< Platformă web pentru o clinică medicală >

# Documentul de proiectare

Cuprins

[1. Introducere 3](#_Toc164090382)

[1.1 Scopul documentului 4](#_Toc164090383)

[2. Prezentare generală și abordări de proiectare 5](#_Toc164090384)

[2.1 Prezentare generală 5](#_Toc164090385)

[2.2 Presupuneri/ Constrângeri/ Riscuri 6](#_Toc164090386)

[2.2.1 Presupuneri 6](#_Toc164090387)

[2.2.2 Constrângeri 6](#_Toc164090388)

[2.2.3 Riscuri 8](#_Toc164090389)

[3. Considerațiii de proiectare 10](#_Toc164090390)

[3.1 Obiective și linii directoare (ghiduri) 11](#_Toc164090391)

[3.2 Metode de dezvoltare 12](#_Toc164090392)

[3.3 Strategii de arhitectură 13](#_Toc164090393)

[4. Arhitectura Sistemului și Proiectarea Arhitecturii 15](#_Toc164090394)

[4.1 Arhitectură hardware 15](#_Toc164090395)

[4.2 Arhitectură software 16](#_Toc164090396)

[4.3 Arhitectura informațiilor 16](#_Toc164090397)

[5. Proiectarea sistemului 19](#_Toc164090398)

[5.1 Proiectarea bazei de date 19](#_Toc164090399)

[5.1.1 Obiecte de date și structuri de date rezultante 19](#_Toc164090400)

[5.1.2 Fișiere și baze de date 20](#_Toc164090401)

[5.2 Interfețe utilizator 22](#_Toc164090402)

[5.2.1 Intrări 23](#_Toc164090403)

[5.2.2 Ieșiri 23](#_Toc164090404)

[5.3 Proiectarea interfețelor cu utilizatorul 24](#_Toc164090405)

[6. Scenarii de utilizare 27](#_Toc164090406)

[7. Proiectare de detaliu 29](#_Toc164090407)

[7.1 Proiectare software de deatliu 29](#_Toc164090408)

[7.2 Proiectare detaliată de securitate 31](#_Toc164090409)

[7.3 Proiectare de detaliu pentru performanța sistemului 31](#_Toc164090410)

[8. Controale pentru verificarea integrității sistemului 33](#_Toc164090411)

## Introducere

Numele proiectului: Platforma web pentru o clinică medicală

Sistemul existent:

Clinica medicală curentă funcționează pe un sistem tradițional de gestionare a pacienților și a serviciilor medicale, care include înregistrări pe hârtie și comunicații manuale între doctori și pacienți.

Sistemul propus:

Propunerea constă în dezvoltarea unei platforme web integrate care să automatizeze procesele de gestionare a pacienților, programări online, prescripții medicale digitale, sistem de plăți și abonamente online, precum și comunicarea între doctori și pacienți. Scopul este de a îmbunătăți accesul la servicii, eficiența administrativă și satisfacția pacienților.

Evoluția așteptată a documentului de proiectare a sistemului:

Documentul de proiectare a sistemului va fi actualizat pe măsură ce proiectul avansează prin fazele sale, de la analiza inițială a cerințelor până la implementarea finală și teste. Acesta va include specificații detaliate pentru toate modulele software, interfețele utilizator, precum și integrările cu alte sisteme.

Considerații de securitate și confidențialitate:

* Protecția datelor: Se va folosi HTTPS pentru toate comunicațiile din cadrul platformei pentru a proteja datele sensibile.
* Accesul la date: Accesul la informațiile medicale va fi strict controlat prin utilizarea autentificării bazate pe roluri. Fiecare categorie de utilizator (administrator, doctor, pacient) va avea acces doar la informațiile necesare pentru îndeplinirea funcțiilor sale.
* Audit și monitorizare: Sistemul va include capabilități de logare detaliată pentru a monitoriza toate acțiunile efectuate pe platformă, facilitând detectarea și investigarea oricăror activități suspecte sau neautorizate.
* Confidențialitate: Platforma va respecta politicile de confidențialitate ale clinicii, asigurând că pacienții sunt informați și își dau consimțământul cu privire la modul în care datele lor personale sunt utilizate și stocate.

### Scopul documentului

Scop general:

Scopul Documentului de Proiectare a Sistemului (SDD) este de a furniza o hartă detaliată pentru dezvoltarea platformei web a clinicii medicale, descrisă în documentul de specificație a cerințelor (SRS). Acest document vizează să detalieze arhitectura tehnică și designul sistemului necesar pentru a satisface cerințele funcționale și non-funcționale stabilite, asigurând că toate aspectele tehnice sunt adresate corespunzător pentru a atinge obiectivele proiectului.

Adaptare la proiectul specific:

SDD este specific adaptat pentru a răspunde nevoilor clinicii medicale în digitalizarea și optimizarea gestionării pacienților și serviciilor medicale. Acesta descrie în detaliu cum platforma va integra sisteme separate de login pentru admini, doctori și pacienți, va gestiona programările online, prescripțiile medicale digitale, și comunicațiile prin chat integrat, pe lângă alte funcționalități.

Componentele cheie ale SDD includ:

* Arhitectura sistemului: Detalierea structurii sistemului, inclusiv front-end, back-end, și baza de date, precum și tehnologiile specifice care vor fi utilizate.
* Descrierea detaliată a componentelor: Specificații pentru fiecare modul al platformei, inclusiv gestionarea autentificării, interfețele utilizator, și logica de afaceri pentru procesarea programărilor și prescripțiilor.
* Planuri de testare: Scenarii detaliate pentru testarea funcționalităților sistemului și asigurarea calității.

