei comenzi, adăugarea unui produs la favorite, posibilitatea de a putea posta un comentariu pe pagina unui produs și comandarea unui tort personalizat.

Adăugarea unui comentariu:

Un utilizator are posibilitatea să posteze unul sau mai multe comentarii pe pagina unui produs. Această acțiune este utilă în cazul în care un utilizator are anumite nelămuriri despre un produs sau vrea să își exprime opinia despre acesta după achiziționare.

Utilizatorul poate șterge oricare din comentariile sale, dar nu are această posibilitate față de comentariile celorlalți utilizatori.

The screenshot shows a web form for adding a comment. At the top, there is a text input field with the placeholder text 'Acesta este un exemplu de comentariu' (This is an example of a comment) and a red button labeled 'Postează' (Post). Below this, there is a section for the user's profile, showing a circular profile picture and the name 'cami'. Underneath the profile, there is another text input field with the same placeholder text 'Acesta este un exemplu de comentariu'.

Fig. 12: Adăugarea unui comentariu



Fiecare produs, dacă avem cursorul poziționat asupra lui, va prezenta trei butoane. Apăsând primul buton utilizatorul poate adăuga produsul în coșul de cumpărături. Al doilea buton adaugă produsul în lista de produse favorite, iar al treilea permite utilizatorului să deschidă pagina produsului.

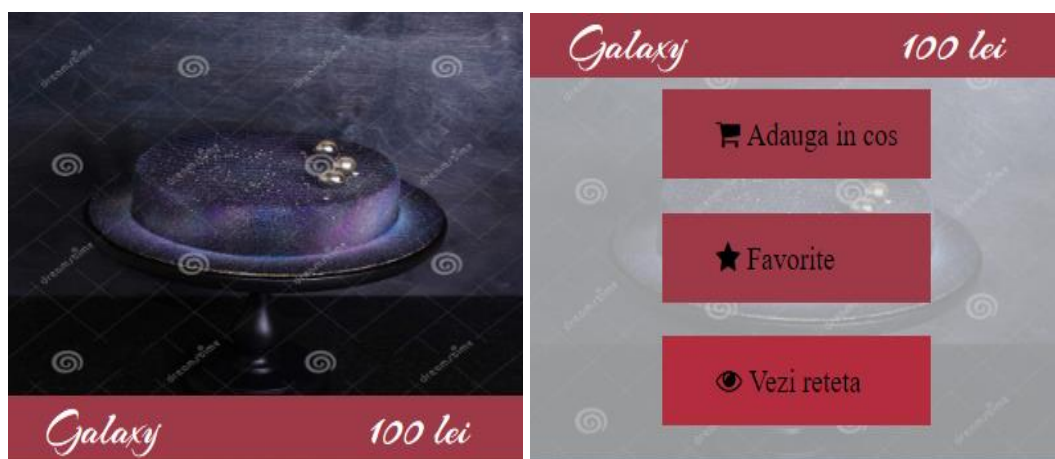


Fig. 13: Butoanele unui produs

#### Adăugarea la favorite:

Adăugarea unui produs în lista de produse favorite se face fie din pagina de căutare, fie din paginile Torturi, Brioșe și Alte prăjituri.

De asemenea, această acțiune se poate realiza și din pagina fiecărui produs.

Odată adăugat un produs la favorite, acesta poate fi găsit în pagina Favorite, pagină care poate fi accesată apăsând butonul „Favorite” din meniul care apare când cursorul este deasupra butonului „Contul meu” sau din contul utilizatorului.

Din pagina Favorite, utilizatorul poate adăuga produsul în coș, poate șterge elementul din Favorite sau poate deschide pagina acelui produs.

