Aplicație Gestiune Restaurant

Bancoș Gabriel, Ciurte Iulian, Gavriș Nicolae, Haidu Eduard, Mirth Kevin, Pop Lucas

Contents

1 Introducere

Proiectul nostru de gestionare a unui restaurant reprezintă o soluție software modernă ce are ca obiectiv administrarea digitalizată a tuturor aspectelor esențiale ale unui restaurant, precum gestionarea meniului, barului, chat-ului intern pentru angajați și a altor procese conexe. Soluția propusă are rolul de a ușura munca utilizatorului, oferind funcționalități specifice care optimizează procesele zilnice și reduc erorile umane.

Această documentație detaliază etapele parcurse pentru dezvoltarea sistemului de gestiune și a fost realizată de o echipă formată din Bancoș Gabriel, Ciurte Iulian, Gavriș Nicolae, Haidu Eduard, Mirth Kevin și Pop Lucas, ca parte a cursului de Ingineria Programelor. Am ales acest proiect deoarece domeniul ospitalității este în continuă dezvoltare, iar soluțiile software pentru gestiune sunt din ce în ce mai solicitate în diverse afaceri, precum restaurante, baruri sau cafenele.

Ne concentrăm pe dezvoltarea backend-ului sistemului, asigurând funcționalități robuste și securizate. Programul propus trebuie să fie ușor de utilizat, să ofere eficiență ridicată și să fie scalabil, astfel încât să poată fi implementat cu succes în mai multe locații.

1.1 Scopul proiectului

Scopul principal al acestui proiect este de a dezvolta o soluție software completă pentru administrarea unui restaurant, având următoarele funcționalități principale: gestionarea bucătăriei și a barului; funcționalitățile de chat intern pentru angajați; administrarea meniului și a rețetarului; generarea de rapoarte detaliate privind stocurile și necesarul de aprovizionare; scalabilitate pentru implementări multiple; siguranță prin prevenirea erorilor umane și protecția datelor.

1.2 Avantajele soluției Software

Programul destinat gestiunii unui restaurant trebuie să ofere următoarele avantaje:

- Eficiență ridicată – automatizarea celor mai importante procese;
- Siguranță – prevenirea furturilor și a erorilor de calcul;
- Rapoarte precise – furnizarea unei imagini clare asupra stocurilor și necesarului;
- Scalabilitate – sistem flexibil, capabil să se adapteze pentru mai multe locații;
- Ușurința de utilizare – interfață intuitivă și accesibilă.

1.3 Personalizare și dezvoltări ulterioare

Software-ul dezvoltat va reprezenta o soluție personalizată pentru nevoile specifice ale clientului. În etapa de proiectare, se va lua în considerare posibilitatea dezvoltarilor ulterioare, astfel încat funcționalități noi să poată fi adăugate în viitor, conform cerințelor beneficiarului.

Prin acest proiect, echipa noastră își propune să ofere o soluție completă, sigură și eficientă, contribuind la digitalizarea și optimizarea proceselor specifice administrării unui restaurant.

2 Specificații

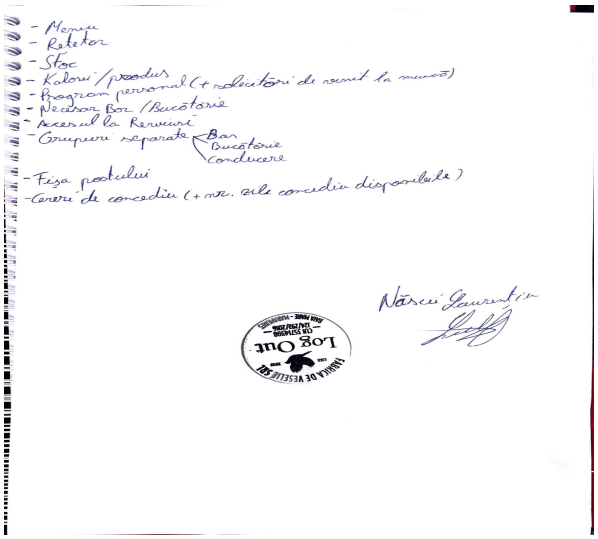
2.1 Identificarea nevoilor beneficiarului

În prima fază se identifică nevoile beneficiarului. Întrebările care trebuie puse sunt următoarele:

- Cine va folosi programul?
- Cum va fi folosit?
- Ce date trebuie introduse?
- Ce date trebuie extrase?

Răspunsurile la aceste întrebări definesc funcționalitățile pe care trebuie să le îndeplinească sistemul. Putem defini o imagine de ansamblu a ceea ce va îndeplini sistemul fară să ne intereseze cum va funcționa în spate.

2.2 Cerințe funcționale



2.2.1 Discuția cu beneficiarul

În urma discuției cu beneficiarul, a fost identificată o serie de operațiuni pe care acesta le desfășoară zilnic în munca sa, în ceea ce privește gestiunea restaurantului. Totodată, luând în considerare ca produsul final realizat pe structura aplicației dezvoltate va necesita accesul mai multor tipuri de utilizatori, aplicația va include și o parte de gestiune a acestora.

2.2.2 Operațiunile identificate

Operațiunile de bază care trebuie incluse în sistem sunt urmatoarele:

- Inserare/ștergere/actualizare de produse.
- Inserare/ștergere/actualizare de utilizatori.

Acestea sunt strict funcționalitățile obligatorii pe care trebuie să le îndeplinească aplicația. Lista detaliată a cerințelor funcționale poate fi vazută în SRS.

