

UNIVERSITATEA „ALEXANDRU IOAN CUZA” DIN IAȘI
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

HealthyFood

**Aplicație web ce promovează producătorii
locali din apropiere**

propusă de

Moroșanu Vlăduț-Haralambie

Sesiunea: Iulie, 2019

Coordonator științific

Prof. Colab. Olariu Florin

UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI
FACULTATEA DE INFORMATICĂ

HealthyFood

Aplicație web ce promovează producătorii
locali din apropiere

Moroșanu Vlăduț-Haralambie

Sesiunea: Iulie, 2019

Coordonator științific

Prof. Colab. Olariu Florin

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 falsificării 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.

Data azi,

Semnătură student

DECLARAȚIE PRIVIND ORIGINALITATE ȘI RESPECTAREA DREPTURILOR DE AUTOR

Prin prezenta declar că Lucrarea de licență cu titlul „HealthyFood - Aplicație web ce promovează producătorii locali din apropiere” este scrisă de mine și nu a mai fost prezentată niciodată la o altă facultate sau instituție de învățământ superior din țară sau străinătate. De asemenea, declar că toate sursele utilizate, inclusiv cele preluate de pe Internet, sunt indicate în lucrare, cu respectarea regulilor de evitare a plagiatului:

- toate fragmentele de text reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și dețin referința precisă a sursei;
- reformularea în cuvinte proprii a textelor scrise de către alți autori deține referința precisă;
- codul sursă, imagini etc. preluate din proiecte open-source sau alte surse sunt utilizate cu respectarea drepturilor de autor și dețin referințe precise;
- rezumarea ideilor altor autori precizează referința precisă la textul original.

Absolvent Moroșanu Vlăduț-Haralambie

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „HealthyFood - Aplicație web ce promovează producătorii locali din apropiere”, 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, 27.06.2019

Absolvent Moroșanu Vlăduț-Haralambie

(semnătura în original)

ACORD PRIVIND PROPRIETATEA DREPTULUI DE AUTOR

Facultatea de Informatică este de acord ca drepturile de autor asupra programelor-calculator, în format executabil și sursă, să aparțină autorului prezentei lucrări, Moroșanu Vlăduț-Haralambie.

Încheierea acestui acord este necesară din următoarele motive:

Extinderea aplicației și îmbunătățirea funcționalităților existente în viitor.

Iași, 27.06.2019

Decan Conf. Dr. Iftene Adrian

(semnătura în original)

Absolvent Moroșanu Vlăduț-Haralambie

(semnătura în original)

Cuprins

Introducere.....	1
Contribuții.....	1
Capitolul 1. Arhitectura aplicației	4
1.1 Arhitectura frontend-ului și a backend-ului	5
1.2 Arhitectura bazei de date.....	6
1.3 Concluzii.....	6
Capitolul 2. Descrierea aplicației	4
2.1 Înregistrare prin validarea e-mail-ului	5
2.2 Autentificare și autorizare	6
2.3 Adăugarea unui anunț	6
2.4 Vizualizarea produselor din apropiere	6
2.5 Gestionarea anunțurilor	6
2.6 Algoritmul pentru calcularea tendințelor	6
2.7 Sugestii pentru căutarea unui produs	6
2.8 Algoritmul utilizat pentru sugerarea altor produse	6
2.9 Comunicarea cu ceilalți utilizatori	6
2.10 Pagina de prezentare a unui vânzător	6
2.11 Secțiune de vizualizare și adăugare de recenzii	6
2.12 Grupul de persoane de încredere	6
2.13 Gestionarea planurilor de livrare	6

2.14 Administrarea contului utilizatorului	6
2.15 Concluzii	6
Capitolul 3. Beneficiile utilizării aplicației.....	4
3.1 Avantaje și noutăți aduse.....	5
3.2 Dezavantaje	6
3.3 Concluzii	3
Capitolul 4. Concluziile lucrării și posibilitatea extinderii aplicației în viitor	4
3.1 Concluzii generale	5
3.2 Funcționalități ce urmează să fie dezvoltate	6
Bibliografie	1
Anexe	1

Introducere

În ultima perioadă, magazinele mari deținute de rețele multinaționale au „invadat” orașele, dar și localitățile din jurul acestora. Având posibilitatea de a importa ușor produse din afara țării, aceste companii și-au dezvoltat strategic magazine de tip supermarket și hipermarket în majoritatea orașelor, iar producătorii locali de alimente au fost înlăturați încetul cu încetul. Cu toate că agricultorii oferă o calitate mai bună a produselor, aceștia nu reușesc să își promoveze afacerile printre multitudinea de oferte ale acestor lanțuri de magazine.

Am realizat această aplicație cu scopul de a crea un spațiu online ce vine în ajutorul persoanelor ce vor să își promoveze produsele alimentare, dar și al clienților ce își doresc găsirea alimentelor naturale mai simplu și mai rapid. Această platformă favorizează micii producători în a-și promova produsele doar clienților din apropiere, distanța fiind calculată pe baza locației utilizatorilor. Fiecare vânzător va deține o pagină de prezentare ce va putea fi personalizată pentru a arăta cât mai atractiv. În același timp, aplicația îmbunătățește timpul de căutare a produselor, fiind realizată pe baza recenziilor vânzătorilor și oferind întotdeauna cele mai atractive alimente.

În acest moment, producătorilor le este dificil să își facă simțită prezența pe rețelele de socializare, fie că postarea se pierde printre alte anunțuri, fie este dedicată persoanelor care nu sunt interesate. Din acest motiv, am dorit crearea unei funcționalități prin care toți utilizatorii din grupul de persoane apropiate vânzătorului vor fi informați dacă acesta face livrări în diferite puncte ale orașului, specificând locația și intervalul orar.

Există diverse aplicații ce vin în ajutorul producătorilor locali, care au ca scop principal, atât postarea de anunțuri alimentare în anumite zone, cât și gestionarea comenzilor și oferirea de servicii de curierat adresate clientului. Printre aplicațiile cele mai populare, se numără următoarele: *PlatFerma*, *MiciProducători* și *PentruLocalnici*.

*PlatFerma*¹ este o aplicație web unde producătorii se pot promova prin adăugarea fermei lor pe hartă. Fiecare utilizator își poate detalia ferma în secțiunea „Portret de fermier” și își poate

¹ <https://platferma.ro/>

seta un loc pe harta fermelor. Clienții se pot abona la noutățile oricărui producător și pot vedea restaurantele ce servesc produse bio. Pentru a-și promova alimentele, un fermier trebuie să contacteze administratorul site-ului, pentru a fi adăugat în lista producătorilor activi.

*MiciProducători*² este o platformă dedicată promovării micilor producători din România, cu scopul de a identifica noi căi de marketing. Vanzătorii pot posta anunțuri adăugând poze și specificând locația acestora. Este disponibilă contactarea vânzătorilor prin numărul de telefon sau prin adresa de e-mail. Aplicația deține și un blog unde administratorii postează diferite noutăți din domeniul alimentar.

*PentruLocalnici*³ este o aplicație destinată antreprenorilor ce dețin o firmă specializată în produse realizate local. Utilizatorii au la dispoziție un formular pentru înscriere cu numele firmei și datele personale. După ce este acceptată cererea de înscriere, fiecare producător își poate detalia activitatea. Clienți pot vedea lista producătorilor pe o rază de 50 de kilometri împreună cu locația sediului firmei.

Obiectivul acestei lucrări de licență este crearea unei aplicații web ce vine în sprijinul producătorilor locali de alimente. În această aplicație, producătorii vor putea adăuga anunțuri cu privire la produsele acestora și vor putea gestiona planuri de livrare pentru comenzile primite. Clienții vor putea găsi alimentele potrivite cu ajutorul distanței calculate pe baza coordonatelor, fiind posibilă afișarea anunțurilor din apropiere. Platforma va oferi suport pentru recenzii (cu privire la vânzători) și va afișa conținut calitativ prin filtrarea anunțurilor ce au primit critici de la ceilalți clienți. Fiind o aplicație în care promovările alimentare se vor schimba de la un anotimp la altul, un algoritm va stabili ce anunțuri vor fi în topul tendințelor în perioada respectivă. Această funcționalitate va avea două roluri: de a informa clienții despre produsele disponibile doar în perioada curentă a anului și de a le sugera alte categorii de alimente dintre cele mai căutate.

