# Cuprins

[Cuprins 1](#_Toc43313245)

[Introducere 2](#_Toc43313246)

[Motivație 2](#_Toc43313247)

[Obiectiv 2](#_Toc43313248)

[Contribuții 3](#_Toc43313249)

[Cerințe funcționale 4](#_Toc43313250)

[1. CAPITOLUL I. Tehnologii folosite: 6](#_Toc43313251)

[1.1. Flutter framework 6](#_Toc43313252)

[1.1.1. Sistemul de operare Android 6](#_Toc43313253)

[1.1.2. Sistemul de operare iOS 8](#_Toc43313254)

[1.2. Web API ASP.NET CORE 10](#_Toc43313255)

[1.3. Angular 11](#_Toc43313256)

[1.4. MongoDB 11](#_Toc43313257)

[2. CAPITOLUL II. Arhitectura aplicației 13](#_Toc43313258)

[2.1. Structura 13](#_Toc43313259)

[2.2. Modulele aplicației 14](#_Toc43313260)

[2.2.1. Modulul Interfață Web 14](#_Toc43313261)

[2.2.2. Modulul interfața Android și iOS 23](#_Toc43313262)

[2.3. Modulul bază de date 24](#_Toc43313263)

[*2.4.* Modulul API *Gateway* 26](#_Toc43313264)

[3. CAPITOLUL III. Detalii de implementare 27](#_Toc43313265)

[Concluzii 28](#_Toc43313266)

[Bibliografie 29](#_Toc43313267)

# Introducere

## Motivație

În această lucrare de licență principala problemă abordată este cea a timpului, a așteptării continue și a statului la coadă, precum și cea a incertitudinii in legătură cu ceea ce am prefera în momentul în care apare pofta de mâncare și am decis să luăm masa la cantină sau restaurant.

Principalul scop al aplicației pentru utilizatori precum studenți și profesori este cel de a reduce timpul de așteptare și de a realiza comenzile de produse într-un ritm mai alert pentru cei înfometați prin generarea unui cod de 5 cifre după ce utilizatorul a ales ce produse dorește să cumpere, și de a pune la dispoziție utilizatorilor din afara personalului, cei ce frecventează această cantină, un sistem de recomandare produse pe baza celor cumpărate anterior.

Scopul aplicației pentru administratorii în cadrul componentei interfeței, adică cei ce fac parte din cadrul personalului este cel de a le ușura munca si de a le oferi un sistem de gestionare a produselor cantinei. Principalele funcționalități pentru acest tip de utilizator sunt următoarele:

* magazinul virtual propriu zis în care se selectează produsele ce doresc a fi cumpărate si generarea unei chitanțe
* generarea unei chitanțe pe baza unui cod de 5 cifre generat automat in momentul in care utilizatorii au ales anumite produse ce doresc a fi cumpărate
* adăugarea de produse noi
* adăugarea de utilizatori noi
* crearea unui meniu pentru ziua curentă

## Obiectiv

Soluția pentru această problemă, descrisă in această lucrare a fost dezvoltarea unei aplicații Web și mobile pentru sistemele de operare Android, respectiv iOS care este ușor de utilizat si oferă funcționalități similare unui magazin virtual, aplicația fiind dezvoltată pentru doua categorii de utilizatori. Prima categorie este cea a administratorilor în cadrul componentei interfețe, persoane ce fac parte din personalul cantinei iar cea de-a doua categorie este cea a utilizatorilor, adică studenților si a profesorilor care frecventează această cantină.

Un magazin virtual/magazin online este un website destinat vânzării de produse si de servicii. În cele mai multe dintre cazuri, magazinul online reprezintă o platformă pe care sunt adăugate produse. În general, magazinele online permit cumpărătorilor să utilizeze funcții de căutare pentru a găsi modele, mărci sau elemente specifice. Clienții online trebuie sa aibă acces la Internet.

Una dintre cele mai vechi forme de tranzacționare desfășurate online a fost procesarea tranzacțiilor online a IBM dezvoltată în anii 1960 și a permis prelucrarea tranzacțiilor financiare in timp real. Apariția de cumpărături on-line, după cum știm astăzi s-a dezvoltat odată cu apariția Internetului. Inițial, această platforma funcționa doar ca un instrument publicitar pentru companii, oferind informații despre produsele sale. Aceasta a avansat rapid de la acest utilitar simplu la tranzacția reală de cumpărături online datorită dezvoltării de pagini Web interactive si de transmisii securizate.

Magazinul virtual are ca scop prezentarea si vinderea unor produse. Majoritatea comercianților mari, corporațiile sau brandurile, își dezvoltă propriile departamente web și își creează magazine virtuale.

Funcționalitățile dezvoltate in această aplicație sunt similare cu cele ale unor platforme precum „FoodPanda”ce conține informații sigure și a fost dezvoltată cu scopul de a informa utilizatorii despre anumite produse, pentru a satisface poftele utilizatorilor.

## Contribuții

Scopul principal al acestei aplicații este cel de a pune la dispoziție, atât utilizatorilor cât și administratorilor din cadrul interfeței un sistem și anumite funcționalități ce fac posibile reducerea timpului de așteptare și realizarea comenzilor de anumite produse într-un ritm mai alert.

Pentru utilizatori, aplicația pune la dispoziție o interfață web atractivă, dar și o interfață mobile pentru două din cele mai cunoscute sisteme de operare din lume (Android și iOS) și oferă diverse funcționalități precum: afișarea meniului din ziua curentă ce conține informații despre fiecare produs în parte ce este pus în vânzare, precum și un sistem de recomandare produse pe baza celor cumpărate anterior. De asemenea utilizatorul își poate alege produsele preferate, iar apoi, prin acționarea unui buton, acestuia îi este generat un cod de 5 cifre, care poate fi folosit ulterior atunci când acestuia îi vine rândul la casă, astfel se reduce timpul de așteptare și comenzile se realizează într-un ritm mai alert.

Pe de altă parte, pentru administratorii din cadrul interfeței, aplicația oferă un sistem de selecție a produselor și de generare a chitanței pe baza produselor selectate din cadrul meniului din ziua curentă, dar și un sistem prin care, introducându-se codul de 5 cifre specificat mai sus, se generează o chitanță automată, fără a mai selecta manual. O altă funcționalitate importantă este cea de creare a meniului pentru ziua curentă, în care, pentru fiecare categorie de produse, se alege din lista de produse disponibile și se adaugă în meniul respectiv. De asemenea, administratorii pot vizualiza care produse mai sunt disponibile, și pentru fiecare produs, cantitatea care mai este disponibilă. Tot aceștia pot adăuga produse noi, pot schimba anumite informații asupra unor anumite produse, pot vedea istoricul vânzărilor și pot căuta informații despre anumite produse.

## Cerințe funcționale

În momentul in care un utilizator va accesa site-ul Web sau își va descărca și instala aplicația, va apărea pagina de *Login.* Utilizatorul se va putea conecta folosind această pagină în cazul in care are deja făcut un cont introducând adresa de email, respectiv parola.

În cazul în care utilizatorul și-a uitat parola, acesta poate acționa atât în interfața mobile cât și în cea Web butonul *Ai uitat parola / Forgot password* care va duce către pagina de recuperare parola.

În cadrul acestei pagini utilizatorul va introduce adresa de email către care va fi trimis un link de resetare parolă. Aici utilizatorul va trebui să introducă de două ori aceeași parolă, astfel încât aceasta să poate să fie modificată. Dupa ce utilizatorul a introdus aceeași parolă de două ori, acesta va fi redirecționat către pagina de *Login* pentru a se putea autentifica.

Atât pentru interfața mobile cât și pentru cea Web, în cazul in care utilizatorul nu are un cont existent se poate naviga către pagina de *Register,* folosind butonul de acțiune *Înregistrare / Sign up* – care va duce către pagina de înregistrare, unde utilizatorul își va crea propriul cont in baza de date. Contul va fi completat în funcție de informațiile despre fiecare utilizator.

Odată conectați, utilizatorii vor avea un meniu cu produsele disponibile din ziua curentă, care conține informații precum: *nume, descriere, preț*, *gramaj*. Utilizatorul poate arunca o privire asupra meniului, urmând ca la casă să comande ceea ce dorește, sau prin intermediul unui buton ar putea să-și selecteze produsele dorite, iar la final să genereze un cod de 5 cifre, urmând apoi când ii vine rândul să comunice codul cu doamnele de la casă și va fi generată automat o chitanță. De asemenea utilizatorul dispune și de un buton de recomandare de produse, ce are in spate un algoritm, care pe baza produselor cumpărate anterior, va afișa produsele similare cu cele anterioare.