Rolul documentului în ciclul de dezvoltare:

SDD servește ca un ghid esențial pentru echipa de dezvoltare, arhitecți de sistem, și testeri, oferind detalii necesare pentru implementarea tehnică a soluțiilor propuse. Acest document asigură că toate părțile implicate în proiect au o înțelegere clară și comună a obiectivelor de proiectare și a așteptărilor, contribuind astfel la eficientizarea procesului de dezvoltare și reducerea riscurilor de deviație de la cerințele inițiale.

## Prezentare generală și abordări de proiectare

### Prezentare generală

Context:

Sistemul propus este o platformă web destinată clinicii medicale, scopul fiind modernizarea și optimizarea proceselor de gestionare a pacienților și serviciilor medicale. Aceasta include facilități de autentificare diferite pentru admini, doctori și pacienți, programări online detaliate, sisteme de abonamente și plăți online, prescripții medicale digitale, și un chat integrat pentru comunicare.

Abordarea de Proiectare:

Documentul indică o abordare centrată pe utilizator, cu funcționalități diverse care să satisfacă nevoile specific diferitelor clase de utilizatori: admini, doctori, și pacienți. Proiectarea va adopta o metodologie agile pentru a permite implementarea flexibilă și iterativă a cerințelor, cu actualizări pe baza feedback-ului continuu de la utilizatori.

Arhitectura sistemului și software-ului

Arhitectura sistemului:

* Front-end: Utilizarea tehnologiilor web moderne precum JavaScript, HTML, și CSS pentru a crea o interfață prietenoasă și accesibilă pentru toate tipurile de utilizatori.
* Back-end: Servere bazate pe PHP și baze de date relaționale pentru gestionarea datelor și interacțiunilor din sistem.
* Interfețe externe: Conexiuni securizate via HTTPS, cu API-uri RESTful pentru integrare cu alte sisteme și servicii externe, cum ar fi portaluri de plăți.

Arhitectura software-ului:

* Securitate: Măsuri robuste de securitate, inclusiv autentificare multi-factor și criptarea datelor pentru protejarea informațiilor medicale și personale ale pacienților.
* Modularitate: Design modular pentru a facilita întreținerea și extinderea ulterioară a sistemului fără impact major asupra componentelor existente.

Obiectivele de proiectare:

* Usabilitate: Asigurarea unei interfețe intuitive, cu acces facil la funcționalitățile necesare fiecărui tip de utilizator, și design responsive pentru diverse dispozitive.
* Securitate: Implementarea celor mai bune practici de securitate pentru protecția datelor.
* Performanță: Sistemul trebuie să fie capabil să gestioneze un volum mare de tranzacții și utilizatori simultani, asigurând un timp de răspuns rapid și fiabilitate.
* Scalabilitate: Capacitatea de a extinde sistemul cu noi funcționalități sau de a îmbunătăți cele existente în funcție de evoluția tehnologiei și a cerințelor de afaceri.

### Presupuneri/ Constrângeri/ Riscuri

#### Presupuneri

Presupuneri:

* Acces la internet: Se presupune că toți utilizatorii sistemului (admini, doctori, pacienți) au acces la o conexiune de internet stabilă și sigură pentru a accesa platforma.
* Disponibilitatea dispozitivelor: Utilizatorii au dispozitive (desktop, laptop, tabletă, smartphone) capabile să ruleze interfețele web moderne.
* Competențe tehnice: Utilizatorii finali au competențele necesare pentru a naviga și utiliza platforma fără dificultăți semnificative. De asemenea, se presupune o adaptabilitate rapidă la noile tehnologii introduse.
* Stabilitatea tehnologică: Tehnologiile selectate pentru dezvoltare (limbaje de programare, framework-uri, baze de date) vor rămâne stabile și suportate pe durata dezvoltării și a utilizării inițiale a sistemului.

#### Constrângeri

1. Mediu hardware și software:

* Constrângere: Platforma trebuie să fie compatibilă cu diverse dispozitive și sisteme de operare, de la desktop-uri la dispozitive mobile.
* Impact: Aceasta necesită un design responsiv și teste extinse pentru a asigura funcționalitatea pe diverse dispozitive și browsere.

1. Mediu utilizator final:

* Constrângere: Utilizatorii sistemului includ admini, doctori, și pacienți, fiecare cu niveluri diferite de competențe tehnologice.
* Impact: Interfața trebuie să fie intuitivă și accesibilă pentru a facilita o experiență utilizator optimă pentru toate categoriile de utilizatori.

1. Disponibilitatea sau volatilitatea resurselor:

* Constrângere: Resursele financiare pentru dezvoltare și întreținere pot fi limitate sau variabile.
* Impact: Pot exista limitări în alegerea tehnologiilor sau în scalabilitatea planificată a sistemului.

1. Conformitatea cu standardele:

* Constrângere: Necessitatea de a respecta reglementările privind protecția datelor personale și de sănătate.
* Impact: Design-ul sistemului trebuie să includă protocoale de securitate și criptare puternice, ceea ce poate crește complexitatea și costurile.

1. Cerințe de Interfață/Protocol:

* Constrângere: Protocoalele de comunicație trebuie să fie sigure și eficiente, utilizând standardizări cum ar fi HTTPS și RESTful API.
* Impact: Alegerea și implementarea acestor protocoale necesită expertiză tehnică și poate influența performanța sistemului.

