**Documentul de specificare a cerințelor**

**Software Requirements Specification**

**(SRS) Document**

**<** **Platformă Web pentru Analiza și Vizualizarea Datelor Medicale pentru îmbunătățirea Deciziilor Clinice>**

**<Data><Versiune>**

**<Buşilă Ionuţ>**

|  |
| --- |
| **Istoricul versiunilor** |

| Versiune | Autor(i) principali | Descriere versiune | Dată |
| --- | --- | --- | --- |
|  |  |  |  |

|  |
| --- |
| **Revizuiri și aprobări** |

Istoric aprobări

| Aprobă | Versiune | Semnătură | Dată |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Istoric revizuiri

| Revizor | Versiune | Semnătură | Dată |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| **Cuprins** |

[1. Introducere 4](#_Toc190689617)

[1.1 Scopul 4](#_Toc190689618)

[1.2 Convenții ale documentului 4](#_Toc190689619)

[1.3 Audiență țintă 4](#_Toc190689620)

[1.4 Sfera de aplicare 4](#_Toc190689621)

[1.5 Referințe 4](#_Toc190689622)

[**2 Descriere generală 4**](#_Toc190689623)

[**2.1 Perspectiva produsului 4**](#_Toc190689624)

[**2.2 Caracteristici ale produsului 4**](#_Toc190689625)

[**2.3 Clase și caracteristici ale utilizatorilor 4**](#_Toc190689626)

[**2.4 Mediul de operare 5**](#_Toc190689627)

[**2.5 Constrângeri de proiectare și de implementare 5**](#_Toc190689628)

[**2.6 Presupuneri și dependențe 5**](#_Toc190689629)

[**3 Cerințele sistemului 5**](#_Toc190689630)

[**3.1 Funcționalitatea 1 5**](#_Toc190689631)

[**3.1.1 Descriere generală 5**](#_Toc190689632)

[**3.1.2 Flux de interacțiune (scenarii de utilizare) 5**](#_Toc190689633)

[**3.1.3 Condiții prealabile și constrângeri 6**](#_Toc190689634)

[**3.1.4 Detaliere cerință 6**](#_Toc190689635)

[**3.1.5 Scenarii de eroare și gestionarea excepțiilor 6**](#_Toc190689636)

[**3.1.5 Dependențe și interacțiuni cu alte funcționalități 6**](#_Toc190689637)

[**3.2 Cerința funcțională 2 6**](#_Toc190689638)

[3.3 .... 6](#_Toc190689639)

[4 Cerințe pentru interfețe 6](#_Toc190689640)

[4.1 Interfețe cu utilizatorul 6](#_Toc190689641)

[4.2 Interfețe hardware 6](#_Toc190689642)

[4.2.1 Configurații Minime Recomandate 7](#_Toc190689643)

[4.2.2 Dispozitive Externe Compatibile 7](#_Toc190689644)

[4.3 Interfețe de comunicare 7](#_Toc190689645)

[4.3.1 Protocoale și Standarde de Comunicare 7](#_Toc190689646)

[4.3.2 Cerințe de Securitate în Comunicare 7](#_Toc190689647)

[4.4 Interfețe software 7](#_Toc190689648)

[4.4.1 Tehnologii Utilizate 7](#_Toc190689649)

[4.4.2 Servicii Externe și API-uri 7](#_Toc190689650)

[5 Cerințe non-funcționale 8](#_Toc190689651)

[5.1 Cerințe de performanță 8](#_Toc190689652)

[5.2 Cerințe de siguranță 8](#_Toc190689653)

[5.3 Cerințe de securitate 8](#_Toc190689654)

[5.4 Atribute de calitate ale software-ului 8](#_Toc190689655)

[6 Alte cerințe 8](#_Toc190689656)

[7 Anexe 8](#_Toc190689657)

[7.1 Anexa A: Glosar 8](#_Toc190689658)

[7.2 Anexa B: Modele de Analiză 8](#_Toc190689659)

[7.3 Anexa C: Listă de Probleme 9](#_Toc190689660)

# Introducere

## Scopul

Scopul documentului.

Planificarea simplificata a consultatiilor, pentru a oferi pacientilor un mod usor de a programa consultatiile medicale, alegand medicul dorit si verificand disponibilitatea acestuia in timp real, economisind tip si reducand riscul de a suprapune programarile. Totodata creste accesibilitatea pacientilor, care pot face programari direct de acasa fara a fi necesar sa sune sau sa mearga fizic la cabinet. Acest lucru este util in special pentru pacienti cu mobilitate redusa, crescand gradul de satisfactie al pacientilor.

Gestionarea eficienta a programarilor de catre medici, care poate vizualiza rapid programarile si gestiona mai usor consultatiile ceea ce ajuta la o planificare mai eficienta a timpului si resurselor. De asemenea ajuta la prevenirea intarzierilor sau absentelor, prin trimiterea de notificari automate pacientilor. Totodata ajuta la reducerea erorilor umane.