Ca metodologie folosită pentru îndeplinirea acestui obiectiv, primul pas a fost realizarea unui studiu de piață în ceea ce privește alternativele producătorilor locali și ce nevoi nu sunt încă îndeplinite cu ajutorul aplicațiilor deja existente. Deoarece mulți vânzători din apropierea orașelor apelează la rețelele sociale pentru a-și promova activitatea, am studiat grupurile de pe

² <http://www.miciproducatori.ro/>

³ <https://pentrulocalnici.ro/>

platforma Facebook destinate comerțului de alimente și am creat o listă cu funcționalitățile elementare ale aplicației. Al doilea pas a constat în crearea unui plan al bazei de date, urmând ca apoi să fie implementată prima versiune. Ultimul pas a fost dezvoltarea aplicației împreună cu realizarea unei liste de priorități ale funcționalităților. În ceea ce privește design-ului platformei, am adăugat o machetă ce cuprinde paginile principale ale aplicației, urmând să dezvolt interfața utilizatorului pe baza schițelor adăugate.

HeathyFood este o aplicație ce vine în ajutorul producătorilor locali de alimente ce vor să își promoveze anunțurile inteligent pe o piață destinată consumului de produse naturale. Platforma ajută clienții să găsească alimente simplu și mai ușor pe baza locației acestora, poate stabili ce anunțuri sunt în topul tendințelor într-o anumită perioadă a anului și oferă sugestii utilizatorilor pe baza acțiunilor acestora. Aplicația pune la dispoziția micilor antreprenori suportul pentru gestionarea livrărilor prin adăugarea planurilor de livrare a comenzilor direct la domiciliul clienților sau în diferite locații ale orașelor.

Am apelat la integrarea cu Google Maps deoarece este mult mai ușoară vizualizarea anunțurilor pe o hartă interactivă, facilitează orientarea vânzătorilor spre adresa de livrare a cumpărătorilor, dar ajută și la calcularea distanței dintre producător și client.

Fiind o aplicație în care promovările alimentare se vor schimba de la un anotimp la altul, am implementat un algoritm ce va stabili ce anunțuri vor fi în topul tendințelor în perioada respectivă a anului. Această funcționalitate are scopul de a informa utilizatorii despre produsele disponibile doar în intervalul curentă, dar și de a le sugera clienților să găsească alte categorii de alimente.

Micii antreprenori ce sunt la început de drum au o dificultate în a-și gestiona livrările, mai ales în cadrul orașelor. În urma studiului efectuat pe rețelele de socializare, acești producători își livrează produsele, fie în anumite puncte principale din oraș, fie prin predarea acestora la domiciliu clienților. Vânzătorul poate planifica ruta livrărilor pentru o anumită zi, iar clienții vor fi înștiințați când comenzile plasate vor sosi la domiciliul acestora.

Lucrarea de licență este structurată pe patru capitole ce descriu funcționalitățile implementate, dar și pașii parcurși pentru realizarea acestei aplicații.

Primul capitol descrie în detaliu arhitectura aplicației și pune în evidență structura backend-ului și pe cea a frontend-ului. Acest capitol conține reprezentarea vizuală a aplicației, dar și detalierea arhitecturii bazei de date.

În capitolul al doilea sunt detaliate funcționalitățile aplicației, cât și pașii urmați pentru realizarea scopului lucrării. Acest capitol conține o descriere minuțioasă a algoritmilor utilizați, împreună cu scenariile de utilizare ale aplicației.

Capitolul al treilea exemplifică beneficiile aduse în raport cu aplicațiile deja existente în acest moment. Se prezintă avantajele și dezavantajele folosirii aplicației în raport cu celelalte alternative.

Ultimul capitol prezintă concluziile generale aduse asupra acestei lucrări și enumeră modurile prin care aplicația poate fi îmbunătățită sau dezvoltată.

Contribuții

Producătorul are la dispoziție o pagină unde își poate promova produsele prin postarea de anunțuri, adăugând poze sugestive. Acesta poate stabili locația anunțului, fie automat prin poziția sa curentă, fie manual căutând o locație.

Deoarece afișarea anunțurilor se bazează pe distanța calculată dintre coordonatele utilizatorului și cele ale anunțului, calitatea rezultatelor oferite este îmbunătățită, evitând produsele ce sunt indisponibile zonei clientului.

Clienții pot vizualiza anunțurile ce se află în topul tendințelor pentru perioada curentă a anului pe baza unui algoritm ce utilizează numărul de accesări ale fiecărui anunț pentru fiecare zi.

Aplicația oferă un suport bun în a le oferi utilizatorilor sugestii de căutare. Suportul pentru sugestii folosește un algoritm ce sortează căutările descrescător, după numărul de accesări din partea altor utilizatori.

Pentru fiecare utilizator ce accesează platforma, pe baza activității anterioare, se va stabili o listă de sugestii de anunțuri ce vor fi afișate. Algoritmul calculează cât este de interesat un client de o anumită categorie de produse.

Fiecare vânzător deține o pagină de prezentare pe care o poate edita și personaliza. În această pagină, producătorul își poate încărca o poză de tip copertă și își poate detalia activitatea sa de agricultor.

Aplicația pune la dispoziție un chat, în care se poate vedea dacă utilizatorul căutat este online, în caz contrar, se va afișa data când acesta a fost activ ultima oară în aplicație.

Platforma conține o secțiune unde utilizatorii pot lăsa recenzii persoanelor cu care au interacționat. Anunțurile vor fi filtrate și vor fi afișate doar cele care provin de la vânzători cu recenzii pozitive.

Utilizatorul deține grupul de persoane de încredere unde poate adăuga, șterge sau bloca alți utilizatori. Scopul acestui grup este de a furniza locația de livrare doar persoanelor de încredere.

Pentru vânzătorii ce pot face livrări, aceștia își pot gestiona planurile de livrare direct din aplicație. În acest fel, clienții vor fi informați automat atunci când livrarea comenzii va fi efectuată.

Aplicația pune la dispoziție două moduri de a gestiona livrările: la domiciliu, sau la o anumită locație pe hartă.

Utilizatorul își poate gestiona setările generale ale contului precum poza de profil, informațiile personale, parola de acces în aplicație, dar și locațiile furnizate de acesta.

O altă funcționalitate oferită de aplicație este de a notifica utilizatorii prin e-mail sau prin atenționare în cadrul platformei atunci când o persoană din grupul utilizatorului postează un anunț nou sau face o livrare nouă.

Capitolul 1. Arhitectura aplicației

Arhitectura platformei este alcătuită din două părți principale: o aplicație web destinată interacțiunii cu utilizatorul și un server de tip API REST, ce conține logica gestionării informațiilor primite de la clienți. Interfața utilizatorilor preia acțiunile acestora și le transmite serverului, primind ca răspuns datele prelucrate. Pentru a dezvolta funcționalități precum suportul pentru sugestii sau vizualizarea anunțurilor ce sunt în tendințe într-o anumită perioadă a anului, există fire de execuție recurente ce rulează la fiecare 15 minute, analizând și prelucrând datele primite de la utilizatori. De asemenea, serverul conține și un serviciu de e-mail ce trimite utilizatorilor un e-mail de verificare atunci când aceștia își creează un cont nou.

