Aplicație pentru gestionarea și prepararea rețetelor culinare

# Documentul de proiectare

Cuprins

[1. Introducere 1](#_Toc197769082)

[1.1 Scopul documentului 1](#_Toc197769083)

[2. Prezentare generală și abordări de proiectare 2](#_Toc197769084)

[2.1 Prezentare generală 2](#_Toc197769085)

[2.2 Presupuneri/ Constrângeri/ Riscuri 2](#_Toc197769086)

[2.2.1 Presupuneri 2](#_Toc197769087)

[2.2.2 Constrângeri 2](#_Toc197769088)

[2.2.3 Riscuri 2](#_Toc197769089)

[3. Considerații de proiectare 3](#_Toc197769090)

[3.1 Obiective și linii directoare (ghiduri) 3](#_Toc197769091)

[3.2 Metode de dezvoltare 3](#_Toc197769092)

[3.3 Strategii de arhitectură 3](#_Toc197769093)

[4. Arhitectura Sistemului și Proiectarea Arhitecturii 4](#_Toc197769094)

[4.1 Vedere logică 4](#_Toc197769095)

[4.2 Arhitectură hardware 4](#_Toc197769096)

[4.3 Arhitectură software 4](#_Toc197769097)

[4.4 Arhitectura informațiilor 4](#_Toc197769098)

[4.5 Arhitectura de comunicații interne 4](#_Toc197769099)

[4.6 Diagrama de arhitectură a sistemului 4](#_Toc197769100)

[5. Proiectarea sistemului 5](#_Toc197769101)

[5.1 Proiectarea bazei de date 5](#_Toc197769102)

[5.1.1 Obiecte de date și structuri de date rezultante 5](#_Toc197769103)

[5.1.2 Fișiere și baze de date 5](#_Toc197769104)

[5.2 Conversii de date 5](#_Toc197769105)

[5.3 Interfețe utilizator 5](#_Toc197769106)

[5.3.1 Intrări 5](#_Toc197769107)

[5.3.2 Ieșiri 6](#_Toc197769108)

[5.4 Proiectarea interfețelor cu utilizatorul 6](#_Toc197769109)

[6. Scenarii de utilizare 7](#_Toc197769110)

[7. Proiectare de detaliu 8](#_Toc197769111)

[7.1 Proiectare hardware de detaliu 8](#_Toc197769112)

[7.2 Proiectare software de detaliu 8](#_Toc197769113)

[7.3 Proiectare detaliată de securitate 8](#_Toc197769114)

[7.4 Proiectare de detaliu pentru performanța sistemului 8](#_Toc197769115)

[7.5 Proiectare detaliată a comunicațiilor interne (între componente) 8](#_Toc197769116)

[8. Controale pentru verificarea integrității sistemului 9](#_Toc197769117)

[Anexa A: Gestiunea modificărilor documentului 10](#_Toc197769118)

[Anexa B: Acronime 11](#_Toc197769119)

## Introducere

Documentul de Proiectare a Sistemului descrie modul în care cerințele funcționale și non-funcționale stabilite în faza de analiză sunt transpuse într-o arhitectură concretă a aplicației pentru gestionarea și prepararea rețetelor culinare. Aplicația are scopul de a permite utilizatorilor autentificați să caute rețete culinare în funcție de denumire sau ingrediente disponibile, să salveze favorite, să genereze planuri de gătit și liste de cumpărături. Documentul descrie structura arhitecturală, tehnologiile folosite, considerente de proiectare, precum și scenarii operaționale care reflectă modul real de utilizare al sistemului.

### Scopul documentului

Scopul acestui document este de a furniza o bază tehnică detaliată echipei de dezvoltare pentru implementarea aplicației, respectând cerințele inițiale ale beneficiarului. Acesta conține deciziile arhitecturale fundamentale, strategii de proiectare, structura componentelor și fluxurile de date. Documentul este adresat echipei de proiect, programatorilor și managerului de proiect, dar anumite secțiuni precum interfețele grafice pot fi partajate și cu utilizatorii finali pentru validare și feedback.