Programul poate colecta date despre programarile pacientilor, tipurile de consultatii, si alte informatii care pot fi analizate pentru a optimiza serviciile oferite si a identifica posibilile imbunatatiri in cadrul cabinetului medical.

## Convenții ale documentului

Metodologiile tipografice urmate în cadrul documentului. De exemplu, orice abrevieri, stilizare tipografică a conținutului sau schimbări de fonturi și semnificația acestora.

VEZI ANEXA A.

## Audiență țintă

Descrie care parte a documentului este destinată fiecărui cititor. Include o listă a tuturor părților interesate ale proiectului, dezvoltatorilor, managerilor de proiect și testerilor pentru o mai bună claritate.

Audienta tinta este alcatuita din 4 tipuri de utilizatori : admini si utilizatori doctori testeri

Pentru utilizatori finali : Scopul aplicatiei, Cerinte de functionare, Interfata cu utilizatorul si Fluxul de interactiuni

Pentru testeri: Fluxul de Interactiuni, Scenariile de eroare si gestionarea exceptiilor, cerinte de fiabilitate.

Pentru administratori: Detaliile tehnice de infrastructura ( Configuratie minima radcomandata), cerinte de securitate si protectia datelor

## Sfera de aplicare

Specifică cum se aliniază obiectivele produsului cu obiectivele generale ale sistemului în care se integrează produsul și conturează beneficiile proiectului pentru afacere.

Aplicatia face parte dintr-un ecosistem mai larg dedicat imbunatatirii calitatii serviciilor, iar obiectivele sale sunt urmatoarele:

* Optimizarea resurselor si eficientizrea operationala
* Imbunatatirea experientei cu utilizatorul
* Reducerea costurilor administrative
* Asigurarea conformitatilor cu prevederile legale
* Reducerea erorilor si a confuziei specifica procesului de a te programa la medic.
* Accesibilitate extinsa
* Analiza si rapoarte pentru imbunatarirea deciziilor.

## Referințe

O listă a altor documente la care face referire documentul SRS, inclusiv surse precum site-uri web sau literatură scrisă.

* Regulemente si standarde legale privind GDPR : <https://eur-lex.europa.eu/eli/reg/2016/679/oj>
* ReactJS - <https://react.dev/learn>
* Standard pentru documentarea cerintelor software: <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:25022:ed-1:v1:en>

# Descriere generală

## Perspectiva produsului

Descrie contextul și originea produsului.

În contextul actual al digitalizării și al progresului tehnologic, sectorul medical generează și gestionează cantități uriașe de date. Creșterea volumului de informații medicale, combinată cu necesitatea unor decizii rapide și precise, a dus la dezvoltarea unor soluții informatice avansate pentru analiza și vizualizarea datelor clinice.

Această platformă web își propune să răspundă acestor cerințe printr-un sistem intuitiv, scalabil și sigur, destinat să sprijine personalul medical în analiza și interpretarea datelor clinice într-un mod mai rapid și mai eficient.

## Caracteristici ale produsului

Un rezumat la nivel înalt al funcțiilor pe care produsul le-ar îndeplini și caracteristicile care urmează să fie incluse.

Platforma web pentru analiza și vizualizarea datelor medicale este concepută pentru a facilita procesul decizional clinic printr-un set de funcționalități avansate. Aceasta va include caracteristici esențiale care permit colectarea, procesarea, analiza și reprezentarea grafică a informațiilor medicale într-un mod eficient și intuitiv.

1. Gestionarea și centralizarea datelor medicale
2. Analiză și procesare avansată a datelor
3. Vizualizare interactivă a datelor
4. Funcționalități de suport pentru decizie
5. Accesibilitate și securitate

## Clase și caracteristici ale utilizatorilor

O categorizare și profilare a utilizatorilor pentru care este destinat produsul și clasificarea lor în diferite clase de utilizatori.

Se identifică utilizatorii vizați și se clasifică pe diferite categorii: de exemplu pe baza frecvenței de utilizare, a subseturilor de funcții utilizate, expertiză tehnică, nivele și privilegii de securitate, etc.

Aplicatie utilizeaza un spectru larg de utilizatori implicati in deciziile clinice. Acestia sunt clasificati pe baza rolului lor in sistem, frecventei de utilizare si functionalitatea platformei.

1. Medici : au acces la datele pacientilor, pot vizualiza istoricul medical al pacientilor, pot analiza datele clinice si interpreta graficele, pot programa pacienti si pot modifica programarile deja facute de pacienti.
2. Pacienti: au acces doar la datele lor, isi pot vizualiza istoricul medical, istoricul programarilor. Pot efectua plati pentru consultatii, se pot programa la anumiti medici in functie de disponibilitatea fiecaruia
3. Administrator: Acces complet la setarile tehnice fara acces la datele clinice ale pacientilor, implementeaza actualizari, monitorizeaza performanta platformei, mentine baza de date si sistemul de autentificare functional.

## Mediul de operare

Specificarea mediului în care este proiectat să opereze produsul incluzând: toate platformele hardware, sistemele de operare, versiuni ale acestora și alte componente software și aplicații cu care trebuie să coexiste fără conflicte.

