**UNIVERSITATEA ROMÂNO – AMERICANĂ**

**FACULTATEA DE INFORMATICĂ MANAGERIALĂ**



REALIZAREA UNEI APLICAȚII INFORMATICE PENTRU GESTIUNEA ACTIVITĂȚII UNEI CAFENELE

Coordonator științific:

Conf.univ.dr Crișan Daniela Alexandra

Absolvent:

Mușețeanu T Corneliu

București 2024

Cuprins

[Introducere 3](#_Toc167792462)

[Oportunitatea și importanța temei propuse 3](#_Toc167792463)

[Cap. 1. Studiul și analiza sistemului existent 4](#_Toc167792464)

[1.1. Prezentarea succinta a unității economico-sociale 4](#_Toc167792465)

[1.2. Principalele activități desfășurate în unitatea economică 5](#_Toc167792466)

[1.3. Studiul sistemului de conducere 6](#_Toc167792467)

[1.4. Studiul sistemului condus 8](#_Toc167792468)

[1.5. Studiul sistemului informațional 9](#_Toc167792469)

[1.5.1. Schema fluxului informațional 9](#_Toc167792470)

[1.5.2. Descrierea circuitului informațional 10](#_Toc167792471)

[1.5.3. Descrierea documentelor utilizate 11](#_Toc167792472)

[1.5.4. Modelul conceptual al prelucrărilor 18](#_Toc167792473)

[1.5.5. Analiza critică a sistemului actual și identificarea neajunsurilor existente în funcționarea sistemului existent 20](#_Toc167792474)

[1.5.6. Direcții de perfecționare a sistemului actual 21](#_Toc167792475)

[Cap. 2. Proiectarea de detaliu a aplicației informatice 22](#_Toc167792476)

[2.1. Definirea obiectivelor aplicației informatice 22](#_Toc167792477)

[2.2. Proiectarea logică și fizica a ieșirilor 23](#_Toc167792478)

[2.2.1 Lista cu toate documentele de ieșire din aplicație 23](#_Toc167792479)

[2.2.2 Machetele tuturor documentelor de ieșire 24](#_Toc167792480)

[2.3. Proiectarea logică și fizica a intrărilor 31](#_Toc167792481)

[2.3.1 Lista cu toate intrările din aplicație 31](#_Toc167792482)

[2.3.2 Machetele pentru toate video formatele din aplicația informatica 32](#_Toc167792483)

[2.4. Proiectarea sistemului de codificare a datelor 37](#_Toc167792484)

[2.5. Proiectarea bazei de date 38](#_Toc167792485)

[2.5.1 Schema relațională a bazei de date 44](#_Toc167792486)

[2.6. Schema de sistem a aplicației 45](#_Toc167792487)

[2.7. Proiectarea interfeței aplicației 46](#_Toc167792488)

[2.8 Alegerea tehnologiei de prelucrare 47](#_Toc167792489)

[2.9 Estimarea necesarului de resurse și a calendarului de realizare 48](#_Toc167792490)

[2.9.1 Activitățile și necesarul de resurse 48](#_Toc167792491)

[2.9.2. Diagrama Gantt 50](#_Toc167792492)

[Cap. 3. Prezentarea produsului software 51](#_Toc167792493)

[3.1 Cerințele platformei hardware și software ale produsului 51](#_Toc167792494)

[3.2 Descrierea funcțiunilor aplicației 52](#_Toc167792495)

[Cap. 4. Eficiența și utilitatea aplicației informatice 54](#_Toc167792496)

[4.1 Condiții privind implementarea aplicației 54](#_Toc167792497)

[4.2 Exploatarea curenta a aplicației 54](#_Toc167792498)

[4.3 Considerații privind eficiența aplicației informatice 54](#_Toc167792499)

[5. Anexe 54](#_Toc167792500)

[6. Bibliografie 55](#_Toc167792501)

# Introducere

## Oportunitatea și importanța temei propuse

Tehnologia se dezvoltă foarte rapid în ultimii ani, asta a condus la faptul că majoritatea procedurilor fizice din procesele de muncă au fost înlocuite cu procese automatizate prin intermediul unor sisteme informatice care a permis în primul rând că etapele pe care le-a înlocuit să devină mult mai rapide și să apară mult mai puține erori care puteau fi cauzate în trecut de factorul uman.

Această lucrare are că obiectiv crearea și implementarea unui sistem informatic în activitatea unei cafenele ‚SD BAR S.R.L’ cu scopul de a automatiza anumite procese pentru a eficientiza procesul de muncă din această societate cu răspundere limitată.

Scopul acestei lucrări este de a crea un sistem informatic cu o bază de date în spate, aplicația informatică va reprezenta un site prin care se vor accesa toate funcționalitățile acesteia. Clienții se vor putea înregistra în aplicația informatică. Motivul pentru care clienții trebuie să se autentifice este pentru a avea acces la istoricul comenzilor, de a putea efectua și vizualiza rezervările și pe viitor ca o etapă de perfecționare a sistemului informatic de a beneficia de reduceri in funcție de cât de fidelitatea acestora.

# Cap. 1. Studiul și analiza sistemului existent

## Prezentarea succinta a unității economico-sociale

“SD BAR SRL” este o societate comercială cu răspundere limitată înființata în anul 2007 de către o singură persoană, firma face parte din industria ospitalității având codul CAEN 5630 - ‚Activitatea în domeniul barurilor și servirii de băuturi’. Firma dată reprezintă un bar care furnizează servicii de vânzarea de băuturi și produse alimentare de complexitate mică, adică în mare parte gustări pentru băuturile oferite.

Obiectivul acestei firme este de a oferi clienților să-i un mediu cât mai relaxant unde ar putea să-și petreacă timpul cu cei apropiați pentru a se bucură de diversele băuturi alcoolice și non-alcoolice cât și de meniul diversificat.

Pârțile implicate:

* Personalul de conducere – persoanele care gestionează diferite aspecte legate de funcționarea corecta și maximizarea profitului firmei.
* Angajații – persoanele care lucrează la aceasta societate și care au ca rol principal servirea clienților.
* Furnizorii – întreprinderile de la care se face aprovizionarea necesară activității.
* Clienții – persoanele fizice care beneficiază de serviciile oferite.

## Principalele activități desfășurate în unitatea economică

În societatea „SD BAR S.R.L” sunt desfășurate următoarele activități:

1. Preluarea comenzilor de către chelneri – această activitatea se efectuează în pașii următori. Clientul vine la localul dat, daca dorește să se rețină pe o perioada mai îndelungată se poate așeză la o masă, daca nu poate face comanda direct la casă. Comanda este preluată de către un chelner care pregătește comanda și o transmite clientului, clientul cere nota de plată sau plătește direct la casa de marcat și primește bonul fiscal.
2. Efectuarea rezervărilor – clienții pot efectua și rezervări la local pentru a se asigura că vor avea o masă liberă pentru ei la ora rezervată. Rezervarea se face telefonic.
3. Aprovizionarea de mărfuri – de partea de aprovizionare se ocupă biroul de aprovizionare, acesta verifica zilnic numărul produselor din gestiune și dacă se observă că un produs din gestiune are stocul foarte mic se efectuează o comanda la furnizor care livrează comanda și biroul de aprovizionare o preia de la acesta.