1. Cerințe pentru depozitarea și distribuția datelor:

* Constrângere: Datele trebuie stocate și gestionate într-un mod care să respecte confidențialitatea și accesibilitatea.
* Impact: Necessită infrastructuri de stocare securizate și redundante pentru a preveni pierderile de date.

1. Cerințe de securitate:

* Constrângere: Sistemul trebuie să protejeze informațiile sensibile ale pacienților și să prevină accesul neautorizat.
* Impact: Design-ul trebuie să includă autentificare multi-factor, criptare și alte măsuri de securitate.

1. Cerințe de performanță:

* Constrângere: Sistemul trebuie să fie capabil să gestioneze un volum mare de utilizatori și tranzacții simultan.
* Impact: Arhitectura trebuie să fie scalabilă și optimizată pentru performanță.

1. Comunicații de rețea:

* Constrângere: Necessitatea unei rețele stabile și rapide pentru a suporta comunicațiile în timp real, cum ar fi chat-ul medical.
* Impact: Poate necesita investiții în hardware de rețea sau soluții de cloud.

1. Cerințe de verificare și validare (Testare):

* Constrângere: Toate componentele sistemului trebuie testate riguros pentru a asigura conformitatea cu cerințele.
* Impact: Procesul de testare poate fi timp-consuming și costisitor, dar este crucial pentru calitatea finală a sistemului.

#### Riscuri

1. Risc tehnologic:

* Descriere: Utilizarea tehnologiilor care devin rapid învechite sau sunt greu de integrat cu alte sisteme.
* Strategii de reducere: Alegerea tehnologiilor stabile și larg acceptate în industrie. Menținerea flexibilității în arhitectura sistemului pentru a permite actualizări ușoare și integrarea cu noi tehnologii pe măsură ce acestea devin disponibile.

1. Risc de securitate a datelor:

* Descriere: Posibilitatea breșelor de securitate care ar putea compromite datele sensibile ale pacienților.
* Strategii de reducere: Implementarea celor mai bune practici de securitate, inclusiv criptarea datelor în tranzit și la repaus, utilizarea autentificării multi-factor și realizarea de audituri de securitate regulate.

1. Risc de non-conformitate cu reglementările:

* Descriere: Nerespectarea reglementărilor privind protecția datelor.
* Strategii de reducere: Colaborarea cu experți în conformitate și reglementări pentru a asigura că toate aspectele sistemului sunt conforme cu legile aplicabile. Implementarea proceselor de conformitate din designul sistemului.

1. Risc de performanță sub așteptări:

* Descriere: Sistemul poate să nu îndeplinească așteptările de performanță sub sarcini grele sau la creșteri ale bazei de utilizatori.
* Strategii de reducere: Proiectarea unei arhitecturi scalabile și realizarea de teste de performanță înainte de lansare. Monitorizarea performanței sistemului post-lansare și ajustarea resurselor conform necesităților.

1. Risc de integrare deficientă:

* Descriere: Integrările cu sisteme externe pot eșua sau pot genera erori, afectând funcționalitatea generală.
* Strategii de reducere: Utilizarea API-urilor standardizate și verificarea compatibilității cu toate sistemele externe înainte de implementare. Realizarea de teste de integrare detaliate.

1. Risc de acceptare a utilizatorilor:

* Descriere: Rezistența utilizatorilor la adoptarea noii platforme din cauza complexității percepute sau a schimbării obiceiurilor de lucru.
* Strategii de reducere: Crearea unei interfețe intuitive și organizarea de sesiuni de formare pentru utilizatori. Colectarea feedback-ului utilizatorilor în fazele timpurii de dezvoltare și ajustarea designului conform acestui feedback.

1. Risc de depășire a bugetului și a termenelor:

* Descriere: Proiectul poate depăși estimările inițiale de cost și timp din cauza unor neașteptate în dezvoltare.
* Strategii de reducere: Planificare detaliată a proiectului și stabilirea unor bugete și termene realiste. Implementarea managementului eficient al proiectului pentru a monitoriza progresul și a identifica deviațiile timpurii.

## Considerațiii de proiectare

1. Evaluarea Tehnologică și Selecția Stack-ului Tehnologic

* Problemă: Alegerea tehnologiilor inadecvate poate duce la probleme de performanță, scalabilitate, sau securitate.
* Soluție: Realizarea unei evaluări tehnologice pentru a determina cel mai potrivit stack tehnologic care să răspundă cerințelor de funcționalitate și performanță ale platformei.

1. Conformitatea cu Reglementările și Standardele de Securitate

* Problemă: Necesitatea de a asigura că platforma este conformă cu toate reglementările relevante.
* Soluție: Consultarea cu experți în legal și securitate pentru a integra cerințele de conformitate și securitate din faza de design.

1. Gestionarea Datelor și Integrările Sistemului

* Problemă: Integrarea eficientă cu alte sisteme existente și gestionarea securizată a datelor.
* Soluție: Analiza sistemelor existente și proiectarea unei arhitecturi de date care să permită integrări fluide și sigure, folosind API-uri standardizate și protocoale de securitate robuste.

1. Capacitatea și Performanța Sistemului

* Problemă: Asigurarea că sistemul poate gestiona volumul de utilizatori și tranzacții așteptat fără degradări ale performanței.
* Soluție: Planificarea testelor de performanță și stres din timp, și proiectarea unei arhitecturi scalabile care poate fi extinsă pe măsură ce crește numărul de utilizatori.

