

UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

**Foodbucks – Aplicație android de rețete
culinare cu filtrare după preț**

propusă de

Dănuț-Petru Puiu

Sesiunea: iulie, 2019

Coordonator științific

Lect. Dr. Alex Mihai Moruz

UNIVERSITATEA "ALEXANDRU-IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ

**Foodbucks – Aplicație android de rețete
culinare cu filtrare după preț**

Dănuț-Petru Puiu

Sesiunea: iulie, 2019

Coordonator științific

Lect. Dr. Alex Mihai Moruz

Avizat,
Îndrumător lucrare de licență,
Lect. Dr. Alex Mihai Moruz.

Data: Semnătura:

Declarație privind originalitatea conținutului lucrării de licență

Subsemnatul **Puiu Dănuț-Petru** domiciliat în **România, jud. Iași, comuna Țuțora, sat Țuțora**, născut la data de **12 iunie 1996**, identificat prin CNP **1960612226751**, absolvent al Facultății de informatică, **Facultatea de informatică** specializarea **informatică**, promoția 2019, 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 **Foodbucks – Aplicație android de rețete culinare cu filtrare după preț** elaborată sub îndrumarea domnului **Lect. Dr. Alex Mihai Moruz**, pe care urmează să o susțin 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 ei 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 diplomă 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:

Semnătura:

Declarație de consimțământ

Prin prezenta declar că sunt de acord ca lucrarea de licență cu titlul **Foodbucks – Aplicație android de rețete culinare cu filtrare după preț**, 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.

Absolvent **Dănuț-Petru Puiu**

Data:

Semnătura:

Cuprins

Introducere	2
Contribuții	4
1 Aplicații similare	5
1.1 Tasty	5
1.2 Recipe Cost Calculator	6
1.3 Yummly	7
2 Descrierea soluției	8
2.1 Unelte software folosite	8
2.2 Descrierea implementării	9
2.2.1 Backend	9
2.2.2 Frontend	15
2.3 Probleme întâmpinate și soluțiile acestora	18
3 Ghidul utilizatorului	19
3.1 Navigarea rețetelor	19
3.2 Adăugarea de rețete noi	25
3.3 Navigarea produselor	26
3.4 Navigarea magazinelor	27
Concluzii	29
Bibliografie	31

Introducere

Gătitul, arta prin care se prepară mâncarea pentru a fi mai ușor și mai savuros consumată, a evoluat în moduri inedite în decursul istoriei. În timpurile preistorice se gătea folosind doar ingrediente ce se găseau deja în natură, fie ele legume, fructe sau condimente naturale, și ingrediente ce proveneau de la animalele vâdate. Cu trecerea timpului și cu introducerea agriculturii și a comerțului, a crescut treptat disponibilitatea ingredientelor prin cultivarea, negoțul sau cumpărarea acestora. Astfel, gătitul a devenit un meșteșug din ce în ce mai simplu de executat. Odată cu revoluția industrială din secolul 19, s-a observat o îndepărtare tot mai mare de ideea de agricultură pentru consumul propriu și apropierea mai mare de agricultura pe scară largă, astfel că oamenii de rând au ajuns, în prezent, în foarte mare măsură să se bazeze pe ingredientele cumpărate. În același timp și din aceleași motive, gătitul a devenit și el o activitate tot mai puțin practică, iar comandatul acasă sau la birou de mâncare a devenit mai eficient din punct de vedere al prețului și al timpului. Astfel, unul dintre factorii importanți în alegerea dintre a găti sau a comanda mâncarea, iar mai apoi, dacă se alege prima dintre acțiuni, ce rețetă mai exactă să fie gătită este prețul necesar pentru executarea rețetei. Un alt factor este timpul pe care rețeta îl necesită.

Lucrarea de față își propune crearea unei aplicații ce le permite utilizatorilor să filtreze și să sorteze rețetele culinare după prețul estimativ pe care acestea îl presupun, luând în considerare tipul fiecărui ingredient și cantitatea acestuia. Acest lucru va fi posibil prin stocarea informațiilor mai multor produse generice, existând posibilitatea ca aceste produse să se afle în mai multe magazine simultan, chiar cu prețuri diferite. Aplicația va încerca totdeauna să ofere prețul optim, considerat ca prețul cel mai mic, al ingredientelor folosite. Desigur, aplicația va pune la dispoziție și metode de filtrare și sortare a rețetelor după ceilalți factori ce le compun, cum ar fi, de exemplu, numele sau timpul de gătit.

Aplicații populare ce există deja pe piață, ce reprezintă unelte ușor de folosit în

scopul gătitului rețetelor culinare, sunt Yummly și Tasty, însă niciuna dintre acestea nu oferă informații legate de prețul estimativ pe care îl presupun rețetele, implicit nici filtrare sau sortare după preț. Motivația din spatele alegerii acestei teme o reprezintă o problemă întâmpinată atunci când, folosind oricare dintre aceste aplicații nu am găsit această funcționalitate, cu toate că factorul prețului este unul foarte important în alegerea rețetelor culinare pentru a fi executate.

În consecință, lucrarea de față își propune implementarea unei aplicații android și a unui Web API ce, lucrând împreună, vor permite adăugarea rețetelor, răsfoirea și citirea acestora pe baza mai multor criterii.

Contribuții

Ideea de bază a acestui proiect nu este una originală, ci mai mult reprezintă o fuziune între două idei care au deja foarte multe implementări populare: aplicație de rețete culinare și aplicație de comparare a prețurilor. Cu toate acestea, pașii aleși pentru conceperea, modelarea și implementarea acestui proiect reprezintă idei personale. Am folosit diferite unelte și librării în cadrul dezvoltării proiectului, fără de care reușita acestuia ar fi generat mult mai multe probleme. Dintre aceste unelte, voi enumera Volley, care facilitează conectarea aplicației android la server, Ngrok, care expune serverul pe rețea și Entity Framework, care permite foliosirea abordării „code-first” în modelarea și menținerea bazei de date. Însă, structura entităților, algoritmi de calculare a prețurilor ingredientelor, a costurilor rețetelor pe partea de server și de creare dinamică a elementelor interfeței grafice împreună cu modalitățile de interpretare a răspunsurilor împreună cu construire cererilor HTML au reprezentat contribuții proprii.

Prin realizarea acestui proiect, am aplicat cunoștințele acumulate în cadrul studiului cu privire la limbajul Java, platforma Android, framework-ul .Net Core, limbajul c și multe alte noțiuni teoretice, pe care le-am executat în mod practic. Prin această muncă, mi-am lărgit cunoștințele în aceste domenii și mi-am dezvoltat abilitățile ce țin de crearea unei aplicații android, împreună cu server-ul web de care depinde.

Capitolul 1

Aplicații similare

După cum este de așteptat, pe piață există deja în momentul de față o mulțime de aplicații ce au ca scop navigarea și filtrarea rețetelor. Aplicația propusă fiind implementată folosind platforma Android, aplicațiile similare cu care va fi aceasta comparată vor fi tot din aceeași platformă. O simplă căutare în magazinul de aplicații Android "Play Store" după cuvântul cheie "recipes" va returna un număr de cel puțin 250 de rezultate. Din toate acestea, vor fi analizate trei exemple de aplicații care încearcă să abordeze, într-un fel sau altul, problema propusă în introducere: Tasty, Recipe Cost Calculator și Yummly.

1.1 Tasty

Aplicația Tasty, dezvoltată de către BuzzFeed, este una dintre cele mai populare aplicații de rețete disponibile în magazinul de aplicații Android "Play Store", numărând peste un milion de descărcări, iar un factor foarte important căruia îi poate fi datorată această popularitate este prezența clipurilor de prezentare a tuturor celor peste 3000 de rețete originale Tasty.

Această aplicație poate fi comparată cu produsul software propus în această lucrare prin faptul că ambele implementări oferă metode de a filtra și de a da calificative rețetelor. Aceste două funcții, cu toate că sunt simple, sunt foarte bine optimizate în cadrul aplicației Tasty, oferind o interfață extrem de intuitivă. Pe lângă acestea, rețetele sunt împărțite și pot fi filtrate pe criterii originale, față de abordarea prezentată în lucrare. Aceste categorii sunt create dinamic, în funcție de trend-urile curente. Exemple de aceste categorii sunt "Distractive la petreceri", unde sunt rețete pentru aperitive și

alte feluri potrivite pentru un anumit tip de evenimente, o alta categorie este "Treabă Nacho", în care se pot explora diferite rețete legate de nacho.