A screenshot of a computer screen

Description automatically generated

*Figura 1, Indicatorii economici*

## Studiul sistemului de conducere

Organigrama firmei:

A black background with white rectangles

Description automatically generated

*Figura 2, Organigrama firmei*

**Studiul sistemului de conducere**

**Proprietar**

* Responsabilități:
  + Deținătorul firmei
  + Responsabil pentru obținerea licențelor necesare pentru operarea localului
  + Asigurarea securității în local
  + Conducerea localului

**Manager General**

* Responsabilități:
  + Recrutarea și gestionarea personalului
  + Formarea graficului de muncă a personalului
  + Implementarea standardelor de serviciu
  + Gestionarea relațiilor cu clienții (obținerea de feedback asigurându-se că cererile și plângerile sunt abordate în mod corespunzător)
  + Raportarea către proprietar

**Manager Bar**

* Responsabilități:
* Gestionarea operațiunilor zilnice legate de tot ce ține de partea de bar (adică verificarea stocului pentru bar)
* Supravegherea subordonaților
* Menținerea unei calități înalte a tuturor comenzilor trimise către clienți

## Studiul sistemului condus

**Biroul de aprovizionare**

Este reprezentat de o singură persoană care are că responsabilitate verificarea stocurilor zilnice și aprovizionarea acestora în cazul în care sunt sub numărul minim necesar stabilit, acesta face comenzile către furnizori și se asigura la primirea mărfii că totul să fie în ordine.

**Contabil**

Persoana care se ocupă de gestionarea contabilității generale, emiterea salariilor către angajați și asigurarea conformității legale (asigurarea că toate activitățile financiare ale barului respectă legile și reglementările fiscale locale și naționale.

**Bucătar**

Este responsabil de gestiunea bucătăriei, prepararea gustărilor mici în avans pentru restul zilei și a prânzurilor în perioada stabilită în care acestea pot fi comandate și asigurarea că toate comenzile trimise în sală sunt la o calitate înaltă.

**Ajutor bucătar**

Ajutorul bucătarului, se ocupă de pregătirea ingredientelor necesare pentru prepararea bucatelor.

**Ospătar**

Preia comenzile de la clienți din sală, le duce la bar și duce comenzile către mese. Se asigură că clienții de la mesele pe care le servesc să nu plece fără a achita.

**Ajutor ospătar (picol)**

Se ocupă de curățarea meselor după plecarea clienților și are grijă că pe mese să fie toate tacâmurile necesare pentru următorii clienți.

**Barman**

Preia comenzile de la ospătari și prepară băuturile alcoolice și non-alcoolice din comenzi.

**Personalul de curățenie**

Se ocupă de menținerea curățeniei generale a localului.

## Studiul sistemului informațional

### A screenshot of a computer screen Description automatically generatedSchema fluxului informațional

*Figura 3, Schema fluxului informațional*

### Descrierea circuitului informațional

1. Verificarea stocului cu ajutorul documentului, registru de stoc.

2. Furnizorul vine cu un catalog digital de produse disponibile.

3. Se completează formularul de comandă de la furnizor.

4. Comanda este preluată de către furnizor.

5. Furnizorul livrează comanda.

6. Preluăm factura pe comandă de la furnizor.

7. Se preia marfa de la furnizor.

8. Se adaugă produsele comandate în registrul de stoc.

9. Clientul plasează comanda la local.

10. Angajatul notează comanda pe o foaie de comandă.

11. Comanda se pregătește.

12. Se actualizează registrul de stoc dacă este cazul.

13. Comanda este transmisa clientului la masă.

14. Clientului primește de la ospătar nota de plată.

15. Clientul efectuează plata la casă.

16. Clientul primește bonul fiscal.

### Descrierea documentelor utilizate

* **Registrul de stoc**

A table of data with numbers

Description automatically generated with medium confidence

*Figura 4, Registrul de stoc*

Registrul de stoc este folosit pentru gestionarea stocurilor din bar, este actualizat la fiecare aprovizionare și la perioade stabilite, de obicei la sfârșitul zilei pentru a ține la curent stocul din local. Este structurat în forma unui tabel în care putem introduce numele produsului, intrările, ieșirile și stocul produsului dat. Cu ajutorul acestui document se i-au deciziile în legătură cu ce cantitate trebuie aprovizionată de la furnizori.

* **Catalogul de produse digitalizat**

A screenshot of a product

Description automatically generated

*Figura 5, Catalog de produse digitalizat*

*Figura 5, Catalog produse (temporar)*

Catalogul de produse este documentul prezentat de către furnizor către biroul de aprovizionare pentru a vedea ce produse sunt disponibile în stocul furnizorului, dacă au introdus produse noi sau daca sunt oferte în perioada dată. Cu ajutorul acestui document se realizează formularul de comanda care este transmis furnizorului.

* **Formularul de comanda**

A document with a list of information

Description automatically generated with medium confidence

*Figura 6, Formularul de comanda*

Documentarul reprezentat în figura 6 este folosit pentru a preciza produsele pe care dorim sa le comandăm de la furnizor:

* + În colțul din stânga sus avem datele companiei care face comanda cu câmpurile: Numele companiei, CUI, Adresa și numărul de înregistrare.
  + În colțul din dreapta sus sunt datele furnizorului.
  + Secțiunea de expediere sunt datele despre expediere care include: adresa codul poștal, strada și informațiile despre plata.
  + Tabelul în care se completează datele produselor care trebuie comandate.
  + În partea de jos se pune semnătura directorului și a contabilului.
* **Factura pe comandă**

A close-up of a document

Description automatically generated

*Figura 7, Factura pe comandă*

Acest document este emis de către furnizor în urma finalizării comenzii, documentul conține în partea de stânga sus datele furnizorului, dreapta sus datele cumpărătorului. Sub denumirea documentului „Factură” avem datele facturii cum ar fi numărul facturii, data și numărul aviz de însoțire a mărfii. Mai jos este prezent un tabel care conține următoarele secțiuni: Nr. crt. Denumirea produselor, U.M (unitatea de măsurare), Cantitatea, Prețul per unitate, Valoare fără TVA și valoarea cu TVA.

Mai jos găsim date privind expediția, totalul de plată. La primirea comenzii dacă totul este în regulă punem și semnătură de primire în partea dreaptă jos.

* **Foaia de comandă**

A screenshot of a phone

Description automatically generated

*Figura 8, Foaia de comandă*

Foaia de comandă în cazul dat, prezentată printr-o macheta, în prezent foaia de comandă este o foaie dintr-un blocnot în care ospătarul scrie ce a comandat clientul și o duce mai departe la bar pentru a se pregăti comanda. Acest document conține masa care a făcut comanda, produsele alimentare comandate, data și numărul comenzii.

