<Nume proiect/ acronim>

# Documentul de proiectare

Cuprins

[1. Introducere 1](#_Toc160527836)

[1.1 Scopul documentului 1](#_Toc160527837)

[2. Prezentare generală și abordări de proiectare 2](#_Toc160527838)

[2.1 Prezentare generală 2](#_Toc160527839)

[2.2 Presupuneri/ Constrângeri/ Riscuri 2](#_Toc160527840)

[2.2.1 Presupuneri 2](#_Toc160527841)

[2.2.2 Constrângeri 2](#_Toc160527842)

[2.2.3 Riscuri 3](#_Toc160527843)

[3. Considerațiii de proiectare 4](#_Toc160527844)

[3.1 Obiective și linii directoare (ghiduri) 4](#_Toc160527845)

[3.2 Metode de dezvoltare 4](#_Toc160527846)

[3.3 Strategii de arhitectură 4](#_Toc160527847)

[4. Arhitectura Sistemului și Proiectarea Arhitecturii 6](#_Toc160527848)

[4.1 Vedere logică 6](#_Toc160527849)

[4.2 Arhitectură hardware 6](#_Toc160527850)

[4.3 Arhitectură software 6](#_Toc160527851)

[4.4 Arhitectura informațiilor 7](#_Toc160527852)

[4.5 Arhitectura de comunicații interne 7](#_Toc160527853)

[4.6 Diagrama de arhitectură a sistemului 8](#_Toc160527854)

[5. Proiectarea sistemului 9](#_Toc160527855)

[5.1 Proiectarea bazei de date 9](#_Toc160527856)

[5.1.1 Obiecte de date și structuri de date rezultante 9](#_Toc160527857)

[5.1.2 Fișiere și baze de date 9](#_Toc160527858)

[5.2 Conversii de date 9](#_Toc160527859)

[5.3 Interfețe utilizator 10](#_Toc160527860)

[5.3.1 Intrări 10](#_Toc160527861)

[5.3.2 Ieșiri 10](#_Toc160527862)

[5.4 Proiectarea interfețelor cu utilizatorul 10](#_Toc160527863)

[6. Scenarii de utilizare 11](#_Toc160527864)

[7. Proiectare de detaliu 12](#_Toc160527865)

[7.1 Proiectare hardware de detaliu 12](#_Toc160527866)

[7.2 Proiectare software de deatliu 12](#_Toc160527867)

[7.3 Proiectare detaliată de securitate 13](#_Toc160527868)

[7.4 Proiectare de detaliu pentru performanța sistemului 14](#_Toc160527869)

[7.5 Proiectare detaliată a comunicațiilor interne (între componente) 14](#_Toc160527870)

[8. Controale pentru verificarea integrității sistemului 15](#_Toc160527871)

[Anexa A: Gestiunea modificărilor documentului 16](#_Toc160527872)

[Anexa B: Acronime 17](#_Toc160527873)

[Anexa C Documente la care se face referire 18](#_Toc160527874)

## Introducere

Instrucțiuni: Furnizați informații de identificare pentru sistemul existent și/sau propus sau situația pentru care se aplică Documentul de Proiectare a Sistemului (de exemplu, numele complet și acronimele proiectului de dezvoltare, sistemul sau situația existentă și propusă, după caz), și evoluția așteptată a documentului. De asemenea, descrieți orice considerații referitor la securitate sau confidențialitate asociate cu utilizarea acestui document..

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

*Instrucțiuni: Furnizați scopul SDD-ului. Acest document ar trebui să fie adaptat pentru a se potrivi nevoilor unui proiect specific.*

Prin acest livrabil se documentează și urmăresc informațiile necesare pentru a defini eficient arhitectura și designul sistemului, în scopul de a oferi echipei de dezvoltare îndrumare asupra arhitecturii sistemului ce urmează să fie dezvoltat.

Documentele de proiectare sunt produse incremental și iterativ pe parcursul ciclului de viață al dezvoltării sistemului, în funcție de circumstanțele particulare ale proiectului de tehnologie informațională (IT) și de metodologia de dezvoltare a sistemului utilizată.

Publicul țintă este managerul de proiect, echipa de proiect și echipa de dezvoltare. 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.

## Prezentare generală și abordări de proiectare

Această secțiune descrie principiile și strategiile care vor fi utilizate ca ghiduri în momentul proiectării și implementării sistemului.

### Prezentare generală

Instrucțiuni: Prezentați succint contextul sistemului și abordarea sau organizarea de bază a proiectării. Oferiți o prezentare generală a arhitecturilor sistemului și software-ului și a obiectivelor de proiectare.