1. Usabilitatea și Accesibilitatea Interfeței Utilizator

* Problemă: Crearea unei interfețe care este ușor de utilizat pentru toți utilizatorii, inclusiv pentru cei cu dizabilități.
* Soluție: Implementarea principiilor de design universal și realizarea testelor de utilizabilitate pentru a asigura că interfața este intuitivă și accesibilă.

1. Planificarea Resurselor și Bugetului

* Problemă: Alocarea insuficientă a resurselor și a bugetului poate duce la întârzieri și compromisuri în calitate.
* Soluție: Dezvoltarea unui plan detaliat de resurse și buget, inclusiv estimări realiste ale costurilor și alocarea adecvată a resurselor umane și tehnice.

1. Strategia de Testare și Validare

* Problemă: Asigurarea că toate componentele sistemului funcționează conform așteptărilor și cerințelor inițiale.
* Soluție: Dezvoltarea unui plan comprehensiv de testare, inclusiv testare automată și manuală, pentru a valida funcționalitatea și performanța înainte de lansare.

### Obiective și linii directoare (ghiduri)

Obiective de design:

* Performanță vs. utilizarea memoriei: Platforma va pune un accent mai mare pe performanță pentru a asigura o experiență rapidă și fluidă pentru utilizatori. Deși utilizarea eficientă a memoriei este importantă, prioritizarea timpului de răspuns rapid și a unor încărcări de pagină agile este crucială pentru satisfacția utilizatorului, având în vedere diversitatea de operații pe care le vor efectua utilizatorii.
* Consistență cu produsele existente: Interfața platformei va urmări să mențină un aspect și o senzație similară cu alte sisteme software utilizate în clinică, pentru a reduce curba de învățare a utilizatorilor și a integra în mod coeziv noua platformă cu infrastructura existentă.

Linii directoare și convenții de codare:

* Standardizarea codului: Adoptarea unor convenții de codare clare și utilizarea unor formate standard în cadrul echipei de dezvoltare. Aceasta va include utilizarea comentariilor adecvate, numirea corespunzătoare a variabilelor și funcțiilor, și structurarea modulară a codului. Motivul este de a asigura că codul este ușor de înțeles, de întreținut și de actualizat de către orice membru al echipei, actual sau viitor.
* Reutilizarea codului: Încurajarea reutilizării componentelor software unde este posibil pentru a reduce timpul de dezvoltare și a crește eficiența. Acest lucru va ajuta la menținerea consistenței în cadrul aplicației și va reduce riscul de erori.

Strategii de design fără implicații arhitecturale majore:

* Alegerea tehnologiilor și a produselor: Deciziile privind tehnologiile specifice (de exemplu, framework-uri, biblioteci) vor fi luate pe baza performanței dovedite și a compatibilității cu sistemele existente, precum și a suportului pe termen lung și a comunității active de dezvoltatori. Acest lucru asigură că platforma poate fi întreținută și extinsă eficient pe măsură ce tehnologiile evoluează și cerințele clinicii se schimbă.
* Designul interfeței utilizatorului: Fiecare aspect al UI va fi proiectat având în vedere accesibilitatea, cu etichete clare, fonturi lizibile și un contrast adecvat. În plus, interfața va fi testată pentru a asigura că este intuitivă și ușor de utilizat de către toate categoriile de utilizatori, inclusiv de către persoanele cu dizabilități.

### Metode de dezvoltare

Abordare de design:

Pentru designul sistemului și software-ul platformei web a clinicii medicale, am adoptat o abordare orientată pe obiecte, completată de utilizarea UML (Unified Modeling Language) pentru modelarea și documentarea designului arhitectural și funcțional. Aceasta permite o reprezentare vizuală clară a componentelor sistemului, interacțiunilor și relațiilor dintre acestea, facilitând astfel comunicarea și înțelegerea între membrii echipei de dezvoltare și stakeholderi.

Tehnologii și standarde utilizate:

* UML: Folosit pentru a crea diagrame de cazuri de utilizare, diagrame de activitate și diagrame de secvență, care ajută la vizualizarea fluxurilor de proces și a interacțiunilor utilizator în cadrul sistemului.
* XML: Utilizat pentru configurarea și integrarea diverselor componente ale sistemului, oferind un format standardizat pentru schimbul de date între diferite module și sisteme externe.
* Prototipare: Implementată în fazele inițiale pentru a obține feedback rapid de la utilizatori și pentru a valida cerințele funcționale înainte de dezvoltarea completă.

Evaluarea metodelor:

Am evaluat și alte abordări, inclusiv dezvoltarea structurată și utilizarea metodelor agile de dezvoltare. Metoda agile a fost parțial integrată în procesul nostru pentru a îmbunătăți flexibilitatea și adaptabilitatea în fazele de dezvoltare, permițând cicluri rapide de feedback și iterare.

Contingențe și planuri alternative:

* Lipsa acordurilor de interfață cu agențiile externe: În cazul în care acordurile pentru integrarea cu sisteme externe întârzie sau sunt incomplete, vom implementa simulări temporare ale acestor sisteme pentru a permite continuarea testării și dezvoltării. Aceasta asigură că progresul în dezvoltarea sistemului nu este împiedicat.
* Arhitecturi instabile: Dacă arhitectura propusă se dovedește a fi instabilă în timpul testelor inițiale, vom reevalua și ajusta designul arhitectural, posibil adoptând noi tehnologii sau abordări care oferă o mai mare stabilitate.
* Planuri de rezervă: În cazul în care tehnologiile alese inițial nu satisfac cerințele de performanță sau securitate, suntem pregătiți să evaluăm și să integram alternative mai robuste.

