<Aplicație informatică pentru managementul și recomandarea rețetelor culinare>

# Documentul de proiectare

Cuprins

[Documentul de proiectare i](#_Toc163583867)

[1. Introducere 1](#_Toc163583868)

[1.1 Scopul documentului 1](#_Toc163583869)

[2. Prezentare generală și abordări de proiectare 2](#_Toc163583870)

[2.1 Prezentare generală 2](#_Toc163583871)

[3. Considerațiii de proiectare 3](#_Toc163583872)

[3.1 Metode de dezvoltare 3](#_Toc163583873)

[3.2 Strategii de arhitectură 8](#_Toc163583874)

[4. Arhitectura Sistemului și Proiectarea Arhitecturii 9](#_Toc163583875)

[4.1 Arhitectură software 9](#_Toc163583876)

[4.2 Arhitectura informațiilor 9](#_Toc163583877)

[4.3 Arhitectura de comunicații interne 10](#_Toc163583878)

[4.4 Diagrama de arhitectură a sistemului 10](#_Toc163583879)

[5. Proiectarea sistemului 11](#_Toc163583880)

[5.1 Proiectarea bazei de date 11](#_Toc163583881)

[5.2 Interfețe utilizator 12](#_Toc163583882)

[5.3 Proiectarea interfețelor cu utilizatorul 12](#_Toc163583883)

[6. Scenarii de utilizare 14](#_Toc163583884)

[7. Proiectare de detaliu 16](#_Toc163583885)

[7.1 Proiectare software de detaliu 16](#_Toc163583886)

## Introducere

Documentul de Proiectare a Sistemului descrie modul în care cerințele funcționale și non-funcționale înregistrate în Documentul de Cerințe se transformă în specificații de proiectare a sistemului, mai tehnice, pe baza cărora se construiește sistemul. Se documentează proiectarea sistemului la un nivel înalt și specificațiile detaliate de proiectare.

Se descriu obiectivele și considerațiile de proiectare, se furnizează o prezentare generală la nivel înalt a arhitecturii sistemului și se descrie proiectarea datelor asociate cu sistemul, precum și interfața om-mașină și scenariile operaționale.

Proiectarea sistemului la nivel înalt este descompusă în continuare în specificații detaliate de proiectare pentru fiecare componentă a sistemului, inclusiv hardware, comunicațiile interne, software, controalele de integritate a sistemului și interfețele externe.

### Scopul documentului

Scopul acestui Document de Proiectare a Sistemului (SDD) este de a furniza o îndrumare detaliată pentru arhitectura și designul sistemului propus sau existent. Acest document este adaptat pentru a se potrivi nevoilor proiectului nostru specific, cu accent pe eficiența și coerența arhitecturii și designului sistemului.

Documentul SDD urmărește să documenteze și să urmărească informațiile necesare pentru a defini arhitectura și designul sistemului într-un mod eficient. Este un livrabil esențial pentru echipa de dezvoltare, oferindu-le îndrumare și informații detaliate pentru a construi și implementa sistemul conform cerințelor stabilite în Documentul de Cerințe.

Aceste documente de proiectare sunt produse incremental și iterativ pe parcursul ciclului de viață al dezvoltării sistemului, adaptându-se circumstanțelor specifice ale proiectului de IT și metodologiei de dezvoltare utilizate. Acest document este destinat a fi utilizat de către managerul de proiect, echipa de proiect și echipa de dezvoltare pentru a asigura o înțelegere clară și coerentă a arhitecturii și designului sistemului.

Anumite părți ale acestui document, cum ar fi interfața cu utilizatorul (UI), pot fi împărtășite cu clientul/utilizatorul și cu alte părți interesate ale căror contribuții/aprobări sunt necesare în UI. Astfel, SDD-ul servește ca un punct de referință pentru toate părțile implicate în proiectul nostru, asigurând o comunicare și colaborare eficientă în timpul dezvoltării sistemului.

## Prezentare generală și abordări de proiectare

### Prezentare generală

Proiectul vizează dezvoltarea unei aplicații de management culinar, care să ofere utilizatorilor posibilitatea de a căuta, salva și organiza rețete culinare, precum și de a comunica și colabora în cadrul grupurilor sociale pentru a planifica și a găti rețete împreună. Aplicația urmărește să ofere o experiență completă și personalizată în domeniul gătitului, adaptată atât începătorilor, cât și bucătarilor avansați.