În concluzie, aplicația Tasty este mult mai restrânsă în numărul de funcționalități și nu oferă niciun fel de interpretare a prețului ingredientelor sau al rețetelor, însă este mult mai dezvoltată la capitolul opțiunilor de filtrare și vizualizare al acestora.

1.2 Recipe Cost Calculator

Aplicația Recipe Cost Calculator, dezvoltată de către Nooglesoft, este o aplicație care nu are o foarte mare popularitate, însă iese în evidență prin posibilitatea returnării prețului estimativ al rețetelor, în funcție de cantitatea și tipul ingredientelor ce le compun. Aceasta este, însă, singura funcționalitate disponibilă.

În esență, utilizatorul este singurul care adaugă date și apoi accesează date Recipe Cost Calculator. Funcțiile disponibile sunt:

- adăugarea manuală de ingrediente, specificând cantitatea, unitatea de măsură și prețul fiecăruia
- adăugarea manuală a rețetelor ce trebuie să folosească doar ingredientele disponibile
- vizualizarea ingredientelor și rețetelor adăugate
- crearea listelor de cumpărături, având ca dată de intrare o rețetă și luând în considerare ingredientele folosite

În afară de aceste funcționalități, aplicația nu pare să aibă nevoie de conexiune la internet și implicit la un server pentru a funcționa, ceea ce reprezintă un plus. Însă, acest plus se transformă într-un dezavantaj luând în considerare faptul că aplicația nu poate să calculeze prețurile, decât bazându-se pe datele primite de la utilizator.

În concluzie, Recipe Cost Calculator implementează un mod prin care ingredientele necesare unei rețete să poată fi cumpărate mai ușor, însă încearcă să rămână cât mai separată de lucrul cu entități de tip "rețetă".

1.3 Yummly

Aplicația Yummly, creată de către dezvoltatorul omonim, este cea mai populară aplicație de rețete din "Play Store", înregistrând un număr de peste 5 milioane de descărcări. În cele 3 panouri pe care le prezintă, "Just for you", "Explore" și "Store", această oferă posibilitatea utilizatorilor de a vizualiza rețete bazate pe preferințe anterioare, de a explora, după anumite criterii, cele peste 2 milioane de rețete disponibile și, respectiv, de a cumpăra ustensile de bucătărie oferite de brand-ul propriu al aplicației.

Printre cele mai importante funcționalități oferite de Yummly, se pot enumera:

- prezentarea rețetelor într-un mod personalizat, pe baza preferințelor utilizatorului și al tendințelor arătate de acesta în cadrul aplicației
- afișarea detaliilor rețetelor, schițate într-un mod eficient și ușor de înțeles
- posibilitatea adăugării oricărui ingredient într-o listă de cumpărături integrată în aplicație, iar, de asemenea, se poate adăuga la lista de cumpărături o listă de ingrediente care îi corespunde unei anumite rețete, cu un singur click
- lista de cumpărături poate fi apoi exportată automat către aplicația Instacart, de unde se vor putea cumpăra ingredientele descrise listă, luând în considerare și cantitatea de care este nevoie

Din punctul de vedere al legăturii dintre conceptele de ingrediente din Yummly și conceptele de produse din Instacart, aceasta este aplicația deja valabilă pe piață care se apropie cel mai mult de conceptul produsului software propus în lucrarea de față. Implementarea Yummly nu oferă niciun fel de indiciu sau posibilitate de filtrare a rețetelor după prețul estimativ pe care acestea îl pot avea după exportarea listei de ingrediente în Instacart.

În concluzie, Yummly implementează, în esență tot procesul gătitului, de la găsirea unei rețete interesante, pe baza celor care au fost apreciate anterior, la cumpărarea ingredientelor, folosind aplicația ajutoare, și cumpărarea ustensilelor folosind funcționalitatea disponibilă în cadrul aplicației.

Capitolul 2

Descrierea soluției

Soluția propusă în cadrul acestei lucrări reprezintă implementarea unui ansamblu format dintr-un server web și o aplicație android. Server-ul web va asigura păstrarea datelor, va implementa o serie de funcții prin care aceste date vor putea fi procesate, și va expune un API prin care funcțiile vor putea fi apelate. Aplicația android va funcționa ca interfață grafică pentru utilizatori și va pune la dispoziție panouri prin care informațiile vor putea fi consumate într-un mod ușor de înțeles.

2.1 Unelte software folosite

Server-ul web este implementat în limbajul de programare C#, cu ajutorul .NET Core 2 Framework și în special cu ajutorul Entity Framework Core. A fost ales acest mod de implementare datorită mai multor factori, dintre care cei notabili sunt:

- avantajul faptului că .NET Core este open source și cross-platform
- are parte de actualizări dese și folositoare din partea dezvoltatorilor de la Microsoft
- integrarea fără sudură cu IDE-ul Visual Studio, care are sintetizat managerului de pachete NuGet, ce face foarte simplă adăugarea pachetelor în cadrul proiectului și administrarea dependențelor
- performanța ridicată a acestui cadru, care, conform testelor realizate de Microsoft, este cel mai rapid, din punctul de vedere al compilării, de pe piață

Implementarea interfeței grafice este în limbajul Java, pentru a fi rulată pe sistemul de operare Android, iar IDE-ul folosit este Android Studio. Platforma aceasta a

fost aleasă pentru că Android este cel mai popular sistem de operare pentru mobile de pe piață și datorită suportului oferit de comunitatea acestuia.

2.2 Descrierea implementării

2.2.1 Backend

Implementarea server-ului web a fost făcută pe baza modelului arhitecturii 3-Tier, care presupune existența a 3 niveluri separate, unul pentru date, unul pentru logică și unul pentru prezentare. Aceste trei niveluri, care sunt efectiv 3 proiecte diferite, trebuie să poată lucra cât mai independent posibil față de celelalte, astfel încât să se poată fi făcute modificări asupra unui nivel, fără a schimba în vreun fel codul din celelalte niveluri, iar în același timp trebuie să lucreze împreună pentru a asigura funcționarea aplicației.

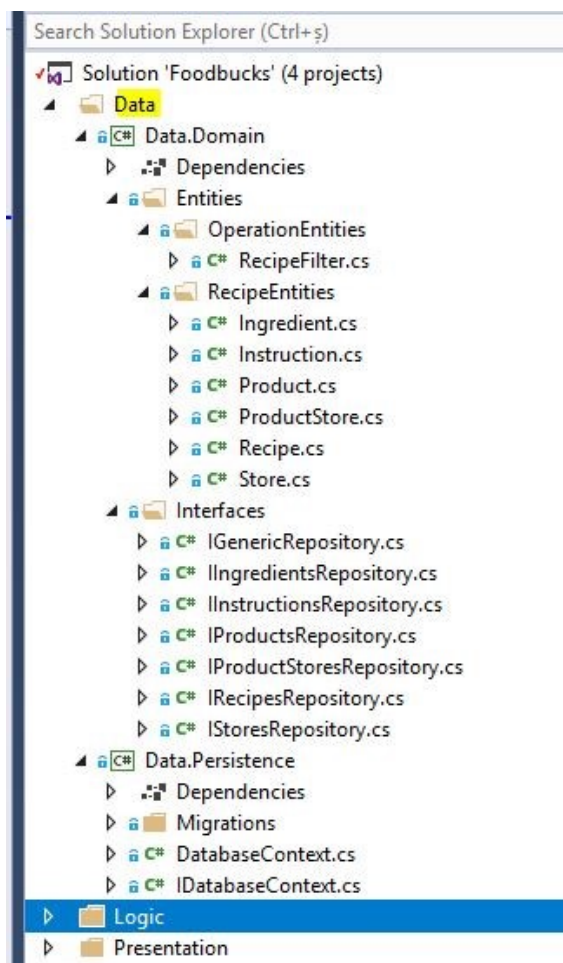


Figura 2.1: Structura nivelului date

În implementarea propusă, administrarea datelor se face în nivelul de date (Figura 2.1). Aceasta este de tipul „Code-First Database Design”, care presupune crearea automată a bazei de date, folosind modelul entităților și al legăturilor dintre acestea descrise în cod și al unui șir de conexiune, care se află în fișierul „appsettings.json” și conține numele server-ului de baze de date, pe lângă alte opțiuni referitoare la acesta. În cadrul nivelului de date se află două proiecte: unul pentru descrierea domeniului datelor și unul care se ocupă de legătura cu baza de date. Domeniul are în componența sa clasele ce descriu modelele entităților ce vor fi folosite de partea de logică și de baza de date. Pe lângă acestea, conține interfețele ce descriu repertoriile de funcții ce vor acționa asupra entităților. Partea de persistență are în componență o interfață a contextului bazei de date și una care implementează această interfață. Legătura dintre baza de date și entități este asigurată de funcțiile oferite de interfața generică „DbContext”. Aceasta pune la dispoziție modalitățile prin care se creează baza de date, se asigură de existența și ajută la popularea acesteia.