## Prezentare generală și abordări de proiectare

### Prezentare generală

Aplicația este structurată modular, fiind organizată în componente care corespund principalelor funcționalități: autentificare, căutare rețete, gestionare favorite, planificator și contul utilizatorului. Sistemul este construit cu o interfață unică, fără schimbare de ferestre, astfel încât utilizatorul navighează fluid prin toate componentele aplicației dintr-o singură interfață principală

### Presupuneri/ Constrângeri/ Riscuri

#### Presupuneri

Se presupune că utilizatorii vor folosi aplicația pe un sistem de operare Windows 10 sau superior, cu un ecran suficient de mare pentru o interfață grafică completă. Se presupune că utilizatorii își vor crea conturi reale și vor respecta structura formularului.

#### Constrângeri

Aplicația este dependentă de platforma Windows Forms și .NET, ceea ce limitează portabilitatea acesteia pe alte sisteme de operare. Toate datele sunt stocate local în SQLite, iar aplicația nu oferă încă suport pentru sincronizare online sau stocare în cloud. Nu există criptare a parolelor, ceea ce impune limitări asupra securității datelor sensibile.

#### Riscuri

Principalele riscuri sunt asociate cu pierderea datelor din baza de date locală, lipsa unui sistem de backup și securitatea insuficientă a credențialelor. În cazul deteriorării fișierului bazei de date, toate datele utilizatorului pot fi pierdute. Neimplementarea criptării poate duce la compromiterea informațiilor în cazul accesului neautorizat la fișiere.

## Considerații de proiectare

### Obiective și linii directoare (ghiduri)

Designul aplicației pune accent pe claritate, simplitate și coerență vizuală. Interfața utilizatorului este menținută minimalistă, folosind componente standard Windows Forms, cu palete de culori constante și controale etichetate clar. Codul este structurat în clase și metode separate pentru fiecare componentă, folosind convenții de denumire clare și comentarii explicative.

### Metode de dezvoltare

Proiectarea și dezvoltarea aplicației au fost realizate într-un mod incremental, folosind modelul de dezvoltare iterativ. Codul este scris în C#, utilizând biblioteci native .NET și accesul la baza de date prin SQLite. Interfața a fost prototipată vizual în Windows Forms Designer, iar logica a fost adăugată treptat, în paralel cu testarea funcționalităților.

### Strategii de arhitectură

Arhitectura aplicației este de tip monolit, dar modulară logic. Fiecare funcționalitate este implementată într-un modul clar separat (de exemplu, login, căutare, favorite), iar logica fiecăruia este izolată. Decizia de a păstra totul într-un singur formular principal a fost luată pentru a simplifica navigarea utilizatorului și pentru a evita multiple ferestre deschise în paralel.

## Arhitectura Sistemului și Proiectarea Arhitecturii

### Vedere logică

Aplicația este organizată logic în patru module principale: autentificare, interfață principală, funcționalitate de căutare și secțiunea de cont al utilizatorului. Fiecare modul gestionează propriul set de funcționalități și interacționează cu celelalte module prin schimbul de date și comenzi directe, folosind instanțe ale componentelor și apeluri directe între clase.

### Arhitectură hardware

Aplicația este destinată rulării pe calculatoare personale sau laptopuri cu arhitectură x64, cu sistem de operare Windows 10 sau superior. Este necesar un procesor de minim 2 GHz, 2 GB memorie RAM și cel puțin 100 MB spațiu liber pe disc pentru stocarea fișierului executabil și a bazei de date SQLite. Arhitectura este complet locală, neexistând cerințe privind conectarea la servere externe sau resurse distribuite.

### Arhitectură software

Sistemul este implementat în C# folosind platforma .NET și biblioteca Windows Forms pentru interfața grafică. Pentru stocarea datelor se utilizează SQLite, o bază de date relațională locală, accesată prin biblioteca System.Data.SQLite. Structura aplicației include clase precum FormLogin (folosită și ca interfață principală), clase statice pentru sesiune și utilitare pentru interogarea bazei de date. Nu sunt utilizate API-uri externe, iar toate datele sunt gestionate local.