Obiective:

1. Gestionare cont utilizatori

* Permite utilizatorilor să își creeze propriile conturi și să le gestioneze

1. Gestionare preferințe

* Permite utilizatorilor să își seteze preferințele alimentare, culinare și să specifice alergiile alimentare

1. Încărcare rețete personale

* Dezvoltarea unei functionalități de încărcare a rețetelor proprii, inclusiv imagini, ingrediente, mod de preparare, detalii nutriționale
* Posibilitatea editării și ștergerii rețetelor încărcate anterior

1. Filtrare

* Implementarea unui sistem de căutare avansată, permițând filtrarea după ingrediente, tip de mâncare, tip de masă

1. Recomandari personalizate

* Crearea unui algoritm de recomandare bazat pe preferintele individuale, istoric de cautare si ingrediente preferate

1. Optimizare pentru dispozitive mobile

* Asigurarea compatibilitatii si experientei de utilizare pe diferite dispozitive

1. Interacțiune socială

* Permite utilizatorilor să interacționeze cu alți utilizatori și să ofere feedback

## Considerațiii de proiectare

### Metode de dezvoltare

Este folosită o abordare structurată pentru a organiza și defini cerințele sistemului, identificând componente și interacțiuni între acestea. Această metodă ajută la clarificarea și definirea cerințelor funcționale și non-funcționale ale sistemului, precum și la structurarea arhitecturii sistemului într-un mod coerent și eficient.

Sunt utilizate principiile orientării pe obiecte pentru a proiecta componente modulare și reutilizabile ale sistemului. Această abordare ne permite să împărțim sistemul în entități logice, care interacționează între ele prin intermediul mesajelor și metodelor. Astfel, putem realiza o dezvoltare flexibilă și ușor extensibilă, care să permită adaptarea sistemului la schimbările ulterioare ale cerințelor.

Sunt folosite practici Agile, cum ar fi dezvoltarea iterativă, colaborarea interdisciplinară și feedback-ul continuu, pentru a răspunde eficient la schimbările cerințelor și pentru a livra rapid funcționalități valoroase. Această abordare ne permite să ne concentrăm pe prioritățile cheie ale proiectului și să ajustăm design-ul și implementarea pe măsură ce obținem feedback de la utilizatori.

Tehnologiile utilizate sunt multiple și vor fi prezentate mai jos.

#### Limbajul HTML

HTML este un limbaj de marcare utilizat pentru structurarea și prezentarea conținutului unei pagini web. Acesta permite organizarea și formatarea textului, adăugarea de imagini, crearea de link-uri și integrarea altor elemente media într-un mod structurat. Deși HTML se concentrează mai mult pe prezentarea informațiilor, există elemente și atribute specifice care pot furniza și informații semantice despre conținutul paginii.

În dezvoltarea web de tip front-end, HTML este utilizat împreună cu CSS (Cascading Style Sheets) pentru a defini stilul și aspectul paginii și cu JavaScript pentru a adăuga funcționalități interactive și dinamice.

Specificațiile și standardele HTML sunt stabilite și menținute de organizația World Wide Web Consortium (W3C) pentru a asigura interoperabilitatea și compatibilitatea între diferitele browsere și dispozitive. CSS este adesea folosit pentru a separa prezentarea de structura HTML, facilitând astfel întreținerea și modificarea ulterioară a paginii.

HTML este un format text utilizat pentru crearea și structurarea paginilor web. Într-adevăr, este posibil să se creeze și să se editeze pagini HTML folosind un editor de text simplu, dar acest lucru necesită cunoștințe avansate de HTML și poate fi o sarcină consumatoare de timp.

Pentru a facilita procesul de creare a paginilor web, au fost dezvoltate și utilizate editorii grafici de tip WYSIWYG (What You See Is What You Get) precum Macromedia Dreamweaver, Adobe GoLive sau Microsoft FrontPage. Aceste programe permit utilizatorilor să creeze pagini web folosind o interfață grafică similară cu cea a unui editor de text obișnuit sau a unui program de procesare a textului. Cu toate acestea, trebuie menționat că editorii grafici pot genera uneori un cod HTML de calitate inferioară sau ineficient din punct de vedere al performanței.

Este important să se aibă în vedere că utilizarea unui editor grafic nu elimină necesitatea de a avea cunoștințe de bază despre HTML și structura paginilor web. Înțelegerea conceptelor și a structurii HTML va permite o mai mare flexibilitate și control asupra paginilor create, iar utilizarea unui editor grafic poate fi considerată un instrument suplimentar pentru a spori eficiența și productivitatea în dezvoltarea web.[[1]](#footnote-1)

#### A blue and white logo Description automatically generatedLimbajul CSS

CSS sau Cascading Style Sheets este un standard folosit pentru a formata și stiliza elementele unui document HTML. În loc să formatezi direct elementele HTML, CSS permite separarea aspectului și prezentării vizuale a paginii de conținutul său.

Prin intermediul CSS, putem specifica aspecte precum culorile, fonturile, dimensiunile, fundalurile și alinierea elementelor HTML. Acest lucru permite o mai mare flexibilitate și control asupra aspectului paginii web.

Un avantaj important al CSS este că permite aplicarea unui set comun de stiluri pentru mai multe pagini. Astfel, putem crea o foaie de stiluri CSS separată, pe care o putem atașa la diferite pagini HTML. Aceasta simplifică procesul de modificare a aspectului global al unui site web, deoarece modificările se reflectă automat în toate paginile care utilizează acea foaie de stiluri.