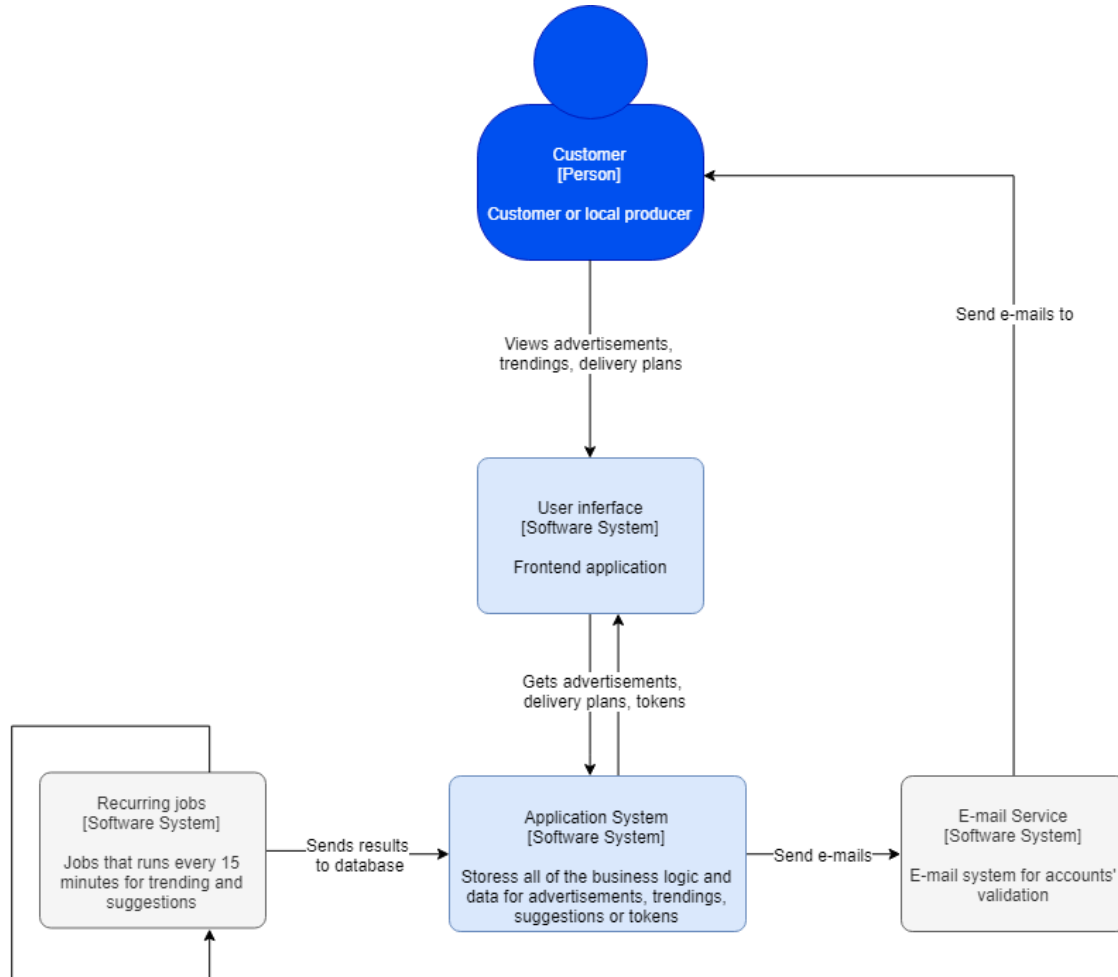


Fig. 1 - Arhitectura aplicației

1.1 Arhitectura frontend-ului și a backend-ului

Interfața utilizatorului este o aplicație de tip *Single-Page Application*, realizată cu ajutorul framework-ului Angular. Funcționalitățile sunt structurate pe module și pe componente, fiecare modul aparținând unei caracteristici ale aplicației. Toate cererile HTTP către server sunt realizate cu ajutorul serviciilor injectate în interiorul componentelor prin mecanismul de *Dependency Injection*.

Backend-ul este o aplicație de tip Web API, realizată cu ajutorul framework-ului .NET Core. Arhitectura serverului este de tip *Three-Layer-Architecture* și conține următoarele componente: un strat pentru datele primite din baza de date, unul ce conține logica gestionării datelor și unul folosit pentru a expune API-urile create. În stratul specific pentru date sunt generate modelele cu ajutorul ORM-ului Entity Framework, dar și clase ce implementează modelul *Repository Pattern*. Printre aceste clase se numără: *AdvertisementsRepository*, *DeliveryRepository*, *RelationshipRepository* sau *PresentationRepository*. Stratul pentru logica aplicației conține servicii diferite pentru fiecare funcționalitate a platformei. De exemplu, această componentă conține clase precum: *EmailService*, *ReviewService*, *DeliveryService* sau *UserService*. Stratul specific pentru prezentare conține API-uri pentru comunicarea cu frontend-ul.

Pentru a exista o comunicare între frontend și backend, pentru fiecare acțiune a utilizatorului se trimit cereri de tip HTTP către server, iar ca răspuns, clientul va primi date în format JSON.

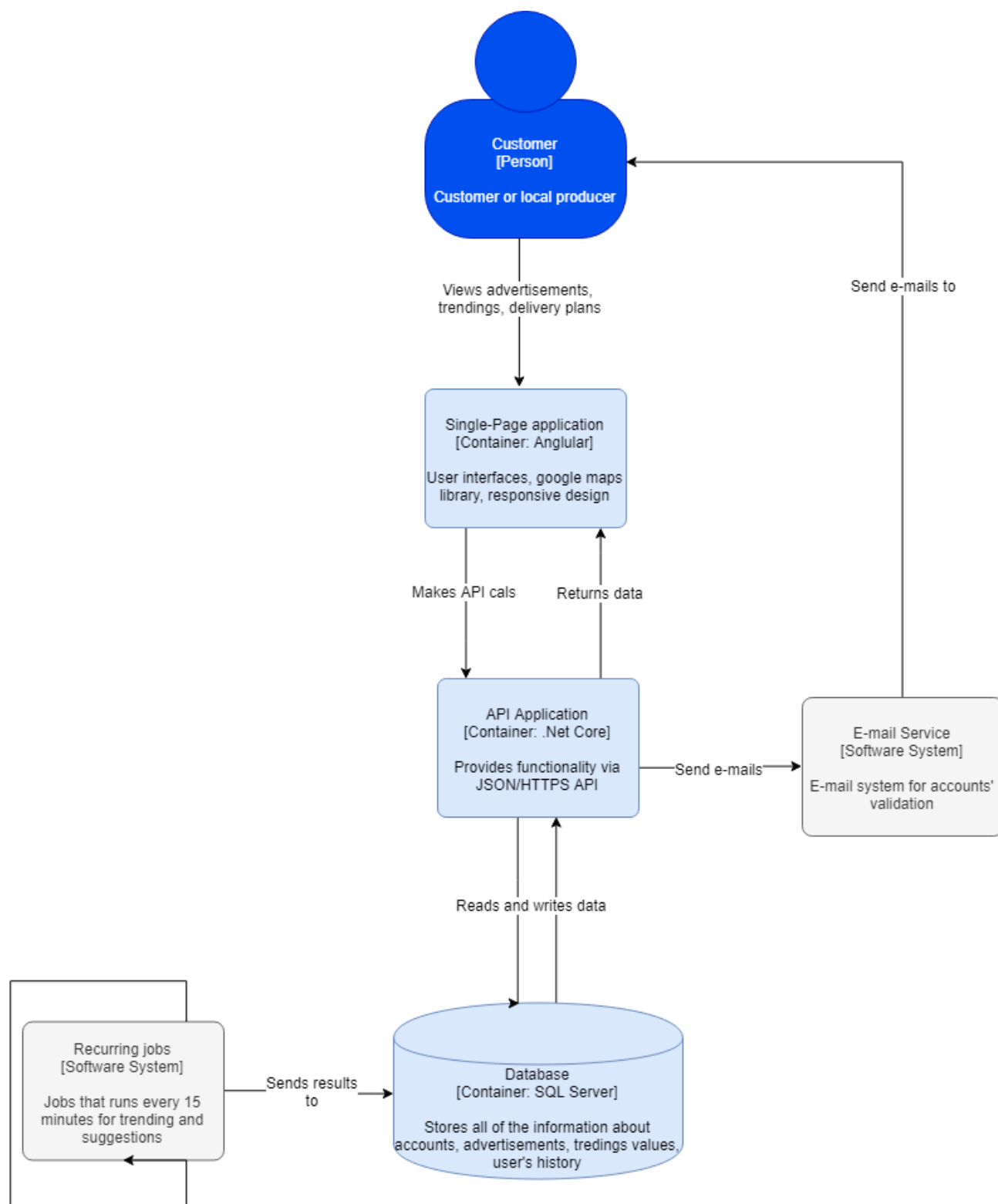


Fig. 2 - Arhitectura frontend-ului și a backend-ului

1.2 Arhitectura bazei de date

Baza de date folosește tehnologia SQL Server pentru a stoca informațiile primite de la server și conține trei părți principale: tabelele și relațiile necesare stocării detaliilor anunțurilor, cele pentru gestionarea planurilor de livrare și tabelele destinate conversațiilor dintre ceilalți utilizatori.

Anunțurile producătorilor sunt stocate în tabela *Advertisements*, ce conține toate informațiile necesare cât și relațiile dintre celelalte tabele, precum *AdvertisementsHistory* sau *Categories*. Tabela *AdvertisementsHistory* conține istoricul utilizatorilor pentru a putea afla care sunt anunțurile în tendințe sau ce produse vor fi sugerate atunci când se accesează aplicația.