* **Nota de plată**

A blank form with a number

Description automatically generated with medium confidence

*Figura 9, Nota de plată*

Nota de plată este documentul prin care se face plata la casă, acesta este cerut de către client, ospătarul completează nota de plată în funcție de ce a comandat clientul. Sus avem datele localului, după care urmează numărul la nota de plată, acesta de obicei semnifică numărul comenzii din ziua respectivă, mai jos avem numărul mesei și numele ospătarului care a servit masa, după urmează un tabel care este completat cu ce a comandat clientul la final fiind totalul comenzii.

* **Bonul fiscal**

A receipt on a counter

Description automatically generated

*Figura 10, Bonul fiscal***A black background with white ovals

Description automatically generated**

Documentul de mai sus este un exemplu de cum trebuie sa arate un bon fiscal emis de casa de marcat, acesta conține în partea de sus datele localului, adresa și codul fiscal, mai jos avem numărul bonului, urmat de produse și cantitatea care a fost comandată și în partea dreapta a acestora avem prețul în funcție de cantitate. Urmează totalul, numele angajatului de la casa și numărul casei. În partea de jos avem total TVA, cotă TVA și data în care a fost emis acest document și ora, urmat la sfârșit de tot Seria și numărul bonului acestea fiind unice.

### Modelul conceptual al prelucrărilor

**A black screen with white ovals

Description automatically generated**

**A black background with white ovals

Description automatically generated**

*Figura 11, Modelul conceptual al prelucrărilor***A black background with white ovals

Description automatically generated**

### Analiza critică a sistemului actual și identificarea neajunsurilor existente în funcționarea sistemului existent

După analiza sistemului informațional actual al societății comerciale SD BAR S.R.L s-au identificat următoarele neajunsuri în funcționarea sistemului existent:

1. Registrul de stoc se actualizează manual, completarea registrului de stoc manuală este un neajuns semnificativ al sistemului existent deoarece ocupă mult timp, actualizarea acestuia se efectuează doar odată pe zi ceea ce poate duce la epuizarea stocului unui produs fără a se observa din timp acest lucru. Se pot produce erori umane care pot afecta deciziile luate la etapa de aprovizionare, sau în cel mai rău caz poate sa fie distrus ceea ce înseamnă că va trebuie sa se întocmeze de la capăt cu toate produsele din stoc.
2. Documentul foaia de comanda în sistemul actual este reprezentat de un blocnot în care este notată comanda manual de către ospătar, aceasta foaie poate fi pierdută ceea ce duce la creșterea timpului de finalizare a unei comenzi.
3. Documentul nota de plată tot se completează manual, din aceasta cauză pot apărea erori umane ceea ce poate duce la conflicte cu clienții și ocupă mult timp.
4. Plata de către client se efectuează doar la casa de marcat, tot un neajuns al sistemului existent datorită faptului că la orele de vârf există un flux foarte mare de clienți care blochează temporar activitatea de preluare și preparare a băuturilor, deoarece aceste activități sunt făcute de aceeași persoană care se ocupă cu prepararea produselor din comandă.
5. Comenzile nu sunt înregistrate în nici un sistem, aceasta poate fi un neajuns al sistemului atunci când se efectuează schimbul de ture la angajați din cauza faptului că în schimbul turelor de muncă există comenzi nefinalizate, informațiile despre aceste comenzi trebuie transmise următorului angajat. în sistemul existent această procedură se efectuează prin ajutorul unui blocnot în care se scrie masa și ce produse au fost comandate la masa respectiva, din cauza acestei metode exista erori umane care pot duce la erori în legătură cu integritatea unor comenzi.

### Direcții de perfecționare a sistemului actual

Luând în considerare neajunsurile determinate în subcapitolul anterior vom avea

următoarele direcții de perfecționare pentru fiecare neajuns determinat:

* Registrul de stoc actualizat manual. O îmbunătățire a acestei etape din fluxul informațional ar fi automatizarea acestui proces prin implementarea unui sistem de gestiune a stocului care să se actualizeze automat pentru produsele care permit acest lucru, iar în cazul în care se produc erori la inserarea de date va fi foarte ușor de corectat aceste erori.
* Foaia de comandă, documentul dat poate fi digitalizat printr-un sistem de gestiune a comenzilor. Acest lucru va permite înregistrarea comenzilor în sistem și vizualizarea comenzilor active. Cu ajutorul acestui sistem comenzile se vor putea monitoriza mult mai ușor.
* Nota de plată, o direcție de perfecționare pentru acest neajuns ar fi tot cu un sistem de gestionare al comenzilor care va genere o notă de plată în funcție de comandă.
* Plata de către client se efectuează doar la casa de marcat, o direcție de perfecționare a acestui neajuns ar fi posibilitatea clienților de a accesa o platformă, în care sa poate crea o comandă fiind prezent în local și la finalizarea acesteia să existe posibilitatea de a achita online cu cardul prin platforma respectivă.
* Comenzile nu sunt înregistrate în nici un sistem, acest neajuns poate fi îmbunătățit tot cu un sistem de gestionare al comenzilor.

# Cap. 2. Proiectarea de detaliu a aplicației informatice

## 2.1. Definirea obiectivelor aplicației informatice

Aplicațiile informatice în prezent sunt prezente aproape în fiecare tip de afaceri, implementarea unei aplicații informatice conduce în cele mai multe cazuri către automatizarea unor procese mulțumită cărora se economisesc bani și timp ceea ce duce la creșterea profiturilor.

Aplicația informatică are ca obiective principale automatizarea și digitalizarea proceselor din sistemul informațional prezentat în capitolul 1. Automatizarea și digitalizarea acestor procese vor permite economisirea timpului de munca și a minimizării erorilor umane.

Aplicația informatică va implementa următoare funcționalități:

* Sistem de gestionare al comenzilor.
* Sistem de gestionare a rezervărilor.
* Panou administrativ cu posibilitatea de generare a rapoartelor.