De asemenea, utilizarea CSS permite separarea clară a conținutului și prezentării, ceea ce facilitează și îmbunătățește accesibilitatea paginilor web. Motorul de căutare și dispozitivele asistive pot accesa și interpreta mai ușor conținutul semantic al paginii, în timp ce stilurile CSS oferă prezentarea adecvată.

CSS3 reprezintă o versiune actualizată a standardului CSS care aduce mai multe funcționalități și atribute noi pentru a sprijini dezvoltarea webdesign-ului modern.

Unul dintre aspectele-cheie ale CSS3 este introducerea de selecționatori mai avansați, care permit selectarea și stilizarea elementelor HTML în mod mai precis și flexibil.

Modulele CSS3 dedicat background-urilor și border-urilor aduc mai multe opțiuni pentru adăugarea de imagini de fundal și stilizarea border-urilor.

CSS3 permite, de asemenea, aplicarea de efecte vizuale asupra textului, cum ar fi umbre, contururi și degradeuri, oferind astfel mai multă flexibilitate în stilizarea textului.

Transformările 2D și 3D adăugate în CSS3 permit rotirea, scalarea și translația elementelor HTML în plan 2D și 3D, ceea ce deschide noi posibilități pentru crearea de animații și efecte vizuale impresionante.

Un alt modul important din CSS3 este cel pentru animații, care permite definirea și controlul tranzițiilor și animațiilor elementelor HTML, adăugând un nivel suplimentar de interactivitate și dinamică paginilor web.[[2]](#footnote-2)

#### A yellow and black logo Description automatically generatedLimbajul JavaScript

JavaScript (JS) este un limbaj de programare orientat obiect, utilizat în principal pentru dezvoltarea de funcționalități interactive în paginile web. Este executat de către browser și permite manipularea și interacțiunea cu elementele și conținutul paginii.

JavaScript este cunoscut pentru utilizarea sa în construirea site-urilor web, adăugând interactivitate și dinamică paginilor. Cu toate acestea, poate fi utilizat și în alte contexte, cum ar fi accesul la obiecte încapsulate în alte aplicații.

Limbajul a fost dezvoltat inițial de către Brendan Eich de la Netscape Communications Corporation și a avut mai multe denumiri înainte de a fi numit JavaScript. Deși are o sintaxă similară cu limbajul Java, JavaScript nu are nicio legătură cu acesta. De fapt, JavaScript are mai multe asemănări cu limbajul Self și se bazează pe conceptul de prototipuri pentru programare orientată obiect.

Versiunile JavaScript sunt standardizate de către ECMA International prin specificația ECMAScript. Ultimele versiuni includ funcționalități avansate precum manipularea XML (prin E4X), suport pentru module și sintaxă îmbunătățită. Fiecare browser are propria implementare a JavaScript, cu suport pentru diferite versiuni ale limbajului.

JavaScript este esențial în dezvoltarea web modernă, oferind posibilități extinse pentru crearea de interfețe interactive, validarea datelor, gestionarea evenimentelor și multe altele. Prin intermediul bibliotecilor și framework-urilor JavaScript, programatorii pot construi aplicații web complexe și dinamice.

JavaScript a primit numele actual, adică JavaScript, în jurul aceluiași timp când Netscape a început să integreze suportul pentru tehnologia Java în browserul web Netscape Navigator. Acest lucru a fost făcut pentru a profita de popularitatea limbajului Java și pentru a atrage mai mulți dezvoltatori către JavaScript.

Pe de altă parte, Microsoft a dezvoltat propria implementare a limbajului JavaScript numită JScript, care a adus unele modificări și extensii față de implementarea Netscape. JScript a fost integrat în platforma Windows și poate fi utilizat pentru scriptarea aplicațiilor care suportă Windows Script, cum ar fi Internet Explorer sau chiar sistemul de operare Windows.[[3]](#footnote-3)

#### React

A blue and black symbol

Description automatically generatedÎn lumea dezvoltării web moderne, React a devenit una dintre cele mai populare și influente biblioteci JavaScript. Creată și întreținută de Facebook, React a revoluționat modul în care dezvoltatorii construiesc interfețe utilizator, aducând o abordare reactivă și eficientă în gestionarea stării aplicațiilor web.

React a fost lansată pentru prima dată în 2013, fiind dezvoltată inițial pentru a gestiona interfețe complexe și dinamice ale platformei Facebook. De atunci, a evoluat constant și a câștigat popularitate în rândul dezvoltatorilor, devenind o alegere frecventă pentru proiectele web de diverse dimensiuni.