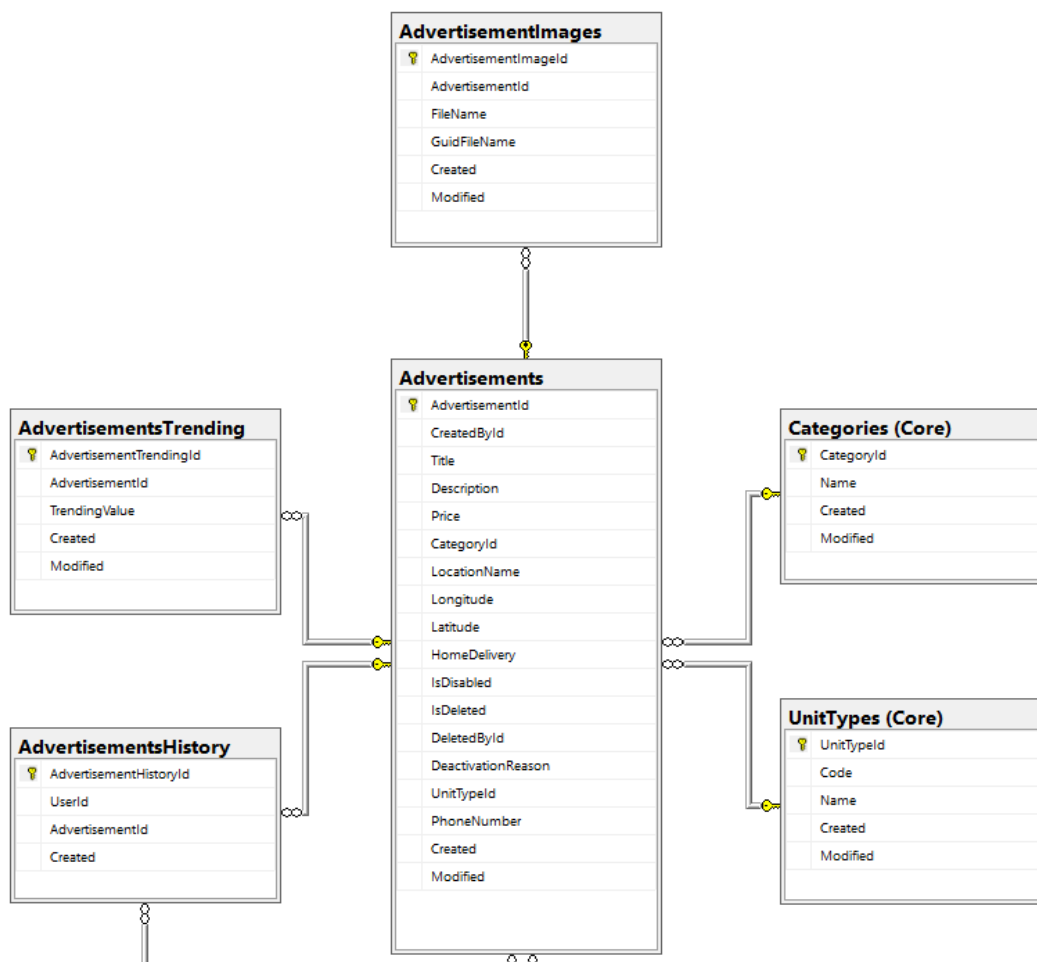


Fig. 3 - Tabelele folosite pentru stocarea anunțurilor

A doua parte principală a bazei de date stochează informațiile livrărilor gestionate de vânzători. Tabela *HomeDeliveryPoints* are rolul de a prelua livrările la domiciliu de la server, iar tabela *LocationDeliveryPoints* stochează detaliile livrărilor în anumite puncte din interiorul orașelor.

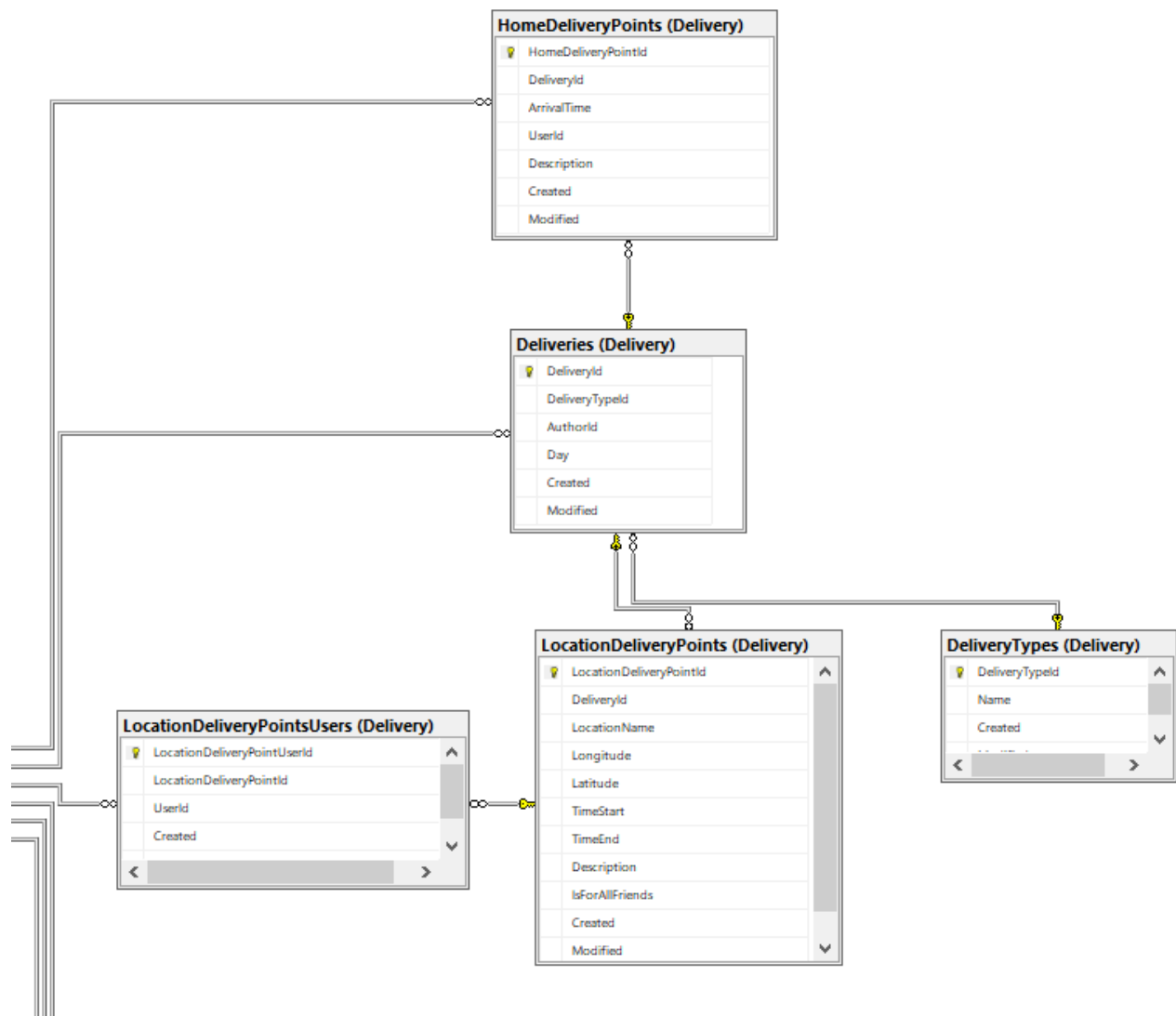


Fig. 4 - Tabelele folosite pentru stocarea planurilor de livrare

Pentru a se putea realiza o comunicare între utilizatori, baza de date conține tabele precum *Chats* (ce are ca scop crearea unei camere virtuale), o tabelă de relaționare *ChatUser* (ce face legătura dintre un utilizator și o cameră creată), și tabela *Messages* (ce conține detaliile fiecărui mesaj transmis în chat).

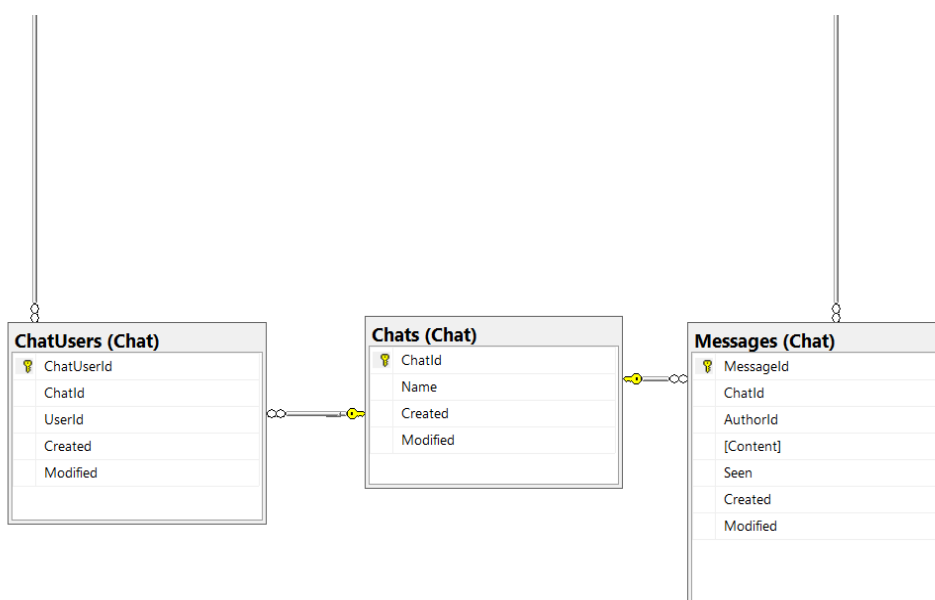


Fig. 5 - Tabelele folosite pentru stocarea conversațiilor

1.3 Concluzii

Capitolul 1 detaliază arhitectura aplicației împreună cu tehnologiile folosite. În această secțiune, am prezentat arhitectura aplicației, structura frontend-ului și cea a backend-ului, dar și diagrama bazei de date. Pentru fiecare arhitectură în parte, am adăugat imagini sugestive.

