UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

**Stai jos**

**RAPORT DE PROIECTARE**

* Perspectiva de dezvoltare -

**Echipa 19**

**Cărpineanu Alexandru**

**Cuciureanu Dragoș-Adrian**

**Vasile George Cristian**

2024

Cuprins

[1. Scopul aplicației 3](#_Toc183120380)

[2. Aria de acoperire a aplicației 3](#_Toc183120381)

[3. Preliminarii 3](#_Toc183120382)

[3.1 Specificații de analiză 3](#_Toc183120383)

[3.2 Specificații de proiectare 5](#_Toc183120384)

[3.3 Specificații tehnologice 6](#_Toc183120385)

[3.4 Componente de dezvoltare 7](#_Toc183120386)

[3.5 Componente de testare 9](#_Toc183120387)

[3.6 Componente de comunicare 9](#_Toc183120388)

[3.7 Componente de învățare 9](#_Toc183120389)

[4. Perspectiva de dezvoltare 10](#_Toc183120390)

# Scopul aplicației

*Stai jos* este un instrument destinat atât clienților, cât și proprietarilor de restaurante, având ca scop simplificarea și optimizarea procesului de rezervare. Tipul aplicației este unul de tip SaaS (Software as a Service), care permite utilizatorilor să își rezerve mese într-un mod rapid și eficient, fără apeluri telefonice sau vizite fizice. Misiunea principală a acestei aplicații este atragerea de clienți noi pentru restaurantele partenere, stimularea consumului prin oferte personalizate și facilități de rezervare, precum și fidelizarea clienților prin intermediul notificărilor, programelor de loialitate și al ofertelor exclusive. Beneficiarii principali ai aplicației sunt proprietarii și managerii de restaurante, care își pot optimiza resursele și pot reduce timpul de administrare, dar și clienții, care beneficiază de o experiență de rezervare ușoară și personalizată. Aplicația contribuie, astfel, la creșterea vizibilității restaurantelor și la îmbunătățirea experienței de consum, oferind clienților o metodă convenabilă de a-și planifica mesele.

# Aria de acoperire a aplicației

*Stai* jos, aplicația de gestionare a rezervărilor la restaurante are o arie de acoperire extinsă, fiind accesibilă atât la nivel local, cât și național, cu posibilitatea de a integra restaurante din diverse orașe sau regiuni. Aplicația este un instrument digital de organizare și planificare, orientat exclusiv către gestionarea rezervărilor și optimizarea interacțiunilor dintre clienți și restaurante. Nu este o platformă de livrare de mâncare și nici un ghid culinar complex, ci se concentrează pe facilitarea procesului de rezervare și pe creșterea gradului de ocupare a restaurantelor. În cadrul aplicației, există trei roluri principale: **adminul**, care gestionează setările generale, utilizatorii și restaurantele partenere; **stafful/personalul restaurantului**, care se ocupă de administrarea rezervărilor, confirmări și gestionarea capacității restaurantului; și **clientul**, care poate căuta restaurante, face rezervări și primi oferte personalizate. Obiectivele principale ale aplicației sunt creșterea accesibilității rezervărilor, reducerea timpului de așteptare prin gestionarea automată a locurilor disponibile și fidelizarea clienților prin oferirea unei experiențe de rezervare rapide și intuitive.

# Preliminarii

Cele mai relevante cerințe de calitate pentru realizarea misiunii aplicației sunt utilizabilitatea, disponibilitatea, securitatea și scalabilitatea, deoarece acestea contribuie în mod direct și predominant la creșterea numărului și valorii comenzilor. Utilizabilitatea asigură o experiență intuitivă, facilitând interacțiunea utilizatorilor cu aplicația, în timp ce disponibilitatea permite accesul permanent la funcționalitățile acesteia, indiferent de programul restaurantului. Securitatea protejează datele sensibile ale utilizatorilor, menținând încrederea acestora, iar scalabilitatea permite extinderea rapidă a capacităților aplicației pentru a face față creșterii cerințelor, asigurând relevanța și competitivitatea acesteia pe termen lung în ecosistemul restaurantelor.