React se bazează pe două concepte cheie: componentele și starea. Componentele sunt blocurile de construcție fundamentale ale interfețelor React, reprezentând porțiuni izolate și reutilizabile ale interfeței utilizator. Acestea pot varia de la simple elemente precum butoane sau casete de text, până la componente complexe, cum ar fi bare laterale sau interfețe de utilizator interactive.

Starea, pe de altă parte, reprezintă datele interne ale unei componente. React introduce un model unidirecțional de flux de date, ceea ce înseamnă că schimbările stării unei componente declanșează actualizări automate ale interfeței utilizator. Această abordare simplifică gestionarea stării și reduce riscul de bug-uri legate de sincronizarea datelor.

Un alt aspect esențial al React este Virtual DOM (Document Object Model). Atunci când starea unei componente se schimbă, React nu actualizează direct DOM-ul, ci creează mai întâi o reprezentare virtuală a acestuia. Acest Virtual DOM este apoi comparat cu DOM-ul real, și doar modificările sunt aplicate eficient. Acest proces minimizează costurile de re-renderizare, contribuind astfel la performanța și eficiența aplicațiilor React.

React promovează puternic principiul reutilizabilității și modularității. Dezvoltatorii pot crea componente independente care pot fi utilizate în mai multe locuri ale aplicației. Această abordare nu numai că reduce duplicarea codului, ci facilitează și întreținerea și scalabilitatea proiectelor.

React este însoțită de un ecosistem bogat de instrumente și biblioteci, cum ar fi Redux pentru gestionarea stării, React Router pentru gestionarea navigării, și multe altele. Aceste resurse extind funcționalitatea și eficiența React, oferind dezvoltatorilor instrumente puternice pentru a construi aplicații web complexe și performante.[[4]](#footnote-4)

#### A yellow folded paper with a black background Description automatically generatedFirebase

Firebase este o suită extinsă de servicii cloud dezvoltată de Google, oferind un set cuprinzător de instrumente și servicii pentru dezvoltarea, gestionarea și scalarea aplicațiilor web și mobile.

Firebase oferă o bază de date în timp real (Realtime Database) care permite sincronizarea datelor în timp real între utilizatori și dispozitive. Acest serviciu JSON bazat pe cloud facilitează dezvoltarea aplicațiilor colaborative și interactive, unde schimbările în date sunt transmise instantaneu.

Furnizează un sistem de autentificare puternic, permițând dezvoltatorilor să integreze ușor autentificarea cu servicii precum Google, Facebook, Twitter și altele. De asemenea, oferă funcționalități robuste de autorizare, permițând controlul precis al accesului utilizatorilor la diferite părți ale aplicației.

Serviciul Firebase Hosting oferă un mediu de găzduire web rapid și sigur pentru aplicații web statice sau dinamice. Utilizând concepte de serverless computing, dezvoltatorii pot concentra eforturile asupra codului și a funcționalității, în timp ce Firebase se ocupă de gestionarea infrastructurii.

Firebase Cloud Functions permite dezvoltatorilor să scrie funcții backend scalabile și serverless. Aceste funcții pot fi declanșate de evenimente în cadrul ecosistemului Firebase sau de alte evenimente din Google Cloud Platform, oferind o modalitate eficientă de a extinde funcționalitățile aplicației.

Firebase Cloud Firestore este un serviciu de bază de date NoSQL care oferă o structură de stocare flexibilă, scalabilă și ușor de utilizat. Documentele Firestore pot fi organizate în colecții și pot fi accesate eficient pentru a satisface cerințele complexe ale aplicațiilor moderne.

Serviciul Cloud Storage pentru Firebase oferă o soluție de stocare a fișierelor în cloud, facilitând încărcarea și descărcarea de fișiere de către utilizatori. Acest serviciu este util în special pentru gestionarea și partajarea fișierelor media în aplicații.

Firebase Analytics furnizează informații detaliate despre comportamentul utilizatorilor în aplicație, permițând dezvoltatorilor să înțeleagă mai bine cum interacționează utilizatorii cu aplicația lor. Serviciul de monitorizare Crashlytics ajută la detectarea și raportarea erorilor, facilitând îmbunătățirea stabilității aplicației.

Firebase Cloud Messaging (FCM) oferă funcționalități puternice pentru trimiterea de notificări push către utilizatori pe diverse platforme, inclusiv Android și iOS.[[5]](#footnote-5)

### Strategii de arhitectură

Am ales să folosesc tehnologii precum React și Firebase pentru eficiența lor în dezvoltarea rapidă și eficientă a aplicației. Aceste instrumente oferă caracteristici puternice și sunt bine integrate, facilitând procesul de creare a interfeței de utilizator și gestionarea datelor în timp real. De asemenea, am luat în considerare scalabilitatea oferită de Firebase, care este esențială pentru o aplicație cu potențial de creștere rapidă a utilizatorilor. Utilizarea bibliotecii React pentru frontend și Firebase pentru backend ne asigură și o performanță optimă a aplicației, oferind o experiență fluidă și interactivă utilizatorilor noștri. În plus, Firebase furnizează o suită de servicii cloud bine integrate, cum ar fi stocarea datelor, autentificarea și hosting-ul aplicației, eliminând astfel nevoia de a configura și gestiona infrastructura proprie.