Capitolul 2. Descrierea aplicației

HeathyFood este o aplicație ce vine în ajutorul producătorilor locali de alimente care vor să își promoveze anunțurile inteligent pe o piață destinată consumului de produse naturale, ajută clienții să găsească produse alimentare într-un mod simplu, pe baza locației acestora și oferă suport micilor antreprenori ce fac livrări în anumite puncte ale orașelor sau direct la domiciliul clienților.

În această aplicație fiecare producător își poate promova local produsele specificând locația anunțului pe harta Google Maps. Producătorul are la dispoziție o pagină unde poate posta anunțuri, își detaliază produsele, adaugă poze sugestive și poate stabili locația anunțului, fie automat (prin poziția sa curentă), fie manual prin identificarea unei locații. Căutarea acestor anunțurilor se bazează pe distanța calculată dintre coordonatele anunțului și cele ale clientului. Prin această funcționalitate se îmbunătățește calitatea rezultatelor oferite și se evită afișarea produselor ce sunt indisponibile zonei clientului.

Clienții pot vizualiza anunțurile ce se află în tendințe, pentru perioada curentă a anului, pe baza unui algoritm. Deoarece ofertele producătorilor se schimbă de la o perioadă la alta a anului, aplicația folosește un algoritm ce stabilește un top al cele mai căutate anunțuri pentru perioada curentă. Clienții vor putea vizualiza produsele cele mai căutate de ceilalți utilizatori și, în același timp, se vor inspira în a găsi alte categorii de produse ce sunt la mare căutare.

Aplicația oferă un suport bun în a propune sugestii de căutare utilizatorilor. De exemplu, atunci când un client caută un produs în secțiunea de căutare, va primi sugestii de tip autocompletare. Primele rezultate vor fi cele din istoricul căutărilor utilizatorului, urmate de cele care sunt în tendințe în acel moment. Suportul de sugestii pentru căutare folosește un algoritm ce sortează căutările descrescător, după numărul de accesări din partea altor utilizatori.

Pentru fiecare utilizator ce accesează platforma, pe baza activității anterioare se va stabili o listă de sugestii de anunțuri ce vor fi afișate. În acest fel, clienții vor găsi întotdeauna produse relevante și vor descoperi anunțuri noi, similare cu cele căutate anterior din categoriile ce prezintă interes.

Fiecare vânzător deține o pagină de prezentare pe care o poate edita și personaliza. În această pagină, producătorul își poate încărca o poză de tip copertă și își poate detalia activitatea

sa. Atunci când alți utilizatori accesează pagina producătorului, vor putea vedea, în plus, numărul de anunțuri adăugate, numărul de persoane apropiate ale acestui vânzător, locația sa unde poate fi găsit în cazul în care acesta nu face livrări, dar și data în care s-a înregistrat în aplicație. În această pagină se poate vedea ratingul vânzătorului, împreună cu recenziile primite de la ceilalți utilizatori.

Aplicația pune la dispoziție un chat unde utilizatorii pot purta o conversație. În acest chat se poate vedea dacă utilizatorul căutat este online sau, în caz contrar, data când acesta a fost activ ultima oară în aplicație. De asemenea, aplicația conține și un istoric al mesajelor fiecărei conversații și oferă utilizatorului conectat notificări atunci când primește mesaje noi.

Utilizatorul poate administra un grup de persoane de încredere, în care adăugă sau șterge alți utilizatori. Scopul acestui grup este de a furniza locația de livrare a utilizatorilor doar persoanelor de încredere. De exemplu, un vânzător poate vedea pentru toate persoanele din grupul acestuia locația lor de livrare, iar clienții pot vedea coordonatele producătorului în cazul în care acesta nu face livrări.

Pentru vânzătorii ce pot face livrări, aceștia își pot gestiona planurile de livrare direct din aplicație. În acest fel, clienții vor fi informați automat atunci când livrarea comenzii va fi efectuată. Aplicația pune la dispoziție două tipuri de a gestiona livrările. Primul tip este livrarea la domiciliu: vânzătorul își creează un plan de livrare și va adăuga persoanele corespunzătoare, va seta data, respectiv ora la care produsul va ajunge la fiecare client. Al doilea tip este cel de livrare la un punct de pe hartă: vânzătorul își poate face planul cu locațiile pe hartă, unde va livra în intervalele orare de timp specificate. Astfel, clienții vor vedea lista persoanelor ce urmează să facă livrări în ziua respectivă, împreună cu punctele aferente de pe hartă.

Utilizatorul își poate gestiona setările generale ale contului precum poza de profil, informațiile personale, parola de acces în aplicație dar și locațiile furnizate de acesta. Utilizatorul poate gestiona două locații. Prima locație reprezintă poziția acestuia pe hartă atunci când aplicația nu poate folosi coordonatele curente, iar a doua reprezintă adresa de livrare vizibilă pentru ceilalți vânzători.

O altă funcționalitate oferită de aplicație este aceea de a notifica utilizatorii prin e-mail sau prin notificare directă în cadrul platformei atunci când o persoană apropiată din grupul utilizatorului postează un anunț nou sau face livrări.

2.1 Înregistrare prin validarea e-mail-ului

Atunci când accesează aplicația, utilizatorii se pot autentifica sau înregistra, dacă nu au deja un cont creat. Înregistrarea în aplicație se realizează prin parcurgerea cu succes a doi pași.

Primul pas constă în completarea unui formular ce conține următoarele câmpuri: nume, prenume, e-mail și parolă. Atunci când utilizatorul apasă butonul de *Register*, se trimite un obiect către server cu datele introduse. Serverul validează informațiile primite și apelează serviciul destinat înregistrării utilizatorilor.

```
if (model == null)
{
    return BadRequest("Invalid user model");
}

Result<UserName> firstName = UserName.Create(model.FirstName);
Result<UserName> lastName = UserName.Create(model.LastName);
Result<Email> email = Email.Create(model.Email);
Result<Password> password = Password.Create(model.Password);

Result result = Result.Combine(firstName, lastName, email, password);
if (result.IsFailure)
{
    return BadRequest(result.Error);
}
```

Fig. 6 – Crearea unui nou utilizator

Pentru a fi confirmată parola introdusă, aceasta trebuie să conțină, în total, minimum opt caractere, dintre care cel puțin unul să fie caracter special și cel puțin unul să fie o cifră. În cadrul serviciului de înregistrare, se formează o nouă parolă de tip *hash*, prin adăugarea unui șir de litere generat ca prefix și criptarea parolei prin procedeul funcției *hash*. Se folosește obiectul *Rfc2898DeriveBytes* ce primește ca parametri parola în formatul *string*, șirul de litere generat, numit *salt*, și numărul de iterații ce vor fi executate atunci când se generează noua parolă.

Al doilea pas presupune validarea adresei de e-mail specificate de utilizator și accesarea link-ului din conținutul e-mail-ului primit. Utilizatorul va fi redirecționat către pagina de confirmare a e-mail-ului, unde trebuie să selecteze tip de utilizator (client sau producător), dar și să specifice locația acestuia în cazul în care, pe viitor, nu va putea folosi funcția de localizare automată. După acest pas, utilizatorul va fi redirecționat către pagina de autentificare.

Când un utilizator încearcă să se autentifice în aplicație prin e-mail și parolă, pentru validarea parolei, se extrage șirul generat inițial, ce corespunde utilizatorului, se generează noua parolă de tip *hash* și se verifică dacă rezultatul corespunde cu ceea ce există deja în baza de date.