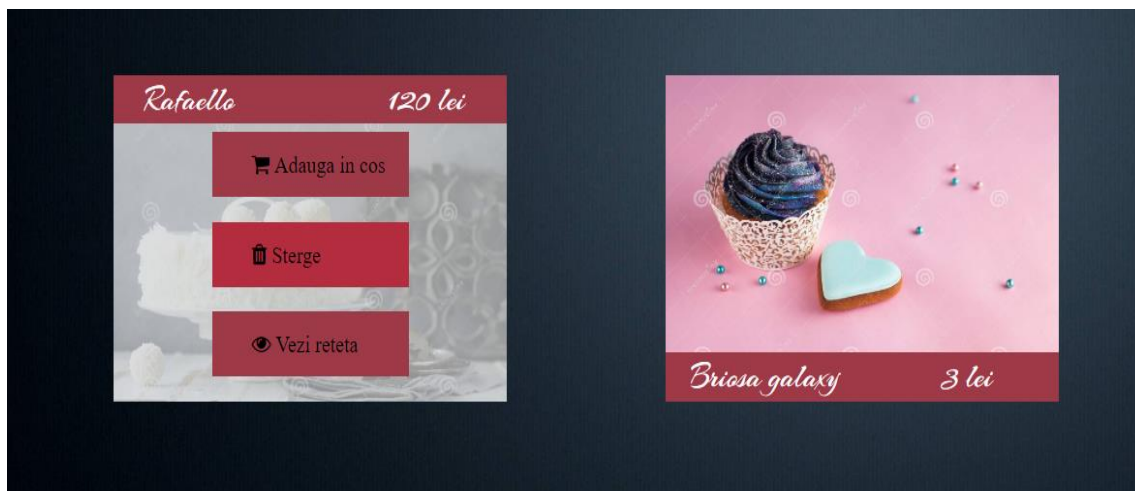


Fig. 14: Produsele din pagina "Favorite"

#### Adăugarea în coșul de cumpărături:

Adăugarea unui produs în coșul de cumpărături se poate face din pagina Favorite și din aceleași pagini ca și adăugarea unui produs la favorite.

De fiecare dată când utilizatorul adaugă un produs în coș, numărul produselor din coșul de cumpărături va fi afișat și va rămâne vizibil pe orice pagina, atât timp cât utilizatorul este autentificat. Utilizatorul nu poate introduce același produs în coș de două ori.

Pentru a elimina un produs din coșul de cumpărături, a schimba cantitatea, a vedea costul total și a trimite comanda, utilizatorul trebuie să acceseze pagina „Coșul meu”.

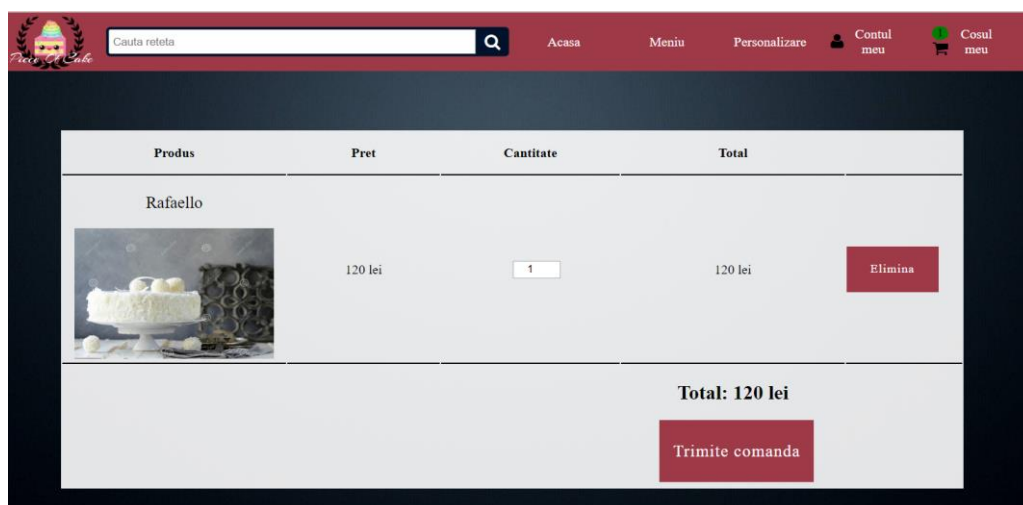


Fig.15: Coșul de cumpărături având produse

În cazul în care nu exista niciun produs în coș, pe pagină va fi afișat mesajul „Nu există produse în coș”.