## Arhitectura Sistemului și Proiectarea Arhitecturii

Arhitectura sistemului pentru aplicația web de management și recomandare a rețetelor culinare se bazează pe o combinație de tehnologii moderne pentru a oferi o experiență de utilizare eficientă și scalabilă. La nivelul frontend-ului, interfața utilizator este dezvoltată utilizând limbajele HTML, CSS și JavaScript, iar pentru gestionarea stării și logicii aplicației se utilizează biblioteca React.

Pagina web este structurată în componente React reutilizabile și bine organizate, permitând o dezvoltare ușoară și eficientă. Componentele se ocupă de diverse aspecte ale interfeței, precum afișarea rețetelor, gestionarea interacțiunii utilizatorului și prezentarea recomandărilor personalizate. CSS-ul este utilizat pentru stilizarea componentelor și pentru a asigura un aspect atractiv și coerent al aplicației.

La nivelul backend-ului, sistemul beneficiază de puterea și scalabilitatea serviciilor Firebase, care include servicii precum Firebase Authentication, Firestore pentru stocarea datelor în timp real și Firebase Hosting pentru implementarea aplicației web. Firebase Authentication asigură securitatea autentificării utilizatorilor, iar Firestore facilitează gestionarea bazelor de date în timp real, oferind un mediu eficient pentru stocarea și accesul la date.

Această arhitectură oferă o experiență de utilizare fluentă, interactivă și personalizată pentru gestionarea și recomandarea rețetelor culinare, beneficiind de tehnologiile moderne și scalabilitatea oferită de Firebase.

### Arhitectură software

Un utilizator introduce o adresă URL în browserul web, prin care trimite o solicitare la serverul web. Fișierele HTML, CSS și JavaScript sunt stocate pe server, în Firebase Realtime Database, apoi sunt trimise către client. Utilizatorul interacționează cu aplicația web, iar serverul procesează informația pe care o trimite înapoi.

### Arhitectura informațiilor

Informatiile stocate in sistem:

* Utilizatori: id utilizator, nume de utilizator, adresa de email, preferinte culinare
* Retete culinare: id reteta, nume reteta, ingredient, cantitati, instructiuni de preparare, imagine reteta, descriere reteta
* Grup de chat: id grup, nume grup, descriere grup, membrii grup, mesaje

Informatiile cu caracter sensibil sunt numele de utilizator, adresa de email si parola fiecarui utilizator, deoarece sunt legate de confidentialitatea acestora si nu trebuie sa permita accesul neautorizat.

Utilizatorii isi introduce manual informatiile personale la crearea contului.

Retetele culinare sunt adaugate de catre utilizatori.

Informatiile despre grupurile de chat si taskuri sunt generate de catre utilizatori in cadrul aplicatiei.

### Arhitectura de comunicații interne

Arhitectura sistemului va fi bazată pe un model client-server, în care aplicația client va comunica cu un server backend pentru accesarea și gestionarea datelor. Serverul backend va folosi o bază de date pentru stocarea informațiilor despre utilizatori, rețete, grupuri sociale și altele.

Aplicația de management și recomandare a rețetelor culinare, implementeaza o arhitectură de rețea care se bazează pe o combinație de echipamente hardware și tehnologii software pentru a facilita transmiterea datelor și comunicarea între utilizatori și servere.

Componente Software:

- Protocol TCP/IP: Folosit pentru a permite comunicația între dispozitivele conectate la rețea.

- HTTPS (HTTP Secure): Utilizat pentru a asigura transmiterea sigură a datelor între browserul utilizatorului și serverele noastre.

În cazul in care conexiunea la internet nu poate fi realizată, pentru a asigura transmiterea datelor, putem utiliza capacitatea dispozitivului utilizatorului de a stoca datele local. Astfel, atunci când utilizatorul este offline, aplicația poate să salveze datele în memoria locală a dispozitivului. Aceste date pot fi ulterior sincronizate cu serverul atunci când conexiunea la internet devine disponibilă din nou.

### Diagrama de arhitectură a sistemului

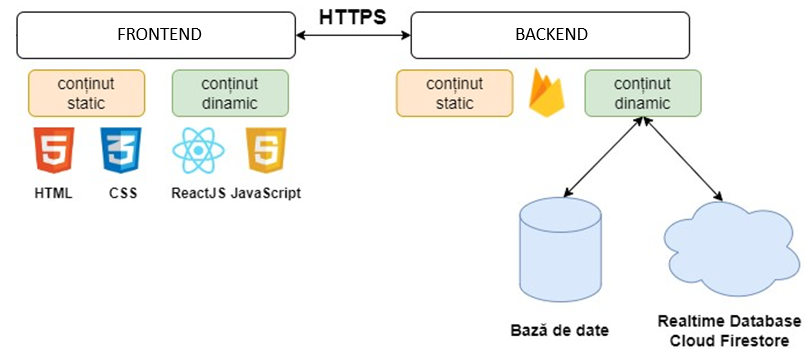


Figura 1 Server Web

## Proiectarea sistemului

### Proiectarea bazei de date

#### Obiecte de date și structuri de date rezultante

1. Utilizator:

- Structura de Date:

- user\_id (String): Identificator unic pentru utilizator.