### Strategii de arhitectură

1. Utilizarea limbajului de programare și a bazei de date

Decizie: Alegerea PHP pentru backend și MySQL pentru baza de date.

Raționament: PHP este larg utilizat în dezvoltarea web datorită flexibilității și suportului extensiv pentru diferite tipuri de baze de date. MySQL este ales pentru robustețea și scalabilitatea sa, fiind compatibil cu PHP, ceea ce facilitează integrarea și dezvoltarea rapidă.

Alternative considerate: Node.js pentru backend și MongoDB ca bază de date non-relațională au fost evaluate, dar PHP și MySQL au fost preferate pentru maturitatea și stabilitatea lor în aplicații web comerciale.

1. Reutilizarea componentelor software existente

Decizie: Integrarea bibliotecilor și framework-urilor existente.

Raționament: Reutilizarea componentelor software reduce timpul de dezvoltare și costurile asociate. Framework-uri precum Laravel (pentru PHP) sunt utilizate pentru a îmbunătăți securitatea și pentru a simplifica implementarea funcționalităților complexe.

Alternative considerate: Dezvoltarea de componente personalizate a fost luată în considerare, dar reutilizarea a fost preferată pentru eficiența costurilor și timpului.

1. Planuri viitoare pentru extinderea sau îmbunătățirea software-ului

Decizie: Proiectare modulară pentru facilitarea extensiei și îmbunătățirilor viitoare.

Raționament: Designul modular permite adăugarea de noi funcționalități sau îmbunătățirea celor existente fără a perturba structura de bază a sistemului, asigurând astfel scalabilitatea și flexibilitatea pe termen lung.

Alternative considerate: Un design monolitic a fost luat în considerare, dar a fost respins datorită dificultăților de întreținere și scalare.

1. Paradigmele interfeței utilizatorului

Decizie: Utilizarea unui design responsiv și accesibil.

Raționament: Un design responsiv asigură că platforma poate fi utilizată eficient pe o varietate de dispozitive, iar accesibilitatea este crucială pentru a servi toți utilizatorii, inclusiv pe cei cu dizabilități.

Alternative considerate: O aplicație dedicată pentru mobile a fost evaluată, dar un design web responsiv a fost ales pentru a reduce costurile și complexitatea dezvoltării multiplelor aplicații native.

1. Gestionarea și persistența datelor

Decizie: Utilizarea unui model de date relaționale și implementarea de backup-uri automate.

Raționament: Un model relațional facilitează interogările complexe și relațiile dintre date, esențiale pentru istoricul medical și administrarea clinicii. Backup-urile automate asigură integritatea și disponibilitatea datelor.

Alternative considerate: Soluții de stocare non-relațională au fost considerate, dar complexitatea relațiilor dintre date în contextul medical a favorizat bazele de date relaționale.

## Arhitectura Sistemului și Proiectarea Arhitecturii

Componente principale ale sistemului:

* Interfața cu utilizatorul (UI): Responsabilă pentru prezentarea informațiilor utilizatorului și colectarea inputurilor de la acesta. Implementată folosind tehnologii web moderne care asigură un design responsiv și accesibil.
* Serverul de aplicații (Logică de afaceri): Aici sunt procesate cererile, se execută logica specifică aplicației și se gestionează tranzacțiile. Această componentă asigură că datele utilizatorilor sunt procesate corect și eficient.
* Baza de date: Stochează toate datele necesare aplicației, inclusiv datele utilizatorilor, istoricul medical, programările și informațiile de autentificare. Integritatea și securitatea datelor sunt prioritare în acest strat

Interacțiunea cu alte aplicații:

Platforma este proiectată să interacționeze cu diverse alte sisteme externe, cum ar fi gateway-uri de plată pentru procesarea tranzacțiilor financiare și sisteme de informații medicale externe pentru acces la istoricul medical al pacienților. Comunicarea se realizează prin intermediul API-urilor RESTful, care asigură un mecanism standardizat și securizat pentru schimbul de date.

Colaborarea între componente:

Componentele sistemului colaborează pentru a procesa cererile utilizatorilor. De exemplu, o cerere de programare a unei consultații este captată de interfața cu utilizatorul, transmisă serverului de aplicații pentru procesare, și rezultatul este stocat în baza de date. Serverul de aplicații răspunde apoi UI-ului care actualizează afișarea pentru utilizator.

### Arhitectură hardware

Componentele hardware

Servere de prezentare:

Descriere: Aceste servere gestionează interfața utilizator și prelucrează cererile clientului, trimițându-le către serverele de aplicații.

Servere de aplicație:

Descriere: Aceste servere procesează cererile de la serverele de prezentare, execută operațiuni de business și accesează baza de date.

Servere de date:

Descriere: Serverele sunt optimizate pentru operațiuni de bază de date cu acces rapid la stocare.

Dispozitive periferice:

Load balancers: Utilizate pentru a distribui traficul de rețea între servere și a optimiza performanța și disponibilitatea.

Acceleratori SSL: Hardware dedicat pentru accelerarea criptării și decriptării SSL, îmbunătățind securitatea și performanța.

Comutatoare și firewall-uri: Asigură conectivitatea internă și protejează rețeaua de accesuri neautorizate.