### Presupuneri/ Constrângeri/ Riscuri

#### Presupuneri

Instrucțiuni: Descrieți orice presupuneri sau dependențe legate de sistem, software și utilizarea sa. Acestea pot viza probleme precum: software sau hardware asociat, sisteme de operare, caracteristicile utilizatorilor finali și posibilele și/sau probabilele modificări ale funcționalității.

#### Constrângeri

Instrucțiuni: Descrieți orice limitări sau constrângeri globale care au un impact semnificativ asupra designului hardware-ului, software-ului și/sau comunicațiilor **s**istemului și descrieți impactul asociat. Astfel de constrângeri pot fi impuse de oricare dintre următoarele (lista nu este exhaustivă):

* Mediu hardware sau software
* Mediu utilizator final
* Disponibilitatea sau volatilitatea resurselor
* Conformitatea cu standardele
* Cerințe de interoperabilitate
* Cerințe de interfață/protocol
* Cerințe de licențiere
* Cerințe pentru depozitarea și distribuția datelor
* Cerințe de securitate (sau alte reglementări similare)
* Limitări ale capacității de memorie sau ale altor resurse
* Cerințe de performanță
* Comunicații de rețea
* Cerințe de verificare și validare (testare)
* Alte mijloace de a aborda obiectivele de calitate
* Alte cerințe descrise în Documentul de Cerințe

#### Riscuri

Instrucțiuni: Descrieți orice riscuri asociate cu designul sistemului și strategiile propuse de reducere a acestora.

## Considerațiii de proiectare

Instrucțiuni: Descrieți problemele care trebuie abordate sau rezolvate înainte de a încerca să elaborați o soluție de design completă.

### Obiective și linii directoare (ghiduri)

Instrucțiuni: Descrieți orice obiective, linii directoare, principii sau priorități care domină sau încorporează designul sistemului și al software-ului. Exemple de astfel de obiective ar putea fi: accentul pe viteză versus utilizarea memoriei; sau utilizare, aspect sau "să se simtă" similar cu un produs existent. Linii directoarele includ ghiduri și convenții de codare. Pentru fiecare ghid, descrieți motivul, cu excepția cazului în care este implicit evident. Descrieți orice politici și/sau strategii de design care nu au implicații arhitecturale majore (adică nu afectează semnificativ organizarea generală a sistemului și structurile sale de nivel înalt), dar care totuși afectează detaliile interfeței și/sau implementarea diferitelor aspecte ale sistemului (de exemplu, alegerea unui produs specific de utilizat).

### Metode de dezvoltare

Instrucțiuni: Descrieți succint metoda sau abordarea folosită pentru designul sistemului și al software-ului (de exemplu, structurat, orientat pe obiecte, prototipare, J2EE, UML, XML, etc.). Dacă una sau mai multe metode formale/publicate au fost adoptate sau adaptate, includeți o referință către o descriere mai detaliată a acestor metode. Dacă au fost luate în considerare mai multe metode serios, menționați fiecare astfel de metodă, împreună cu o explicație succintă a motivului pentru care întreaga metodă sau o parte din ea a fost utilizată sau nu. Descrieți orice contingente care ar putea apărea în designul sistemului și al software-ului care ar putea schimba direcția de dezvoltare. Posibilitățile includ lipsa acordurilor de interfață cu agențiile externe sau arhitecturile instabile în momentul în care documentul este pregătit. Abordați orice soluții de rezervă posibile sau planuri alternative.

### Strategii de arhitectură

Instrucțiuni: Descrieți orice decizii și/sau strategii de design care afectează organizarea generală a sistemului și structurile sale de nivel superior. Aceste strategii ar trebui să ofere o înțelegere a principalelor abstractizări și mecanisme utilizate în arhitectura sistemului. Descrieți raționamentul utilizat pentru fiecare decizie și/sau strategie (posibil referindu-vă la obiectivele și principiile de design anterior menționate) și modul în care au fost echilibrate sau compensat obiectivele sau prioritățile de designAtunci când descrieți o decizie de design, discutați orice alte alternative semnificative care au fost luate în considerare și motivele pentru respingerea lor (precum și motivele pentru acceptarea alternativei în cele din urmă alese).

Exemple de decizii de design ar putea include (dar nu sunt limitate la) următoarele:

* • Utilizarea unui anumit tip de produs (limbaj de programare, bază de date, bibliotecă, produs comercial disponibil pe piață etc.)
* • Reutilizarea componentelor software existente pentru implementarea diferitelor părți/caracteristici ale sistemului
* • Planuri viitoare pentru extinderea sau îmbunătățirea software-ului
* • Paradigmele interfeței utilizatorului (sau modelele de intrare și ieșire ale sistemului)
* • Paradigmele interfeței hardware și/sau software
* • Detectarea și recuperarea erorilor
* • Politicile de gestionare a memoriei
* • Bazele de date externe și/sau gestionarea și persistența datelor
* • Date distribuite sau control asupra unei rețele
* • Abordări generalizate pentru control
* • Concurgență și sincronizare
* • Mecanisme de comunicare
* • Managementul altor resurse