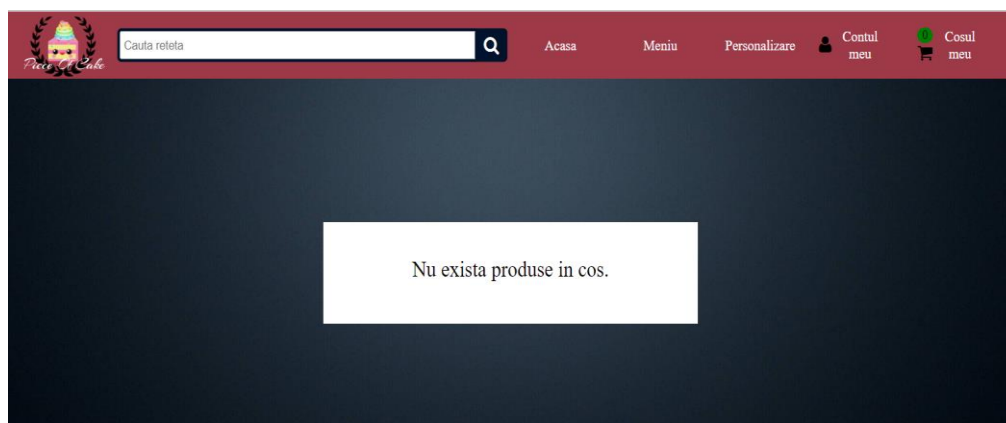


Fig. 16: Coșul de cumpărături fără produse

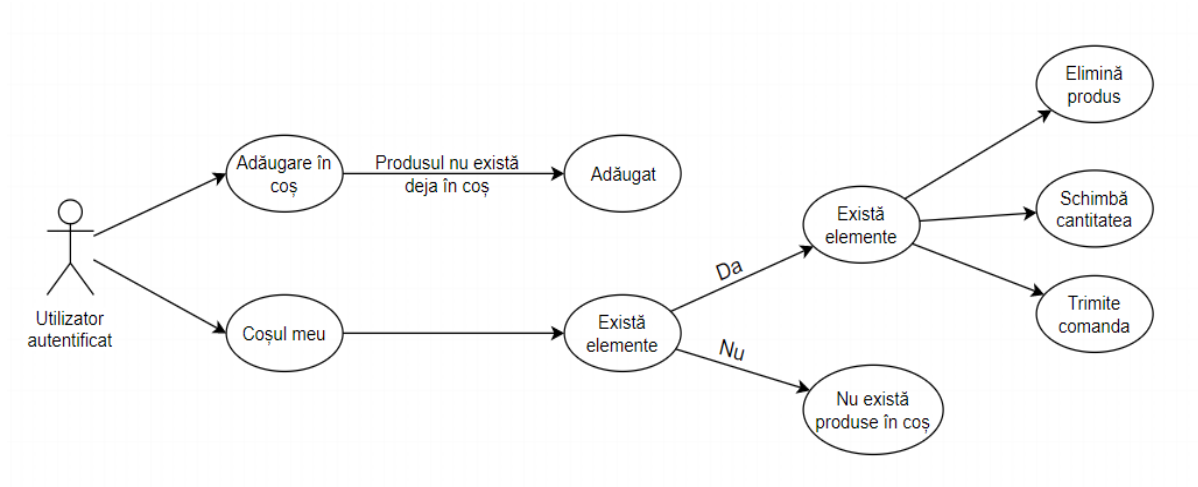


Fig. 17: Interacțiunea unui client cu coșul de cumpărături

#### Comandă personalizată:

Din pagina „Personalizare”, un utilizator își poate crea un tort pe placul său, având posibilitatea să aleagă forma tortului, câte niveluri să aibă, dimensiunea și aroma blatului, ce fel de cremă să conțină tortul în interior, cu ce să fie acoperit și, de asemenea, să își aleagă decorațiunile.