### Arhitectură software

Componente ale stratului de prezentare:

HTML/CSS/JavaScript si framework-uri ale acestora: Aceste tehnologii standard sunt utilizate pentru a construi interfețele cu utilizatorul, oferind o experiență bogată și interactivă.

Componente ale stratului de date:

MySQL: Sistemul de gestionare a bazelor de date relaționale, utilizat pentru stocarea și interogarea datelor pacienților, programărilor și informațiilor medicale.

Instrumente de inginerie software:

Git: Sistem de control al versiunilor utilizat pentru gestionarea codului sursă al aplicației.

### Arhitectura informațiilor

Tipuri de date și formatul Lor:

Date despre pacienți:

Descriere: Informații personale ale pacienților, inclusiv nume, adresă, număr de telefon, e-mail, date de naștere și istoric medical.

Sensibilitate: Aceste informații sunt cu caracter sensibil, deoarece implică date personale și de sănătate.

Format: Datele sunt colectate electronic, fie prin formulare online la înregistrare, fie prin introducere manuală de către personalul clinic.

Programările pacienților:

Descriere: Detalii despre programările făcute de pacienți, inclusiv data, ora și medicul alocat.

Sensibilitate: Informații cu caracter confidențial, necesitând protecție adecvată.

Format: Informațiile sunt generate și stocate electronic prin sistemul de gestionare a programărilor al platformei.

Prescripții medicale:

Descriere: Detalii despre medicamentele prescrise de medici, inclusiv numele medicamentului, dozajul și instrucțiunile de utilizare.

Sensibilitate: Aceste date sunt sensibile, implicând informații despre tratamentul medical al pacienților.

Format: Datele sunt generate de medici și introduse în sistem electronic.

Date financiare:

Descriere: Informații despre plățile efectuate de pacienți, inclusiv detaliile tranzacțiilor, metoda de plată și istoricul facturilor.

Sensibilitate: Informații financiare sensibile care necesită securitate înaltă.

Format: Colectate și stocate electronic prin interfața sistemului cu gateway-uri de plată și module financiare.

Interacțiuni și comunicări:

Descriere: Loguri ale comunicațiilor între pacienți și medici, inclusiv mesaje din chat și note medicale.

Sensibilitate: Conțin informații cu caracter privat între medic și pacient.

Format: Informațiile sunt stocate electronic, capturate automat de sistemul de mesagerie al platformei.

Sursa datelor:

Pacienții: Majoritatea datelor personale și a programărilor sunt furnizate direct de pacienți prin interfața utilizatorului a platformei.

Medici și personal medical: Informații medicale, prescripții și note despre starea pacienților sunt introduse de medici și personalul clinic.

Sisteme externe: Datele financiare pot fi furnizate prin intermediul procesatorilor de plăți externi integrati în sistem.

## Proiectarea sistemului

### Proiectarea bazei de date

A screenshot of a computer

Description automatically generated

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

**Tabela: programari**

id: int(11)

id\_medic: int(11)

id\_pacient: int(11)

data: date

ora: time

durata: int(11)

status: varchar(50)

**Tabela: user\_form**

id: int(255)

name: varchar(255)

email: varchar(255)

password: varchar(255)

user\_type: varchar(255)

**Tabela: servicii**

id: int(11)

nume: varchar(255)

descriere: text

imagine: varchar(255)

**Tabela: medici**

id: int(11)

nume: varchar(255)

specialitate: varchar(255)

imagine: varchar(255)

**Tabela: reviewuri**

id: int(11)

nume: varchar(255)

review: text

data\_review: date

aprobare: tinyint(1)

specializare: varchar(255)

#### Fișiere și baze de date

**Fișierul de bază de date:**

Nume: clinica\_db

**Tabelul Programări:**

Nume: programari

Atribute:

id: int(11), cheie primară

id\_medic: int(11), cheie străină referință la tabelul medici

id\_pacient: int(11), cheie străină referință la tabelul user\_form

data: date

ora: time

durata: int(11)

status: varchar(50)

Locație: În fișierul de bază de date clinica\_db

**Tabelul User\_Form:**

Nume: user\_form

Atribute:

id: int(255), cheie primară

name: varchar(255)

email: varchar(255)

password: varchar(255)

user\_type: varchar(255)

Locație: În fișierul de bază de date clinica\_db

**Tabelul Servicii:**

Nume: servicii

Atribute:

id: int(11), cheie primară

nume: varchar(255)

descriere: text

imagine: varchar(255)

Locație: În fișierul de bază de date clinica\_db

**Tabelul Medici:**

Nume: medici

Atribute:

id: int(11), cheie primară

nume: varchar(255)

specialitate: varchar(255)

imagine: varchar(255)

Locație: În fișierul de bază de date clinica\_db

**Tabelul Reviewuri:**

Nume: reviewuri

Atribute:

id: int(11), cheie primară

nume: varchar(255)

review: text

data\_review: date

aprobare: tinyint(1)

specializare: varchar(255)

Locație: În fișierul de bază de date clinica\_db

##### Fișiere non-DBMS

Fișiere de Configurare

* Utilizare: Intrare
* Descriere: Stochează parametrii de configurare necesari pentru inițializarea și operarea sistemului, cum ar fi conexiuni la baza de date, setări de rețea și configurări de securitate.

Fișiere Log (Jurnale)

* Utilizare: Ieșire (pot fi și de intrare în cazul analizei logurilor)
* Descriere: Înregistrează evenimentele sistemului, erorile și alte informații operaționale critice.