Produsul este proiectat sa opereze pe orice browser web ( Google Chrome, Mozilla Firefox, Microsoft Edge, Safari). Functioneaza pe Windows, Linux, MacOS.

## Constrângeri de proiectare și de implementare

Orice factori limitativi care ar putea reprezenta provocări pentru dezvoltarea produsului. Acestea includ atât constrângeri de proiectare, cât și de implementare. De exemplu: politici la nivel de companie sau legislative, limitări hardware (cerințe de timp, cerințe de memorie); interfețe către alte aplicații; tehnologii, instrumente și baze de date specifice de utilizat; operațiuni paralele; cerințe de limbaj; protocoale de comunicare; considerații de securitate; convenții de proiectare sau standarde de programare (de exemplu, dacă organizația clientului va fi responsabilă pentru întreținerea software-ului livrat).

1. Constrageri de functionalitate

* Respectarea reglumaentelor GDPR
* Stocarea si transmiterea datelor pacientilor
* Securitate robusta pentru autorizare si autentificare
* Gestionarea corecta a programarilor pentru acelasi medic si a zonelor orare
* Design responsive si usor de utilizat atat pe mobil cat si pe desktop
* Notificari inteligente

1. Constrangeri tehnice

* Gestionarea bazei de date
* AI-ul trebuie sa ofere raspunsuri relevante fara sa afecteze experienta utilizatorului
* Actualizarea in timp real a programarilor atat pentru utilizatori cat si pentru doctori
* Gestionarea de conflicte atunci cand doua persoane incearca sa rezerve acelasi interval de timp.

## Presupuneri și dependențe

O listă a tuturor presupunerilor făcute în legătură cu produsul proiectat și mediul de operare, împreună cu orice dependențe externe care ar putea afecta proiectul.

Presupuneri :

1. Utilizatori aplicatie trebuie trebuia sa poata accesa internetul si sa detina dispozitive compatibile. Se presupune ca viteza conexiuni este suficient de rapida pentru a incarca datele fara intarzieri majore si ca utilizeaza un browser modern ( Chrome,Edge,Firefox,Safari)
2. Aplicatia va fi utilizata intr-un mediu securizat. Se presupune ca dispozitivele utilizatorilor nu sunt compromise si ca nu exista atacuri directe asupra acestora si ca utilizatorii vor umra bune practici de securitate de exemplu: parole sigure, autentificare cu doi factori, etc.
3. Personalul medical vor avea abilitati tehnice de baza. Se presupune ca acestia stiu sa foloseasca interfete web si sa gestioneze programarile pacientilor fara instruier intensiva.
4. Sistemele de notificare (email/SMS) vor avea un uptime ridicat si vor livra mesajele in timp util.
5. Aplicatia va fii scalabila in functie de numarul de utilizatori, poate sustine cresterea de utilizatori fara probleme majore de performanta.
6. AI-ul utilizat are o rata acceptabila de acuratete si poate oferi sugestii utile pentru pacienti si medici.

Dependente :

1. Aplicatiile React depind de un backand care gestioneaza autenficiarea, stocarea datelor pacientilor si programarilor. Depinde de o baza de date performanta pentru stocarea si accesarea rapida a informatiilor.
2. Servicii terte pentru autentificare si securitate : Google Oauth, Firebase Authentication
3. Biblioteci si framework-uri externe
4. Reglementari si conformitate legala.

# Cerințele sistemului

Toate cerințele din cadrul sistemului sau sub-sistemului pentru a determina rezultatul pe care se așteaptă să-l ofere produsul în raport cu intrarea dată. Acestea constau în cerințe de design, cerințe grafice, cerințe de sistem de operare.

*Acest șablon ilustrează organizarea cerințelor funcționale pentru produs în funcție de caracteristicile sistemului, principalele servicii furnizate de produs. Poți prefera să organizezi această secțiune în funcție de cazul de utilizare, modul de operare, clasă de utilizatori, clasă de obiecte, ierarhie funcțională sau combinații ale acestora, orice ar face cel mai mult sens logic pentru produsul tău.*

## Funcționalitatea 1 – NUME FUNCTIONALITE

Utilizatorii trebuie sa poata crea conturi sa se autentifice.

### Descriere generală

* Separarea rolurilor, respectare GDPR, separarea atributiilor pe site.
* Medici trebuie sa poata gestiona programarile si sa consulte istoricul pacientilor. Pacientii trebuie sa poata vizualiza disponibilitatea medicilor si sa programeze consultatii. Administratorii trebuie sa poata configura serviciile platformei si sa gestioneze utilizatorii.
* Gestionarea corecta a utilizatorilor este importanta pentru a preveni accesul neautorizat la functionalitati specifice.