- email (String): Adresa de email a utilizatorului.

- password (String): Parola utilizatorului criptată.

- nume\_utilizator (String): numele utilizatorului.

- prenume\_utilizator (String): prenumele utilizatorului.

- imagine\_profil (String): imaginea de profil a utilizatorului.

- preferinte\_utilizator (Array of Strings): preferintele utilizatorului

2. Rețetă:

- Structura de Date:

- recipe\_id (String): Identificator unic pentru rețetă.

- titlu (String): Titlul rețetei.

- ingrediente (Array of Strings): Lista de ingrediente necesare pentru rețetă.

- instructiuni (Array of Strings): Instrucțiunile de preparare pentru rețetă.

- categorie\_reteta (String): categoria in care se afla reteta.

- timp\_preparare\_reteta (Number): timpul de preparare al retetei.

- dificultate\_reteta (String): nivelul de dificultate al retetei.

3. Recenzie:

- Structura de Date:

- recenzie\_id (String): Identificator unic pentru recenzie.

- user\_id (String): Identificatorul unic al utilizatorului care a scris recenzia.

- reteta\_id (String): Identificatorul unic al rețetei la care se referă recenzia.

- rating (Number): Scorul acordat rețetei (opțional).

- comment (String): Comentariul scris de utilizator.

- data\_postarii (Date): Data și ora la care recenzia a fost postată.

- aprecieri (Number): numarul de aprecieri.

4. Grup de Chat:

- Structura de Date:

- chat\_id (String): Identificator unic pentru grupul de chat.

- membri (Array of Strings): Lista de membri ai grupului.

- mesaje (Array of Objects): Lista de mesaje trimise în cadrul grupului, fiecare mesaj având un ID, expeditor, conținut.

5. Task:

- Structura de Date:

- task\_id (String): Identificator unic pentru task.

- descriere (String): Descrierea task-ului.

- status (String): Starea task-ului (îndeplinit, în lucru)

- asignat\_catre (String): Utilizatorul căruia i-a fost asignat task-ul.

### Interfețe utilizator

1. Utilizatori obișnuiți

Acești utilizatori reprezintă consumatorii principali ai aplicației de gestionare și recomandare a rețetelor culinare. Ei vor accesa aplicația pentru a găsi și adăuga rețete, a salva rețete preferate, a adăuga ingrediente în lista de cumpărături și a interacționa cu funcționalitățile de recenzii și comentarii.

2. Administratori

Acești utilizatori au responsabilitatea de a monitoriza și a gestiona activitatea comunității, inclusiv revizuirea și moderarea recenziilor și comentariilor, aplicarea regulilor comunității și gestionarea utilizatorilor.

### Proiectarea interfețelor cu utilizatorul

1. Pagina de login/register:

Utilizatorii vor avea posibilitatea de a se autentifica sau de a-și crea un cont nou folosind pagina de login/register. Aici vor introduce credențialele lor de acces sau vor completa un formular de înregistrare cu informațiile necesare.

2. Ecranul principal:

Aici, utilizatorul va fi întâmpinat cu retete recomandate de aplicație, alături de butoane pentru a aprecia (like) sau a dezaprecia (dislike) fiecare retetă. Dacă utilizatorul apreciază o retetă, aceasta va fi salvată în secțiunea "Favorite". De asemenea, pe acest ecran vor fi afișate toate retetele, alături de un filtru de căutare după ingrediente și alte categorii.

3. "Retetele mele":

Utilizatorul va putea adăuga propriile retete folosind butonul "Adaugă rețetă" și va putea vizualiza și gestiona retetele postate pe această pagină. De asemenea, retetele pot fi editate sau șterse conform preferințelor utilizatorului.

4. "Grupuri":

Utilizatorul va putea interacționa cu grupurile afișate pe ecran pentru a intra sau a crea un grup (public sau privat) prin intermediul butonului "Creează grup". După ce devine membru al unui grup, utilizatorul va putea trimite mesaje și va avea posibilitatea de a primi sau crea taskuri prin butoane sugestive.

5. "Favorite":

Utilizatorul va putea vizualiza retetele apreciate și informațiile acestora pe pagina "Favorite". De asemenea, va avea opțiunea de a le șterge din lista de aprecieri dacă dorește.

6. "Setări":

Utilizatorul va putea să își modifice detaliile contului, precum și preferințele și alergiile, accesând secțiunea "Setări".