### Interfețe utilizator

1. Pacienții (Utilizatorii Externi):

Responsabilități: Programări pentru consultații, vizualizarea informațiilor personale și medicale, comunicarea cu medicii și personalul clinic.

1. Medicii:

Responsabilități: Revizuirea programărilor, acordarea retetelor, comunicarea cu pacienții.

1. Personalul administrativ:

Responsabilități: Introducerea datelor pacienților si a medicilor, gestionarea programărilor, asigurarea suportului administrativ.

#### Intrări

Ecrane de introducere a datelor:

* Utilizatori: Ecrane web formate din formulare de introducere pentru înregistrarea și gestionarea profilurilor pacienților, programările la medici, și feedback-ul post-consultație.
* Medici: Interfețe pentru introducerea detaliilor consultațiilor, prescripțiilor și actualizări ale istoricului medical al pacienților.
* Personal administrativ: Formulare pentru gestionarea programărilor, facturarea serviciilor și administrarea detaliilor contului pacientului.

Ecranele de introducere a datelor sunt direct conectate la baza de date a sistemului, unde informațiile sunt stocate și procesate.

Sistemul furnizează mesaje de eroare clare și sugestive în cazul în care datele introduse nu trec validările, ghidând utilizatorul pentru corectarea informațiilor. De asemenea, mesaje de confirmare sunt afișate la completarea cu succes a tranzacțiilor sau actualizărilor.

#### Ieșiri

1. Raportul de programări zilnice:

Scopul ieșirii: Acest raport este destinat medicilor și personalului administrativ pentru a planifica și coordona activitățile zilnice ale clinicii.

1. Ecranul de istoric medical al pacientului:

Scopul ieșirii: Acest ecran ajută medicii să acceseze rapid istoricul medical al pacienților pentru o mai bună evaluare și tratament.

1. Dashboard-ul de statistici și tendințe:

Scopul ieșirii: Acest dashboard este destinat adminului pentru a monitoriza și îmbunătăți operațiunile clinice.

### Proiectarea interfețelor cu utilizatorul

A computer screen shot of a computer

Description automatically generated

*Figură 1 - Pagina de home*

A screenshot of a computer

Description automatically generated

*Figură 2 - Pagina de logare*

A screenshot of a computer

Description automatically generated

*Figură 3 - Pagina adminului*

A screenshot of a computer

Description automatically generated

*Figură 4 - Pagina pacientului*

A screenshot of a computer

Description automatically generated

*Figură 5 - Pagina doctorului*

## Scenarii de utilizare

Scenariul 1: Programarea unei consultații

Utilizatori implicați: Pacient, medic, admin

Execuție:

Pacientul se autentifică în platforma web.

Pacientul accesează secțiunea de programări și selectează data și ora dorită.

Sistemul verifică disponibilitatea medicului și confirmă programarea.

Medicul și adminul primesc notificări despre noua programare.

Pacientul primește o confirmare prin email sau SMS cu detaliile programării.

Scenariul 2: Accesarea istoricului medical

Utilizatori implicați: Medic

Execuție:

Medicul se autentifică în sistem.

Medicul accesează profilul pacientului și revizuiește istoricul medical.

Medicul adaugă noi informații medicale (rețeta).

Informațiile sunt salvate și criptate în baza de date.

Scenariul 3: Gestionarea plăților

Utilizatori implicați: Pacient, admin

Execuție:

Pacientul efectuează plata online prin gateway-ul de plăți integrat.

Adminul verifică starea plății în sistem și actualizează statusul acesteia.

Pacientul primește o confirmare a plății.

Scenariul 4: Utilizarea dashboard-ului de administrație

Utilizatori implicați: Admin

Execuție:

Autentificare.

Adminul accesează sistemul și se autentifică folosind credențialele sale securizate.

Navigarea către dashboard-ul de administrație.

După autentificare, adminul o sa fie redirectionat in pagina de admin care oferă controale extensive asupra tuturor aspectelor operaționale ale clinicii.

Scenariul 5: Gestionarea feedback-ului pacienților

Utilizatori implicați: Pacient, admin

Execuție:

Pacientul se autentifică în portalul pacientului pe platforma web a clinicii, dar poate scrie feedback chiar daca nu este autentificat.

După finalizarea unei consultații, pacientul navighează către secțiunea „Feedback” unde completează un formular evaluând experiența sa.

Feedback-ul este trimis și stocat în baza de date.

Monitorizarea feedback-ului de către admin.

Adminul verifică periodic noile intrări de feedback în sistem si are posibilitatea sa dea acces pe pegina doar feedback-urilor cu limbaj adecvat.

Scenariul 6: Comunicarea prin chat între doctor și pacient

Utilizatori implicați: Pacient, doctor

Execuție: Doctorul și pacientul se autentifică în sistem folosind credențialele lor securizate.

După autentificare, pacientul accesează secțiunea din interfața utilizatorului pentru a începe o conversație cu doctorul.

Doctorul vizualizează mesajul, răspunde și oferă informații medicale necesare sau clarificări.

## Proiectare de detaliu

Dezvoltarea hardware

Prima etapa este de selectare a serverelor robuste care pot suporta aplicațiile web și baza de date a clinicii, precum și dispozitive de rețea adecvate pentru a asigura securitatea și eficiența comunicațiilor. Stațiile de lucru pentru personalul clinicii trebuie să fie compatibile cu cerințele software-ului.