Entitățile ilustrate în Figura 2.2, cu care lucrează aplicația propusă, împreună cu câmpurile acestora sunt:

- Magazin, cu atributele nume, descriere și website. Acesta are rolul de a păstra informațiile unui magazin și va fi legat de o mulțime de produse, semnificând apartenența produselor la un anumit magazin.
- Produs, cu atributele nume, brand, cantitate, unitate de măsură și sistem de măsură. Din acesta se vor extrage informațiile necesare creării ingredientelor fiecărei rețete.
- O entitate care are rolul să rezolve relația „Many-to-many” dintre produse și magazine, denumită produs-magazin. Aceasta are ca atribute id-ul unui magazin, id-ul unui produs și prețul produsului în magazinul respectiv.
- Ingredient, care are ca atribute id-ul unei rețete, id-ul unui produs-magazin, numărul de produse necesare, cantitatea, unitatea de măsură și costul este entitatea ce descrie un ingredient din rețetă. Costul este calculat în funcție de cantitatea și unitatea de măsură folosite, cu ajutorul informațiilor oferite de entitatea produs-magazin legată.
- Instrucțiune, cu atributele id-ul unei rețete, descrierea și numărul instrucțiunii. Aceasta este folosită de către rețetă.

- Rețetă, entitatea principală a aplicației, are atributele nume, descriere, porții, calorii, aprecieri, voturi, notă și cost.
- Filtru, este o entitate ce nu va fi folosită în cadrul bazei de date, ci doar pentru a menține informațiile referitoare la filtrele aplicate pe lista de rețete la un moment dat.

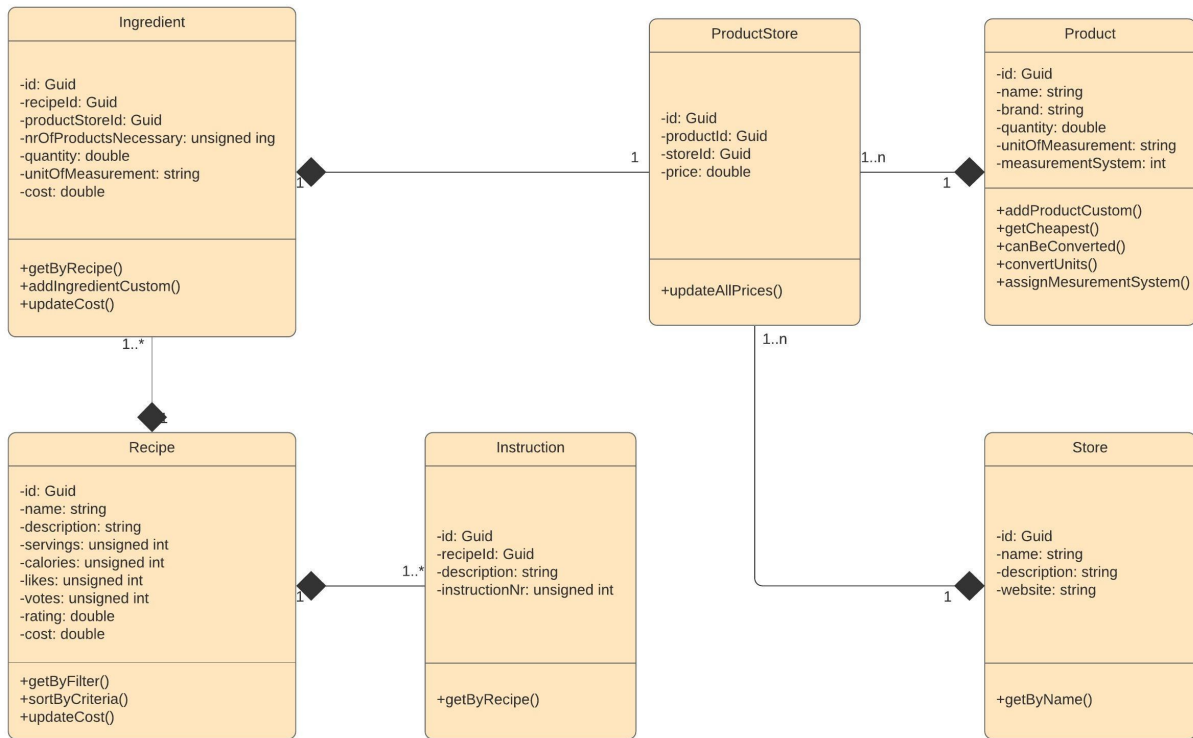


Figura 2.2: Diagrama UML

Pentru crearea și actualizarea schemei bazei de date pe baza entităților se folosesc migrări, ce sunt executate prin metodele puse la dispoziție de „Entity Framework Core”. Se folosesc două comenzi: „add-migration”, care creează script-urile SQL necesare la crearea bazei de date, și „update-database” ce rulează efectiv script-urile menționate.

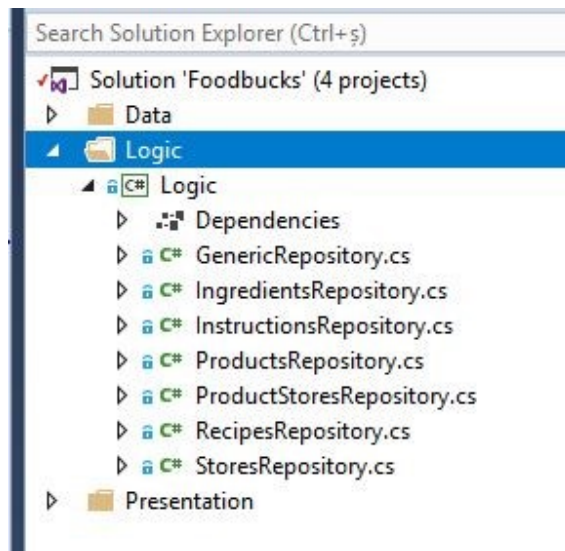


Figura 2.3: Structura nivelului logică

Mergând mai departe în structura pe trei niveluri, la al doilea nivel se află partea de logică (Figura 2.3), ce conține implementarea repertoriilor entităților. Cele șapte clase ce implementează interfețele descrise în nivelul de date sunt:

- Repertoriul generic, ce este o clasă abstractă, implementează funcțiile generice prin care se poate adăuga, edita, șterge sau returna orice entitate după id. Conține și funcția care returnează întregul repertoriu de entități în orice moment. Acest repertoriu acționează ca legătura de bază dintre contextul bazei de date și mulțimea de entități care este folosită la un moment dat.
- Repertoriul ingredientelor implementează funcțiile prin care se obțin ingredientele după nume sau id-ul rețetei, adăugarea unui ingredient și actualizarea costului.
- Repertoriul instrucțiunilor, ce implementează funcțiile de adăugare a noilor instrucțiuni și de obținere a acestora pentru fiecare rețetă.
- Repertoriul produselor conține definițiile unităților de măsură, împreună cu factorii de conversie și sistemul măsurii (masă, volum sau bucată). Pe lângă acestea, sunt implementate funcțiile de adăugare a unui produs nou, în care se face și legătura cu magazinul și, implicit, se setează prețul produsului din magazinul ales. În acest repertoriu sunt implementate și funcțiile care ajută la returnarea produsului optim, în funcție de nevoile unui ingredient dintr-o anumită rețetă (cantitate și unitate de măsură).