2.3 Analiza MoSCoW

Prioritizarea cerințelor permite echipei care lucrează la proiect să înțeleagă cantitatea de efort și resursele necesare pentru realizarea lucrării, îmbunătățindu-se astfel gestionarea timpului și crescând probabilitatea ca proiectul să fie finalizat la timp.Aplicând metoda MoSCoW asupra proiectului în cauză, se ajunge la urmatoarele rezultate

Must-have

- Meniu - este o listă digitală care prezintă preparatele disponibile în restaurant
- Rețetar - o listă detaliată a ingredientelor și a cantităților necesare pentru prepararea fiecărui produs din meniu
- Stock - listă cu stock-ul disponibil
- Management personal

- Necesar bar/bucătărie - ingredientele out of stock care trebuiesc cumpărate
- Acces diferențiat pe grupuri de utilizatori -

Should-have

- Macronutrienți per produs - listă cu macronutrienții dupa fiecare produs
- Rapoarte de stock
- Scalabilitate

Could-have

- Sincronizare în timp real al datelor
- Interfața intuitivă și intrucțiuni clare
- Securitate avansată

Won't-have

- Monitorizare în timp real al performanței angajaților

3 Analiză cerințe

3.1 Cazuri de utilizare

Sistemul de gestiune a unui restaurant răspunde la cerințele funcționale prin intermediul mai multor cazuri de utilizare, fiecare fiind asociat unui actor specific. Actorii principali identificați sunt Managerul, Bucătarul și Barmanul. Aceștia interacționează cu sistemul pentru a realiza diferite activități legate de administrarea meniului, gestionarea personalului și monitorizarea resurselor necesare.

Angajat

Post	Barman
Nume:	Andrei
Sex:	Masculin
Vârsta:	23
Stare civilă:	Necăsătorit
Ocupație:	Fotbal și sală
Personalitate:	Extrovertit, amabil cu clienții, bine dispus
Domiciliu:	Baia Mare
Hobby-uri:	Ieșit în oraș cu prietenii și sală
Aspirații:	Vrea să-și îmbunătățească abilitățile sociale
Așteptări:	Un venit sigur

Angajat

Post:	Bucătar
Nume:	Monica
Sex:	Feminin
Vârsta:	19
Stare civilă:	Necăsătorită
Ocupație:	Manichiuristă
Personalitate:	Extrovertită, amabilă cu clienții, bine dispusă
Domiciliu:	Baia Sprie
Hobby-uri:	Ieșit în oraș cu fetele și mani pedi
Aspirații:	Vrea să-și îmbunătățească abilitățile sociale
Așteptări:	Bani de buzunar

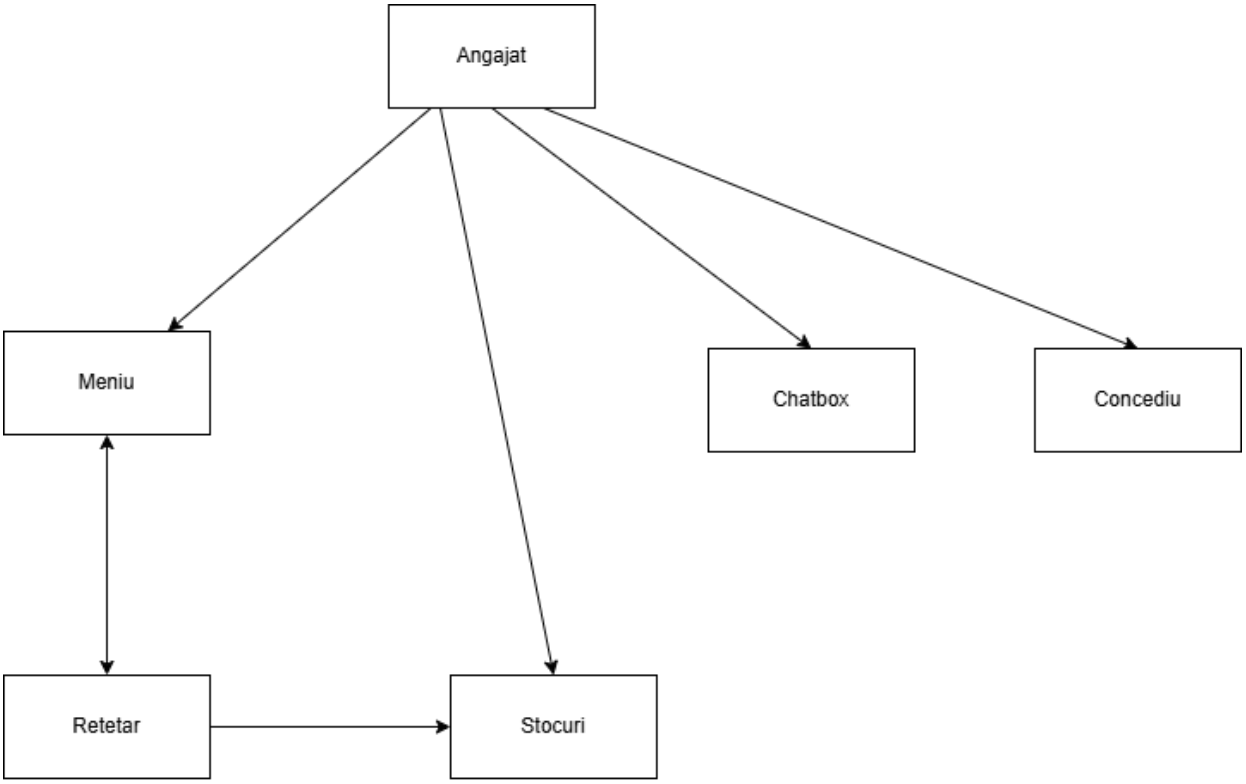
Administrator

Post:	Director
Nume:	Călin
Sex:	Masculin
Vârsta:	48
Obiectiv:	Antreprenor
Stare civilă:	Căsătorit
Personalitate:	Extrovertit, amabil cu angajații, bine dispus
Domiciliu:	Baia Mare
Hobby-uri:	Ieșit în oraș cu prietenii și sală
Aspirații:	Vrea să-și îmbunătățească abilitățile sociale
Așteptări:	Un venit sigur