Pe de altă parte, pentru administratorii din cadrul interfeței, după ce s-au conectat, aceștia vor dispune de magazinul virtual propriu-zis in care vor putea selecta produsele ce sunt dorite si urmează a fi cumpărate de utilizatorii precum profesori și studenți, sau așa cum am zis mai sus vor putea introduce codul de 5 cifre comunicat de către ceilalți utilizatori și se va genera o chitanță automat. In cadrul acestei pagini vor fi afișate doar produsele care mai sunt disponibile.

A doua funcționalitate de care dispune această categorie de utilizatori este aceea de a introduce un produs nou în baza de date, prin completarea unor câmpuri în care sunt adăugate informații despre produsul respectiv. Printre alte funcționalități se enumeră cea de adăugare admin nou, în care sunt completate informații despre persoana ce urmează să i se atribuie această funcție și crearea de meniu pentru ziua curentă în care pentru fiecare categorie de produse putem selecta din produsele disponibile din ziua curentă și care fac parte din această categorie de produse. După ce meniul a fost creat, acesta va putea fi vizualizat de către utilizatori. Printre alte funcționalități ale aplicației se enumeră cea de vizualizare a informațiilor despre produs, cea de schimbare a informațiilor unui produs în cazul în care se produc modificări, precum si cea de vizualizare a istoricului vânzărilor dintr-o anumită zi, ce conține informații precum: numărul chitanței, numele produselor, precum si cantitățile produselor vândute din acea zi și prețul total a unei chitanțe.

# CAPITOLUL I. Tehnologii folosite:

Pentru dezvoltarea aplicației „Cantina Gaudeamus” au fost folosite următoarele tehnologii:

* Flutter
  + Sistemul de operare Android
  + Sistemul de operare iOS
* Web API ASP.NET Core
* Angular
* MongoDB

## Flutter framework

*Flutter* este un dezvoltator SDK mobil „open-source”, dezvoltat de Google, care este folosit împreună cu limbajul de programare *Dart*, limbaj orientat-obiect. Este o tehnologie relativ nouă și poate fi folosit pentru a dezvolta aplicații de înaltă performanță pentru Android și IOS cu aspect nativ dintr-un singur cod sursă de bază. Scopul este de a permite dezvoltatorilor să furnizeze aplicații performante pe diferite platforme.

Unul dintre motivele principale pentru care am ales să lucrez cu această tehnologie este portabilitatea, ceea ce înseamnă ca pot să dezvolt aplicația folosind un singur cod sursa de bază pentru Android, cât și pentru iOS, al doilea motiv ar fi că flutter ne ajută să dezvoltăm aplicații de înainte performanțe pe diferite platforme (Android, iOS, Desktop), iar al treilea motiv, dar nu cel din urmă ar fi gradul ridicat de integrare cu alte limbaje de programare.

### Sistemul de operare Android

Android este o platformă software și un sistem de operare pentru dispozitive și telefoane mobile bazată pe nucleul Linux, dezvoltată inițial de compania Google, iar mai târziu de consorțiul comercial Open Handset Allince.

Android permite dezvoltatorilor să scrie un cod gestionat în limbajul Java, controlând dispozitivul prin intermediul bibliotecilor Java dezvoltate de Google. Aplicațiile scrise in C și în alte limbaje pot fi compilate în cod mașină ARM și executate, dar acest model de dezvoltare nu este sprijinit oficial de către Google.

Lansarea platformei Android la 5 noiembrie 2007 a fost anunțată prin fondarea Open Handset Alliance. un consorțiu de 48 de companii de hardware, software și de telecomunicații, consacrat dezvoltării de standarde deschide pentru dispozitive mobile. Google a lansat cea mai mare parte a codului Android sub licența Apache. Google a dezvoltat și alte sisteme de operare bazate pe Android: Wear OS pentru ceasuri inteligente, Android TV pentru SmartTV si Android Auto pentru autoturisme.

Începând cu 21 octombrie 2008, Android a fost disponibil ca Open Source. Google a deschis întregul cod sursă, care anterior era indisponibil, sub licența Apache. Sub licența Apache producătorii sunt liberi să adauge extensii proprietare, fără a le face disponibile comunității open source. În timp ce contribuțiile Google la această platformă se așteaptă să rămână open source, numărul versiunilor derivate ar putea exploda, folosind o varietate de licențe. De asemenea, de-a lungul anilor, au fost dezvoltate numeroase versiuni de android, așa cum se poate vedea în Fig. 1.

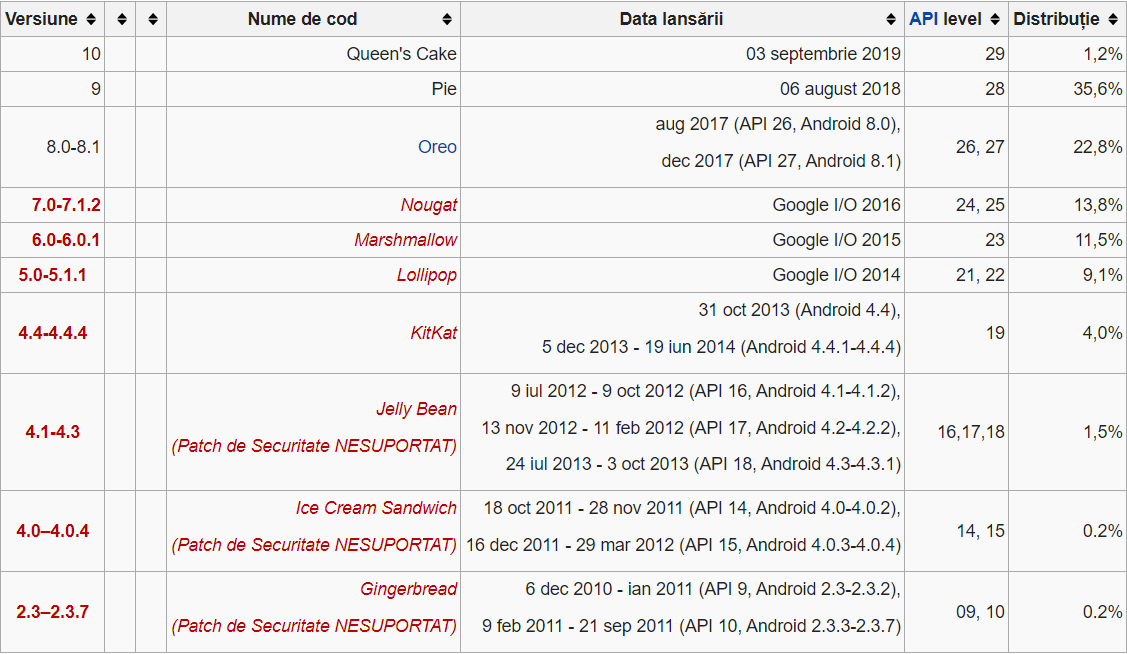
Versiuni Android:

Fig. 1

Sistemul de operare Android cuprinde o gamă largă de caracteristici și funcționalități printre care se enumeră următoarele:

* o interfață ce atrage utilizatorul;
* *stocare,* ce foloseste SQLite, o bază de date relațională ce permite utilizarea eficientă a resurelor;
* *conectivitatea* prin diverse modalități, printer care se enumeră: GSM/EDGE, IDEN, CMDA, EV-DO, UMTS, Bluetooth, Wi-Fi, 4G, NFC, WiMAX;
* *mesagerie* SMS și MMS;
* *navigarea pe Internet* bazată pe motorul open source pentru navigare WebKit împreună cu motorul JavaScript de la Chrome V8 suportând HTML5 și CSS3.
* *multi-touch*, care suportă posibilitatea de contact în mai multe puncte concomitent;
* *multi-tasking*; GCM(Google Cloud Messaging) permițând dezvoltatorilor expedierea de date de dimensiuni reduse, în lipsa unei soluții de sincronizare proprietară.
* *Android Beam*, prin care utilizatorii partajează conținut instant prin apropierea dispozitivelor respective.
* *WiFi direct*, care permite interconectarea între diverse dispozitive, de la o distanță relativ mare, având de asemenea o lățime de bandă mare;
* *multi-tasking*; GCM(Google Cloud Messaging) permițând dezvoltatorilor expedierea de date de dimensiuni reduse, în lipsa unei soluții de sincronizare proprietară.

### Sistemul de operare iOS

iOS este un sistem de operare pentru dispozitive mobile creat si dezvoltat de Apple Inc. Este sistemul de operare care alimentează în prezent multe dispozitive ale companiilor mobile, inclusiv iPhone si iPod Touch. Este al doilea cel mai popular sistem de operare mobil la nivel mondial după Android. Este baza altor sisteme de operare realizate de Apple Inc, cum ar fi iPadOS, tvOS si watchOS.