2.2 Autentificare și autorizare

Fiecare utilizator va avea un rol în aplicație, fie vânzător, fie client. Atunci când acesta se autentifică, după validarea parolei, se face o verificare a rolului și se generează un *token* unic, virtual, cu ajutorul unei chei private. Acest *token* este numit *JSON Web Token* și este folosit ca metodă de securitate pentru API-urile expuse. La generarea acestuia, se introduc informațiile utilizatorului ce vor fi folosite de aplicația Angular, precum numele, prenumele, rolul sau e-mail-ul. După ce a fost primit ca răspuns al cererii HTTP, *tokenul* va fi stocat în secțiunea *local storage* al browser-ului curent și va fi introdus automat în fiecare cerere HTTP, urmând să fie extras și verificat de serverul aplicației.

Pentru autorizarea API-urilor, s-a implementat un mecanism ce verifică rolul utilizatorului din *tokenul* primit și decide dacă acesta are acces la informațiile oferite de API. În caz contrar, se va returna codul 403, ca răspuns al cererii.

2.3 Adăugarea unui anunț

Vânzătorii au la dispoziție funcționalitatea de a adăuga anunțuri cu privire la produsele acestora. În această pagină, producătorii pot completa titlul, descrierea, numărul de telefon, prețul și locația anunțului. Se pot adăuga până la cinci poze, fiecare imagine având dimensiunea maximă de 5MB.

Pentru o vizualizare mai bună a poziției anunțului, se folosește harta obținută de la integrarea cu Google Maps. Atunci când utilizatorul accesează această pagină, în partea dreaptă se încarcă harta la coordonatele locației acestuia. API-urile Google Maps oferă avantaje, precum selectarea locației direct pe hartă sau conversia latitudinii/longitudinii la numele zonei selectate. Pentru a specifica locația anunțului, se poate selecta poziția curentă sau se poate folosi căutarea locațiilor în câmpul de căutare. În final, formularul va conține două câmpuri în plus, corespunzătoare latitudinii și longitudinii.

La trimiterea unui anunț către server, se extrage fiecare imagine din formular, se redenumeste sub forma unui șir de litere generat aleatoriu (pentru a se evita conflictele) și se salvează pe disc. În acest mod, în baza de date se va salva doar calea către imaginea anunțului și, astfel, se evită supraîncărcarea tabelelor.

```
var advertisementImage = new AdvertisementImageBlModel()
{
    FileName = image.Name,
    AdvertisementId = advertisementId,
    GuidFileName = Guid.NewGuid() + Path.GetExtension(image.Name),
};

advertisementImages.Add(advertisementImage);
var fullPath = Path.Combine(folderName, advertisementImage.GuidFileName);

byte[] bytes = Convert.FromBase64String(image.Base64.Split(',')[1]);
System.IO.File.WriteAllBytes(fullPath, bytes);
```

Fig. 7 – Stocarea imaginilor pe hard-disk

2.4 Vizualizarea produselor din apropiere

Produsele vânzătorilor sunt vizualizate pe baza distanței calculate dintre coordonatele anunțului și cele ale utilizatorului. La deschiderea paginii de căutare a produselor, aplicația Angular apelează serviciul de localizare folosit de browser-ul web, preia punctul geografic al utilizatorului și îl trimite către server. Utilizatorul poate specifica distanța maximă pentru a primi

doar anunțurile accesibile, iar serverul calculează distanțele corespunzătoare și identifică acele produse ce se încadrează în intervalul distanței specificat de utilizator.

O primă problema în dezvoltarea acestei funcționalități este aceea că se poate calcula distanța dintre două puncte doar cu ajutorul API-urilor Google Maps, ceea ce înseamnă aducerea tuturor anunțurilor în aplicația frontend și filtrarea acestora. Abordarea nu este benefică vitezei de încărcare a platformei, deoarece pot exista mii de anunțuri salvate în baza de date, ce vor încărcate pentru a fi filtrate.

Soluția acestei probleme a fost implementarea unei funcții în cadrul backend-ului, ce poate calcula o distanță aproximativă cu cea calculată de Google Maps. Pentru a optimiza viteza de încărcare a anunțurilor, am folosit o funcție la nivelul bazei de date ce primește ca parametri două puncte în plan și calculează distanța dintre acestea, luând în considerare circumferința pământului. Viteza de încărcare a fost îmbunătățită semnificativ, iar anunțurile pot fi filtrate pe partea serverului.

```

ALTER FUNCTION [Core].[GetDistance]
(
    @latitude1 DECIMAL(24, 18),
    @longitude1 DECIMAL(24, 18),
    @latitude2 DECIMAL(24, 18),
    @longitude2 DECIMAL(24, 18)
)
RETURNS DECIMAL(24, 18)
AS
BEGIN
    DECLARE @deg2rad DECIMAL(24, 18) = PI() / 180.00;
    DECLARE @dlong DECIMAL(24, 18) =
        (@longitude2 - @longitude1) * @deg2rad;
    DECLARE @dlat DECIMAL(24, 18) = (@latitude2 - @latitude1) * @deg2rad;
    DECLARE @a DECIMAL(24, 18) =
        POWER(SIN(@dlat / 2.0), 2) + COS(@latitude1 * @deg2rad) *
        COS(@latitude2 * @deg2rad) * POWER(SIN(@dlong / 2.0), 2);

    DECLARE @c DECIMAL(24, 18) = 2 * ATN2(SQRT(@a), SQRT(1-@a));
    DECLARE @d DECIMAL(24, 18) = 6367 * @c
    RETURN @d
END

```

Fig. 8 - Calcularea distanței dintre două puncte în plan⁴

⁴ Calculate distance, bearing and more between Latitude/Longitude points: [online]. [Accesat: 20 mai 2019]. Disponibil la adresa: <https://www.movable-type.co.uk/scripts/latlong.html>

Anunțurile primite de la server vor fi afișate în pagina de căutare, împreună cu prima imagine atribuită fiecărui produs. Având în vedere că o imagine poate avea o dimensiune de până la 5MB, se produce un trafic mare de date la încărcarea pozelor anunțurilor. Componenta destinată vizualizării anunțurilor conține o paginare implementată pe partea serverului, cu scopul de a nu o supraîncărca și de a permite vizualizarea a maximum 20 de anunțuri pentru fiecare secțiune. Dacă fiecare imagine ar avea o dimensiune de aproximativ 5MB, traficul de date necesar pentru vizionarea produselor ar fi de cel mult 100MB, ceea ce înseamnă un volum de date imens, mai ales pentru dispozitivele mobile.

Ca o soluție la această problemă, s-a implementat procesarea primei imagini a fiecărui anunț adăugat. Imaginea este redimensionată la 400x400 pixeli pentru a se potrivi componentei de tip card a prezentării anunțului. Această abordare scade din dimensiunea imaginii cu până la 99,8%, ajungând la o proporție de aproximativ 10KB. În acest fel, pentru o încărcare a celor 20 de anunțuri, va fi nevoie de doar 200KB de trafic de date.

```
var file = new FileInfo(cardFilePath);  
MagickImage sprite = new MagickImage(file);  
sprite.Format = MagickFormat.Jpeg;  
sprite.Quality = 30;  
sprite.Resize(400, 0);  
sprite.Write(file);
```

Fig. 9 – Prelucrarea unei imagini

2.5 Gestionarea anunțurilor

Producătorii au la dispoziție funcționalitatea de a edita sau șterge anunțurile postate. Această componentă ajută producătorii să își gestioneze afacerea prin actualizarea frecventă a anunțurilor sau ștergea acestora definitiv. Deoarece toate imaginile sunt stocate pe memoria hard-disk-ului atunci când se actualizează un anunț, în momentul în care producătorul dorește să efectueze o modificare, serverul va trimite doar adresa URL a imaginilor, și nu întreaga lor dimensiune. Atunci când vânzătorul modifică imaginile unui anunț, se vor trimite către backend doar imaginile noi adăugate, celelalte având nemodificată adresa URL.