3.2 Arhitectura generală

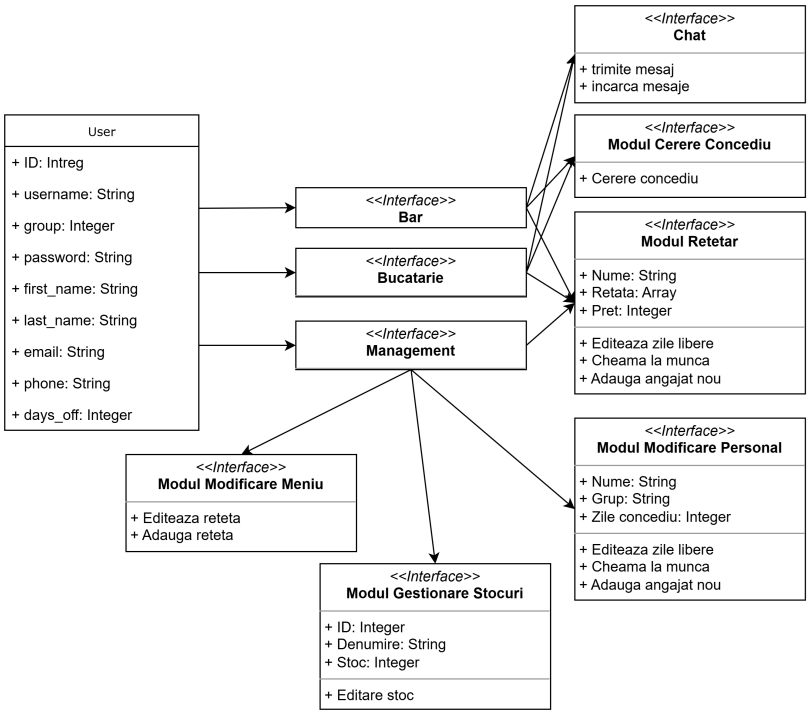
Schema prezentată mai jos reprezintă o arhitectură pentru un sistem de gestionare într-un restaurant. Modulul central **Angajat** este conectat cu restul componentelor, facilitând gestionarea și comunicarea informațiilor.

- **Meniu:** Permite gestionarea și actualizarea meniurilor.
- **Rețetar:** Gestionează rețetele și detaliile despre ingrediente.
- **Stocuri:** Monitorizează și acutalizează stocurile disponibile.
- **Chatbox:** Asigură comunicarea între angajați.
- **Concediu:** Administrează cererile de concediu ale angajaților.



3.3 Arhitectura detaliată-Diagrama de clase

1. **Clasa Principală (User)**
   - **ID** - Identificator unic al utilizatorului.
   - **username** - Numele de utilizator.
   - **group** - Grupul din care face parte utilizatorul.
   - **password** - Parola utilizatorului.
   - **first_name** - Prenumele utilizatorului.
   - **last_name** - Prenumele utilizatorului.
   - **email** - Adresa de email.
   - **phone** - Număr de telefon.
   - **days_off** - Numărul de zile libere disponibile.
2. **Interfețe Funcționale**
   (a) **Chat**
      - **trimite mesaj** - Permite trimitere de mesaje.
      - **încarcă mesaje** - Încarcă mesajele din sistem.



(b) **Modul Cerere Concediu**

- **Cerere concediu** - Funcționalitățile pentru depunerea unei cereri de concediu

(c) **Modul Rețetar**

- **Nume** - Numele rețetei.
- **Rețeta** - Lista de ingrediente sau pași din rețetă.
- **Preț** - Prețul ingredientelor.

Funcționalități suplimetare:

- **Editează zile libere**
- **Cheamă la muncă**
- **Adaugă angajat nou**

(d) **Modul Modificare Personal**

- **Nume** - Numele angajatului.
- **Grup** - Grupul din care face parte.
- **Zile concediu** - Zilele de concediu disponibile.

Funcționalități specifice:

- **Editează zile libere**
- **Cheamă la muncă**
- **Adaugă angajat nou**

(e) **Modul Gestionare Stocuri**

- **ID** - Identificator unic pentru produs.
- **Denumire** - Numele produsului.
- **Stoc** - Cantitatea disponibilă în stoc.
- **Editare stoc** - Permite actualizarea cantității stocului.

3. **Interfețe Generale**

- **Bar**
- **Bucătărie**
- **Management**

4 State Of The Art

Pentru început vom începe cu o scurtă descriere despre ce înseamă "State Of The Art".

State Of The Art reprezintă nivelul actual al dezvoltării într-un anumit domeniu sau tehnologie, în cazul nostru "Gestiunea unui restaurant".

1. **Digitalizarea și automatizarea proceselor în restaurante**

   Conform cu [4] autorul prezintă câteva aspecte pentru digitalizarea și automatizarea proceselor în restaurante. Unul dintre aspectele esențiale extrase din articolul prezentat este importanța tehnologiilor moderne pentru eficientizaarea operatiunilor intr-un restaurant. Conform autorilor, digitalizarea și automatizarea proceselor pot imbunatati semnificativ gestionarea resurselor și reducerea erorilor. De exemplu evidenta stocurilor in timp real.

   Un al punct important este optimizarea resurselor umane. Automatizarea proceselor de programare și gestionare a concediilor ajută la eficientizarea muncii personalului. Aceste tehnologii permit astfel, o gestionare mai eficientă a resurselor umane și o mai buna coordonare.

2. **Sisteme informatice pentru managementul resurselor într-un restaurant**

   Conform cu [1] autorul subliniază implementarea unui sistem ERP integrat într-un restaurant poate adresa o gamă larga de nevoi, inclusiv gestionarea meniurilor și a resurselor umane. Un sistem ERP bine implementat poate centraliza meniurile, inclusiv informațiile despre ingrediente, calorii, etc., astfel încat toate echipele implicate (bucătari, manageri, barmani) să aibă acces la date actualizate în timp real.

   Un alt beneficiu important al sistemelor ERP este capacitatea de a automatizaa procesele de resurse umane, cum ar fi gestionarea cererilor de concediu și aporbarile. Astfel, sistemul poate monitoriza zilele de concediu ale angajaților. Aceste funcționalități permit reducerea timpului necesar pentru gestionarea manuală a acestor procese.