Dezvoltat inițial în 2007 pentru iPhone de primă generație, iOS a fost extins de atunci pentru a sprijini alte dispozitive Apple, cum ar fi iPod Touch si iPad. Incepând cu luna martie 2018, App Store apple conține mai mult de 2.1 milioane de aplicații iOS, dintre care 1 milion sunt native pentru iPad. Aceste aplicații mobile au fost descărcate de peste 130 de miliarde de ori.

Interfața de utilizator iOS se bazează pe o manipulare directă, folosind gesturi *multi-touch*. Elementele de control alte interfeței constau în glisiere, întrerupătoare și butoane. Interacțiunea cu sistemul de operare include gesturi precum *swipe, tap, pinch si* pinch invers, toate având definiții specifice în contextul sistemului de operare iOS si al interfeței sale *multi-touch*. Apple a fost lăudat în mod semnificativ pentru încorporarea funcțiilor de accesibilitate completă în iOS, permițând utilizatorilor cu dizabilități de vedere și auz să-și folosească corect produsele.

De asemenea, sistemul de operare iOS are numeroase caracteristici și funcționalități printre care se enumeră următoarele:

* *ecranul de pornire,* redat de SpringBoard, afișează pictogramele aplicației în partea de jos în care utilizatorul își poate fixa aplicațiile utilizate cel mai des.
* *centru de notificare,* care permite utilizatorilor să vizualizeze un istoric al notificărilor
* *accesibilitate;* iOS oferă diverse funcții de accesibilitate pentru a ajuta utilizatorii cu dizabilități de vedere și auz. O caracteristică principală este VoiceOver, ce oferă informații despre citirea vocii pe ecran, inclusiv butoane contextuale, pictograme și alte elemente de interfață pentru utilizator și permite acestuia să navigheze în sistemul de operare prin gesturi
* *multi-tasking, comutarea aplicațiilor, finalizarea sarcinilor*
* *Siri* care este un asistent personal inteligent integrat în iOS. Asistentul folosește interogări vocale și o interfață de utilizare a limbajului natural pentru a răspunde la întrebări, a face recomandări și a efectua acțiuni prin delegarea cererilor către un set de servicii Internet. Software-ul se adaptează la utilizările, căutările și preferințele limbajului individual al utilizatorilor, cu utilizare continuă. Rezultatele returnate sunt individualizate. *Siri* acceptă o gamă largă de comenzi ale utilizatorului, inclusiv efectuarea acțiunilor telefonice, verificarea informațiilor de bază, programarea evenimentelor, gestionarea setărilor dispozitivului, căutarea pe internet, navigarea în zone și găsirea informațiilor despre divertisment.
* *rețea de jocuri* *multiplayer* care permite utilizatorilor să invite prietenii să joace un joc, să înceapă un joc prin realizarea meciurilor, să urmărească realizările lor și să compare scorurile lor înalte pe un clasament.
* *platformă hardware* care are arhitectura ARM. Versiunile iOS 7 pot fi rulate numai pe dispozitive iOS cu procesare ARM pe 32 de biți.

## Web API ASP.NET CORE

Serviciile web sunt servere web create speciale pentru a îndeplini nevoile unui site sau a unei aplicații. Programele client folosesc interfața de programare a aplicațiilor (Application Programming Interface – API) pentru a comunica cu serviciile web. General vorbind, un API dispune de un set de date și de acțiuni pentru a facilita interacțiunea dintre anumite programe ale calculatorului și le permite să realizeze schimb de informații între ele.

API reprezintă un set de definiții de sub-programe, protocoale și unele pentru programarea de aplicații software. Un API poate fi pentru un sistem web, sistem de operare, sistem de baze de date, Hardware sau biblioteci software.

REST (Representational state transfer) este o arhitectură software care definește un set de constrângeri care trebuie utilizate pentru crearea serviciilor Web. Serviciile web care se conformează stilului arhitectural REST, numi “RESTful Web services”, asigură interoperabilitatea între sistemele informatice de pe Internet. Serviciile web RESTful permit sistemelor solicitante să acceseze si să manipuleze reprezentările textuale are resurselor Web, folosind un set uniform și predefinit de operații.

Un RESTful API este o interfață de programare a aplicațiilor care utilizează cereri HTTP pentru a obține, a actualiza, a șterge sau a face anumite operații cu anumite date. Un API pentru un site web este cod ce permite ca două programe software să comunice între ele.

=== adaugare imagine

Am ales să folosesc Web API.NET CORE din următoarele motive:

* dezvoltatorul deține controlul asupra transmiterii și răspunsurilor la mesajele protocolului HTTP
* API-urile ASP.NET oferă un nivel ridicat de abstractizare cu ajutorul căruia dezvoltatorii pot crea API-uri web care pot încapsula HttpMessageHandler
* arhitectura API-urilor web este foarte ușoară, ceea ce le face o alternativă perfectă pentru dezvoltatori atunci când doresc să construiască aplicații pentru dispozitive cu lățime de bandă limitată
* este simplu, scalabil si robust, deoarece acceptă toate funcțiile MVC, cum ar fi rutarea, controlerele, rezultatele acțiunii, filtrul sau injecția de dependență

## Angular

Angular este o platformă pentru construirea aplicațiilor client utilizând HTML si TypeScript. Angular este scris cu TypeScript și implementează funcționalități de bază și opționale ca un set de biblioteci TypeScript pe care le importăm din aplicații.

Arhitectura unei aplicații Angular se bazează pe concepte fundamentale. Blocurile de bază sunt NgModules, care oferă un context de compilare pentru componente. NgModules colectează codul aferent de seturi funcționale. O aplicație Angular este definită de un set de NgModule. O aplicație are întotdeauna cel puțin un modul rădăcină.

Componentele definesc pagini web (vizualizări), care oferă funcționalități specifice care nu sunt legate direct de alte pagini web. Furnizorii de servicii pot fi integrați în componente ca și dependențe, ceea ce face codul sursă să fie modular, reutilizabil și eficient. Atât componentele, cât și serviciile sunt pur si simplu clase, cu decoratori care marchează tipul lor și oferă metadate care îi spun platformei cum sa le folosească.

Componentele unei aplicații definesc de obicei mai multe pagini web, aranjate ierarhic. Angular oferă serviciul de rutare (Router) pentru a ne ajuta să definim căile de navigare între paginile web. Rutarea oferta funcții sofisticate de navigare în browser.

## MongoDB

MongoDB este o bază de date ce stochează date *NoSql* (Not only sql), adică o bază de date ne-relațională în care informațiile sunt stocate în documente/colecții, ci nu în tabele. O astfel de bază de date permite stocarea unor volume foarte mari de informații intr-un timp scurt, față de o bază de date *SQL*. O colecție de date MongoDB este similară ca si format cu o dată de tip dicționar ce se folosește in cadrul fișierelor .json.

Avantajul folosirii colecțiilor:

* cea mai naturală si productivă metodă de a lucra cu date
* suporta *array* și obiecte ca și valori
* permite scheme flexibile și dinamice

Exemplu de colecție *MongoDB*:

{

“\_id” : “ “

“firsName” : “Vasile”,

“lastName” : “Lazar”,

“address” : {

“street” : “Mihai Eminescu”,

“city” : “Galati”,

“number” : “361”,

},

“hobbies”:[“tennis”,“football”]  
}

Principalele motive pentru care am folosit această bază de date:

* este un limbaj de interogare bogat și expresiv, care ne permite să filtrăm și să sortăm după orice criteriu
* suport pentru agregări și alte cazuri moderne de utilizare, cum ar fi căutare în grafic și căutare de text
* întrebările sunt și ele la rândul lor sub format JSON, deci sunt ușor de compus
* asistență pentru *join* la interogări
* două tipuri de relații în loc de una: de referință și incorporare
* permite stocarea datelor în colecții sub mai multe forme (Array, Object)

# CAPITOLUL II. Arhitectura aplicației

În acest capitol vom pune în evidență arhitectura aplicației **Cantina Gaudeamus,** principale module aplicației, structura și modul de comunicare între modulele aplicației.

Aplicația a fost dezvoltată astfel încât să ofere o soluție cât mai eficientă și ușor de utilizat de către cei ce o folosesc. Arhitectura este construită astfel încât să utilizeze resurse de memorie cât mai puține și care să ofere un timp de execuție, respectiv de încărcare cât mai scurt.

Aplicația a fost dezvoltată pentru partea de web si pentru cele mai cunoscute sisteme de operare *mobile* de la ora actuală (Android, iOS). Pentru a putea folosi această aplicație este nevoie de conexiune la Internet.

## Structura

Așa cum se poate observa în structura aplicației este una clasică, în care avem o interfață mobile pentru Android, respectiv iOS și web și o „poartă de acces” care face legătura între interfață și baza de date.