2.6 Algoritmul pentru calcularea tendințelor

Aplicația pune la dispoziție o pagină unde utilizatorii pot vizualiza anunțurile care sunt în topul tendințelor pentru o anumită perioadă. Pentru a identifica produsele cele mai vizualizate, se înregistrează fiecare acțiune a utilizatorilor atunci când aceștia vizualizează produsele. Datele se salvează în tabela *AdvertisementsHistory*, iar fiecare linie conține id-ul utilizatorului, id-ul anunțului și data când a fost accesat respectivul anunț.

```
public Result AddAdvertisementHistory(long userId, long advertisementId)
{
    var advertisementHistory = new AdvertisementHistory()
    {
        AdvertisementId = advertisementId,
        UserId = userId
    };
    return
    _advertisementRepository.AddAdvertisementHistory(advertisementHistory);
}
```

Fig. 10 - Stocarea istoricului utilizatorilor

Algoritmul parcurge anunțurile din baza de date și extrage pentru fiecare: numărul de accesări din ultimele șase zile și numărul de vizionări din ziua curentă. Acest algoritm calculează deviația standard pe baza eșantionului de accesări anterioare. Pentru calcularea rezultatului, primul pas este adăugarea mediei acestui eșantion, într-o variabilă. Pentru pasul al doilea, se extrage fiecare număr din vector și se calculează varianța⁵ acestuia. Pasul al treilea constă în preluarea tuturor varianțelor calculate, care, mai apoi, se ridică la pătrat și, într-un final, se face suma acestora. Pentru pasul al patrulea, se stochează deviația standard în variabila *standardDeviation*, fiind rezultatul dintre împărțirea sumei calculate anterior la numărul de elemente din eșantion. La final, se calculează varianța dintre accesările din ziua respectivă și media accesărilor din eșantion, apoi se împarte la deviația standard. Rezultatul reprezintă un număr de tip *double*, iar dacă este

⁵ Varianță: media aritmetică a pătratelor abaterilor valorilor individuale ale unui șir statistic de experiențe, de la media aritmetică a șirului respectiv (*Dicționarul explicativ al limbii române*: [online]. [Accesat: 23 mai 2019]. Disponibil la adresa: <https://dexonline.ro/definitie/varian%C8%9B%C4%83>)

mai mare decât valoarea 1, se deduce că anunțul deține mai multe accesări în ziua curentă în raport cu cele zile anterioare. Dacă are valoarea mai mică decât valoarea 1, interesul celorlalți utilizatori pentru acel anunț a scăzut.

```
public class TrendingHelper
{
    public static double CalculateTrendingValue(
        long value, List<long> previousValues)
    {
        double sum = 0;

        double count = previousValues.Count;
        double average = previousValues.Average();

        foreach (var previousValue in previousValues)
        {
            sum += Math.Pow(previousValue - average, 2);
        }

        var standardDeviation = Math.Sqrt(sum / count);
        return standardDeviation == 0 ?
            (value - average) / 1 :
            (value - average) / standardDeviation;
    }
}
```

Fig. 11 - Algoritmul pentru calcularea tendințelor⁶

Deoarece numărul de vizionări ale produselor se vor schimba pe baza activității utilizatorilor, a fost implementată o metodă care rulează un fir de execuție recurent la fiecare 15 minute și are ca scop actualizarea valorilor specifice tendinței anunțului. Pentru fiecare execuție a acestui fir, se parcurg toate anunțurile din baza de date și se apelează funcția de calculare a valorii tendinței. Valoarea returnată se salvează în tabela *AdvertisementsTreding*, corespunzătoare id-ului

⁶ *Calculating standard deviation step by step*: [online]. [Accesat: 25 mai 2019]. Disponibil la adresa: <https://www.khanacademy.org/math/probability/data-distributions-a1/summarizing-spread-distributions/a/calculating-standard-deviation-step-by-step>

anunțului. Pentru afișarea produselor ce sunt în tendințe, se parcurge această tabelă și se extrag liniile ce au valoarea tendinței mai mari decât valoarea 1.

2.7 Sugestii pentru căutarea unui produs

Aplicația oferă sugestii de tip autocompletare atunci când utilizatorul introduce text în câmpul destinat căutării produselor. În momentul în care se introduc caractere în acel câmp, se trimite o cerere către server pentru a vedea ce sugestii se potrivesc cu textul introdus. Ca răspuns, va fi returnat un vector de sugestii, ce conține numele și id-ul acestor propuneri.

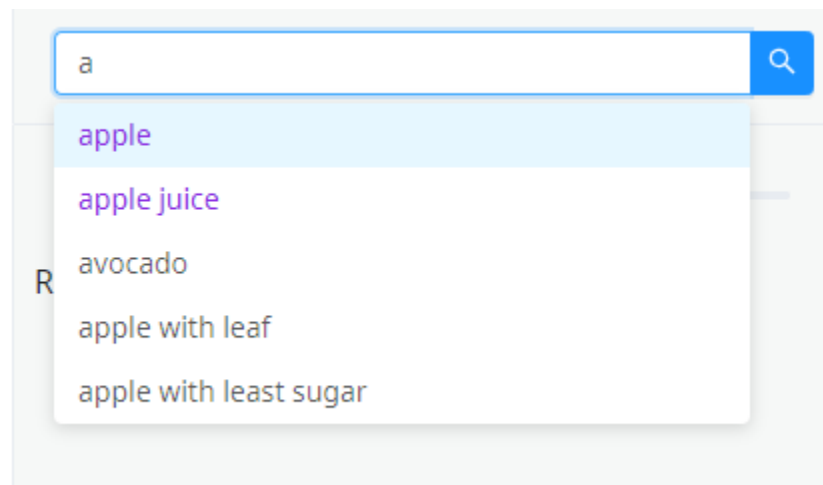


Fig. 12 – Exemplu de sugestii pentru o căutare

Serverul memorează dacă utilizatorul selectează o sugestie din cele oferite și calculează ordinea apariției sugestiilor după cele mai căutate rezultate. De exemplu, sugestia *avocado* va apărea înaintea sugestiei *apple with leaf* deoarece deține mai multe căutări anterioare. Pentru a o oferi o experiență de utilizare mai bună a utilizatorilor, aplicația oferă suport și pentru istoricul căutărilor anterioare ale unui utilizator, dar și pentru sugestiile oferite ce sunt la mare căutare în ziua respectivă, calculate cu ajutorul aceluiași algoritm prezentat anterior. Culoarea mov reprezintă sugestiile ce au mai fost căutate de utilizator, iar culoarea portocaliu reprezintă căutările ce sunt în tendințe.

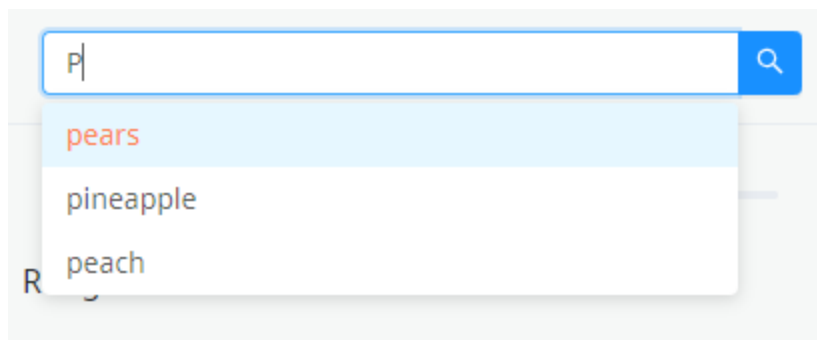


Fig. 13 – Sugestie de căutare ce se află în tendințe

2.8 Algoritmul utilizat pentru sugerarea altor produse

Utilizatorii vor primi sugestii atunci când accesează pagina de căutare. Sugestiile sunt calculate în raport cu activitatea acestora. Atunci când utilizatorul accesează pagina principală, se trimite o cerere către server. Backend-ul primește cererea utilizatorului și calculează, pe baza istoricului memorat, cele mai vizionate categorii de produse. Algoritmul primește istoricul utilizatorului și returnează o listă de categorii. Se extrage o serie de produse și se verifică dacă au fost sau nu vizionate anterior. În cazul în care utilizatorul nu a accesat anterior anunțul, va fi adăugat în lista pentru sugestii.

2.9 Comunicarea cu ceilalți utilizatori

Platforma conține o pagină destinată comunicării între utilizatori. Atunci când se va selecta o conversație, se poate vedea dacă destinatarul este online, se pot vedea mesajele trimise anterior și se pot vizualiza, în dreapta, paginii, detaliile conversației. Aplicația poate memora până la 50 de mesaje trimise între utilizatori. Pentru a se realiza o comunicare instantanee între utilizatori s-a folosit o conexiune de tip *Web Socket*.