## Arhitectura Sistemului și Proiectarea Arhitecturii

Arhitectura sistemului software pentru gestionarea și prelucrarea comenzilor într-un restaurant, cu aplicație web pentru digitalizarea procesului de plasare a comenzilor, utilizează tehnologii moderne precum HTML, CSS, JavaScript, React și Firebase. Interfața cu clientul, dezvoltată în React, oferă o experiență interactivă utilizatorilor, inclusiv funcționalități precum vizualizarea meniului, adăugarea produselor în coș și finalizarea comenzilor. Această interfață include acum și o pagină de autentificare, gestionată de componente React, care solicită utilizatorilor să introducă credențialele (nume de utilizator și parolă).

Serverul aplicației, construit în React și Firebase, integrează serviciul Firebase Authentication pentru gestionarea procesului de autentificare a utilizatorilor. Funcționalitățile avansate ale aplicației, precum plasarea comenzilor și vizualizarea istoricului, sunt accesibile doar utilizatorilor autentificați. Baza de date, Firebase Cloud Firestore, a fost extinsă pentru a personaliza accesul la date în funcție de utilizatorii autentificați, astfel încât fiecare client să aibă acces doar la propriul istoric de comenzi și informații personale.

În ceea ce privește sistemul de procesare a plăților, acesta integrează Firebase Authentication pentru a asigura securitatea și accesul doar pentru utilizatorii autentificați. Utilizează servicii precum Firebase Functions pentru gestionarea securizată și procesarea plăților, în concordanță cu autentificarea efectuată.

Sistemul de comunicare cu bucătăria, bazat pe Firebase și React, actualizează în timp real starea comenzilor în funcție de feedback-ul primit de la bucătărie. În plus, permite vizualizarea stării comenzilor anterioare doar pentru utilizatorii autentificați. Aceste adăugari de funcționalități de autentificare îmbunătățesc securitatea și personalizarea experienței utilizatorilor, asigurând că accesul la date și funcționalități este adaptat identității fiecărui utilizator.

Instrucțiuni: Descrieți arhitectura sistemului, modul în care aplicația interacționează cu alte aplicații. Nu este neapărat nevoie să detaliați cum funcționează aplicația în sine, ci mai degrabă cum datele corespunzătoare sunt corect transferate între componente. Furnizați o prezentare generală a modului în care funcționalitatea și responsabilitățile sistemului au fost partitionate și apoi atribuite sub-sisteme sau componente. Nu intrați prea mult în detalii despre componentele individuale în această secțiune. O secțiune ulterioară a documentului va furniza descrieri detaliate ale componentelor. Scopul principal aici este de a obține o înțelegere generală a modului și motivelor pentru care sistemul a fost descompus și cum părțile individuale lucrează împreună pentru a furniza funcționalitatea dorită.

Descrieți cum sistemul a fost dezmembrat în componente/subsistemele sale (identificând fiecare componentă/subsistem de nivel superior și rolurile/responsabilitățile atribuite acestuia). Descrieți modul în care componentele de nivel superior colaborează între ele pentru a obține rezultatele dorite. Furnizați o anumită justificare pentru alegerea acestei decompuneri particulare a sistemului (poate discutați alte decompuneri propuse și motivele pentru care au fost respinse).

Sistemul software pentru gestionarea și prelucrarea comenzilor într-un restaurant, cu aplicație web pentru digitalizarea procesului de plasare a comenzilor, a fost dezmembrat în mai multe componente sau subsisteme, fiecare cu roluri și responsabilități bine definite, contribuind astfel la funcționarea eficientă a întregului sistem.

Unul dintre principalele subsisteme este reprezentat de Interfața cu Clientul (Client-Side Component), responsabilă pentru interacțiunea directă cu utilizatorii. Aceasta include funcționalități precum afișarea meniului, adăugarea produselor în coș și finalizarea comenzilor. Colaborează strâns cu serverul aplicației pentru a transmite comenzile și pentru a primi actualizări privind starea meselor.

La nivel de server, avem Serverul Aplicației (Server-Side Component), care își asumă sarcini cruciale precum procesarea comenzilor, validarea datelor și gestionarea stării meselor și a meniului. Interacționează în mod activ cu baza de date pentru a accesa și actualiza informațiile necesare, precum și cu sistemul de procesare a plăților pentru finalizarea tranzacțiilor financiare.

Un alt element esențial al sistemului este Baza de Date (Data Storage Component), care asigură stocarea persistentă a informațiilor despre meniu, starea meselor și istoricul comenzilor. Aceasta este accesată și actualizată de către server pentru a asigura consistența datelor în întregul sistem.