## 2.2. Proiectarea logică și fizica a ieșirilor

### 2.2.1 Lista cu toate documentele de ieșire din aplicație

|  |  |  |
| --- | --- | --- |
| Nr. crt. | Denumire raport | Descriere raport |
| 1 | Raport produse stoc | Raportul va fi generat din panoul administrativ pentru a se putea vizualiza datele despre fiecare produs și numărul de bucăți disponibil în stoc. |
| 2 | Raport produse pentru aprovizionare | Raportul va putea fi generat din panou administrativ, care va conține toate produsele ce necesita aprovizionare cât mai curând posibil din cauza faptului că stocul acestor produse este sub o limită stabilită |
| 3 | Raport cu tips-ul pentru fiecare angajat | Raportul cu tips-ul pentru fiecare angajat va fi generat din panoul administrativ pentru a se vizualiza suma de tips pe care un angajat a câștigat-o într-o anumită perioadă de timp |
| 4 | Raport cu comenzile pe o anumită perioadă | Raportul cu comenzi se va genera din panoul administrativ în cazul în care va fi necesar vizualizarea tuturor comenzilor pe o anumită perioadă de timp |
| 5 | Raport cu încasări din comenzi pe perioade de timp | Acest raport se generează la închiderea casei pentru a se vizualiza suma comenzilor din ziua respectivă, sau din panoul administrativ pentru a vizualiza veniturile pe o anumită perioadă de timp |
| 6 | Raport cu datele clienților înregistrați | Un raport generat în panoul administrativ cu datele clienților |
| 7 | Raport cu toate rezervările | Raportul se generează din panoul administrativ cu toate rezervările pe o anumită perioadă de timp |
| 8 | Bonul fiscal | Documentul de ieșire generat la finalizarea unei comenzi |
| 9 | Foaia de comandă | Documentul de ieșire care va putea fi generat la alegerea angajatului în cazul în care va trebui sa transmită datele comenzii unui alt angajat |
| 10 | Nota de plată | Documentul de ieșire generat de către angajat la cererea clientului pentru a putea efectua plata pentru comanda plasată |

*Tabelul nr. 1, Lista cu toate documentele de ieșire*

### 2.2.2 Machetele tuturor documentelor de ieșire

* Raport produse stoc

A white sheet with black text

Description automatically generated

*Figura 12, machetă raport stoc*

* Raport produse pentru aprovizionare

A paper with a number of lines

Description automatically generated with medium confidence

*Figura 13, Machetă raport produse aprovizionare*

* Raport tips pentru fiecare angajat

A white sheet with black text

Description automatically generated

*Figura 14, Machetă raport tips per angajat*

* Raport comenzi

A white sheet with black text

Description automatically generated

*Figura 15, Machetă raport comenzi*

* Raport încasări

A white card with black text

Description automatically generated

*Figura 16, Machetă raport venituri*

* Raport datele clienților

A white sheet with black text

Description automatically generated

*Figura 17, Machetă raport utilizatori*

* Raport rezervări

A white sheet with black text

Description automatically generated

*Figura 18, Machetă raport rezervări*

* Bonul fiscal

A screenshot of a receipt

Description automatically generated

*Figura 19, Machetă bon fiscal*

* Foaia de comandă

A screen shot of a document

Description automatically generated

*Figura 20, Machetă foaia de comandă*

* Nota de plată

A screenshot of a document

Description automatically generated

*Figura 21, Machetă nota de plată*

## 2.3. Proiectarea logică și fizica a intrărilor

### 2.3.1 Lista cu toate intrările din aplicație

*Tabelul nr. 2, Lista cu toate situațiile de intrare*

|  |  |  |
| --- | --- | --- |
| Nr. Crt. | Denumire situație intrare | Descriere situație intrare |
| 1 | Autentificarea în aplicație | Video-formatul în care se vor introduce datele user-ului pentru a putea accesa funcționalitățile aplicației |
| 2 | Înregistrarea unui nou client | Video formatul prin care se introduc datele necesare înregistrării unui nou client în aplicație |
| 3 | Înregistrarea unui nou angajat | Video formatul prin care se introduc datele necesare înregistrării unui nou angajat în aplicație |
| 4 | Creare unei rezervări de către client | Video formatul pentru introducerea datelor necesare creării unei noi rezervări de către client |
| 5 | Crearea unei rezervări de către angajat | Video formatul pentru crearea unei noi rezervări de către angajat |
| 6 | Adăugarea unui produs nou | Video formatul pentru adăugarea unui produs nou în sistem din panoul administrativ |
| 7 | Adăugarea unui produs complex | Video formatul pentru adăugarea unui produs complex în sistem din panou administrativ |
| 8 | Echilibrarea de stoc | Video formatul pentru echilibrarea stocului de produse |
| 9 | Adăugarea unei mese | Video formatul pentru adăugarea datelor necesare pentru a înregistra o masă nouă în sistem |
| 10 | Adăugarea unei noi categorii | Video formatul pentru adăugarea unei noi categorii în sistem |

### 2.3.2 Machetele pentru toate video formatele din aplicația informatica

**A screenshot of a login screen

Description automatically generated**

*Figura 22, Machetă autentificare*

**A screenshot of a registration form

Description automatically generated**

*Figura 23, Machetă creare rezervare client*

**A screenshot of a login form

Description automatically generated**

*Figura 24, Machetă înregistrare utilizator*

**A screenshot of a phone registration form

Description automatically generated**

*Figura 25, Machetă creare rezervare angajat*

**A screenshot of a computer

Description automatically generated**

*Figura 26, Machetă creare utilizator angajat*

**A screenshot of a computer

Description automatically generated**

*Figura 27, Machetă adăugare produs*

**A screenshot of a computer

Description automatically generated**

*Figura 28, Machetă adăugare produs nou complex*

**A screenshot of a computer

Description automatically generated**

*Figura 29, Machetă adăugare masă nouă*

**A screenshot of a computer

Description automatically generated**

*Figura 30, Machetă adăugare categorie nouă*

**A screenshot of a computer

Description automatically generated**

*Figura 31, Machetă creare echilibrare stoc*

## 2.4. Proiectarea sistemului de codificare a datelor

Aplicația informatic folosește un sistem de codificare bazat pe valoare cheii primare pentru identificarea unică a unei înregistrări din baza de date, aceasta fiind aplicată pentru fiecare tabel din baza de date.

Un alt sistem de codificare de tip numeric folosit în aplicația informatică este utilizat pentru stabilirea statusului unei comenzi, aceste coduri sunt definite astfel:

*Tabelul nr. 3, Sistemul de codificare pentru comenzi*

|  |  |
| --- | --- |
| Cod | Semnificația |
| 1 | Comandă creată |
| 2 | Comandă acceptată |
| 3 | Comandă predată |
| 4 | Comandă finalizată |
| 5 | Comandă anulată |

Sistemul de crearea și gestionarea a rezervărilor include un sistem de codificare de tip boolean pentru stabilirea statusului unei rezervări, aceste coduri sunt definite astfel:

*Tabelul nr. 4, Sistemul de codificare pentru rezervări*

|  |  |
| --- | --- |
| Cod | Semnificație |
| True | Rezervare confirmată |
| False | Verificare rezervare |

Un alt sistem de codificare este prezent pentru stabilirea statusului unei mese din local, acest sistem de codificare este de tip boolean fiind definit astfel;

*Tabelul nr. 5, Sistemul de codificare pentru mese*

|  |  |
| --- | --- |
| Cod | Semnificație |
| True | Masă ocupată |
| False | Masă liberă |

## 2.5. Proiectarea bazei de date

Prin aplicația informatică se gestionează mai multe parți al sistemului informațional analizat în capitolul 1, din aceasta cauze este necesar proiectarea unei baze de date în care se vor stoca toate datele necesare pentru funcționarea corectă a tuturor modulelor. Fiecare tabel din baza de date conține doar atributele necesare pentru îndeplinirea funcționalităților stabilite cu clientul.