În momentul în care o conversație se încarcă, serverul primește o cerere de conectare la canalul mesajelor, aceasta conținând id-ul conexiunii și id-ul utilizatorului. Acest serviciu include două funcții ce sunt utilizate în aplicație: prima funcție preia toate cererile de înscriere ale

utilizatorilor în canal, iar a doua funcție este folosită cu scopul de a monitoriza conexiunile care se deconectează de la acest canal. Toate aceste informații se stochează în baza de date, în tabela *UserConnections*.

```
public override async Task OnConnectedAsync()
{
    int userId = int.Parse(Context.User.Claims
        .First(item => item.Type == "userId").Value);
    RegisterConnection(userId, Context.ConnectionId);

    var userConnectionsResult = _messageRepository.GetUsersPartners(userId);
    foreach (var connection in userConnectionsResult.Value)
    {
        var userIdResult =
            _messageRepository.GetUserIdByConnectionId(connection);
        await GetChatsByUserId(userIdResult.Value, connection);
    }

    await base.OnConnectedAsync();
}
```

Fig. 14 – Funcția care primește cererile de înscriere în canal

```

public override async Task OnDisconnectedAsync(Exception exception)
{
    _messageRepository.UpdateUserConnectionOffline(Context.ConnectionId);
    var userIdResult =
        _messageRepository.GetUserIdByConnectionId(Context.ConnectionId);

    var chats = _messageRepository.GetChatsForUserId(userIdResult.Value);
    foreach (var chat in chats.Value)
    {
        await Groups.RemoveFromGroupAsync(Context.ConnectionId, chat.Name);
    }

    // notify all users about this event
    var userConnectionsResult =
        _messageRepository.GetUsersPartners(userIdResult.Value);
    foreach (var connection in userConnectionsResult.Value)
    {
        var userId2Result =
            _messageRepository.GetUserIdByConnectionId(connection);
        await GetChatsByUserId(userId2Result.Value, connection);
    }

    await base.OnDisconnectedAsync(exception);
}

```

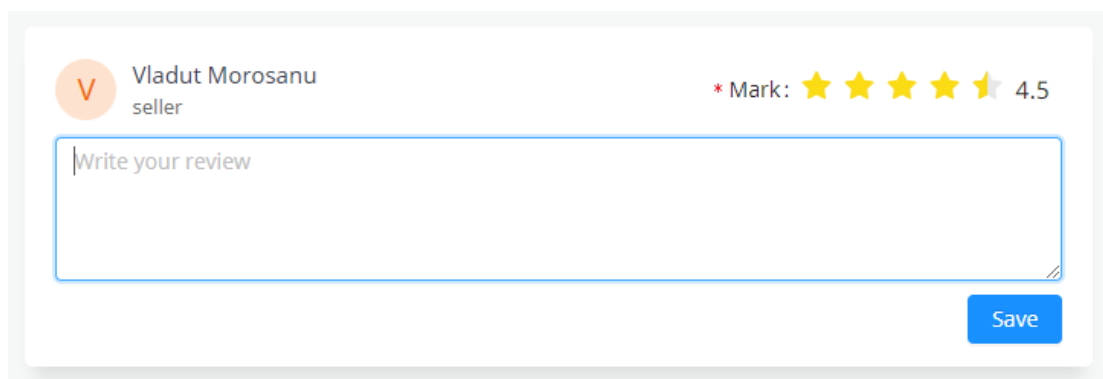
Fig. 15 – Funcția apelată la întreruperea unei conexiuni

2.10 Pagina de prezentare a unui vânzător

Fiecare producător deține o pagină de prezentare unde își poate edita activitatea și își poate schimba poza de copertă a profilului. În momentul în care un client vizualizează pagina de prezentare a unui vânzător, vor fi prezente și informațiile despre acesta, precum: numărul anunțurilor adăugate, data înscrierii în aplicație dar și locația sa. Serverul salvează pe hard-disk imaginea primită în momentul în care utilizatorul își actualizează pagina.

2.11 Secțiune de vizualizare și adăugare de recenzii

Fiecare utilizator poate primi recenzii din partea persoanelor cu care a interacționat. Atunci când utilizatorul scrie o recenzie altei persoane, acesta trebuie să specifice nota recenziei (o valoare între 1 și 5) și un mesaj sugestiv prin care exemplifică motivul alegerii notei. Recenziile sunt vizibile pe pagina fiecărui utilizator. Aceasta funcționalitate are rolul de a filtra anunțurile producătorilor ce au primit recenzii negative, dar și de a oferi informații despre activitatea producătorului.



The image shows a user interface for adding a review. At the top left, there is a profile icon with the letter 'V' and the name 'Vladut Morosanu' with the role 'seller' below it. To the right, it says '* Mark:' followed by five yellow stars, the last one being half-filled, and the number '4.5'. Below this is a large text input area with the placeholder text 'Write your review'. At the bottom right of the input area is a blue button labeled 'Save'.

Fig. 16 – Adăugarea unei recenzii

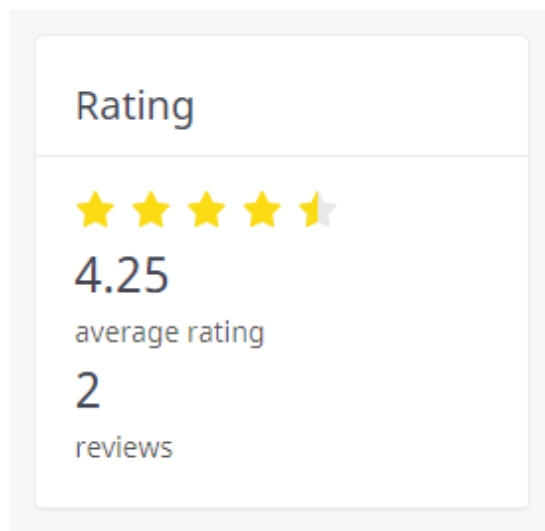


Fig. 17 – Vizualizarea mediei notelor ale unui utilizator

2.12 Grupul de persoane de încredere

Fiecare utilizator își poate gestiona grupul de persoane de încredere. Acesta poate trimite cereri de înscriere în grupul altor persoane și poate accepta sau șterge cererile primite din partea altor utilizatori. Această funcție are rolul de a securiza vizualizarea datelor utilizatorilor din partea persoanelor necunoscute. De exemplu: un producător poate vedea doar adresele de livrare a persoanelor din grupul acestuia.

2.13 Gestionarea planurilor de livrare

Producătorii își pot planifica livrările solicitate în pagina *Add delivery*. Aici, pagină aceștia își pot adăuga două tipuri de livrare: livrare la adresa clienților și livrare în mai multe puncte ale orașelor.

Prin selectarea planului de livrare la adresa clienților, se poate crea o listă de persoane a căror livrări vor fi efectuate în ziua selectată de producător. Pentru fiecare persoană adăugată în planul de livrare, se folosește adresa de livrare a clienților, dar se va specifica descrierea comenzii și ora la care va sosi vânzătorul. În momentul în care producătorul salvează planul de livrare, clientul va fi informat prin notificare, în cadrul aplicației, și prin e-mail.

Al doilea plan de livrare constă în adăugarea a mai multor puncte pe hartă și specificarea orelor în care vânzătorul va fi prezent în locul respectiv. Pentru fiecare punct de livrare, acesta poate invita toate persoanele din grupul acestuia sau poate invita doar anumite persoane. Atunci când se va salva livrarea, ceilalți clienți vor vedea punctele de pe harta în care vânzătorul va face comerț.

```
public Result CreateHomeDelivery(CreateHomeDeliveryBlModel model)
{
    var delivery = model.GetDeliveryDataModel();
    var deliveryResult = _deliveryRepository.CreateDelivery(delivery);
    if (deliveryResult.IsFailure)
    {
        return Result.Fail(deliveryResult.Error);
    }
    ...
}
```

Fig. 18 – Adăugarea unui plan de livrare

2.14 Administrarea contului utilizatorului

În pagina *Settings*, fiecare utilizator își poate edita date personale, își poate schimba imaginea de profil, poate schimba parola, își poate administra adresele înregistrate în aplicație sau poate seta dacă acesta dorește să primească e-mail-uri. Fiecare utilizator deține două adrese, prima adresa reprezintă adresa care este folosită în calcularea distanțelor (în cazul în care acesta nu are acces la locația lui curentă) și a doua reprezintă adresa de livrare ce va fi vizualizată de ceilalți membri ai grupului acestuia.

2.15 Concluzii

Acest capitol are rolul de a crea o imagine de ansamblu asupra funcționalităților aplicației. Fiecare subcapitol descrie o funcționalitate, pașii parcurși pentru dezvoltare și rezultatul final.

Bibliografie

Calculate distance, bearing and more between Latitude/Longitude points: [online]. [Accesat: 20 mai 2019]. Disponibil la adresa: <https://www.movable-type.co.uk/scripts/latlong.html>

Calculating standard deviation step by step: [online]. [Accesat: 25 mai 2019]. Disponibil la adresa: <https://www.khanacademy.org/math/probability/data-distributions-a1/summarizing-spread-distributions/a/calculating-standard-deviation-step-by-step>