- Repertoriul produselor-magazinelor este implementat doar cu scopul de a oferi acces la datele entităților pe care îl reprezintă și implementează funcțiile de obținere a entităților după magazin, produs și a id-ului magazinului care vinde un anumit produs cu prețul cel mai mic.
- Repertoriul rețetelor, conține funcțiile de adăugare a noilor rețete, de obținere a rețetelor, de a aplica funcțiile de sortare și de filtrare asupra listelor de rețete și de a actualiza prețul unei rețete, recalculându-l după noile prețuri ale produselor.
- Repertoriul magazinelor, ce oferă funcții de extragere a magazinelor după nume și de verificare a existenței acestora după nume.

Ideal, aplicația propusă ar trebui să primească informațiile cu privire la detaliile produselor și al magazinelor de la un API terț, care ar returna, la cererile server-ului aplicației propuse, prețurile actualizate ale produselor, pe lângă celelalte detalii ale acestora. Detaliile ar trebui să fie preluate din mai multe surse, reprezentate de magazine diferite, pentru a asigura o arie cât mai mare de prețuri posibile a fiecărui produs. În implementarea prezentată, însă, toate informațiile sunt adăugate manual de către administratorul aplicației.

Mecanismul de adăugare a unui nou ingredient, ce este implementat în cadrul repertoriului ingredientelor, se execută astfel:

- Se verifică dacă există un produs cu numele ingredientului adăugat și în caz că nu există, se adaugă un nou produs cu prețul setat pe nedefinit.
- În caz că un astfel de produs există, se trece la căutarea magazinului în care există, se caută magazinul în care acest produs are prețul cel mai mic, iar dacă nu se găsește în niciun magazin, atunci se returnează cu cost nedefinit.
- După ce se găsește magazinul căutat mai sus, se calculează numărul de produse necesare pentru a acoperi cantitatea cerută de ingredient.
- Se înmulțește numărul de produse cu prețul acestuia și se calculează costul.

Pentru calcularea și actualizarea prețului rețetelor se folosește mecanismul de mai sus pentru a calcula prețul tuturor ingredientelor, ca mai apoi să se adune prețul acestora.

Funcția de filtrare, implementată în interiorul repertoriului rețetelor, primește ca dată de intrare un obiect de tip filtru în care se află informațiile care descriu toate filtrele ce trebuie aplicate asupra rețetelor. Execuția acestei funcții se desfășoară astfel: în primul rând se obține lista cu toate rețetele, apoi se filtrează acestea în funcție de cost, nume, timp de gătire, brand-urile pe care le conține, ingredientele pe care le conține și magazinele de unde sunt alese produsele. Se returnează, apoi, lista de rețete care are toate criteriile îndeplinite.

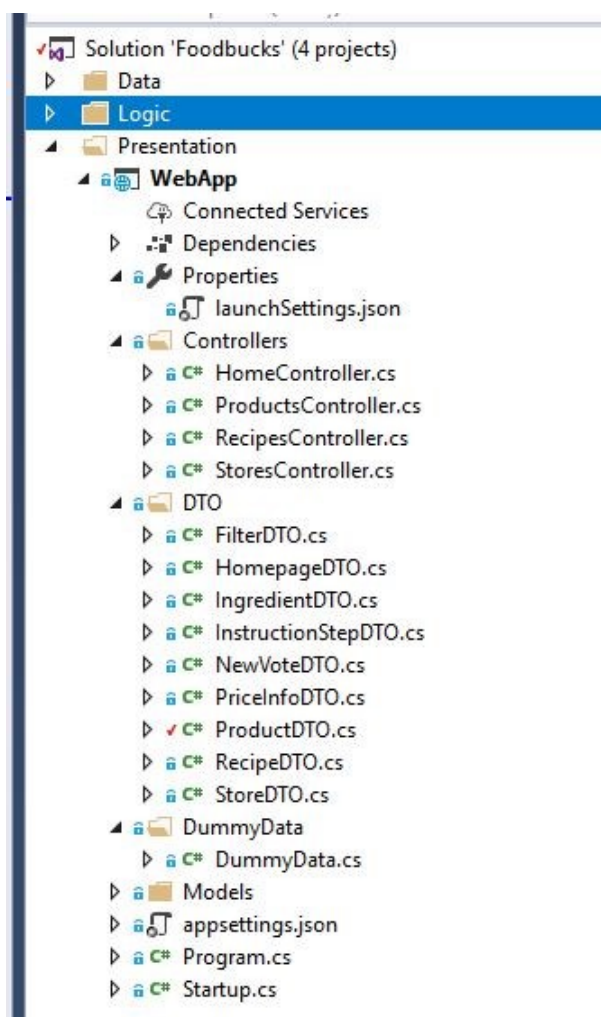


Figura 2.4: Structura nivelului prezentare

Următorul nivel este prezentarea (Figura 2.4). Acesta constă într-un proiect de tipul „ASP.NET Core Web Application”, ce funcționează ca API ce expune (prezintă) logica soluției. În componența sa găsim controlerele, DTO-urile și codul care generează datele machetă, dacă este detectat faptul că bazele de date sunt goale. Controlerele returnează și așteaptă informațiile, asincron, sub formă de JSON-uri.

Controlerele sunt definite pentru produse, magazine și rețete. Cel pentru produse oferă funcțiile prin care noi produse pot fi adăugate prin verbul POST cu un JSON de tipul DTO-ului produsului în corpul cererii. Pe lângă aceasta, există definite și implementări ale funcțiilor ce îndeplinesc funcționalitatea verbului GET, pentru detaliile unui produs, o listă cu toate produsele și o listă cu prețurile unui produs. Controlerul magazinelor are, în esență, implementate același tip de funcții, doar că raportate la entitatea pe care îl reprezintă. Controlerul rețetelor oferă, pe lângă funcțiile standard, funcțiile ce returnează lista de rețete pe baza unui anumit filtru sau al unui anumit criteriu de sortare.

S-a ales folosirea DTO-urilor deoarece s-a dorit ca fiecare cerere la server să returneze cât mai multe date posibile, pentru a scădea numărul acestor cereri. Cu cât numărul este mai mare, cu atât se încarcă nivelul de procesare al server-ului, iar eficientizarea procesului a fost unul dintre obiectivele implementării.

Clasa datelor machetă se ocupă de verificarea dacă baza de date nu are deloc date și, dacă este cazul, creează obiecte exemplu și le adaugă în aceasta pentru cazurile când se testează funcționarea aplicației.

Pentru rularea serverul-ui, sunt folosite serviciile de informare internet (IIS 10.0) integrate în Visual Studio 2019. Pentru găzduirea server-ului la o adresă web, este folosită unealta ngrok 2.3.29, versiunea gratuită. Limitarea acestei versiuni este timpul maxim de 8 ore de găzduire, înaintea închiderii server-ului.

2.2.2 Frontend

Partea de interfață, implementată în platforma android, se folosește de librăria HTTP Volley, care facilitează conectarea la serviciile oferite de API-ul server-ului implementat în partea de backend. Pentru a folosi aceste servicii, au fost create clasele java „AppConfig” și „AppController”. În clasa pentru configurații sunt definite macro-urile care conțin adresa HTTPS expusă de ngrok și șirurile de caractere necesare la executarea metodelor POST și GET: obținerea tuturor rețetelor, adăugarea unei noi rețete, obținerea detaliilor unei rețete, obținerea rețetelor după nume (pentru funcția de căutare), ștergerea unei rețete, adăugarea unui vot, obținerea rețetelor după un anumit criteriu de sortare, adăugarea unui filtru peste rețete, obținerea tuturor produselor, adăugarea unui produs nou, obținerea detaliilor unui produs, obținerea tuturor

magazinelor, adăugarea unui magazin și obținerea detaliilor unui magazin. În clasa controlerului aplicației se află funcțiile ce construiesc coada de cereri către server.

Activitățile aplicației sunt: activitatea principală (de navigare a rețetelor), activitatea detaliilor unei rețete, activitatea adăugării unei rețete, activitatea magazinelor, activitatea detaliilor unui magazin, activitatea produselor și activitatea prețurilor unui produs.