Baza de date conține următoarele tabele:

* Tabela comenzi

*Tabelul nr. 6, Comenzi*

|  |  |
| --- | --- |
| Denumire coloană | Tip de data |
| Cod\_comandă | Număr întreg |
| Dată\_comandă | Dată |
| Status\_comandă | Număr întreg |
| Cod\_client | Număr întreg |
| Cod\_angajat | Număr întreg |
| Cod\_masă | Număr întreg |
| Tips | Număr real |

Tabela nr. 6 este definită pentru stocarea comenzilor în sistem, tabela conține atributul codul comenzii care reprezintă codul unic al fiecărei comenzi, data la care s-a creat comanda, statusul comenzii, tips-ul pentru angajat, codul clientului care a făcut comanda și codul unic al angajatului care a preluat comanda și este responsabilul pe ea.

* Tabela produse

*Tabelul nr. 7, Produse*

|  |  |
| --- | --- |
| Denumire coloană | Tip de data |
| Cod\_produs | Număr întreg |
| Nume\_produs | Șir de caractere |
| Preț\_unitar | Număr real |
| Unitate\_măsurare | Șir de caractere |
| Valabil\_consumator | Boolean |
| Produs\_complex | Boolean |
| Cod\_categorie | Număr întreg |
| Cantitate\_disponibilă | Număr întreg |
| Cantitate\_aprovizionare | Număr întreg |

Tabela nr. 7 conține toate datele necesare pentru produsele din stoc, aici se regăsesc atributele pentru numele produsului, prețul unitar al produsului, unitatea de măsurare, valabil consumator este un atribut prin care se face verificarea pentru a ști daca produsul se poate vinde direct consumatorului sau nu este un produs dedicat vânzării directe, aceste produse care nu sunt dedicate vânzării directe deseori sunt componente ale produselor complexe.

* Tabela produse\_comanda

*Tabelul nr. 8, Produse - comandă*

|  |  |
| --- | --- |
| Denumire coloană | Tip de data |
| Cod\_produs\_comandă | Număr întreg |
| Cod\_comandă | Număr întreg |
| Cod\_produs | Număr întreg |
| Cantitate comandată | Număr întreg |
| Preț unitar | Număr real |

Tabelul nr. 8 este un tabel de legătura pentru tabela nr. 7 și tabela nr. 6 pentru a se identifica toate produsele care aparțin unei comenzi, această tabelă conține 2 câmpuri care sunt prețul unitar și cantitatea produselor comandate, s-a adăugat prețul unitar și în această tabel , deși se regăsește și în tabela nr. 7 deoarece în cazul în care se va schimba prețul unor produse sa nu se afecteze comenzile vechi din sistem sau dacă se va dori de vizualizat evoluția prețului a unui produs.

* Tabela categorii

*Tabelul nr. 9, Categorii*

|  |  |
| --- | --- |
| Denumire coloană | Tip de date |
| Cod\_categorie | Număr întreg |
| Nume\_categorie | Șir de caractere |

Tabela nr. 9 este pentru adăugarea sau modificarea categoriilor pentru

produsele din stoc, s-a adăugat aceasta tabelă pentru gestionarea mai ușoara a categoriilor.

* Tabela rezervări

*Tabelul nr. 10, Rezervări*

|  |  |
| --- | --- |
| Denumire coloană | Tip de date |
| Cod\_rezervare | Număr întreg |
| Dată\_rezervare | Dată |
| Număr oaspeți | Număr întreg |
| Status\_rezervare | Boolean |
| Ore\_rezervate | Număr întreg |
| Cod\_client | Număr întreg |
| Cod\_masă | Număr întreg |
| Prenume | Șir de caractere |
| Nume | Șir de caractere |
| Număr\_telefon | Șir de caractere |

Tabela nr. 10 este pentru gestionarea rezervărilor din aplicația informatică, această tabelă conține toate datele minime necesare pentru crearea și vizualizarea rezervărilor create de către clienți sau angajați utilizând aplicația informatică, atributele prenume, nume, și număr telefon pot fi adăugate doar de către utilizatorii cu rolul de angajați.

* Tabela organizație

*Tabelul nr. 11, Organizații*

|  |  |
| --- | --- |
| Denumire coloană | Tip de date |
| Cod\_organizație | Număr întreg |
| Nume | Șir de caractere |
| Adresă | Șir de caractere |
| Adresă\_logo | Șir de caractere |
| Status\_funcționare | Boolean |

Tabela nr. 11 face parte din baza de date pentru a stoca informațiile firmei și pentru a se putea modifica statusul localului, în închis sau deschis, pentru a se putea efectua comenzi în aplicația informatică doar în cazul în care localul este deschis.

* Tabela clienți

*Tabelul nr. 12, Clienți*

|  |  |
| --- | --- |
| Denumire coloană | Tip de data |
| Cod\_client | Număr întreg |
| Nume | Șir de caractere |
| Prenume | Șir de caractere |
| Email | Șir de caractere |
| Cod\_user | Șir de caractere |
| Număr\_telefon | Șir de caractere |

Tabela nr. 12 este dedicată pentru stocarea datelor clienților necesare pentru conectarea în aplicația informatică, atributul cod user este pentru stocarea codului care face legătura cu o tabela din baza de date generata pentru funcționalitățile de autentificare și autorizare.

* Tabela angajat

*Tabelul nr. 13, Angajați*

|  |  |
| --- | --- |
| Denumire coloană | Tip de data |
| Cod\_angajat | Număr întreg |
| Nume | Șir de caractere |
| Prenume | Șir de caractere |
| Email | Șir de caractere |
| Salariu\_fix\_brut | Număr real |
| Cod\_user | Șir de caractere |

Tabela nr. 13 conține datele angajaților, tabela dată tot conține codul pentru user care este necesar pentru a face referința la baza de date care efectuează autentificarea și autorizarea în aplicația informatică. S-a mers pe principiul datelor minime necesare.

* Tabela componente\_produs\_complex

*Tabelul nr. 14, CPC*

|  |  |
| --- | --- |
| Denumire coloană | Tip de data |
| Cod\_cpc | Număr întreg |
| Cod\_produs\_complex | Număr întreg |
| Cod\_produs\_component | Număr întreg |

Tabelul nr. 14 este un tabel de legătura, doar că face legătura dintre mai multe produse, pentru identificarea produselor componente pentru un produs complex, de exemplu la cafenea sunt multe produse te tip cocktail care au mai multe ingrediente, în aceste ingrediente pot sa intre și sticle de 330ml de coca cola sau alte tipuri de sucuri necesare pentru prepararea băuturii și sa realizat această tabelă pentru a se putea scădea din stoc automat produsele care fac parte dintr-un produs complex și care permite scăderea automată din stoc.