# Specificații de analiză

Misiunea este punerea la dispoziție a unei platforme atractive care să mărească numărul de comenzi cât și valoarea acestora.

Ob1. Clienții doresc **să facă o rezervare** și vadă produsele din meniu de la restaurant pentru realizarea misiunii.

Ob1.1. Ca și client, vreau să pot accesa aplicația 24/7, astfel încât să pot face o rezervare chiar și în afara orelor de funcționare ale restaurantului pentru a mări numărul de comenzi. => cerința de calitate 2 (disponibilitate) sub formă de user story

Ob1.2. Ca și client, vreau să pot face o rezervare rapid la un anumit restaurant și la o anumită oră, astfel încât să îmi asigur un loc la restaurant în mai puțin de câteva minute pentru a mări numărul de comenzi. => cerința de calitate 1 (utilizabilitate) sub formă de user story

Ob1.3. Ca și client, vreau să primesc o notificare (de preferat pe email, dar inițial din aplicație), atunci când realizez o rezervare, pentru o experiență de utilizare bună.

Ob2. Personalul restaurantului se ocupă de **gestionarea meniurilor, rezervărilor și al comenzilor** pentru realizarea misiunii.

Ob2.1. Ca și client, vreau să pot vedea produsele din meniu în aplicație în mod rapid și intuitiv pentru a mări valoarea comenzii.

Ob2.2. Ca si personal al restaurantului, vreau să pot adăuga produsul ales de client rapid și ușor la comanda sa, pentru a crește eficiența procesului și astfel și interacțiunea clientului.

Ob2.3. Ca și client, vreau să pot plăti nota într-un mod facil din aplicație într-un mod securizat ce-mi va păstra confidențialitatea datelor, pentru a reveni în viitor și sa cresc numărul de comenzi. => cerința de calitate 3 (securitate) sub formă de user story

Ob3. Administratorii sistemului asigură buna funcționare și climatul de încredere al sistemului pentru realizarea misiunii.

Ob4. Sistemul furnizează în baza datelor caracteristici și **funcționalități avansate și la curent cu standardele industriei** pentru realizarea misiunii.

Ob4.1. Ca și dezvoltator, vreau ca aplicația să ofere funcționalitatea de a da feedback individual la fiecare produs cumpărat de către client.

Ob4.2. Ca și dezvoltator, vreau să pun la dispoziția clienților **elemente de gamification** pentru a le crește retenția.

Ob4.3.1. Ca și client, doresc să primesc rewarduri (sub forma de învârtirea unei roți) atunci când dau feedback la produse individuale, pentru a crește fidelitatea clientului și a mări numărul viitor de rezervări.

Ob4.3.2. Ca și client, doresc să primesc puncte atunci când realizez o rezervare la un anumit restaurant, pentru a crește fidelitatea clientului și a mări numărul viitor de rezervări.

Ob4.3. Ca și sistem, suport un volum mare de rezervări pentru a asigura funcționarea corectă și satisfăcătoare indiferent de numărul de rezervări introduse. => cerința de calitate 3 (scalabilitate) sub formă de user story

Ob4.4. Ca și dezvoltator, vreau să pot testa rapid și eficient noile funcționalități, astfel încât să asigur calitatea și stabilitatea aplicației.

Ob4.5 Ca și sistem, am nevoie de o bază de date bine definită pentru a stoca toate datele și informațiile.

# Specificații de proiectare

US1. Utilizabilitate: Ca și client, vreau să pot face o rezervare rapid la o anumită masă și la o anumită oră, astfel încât să îmi asigur un loc la restaurant în mai puțin de câteva minute pentru a mări numărul de comenzi.