7. "Lista de cumpărături":

Utilizatorul va putea să își scrie propria listă de cumpărături, salvând-o cu ajutorul butonului "Adaugă listă nouă de cumpărături". De asemenea, va putea să editeze sau să șteargă liste vechi. Dacă apasă pe butonul "Cumpără", utilizatorul va fi redirecționat către pagina unui supermarket online, unde ingredientele din lista sa vor fi automat selectate și adăugate în coșul de cumpărături pentru a facilita gestionarea cumpărăturilor.

## Scenarii de utilizare

Scenariu operațional: Crearea și Publicarea unei Rețete

1. Utilizatorul accesează aplicația web de management și recomandare a rețetelor culinare folosind un browser web.

2. Utilizatorul se autentifică în contul său sau își creează un cont nou, furnizând adresa de e-mail și parola sau folosind autentificarea socială.

3. După autentificare, utilizatorul navighează la secțiunea "Retetele mele" din bara cu meniul principal al aplicației.

4. Utilizatorul completează formularul de creare a rețetei, introducând un titlu pentru rețetă, o descriere, lista de ingrediente necesare și instrucțiunile pas cu pas pentru pregătirea rețetei.

5. Utilizatorul poate atașa o imagine rețetei, pentru a oferi o prezentare vizuală a acesteia.

6. După completarea formularului, utilizatorul apasă butonul "Publică Rețetă" pentru a finaliza procesul de creare și publicare a rețetei.

7. Sistemul validează informațiile introduse de utilizator și stochează rețeta în baza de date Firestore.

8. Utilizatorul vede reteta publicata in sectiunea „Retetele mele”.

Scenariu operațional: Utilizarea Grupurilor de Chat pentru Organizarea și Comunicarea în Grup

1. Utilizatorul se autentifică în aplicația web de management și recomandare a rețetelor culinare și navighează la secțiunea "Grupuri" din meniul principal al aplicației.

2. Utilizatorul poate vizualiza lista de grupuri de chat disponibile sau poate alege să creeze un nou grup de chat.

3. Pentru a crea un nou grup de chat, utilizatorul completează un formular care include un nume pentru grup, o scurtă descriere a scopului sau temei grupului și poate selecta dacă grupul va fi public sau privat.

4. Dacă utilizatorul alege să creeze un grup privat, va trebui să stabilească o parolă pentru accesul la grup.

5. După crearea sau selectarea unui grup de chat, utilizatorul poate vedea lista membrilor grupului și poate invita alți utilizatori să se alăture grupului.

6. Membrii grupului pot comunica între ei utilizând funcția de chat integrată în aplicație, unde pot discuta despre rețete, planifica întâlniri pentru a găti împreună sau organiza diverse activități legate de gătit.

7. Utilizatorii pot vedea task-urile atribuite grupului, cum ar fi cine aduce anumite ingrediente sau ustensile pentru o întâlnire de gătit.

8. Administratorii grupului au capacitatea de a gestiona membrii grupului, de a adăuga sau elimina membri și de a gestiona task-urile grupului.

9. Grupurile de chat pot fi accesate oricând de către utilizatori, iar istoricul conversațiilor este păstrat pentru a permite accesul la conversațiile anterioare și pentru a menține coerența și contextul discuțiilor.

Scenariul operațional: Generarea automată a listei de cumpărături

1. Utilizatorul accesează aplicația web de management și recomandare a rețetelor culinare folosind un browser web și se autentifică în contul său.

2. După autentificare, utilizatorul navighează pe pagina principală pentru a căuta rețeta dorită.

3. Utilizatorul găsește o rețetă pe care dorește să o pregătească și apasă pe butonul “Vezi mai mult” pentru a vedea lista de ingrediente necesare și instrucțiunile de preparare.

4. În dreptul listei de ingrediente, utilizatorul observă opțiunea "Adaugă la listă de cumpărături". Apasă pe această opțiune pentru a adăuga automat toate ingredientele necesare pentru rețetă în lista de cumpărături.

5. Sistemul preia automat toate informațiile despre ingredientele necesare din rețeta selectată și le adaugă în lista de cumpărături a utilizatorului.

6. Utilizatorul poate reveni la lista de cumpărături accesând secțiunea corespunzătoare din meniul principal al aplicației pentru a vedea și gestiona lista de cumpărături generată automat.

7. În lista de cumpărături, utilizatorul poate vedea toate ingredientele necesare pentru rețeta selectată, precum și cantitățile corespunzătoare pentru fiecare ingredient.

8. Utilizatorul poate edita lista de cumpărături, eliminând sau adăugând alte produse necesare, în funcție de preferințele și necesitățile sale.