3. **Sisteme front-end pentru managementul meniurilor**

   În referința [3] este prezentată importanța sistemelor front-end pentru managementul meniurilor. Designul digital al meniurilor în industria ospitalității influentează semnificativ satisfacția clientilor. În special, mai multe aspecte ale meniurilor, precum descrierea detaliată a produselor, diversitatea opțiunilor și design-ul meniului(digital sau tradițional), joacă un rol important în satisfacția consumatorilor. De asemenea, se subliniează faptul că meniurile digitale, fie cu touchscren, fie non-touchscrenn, pot contribui la economisirea timpului și reducerea erorilor angajaților, ceea ce adaugă valoare experienței clientului.

4. **Sisteme de gestionare a personalului și automatizarea proceselor HR**

   Conform cu [7] autorul face referire la inovarea proceselor de HR cu tehnologii digitale, lucrearea explorând modul în care cunoștințele externe pot fi utilizate în procesele de inovare a resurselor umane. În mare, principalele puncte sunt **Impactul tehologiilor digitale în domeniul HR** unde tehonologiile digitale permit automatizarea și eficientizarea proceselor in HR, de asemenea și reducerea costurilor și bineinteles creșterea productivității restaurantului.**Inovarea in HR**, aceasta implică folosirea cunoștințelor interne dar și cele externe, acestea combinându-se astfel pentru a crea soluții noi. Așadar în acest articol ne este prezentată importanța implementării unui cadru strategic pentru a digitaliza partea de HR.

   Conform cu [8] autorul acestei lucrări face referire la automatizarea proceselor HR, beneficiile automatizării și impactul acesteia asupra efiienței organizaționale. Autorul începe cu definirea automatizării HR unde aceasta ne este reprezentată precum o unealtă, adică este definită ca înlocuirea operațiunilor manuale. Scopul acesteia este de a reduce sarcinile repetitive și care consumă timp. Beneficiile automatizării proceselor HR sunt creșterea productivitătii cu ajutorul eliminării redundanțelor și a lucrurilor ineficiente și de asemenea creșterea transparenței și a controlului.

5. **Arhitecturi sofware moderne pentru aplicații restaurant**

   Conform cu [5] unde este vorba despre necesitatea arhitecturii software în dezvoltarea modernă și identificarea punctelor de focus pentru a crea o arhitectură pentru o aplicație. Contextul

aplicației este tipul acesteia fiind un produs care folosește anumite tehnologii de analiză modernă a datelor, a cazurile de utilizare care implică monitorizarea utilizării unui restaurant. Ne este prezentată și importanța designului arhitecturii aplicației care se concentrează pe scalabilitate și adaptibilitate.

6. **Securitatea datelor în aplicațiile de gestionare**

   Conform cu [2] Cihan Cobanoglu a interogat mai multe restaurante printr-un sondaj cu privire la măsurile de securitate folosite și la cele mai frecvente atacuri cibernetice. În urma sondajului, acesta a observat că doar 24% din restaurantele interogate își verifică vulnerabilitățile și majoritatea atacurilor pe care le suferă nu sunt raportate autorităților, temându-se de publicitatea negativă. Cobanoglu a constatat că cele mai frecvente tipuri de atacuri sunt viruși, abuzuri interne, furturi de dispozitive, spoofing și accesul neautorizat, iar aceste atacuri vin adesea de la hackeri independenți și angajați nemulțumiți. Iar cele mai utilizate mijloace de protecție sunt programele antivirus, firewall hardware și diferite măsuri fizice de securitate. În urma acestor observații, Cobanoglu recomandă ca pentru îmbunătățirea securității să se folosească instrumente biometrice (precum identificarea prin amprentă sau recunoaștere facială), personalul să fie instruit regulat cu privire la securitatea cibernetică și să se implementeze politici stricte pentru acces la sistem și angajații să folosească parole complexe.

7. **Impactul implementării sistemelor de gestiune asupra restaurantelor**

   Conform cu [9] unde autorii fac referință la impactul implementării sistemelor de gestiune asupra restaurantelor și importanța acestora. Acest studiu se adresează cercetătorilor din domeniu tehnologic și managementului, celor din industria resturantelor. Lucrarea oferă o analiză a modului in care sistemele de gestionare a restaurantelor(RMS) sunt aplicate și bineînțeles cum acestea influențează industria serviciilor alimentare. Ce sunt RMS? Sunt niște sisteme software care au fost concepute pentru optimizarea și gestionarea operațiunilor din restaurante. Acestea includ raportarea financiară, gestionarea stocurilor și programarea personalului. Beneficile sunt creșterea eficienței, reducerea costurilor, digitalizarea și gestionarea operațiunilor. Impactul tehnologic asupra restaurantelor din cele prezentate, este unul important deoarece tehnologia este un lucru esențial pentru competivitate și adaptare la diferite condiții, de asemena contribuie la productivitate și profitabilitate.

8. **Utilizarea cloud computing pentru gestionarea datelor**

   Conform cu [10], autorii lucrării povestesc despre tehonologia cloud, un model care ne permite furnizarea de informații prin intermediul internetului. În loc ca utilizatorii să întrețină echipamentele hardware sau să folosească aplicații software, se pot închiria aceste resurse de la furnizorii de cloud, reducând totodată costurile de operare. Un aspect foarte important care este menționat în lucrare, este capacitatea furnizării de date în funcție de cerințele variate de calitate a serviciului, incluzând, latența, performanța si fiabilitatea. În această lucrare sunt analizate platformele cloud de top(Amazon Web Services, Microsofft Azure și Google App Engine), performanța acestora, diponibilitatea și fiabilitatea, cauzele din spatele erorilor.

9. **Optimizarea arhitecturii software prin utilizarea bazelor de date**

   Conform cu [6], autorii oferă o analiză sistematică privind arhitecturile software și performanțele bazelor de date SQL și NoSQL. Obiectivele principale ale acestui studiu sunt: examinarea arhitecturilor software ale bazelor de date SQL și NoSQL în contextul procesării volumelor mari de date; compararea performanțelor între bazele de date SQL și NoSQL cu ajutorul analizei unor anumite aspecte precum scalabilitatea, disponibilitatea, consistența și capacitatea de distribuire a datelor. În mare, bazele de date SQL sunt mai potrivite pentru aplicații de procesare a tranzacțiilor datorită structurii rigide. Bazele de date NoSQL sunt folosite în analiza datelor mari, acest fapt se datorează scalabilității și flexibilității în gestionarea datelor. Autorii recomandă dezvoltarea de API-uri unice și alegera tipului de bază de date în funcție de preferințele utilizatorului. Așadar, acest studiu oferă o perspectivă asupra limitelor și avantajelor bazelor de date SQL și NoSQL și alegerii arhitecturii potrivite în funcție de cerințele utilizatorului.