US1.1. Interfețe intuitive: Ca și client, vreau să pot face o rezervare rapid la un anumit restaurant și la o anumită oră utilizând o interfață intuitivă organizată în pagini și formulare dedicate, astfel încât să îmi asigur un loc la restaurant în mai puțin de câteva minute pentru a mări numărul de comenzi.

US1.2. Step Wizard: Ca și client, vreau să fiu printr-un model de tip „step wizard” pentru a face o rezervare la un anumit restaurant și la o anumită oră astfel încât să îmi asigur un loc la restaurant în mai puțin de câteva minute pentru a mări numărul de comenzi.

US2. Disponibilitate: Ca și client, vreau să pot accesa aplicația 24/7, astfel încât să pot face o rezervare chiar și în afara orelor de funcționare ale restaurantului pentru a mări numărul de comenzi.

US2.1. Sisteme de monitorizare: Ca și administrator, vreau ca sistemul să fie monitorizat în timp real de sisteme robuste precum Grafana pentru orice eroare ce poate duce la downtime în cadrul aplicației pentru a maximiza timpul în care aplicația este disponibilă.

US2.2. Sisteme de monitorizare: Ca și administrator, vreau ca sistemul mă notifice imediat ce a fost monitorizată o eroare ce poate duce la downtime în cadrul aplicației pentru a maximiza timpul în care aplicația este disponibilă.

US3. Securitate: Ca și client, vreau să pot plăti nota într-un mod facil din aplicație într-un mod securizat ce-mi va păstra confidențialitatea datelor, pentru a putea reveni în viitor și să cresc numărul de comenzi.

US3.1. Secure payment provider: Ca și client, vreau să pot plăti nota într-un mod facil din aplicație prin intermediul unui secure payment provider (precum 3D Secure), ce-mi va păstra confidențialitatea datelor, pentru a putea reveni în viitor și să cresc numărul de comenzi.

US4. Scalabilitate: Ca și sistem am nevoie de abilitatea de a gestiona eficient și fără întreruperi orice volum de utilizatori pentru a asigura o experiență de utilizare de înaltă calitate.

US4.1 Ca și sistem, vreau implementarea unui sistem de caching pentru a reduce încărcarea serverelor și a accelera timpul de răspuns la solicitările frecvente.

US4.2 Ca și sistem, doresc implementarea unor soluții de monitorizare continuă a performanței și ajustarea periodică a resurselor și configurărilor pentru a menține scalabilitatea și performanța optimă.

# Specificații tehnologice

Tehnologia pentru Back End: .NET cu arhitectură MVC

Pentru back end-ul aplicației de management al restaurantelor, recomand utilizarea .NET Framework datorită flexibilității și performanței sale ridicate în dezvoltarea aplicațiilor web scalabile. .NET oferă o modularitate excelentă prin intermediul arhitecturii MVC, care facilitează separarea logicii aplicației de interfață și accesul la date, contribuind astfel la o întreținere mai ușoară a codului.

Framework-ul include suport nativ pentru capturarea și gestionarea erorilor, ceea ce simplifică implementarea tacticilor arhitecturale pentru monitorizare și disponibilitate optimă.

În ceea ce privește securitatea, ASP.NET Core Identity și integrarea cu furnizori terți (OAuth, 3D Secure) asigură protecția datelor utilizatorilor și tranzacțiilor efectuate. Respectarea principiilor SOLID și a arhitecturii Clean Architecture contribuie la crearea unui cod clar și extensibil, facilitând integrarea noilor funcționalități și creșterea continuă a aplicației.

Tehnologia pentru Front End: React

Pentru front end-ul aplicației, propun utilizarea React, un framework performant și flexibil, potrivit pentru dezvoltarea interfețelor de utilizator moderne și intuitive. React facilitează organizarea componentelor reutilizabile, ceea ce permite dezvoltarea unei structuri front end scalabile și eficiente.