A doua etapa este configurarea retelei si anume instalarea si configurarea hardware-ului de rețea, inclusiv routere și firewall-uri, pentru a suporta traficul intern și accesul securizat la internet.

A treia etapa este implementarea de soluții de securitate pentru a proteja infrastructura fizică și pentru a preveni accesul neautorizat.

Dezvoltarea software

Prima etapa este instalarea sistemelor de operare compatibile pe servere și stații de lucru, dezvoltarea aplicatiilor server care vor rula sistemul, și configurarea bazei de date „clinica\_db” pentru a gestiona eficient datele.

A doua etapa este asigurarea integrarii cu software-ul extern necesar, utilizând API-uri pentru a conecta diferite servicii, cum ar fi sistemele de plăți online sau alte sisteme medicale.

A treia etapa este crearea de interfete user-friendly pentru diferite categorii de utilizatori (pacienți, medici, personal administrativ), asigurând accesibilitate și ușurință în utilizare.

Ultima etapa este efectuarea de teste ample pentru a verifica integrarea corectă a componentelor hardware și software și pentru a asigura că sistemul îndeplinește toate cerințele de performanță și securitate.

### Proiectare software de deatliu

Serviciul de autentificare

Identificator: AuthService

Clasificare: Securitate

Definiție: Gestionează autentificarea utilizatorilor și menține securitatea sesiunilor.

Cerințe: Asigură autentificarea sigură și gestionarea sesiunilor pentru diverse tipuri de conturi.

Constrângeri: Trebuie să fie conform cu GDPR și alte standarde de securitate a datelor.

Procesare: Validează credențialele și emite token-uri de sesiune.

Serviciul de gestionare a programărilor

Identificator: AppointmentService

Clasificare: Aplicație

Definiție: Permite utilizatorilor să creeze, vizualizeze și modifice programări.

Cerințe: Interfață intuitivă și sincronizare în timp real cu calendarele medicilor.

Constrângeri: Interacțiunile trebuie să fie reflectate instantaneu pentru a evita suprapunerile.

Procesare: Include logica pentru alocarea timpilor disponibili și notificările aferente.

Serviciul de management al pacienților si doctorilor

Identificator: ManagementService

Clasificare: Management de date

Definiție: Administrează informațiile despre pacienți, inclusiv istoricul medical si informatiile despre doctorilor.

Cerințe: Securizarea și confidențialitatea datelor pacienților si ale doctorilor.

Constrângeri: Datele trebuie actualizate și accesibile doar personalului autorizat.

Procesare: Stochează și prelucrează informații detaliate despre starea de sănătate a pacienților si informatiile despre doctori.

Serviciul de plăți

Identificator: BillingService

Clasificare: Financiar

Definiție: Procesează plățile.

Cerințe: Integrare cu gateway-uri de plată și conformitate financiară.

Constrângeri: Procesarea rapidă și sigură a tranzacțiilor financiare.

Procesare: Acceptă plăți și raportează tranzacțiile.

Serviciul de comunicare internă

Identificator: CommunicationService

Clasificare: Comunicare

Definiție: Facilitează schimbul de mesaje între medici și pacienți.

Cerințe: Livrarea mesajelor în timp real și notificări adecvate.

Constrângeri: Confidențialitatea comunicațiilor și stocarea mesajelor securizată.

Procesare: Trimiterea și primirea mesajelor.

### Proiectare detaliată de securitate

Jurnalizare și auditare

Avem servere dedicate pentru înregistrarea tuturor activităților sistemului, care sunt protejate și accesibile doar personalului de securitate autorizat, facilitând auditurile periodice eficiente.

Criptare

Toate comunicațiile de date între servere și terminale sunt criptate folosind standarde avansate pentru a proteja integritatea și confidențialitatea informațiilor transmise. De asemenea, implementăm criptarea completă a discurilor pentru toate dispozitivele de stocare.

Utilizarea porturilor de rețea

Firewall-urile hardware sunt configurate pentru a controla traficul de rețea, permitând numai traficul legitim și închizând porturile neutilizate pentru a minimiza riscurile de securitate.

Detectare și prevenire a intruziunilor

Sistemele de detectare și prevenire a intruziunilor (IDS/IPS) sunt active atât la nivel de rețea, cât și la nivel de host pentru a identifica și a contracara activitățile suspecte sau maligne.

### Proiectare de detaliu pentru performanța sistemului

Capacitate și volum

Serverele de aplicație trebuie să suporte până la 1.000 de utilizatori simultan, procesând 100 de tranzacții pe secundă. Stocarea datelor este proiectată să înceapă de la 10 TB, extensibilă în funcție de necesități.

Performanță

Sistemul este optimizat pentru a răspunde la solicitări în mai puțin de 2 secunde și este proiectat pentru o disponibilitate de 99.9%, ceea ce permite un timp maxim de nefuncționare de 8.76 ore pe an.

Backup și recuperare

Strategia de backup include backup-uri incrementale zilnice și complete săptămânale, stocate în locații multiple, inclusiv în cloud.

## Controale pentru verificarea integrității sistemului

Securitate internă

Se implementează controlul accesului bazat pe roluri pentru a limita accesul la datele critice doar pentru utilizatorii autorizați în funcție de nevoile lor operaționale. Sistemul de gestionare a identităților va asigura că fiecare utilizator are permisiunile corespunzătoare pentru a accesa sau modifica datele.

Procese de verificare

Procesele de adăugare, ștergere sau actualizare a datelor vor necesita verificări și aprobări multiple pentru a asigura integritatea datelor.