Activitatea principală, care poate fi considerată și activitatea rețetelor, are, din punctul de vedere al interfeței grafice, o schemă formată dintr-o casetă text, unde se poate insera un șir de caractere, un buton pentru căutare, un buton pentru filtrare și un ultim buton pentru sortarea rețetelor. În momentul în care aplicația este deschisă și ori de câte ori este acționat butonul de navigare din meniul din stânga, se execută funcția „onCreate” a acestei activități, care trimite o cerere HTTP de tipul GET pentru o listă de obiecte de tipul JSON, care descriu în mod sumar toate rețetele disponibile. După primirea răspunsului dorit, sunt create un număr de butoane egal cu numărul de rețete primite, apoi acestea sunt afișate în activitatea principală. Apăsarea oricărui dintre aceste butoane pornește activitatea detaliilor rețetei. Funcția de căutare este asigurată de caseta de text și butonul de căutare. În momentul în care se apasă pe butonul de căutare, se evaluează textul din interiorul casetei text și se trimite o cerere de tipul GET la adresa pentru căutarea după nume a server-ului, cu textul din casetă concatenat. Această cerere returnează o listă de obiecte JSON la fel ca mai sus. Pe apăsarea butonului de filtrare se deschide o fereastră de tip pop-up care conține un formular în care se pot adăuga filtre după nume, costul maxim, nota minimă, numărul minim de voturi, timpul maxim de gătire în minute, numele produselor incluse, numele brandurilor incluse, și magazinele preferate, separate prin virgulă. La acționarea butonului de filtrare, cu datele introduse se construiește obiectul JSON ce va fi trimis în corpul metodei POST către server, ca să se primească lista de obiecte JSON ce descriu rețetele filtrate. Implementarea sortării este asemănătoare cu cea a filtrării, diferența fiind în modelul formularului, ce conține în acest caz două filatoare, unul pentru criteriul de sortare, care poate fi după nume, cost, notă, numărul de aprecieri, numărul de porții sau timpul de gătire și unul pentru tipul sortării, care poate fi ascendent sau descendent.

Activitatea detaliilor unei rețete, are în componența interfeței grafice mai multe obiecte de tipul TextView, așezate vertical, în care se găsesc detaliile rețetei. Pe lângă acestea există două zone goale, una pentru ingrediente și una pentru instrucțiuni. În

momentul în care această activitate este creată, se evaluează id-ul primit prin intenția din activitatea principală. Acest id este concatenat la adresa ngrok care trimite ca răspuns JSON-ul ce conține detaliile unei rețete, iar apoi se execută metoda POST asupra acestei adrese. Răspunsul primit este folosit pentru a umple obiectele de tip `TextView` și informațiile referitoare la ingredientele și instrucțiunile rețetei alese. În partea de jos a schiței se află butonul de apreciere și cel de ne-apreciere, iar atunci când unul dintre acestea este acționat, se trimite o cerere de tipul PUT către server cu informația noului vot, iar răspunsul la cerere produce modificarea în direct a notei rețetei.

Activitatea de adăugare a unei noi rețete este reprezentată vizual de un formular care are câmpurile de tip text pentru nume, descriere, numărul de porții, calorii, timpul de gătire, butoane pentru adăugarea de noi rețete și instrucțiuni și butoane pentru înlăturarea acestora. Când se acționează butonul de adăugare a unui ingredient se creează o nouă linie în formular, care are o casetă text pentru numele ingredientului, una pentru cantitatea acestuia și una pentru unitatea de măsură. Când se acționează butonul de adăugare a unei instrucțiuni se creează o nouă linie care conține un text simplu ce descrie numărul instrucțiunii și o casetă text în care trebuie scris textul instrucțiunii. La acționarea asupra butonului de înlăturarea a unui ingredient sau a unei instrucțiuni se șterge ultima linie a instrucțiunilor sau a ingredientelor adăugată. În partea de jos a paginii se află butonul de adăugare a rețetei, care, dacă este acționat în timp ce oricare dintre câmpuri este gol, nu există ingrediente sau nu există instrucțiuni, va produce afișarea unui mesaj de avertizare și nu va trimite nicio cerere către server. Dacă formularul este completat corect în momentul acționării butonului, se va construi obiectul JSON pe baza datelor introduse și se va trimite în corpul cererii de tip POST către adresa ngrok pentru adăugarea de noi rețete. Dacă răspunsul la cerere este pozitiv, atunci se pornește activitatea detaliilor rețetei nou create.

Activitatea magazinelor este o listă de obiecte de tip text, peste care se poate acționa cu click. În momentul creării, se execută o cerere de tip GET către adresa ngrok pentru primirea listei de magazine. Această listă este interpretată și se creează dinamic obiectele de tip text pentru fiecare dintre magazine, fiecare dintre acestea conținând numele, descrierea și site-ul web al magazinului. La acționarea asupra oricărui dintre aceste obiecte de tip text se deschide activitatea detaliilor magazinului. La crearea activității detaliilor unui magazin se trimite o cerere de tip GET către server, care returnează lista produselor care se află în acel magazin. Cu ajutorul datelor din lista primită se creează dinamic mai multe obiecte de tip text care vor produce vizual catalogul de

produse al magazinului.

Activitatea produselor este foarte asemănătoare ca implementare cu activitatea magazinelor, iar atunci când se acționează asupra unui produs, se deschide activitatea detaliilor produsului, ce reprezintă o listă cu magazinele în care acesta se găsește, împreună cu prețurile respective.

2.3 Probleme întâmpinate și soluțiile acestora

1. La adăugarea unui nou ingredient se poate seta ca unitate de măsură o unitate ce nu poate fi convertită la unitatea niciunui produs corespunzător, de exemplu când ingredientul este 200 de grame dintr-un anumit produs, însă toate obiectele din baza de date ce descriu acel produs au ca unitate de măsură mililitrul sau litrul, iar gramele (masă) nu pot fi convertite la mililitri (volum). Pentru a evita erorile generate de această problemă, a fost adăugat atributul „sistem de măsură”, care poate avea unul din trei valori: masă, volum sau la bucată. Astfel, atunci când un ingredient nou este adăugat, se obține sistemul său de măsură pe baza unității de măsură. Se caută apoi în baza de date produsele care se potrivesc în primul rând cu numele și sistemul de măsură, iar dacă nu se găsesc astfel de potriviri, costul ingredientului este ales ca fiind egal cu prețul celui mai ieftin produs care se potrivește ca nume.

2. Librăria volley funcționează implicit cu obiecte de tip JSON și cu liste de obiecte de acest tip. La fel, cererile HTTP se pot face doar dacă obiectul returnat (răspunsul HTTP) și corpul cererii (dacă acesta există) sunt de același tip. În cazul folosirii filtrelor, însă, aplicația android are nevoie să trimită un obiect de tip JSON, dar așteaptă să primească o listă de obiecte JSON. Oricât s-a încercat, nu s-a putut reuși comportamentul corect folosind tipurile de obiecte standard. O opțiune puțin cam complicată ar fi fost modificarea la nivelul serverului al tipurilor de obiecte JSON, ca acestea să potrivească cererile și răspunsurile. Soluția, însă, a venit prin implementarea unei clase personalizate de cereri JSON, care se așteaptă să trimită un obiect și să primească o listă. Astfel, funcționalitatea a fost implementată cu succes.

Capitolul 3

Ghidul utilizatorului

Acest capitol conține ghidul de utilizare al aplicației android. Aplicația este în limba engleză și are nevoie de conexiune permanentă la internet pentru a funcționa corect.

3.1 Navigarea rețetelor

La deschiderea aplicației se vor încărca toate rețetele, cum arată în Figura 3.1, începând cu cele mai noi adăugate.

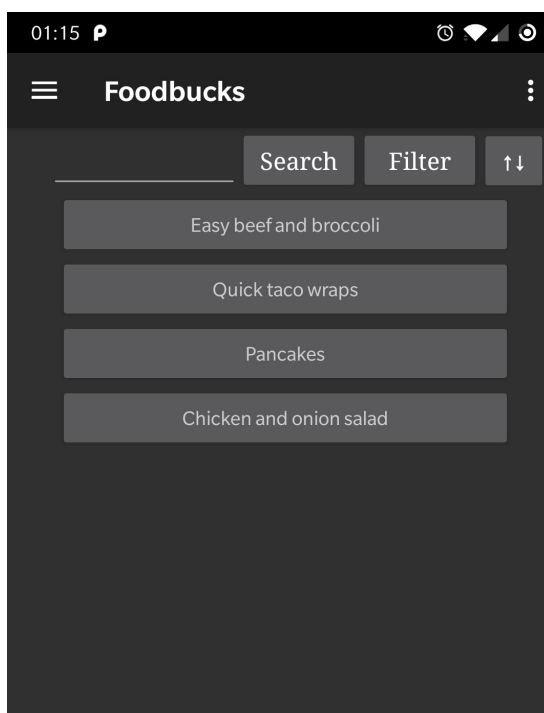


Figura 3.1: Pagina de pornire

Se poate folosi funcția de căutare după nume, scriind text în caseta din stânga butonului „Search” și acționând apoi butonul (Figura 3.2).