Prin utilizarea React Router, aplicația poate fi organizată în pagini distincte pentru fiecare funcționalitate, oferind o experiență de utilizator fluidă și intuitivă. În plus, React permite integrarea tacticilor avansate de UX, cum ar fi modelele interactive sau fluxurile ghidate pentru utilizatori (de exemplu, modelul „step wizard”), îmbunătățind navigarea și utilizabilitatea aplicației.

# Componente de dezvoltare

C1. Baza de date

C1.1. Modelare

C1.2. Seedare

C2. User

C2.1 Client

C2.2 Staff

C2.3 Admin

C3. Restaurant

C4. Rezervare

C5. Produs

C6. Comandă

C7. Beneficiu

C7.1 Beneficiu bazat pe puncte

C7.2 Beneficiu bazat pe roata

F1. Fluxul de realizare de rezervări și comenzi

US1. Ca și client, vreau să pot accesa aplicația 24/7, astfel încât să pot face o rezervare chiar și în afara orelor de funcționare ale restaurantului pentru a mări numărul de comenzi. => componentele C1.1, C2.1, C3, C4

US2. Ca și client, vreau să pot face **o rezervare rapid la un anumit restaurant și la o anumită oră**, astfel încât să îmi asigur un loc la restaurant în mai puțin de câteva minute pentru a mări numărul de comenzi.

US2.1. Ca și client, vreau să pot vedea lista de restaurante existentă și selecta unul dintre acestea. => componentele C1.1, C2.1, C3, C4

US2.2. Ca și client, vreau să pot vedea și selecta o ora de la restaurantul ales pentru a putea face o rezervare.. => componentele C1.1, C2.1, C3, C4, C7.1

US2.3. Ca și client, vreau să primesc o notificare (de preferat pe email, dar inițial din aplicație), atunci când realizez o rezervare, pentru o experiență de utilizare bună. => componentele C1.1, C2.1, C4

US2.4. Ca și client, vreau să pot gestiona rezervările mele active. Prin gestionare ne referim la posibilitatea anulării/ editării acestora. => componentele C1.1, C2.1, C4

US3. Ca și client, vreau să pot vedea meniul restaurantului la care am rezervarea activă din pending, pentru a putea ulterior să fac o comandă în mod înștiințat. => componentele C1.1, C2.1, C5, C6

US4. Ca și client, trebuie să pot face o recenzie la produsele comandate după ce staff-ul adaugă comanda la rezervarea mea. => componentele C1.1, C2.1, C5, C6

F2. Fluxul de creare/updatare restaurante și comenzi

US1. Ca și staff de restaurant, doresc să pot să adaug restaurantul meu în aplicație pentru a le putea oferi clienților posibilitatea să facă rezervări în el. => componentele C1.1, C2.2, C3

US2. Ca și staff de restaurant, doresc să pot gestiona locurile libere din restaurant, pentru a putea oferi clienților posibilitatea să facă rezervări doar la ore și mese care sunt libere. => componentele C1.1, C2.2, C3, C4

US3. Ca și staff de restaurant, doresc să pot gestiona (CRUD) produse din meniul restaurantului care va fi valabil pentru client. => componentele C1.1, C2.2, C5

US4. Ca și staff de restaurant, doresc să asociez produsele comandate de client la rezervarea acestuia pentru a îi da posibilitatea să adauge review-uri pentru aceste produse. => componentele C1.1, C2.2, C4, C5

US5. Ca și staff de restaurant, doresc să pot gestiona rezervările restaurantului meu, pentru a putea să le închid când clienții pleacă de la locație. => componentele C1.1, C2.2, C4

F3. Fluxul de gamification

US1. Ca și client, doresc să pot primi puncte de fidelitate în momentul schimbării stării rezervărilor mele din active în închise. => componentele C1.1, C2.1, C4, C7.1

US2. Ca și client, doresc să pot accesa o lista de beneficii oferite de restaurante ca să pot cheltui punctele strânse prin efectuarea de rezervări pe aceste beneficii. => componentele C1.1, C2.1, C7.1