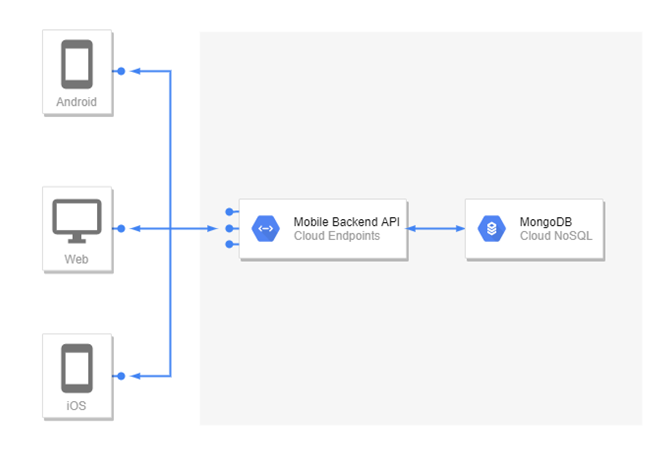


Fig. 2

## Modulele aplicației

În continuare vă voi prezenta modulele aplicației detaliat și ce conțin acestea, respectiv cum funcționează .

### Modulul Interfață Web

Unul dintre cele trei module de interfață este cel pentru Web.

Culorile alese în cadrul acestei interfețe sunt albastru, alb și mov, deoarece albastrul este o culoare rece, movul este o culoare care provoacă senzația de calmare, iar albul simbolizează puritatea, liniștea, toate acestea la un loc tind să fie mai plăcute și mai relaxante.

La prima utilizare, utilizatorul este întâmpinat de pagina de *Login,* în care utilizatorul trebuie să introducă anumite date, acestea fiind adresa de email și parola. De precizat faptul că în cadrul acestei aplicații exista două categorii de utilizatori, aceștia fiind: utilizatori simpli (studenți, profesori, persoane ce frecventează cantina) si administratorul în cadrul componentei de interfață (casierițele, persoane ce fac parte din personalul cantinei)**.** Dacă datele introduse sunt corecte atunci poate începe utilizarea aplicației.



Fig. 3

În cazul în care utilizatorul și-a uitat parola, acesta poate acționa atât în interfața mobile cât și în cea Web butonul *Ai uitat parola / Forgot password* care va duce către pagina de recuperare parola, unde se introduce adresa de email, iar utilizatorul va primi ulterior pe adresa introdusă un mail cu parola.

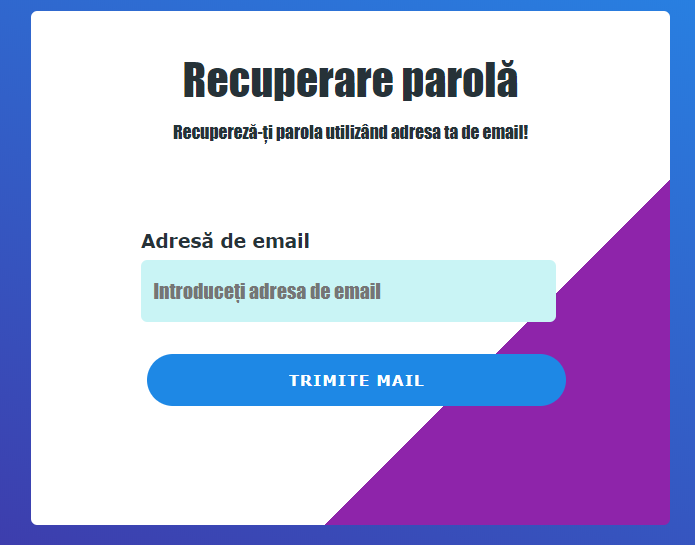


Fig. 4

În Fig. 5 se poate observa pagina de resetare parolă. Aici utilizatorul va trebui să intrudă de două ori aceeasi parolă, astfel încât aceasta să poate să fie modificată. Dupa ce utilizatorul a introdus aceeași parolă de două ori, acesta va fi redirecționat către pagina de *Login* pentru a se putea autentifica.



Fig. 5

În cazul in care utilizatorul nu are deja un cont, in partea de jos a paginii *Login* există un buton de acțiune ce redirecționează către pagina de *Register*. Principala funcționalitate în cadrul acestei pagini este aceea de înregistrare a unui utilizator nou prin completarea a trei câmpuri: *email, parola* și *confirmare parolă.* De precizat faptul că *parola* și *confirmare parolă* trebuie să coincidă și să fie identice.



Fig. 6

Pentru utilizator, după ce acesta se conectează la aplicație, va fi întâmpinat de un ecran asemenea unui meniu, ce conține informații despre produsele disponibile din ziua curentă, informații precum: nume, descriere/ingrediente, preț. De asemenea fiecare produs este însoțit de două butoane. Primul buton este cel de adăugare produs, iar al doilea este cel de ștergere produs. În cadrul acestei pagini utilizatorul poate pur și simplu doar să se informeze să vadă ce produse sunt disponibile in ziua curentă, sau poate sa aleagă din lista de produse disponibile.

In partea de jos a paginii se află un container ce conține informații despre prețul total al produselor ce au fost selectate și un buton *Generare cod*, care după cum spune, după ce utilizatorul a finalizat de selectat produsele ce dorește să le cumpere, acționând acest buton va fi generat un cod de 5 cifre care va apărea în partea de sus a paginii, acest cod poate fi folosit ulterior în momentul în care utilizatorul ajunge la casă și nu mai trebuie să comunice casieriței ce produse dorește să cumpere, ci va comunica doar codul generat, care va genera chitanța automat așa cum se descrie in partea ce urmează.