### Flux de interacțiune (scenarii de utilizare)

* Un pacient isi creeaza un cont si programeaza o consultative.
* Raspunsul din parte sistemului daca toate datele sunt corecte : contul este create cu success, email ul de confirmare este trimis, programarea este inregistrata si notificare este generate. Daca existra erori sistemul afiseaza mesaje de eroare pentru datele invalide, daca slotul de programare este ocupat, pacientul primeste o sugestie pentru o alta ora.
* Diagrama fluxului de evenimente (opțional)

[Pacient] -- (Accesează pagina principală) --> [Sistem]

[Pacient] -- (Completează formularul de înregistrare) --> [Sistem]

[Sistem] -- (Verifică datele și trimite email de confirmare) --> [Pacient]

[Pacient] -- (Confirmă email-ul) --> [Sistem]

[Pacient] -- (Se autentifică și selectează medic + data) --> [Sistem]

[Sistem] -- (Verifică disponibilitatea și confirmă programarea) --> [Pacient]

[Sistem] -- (Trimite notificare prin email/SMS) --> [Pacient]

### Condiții prealabile și constrângeri

### Detaliere cerință

Aceasta este lista cerințelor care definesc exact comportamentul funcționalității. Fiecare cerință trebuie să fie:

* **Clară** (fără ambiguități)
* **Testabilă** (poate fi verificată printr-un test).

1. Cerinte generale :

* REQ-1: Sistemul trebuie sa permita inregistrarea si autentificare utilizatorilor utilizand email si parola.
* REQ-2: Utilizatori trebuie sa poata reseta parola printr-un link de recuperare trimis pe email.
* REQ-3: Sistemul trebuie sa ofere autorizare pe baza de roluri (Pacient, Medic, Administrator)

1. Cerinte Programari Pacienti

* REQ-4: Pacientii trebuie sa poata vizualiza lista medicilor disponibili si sa isi aleaga unul.
* REQ-5: Sistemul trebuie sa permita selectarea unei date si ora disponibile pentru programare.
* REQ-6: Dupa finalizarea programari, pacientul trebuie sa primeasaca un email de confirmare.
* REQ-7: Sistemul trebuie sa permita anularea sau reprogramarea unei consultatii, cu notificarea automata a medicului.
* REQ-8: Daca pacientul introduce simtome, acestea trebuie sa fie transmise catre modulul AI pentru analiza.

1. Cerinte Medici

* REQ-9: Medici trebuie sa aiba acces la lista programarilor intr-un format structurat (calendar, lista)
* REQ-10: Medicul trebuie sa poata marca o consultatie ca finalizata, anulata sau reprogramata.
* REQ-11: Sistemul trebuie sa permita medicilor sa adauge note si recomandari medicale pentru fiecare programare/pacient.
* REQ-12: Mediucl trebuie sa primeasca notificari in timp real pentru modificarile programarilor in aplcatie/email/sms

1. Cerinte Administrator

* REQ-13: Administratorul trebuie sa poata gestiona conturile utilizatorilor ( activare/dezactivare, restare parola)
* REQ-14: Sistemul trebuie sa permita configurarea orelor de lucru ale medicilor si disponibilitatile acestora.
* REQ-15: Administratori trebuie sa poata vizualiza statistici despre utilizarea platformei(numar de programari,utilizatori activi, etc.)

1. Cerinte pentru Notificari si Securitate

* REQ-16: Sistemul trebuie sa trimita notificari prin Email/SMS pentru confirmarea si reamintirea programarilor
* REQ-17: Autentificarea trebuie sa fie ptorejata prin parole criptate si protectie impotriva atacuriilor brute-force.
* REQ-18: Sistemul trebuie sa aiba logarea a activitatilor pentru monitorizarea accesului la date sensibile.

*Exemplu:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Cerință** | **Descriere** | **Prioritate** | **Criterii de acceptanță** |
| **REQ-1** | Utilizatorul trebuie să poată introduce date în formularul X | Ridicată | Câmpurile acceptă date valide |
| **REQ-2** | Sistemul trebuie să valideze automat datele introduse | Medie | Se afișează un mesaj de eroare pentru date incorecte |

### ****Scenarii de eroare și gestionarea excepțiilor – pt functionalitatile de mai sus.****

* Utilizatorul introduce date invalide de inregistrare/autentificare: campuri incomplete/email invalid/parola prea scurta. Se afiseaza mesaje de eroare clar definite ( Email invalid. Introduceti un email valid). Campurile invalide sunt evidentiate visual(contur rosu). Sistemul valideaza parolele si cere o parola mai sigura.(parole trebuie sa fie de cel putin 8 caractere, o litera mare si un symbol special).
* Serverul nu raspunde sau conexiunea la internet este intrerupta: Afisarea unui mesaj de eraore si oferirea de optiuni de reincarcare, salvarea teporara a datelor introduce de utilizator in local storage pentru a preveni pierderea progresului (Eroare de conexiune! Verificati internetul si incercati din nou.)
* Alte erori prevazute: Parola gresita la autentificare, dupa 3 incercari esuate, blocarea temporara a contului. Pacientul incearca sa se programeze la un medic inexistent ( id invalid sau medical a fost sters), se ofera un mesaj (Medicul selectat nu mai este disponibil) si se ofera o lista cu medici activi. Incercarea de a sterge un cont care are programari active: Mesaj de eroare : “Nu poti sterge conturi cu programari active”, se cere anularea programarilor inainte de stergere.