Alegerea acestei decompuneri a sistemului a fost motivată de dorința de a separa eficient responsabilitățile și de a facilita dezvoltarea modulară și scalabilitatea. Această abordare permite gestionarea eficientă a interacțiunilor între componente, asigurând un grad optim de coeziune și cuplaj redus. Alte propuneri de decompunere au fost luate în considerare, dar această structură modulară a fost preferată pentru a susține extensibilitatea și adaptabilitatea sistemului la viitoarele evoluții.

Faceți uz de tiparele de proiectare ori de câte ori este posibil, fie pentru a descrie părți ale arhitecturii (în format de tipar), fie pentru a face referire la elementele arhitecturii care le utilizează. Furnizați o justificare pentru alegerea unui algoritm sau idiom de programare (sau tipar de proiectare) pentru a implementa părți ale funcționalității sistemului.

În cadrul arhitecturii sistemului software pentru gestionarea comenzilor în restaurant, există oportunități de aplicare a tiparelor de proiectare pentru a îmbunătăți structura și eficiența codului. Un astfel de exemplu ar putea fi utilizarea tiparului de proiectare Observator (Observer Pattern) pentru gestionarea actualizărilor stării comenzilor în timp real.

Observator (Observer Pattern) pentru Actualizările în Timp Real:

Un tipar Observator poate fi aplicat în implementarea sistemului de comunicare cu bucătăria și a interfeței cu clientul. Astfel, bucătăria și interfața cu clientul pot fi văzute ca obiecte observabile, iar componente precum starea comenzilor sau statusul preparării pot fi observatori. Atunci când starea comenzii se schimbă (de exemplu, o comandă este pregătită în bucătărie), observatorii (interfața cu clientul) sunt notificați, permițând actualizarea în timp real a informațiilor afișate către utilizator.

Justificare: Această alegere se bazează pe nevoia de a actualiza starea comenzilor în timp real, fără a depinde de interogări periodice către server. Observatorul oferă o modalitate eficientă și dezcuplată de a notifica diverse părți ale sistemului despre schimbări de stare, asigurând coeziunea și lăsând loc pentru extensibilitate.

Fabrica (Factory Pattern) pentru Crearea Componentelor Interfeței cu Clientul:

Pentru crearea interfeței cu clientul, poate fi utilizat un tipar de proiectare Fabrica, care permite crearea de obiecte concrete (de exemplu, butoane sau elemente de meniu) fără a specifica clasa exactă a obiectului care urmează să fie creat.

Justificare: Utilizarea tiparului Fabrica aduce beneficii în modularitatea și extensibilitatea codului, deoarece adăugarea sau schimbarea tipurilor de componente ale interfeței cu clientul se realizează fără a modifica codul existent. Prin intermediul unei fabrici, pot fi create și gestionate diverse obiecte ale interfeței, facilitând astfel mentenanța și dezvoltarea viitoare a aplicației.

Aceste alegeri de tipare de proiectare au fost făcute pentru a îmbunătăți structura sistemului, facilitând întreținerea, extensibilitatea și o separare clară a responsabilităților în cadrul aplicației.

Această secțiune conturează designul arhitecturii sistemului și hardware-ului.

### Vedere logică

Instrucțiuni: Introduceți orice vederi logice relevante sau furnizați o referință către locul în care sunt stocate.

### Arhitectură hardware

Instrucțiuni: Descrieți hardware-ul și organizația generală a sistemului, indicând dacă sistemul de procesare este distribuit sau centralizat. Identificați tipul, numărul și locația tuturor componentelor hardware, inclusiv serverele de prezentare, aplicație și date și orice dispozitive periferice (de exemplu, load balancers, acceleratori SSL, comutatoare, firewall-uri), cu o descriere succintă a fiecărui element și diagrame care arată conectivitatea între componente, împreună cu firewall-urile, porturile și benzile de rețea utilizate (de exemplu, banda de management). Includeți estimările resurselor pentru capacitatea procesorului, memoria, stocarea online și stocarea auxiliară.

### Arhitectură software

Instrucțiuni: Descrieți toate componentele software. Enumerați elemente logice precum componente (de exemplu, JSP în stratul de prezentare, JNDI în stratul de aplicație, EJB și JDBC în stratul de date), platforme de baze de date, limbaje de programare, compilatoare, utilitare, sisteme de operare, software de comunicații, instrumente de inginerie software asistate de calculator, produse software externe – software comercial disponibil pe piață, framework-uri open source etc., cu o scurtă descriere a funcției fiecărui element și a oricăror informații de identificare, cum ar fi producătorul, numărul versiunii, numărul și tipurile de licențe necesare etc., dacă este cazul. Identificați toate elementele de configurare a software-ului computerului și interfețele de programare a aplicațiilor (APIs), inclusiv numele, tipul, scopul și funcția pentru fiecare; interfețele, mesajele și protocoalele pentru acele elemente; și raționamentul pentru designul arhitectural al software-ului.