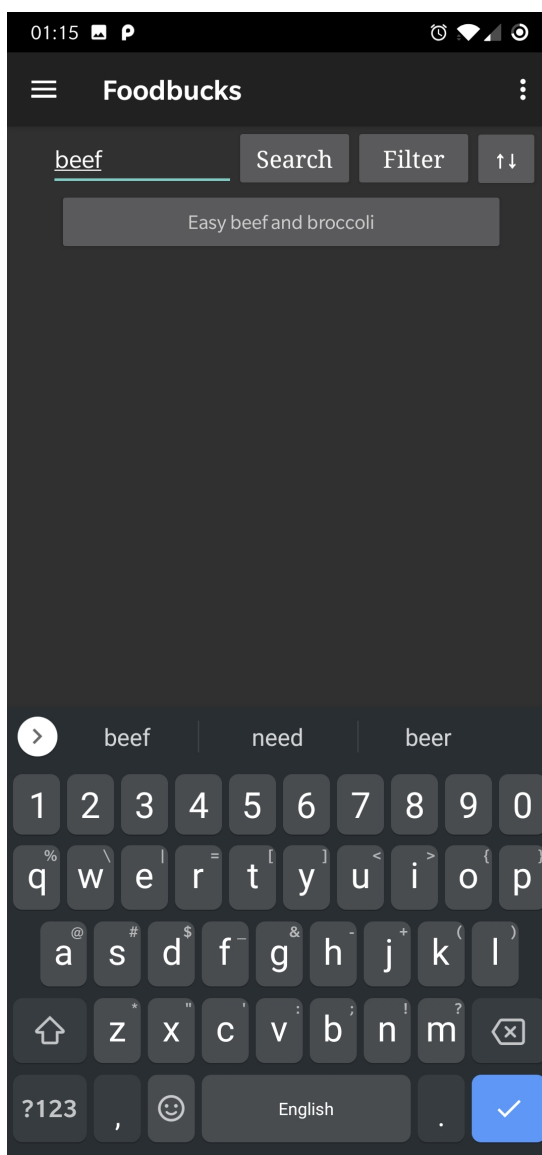


Figura 3.2: Funcția de căutare

Este de notat faptul că atunci când se folosește funcția de căutare, se caută doar după numele rețetelor, nu și după de scriere sau oricare din celelalte attribute.

Acționând butonul „Filter” din pagina principală se deschide fereastra de tip pop-up ce conține formularul pentru filtru (Figura 3.3). Prin acționarea butonului „Cancel” se revine la fereastra principală, fără a fi aplicat niciun filtru. Dacă se acționează asupra butonului „Filter” cu un anumit câmp gol, atunci nu se va filtra după acel câmp, asta însemnând că filtrarea cu formularul nemodificat returnează lista cu toate rețetele.

Câmpurile trebuie completate astfel: costul este un număr cu zecimale, rating-ul este ales din filatorul ce conține numerele de la 0 la 10, numărul de voturi este un

întreg, timpul maxim de gătire este un întreg și se măsoară în minute, iar produsele incluse, brand-urile și magazinele, dacă sunt mai multe, se separă prin virgule.

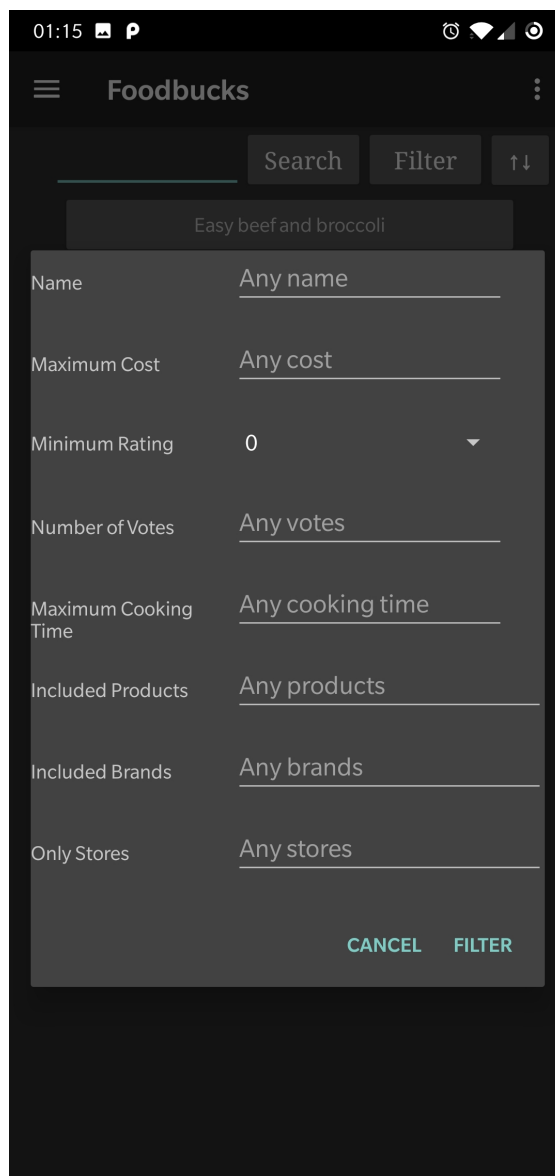


Figura 3.3: Funcția de filtrare

Filtrarea returnează apoi o nouă listă de rețete, ce îndeplinesc condițiile impuse. Această listă de rețete va fi afișată în modelul activității principale.

Acționând pe butonul de sortare, se deschide fereastra de tip pop-ul pentru alegerea tipului sortării (Figura 3.4). Criteriile de sortare din care se poate alege sunt numele, costul, nota, numărul de aprecieri, numărul de porții și timpul de preparare. Tipul de sortare poate fi crescător sau descrescător. După ce se aplică un anumit tip de sortare, fereastra de sortare se închide și cadrul ce conține lista de rețete se reîncarcă, respectând tipul de sortare ales.

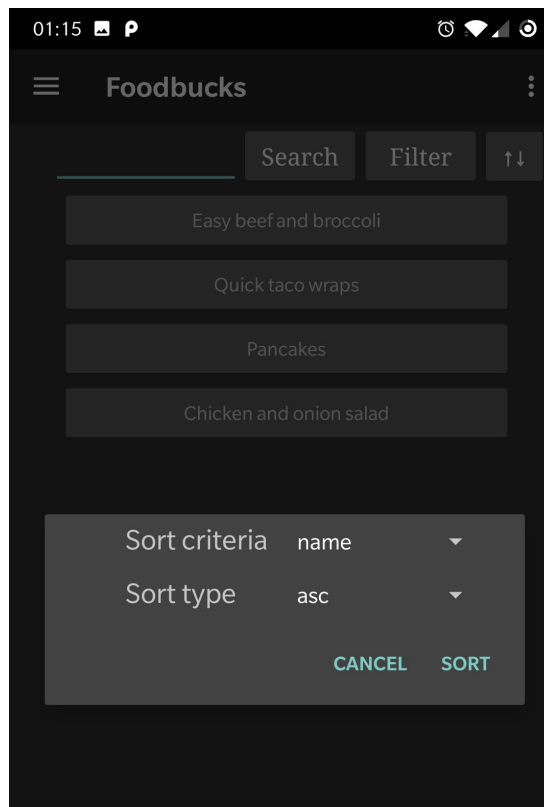


Figura 3.4: Funcția de sortare

După alegerea filtrării și sortării dorite, se poate găsi cu ușurință rețeta căutată. Prin acționarea butonului rețetei se deschide activitatea ce conține detaliile rețetei (Figura 3.5). În cazul luat ca exemplu avem o rețetă de clătite, în limba engleză. De sus în jos, avem informațiile referitoare la numele rețetei, descrierea, numărul de calorii, timpul de gătire, numărul de porții, costul, numărul de stele (nota), ingredientele și instrucțiunile. Costul este calculat ca suma costurilor fiecărui ingredient, care sunt afișate în secțiunea ingredientelor.