US3. Ca și client, vreau să-mi pot folosi punctele pentru a obține diverse beneficii oferite de restaurantele partenere. => componentele C1.1, C2.1, C7.1

US4. Ca și client, trebuie să am oportunitatea să particip la infrastructura de gamification implementată în aplicație, la finalul recenziei. Astfel voi putea primi beneficii care să mă facă să dau recenzii cât mai des, îndeplinind astfel misiunea aplicației. => componentele C1.1, C2.1, C7.2

US5. Ca și client, doresc să pot folosi beneficiile primite prin infrastructura de gamification. => componentele C1.1, C2.2, C7.1, C7.2

F4. Fluxul de admin

US1. Ca și admin, doresc să pot gestiona toate datele din aplicație pentru a asigura buna funcționare a acesteia. => toate componentele

F5. Fluxul tehnic

US1. Ca și dezvoltator, am nevoie de o bază de date bine modelată și normalizată pentru a asigura dezvoltarea coerentă a aplicației si stocarea cu logică a datelor utilizatorilor. => componentele C1.1, C1.2

US2. Ca și dezvoltator, doresc să integrez vendori securizati de payment pentru a asigura securitatea plăților efectuate în aplicație. => componenta C6

US3. Ca și dezvoltator, doresc să hostez aplicația folosind un serviciu de cloud consacrat pentru a asigura disponibilitatea și scalabilitatea aplicației. => toate componentele

# Componente de testare

1. Testare unitara folosind framework-uri specifice ASP .NET (MSTest/XUnit)
2. Testarea performanței pentru sincronizarea rezervărilor în timp real.
3. Testarea de securitate pentru protecția datelor.

# Componente de comunicare

1. Comunicare sincronă: ședințe de tip SCRUM, de ex Daily (ce probabil va fi cam de 2 ori pe săptămână), Retro și Demo.
2. Comunicare asincronă: utilizarea unui sistem de ticketing (poate vom folosi GitHub) și platforme online pentru colaborare (vom folosi Microsoft Teams și Discord).

# Componente de învățare

1. Sesiuni de coaching pentru .NET si React
2. Invățarea per total a anumitor cunoștințe de Software Engineering, nu neapărat legat direct de programare, ci mai mult de Soft Skills.

# Perspectiva de dezvoltare

În primele două sprinturi (până în deadline-ul intermediar) echipa o să se concentreze să implementeze tot F1, cât de mult se poate din F3 și story-urile necesare din F5 (US1 pentru baza de date), restul se va mock-ui. Astfel, EPICul este format din F1, F3 și F5.

Pentru cazul minimal implementăm tot US1 din F5, tot F1 (cu adăugarea de comandă mockuită de către client) și parțial F3 (măcar US1 și preferabil US4). Pentru cazul maximal încercăm să implementăm tot F3.

În cadrul sprinturilor vom granulariza și mai mult usuer storiesurile unde este nevoie.

**Sprint 1:**

1. F5 - US1
2. F1 - US1, US2 (tot US2, adica US2.1, US2.2, US2.3 și US2.4), US3

**Sprint 2:**

1. F1 - US4
2. F3 - US1 și US4
3. (Stretch goal) F3 – US2, US3 și US5

Celelalte două sprinturi vor acoperi terminarea F3 (daca acesta nu a fost complet terminat in primele 2 sprinturi) și implementarea a cât mai multe stories din F2 și F5.

**Sprint 3:**

1. (Dacă nu au fost implementate în Sprintul 2) F3 – US2, US3 și US5
2. F2 - US1, US2, US3, US4 și US5

**Sprint 4:**

1. F5 - US2 și US3
2. (Stretch goal) F4

Flow-ul F4 nu este prioritar deci nu intră în scope-ul acestor 4 stories, rămâne în afara epicului, și va fi implementat doar dacă totul este terminat mai devreme decât a fost planificat.