Prețul acestui tort ca fi afișat în timp real și va fi calculat în funcție de alegerile utilizatorului.

**Comenzi personalizate**

Forma:  Niveluri:  Blat:

Crema:  Acoperit cu:  Decoratiuni:

Adauga poza:  No file chosen

Indicatii:

Nume:  Prenume:  Telefon:

Judet:  Localitate:  Str.,Nr.:

Data livrarii:

Total: 0 lei

Fig. 18: Comandă personalizată

#### Trimiterea comenzii:

În urma apăsării butonului ”Trimite comanda” utilizatorul va fi redirectionat spre un chestionar. Acesta trebuie să introducă date personale precum numele de familie și prenumele, numărul de telefon la care poate fi contactat, adresa de livrare și data la care dorește să îi fie livrată comanda.

De asemenea, produsele comandate de acesta pot fi vizionate sub chestionar, cu scopul de a permite clientului să facă o ultimă verificare înainte de plasarea comenzii.

Nume:  Prenom:  Telefon:   
 Judet:  Localitate:  Str.,Nr.:   
 Data livrării:

1xRafaello	120 lei
<b>Total: 120 lei</b>	

**Plaseaza comanda**

Fig.19: Trimiterea comenzii

După apăsarea butonului ”Pasează comanda”, cererea, împreună cu datele aferente acesteia, va fi stocată în baza de date, iar produsele din coșul de cumpărături vor fi șterse.

Clientul va primi un e-mail pentru a fi anunțat că a fost plasată comanda iar administratorul va primi o notificare i va putea vizualiza comanda.

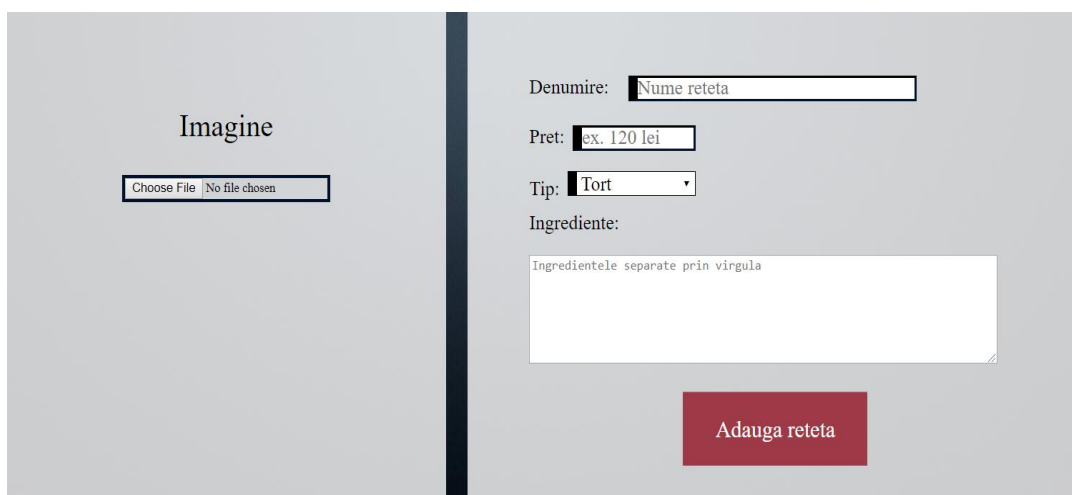
## 2. Utilizator de tip administrator

Un utilizator de tip administrator are acces la mai multe acțiuni decât un utilizator de tip client, întrucât acesta este cel ce adaugă produse în meniu și preia comenzile.