Pentru a evalua rețeta se pot folosi butoanele „Like” sau „Dislike”. Odată ce unul dintre voturi a fost exprimat, nu se poate vota din nou la fel, butonul blocându-se. Se poate face click pe instrucțiuni, această acțiune schimbând culoarea în verde a fundalului, pentru a le marca în vederea urmăririi progresului (Figura 3.6). Cu ajutorul butonului „Înapoi” al telefonului, se revine la pagina principală. De acolo se poate deschide meniul lateral.

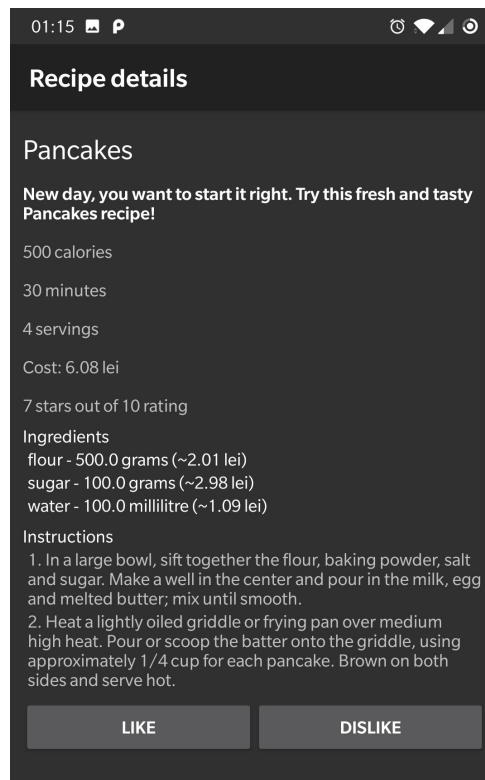


Figura 3.5: Detaliile unei rețete

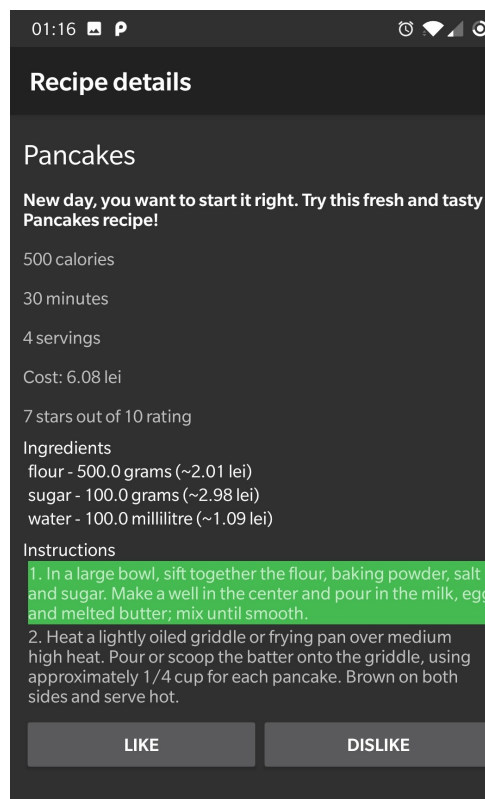


Figura 3.6: Marcarea instrucțiunilor unei rețete

Meniul lateral conține butoanele prin care se pot porni activitățile de navigare a rețetelor, adăugare a unei noi rețete, navigare a produselor și navigare a magazinelor (Figura 3.7).

Dacă este deja deschisă aplicația în cadrul activității principale și se acționează butonul de navigare a rețetelor „Browse Recipes”, atunci meniul este restrâns și se restartează activitatea principală, readucându-se lista de rețete de pe server.

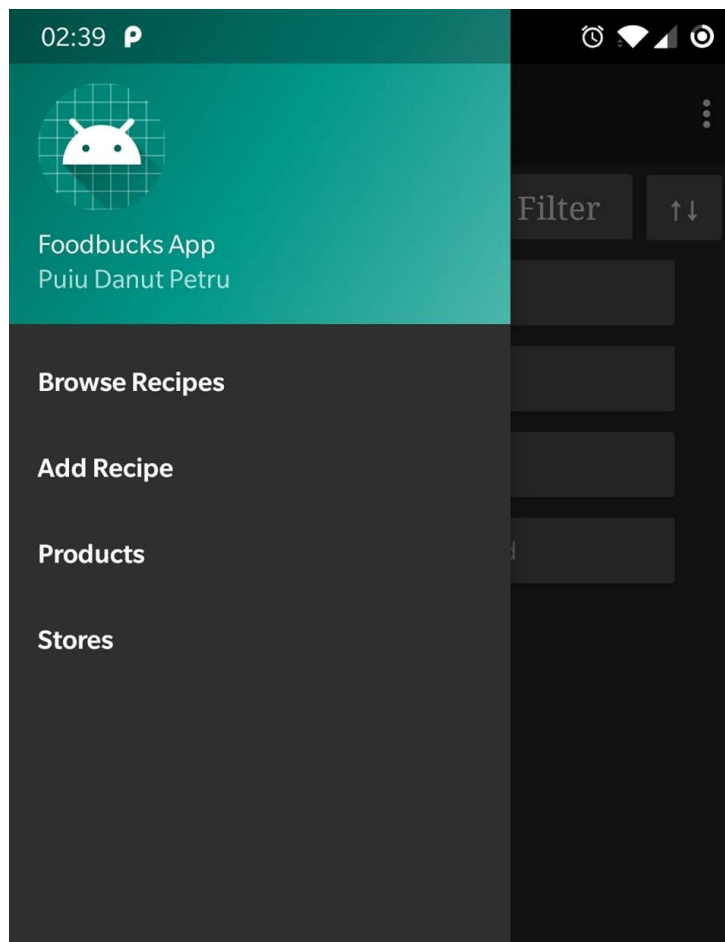


Figura 3.7: Meniul lateral

Dând click pe butonul „Add Recipe” pornește activitatea de adăugare a unei noi rețete.

3.2 Adăugarea de rețete noi

Adăugarea unei rețete noi presupune completarea integrală a formularului ilustrat în Figura 3.8.

01:16 [notification icon] [profile icon]

Add a new recipe

Name

Description

Servings

Calories

Cooking Time

Ingredients

▼

Instructions

1.

2.

Figura 3.8: Adăugare rețetă

Datele de intrare trebuie să fie de următoarele tipuri: numele și descrierea sunt șiruri de caractere, porțiile, numărul de calorii și timpul de gătire în minute sunt numere întregi. Pentru adăugarea de ingrediente noi se acționează butonul „Add ingredient”, iar pentru ștergerea ultimului ingredient adăugat se apasă butonul „Remove last ingredient”. Fiecare dintre ingrediente trebuie să aibă casetele completate, cea din stânga conține numele ingredientului, iar cea din dreapta este un număr cu zecimale ce reprezintă cantitatea în unitatea de măsură aleasă din filatorul de unități de măsură. Pentru adăugarea de instrucțiuni se apasă pe butonul „Add instruction”, iar pentru ștergerea ultimei instrucțiuni adăugate se apasă butonul „Remove last instruction”. Toate casetele text ale ingredientelor trebuie completate pentru a se adăuga rețeta. După ce toate datele au fost adăugate corespunzător, se apasă pe butonul din partea de jos „Add recipe”. În cazul în care nu există erori, va apărea o notificare de tip toast care va anunța faptul că rețeta a fost creată cu succes și se va deschide activitatea cu detaliile rețetei nou adăugate, ca în Figura 3.5.

3.3 Navigarea produselor

Dacă din meniul lateral (Figura 3.7) se acționează butonul „Products”, se deschide activitatea care conține lista de produse, ilustrată în Figura 3.9.