### Dependențe și interacțiuni cu alte funcționalități

* Autentificare si gestionarea conturilor (medici, pacienti, administrator) necesita baza de date pentru a verifica utilizatorii si drepturile de access.
* Programarea consultatiilor depinde de disponibilitatea medicilor
* Notificarile Email/SMS depinde de existenta unei programari valide si de datele de contact ale utilizatorului ( se presupune ca datele de contact sunt corecte)
* Gestionarea programarilor de catre medici necesita autentificare si verificare dreputirlor de access.
* Administrarea utilizatorilor si setari depinde de existenta conturilor de utilizatori .
* Analiza simptomelor folosind AI depinde de introducerea corecta a simtomelor de catre pacient, oferind un raport care poate fi vizualizat de medic.

## Cerința funcțională 2

## ....

# Cerințe pentru interfețe

Această secțiune descrie modul în care sistemul interacționează cu utilizatorii, hardware-ul, alte sisteme software și rețele de comunicație.

## Interfețe cu utilizatorul – interfata de tip pagina web

Această secțiune descrie interfața grafică sau text-based a sistemului, incluzând **aspecte de design și cerințe de accesibilitate**. Aici se pot include ecrane demonstrative (funcționalitățile care ar apărea pe fiecare ecran, mesajele care urmează să fie afișate pe fiecare ecran și ghidurile de stil care urmează să fie utilizate). Detaliile designului interfeței utilizatorului ar trebui să fie documentate într-o specificație separată a interfeței utilizatorului.

## Interfețe hardware

Această secțiune descrie **cerințele minime hardware** pentru funcționarea sistemului și **dispozitivele externe compatibile**.

### ****Configurații Minime Recomandate****

* **PC/Laptop:** Intel Core i3(8th gen)/ AMD Ryzen 3 sau echivalent , 4GB RAM (8 GB recomandat), Minim 500MB liberi spatiu de stocare temporara si cache, browser compatibil: Chrome, Edge, Firefox, Safari(versiuni recente), conexiune la internet minim 5Mbps pentru access fluid
* **Dispozitive mobile:** Qualcomm Snapdragon 665/Apple A10/MediaTek Helio G80, 3GB RAM (4GB recomandat), minim 300MB liberi pentru cache si fisiere temporare, Android 9+ sau IOS 13+, minim 4G(WIFI recomandat)

## Interfețe de comunicare

Protocol de comunicare frontend si backend :

* **REST API + WEBSOCKETS**
* **Protocol HTTPS pentru Securitate**
* **Format date: JSON**
* **Autentificare JWT pentru securizarea cererilor**
* **Rate Limiting pentru prevenirea atacurilor DoS**
* **WebSockets pentru actualizare in timp real.(notificari programari)**
* **Comunicare NoSQL (MONGODB)**

### ****Protocoale și Standarde de Comunicare****

* **[Protocol 1]** – [Ex. HTTP/HTTPS pentru interacțiunea client-server]
* **[Protocol 2]** – [Ex. WebSockets pentru notificări în timp real]
* **[Protocol 3]** – [Ex. MQTT pentru integrare cu dispozitive IoT]

### ****Cerințe de Securitate în Comunicare****

* Toate datele transmise trebuie să fie **criptate**(ex. TLS 1.2+).
* Autentificarea utilizatorilor în sistem trebuie să respecte **OAuth 2.0** sau alt protocol standard.

## Interfețe software

* Servicii de autentificare externa : Google OAuth, Facebook Login, Apple Sign In. Permite utilizatorilor sa se autentifice folosind conturi de Google, Facebook sau Apple pentru a simplifica procesul de autentificare, prin utilizare unor servicii externe deja verificate.
* Servicii de plati online: Stripe/PayPal, pentru procesarea platilor online oferind utilizatorilor optiunea de a plati consultatiile medicale online.
* Interfata cu Serviciile de Notificare Email si SMS. ( SendGrid si Twilio). Pentru notificarile de confirmare, reprogramare sau anulare a consultatiilor, platforma utilizeaza Twillo pentru SMS si SendGrid pentru emailuri.
* Interfata cu AI. Platforma integreaza un serviciu AI care analizeaza simtomele pacientilor si ofera recomandarile de tratament, folosint un API extern.
* Interfata de Stocare a Datelor, pentru stocarea documentelor medicale platforma integreaza servicii de cloud storage externe. Aceasta permite stocare fisierelor pacientilor intr-un mod securizat si accesibil.