### Arhitectura informațiilor

Aplicația stochează informații despre utilizatori (nume, parolă, email), rețete (nume, ingrediente, timp, pași, dietă, număr de porții) și relația dintre utilizatori și rețetele salvate ca favorite. Datele sunt stocate în format electronic, în baza de date SQLite, fără suport pentru fișiere externe sau tipărite. Informațiile nu sunt considerate sensibile, cu excepția parolelor, care în prezent nu sunt criptate.

### Arhitectura de comunicații interne

Fiind o aplicație desktop monolitică, comunicarea între componente este realizată direct, prin instanțierea de clase și apeluri de metode. Nu există un sistem de comunicație bazat pe rețea, iar toate operațiunile se realizează local. Componentele grafice transmit evenimente către logica aplicației, care comunică direct cu baza de date prin interfețe SQLite.

### Diagrama de arhitectură a sistemului

Structura sistemului este formată dintr-un modul principal (FormLogin), care inițial afișează interfața de autentificare, iar ulterior, după autentificare reușită, transformă interfața în zona principală a aplicației. Componenta de bază controlează afișarea conținutului, butoanele de navigare și interacțiunea cu baza de date. Modulele pentru căutare, favorite și contul utilizatorului sunt gestionate dinamic prin panouri și controale. Baza de date SQLite este componenta centrală pentru persistența datelor și este accesată direct din codul aplicației.

## Proiectarea sistemului

### Proiectarea bazei de date

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

Structura de date principală este reprezentată de tabelele bazei de date SQLite: Users, Recipes, Favorites și Planner. Fiecare tabel conține coloane clare pentru câmpurile relevante, cum ar fi Username, Password, Email pentru utilizatori sau Title, Ingredients, Steps pentru rețete. Structurile sunt relaționale, cu legături prin ID-uri unice și asocieri many-to-one între utilizatori și favorite.

#### Fișiere și baze de date

Fișierul bazei de date SQLite este salvat local în directorul aplicației și conține toate tabelele necesare funcționării aplicației. Fișierul este singular, neindexat extern, și se accesează direct prin comenzi SQL prin interfața System.Data.SQLite.

##### Baze de date

Baza de date este denumită „retete.db” și este structurată în următoarele tabele: Users (Id, Username, Password, Email), Recipes (Id, Title, Ingredients, Steps, Time, Difficulty, Diet), Favorites (Id, UserId, RecipeId), Planner (Id, UserId, RecipeId, DayOfWeek). Toate tabelele au chei primare, iar tabelele Favorites și Planner folosesc chei externe.

##### Fișiere non-DBMS

Nu sunt utilizate fișiere non-DBMS în afară de baza de date principală. Nu există fișiere temporare sau log-uri externe. Toate operațiile de scriere și citire sunt gestionate exclusiv prin baza de date. Datele sunt accesate și validate în timp real, fără fișiere intermediare.

### Conversii de date

Nu sunt necesare conversii de date externe. Toate datele sunt introduse direct de către utilizator prin interfața grafică și stocate conform structurii bazei de date SQLite. Nu există proces de migrare sau preluare a datelor din alte surse.

### Interfețe utilizator

Există o singură clasă de utilizator: utilizatorul final, care interacționează direct cu aplicația pentru a căuta rețete, a salva favorite, a crea planuri și a modifica contul. Se estimează un număr redus de utilizatori simultani, întrucât aplicația este locală. Numărul total de utilizatori este determinat de utilizarea aplicației pe fiecare sistem în parte.

#### Intrări

Intrările includ: nume de utilizator, parolă, adresă email (la înregistrare), text introdus pentru căutarea de rețete, selecție de ingrediente, selecția zilei pentru planificare. Datele sunt introduse prin câmpuri de tip TextBox, ComboBox și CheckedListBox în interfața Windows Forms.