5 Implementare

Aplicația este dezvoltată folosind limbajul PHP și framework-ul folosit pentru front-end este Bootstrap 5. PHP este un limbaj de programare utilizat pentru dezvoltarea aplicațiilor web, fiind capabil să gestioneze logica de server și interacțiunea cu bazele de date, iar Bootstrap este un framework CSS care facilitează dezvoltarea rapidă a interfețelor grafice, prin clase predefinite pentru elemente HTML, cum ar fi butoane, tabele, formulare și sidebar.

5.1 Structura aplicației

Aplicația noastră utilizeaza o abordare simplă, fară un model arhitectural precum MVC. În acest caz avem:

- PHP se ocupă de gestionarea logicii aplicației și interacțiunea cu datele. Codul PHP este integrat direct în fișierele HTML, fiind folosit pentru procesarea inputului primit de la utilizatori, generarea conținutului și comunicarea cu baza de date. Spre exemplu, datele trimise printr-un formular sunt validate și prelucrate într-un fisier PHP.
- Bootstrap este responsabil pentru partea vizuală a aplicației. Bootstrap oferă o serie de clase și componente gata de utilizat, permițând astfel crearea unor interfețe moderne și estetice, fără a fi necesară o dezvoltare CSS de la zero. Prin utilizarea claselor Bootstrap, se pot defini usor dimensiunile, culorile, alinierile sau stilurile diferitelor componente ale interfeței, cum ar fi:
  - **Butoane**
  - **Formulare**
  - **Tabele**
  - **Grid System** pentru organizarea layout-ului aplicației

5.2 Interacțiunea utilizatorului

Utilizatorul interacționează cu aplicația prin intermediul formularelor sau a butoanelor definite cu ajutorul Boostrap. Aceste elemente sunt plasate într-o pagina HTML, iar datele sunt trimise către server folosind metodele GET sau POST. Fișierele PHP preiau datele primite, le validează și realizează diverse acțiuni, precum utlizarea bazei de date sau generarea unor răspunsuri pe pagina web.

Rezultatul procesării datelor este trimis înapoi către utilizator sub forma de conținut HTML, cu elemente Bootstrap prentru o prezentare atractivă. Astfel, prin combinarea PHP cu Bootstrap, aplicația oferă atât o logică funcțională, cât și o interfață intuitivă pentru utilizatori.

5.3 PHP Doc

Doxygen este un generator de documentație pentru limbajul PHP pentru a genera documentație în format HTML din codul sursă. În continuare sunt prezentate capturi de ecran ale documentației codului sursă.

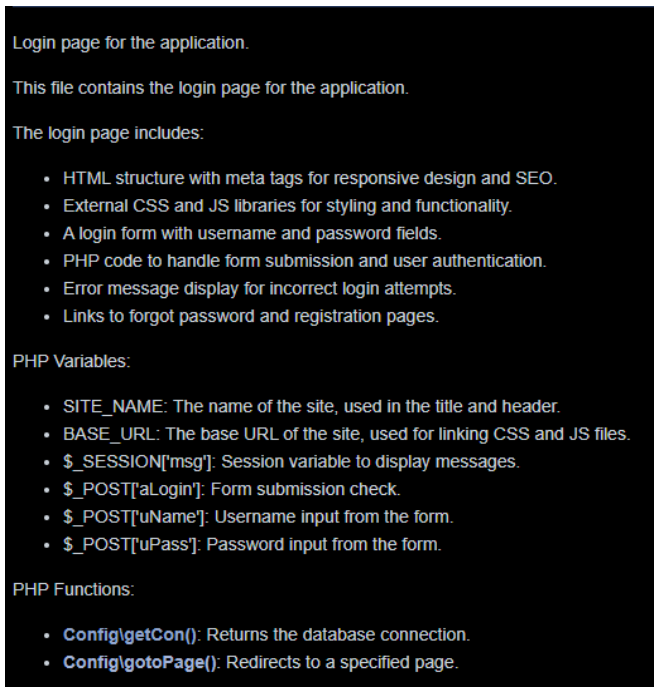
5.4 Tehnologii Folosite

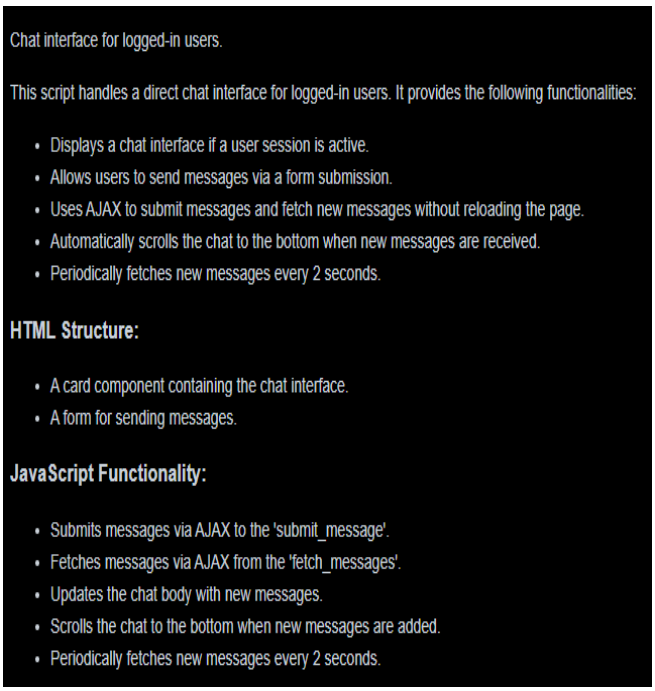
Bootstrap Pentru dezvoltarea interfeței grafice pe care o va vedea utilizatorul folosim framework-ul numit Bootstrap, care ne permite o dezvoltare mai rapidă a interfeței. Acesta ne pune la dispoziție elemente precum: container, row, button, input, dropdown oferindu-ne o cale mai usoară pentru dezvoltarea interfeței.