### Tehnologii Utilizate

* **Backend:** Node.js, Express.js, Django, Flask,
* **Frontend:** React, Redux, Vue.js , Angular
* **Bază de date:** MongoDB

### ****Servicii Externe și API-uri****

| **Serviciu/Interfață** | **Utilizare** |
| --- | --- |
| **[Serviciu/API 1]** | Autentificare [GoogleOAuth,Facebook Login, Apple SignIn] |
| **[Serviciu/API 2]** | Stripe/PayPal API pentru plati. |
| **[Serviciu/API 3]** | Amazon S3 API pentru stocare. |

# Cerințe non-funcționale

Sistemul trebuie sa fie fiabil, sa fie disponibil 99% din timp, cu exceptia perioadelor de intretinere planificata. In cazul unor erori, sistemul trebuie sa se recupereze automat fara a afecta datele utilizatorilor sau operatiunile in curs.

Aplicatia trebuie sa suporte un numar cat mai mare de utilizatori (minim 1000 utilizatori activi pe zi in prima faza. Platforma trebuie sa poata fii scalata pe masur ace numarul de utilizatori creste, utilizand solutii precum load balancing sau orechestrare de containere.

## Cerințe de performanță

Timpul de raspuns al aplicatiei nu trebuie sa depaseasca 2 secunde pentru operatiunile majore ( incarcarea paginilor, trimiterea si procesarea unui formular). Sistemul trebuie sa permita procesarea a cel putin 50 de programari simultan pentru a asigura o utilizare continua si fluida a sistemului.

In cazul aplicatiei mobile, timpul de incarcare nu trebuie sa depaseasca 5 secunde, luand in considerare conexiunile mai lente.

## Cerințe de siguranță

Sistemul va implementa autentificare pe mai multi factori pentru acecsul la functionalitati sensibile, precum gestionarea programarilor sau vizualizarea istoricului medical.

Fiecare utilizator va avea permisiuni diferite in functie de rolul lor. Accesul la datele medicale va fi restrictionat la medicii care au in ingrijire pacientul.

Aplicatia va fi protejata impotriva celor mai comune tipuri de atacuri, precum XSS, CSRF, DoS. Se implementeaza un sistem de detectia a atacurilor si un sistem de avertizare in caz de tentative de acces nautorizat.

Toate accesarile sensibile si modificarile de date vor fi logate pentru a putea fi audiate ulterior. Sistemul include un jurnal de audit al accesului, care sa permita urmarirea activitatii fiecarui utilizator.

Aplicatia include backup-uri regulate si planuri de recuperare a datelor in caz de pierdere.

## Cerințe de securitate

Aplicatia respecta principiul transparentei. Utilizatori sunt informati clar despre modul in care datele lor personale sunt colectate, procesate, stocate. Este obtinut consimtamantul explicit al utilizatorului pentru prelucrarea datelor lor personale.

## Atribute de calitate ale software-ului

Detalierea calităților suplimentare care trebuie incorporate în software, cum ar fi *ușurința în întreținere, adaptabilitatea, flexibilitatea, utilizabilitatea, fiabilitatea, portabilitatea* etc.

Codul estre structurat si documentat clar, astfel incat orice dezvoltator sa poata intelege si sa modifice sau adauge noi functionalitati fara dificultate.

Design-ul este flexibil, foloseste principii de arhitectura bazata pe microsfervicii, API-uri standardizate si intefete flexibile care permit integrarea cu noi tehnologii sau modificarea proceselor fara a necesita rescrierea intregii aplicatii.

UI-ul este simplu si intuitiv, folosind un „user-centered design” si testarea interfetei cu utilizatori reali. De asemenea se asigura compatibilitatea cu dispozitive mobile si desktop.

Fiabilitatea este asigurata prin monitorizarea performantei si logarea erorilor pentru a detecta rapid orice problema care ar putea aparea.

Portabilitatea este asigurata prin folosirea unor framework-uri multiplatforma. Totodata utilizarea containerelor (Docker) poate facilita migrarea si rularea aplicatiei pe mai multe platforme.

Am utilizat diferite tehnici de cache si load balanceing pentru a imbunatatii performanta in cazul unui numar mare de utilizatori, pentru a reduce timpul de raspuns al aplicatiei si pentru a gestiona eficient resursele (memorie,CPU, banda)

# Alte cerințe

Definiți orice alte cerințe care nu sunt acoperite în altă parte a SRS. Acestea ar putea include cerințe legate de baze de date, cerințe de internaționalizare, cerințe legale, obiective de reutilizare pentru proiect și altele asemenea. Adăugați orice secțiuni noi care sunt de interes pentru proiect.

Respectarea reglementarilor GDPR.

Toate datele cu caracter personal trebuie sa fie procesate conform regulilor in vigoare.

Confidentialitatea datelor de sanatate.

Avand in vedere ca aplicatia va gestiona informatii sensibile privind sanatatea, trebuie sa se asigure conformitatea cu regulile locale si internationale privind confidentialitatea datelor de sanatate.

# Anexe

## Anexa A: Glosar

<Definiți toți termenii necesari pentru a interpreta corect SRS-ul, inclusiv acronime și abrevieri. Puteți dori să construiți un glosar separat care să cuprindă mai multe proiecte sau întreaga organizație și să includeți doar termeni specifici unui singur proiect în fiecare SRS.>

**API** (Application Programming Interface): Interfața de programare a aplicațiilor, un set de reguli și specificații care permite ca aplicațiile software să comunice între ele.