#### Ieșiri

Ieșirile sunt sub formă de liste vizuale cu rețete, detalii despre rețete, liste de favorite și planuri zilnice. Acestea sunt afișate în ListView-uri sau panouri dinamice. Utilizatorii primesc mesaje de confirmare pentru salvare, eroare sau acțiuni nereușite.

### Proiectarea interfețelor cu utilizatorul

Interfața utilizatorului este proiectată într-un singur formular principal. La pornire, se afișează formularul de login, iar după autentificare interfața se transformă într-un tablou de control. Navigarea se face prin butoane: „Căutare Rețete”, „Contul Meu” și altele. Zonele de conținut sunt încărcate dinamic într-un panou central, fără a deschide formulare suplimentare.

## Scenarii de utilizare

Funcționalitatea aplicației este ilustrată printr-un set de scenarii de utilizare care surprind pașii tipici și așteptările utilizatorului în cadrul procesului de interacțiune cu sistemul. Aceste scenarii sunt concepute pentru a acoperi ciclul complet de utilizare a aplicației, începând de la accesul inițial și până la gestionarea personalizată a conținutului.

Un scenariu de bază pornește din momentul în care utilizatorul lansează aplicația de pe desktop. La deschidere, aplicația afișează interfața de autentificare, unde utilizatorul introduce numele de utilizator și parola în câmpurile dedicate. Dacă datele sunt valide, aplicația validează accesul și încarcă interfața principală, personalizată pentru utilizatorul curent. În cazul în care datele sunt greșite sau incomplete, sistemul răspunde prompt cu un mesaj de eroare care solicită reintroducerea informațiilor corecte.

Odată autentificat, utilizatorul este întâmpinat de o interfață organizată logic, unde poate selecta opțiunea „Căutare Rețete”. Aplicația îi oferă două alternative: fie căutare directă după denumirea unei rețete, fie introducerea unei liste de ingrediente disponibile în gospodărie. În ambele cazuri, aplicația interoghează baza de date locală și afișează o listă cu rețetele care corespund criteriilor de căutare. Fiecare rețetă afișată conține un rezumat care include titlul, timpul de preparare, nivelul de dificultate și numărul de porții.

După selectarea unei rețete, utilizatorul este redirecționat către o pagină detaliată, unde poate vizualiza lista completă a ingredientelor necesare, detalii nutriționale, tipul de dietă (normală, vegetariană, vegană, proteică) și pașii de preparare organizați secvențial. În partea inferioară a acestei pagini, utilizatorul poate apăsa pe butonul „Salvează la favorite”, moment în care rețeta este stocată în secțiunea personală.

Accesând ulterior secțiunea „Contul Meu”, utilizatorul poate vizualiza toate rețetele favorite salvate anterior, într-un format de listă simplu și clar. De aici poate relua lectura rețetei sau o poate include într-un planificator săptămânal, unde selectează ziua și ora la care dorește să o prepare. Tot din această secțiune, utilizatorul poate actualiza parola contului sau se poate deloga din aplicație.

Toate acțiunile utilizatorului sunt însoțite de validări riguroase, inclusiv verificarea câmpurilor obligatorii, restricții de formatare (ex. adrese email valide) și feedback vizual prin mesaje informative. În cazul în care intervin erori, cum ar fi lipsa rezultatelor la o căutare sau tentative de salvare duplicată în favorite, aplicația informează imediat utilizatorul prin mesaje clare și neintruzive. Astfel, întregul parcurs este gândit să ofere o experiență coerentă, predictibilă și sigură pentru utilizatorul final.

## Proiectare de detaliu

### Proiectare hardware de detaliu

Aplicația necesită un sistem minim cu procesor dual-core, minim 2 GB RAM și spațiu de stocare de aproximativ 100 MB. Nu există cerințe hardware speciale. Toate funcționalitățile rulează pe arhitectură locală, fără conectivitate rețea sau module hardware externe.