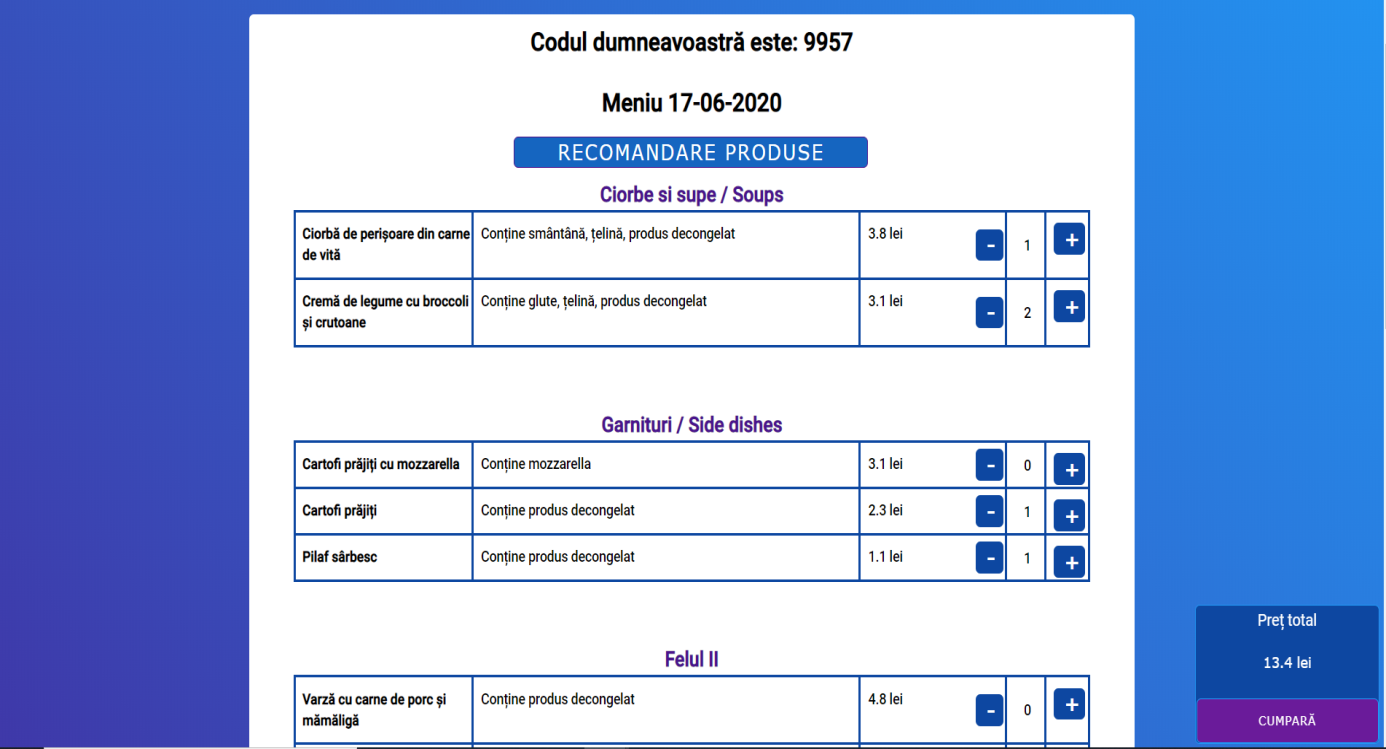


Fig. 7

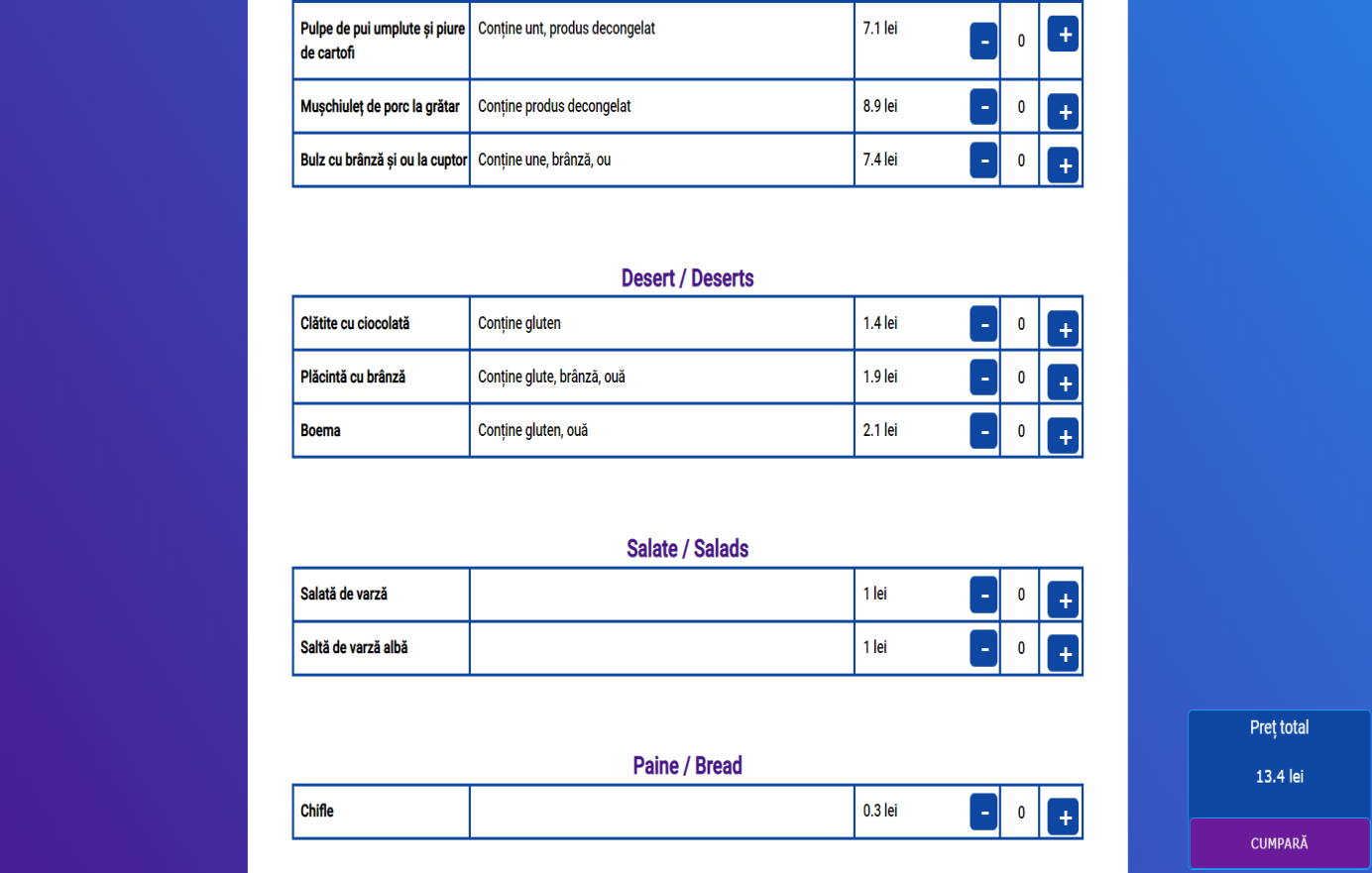


Fig. 8

Pentru administratorul în cadrul componentei interfeței, după ce acesta se conectează la aplicație, va fi întâmpinat de un ecran ce conține în partea de sus(header) numele aplicației și doua butoane: *Logout* și *Users* ca în Fig. 9. Acționarea butonului *Logout* va redirecționa administratorul către pagina de *Login*, iar acționarea butonului *Users* va redirecționa administratorul către pagina utilizatorului.



Fig. 9

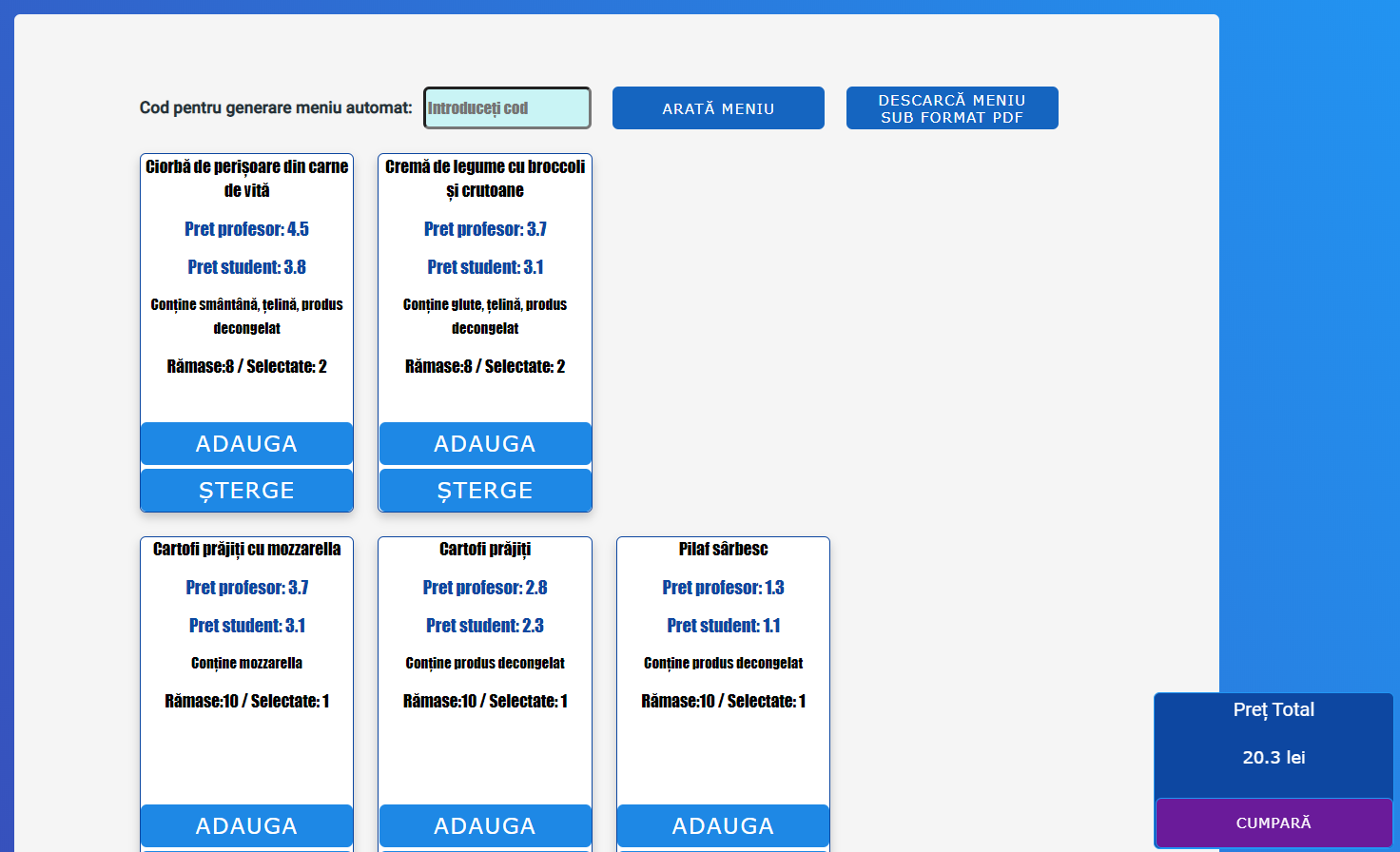
În partea stânga a paginii avem o bară de navigare ce permite redirecționarea către alte pagini ale aplicației, acestea fiind: *Magazin, Creare meniu, Adăugare produs, Detalii produse, Actualizare produse, Adăugare admin, Istoric vânzări.* Bara de navigare se poate vedea în Fig. 10.