**DBMS** (Database Management System): Sistemul de management al bazelor de date, software care permite crearea, gestionarea și manipularea bazelor de date.

**Frontend**: Partea aplicației cu care utilizatorul interacționează direct, inclusiv design-ul vizual și interfețele de utilizator (UI).

**Backend**: Partea aplicației care se ocupă cu procesarea logicii, gestionarea bazelor de date și răspunsul la cererile provenite de la frontend.

**UI** (User Interface): Interfața utilizatorului, componenta grafică a aplicației care permite interacțiunea utilizatorului cu software-ul.

**UX** (User Experience): Experiența utilizatorului, care se referă la modul în care un utilizator percepe și interacționează cu aplicația.

**SQL** (Structured Query Language): Limbajul standard pentru interogarea și manipularea bazelor de date relaționale.

**NoSQL**: Tip de bază de date care nu folosește SQL pentru interogarea datelor și care este orientată spre stocarea unor volume mari de date semi-structurate sau nestructurate.

**SSL/TLS** (Secure Sockets Layer / Transport Layer Security): Protocoale de criptare care asigură securitatea comunicațiilor pe internet.

**GDPR** (General Data Protection Regulation): Regulament european privind protecția datelor cu caracter personal, care reglementează modul în care datele trebuie colectate, procesate și stocate.

**Pacient**: Utilizatorul aplicației care beneficiază de servicii medicale, se programează pentru consultații și trimite simptome către AI pentru evaluare.

**Doctor**: Utilizatorul aplicației care gestionează programările pacienților și oferă tratamente pe baza analizei efectuate de sistemul AI.

**Administrator**: Utilizatorul aplicației care se ocupă de gestionarea și întreținerea platformei, inclusiv de crearea conturilor și monitorizarea performanței aplicației.

**AI** (Inteligență Artificială): Sistemul care analizează simptomele pacienților și sugerează posibile tratamente pentru doctori, pe baza algoritmilor de învățare automată și procesare a limbajului natural.

**Cont** **Pacient**: Contul utilizatorului pacient, care include informațiile personale ale pacientului, istoricul medical, programările și rezultatele consultațiilor.

**Cont** **Doctor**: Contul utilizatorului doctor, care include informațiile profesionale ale doctorului, programările cu pacienții, istoricul consultațiilor și recomandările AI.

**Cont** **Administrator**: Contul utilizatorului administrator, care permite gestionarea tuturor aspectelor platformei, incluzând managementul conturilor utilizatorilor, monitorizarea activității și întreținerea sistemului.

**Programare**: Acțiunea de a seta o consultație între pacient și doctor, în baza disponibilității acestora.

**DeepSeek** **AI**: Replica AI care ajută pacientul să înțeleagă mai bine problemele sale de sănătate, bazându-se pe informațiile oferite de acesta.

**Flux** **de** **lucru** **al** **pacientului**: Secvența de acțiuni prin care pacientul se înregistrează pe platformă, trimite simptome, primește evaluări AI și se programează pentru o consultație.

**Flux** **de** **lucru** **al** **doctorului**: Secvența de acțiuni prin care doctorul primește informațiile pacientului, consultă recomandările AI, stabilește tratamente și gestionează programările.

**Sistem** **de** **autentificare**: Procesul prin care utilizatorii (pacienți, doctori, administratori) își confirmă identitatea pentru a accesa aplicația.

**Roluri** **de** **utilizator**: Permisiuni și drepturi asociate fiecărui tip de utilizator (pacient, doctor, administrator) în aplicație.

**Interfață** **de** **programare** **a** **aplicațiilor** (API) externă: Servicii externe de comunicare cu aplicația pentru procesarea plăților, integrarea cu sisteme de sănătate externe sau pentru verificarea identității utilizatorilor.

**Rezervare** **programare**: Procesul prin care un pacient își fixează un loc într-o consultație cu un doctor pe baza disponibilității acestuia.

**Acronime și Abrevieri**

* **API** - Application Programming Interface
* **DBMS** - Database Management System
* **UI** - User Interface
* **UX** - User Experience
* **SaaS** - Software as a Service
* **GDPR** - General Data Protection Regulation
* **HIPAA** - Health Insurance Portability and Accountability Act
* **SSL** - Secure Sockets Layer
* **TLS** - Transport Layer Security
* **SQL** - Structured Query Language
* **NoSQL** - Not Only SQL
* **AI** - Artificial Intelligence
* **IP** - Internet Protocol
* **URL** - Uniform Resource Locator
* **XML** - Extensible Markup Language
* **JSON** - JavaScript Object Notation

## Anexa B: Modele de Analiză

<Opțional, includeți orice modele de analiză relevante, cum ar fi diagramele de flux de date, diagramele de clasă, diagramele de tranziție de stare sau diagramele de relații entitate-asociere.>

**Diagrama de Flux de Date**

[Utilizator]

|

v

[Aplicație Web]

|

v

[Backend - Procesare Date]

|

v

[Aplicație Bază de Date]