Includeți module de software care sunt funcții, subrutine sau clase. Utilizați diagrame ierarhice funcționale, diagrame de organizare structurată (de exemplu, diagrame de structură), sau diagrame orientate pe obiecte care arată diferitele niveluri de segmentare până la nivelul cel mai jos. Toate caracteristicile din diagrame ar trebui să aibă numere de referință și nume.

Dacă este necesar, descrieți cum o componentă a fost împărțită în subcomponente, precum și relațiile și interacțiunile dintre subcomponente. Continuați în cât mai multe niveluri/subsecțiuni de discuție cât este necesar pentru a oferi o înțelegere de nivel înalt a întregului sistem sau subsistem, lăsând detaliile pentru includerea într-o secțiune ulterioară a documentului. Includeți diagrame de flux de date și furnizează fluxul fizic al proceselor și datelor legate de fluxul logic al proceselor și datelor dezagregat la nivelul procesului primitiv (descriind cum fiecare intrare este procesată/transformată în ieșirea rezultată). Dacă există părți ale sistemului care au existat deja înainte de începerea acestui efort de dezvoltare, atunci descrieți doar relațiile și interacțiunile dintre părțile vechi și cele noi. Părțile preexistente care sunt modificate sau îmbunătățite trebuie descrise doar în măsura în care este necesar pentru a oferi o înțelegere suficientă a naturii modificărilor care se fac.

### Arhitectura informațiilor

Instrucțiuni: Descrieți informațiile care vor fi stocate în sistem (de exemplu, informații despre beneficiari, date despre cereri etc.). Identificați dacă vreuna dintre informații este informație cu caracter sensibil.

Identificați toate datele (precum și formatul datelor - hârtie, introducere manuală, date electronice) furnizate sistemului, precum și cine/ce furnizează datele.

### Arhitectura de comunicații interne

Instrucțiuni: Furnizați o descriere detaliată a rețelei de comunicații a sistemului, indicând arhitectura de comunicații implementată și modul în care componentele sistemului sunt conectate. Includeți descrieri ale echipamentelor necesare (de exemplu, hub-uri, routere, transmițătoare, module de comunicații, firewall-uri, porturi etc.). Furnizați o diagramă care să ilustreze fluxul de comunicații între componentele sistemului și subsisteme.

Includeți estimări ale resurselor necesare pentru capacitatea rețelei de comunicații (LAN și WAN) necesară pentru instalarea și executarea fiecărei aplicații pe fiecare platformă.

### Diagrama de arhitectură a sistemului

Instrucțiuni: Utilizând proiectarea hardware, software, de comunicații și de informații descrise mai sus, redați structura generală și integrată a sistemului.

## Proiectarea sistemului

### Proiectarea bazei de date

Instrucțiuni: Descrieți proiectarea tuturor fișierelor de sistem de gestionare a bazelor de date (DBMS) și a fișierelor non-DBMS asociate cu sistemul. Furnizați un dicționar de date cuprinzător care să conțină numele elementului de date, tipul, lungimea, sursa, regulile de validare, întreținerea (creare, citire, actualizare, ștergere (CRUD)), stocarea datelor, ieșirile, aliasurile și descrierea.

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

Instrucțiuni: Pentru fiecare obiect de date funcțional, specificați structura/ile de date care va/vor fi utilizată/e pentru a stoca și procesa datele. Descrieți orice structuri de date care sunt o parte importantă a sistemului, inclusiv structurile de date majore care sunt transmise între componente.

#### Fișiere și baze de date

Instrucțiuni: Creați un model de date fizic care descrie stocarea și manipularea datelor în cadrul arhitecturii sistemului. Descrieți structurile de fișiere și locațiile acestora.

##### Baze de date

Instrucțiuni: Furnizați proiectarea detaliată a fișierelor DBMS.

##### Fișiere non-DBMS

Instrucțiuni: Furnizați descrierea detaliată a tuturor fișierelor non-DBMS și includeți o descriere narativă a utilizării fiecărui fișier care identifică dacă fișierul este utilizat pentru intrare, ieșire sau ambele, și dacă fișierul este un fișier temporar. De asemenea, oferiți o indicație a modulului care citește și scrie fișierul și includeți structurile de fișiere (faceți referire la dicționarul de date). În funcție de caz, informațiile despre structura fișierului ar trebui să includă următoarele:

* Structurile de înregistrare, cheile de înregistrare sau indexele și elementele de date referite în înregistrări
* Lungimea înregistrării (lungime fixă sau maximă variabilă) și factorii de blocare
* Estimarea dimensiunii fișierului sau volumului de date din fișier
* Definiția frecvenței de actualizare a fișierului (dacă fișierul face parte dintr-un sistem bazat pe tranzacții online, furnizați numărul estimat de tranzacții per unitate de timp și media statistică, modul și distribuția acestor tranzacții.)
* Specificații de backup și recuperare

### Conversii de date

Instrucțiuni: Inserați orice documente care descriu conversiile de date necesare sau furnizați o referință către locul unde sunt stocate.