Adăugare produs:

Administratorul poate adăuga un produs din pagina ”Adaugă rețetă”. Câmpurile necesare completării de către administrator sunt constituite de către o poză reprezentativă a produsului, numele unic a acestuia, prețul și tipul produsului (tort, brișă sau altele) și ingredientele principale, separate prin virgulă.

Adăugarea ingredientelor este necesară în scopul detectării alergenilor din produs, de către un utilizator de tip client.



Imagine

Choose File No file chosen

Denumire:

Pret:

Tip:

Ingrediente:

Ingredientele separate prin virgula

Fig. 20: Adăugarea unui nou produs

#### Preluarea comenzii:

În urma plasării unei comenzi și de către client, administratorul va primi o notificare ce îl va anunța că o nouă comandă a fost plasată. Această notificare este constituită dintr-un număr ce reprezintă câte comenzi au plasate de un client, dar nedeschise de administrator, altfel spus, nevăzute.



Logo: Patisserie Cube

Cauta reteta

Acasa Menu Personalizare Contul meu Comenzi primite

Toate comenzile	Nevazute	In lucru	Finalizate
<div>Comanda: 5d0dc40bbfc9c3403897ef41 Total: 220 lei Status: in procesare</div> <div>Vezi comanda In lucru Finalizare</div>			
<div>Comanda: 5d0e76fae2ca700820f2565f Total: 120 lei Status: in procesare</div> <div>Vezi comanda In lucru Finalizare</div>			

Fig. 21: Comenzi nevăzute

Accesând pagina "Comenzi primite" administratorul va fi capabil să vadă toate comenzile trimise de clienții săi. Aceste comenzi pot fi filtrate după criterii precum nevăzute sau după statutul comenzilor (în lucru sau finalizate).

Aceste comenzi vor fi inițial afișate sub forma: id comandă, total de plată, stadiul comenzii. Alături de aceste date se află butonul "Vezi comanda", care permite administratorului să vadă comanda în detaliu și, dacă este accesat din comenzile nevăzute, schimbă statutul comenzii din "în procesare" în "preluat". Butonul "În lucru" schimbă statutul comenzii în statutul "în lucru" și butonul "Finalizare" schimbă statutul în "finalizat".

Comanda: 5d0dc40bbfc9c3403897ef41	Total: 220 lei	Status: în procesare	Vezi comanda	In lucru	Finalizare
Comanda: 5d0dd4274aaf03394875e2bd	Total: 220 lei	Status: în lucru	Vezi comanda	In lucru	Finalizare
Comanda: 5d0dd4e74aaf03394875e2be	Total: 220 lei	Status: finalizat	Vezi comanda	In lucru	Finalizare

Fig. 22: Afișarea comenzilor în "Toate comenzile"

Apăsând butonul "Vezi comanda", administratorul va putea vedea comanda detaliată, aceasta fiind compusă din datele destinatarului (nume, număr de telefon și adresă) și produsele comandate, iar numărul din notificare se va schimba automat.

**Date destinatar:**

Nume destinatar: Bacaoanu Camelia  
 Telefon: 0736265981  
 Adresa: Jud. Neamt, Loc. Roman, Str. Antipa, Nr. 3

**Comanda:**

1xRafaello	120 lei
<b>Total: 120 lei</b>	

Fig. 23: Comandă detaliată

**Concluzii:**

Aplicația „Piece of Cake” permite posesia unui restaurant, în care clienții pot intra, pot consulta meniul și comanda, fără ca administratorul să dețină un spațiu fizic, aceasta fiind foarte benefic din punct de vedere financiar.

Fiind o aplicație online, personalul poate fi considerabil redus. Chelnerii sau persoane care să stea la recepție pentru a prelua comenzi nu sunt necesare. Personalul care se ocupă cu realizarea produselor și cei care se ocupă cu livrările sunt minimul necesar.