**Diagrama de Tranziție de Stare**

**[Inițial] ---> [Confirmată] ---> [Finalizată]**

**| |**

**v v**

**[Anulată] <-------------------|**

**Diagrama de Flux a Interacțiunii Utilizatorului**

[Start]

|

v

[Autentificare]

|

v

[Selecție doctor]

|

v

[Verificare disponibilitate] <--- [Orar doctor]

|

v

[Confirmare programare]

|

v

[Generare notificare pacient]

|

v

[Final]

## Anexa C: Listă de Probleme

<Aceasta este o listă dinamică a problemelor de cerințe deschise care rămân de rezolvat, incluzând aspecte care urmează a fi decise în viitor - decizii în așteptare, informații necesare, conflicte așteptând rezolvare și altele asemenea.>

Decizia cu privire la integrarea unui sistem extern de autentificare (Google, Facebook, etc.) sau dezvoltarea unui sistem intern de autentificare este încă în așteptare.

Nu am stabilit clar care sunt permisiunile exacte pentru diferitele roluri de utilizatori (pacient, doctor, administrator), mai ales în ceea ce privește accesul la istoricul medical al pacientului.

Specificarea unor cerinte de performanta clar definite.

Politica de backup.

Conflicte intre securitatea datelor si experinta utilizatorului : echilibru intre criptarea datelor, autentificarea complexa si nevoia de a oferi o experienta rapida si usor de utilizat pentru pacient si doctor.

Stabilirea exacta a tipului de notificari care vor fi trimise pacientilor si doctorilor.

Testarea compatibilitatii cu diferite platforme mobile si versiuni.

# Exemplu (parțial) de completare

## ****Funcționalitate: Autentificare Utilizator****

### ****3.1.1 Descriere Generală****

Această funcționalitate permite utilizatorilor să se autentifice în sistem folosind o adresă de email și o parolă. Este o funcționalitate esențială pentru gestionarea accesului utilizatorilor.

### ****3.1.2 Flux de Interacțiune****

1. Utilizatorul introduce email-ul și parola
2. Sistemul validează datele
3. Dacă datele sunt corecte → utilizatorul este autentificat
4. Dacă datele sunt greșite → apare un mesaj de eroare

### ****3.1.3 Condiții Prealabile și Constrângeri****

* Utilizatorul trebuie să aibă un cont activ
* Sistemul trebuie să fie conectat la baza de date pentru validarea acreditărilor

### ****3.1.4 Cerințe Funcționale Detaliate****

| **Cerință** | **Descriere** | **Prioritate** | **Criterii de acceptanță** |
| --- | --- | --- | --- |
| **REQ-1** | Constrangere caractere speciale destinate introducerii unui format corect de email | Mediu | Campul accepta doar date introduce ce respecta formatul email@test.ro |
| **REQ-2** | Verificarea in baza de date a existentei datelor introduse pentru autentificare | Ridicata | Se verifica daca in baza de date exista inregistrari specifice numelui si parolei din campurile de logare |

### ****3.1.5 Scenarii de Eroare și Gestionare a Excepțiilor****

* Dacă utilizatorul introduce o parolă greșită de 5 ori → cont blocat temporar
* Dacă baza de date nu este accesibilă → mesaj de eroare „Serviciu indisponibil”

### ****3.1.6 Dependențe și Interacțiuni cu Alte Funcționalități****

* Depinde de sistemul de gestionare a utilizatorilor
* Se integrează cu sistemul de recuperare a parolei

**4.1 Interfețe cu utilizatorul**

* Sistemul trebuie să ofere o interfață grafică intuitivă, optimizată pentru utilizare pe desktop și mobil.
* Ecranele principale trebuie să includă:
  + **Ecran de autentificare:** câmp pentru email/parolă, buton „Login”
  + **Ecran principal:** navigare prin meniul aplicației
  + **Ecran de setări:** opțiuni de personalizare a contului

**4.2 Interfețe Hardware**

* Aplicația trebuie să ruleze pe următoarele configurații minime:
  + **PC/Laptop:** Procesor minim Intel i5, 8GB RAM, Windows 10/Linux
  + **Smartphone:** Android 9+ sau iOS 14+
* Dispozitive externe compatibile: cititoare de amprente, module NFC

**4.3 Interfețe de Comunicare**

* Aplicația va folosi următoarele protocoale:
  + **HTTP/HTTPS** pentru interacțiunea client-server
  + **WebSockets** pentru notificări în timp real
  + **MQTT** pentru transmiterea datelor IoT
* Datele trebuie să fie criptate utilizând **TLS 1.2+** pentru a asigura securitatea comunicațiilor.

**4.4 Interfețe Software**

* Backend-ul aplicației se va baza pe **Django + MySQL**.
* Aplicația va interacționa cu următoarele API-uri externe:
  + **Google Maps API** pentru localizare
  + **Stripe API** pentru procesarea plăților
  + **Firebase Authentication** pentru gestionarea conturilor de utilizatori