9. Atunci când utilizatorul este gata să facă cumpărăturile, poate apăsa pe butonul "Cumpără" sau "Trimite la supermarket", iar sistemul îl va redirecționa către pagina unui supermarket online, unde ingredientele din lista de cumpărături vor fi automat selectate și adăugate în coșul de cumpărături pentru a finaliza procesul de achiziție.

10. Utilizatorul finalizează procesul de cumpărare pe platforma supermarketului online, iar apoi se întoarce la aplicația noastră web pentru a continua navigarea sau a accesa alte funcționalități disponibile.

## Proiectare de detaliu

### Proiectare software de detaliu

Utilizatorul începe accesând aplicația web printr-un browser obișnuit. Aici, el este întâmpinat cu pagina principală a aplicației. Pentru a avea acces la funcționalitățile complete ale aplicației, utilizatorul se autentifică în contul său existent sau își creează un cont nou, introducând adresa de e-mail și parola corespunzătoare sau folosind autentificarea socială.

Utilizatorul navighează la secțiunea "Retetele mele" din bara de meniu principal al aplicației. În această secțiune, utilizatorul completează un formular specific pentru crearea unei noi rețete. Aici, el introducere un titlu pentru rețetă, o descriere detaliată, lista de ingrediente necesare și instrucțiunile pas cu pas pentru pregătirea rețetei. De asemenea, utilizatorul are opțiunea de a atașa o imagine rețetei, pentru a oferi o prezentare vizuală a acesteia. După completarea formularului, utilizatorul finalizează procesul apăsând butonul "Publică Rețetă". Sistemul efectuează o validare a informațiilor introduse de utilizator pentru a se asigura că toate câmpurile necesare sunt completate corect și apoi stochează rețeta în baza de date Firestore. Utilizatorul poate vedea rețeta publicată în secțiunea "Retetele mele", fiind acum disponibilă pentru vizualizare și apreciere de către ceilalți utilizatori ai aplicației.

Utilizatorul navighează la secțiunea "Grupuri" din meniul principal al aplicației. Aici, utilizatorul poate vedea lista de grupuri de chat disponibile sau poate decide să creeze un nou grup de chat pentru o anumită temă sau scop. Pentru a crea un nou grup de chat, utilizatorul completează un formular care include un nume pentru grup, o scurtă descriere a scopului sau temei grupului și poate selecta opțiunea de a face grupul public sau privat. Dacă utilizatorul alege să creeze un grup privat, va trebui să stabilească și o parolă pentru accesul la grup. După crearea sau selectarea unui grup de chat, utilizatorul poate vedea lista membrilor grupului și poate invita alți utilizatori să se alăture grupului. Membrii grupului pot comunica între ei utilizând funcția de chat integrată în aplicație, unde pot discuta despre rețete, planifica întâlniri pentru a găti împreună sau organiza diverse activități legate de gătit. Utilizatorii pot vedea task-urile atribuite grupului, cum ar fi cine aduce anumite ingrediente sau ustensile pentru o întâlnire de gătit. Administratorii grupului au capacitatea de a gestiona membrii grupului, de a adăuga sau elimina membri și de a gestiona task-urile grupului, asigurând astfel o organizare eficientă și clară a activităților grupului. Grupurile de chat pot fi accesate oricând de către utilizatori, iar istoricul conversațiilor este păstrat pentru a permite accesul la conversațiile anterioare și pentru a menține coerența și contextul discuțiilor.

Utilizatorul navighează pe pagina principală a aplicației pentru a căuta rețeta dorită, fie folosind funcția de căutare, fie navigând prin lista de rețete recomandate. După ce găsește o rețetă pe care dorește să o pregătească, utilizatorul accesează pagina rețetei și vizualizează lista de ingrediente necesare și instrucțiunile de preparare. Utilizatorul apasă butonul "Adaugă la listă de cumpărături" și sistemul preia automat informațiile despre ingredientele necesare, adăugându-le în lista de cumpărături a utilizatorului. După ce ingredientele sunt adăugate, utilizatorul poate accesa lista de cumpărături din meniul principal al aplicației pentru a vedea și gestiona conținutul său. Aici, poate edita lista, eliminând sau adăugând produse necesare suplimentare. Când este pregătit să facă cumpărăturile, utilizatorul apasă butonul "Cumpără" sau "Trimite la supermarket", iar sistemul îl redirecționează către pagina unui supermarket online, unde ingredientele din lista de cumpărături sunt automat selectate și adăugate în coșul de cumpărături. După finalizarea achiziției pe platforma supermarketului online, utilizatorul se întoarce la aplicația noastră web pentru a continua utilizarea acesteia sau pentru a accesa alte funcționalități disponibile.

1. (HyperText Markup Language, 2024) [↑](#footnote-ref-1)
2. (Cascading Style Sheets, 2024) [↑](#footnote-ref-2)
3. (JavaScript, 2024) [↑](#footnote-ref-3)
4. (React, 2024) [↑](#footnote-ref-4)
5. (Firebase, 2024) [↑](#footnote-ref-5)