5.5 Conținutul aplicației web

Aplicația noastră conține următoarele pagini esențiale precum: profil, chat box, meniu, angajați, dar ulterior pot fi adăugate și alte pagini.

Interfața paginii web este simplă și intuitivă, iar aceastea se prezintă astfel:





5.5.1 Sidebar-ul

Oferă o navigare rapidă între diferitele secțiuni ale aplicației și conține un meniu vertical care include:

- Direct chat
- Employees
- Menu

5.5.2 Header

Este situat în partea superioară a paginii și include:

- Logo-ul aplicației
- Profilul utilizatorului cu poza și nume

5.5.3 Conținutul fiecărei pagini

- Pagina "Direct Chat" prezintă un chat box, text box și butonul "Send".
- Pagina "Employees" prezintă lista cu angajații, iar în dreptul fiecăruia este funcția acestuia și zilele de concediu rămase.Totodată din această pagină la apăsarea pe numele oricărui angajat vom fi redirecționați către profilul acestuia.
- Pagina "Menu" prezintă produsele din meniu, împărțite pe mai multe pagini.

6 Testare și validare

Testarea și validare aplicațiilor sunt activități efectuate frecvent de către fiecare tester software. Scopul acestora este de a ne asigura că aplicația funcționează corect conform cerințelor utilizatorului.

Testarea se referă la procesul de verificare a funcționalității și performanței unei aplicații prin identificarea erorilor și defectelor și a comportamentelor neprevăzute, prin rularea unor seturi de teste.

Validarea, pe de alta parte, este procesul prin care se confirmă dacă aplicația îndeplinește cerințele utilizatorului.

6.1 Unit Test Login

Unit test-ul prezentat este scris în PHP, unde se creează două mock-uri (dummy) și utilizăm testing framework-ul PEST pentru a executa testele. Scopul principal al testelor este de a verifica integritatea funcților utilizate precum și a query-urilor aflate în controllerul de login.

```php
<?php

use App\Controller\LoginController;
use Mockery;
use PHPUnit\Framework\TestCase;

beforeEach(function () {
    $this->mockPDO = Mockery::mock(PDO::class);
    $this->mockStatement = Mockery::mock(PDOStatement::class);
    $this->controller = new LoginController($this->mockPDO);
});

afterEach(function () {
    Mockery::close();
});

it('can verify passwords correctly', function () {
    $password = 'password123';
    $userPassword = 'passw2ord123';

    expect($this->controller->verifyPassword($password, $userPassword))->toBeFalse();
});

it('returns false for wrong password in performLogin', function () {
    $this->mockPDO->shouldReceive('prepare')
        ->once()
        ->with('SELECT * FROM `users` WHERE `username` = :username')
        ->andReturn($this->mockStatement);

    $this->mockStatement->shouldReceive('bindParam')
        ->once()
        ->with(':username', 'testuser');

    $this->mockStatement->shouldReceive('execute')->once();

    $this->mockStatement->shouldReceive('rowCount')->once()->andReturn(0);
```

```php
    expect($this->controller->performLogin('testuser', 'wrongpassword'))->toBeFalse();
    expect($_SESSION['msg'])->toContain('Wrong password or username!');
});

it('returns true for correct password in performLogin', function () {
    $this->mockPDO->shouldReceive('prepare')
        ->once()
        ->with('SELECT * FROM `users` WHERE `username` = :username')
        ->andReturn($this->mockStatement);

    $this->mockStatement->shouldReceive('bindParam')
        ->once()
        ->with(':username', 'admin');

    $this->mockStatement->shouldReceive('execute')->once();

    $this->mockStatement->shouldReceive('rowCount')->once()->andReturn(1);
    $user = (object) ['id' => 1, 'password' => 'admin'];
    $this->mockStatement->shouldReceive('fetch')->once()->andReturn($user);

    expect($this->controller->performLogin('admin', 'admin'))->toBeTrue();
    expect($_SESSION['user'])->toBe(1);
});

it('handles PDO exception gracefully', function () {
    $this->mockPDO->shouldReceive('prepare')->once()->andThrow(new PDOException("Database error"));
    expect($this->controller->performLogin('test', 'parola'))->toBeFalse();
    expect($_SESSION['msg'])->toContain('An error occurred. Please try again later.');
});
```