Fig. 10

După ce realizează conectarea, administratorul este întâmpinat de pagina *Magazin* din bara de navigare a aplicației. Această pagina cuprinde o listă cu produsele disponibile în ziua curentă. Fiecare produs are informații despre: nume, preț profesor, preț student, descriere/ingrediente, precum și câte produse de acest fel sunt disponibile la momentul respectiv. De asemenea fiecare produs conține doua butoane, unul de adăugare produs și altul de ștergere produs. Acționarea acestora adaugă, respectiv șterge produsul selectat din lista celor ce urmează a fi cumpărate.

Rolul administratorilor interfeței în cadrul acestei pagini este de a selecta produsele ce doresc a fi cumpărate de către utilizatori și de a genera chitanța. Însă pentru a realiza acest lucru am pus la dispoziție o metoda mai simplă și mult mai rapidă, și aceea de a introduce in câmpul de sus al paginii un cod de 5 cifre care a fost generat în pagina utilizatorilor. Acest cod va selecta automat produsele dorite de către utilizator și va genera chitanța in funcție de produsele selectate. Acest lucru se face prin acționarea butonului *Arata meniu*. În partea dreaptă din jos a paginii, se află un container ce conține informații despre prețul total în urma selectării produselor și un buton *Cumpără* ce generează automat chitanța. Mai jos se poate vedea tot ceea ce am descris până aici.



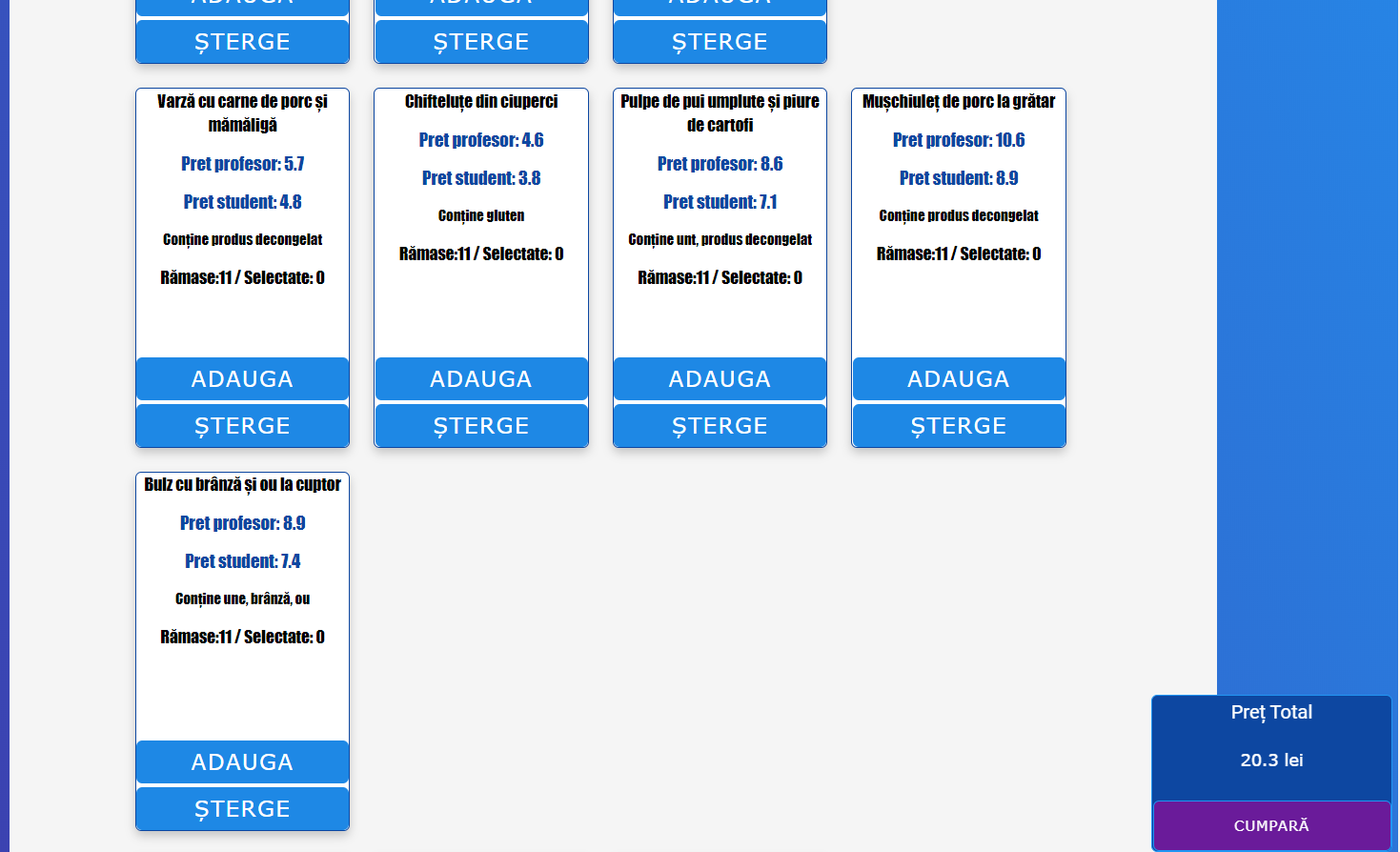


Fig. 11

Al doilea buton din bara de navigare din partea stângă a paginii este cel de *Creare meniu,* în care după cum spune și numele, administratorul poate crea meniul pentru ziua curentă sau pentru altă zi. Această pagină conține o serie de câmpuri ce sunt necesare a fi completate de către administrator. Primul câmp este cel de introducere a datei in care se dorește să fie afișat meniul. În următoarele câmpuri pentru fiecare categorie de produse, se pot adăuga produsele dorite și numărul de produse disponibile de acest tip. Categoriile de produse sunt: *Ciorbe, garnituri, Felul II, salate, apă, etc.* De asemenea, pentru fiecare categorie de produse există și un buton de *Adăugare produs*, care in momentul în care este acționat se poate adăuga un nou produs pentru categoria respectivă. La final, după ce au fost selectate produsele, se poate crea meniul, acționând butonul *Creare meniu,*care va crea automat meniul ce va putea fi vizualizat de către utilizatori în ziua curentă. Pagina de creare meniu se poate vedea în Fig. 12.