* Tabela balansare\_stoc

*Tabelul nr. 15, Balansare stoc*

|  |  |
| --- | --- |
| Denumire coloană | Tip de date |
| Cod\_balansare\_stoc | Număr întreg |
| Dată\_balansare | Data |
| Cod\_produs | Număr întreg |
| Cantitate\_scăzută | Număr întreg |
| Categorie\_balansare | Număr întreg |

Tabelul nr. 15 este pentru stocarea informației necesare balansărilor de stoc, deoarece într-o cafenea sunt foarte multe produse care nu pot fi scăzute automat din stoc, s-a realizat această tabelă prin care se face scăderea manuală a cantității produselor selectate din diferite motive, motivul scăderii fiind reprezentat de o categorie de balansare.

* Tabela categorii\_balansare

*Tabelul nr. 16, Categorii balansare*

|  |  |
| --- | --- |
| Denumire coloană | Tip de dată |
| Cod\_categorie\_balansare | Număr întreg |
| Nume\_categorie | Șir de caractere |

Tabelul nr. 16 este pentru stocarea categoriilor de balansare pentru a ușura crearea unei balansări de stoc în care trebuie introdusa categorie de balansare.

* Tabela mese

*Tabelul nr. 17, Mese*

|  |  |
| --- | --- |
| Denumire coloană | Tip de dată |
| Cod\_masă | Număr întreg |
| Capacitate\_masă | Număr întreg |
| Status\_masă | Boolean |

Tabelul nr. 17 este pentru stocarea informațiilor despre mesele localului, și pentru vizualizarea statusului mesei, care poate fi ocupată sau liberă.

## 2.5.1 Schema relațională a bazei de date

A screenshot of a computer screen

Description automatically generated

*Figura 32, Schema relațională a bazei de date***A black background with white ovals

Description automatically generated**

## 2.6. Schema de sistem a aplicației

A computer screen shot of a diagram

Description automatically generated

*Figura 33, Schema de sistem a aplicației*

## 2.7. Proiectarea interfeței aplicației

Proiectarea interfeței aplicației informatice a fost gândită pentru a putea fi compatibilă și cu dispozitivele mobile deoarece că va fi folosită în mare parte de către clienți și de către angajați pe dispozitive mobile.

**Screen-uri cu interfețele din aplicație**

## 2.8 Alegerea tehnologiei de prelucrare

În urma studiului sistemului informațional și a cerințelor pentru aplicația informatica, s-a decis utilizarea următoarelor tehnologii pentru dezvoltarea aplicații informatice:

Pentru realizarea bazei de date sa ales SGBD[[1]](#footnote-1)-ul SQL Server. SQL Server este

dezvoltat de compania Microsoft și este folosit pentru stocarea și gestionarea datelor, folosind limbajul SQL[[2]](#footnote-2) pentru manipularea și interogarea de date. Principalul motiv pentru alegerea acestui SGBD este din cauza că pentru realizarea parții de backend[[3]](#footnote-3) se folosește ASP.NET Core WEB API[[4]](#footnote-4). SQL Server este foarte performat și scalabil, permițând gestionarea volumelor mari de date, deci dacă pe viitor se va dori extinderea aplicației cu noi funcționalități, baza de date nu va fi o problema. SQL Server nu este gratis pentru utilizarea în producție, în cazul dat se folosește versiunea Express care este gratuită și se utilizează pentru scopurile de dezvoltare.

Aplicația este dezvoltată cu AST.NET Core Web API. Tehnologia .Net este dezvoltată de compania Microsoft, este o tehnologie gratuită care utilizează în mare parte limbajul de programare C#. Limbajul C# este un limbaj de programare OOP[[5]](#footnote-5) și se alfa în top 5 la cele mai utilizate limbaje de programare după datele platformei GitHub. În proiect se folosește Entity Framework care ușurează mult interacțiunea din cod cu baza de date. Prin Entity Framework toate interacțiunea cu baza de date pot să fie realizate cu ajutorul unor metode predefinite, minimizând necesitatea de a scrie cod SQL.

Dezvoltarea de frontend[[6]](#footnote-6) a aplicației este realizată în Angular. Angular este framework web, care se folosește pentru crearea aplicațiilor web de tip o singură pagină. Angular folosește limbajul typescript dezvoltat de compania Google. Angular permite crearea aplicațiilor web compatibile pe toate dispozitivele. Această tehnologie a fost aleasă datorită faptului că se bazează pe module și componente, ceea ce înseamnă ca atunci când se trece de la o pagină la alta, nu se încarcă o pagină web nouă, ci doar se încarcă un alt component pe pagina principala care face ca utilizarea aplicației sa fie mult mai plăcută pentru consumatorii acesteia.

## 2.9 Estimarea necesarului de resurse și a calendarului de realizare

Estimarea necesarului de resurse și a calendarului de realizare este realizat cu ideea în care sunt disponibile toate tehnologiile necesare ca și o echipă completă pentru dezvoltarea aplicației informatice deși în proiectul dat aplicația informatică este realizată de o singură persoană.

### 2.9.1 Activitățile și necesarul de resurse

1. **Studiul sistemului existent**

|  |  |  |
| --- | --- | --- |
| Activitate | Personal alocat | Perioada de timp alocata |
| Analiza fluxului informațional | Analist de sistem | 7 zile |
| Determinarea documentelor necesare | Analist de sistem | 4 zile |

*Tabelul nr. 18, resursele pentru studiul sistemului existent*

1. **Proiectarea de detaliu**

|  |  |  |
| --- | --- | --- |
| Activitate | Personal alocat | Perioada de timp alocata |
| Proiectarea intrărilor | Analist de sistem  Designer aplicație | 7 zile |
| Proiectarea ieșirilor | Analist de sistem  Designer aplicație | 7 zile |
| Proiectarea bazei de date |  | 10 zile |
| Proiectarea sistemului de codificare | Analist de sistem | 2 zile |
| Proiectarea interfețelor | Designer aplicație | 30 zile |
| Realizarea planului de achiziție software și hardware | Analist de sistem  Manager pe proiect | 5 zile |

*Tabelul nr. 19, resursele pentru proiectarea de detaliu*

1. **Programarea aplicației**

|  |  |  |
| --- | --- | --- |
| Activitate | Personal alocat | Perioada de timp alocata |
| Programare backend | Programator 1  Programator 2 | 45 zile |
| Programare frontend | Programator 3  Programator 1 | 45 zile |
| Testare module | Tester 1  Tester 2 | 20 zile |
| Dezvoltarea documentației | Programator 1  Programator 2  Programator 3 | 9 zile |