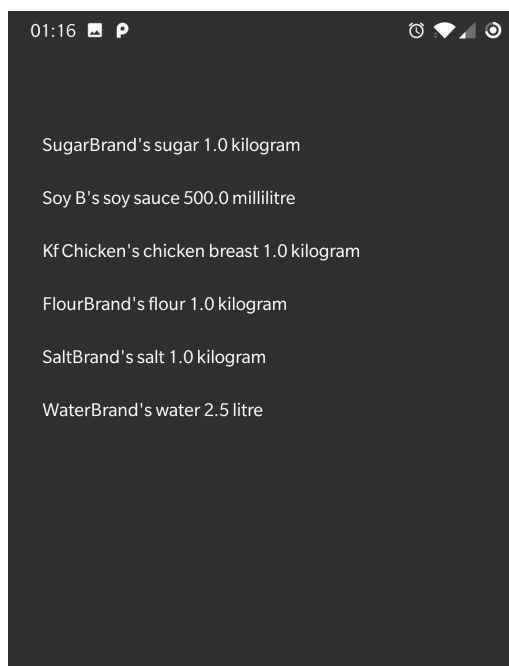


Figura 3.9: Produse

În cadrul acestei scheme sunt listate toate produsele disponibile, împreună cu numele brand-ului și cantitatea lor. Acționând pe oricare dintre ele se deschide activitatea ce descrie magazinele unde se găsește produsul, împreună cu prețurile lui. În exemplul arătat, acționând asupra produsului „sugar” al brand-ului fictiv „SugarBrand”, se deschide activitatea ilustrată în Figura 3.10.

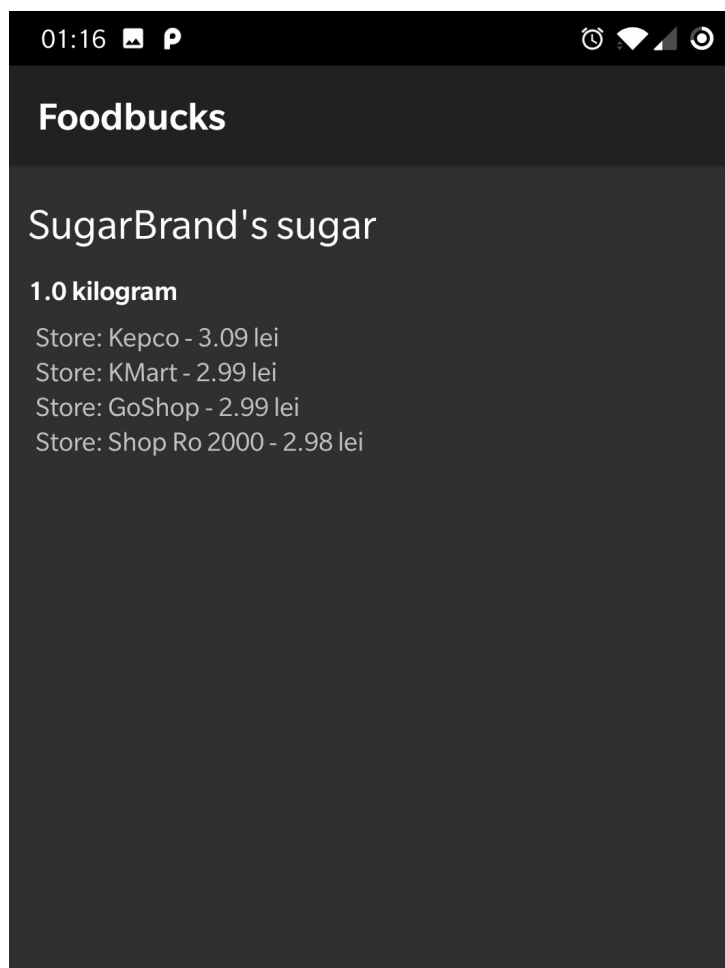


Figura 3.10: Detalii produs

3.4 Navigarea magazinelor

Pentru navigarea magazinelor, se poate acționa butonul „Stores” din panoul lateral (Figura 3.7). Acesta deschide activitatea ce conține lista magazinelor, care reprezintă o schemă de obiecte de tip text, fiecare având în componența sa numele, descrierea și adresa web a magazinului (Figura 3.11).



Figura 3.11: Magazine

Concluzii

Cu trecerea timpului, am început să apreciez din ce în ce mai mult arta gătitului, aceasta fiind unul dintre factorii definitorii ai unor culturi. Am încerca să folosesc cât mai multe aplicații de rețete pentru a le studia funcțiile și a le identifica pe cele ce le lipsesc, dar și pe cele cu care ies în evidență. Fiecare dintre acestea are puncte forte și puncte slabe, însă niciuna nu s-a arătat a fi aplicația completă la care m-am așteptat.

Aplicația Foodbucks reprezintă un prototip al încercării mele de a crea aplicația perfectă de rețete, care ajută utilizatorul să gătească eficient și ușor din prima secundă când a deschis-o, până în momentul când a terminat de servit bucatele.

Consider că aplicația poate fi îmbunătățită din punctul de vedere al proiectării vizuale, cât și din punctul de vedere al funcționalităților, prin mai multe feluri, din care vreau să menționez:

- Adăugarea autentificării utilizatorilor, prin metode clasice și securizate. În faza inițială a proiectului, această parte a fost luată în vedere, însă s-a încercat în principal implementarea funcțiilor de bază ale aplicației: posibilitatea de a filtra rețete.
- Posibilitatea adăugării și vizualizării de imagini cu produsele gătite și a clipurilor de demonstrație pentru pașii rețetelor.
- Adăugarea atributului care conține locația magazinului fizic, astfel încât atunci când se face căutarea produselor în magazine, să se prefere magazinele care sunt fizic mai apropiate de utilizatori.
- Posibilitatea utilizatorilor de a trece pe o „listă neagră” anumite magazine sau brand-uri, astfel încât, atunci când aceștia accesează o anumită rețetă, entitățile ignorate să nu fie luate în considerare la calculul costului și să nu apară în niciun fel în cadrul interfeței grafice.

- Adăugarea unei funcții de găsim a celei mai eficiente combinații de produse, având un anumit tip de produs și o anumită cantitate, folosind informațiile deja disponibile.
- Crearea unei interfețe grafice mai fluide și proiectate mai aproape de design-ul avut în stadiul de înaintea începerii implementării aplicației, care nu a mai fost posibil deoarece s-a încercat o concentrare mai mare asupra asigurării funcționării elementelor de bază, mai mult decât asupra asigurării unui model frumos din punct de vedere estetic.

Aplicația propusă nu este o implementare ideală a rezolvării problemei introduse la începutul lucrării. Unul dintre defectele implementării propuse este dat de dependența de persistența datelor oferite de magazine și eventualele mici erori apărute din cauza structurii datelor păstrate în interiorul aplicației propuse, în comparație cu structura datelor ce ar putea fi primite de la serviciile terțe. Un alt defect este reprezentat de riscurile legale pe care această abordare le presupune, prin faptul că o astfel de implementare poate influența foarte mult consumatorii și, prin extensie, piața. O astfel de aplicație poate fi acuzată că favorizează într-un fel sau altul anumite afaceri, în detrimentul altora, iar acest fapt poate fi sancționat legal. Trecând peste inconvenientele descrise mai sus, aplicația poate fi viabilă prin faptul că oferă o cale simplă și inedită prin care utilizatorii pot găsi rețete în funcție de bugetul de care dispun, printre alte preferințe.

În concluzie, aplicația propusă ar putea, cu mici ajustări, să reprezinte o soluție adecvată problemei eficientizării procesului gătitului, prin oferirea uneltelor necesare căutării rețetelor culinare, informării asupra costului acestora, al prețurilor ingredientelor, împreună cu locurile de unde pot fi procurate și posibilitatea de adăugare a rețetelor noi și originale.

Bibliografie

1. Dragoş Gavriluţ, *Cursuri Android Programming*, 2019
2. Alois Kraus, <https://aloiskraus.wordpress.com/2018/11/25/how-fast-can-you-get-with-net-core/>, 2018
3. Saineshwar Bageri, <https://www.c-sharpcorner.com/UploadFile/4d9083/create-and-implement-3-tier-architecture-in-Asp-Net/>, 2018
4. Valdio Veliu, <https://www.sitepoint.com/volley-a-networking-library-for-android/>, 2016
5. Abdullah Kajamohideen, <https://vmokshagroup.com/blog/expose-your-localhost-to-web-in-50-seconds-using-ngrok/>, 2015
6. Diffen LLC, n.d. Web., <https://www.diffen.com/difference/GET-vs-POST-HTTP-Requests/>, 2019
7. Brice Lambson et al, <https://docs.microsoft.com/en-us/ef/core/managing-schemas/migrations/>, 2018