6.2 Unit Test ChatBox

Unit test-ul prezentat este scris in PHP, unde se creeaza un mock si utilizam testing framework-ul PHPUnit pentrui a executa testele.Scopul principal al testelor este de a verifica integritatea funcților utilizate.

```php
You, 3 weeks ago | 1 author (You)
<?php

use PHPUnit\Framework\TestCase;

Run tests
class SubmitMessageTest extends TestCase
{
    private $config;
    private $pdo;
    private $stmt;

    protected function setUp(): void
    {
        // Create a mock for the Config class
        $this->config = $this->createMock(Config::class);

        // Create a mock for the PDO class
        $this->pdo = $this->createMock(PDO::class);

        // Create a mock for the PDOStatement class
        $this->stmt = $this->createMock(PDOStatement::class);

        // Configure the Config mock to return the PDO mock
        $this->config->method('getCon')->willReturn($this->pdo);

        // Configure the PDO mock to return the PDOStatement mock
        $this->pdo->method('prepare')->willReturn($this->stmt);
    }

    Run test
    public function testSubmit()
    {
        $_SERVER['REQUEST_METHOD'] = 'POST';
        $_POST['sender'] = 'test_sender';
        $_POST['message'] = 'test_message';

        // Call the submit method
        $this->submit();

        $this->assertEquals('test_sender', $_POST['sender']);
        $this->assertEquals('test_message', $_POST['message']);
    }

    public function submit()
    {
        if ($_SERVER['REQUEST_METHOD'] === 'POST') {
            $sender = $_POST['sender'] ?? 'test_sender';
            $message = $_POST['message'] ?? 'test_message';

            if (!empty($message)) {
                $stmt = $this->pdo->prepare('INSERT INTO messages (sender, data, message) VALUES (?, ?, ?)');
                $stmt->execute([$sender, $this->config->getData(), $message]);
                $_POST['message'] = 'test_message';
```

```php
    }

    Run test
    public function testSubmit()
    {
        $_SERVER['REQUEST_METHOD'] = 'POST';
        $_POST['sender'] = 'test_sender';
        $_POST['message'] = 'test_message';

        // Call the submit method
        $this->submit();

        $this->assertEquals('test_sender', $_POST['sender']);
        $this->assertEquals('test_message', $_POST['message']);
    }

    public function submit()
    {
        if ($_SERVER['REQUEST_METHOD'] === 'POST') {
            $sender = $_POST['sender'] ?? 'test_sender';
            $message = $_POST['message'] ?? 'test_message';

            if (!empty($message)) {
                $stmt = $this->pdo->prepare('INSERT INTO messages (sender, data, message) VALUES (?, ?, ?)');
                $stmt->execute([$sender, $this->config->getData(), $message]);
                $_POST['message'] = 'test_message';
            }
        }
    }
}

class Config      You, 3 weeks ago • + submit message unit test
{
    public function getCon()
    {
        // Mocked PDO connection
        $pdo = $this->createMock(PDO::class);
        $stmt = $this->createMock(PDOStatement::class);

        $pdo->method('prepare')->willReturn($stmt);
        $stmt->method('execute')->willReturn(true);

        return $pdo;
    }

    public function getData()
    {
        return 'group_data';
    }
}
```

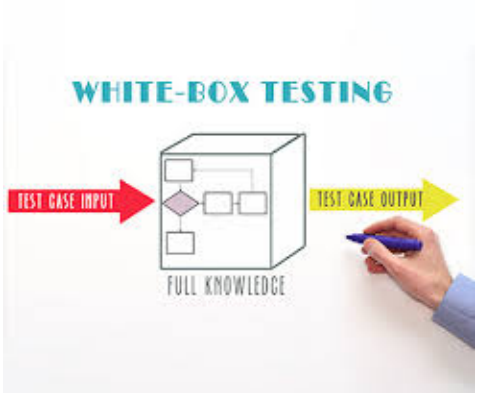
6.3 Testare Black Box

Definiție

Testerul care face testarea nu are cunoștințe despre structura internă sau modul de implemetare a codului sursă. Testerul va face testarea introducând date de intrare și va verifica ieșirile pentru a vedea dacă rezultatele corespund cerințelor.

6.4 Testare White Box

Definiție
Testerul care face testarea are acces complet la cod și cunoaște implemetarea internă a aplicației.



7 Rezultate

Mai jos este rezultaul unit test-ului de la Login care execută următoarele teste:
Primul test verifică dacă două parole sunt identice.
Al doilea test verifică dacă în momentul în care introduci username-ul sau parola gresită, testul să fie picat cu succes.
Al treilea test verifică dacă în momentul în care introduci usename-ul și parola corect, testul să fie trecut cu succes.
Al patrulea test simulează un mock PDO pentru baza de date care aruncă o excepție de tip PDOEXception, după care apelează metoda de performLogin cu credențiale false și se asteaptă să fie fals, totodată verifică dacă în mesajul salvat în variabila de sesiune **msg** corespunde cu cel de eroare din perfomrLogin.

```
PASS Tests\Unit\LoginControllerTest
✓ it can verify passwords correctly                                    0.02s
✓ it returns false for wrong password in performLogin                  0.01s
✓ it returns true for correct password in performLogin
✓ it handles PDO exception gracefully

Tests:    6 passed (19 assertions)
Duration: 0.26s
```

Mai jos este rezultatul unit test-ului de la ChatBox care execută urmatoarele teste:
Primul test verifică dacă se adaugă mesajul trimis de către utiliyator în baya de date.
Al doilea test verifică dacă mesajul scris este la fel cu cel adăugat în baza de date.

```
PHPUnit 11.5.0 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.4.1
Configuration: /Users/eduhaidu/restaurant-management/phpunit.xml

.                                                                   1 / 1 (100%)

Time: 00:00.015, Memory: 8.00 MB

OK (1 test, 2 assertions)
 *  Terminal will be reused by tasks, press any key to close it.
```

8 Concluzii

În concluzie acest ultim capitol prezintă rezultatele obținute și obiectivele care au fost atinse.

De asemenea sunt prezentate si niste dezvoltari ulterioare ale aplicatiei.

8.1 Realizări

Aplicația dezvoltată în cadrul proiectului îndeplinește întocmai nevoile propuse de către beneficiar.

În aplicația dezvoltată au fost implementate funcționaltățile de stoc, program personal, grupuri separate, cereri de concediu.

Așadar, cerințele principale au fost realizate în totalitate. Produsul final este o aplicație PHP, simplă dar totodata complexă din punct de vedere al informațiilor oferite către utiliaztorul aplicației. Am încercat să îi oferim aplicației o interfață cât mai intuitivă și mai ușor de folosit pentru mai multe tipuri de persona.

8.2 Dezvoltări ulterioare

Bineînteles că sistemul dezvoltat poate fi îmbunătățit prin adăugarea de noi funcționalități și prin optimizarea celor existente.De asemenea, viitoarele dezvoltări vor include creșterea performanței aplicației.

Mai jos este o listă cu funcționalitățile care ar putea fi adăugate în viitor sistemului: Buton de alarmă (la apăsarea acestuia în caz de urgență este sesizată echipa de pază), Notificare lipsă stock

9 Experiența de muncă în echipă

Munca în echipă este un lucru esențial în dezvoltarea de proiecte, oferind oportunitatea echipei de a-și împărți sarcinile în funcție de cunoștințele fiecărui membru. Totodată munca în echipă scoate la suprafață probleme apărute pe parcursul proiectului, dar de asemenea, ne ajută și la acumularea unor noi cunoștințe.