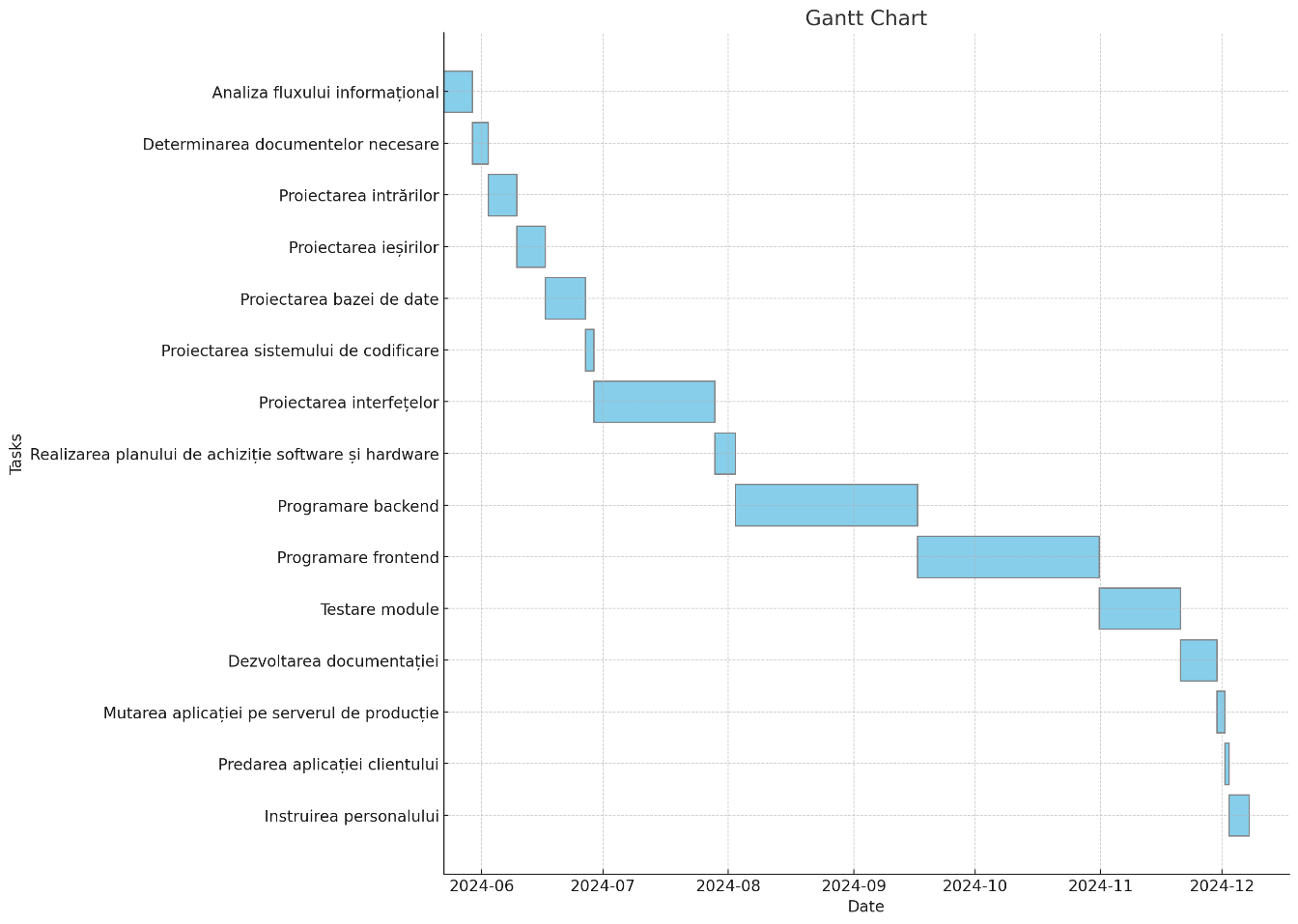
*Tabelul nr. 20, resursele pentru programarea aplicației*

1. Implementarea aplicației în sistem

|  |  |  |
| --- | --- | --- |
| Activitate | Personal alocat | Perioada de timp alocata |
| Mutarea aplicației pe serverul de producție | Inginer baze de date  Programator 1 | 2 zile |
| Predarea aplicației clientului | Manager pe proiect  Analist de sistem | 1 zile |
| Instruirea personalului | Tester 1  Programator 1 | 5 zile |

*Tabelul nr. 21, resursele pentru implementarea aplicației în sistem*

### 2.9.2. Diagrama Gantt



*Figura 34, Diagrama Gantt*

În figura 34 sunt reprezentate toate activitățile pentru finalizarea proiectului și durata de timp pentru îndeplinirea fiecărei activități. Durata totală de realizarea a aplicației informatice este de 200 de zile.

# Cap. 3. Prezentarea produsului software

## 3.1 Cerințele platformei hardware și software ale produsului

Aplicația informatică va fi de tip web, pentru a putea lansa aplicația în mediul public vom avea nevoie de un server pentru hosting în care vom ține atât API-ul cât și aplicația Angular.

Decizia de a rula ambele proiecte pe același server a fost luată din motivul că aplicația este dezvoltată pentru un singur client, clientul fiind o cafenea nu va avea un trafic foarte mare de utilizatori, deoarece aplicația va fi folosita constant doar de personal, iar clienții o vor folosi doar când sunt prezenți în localul dat. Un alt motiv pentru folosirea unui singur server pentru hosting este o latență mai mică pentru afișarea informației necesare în interfața aplicației, deoarece toate cererile din aplicația angular sunt trimise către API pe același server. Totuși decizia de a dezvolta aplicația cu o separare între backend și frontend a fost luată pentru cazul în care numărul utilizatorilor va crește se va putea rula aplicația pe servere de hosting separate pentru backend și frontend ceea ce va crește scalabilitatea aplicației, acest lucru fiind limitat pentru un singur server pentru hosting.

Pentru o performanță optimă a funcționalității aplicației cerințele hardware pentru server-ul de hosting vor fi:

* 8 procesoare virtuale
* 16 GB ram
* 200 GB SSD pentru stocarea datelor necesare
* Conexiune la internet, preferat de 10Gbps

Cerințele software pentru rularea aplicației informatice sunt următoarele:

* Un sistem de operare instalat pe server, linux server
* Vom avea nevoie de Nginx[[7]](#footnote-7) pentru a putea rula aplicația Angular
* .Net Runtime pentru partea de backend
* SQL Server instalat pe server, pentru baza de date

## 3.2 Descrierea funcțiunilor aplicației

Aplicația informatică este împărțita în 3 parți, acestea fiind funcționalitățile pentru utilizatorii de tip client, funcționalitățile pentru utilizatorii de tip angajat și funcționalitățile pentru administrator. Pe lângă acestea mai avem funcționalitățile pentru autentificarea și autorizarea utilizatorilor și funcționalitățile pentru interacțiunea cu baza de date.