### Proiectare software de detaliu

Serviciile software sunt organizate în clase C# corespunzătoare modulelor aplicației. Clasa principală gestionează autentificarea și transformarea interfeței. Clase auxiliare includ manipularea bazei de date, gestionarea sesiunii și logica pentru favorite și planificator. Structura este modulară, cu metode clare și comentarii. Interfața este gestionată prin evenimente și apeluri de metode asincrone interne.

### Proiectare detaliată de securitate

Autentificarea se face prin verificarea locală a credentialelor din baza de date. Nu este implementată criptarea parolelor, dar sistemul validează corect structura și previne accesul neautorizat. Datele nu sunt transmise în rețea, eliminând riscul de interceptare. Aplicația poate fi extinsă ulterior pentru criptarea datelor sau jurnalizarea accesului.

### Proiectare de detaliu pentru performanța sistemului

Performanța sistemului este optimizată prin folosirea unei baze de date locale, cu interogări simple și rapide. Așteptările sunt de răspuns instantaneu la operații normale și timp de încărcare sub două secunde. Fiabilitatea este asigurată printr-un model simplu, fără concurență sau procese paralele. Nu sunt implementate mecanisme de backup, dar baza de date poate fi copiată manual pentru salvare.

### Proiectare detaliată a comunicațiilor interne (între componente)

Comunicarea între componentele aplicației se realizează local, prin apeluri directe între metode și clase. Evenimentele din interfața grafică sunt legate de logica aplicației prin funcții de tip handler. Nu există protocoale sau mesaje, întrucât aplicația este complet locală. Structura este simplă, cu datele transmise între clasele UI, sesiune și accesul la bază de date.

## Controale pentru verificarea integrității sistemului

Accesul la funcționalitățile aplicației este protejat printr-un mecanism de autentificare care validează identitatea utilizatorului înainte de a permite accesul la datele și opțiunile disponibile. Nicio funcționalitate legată de rețete, favorite, planificator sau cont nu este accesibilă în afara unei sesiuni autentificate, asigurând astfel un prim nivel de control al accesului. Datele unui utilizator sunt strict izolate și pot fi vizualizate sau modificate doar de către deținătorul acelui cont.

Sistemul implementează validări riguroase pentru toate câmpurile de introducere, acoperind atât prezența datelor (nepermiterea lăsării câmpurilor esențiale goale), cât și formatul acestora. De exemplu, adresele de email sunt verificate să conțină caractere specifice, iar parolele sunt supuse unei minime verificări de structură. Aceste controale contribuie la menținerea integrității datelor și previn salvarea de informații corupte sau incomplete în baza de date.

Fiecare operație importantă – cum ar fi adăugarea unei rețete la favorite, modificarea programului din planificator sau schimbarea parolei – este însoțită de mesaje vizuale de confirmare, astfel încât utilizatorul să primească feedback imediat și clar privind succesul sau eșecul acțiunii. În cazul apariției unei erori de sistem sau a unei condiții neașteptate (de exemplu, accesarea unei rețete care a fost ștearsă între timp), aplicația răspunde printr-un mesaj descriptiv care informează utilizatorul și previne întreruperea funcționării aplicației.

Deși în versiunea actuală nu este implementat un mecanism de auditare a activităților sau jurnalizare a accesului, arhitectura aplicației este gândită să permită adăugarea acestui tip de funcționalitate într-o versiune ulterioară. Aceasta ar putea include înregistrarea operațiunilor sensibile, identificarea terminalului, ora și tipul acțiunii efectuate pentru a îmbunătăți transparența și securitatea generală a aplicației.

Anexa A: Gestiunea modificărilor documentului

Tabel 1 – Înregistrarea modificărilor asupreaa documentului curent

| versiune | Data | Autorul/Deținătorul | Descriere |
| --- | --- | --- | --- |
| 1.0 | 02/05/2025 | Draghici Alin Ionut | Versiune inițială completă a documentului de proiectare |

Anexa B: Acronime

Tabel 2 - Acronime

| Acronim | Forma completă |
| --- | --- |
| UI | User Interface |
| DB | Database |
| SRS | Software Requirements Specification |