### Interfețe utilizator

Instrucțiuni: Furnizați o descriere a fiecărei clase de utilizatori sau roluri asociate cu sistemul. O clasă de utilizatori se diferențiază în funcție de modurile în care utilizatorii interacționează cu sistemul sau situația propusă. Factorii care disting o clasă de utilizatori includ responsabilitățile comune, nivelurile de competență, activitățile de lucru și modurile de interacțiune cu sistemul. În acest context, un utilizator este oricine interacționează cu sistemul propus, inclusiv utilizatori operaționali, personal de introducere a datelor, operatori de sistem, personal de suport operațional, cei care întrețin sistemul și formatori. Pentru fiecare clasă de utilizatori, furnizați estimări ale numărului total de utilizatori anticipați, un număr maxim de utilizatori simultani și numărul de utilizatori externi.

#### Intrări

Instrucțiuni: Furnizați o descriere a mijloacelor de intrare folosite de utilizator/operator pentru a furniza informații sistemului.

Arătați o mapare către fluxurile de date la nivel înalt (de exemplu, ecranele de introducere a datelor, cititoarele optice de caractere, scanerele de coduri de bare, etc.). Dacă este cazul, tipurile de înregistrări de intrare, structurile de fișiere și structurile de baze de date furnizate în secțiunea pentru Proiectarea datelor pot fi referite. Includeți definițiile elementelor de date sau faceți referire la dicționarul de date.

Furnizați aspectul tuturor ecranelor de intrare de date sau interfețe grafice (GUI-uri) (de exemplu, ferestre). Definiți toate elementele de date asociate fiecărui ecran sau GUI sau faceți referire la dicționarul de date. Furnizați criterii de editare pentru elementele de date, inclusiv valori specifice, interval de valori, obligatoriu/opțional, valori alfanumerice și lungime. Discutați controalele de introducere a datelor pentru a preveni trecerea peste editare. Discutați mesajele diverse asociate cu intrările utilizatorului/operatorului.

#### Ieșiri

Instrucțiuni: Ieșirile sistemului includ rapoarte, ecrane de afișare a datelor și interfețe grafice, rezultatele interogărilor, etc. Fișierele de ieșire descrise în secțiunea pentru Proiectarea Datelor pot fi referite. Ar trebui să fie furnizate următoarele, dacă este potrivit:

• Identificarea codurilor și numelor pentru rapoarte și ecranele de afișare a datelor

• Descrierea conținutului raportului și al ecranului (furnizați o reprezentare grafică a fiecărei structuri și definiți toate elementele de date asociate cu structura sau faceți referire la dicționarul de date)

• Descrierea scopului ieșirii, inclusiv identificarea utilizatorilor principali

• Descrierea oricăror restricții de acces sau considerații de securitate

### Proiectarea interfețelor cu utilizatorul

Instrucțiuni: Introduceți interfețele proiectate împreună cu descrierea lor.

## Scenarii de utilizare

Instrucțiuni: Descrieți funcționalitatea generală a sistemului din perspectiva utilizatorilor și furnizați un flux de execuție sau operațional al sistemului prin scenarii operaționale care oferă descrieri pas cu pas despre modul în care sistemul propus ar trebui să funcționeze și să interacționeze cu utilizatorii săi și interfețele externe într-un set dat de circumstanțe. Scenariile leagă împreună toate părțile sistemului, utilizatorii și alte entități prin descrierea modului în care interacționează, și pot fi folosite și pentru a descrie ce nu ar trebui să facă sistemul.

Scenariile de utilizare ar trebui să fie descrise pentru toate modurile operaționale, tranzacțiile și toate clasele de utilizatori identificate pentru sistemul propusFiecare scenariu ar trebui să includă evenimente, acțiuni, stimuli, informații și interacțiuni în măsura în care este necesar pentru a oferi o înțelegere cuprinzătoare a aspectelor operaționale ale sistemului propus.

## Proiectare de detaliu

Instrucțiuni: Furnizați informațiile necesare echipei de dezvoltare a sistemului pentru a construi și integra efectiv componentele hardware, pentru a codifica și integra componentele software, și pentru a interconecta segmentele hardware și software într-un produs funcțional.

În plus, adresați procedurile detaliate pentru combinarea pachetelor separate externe într-un singur sistem.