Pentru partea de autentificare și autorizarea a utilizatorilor se folosește Identity din ASP.NET Core. Identity este un sistem de gestionarea a funcționalităților de autentificare și autorizare din cadrul unei aplicații. La implementarea acestui sistem în aplicația se generează o baza de date în SQL Server prin metoda Code-First[[8]](#footnote-8) care folosește migrarea claselor din aplicație în tabele relaționate în baza de date. Prin acest sistem putem gestiona datele utilizatorilor, să setăm rolurile acestora, implementarea sistemul de confirmare prin email, autentificarea în aplicația prin metode externe cum ar fi prin google, facebook, microsoft și multe altele. Baza de date generata de către Identity a fost separată de baza de date a aplicației din motive tehnice pentru a evita conflictele posibile.

Pentru autorizarea utilizatorilor se folosește jwt token care reprezintă un cod unic pentru fiecare utilizator din care se extrag informațiile necesare pentru realizarea autorizării, cum ar fi rolul utilizatorului. Acest cod unic este generat la logarea utilizatorului în sistem.

Acest token este necesar pentru accesul la anumite funcționalități a aplicației, fără el utilizatorii nu vor avea acces la funcționalitățile care necesită autorizare.

Principalele funcționalități pentru utilizatorii cu rolul de client sunt următoarele:

* Creare unei comenzi noi

Pentru crearea unei comenzi în aplicația informatică, clientul va trebui să efectueze autentificarea în aplicație, să adauge în coș produsele pe care le dorește, după pentru a putea crea comanda va fi necesar sa scaneze codul qr de pe masă pentru a se putea identifica numărul mesei la care sa efectuat comanda și pentru a preveni situațiile în care utilizatorii încearcă sa efectueze o comanda falsă.

* Crearea unei rezervări

Clientul are acces și la funcționalitatea de creare a unei rezervări, odată creată rezervare trebuie confirmată de către un angajat pentru a putea fi validă.

Funcționalitățile principale pentru utilizatorii cu rolul de angajat:

* Modulul POS pentru casa de marcat.

O interfață grafică care permite crearea și modificarea comenzilor plasate de către clienții care nu doresc utilizarea aplicației informatice.

* Confirmarea rezervărilor și a comenzilor.

Sunt două funcționalități separate pentru confirmarea comenzilor plasate de către clienți în aplicația informatica, odată confirmată responsabilitatea pe îndeplinirea comenzii este a angajatului care a confirmat-o. Iar confirmarea rezervărilor este efectuată de oricare din personalul locației, în cazul în care o rezervare nu poate fi confirmată din diferite motive, un angajat trebuie sa contacteze persoana care a creat rezervare și în funcție de alegerea acestuia să o șteargă sau să modifice data rezervării.

* Funcționalitatea balansării de stoc.

Într-un local de tip cafenea, deseori sunt cazuri de risipă a unor produse. De exemplu spargerea unei sticle de alcool sau accidente cu comanda unui client pană ca aceasta sa fie livrata, din aceasta cauza s-a introdus aceasta funcționalitate pentru a balansa stocul în aceste cazuri.

* Funcționalitatea pentru crearea rezervărilor.

S-a introdus și funcționalitatea pentru crearea rezervărilor în local de către angajați pentru a crea rezervări pe perioade mai îndelungate și pentru clienții care nu utilizează aplicația informatică.

Funcționalitățile principale pentru administrator:

* Adăugarea și modificarea datelor din baza de date.

Efectuarea operațiilor de tip CRUD [[9]](#footnote-9) pe tabele care permit acest lucru.

* Generarea rapoartelor.
* Vizualizarea grafurilor pentru evoluția vânzărilor zilnice.
* Blocarea accesului utilizatorilor la aplicație.

Funcționalitățile de mai sus sunt doar cele principale, aceste funcționalități mai au alte parți minore.

# Cap. 4. Eficiența și utilitatea aplicației informatice

## 4.1 Condiții privind implementarea aplicației

## 4.2 Exploatarea curenta a aplicației

## 4.3 Considerații privind eficiența aplicației informatice

# 5. Anexe

# 6. Bibliografie

**Cărți:**

Aristeidis , B., & Pablo, D. (2023). *Learning Angular: A no-nonsense guide to building web applications with Angular 15.* Packt Publishing.

Botezatu, M., Botezatu, C., & Căruțașu, G. (2016). *Managementul proiectelor informatice.* Editura Pro Universitaria.

Crișan, D. A. (2015). *Programarea aplicatiilor folosind limbajul C# și platforma .NET.* Editura Pro Universitaria.

Ionel, I., & Botezatu, C. (2007). *Proiectarea sistemelor informatice.* București: Pro Universitaria.

Jamie, K., & Brian, W. (2014). *ASP.NET Web API 2: Building a REST Service from Start to Finish.* Apress.

Price, M. J. (2023). *C# 12 and .NET 8 - Modern Cross-Platform Development Fundamentals.* Packt Publishing.

Schuster, S. a. (2023). *Building Web APIs with ASP.NET Core.*

**Documentație:**

*ASP.NET Core web API documentation*. (2024, 04 25). Preluat de pe Microsoft: <https://learn.microsoft.com/en-us/aspnet/core/tutorials/web-api-help-pages-using-swagger?view=aspnetcore-8.0>

*Entity Framework documentation hub*. (2024). Preluat de pe Microsoft.com: https://learn.microsoft.com/en-us/ef/

*Introduction to the Angular docs*. (2023). Preluat de pe Angular: https://angular.io/docs

*Microsoft identity platform documentation*. (fără an). Preluat de pe Microsoft.com: https://learn.microsoft.com/en-us/entra/identity-platform/

*SQL Server technical documentation*. (fără an). Preluat de pe Microsoft.com: https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16

*npm Docs*. (fără an). Preluat de pe npmjs.com: https://docs.npmjs.com/

1. SGBD – sistem de gestionare a bazelor de date [↑](#footnote-ref-1)
2. SQL – limbaj de programare pentru interacțiunea cu baza de date. [↑](#footnote-ref-2)
3. Backend – constă în infrastructura aplicației, partea unde se face logica pentru funcționarea tuturor modulelor. [↑](#footnote-ref-3)
4. API – Application Programming Interface, reprezintă un set de definiții de sub-programe, protocoale și unelte pentru programarea de aplicații și software. [↑](#footnote-ref-4)
5. OOP – object oriented proggraming, limbaj de programare baza pe clase și obiecte. [↑](#footnote-ref-5)
6. Frontend – constă în partea vizuală a aplicației informatice, aici se realizează implementarea în aplicație a funcționalităților. [↑](#footnote-ref-6)
7. Nginx – un server web dezvoltat pentru o performanta maxima și stabila pentru gestionarea unui număr mare de conexiuni simultane. [↑](#footnote-ref-7)
8. Code-First – este o metodă de creare și modificare a unei baze de date prin care mai întâi se scriu clasele în aplicație și după aceste clase sunt migrate în baza de date sub forma de tabele relaționate. [↑](#footnote-ref-8)
9. CRUD – creat, read, update, delete pentru interacțiunea cu baza de date. [↑](#footnote-ref-9)