Echipa noastră a fost formată din 6 membri, fiecărui membru atribuindu-se un anumit task.

Proiectul a fost structurat pe mai multe etape având deadline-uri o data la două săptămâni.

Colaborarea și munca în echipă au fost de cele mai multe ori bune, totodata având și anumite provocări care țineau de structurarea aplicației. În mare parte nu au fost probleme, mai putin problema sincronizării timpului de întâlnire pentru a lucra.

În concluzie acest proiect ne-a ajutat să ne îmbunătățim munca în echipă, având în vedere că nu știam ce abilități are fiecare membru și totodata descoperirea unor noi domenii de lucru.

Ne-a ajutat să avem o privire de ansamblu diferită față de cea pe care o aveam înainte de a lucra la proiect.

În următorul tabel avem marcate pentru fiecare mebru al echipei, capitolele unde au fost aduse cele mai semnificative contribuții de către fiecare membru. Tabelul prezentat este menit să exemplifice activitatea fiecărui membru al echipei. Acest tabel este strict orientativ deoarce membrii echipei au contribuit la mai multe capitole având un nivel de implicare nu la fel de semnificativ.

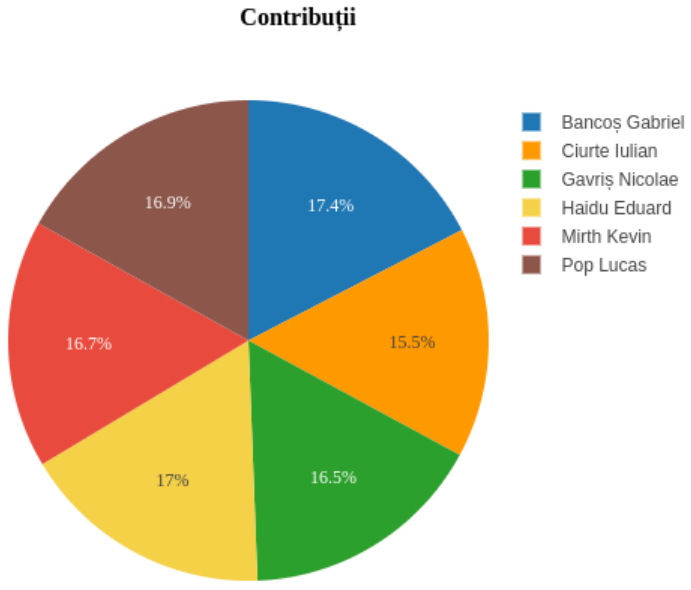
Tabel Contribuții Proiect

capitol\membru	Bancoș Gabriel	Ciurte Iulian	Gavriș Nicolae	Haidu Eduard	Mirth Kevin	Pop Lucas
Documentație					x	x
Prezentare			x	x		
Programare	x			x		x
Testare	x			x		
Diagrame		x	x	x	x	

Tabel Contribuții Documentație

În cea mai mare parte membrii echipei Mirth Kevin și Pop Lucas au făcut documentația,dar ceilalți membrii au contribuit și ei cu câteva idei pe parcurs.

capitol\membru	Bancoș Gabriel	Ciurte Iulian	Gavriș Nicolae	Haidu Eduard	Mirth Kevin	Pop Lucas
Introducere	x	x	x	x	x	x
Specificații	x		x		x	x
Analiză cerințe			x	x	x	x
State Of The Art	x	x	x	x	x	x
Implementare					x	x
Testare și validare	x			x	x	x
Rezultate	x			x	x	x
Concluzii		x	x		x	x



References

[1] Paula Serdeira Azevedo, Mário Romão, and Efigénio Rebelo. Advantages, limitations and solutions in the use of erp systems (enterprise resource planning) – a case study in the hospitality industry. *Procedia Technology*, 5:264–272, 2012. 4th Conference of ENTERprise Information Systems – aligning technology, organizations and people (CENTERIS 2012).

[2] Cihan Cobanoglu. A critical look at restaurant network security: attacks, prevention tools, and practices. *Journal of Foodservice Business Research*, 10(1):31–50, 2007.

[3] Cristina Cristina, Vivy Indrianti, Tri Wiyana, and Dendy Rosma. *The Impact of Digital Menu Design, Menu Item Description, and Menu Variety on Consumer Satisfaction in the Hospitality Industry, Especially in Restaurants*, pages 97–104. Springer Nature Switzerland, Cham, 2025.

[4] Fathi Mohamed Daradkeh, Thowayeb H. Hassan, Tatiana Palei, Mohamed Y. Helal, Sanaa Mabrouk, Mahmoud I. Saleh, Amany E. Salem, and Nabila N. Elshawarbi. Enhancing digital presence for maximizing customer value in fast-food restaurants. *Sustainability*, 15(7), 2023.

[5] Roni Jumpponen. Modern software architecture. 2021.

[6] Wisal Khan, Teerath Kumar, Cheng Zhang, Kislay Raj, Arunabha M Roy, and Bin Luo. Sql and nosql database software architecture performance analysis and assessments—a systematic literature review. *Big Data and Cognitive Computing*, 7(2):97, 2023.

[7] Päivi Laitala. Hr processes innovation with digital technologies: external knowledge as a source for new knowledge creation in innovation process. 2023.

[8] Shubham Shekhar Mishra, Monica Kunte, Netra Neelam, Sanjay Bhattacharya, and Preeti Mulay. Hr process automation: A bibliometric analysis. *Library Philosophy and Practice*, pages 1–12, 2021.

[9] Asiyatul Nabilah Rosnan, Nur Hidayah Che Ahmat, and Norfezah Md Nor. An overview of the application of restaurant management systems in the foodservice. *Journal of Tourism, Hospitality and Culinary Arts*, 15(2):191–204, 2023.

[10] Liang Zhao, Sherif Sakr, Anna Liu, and Athman Bouguettaya. *Cloud data management*. Springer, 2014.