De asemenea, deoarece aplicația are un coș de cumpărături, administratorul nu este nevoit să preia comenzile prin intermediul telefonului. Comenzile vor fi scrise și clare iar aceasta duce la scăderea posibilității ca o comandă să fie înțeleasă greșit.

Aplicația „Piece of Cake” este benefică și pentru clienți. Din ce în ce mai mulți tineri prezintă diverse forme de anxietate când vine vorba de apelurile telefonice. Mulți preferă să nu mai comande decât să realizeze comanda prin intermediul unui apel.

Sunt multe comenzi care sunt plasate însă nu sunt onorate din motive variate, precum lipsa produsului de pe stoc, pierderea acestuia în timpul livrării sau faptul că acea comandă este, fără intenție, trecută cu vederea. Prin sistemul de monitorizare al comenzilor clientul poate verifica stadiul în care se află comanda sa și poate contacta comerciantul în cazul în care realizează prezența unei probleme.

Desigur, aplicația prezentată nu este perfectă. Aceasta ar putea fi extinsă și îmbunătățită prin adăugarea unor funcții precum:

- Posibilitatea ca un client să poată plăti cu cardul;
- Adăugare de filtre multiple pe pagina de de meniu;
- Înlocuire chestionarului din pagina Personalizare cu un joc interactiv, drag and drop, care îi permite utilizatorului să își formeze o imagine mai exactă asupra aspectului comenzii sale personalizate;
- Adăugarea unei note pe pagina unui produs, dată de voturile clienților.



## Bibliografie:

1. Mongoose și mongoDb: <https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/>
2. Typegoose: <https://www.npmjs.com/package/typegoose>
3. Multiselect: <https://www.jqueryscript.net/form/Drop-Down-Combo-Tree.html>
4. Funcționalități coș: <https://www.youtube.com/watch?v=YeFzkC2awTM&t=649s>
5. Cumpărături online: <http://www.businessdictionary.com/definition/online-shopping.html>
6. Avantaje și dezavantaje ale cumpărăturilor online: <https://bunadimineata.ro/de-dimineata/avantajele-si-dezavantajele-cumparaturilor-online/>
7. Prima literă mare: <https://flaviocopes.com/how-to-uppercase-first-letter-javascript/>
8. Node.js: [https://www.w3schools.com/nodejs/nodejs\\_email.asp](https://www.w3schools.com/nodejs/nodejs_email.asp)
9. Flex: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
10. Grid: <https://css-tricks.com/snippets/css/complete-guide-grid/>
11. Calendar: <https://jqueryui.com/datepicker/>
12. Statistici: <https://www.gpec.ro/blog/raportul-pietei-ecommerce-gpec-2018-romanii-au-facut-cumparaturi-online-de-pest-3-5-miliarde-euro-in-2018>
13. Statistici: <https://ec.europa.eu/eurostat/statistics-explained/pdfscache/46776.pdf>
14. Pizza hut: [https://www.pizzahutdelivery.ro/pizza/pizza-creata-de-tine/size\\_medium/base\\_trad](https://www.pizzahutdelivery.ro/pizza/pizza-creata-de-tine/size_medium/base_trad)
15. Tinerii și apelurile telefonice: <https://www.theguardian.com/commentisfree/2016/aug/26/whatsapp-phone-calls-smartphone-messaging-millennials>
16. Diagrame: <https://www.draw.io/>
17. Split: [https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref\\_split4](https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_split4)
18. Liste în hrml: [https://www.w3schools.com/html/tryit.asp?filename=tryhtml\\_lists\\_unordered\\_none](https://www.w3schools.com/html/tryit.asp?filename=tryhtml_lists_unordered_none)
19. Nodemailer: [https://www.w3schools.com/nodejs/nodejs\\_email.asp](https://www.w3schools.com/nodejs/nodejs_email.asp)
20. Regex în Node.js: <https://xuguoming.wordpress.com/2015/02/11/using-variable-regex-with-mongodb-query-in-node-js/>