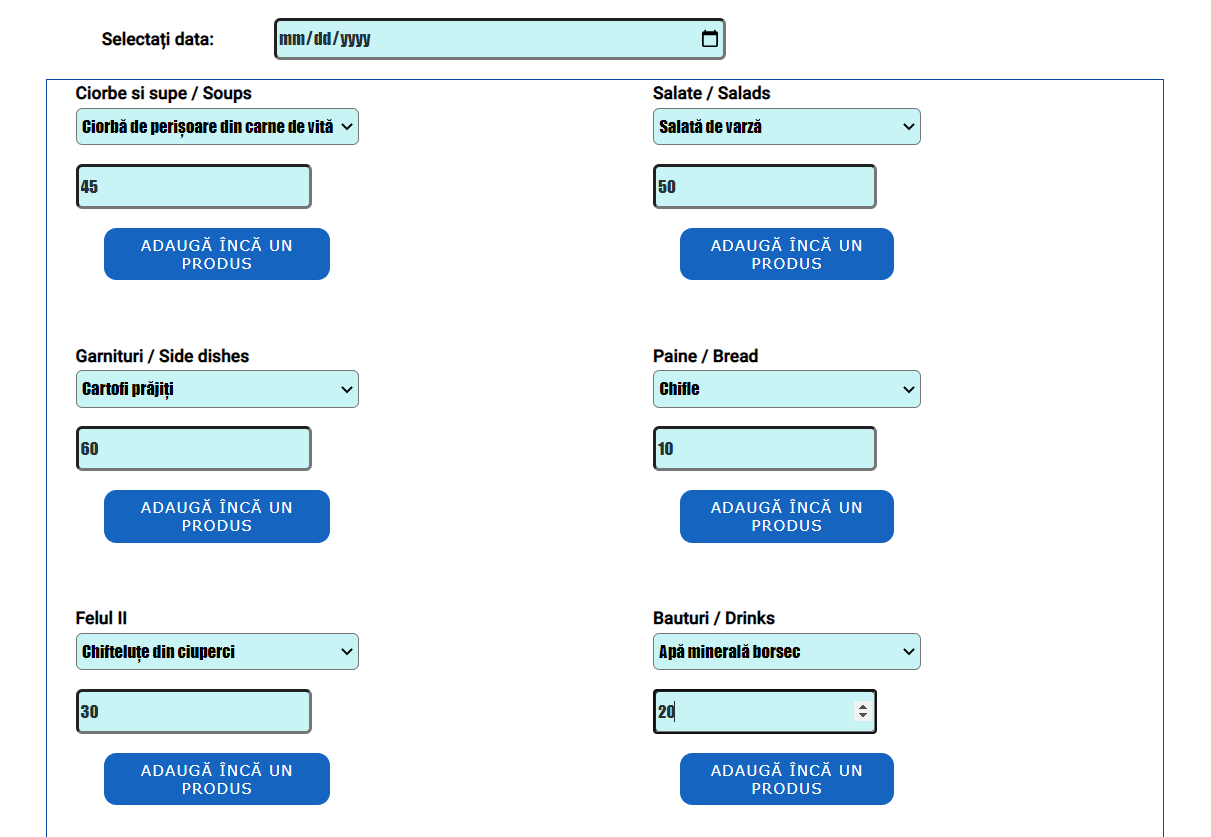


Fig. 12

Al treilea buton din bara de navigare din partea stângă a paginii este cel de *Adăugare produs*, care ne va redirecționa către pagina în care *vrem* sa adăugam un produs nou în lista de produse deja disponibile, care poate fi folosit în momentul în care în cadrul cantinei a apărut un produs nou și se dorește a fi pus la cumpărare. Această pagină conține anumite câmpuri ce trebuie completate de administrator. Aceste câmpuri sunt: *nume produs, categorie produs, preț profesor, preț student, descriere, gramaj.* În momentul în care câmpurile au fost completate, produsul va fi adăugat automat în baza de date. Pagina de *Adăugare produs* poate fi văzută în Fig. 13.

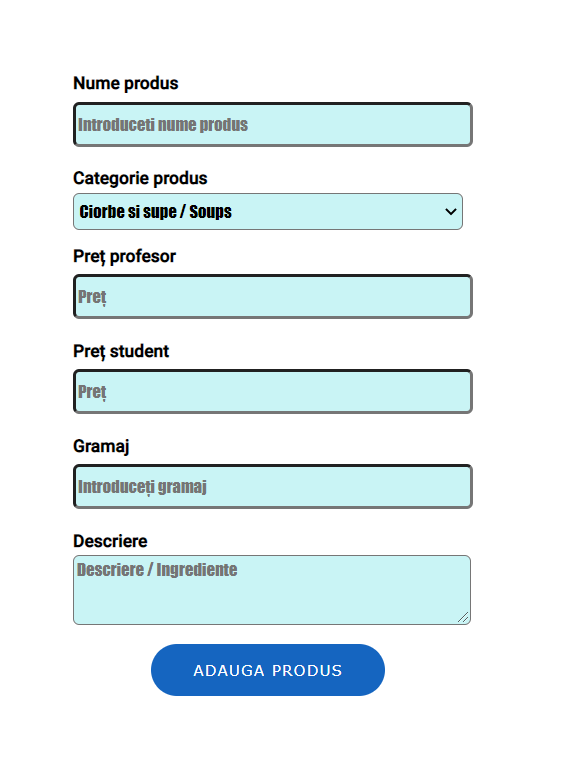


Fig. 13

Cel de-al patrulea buton din bara de navigare este cel de detalii produse, ce oferă un serviciu de căutare a produselor din lista totala de produse și conține o listă cu toate produsele, precum si informațiile fiecărui produs așa cum se poate vedea în Fig. 14.

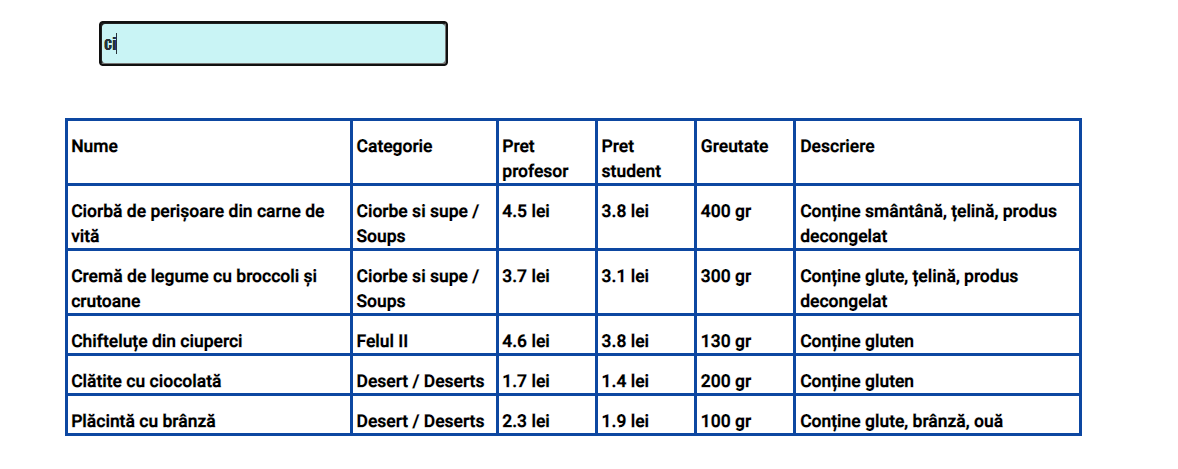


Fig. 14

Cel de-al cincilea buton din bata de navigare este cel de *Actualizare produse*. Acesta poate fi folosit, în special, în cazul in care vrem să modificam informațiile despre un anumit produs. Se completează doar câmpurile ce se doresc a fi modificate. În cazul în care nu va fi modificată nicio informație atunci se va afișa o eroare. Această pagina arată ca în figura ce urmează.

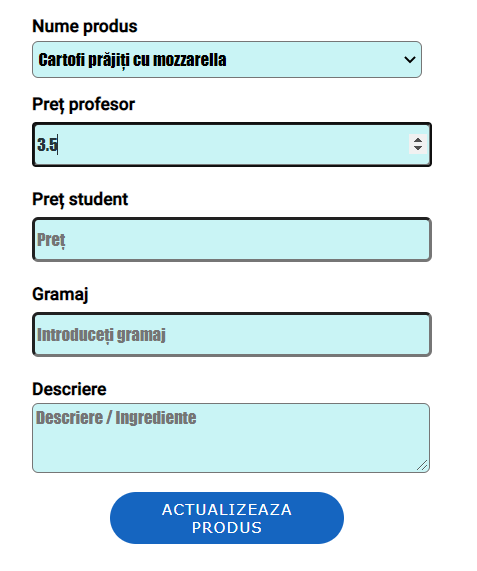


Fig. 15

Cel de al șaselea buton (Fig. 16) din bara de navigare este cel în care administratorul trebuie să introducă o adresă de email, iar persoana ce are acea adresă de email va primi la rândul ei rolul de administrator în cadrul interfeței.

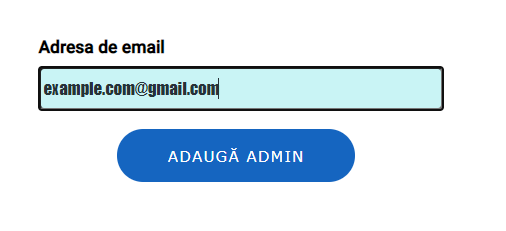


Fig. 16

Ultimul, dar nu cel din urmă buton din bara de navigare este cel de *Istoric vânzări.* În cadrul acestei pagini utilizator trebuie să aleagă o dată din care dorește să obțină informațiile despre vânzările din ziua respectivă. Informațiile ce vor fi afișate pentru fiecare vânzare din ziua selectată sunt: codul chitanței, numele produselor ce au fost vândute, precum și numărul de produse vândute și prețul total pentru fiecare vânzare/chitanță în parte. Istoricul arată așa cum puteți vedea în Fig. 17.