### Proiectare hardware de detaliu

Instrucțiuni: Furnizați suficiente informații detaliate despre fiecare dintre componentele hardware individuale pentru a construi și/sau achiziționa corect toate componentele hardware pentru sistem (sau pentru a integra articole COTS). Dacă există multe componente sau dacă documentația componentelor este extensivă, plasați-o într-un anex. Adăugați diagrame și informații suplimentare, dacă este necesar, pentru a descrie fiecare componentă și funcțiile sale adecvat. Ar trebui să fie urmate practici standard din industrie pentru specificațiile componentelor.

Pentru componente externe, identificați furnizorul specific și denumirile și numerele de model corespunzătoare. Includeți următoarele informații în proiectele detaliate ale componentelor, după caz:

* Cerințe de intrare de alimentare pentru fiecare componentă
* Impedanțe și stări logice ale semnalului
* Specificații ale conectorilor (serial/paralel, 11 pini, mascul/femelă, etc.)
* Specificații ale spațiului de memorie și/sau stocare
* Cerințe ale procesorului (viteză și funcționalitate)
* Reprezentare grafică care prezintă numărul de articole hardware (de ex., servere, dispozitive I/O, monitoare, imprimante etc.), și poziționarea relativă a componentelor între ele

Acestea pot fi introduse ca anexe ale acestui document

### Proiectare software de deatliu

Instrucțiuni: Furnizați o descriere detaliată pentru fiecare serviciu software de sistem care abordează următoarele atribute ale serviciului software. O mare parte din informațiile care apar în această secțiune ar trebui să fie conținute în antete/prologuri și secțiunile de comentarii ale codului sursă pentru fiecare componentă, subsistem, modul și subrutină. În cazul în care este așa, această secțiune poate consta în mare parte din referințe sau extrase din diagramele anotate și codul sursă. Orice diagrame sau extrase din codul sursă referite ar trebui furnizate la orice revizii de proiectare.

* Identificator serviciu - Identificatorul unic și/sau numele serviciului software
* Clasificare - Tipul de serviciu (de exemplu, aplicație, serviciu de date, etc.)
* Definiție - Scopul specific și semnificația semantică a serviciului
* Cerințe - Cerințele specifice funcționale sau nonfuncționale pe care serviciul le satisface
* Structuri de date interne - Structurile de date interne pentru serviciu
* Constraingeri - Orice relevante, presupuneri, limite sau constrângeri pentru serviciu (aceasta ar trebui să includă constrângerile de timp, de stocare sau de stare a serviciului și ar putea include reguli pentru interacțiunea cu serviciul (cuprinzând precondiții, postcondiții, invariante, alte constrângeri privind valorile de intrare sau ieșire și valorile locale sau globale, formatele datelor și accesul la date, sincronizarea, excepțiile, etc.))
* Compoziție - O descriere a utilizării și semnificației subserviciilor care fac parte din serviciu
* Utilizatori/Interacțiuni - O descriere a colaborării serviciului cu alte servicii (ce alte servicii utilizează această entitate? ce alte servicii utilizează această entitate (inclusiv orice efecte secundare pe care le-ar putea avea acest serviciu asupra altor părți ale sistemului)? acest lucru include metoda de interacțiune, precum și interacțiunea în sine. Designurile orientate pe obiecte ar trebui să includă o descriere a oricăror sub-clase, super-clase și meta-clase cunoscute sau anticipate)
* Procesare - O descriere precisă a modului în care serviciul își îndeplinește responsabilitățile (aceasta ar trebui să cuprindă o descriere a oricăror algoritmi utilizați; modificări ale stării; complexitate relevantă în timp sau spațiu; concurență; metode de creare, inițializare și curățare; și gestionarea condițiilor excepționale)
* Interfețe/Exporturi - Setul de servicii (resurse, tipuri de date, constante, subrutine și excepții) pe care serviciul le oferă (definiția sau declarația precisă a fiecărui astfel de element ar trebui să fie prezentă, împreună cu comentarii sau adnotări care descriu semnificațiile valorilor, parametrilor etc.; pentru fiecare element de serviciu descris, includeți sau furnizați o referință în discuția sa la o descriere a atributelor sale importante ale serviciului software (Identificator Componentă, Clasificare, Limbaj, Estimare SLOC, Definiție, Responsabilități, Cerințe, Structuri de Date Interne, Constraingeri, Compoziție, Utilizări/Interacțiuni, Resurse, Procesare și Interfețe/Exporturi))
* Raportare Design și Integrare - Dacă este inclus, furnizați detalii despre traficul și volumele de date.

### Proiectare detaliată de securitate

Instrucțiuni: Furnizați informații detaliate pentru fiecare dintre componentele individuale de securitate hardware. Specificați elementele de mai jos, după cum este necesar.