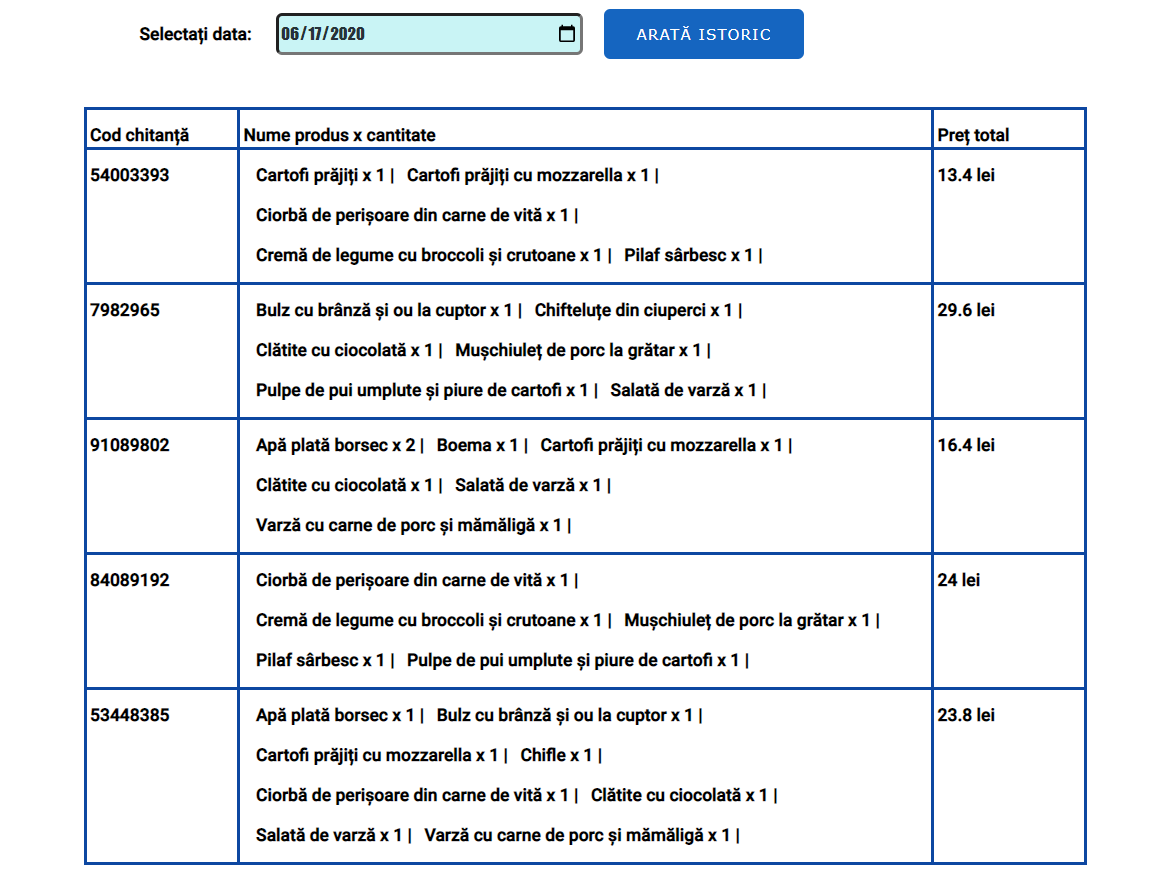


Fig. 17

### Modulul interfața Android și iOS

Acest modul a fost dezvoltat doar pentru utilizatori, nu și pentru administratorii în cadrul componentei interfeței. Paginile din această componentă sunt similare cu cele descrise în 2.2.1. Paleta de culori aleasă pentru această componentă au fost movul și albastrul, ce tind să fie mai plăcute și mai relaxante. La fel ca și la modulul de interfață web, aici avem o pagină de *Login*, în care atât utilizatorii cât și administratorii vor introduce numele și parola corecte, după care vor fi redirecționați către pagina unde se află meniul, în cazul în care aceștia au introdus datele lor corecte. În caz contrar, va fi afișat un mesaj de eroare. Pagina de *Login* din cadrul acestui modul poate fi vizualizată în Fig. 18 de mai jos.

|  |  |
| --- | --- |
| Fig. 18 | Cum am menționat și in 2.2.1, există și aici un buton în care utilizatorul va fi redirecționat către pagina de *Register*, în cazul în care acesta nu are un cont existent in baza de date. De asemenea există și un buton de *Forgot password* în cazul neplăcut în care utilizatorul și-a uitat parola și dorește să o recupereze. Paginile de *Register*, respectiv *Forgot password* pot fi vizualizate în Fig. 19 și Fig. 20. |

|  |  |
| --- | --- |
| Fig. 19 | Fig. 20 |

După ce utilizatorul s-a autentificat cu succes pe aplicația din cadrul interfeței mobile, acesta va fi întâmpinat de un meniu, ce conține informații despre produsele disponibile spre vânzare din ziua curentă. In partea de sus a paginii utilizatorul dispune de un buton de deconectare-*Logout* și un text ce conține informații despre data din care este afișat meniu. Informațiile din cadrul meniului sunt: numele produslui, prețul produsului, descriere produs și numărul de produse selectate. De asemenea utilizatorul dispune de 2 butoane; unul de adăgare produs și altul de ștergere produs. Deasupra meniului utilizatorul dispune de un buton de recomandare produse. Rolul acestuia este de produsele ce au mai fost cumpărate anterior de către utilizator. În partea de jos a paginii se află un text cu informații despre prețul total, precum și un buton de *Buy*. Acționarea acestui buton conduce la generarea unui cod de 5 cifre care va fi afișat ulterior deasupra meniului. Tot ce am desris în cadrul acestei pagini se poate vedea în Fig. 21, respectiv Fig. 22.

|  |  |
| --- | --- |
| Fig. 21 | Fig. 22 |

## Modulul bază de date

Baza de date are un rol foarte important în cadrul unei aplicații, rolul acesteia fiind cel de stocare a informațiilor si de asigurare a integrității datelor. Pentru acest modul am ales o bază de date *NoSQL MongoDB****,*** deoarece datele sunt stocate în colecții ce au un format asemănător unei date de tip dicționar ceea ce le face mult mai ușor de utilizat și datorită complexității timp și spațiu ale obținerii de soluții.

Baza de date este formată dintr-un număr de 4 colecții, după cum se poate observa și în figura == de adaugat diagramă baze de date

Prima colecție este cea a utilizatorilor - *userlist*. În cadrul acestei colecții, în fiecare document sunt stocate informații despre fiecare utilizator în parte; pentru fiecare utilizator avem un identificator unic *\_id* de tipul int32, *email* și *password*  care sunt de tip *string*, ce reprezintă adresa de email al fiecărui utilizator în parte, respectiv parola. De menționat faptul că *password* este un hash SHA-256 de lungime fixă de 64 caractere. De asemenea pentru fiecare utilizator mai avem *type* și *role* de tip *string* ce reprezintă tipul utilizatorului, respectiv rolul acestuia, precum și un *token* care este un *string* de dimensiune mare.

Cea de-a doua colecție este cea a produselor - *products*. În cadrul acestei colecții sunt stocate informații despre fiecare produs în parte; de asemenea și pentru acestea avem un identificator unic *\_id* de tip int32. Fiecare document mai conține și următoarele câmpuri: *name, category, description*  ce sunt de tip *string* și conțin informații despre numele, categoria, respectiv descrierea produsului și *professorPrice, studentPrice, weight* ce sunt de tip *int*, respectiv *double* și conțin informații despre prețul pentru profesori, cel pentru student și gramajul.

Cea de-a treia colecție este cea a meniurilor ­- *menus*. Pentru fiecare document din cadrul acestei colecții avem un identificator unic *\_id* de tip *int*, *dateMenu* de tip *Date* ce reprezintă data în care a fost creat meniul respectiv și *productsIdAndAmounts* care este de tipul *Object* și conține informații sub forma unui dicționar în format JSON, în care cheia reprezintă id-ul produsului, iar valoarea reprezintă cantitatea disponibila pentru produsul respectiv în acea zi.

Cea de-a patra colecție este cea a comenzilor – *orders.* La fel ca la colecțiile prezentate mai sus, și pentru această colecție avem un identificator unic *\_id* de tip int. De asemenea mai avem următoarele: *idUser* de tip *int* ce reprezintă id-ul utilizatorului pentru care a fost generat acest cod; *idProducAndAmount* de tip Object, ce reține informații cantitatea ce se dorește a fi cumpărată pentru fiecare produs în parte; *code* de tip *int32* ce reprezintă codul generat automat care mai târziu poate fi folosit pentru achiziționarea de produse; *date* de tip *Date* ce reține data în care a fost generată comanda respectivă și *totalPrice* de tip *double* ce reprezintă prețul total asociat comenzii.

Ultima colecție este cea a istoricului comenzilor – *history.* Acest document conține un identificator unic *\_id* de tip *int32* de 6 cifre, ce reprezintă numărul chitanței. De asemenea acesta reține data în care a fost generată această chitanța – *date,* numele produselor precum și cantitatea acestora într-un obiect sub format JSON – *nameProductsAndAmounts,* precum și prețul total al acestora – *totalPrice.*

*== de adăgat schema*

## Modulul API *Gateway*

Pentru implementarea acestui modul am folosit modelul de lucru *ASP.NET CORE* împreună cu limbajul de programare C#.

=== de ce am folosit aceasta tehnologi?

=== la fel, de completat după ce termin aplicația în întregime

# CAPITOLUL III. Detalii de implementare

=== aici voi explica cum am gândit codul și ce am făcut la fiecare parte de cod

# Concluzii

# Bibliografie

Kouwer, B. J. (1949). *Colors and Their Character: A Psychological Study.*