• Autentificare

• Autorizare

• Jurnalizare(arhivare) și auditare

• Criptare

• Utilizarea porturilor de rețea

• Detectare și prevenire a intruziunilor (în special dacă este găzduită într-un centru de date care nu aparține de CMS).

### Proiectare de detaliu pentru performanța sistemului

Instrucțiuni: Furnizați o reprezentare grafică cu informații detaliate pentru fiecare dintre componentele hardware individuale de performanță și fiabilitate, pentru a include elementele de mai jos:

• Cerințe/estimări de capacitate și volum

• Așteptări de performanță

• Cerințe de disponibilitate

• Proiectare de performanță pentru a îndeplini cerințele de capacitate

• Proiectare de fiabilitate pentru a îndeplini cerințele de disponibilitate

• Proiectare de backup, recuperare și arhivare

Identificați punctele unice de eșec și, dacă este relevant, descrieți proiectarea de disponibilitate ridicată (de exemplu, clustering).

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

Instrucțiuni: Dacă sistemul include mai mult de o componentă, este posibil să existe o cerință pentru comunicații interne pentru a schimba informații, a furniza comenzi sau a susține funcțiile de intrare/ieșire. Furnizați suficiente informații detaliate despre proiectarea comunicațiilor pentru a construi și/sau a achiziționa corect componentele de comunicații pentru sistem. Includeți următoarele informații în proiectările detaliate ale componentelor, după caz:

* Numărul de servere și clienți care urmează să fie incluși pe fiecare rețea de zonă
* Specificații pentru cerințele de sincronizare și control al busului
* Format(e) pentru datele schimbate între componente
* Reprezentare grafică a conectivității între componente, arătând direcția fluxului de date (dacă este aplicabil), și distanțele aproximative între componente (informațiile ar trebui să furnizeze suficient detaliu pentru a susține achiziționarea hardware-ului pentru finalizarea instalării într-o anumită locație)
* Topologia LAN

## Controale pentru verificarea integrității sistemului

Instrucțiuni: Furnizați specificații de proiectare pentru următoarele nivele minime de control și orice controale suplimentare adecvate sau necesare:

* Securitate internă pentru a restricționa accesul la datele critice doar pentru acele tipuri de acces necesare de către utilizatori/operatori
* Proceduri de audit pentru a îndeplini cerințele de control, raportare și perioade de reținere pentru rapoartele operaționale și de management
* Piste de auditare a aplicațiilor pentru a audita dinamic accesul la recuperare la datele critice designate
* Tabele standard care urmează să fie utilizate sau solicitate pentru validarea câmpurilor de date
* Procese de verificare pentru adăugarea, ștergerea sau actualizarea datelor critice
* Capacitatea de a identifica toate informațiile de auditare prin identificarea utilizatorului, identificarea terminalului de rețea, dată, oră și datele accesate sau modificate.

Anexa A: Gestiunea modificărilor documentului

Instrucțiuni: Furnizați informații despre modul în care dezvoltarea și distribuția documentului va fi controlată și urmărită. Utilizați tabelul de mai jos pentru a furniza numărul de versiune, data versiunii, autorul/deținătorul versiunii și o scurtă descriere a motivului pentru crearea versiunii revizuite.

Tabel 1 – Înregistrarea modificărilor asupreaa documentului curent

| versiune | Data | Autorul/Deținătorul | Descriere |
| --- | --- | --- | --- |
| <X.X> | <ZZ/LL/AAAA> | <nume autor> | <Descrierea modificării> |
| <X.X> | <ZZ/LL/AAAA> | <nume autor> | <Descrierea modificării> |
| <X.X> | <ZZ/LL/AAAA> | <nume autor> | <Descrierea modificării> |

Anexa B: Acronime

*Instrucțiuni: Furnizați o listă de acronime și traduceri literale asociate utilizate în cadrul documentului. Enumerați acronimele în ordine alfabetică folosind un format tabular, așa cum este ilustrat mai jos.*

Tabel 2 - Acronime

| Acronim | Forma completă |
| --- | --- |
| <Acronim> | <Forma completă> |
| <Acronim> | <Forma completă> |
| <Acronim> | <Forma completă> |

Anexa C Documente la care se face referire

*Instrucțiuni: Sintetizați relația acestui document cu alte documente relevante. Furnizați informații de identificare pentru toate documentele folosite pentru a ajunge la și/sau referite în acest document (de exemplu, documente conexe și/sau asociate, documente prealabile, documentație tehnică relevantă, etc.).*

Tabel 3 – Documente la care se facce referire

| Nume document | Locație sau URL | Dată emitere document |
| --- | --- | --- |
| < Nume document > | <Locație sau URL> | <ZZ/LL/AAAA> |
| < Nume document > | <Locație sau URL> | <ZZ/LL/AAAA> |
| < Nume document > | <Locație sau URL> | <ZZ/LL/AAAA> |