Exemplu gratuit, cu comentarii: licenta-disertatie-romania.com/domenii/informatica/

UNIVERSITATEA [NUMELE UNIVERSITĂȚII]
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ
Programul de studii: Informatică

LUCRARE DE LICENȚĂ

Aplicație web pentru programări online într-un cabinet medical

Proiectare, implementare și testare

EXEMPLU FICTIV – structura, tabelele și figurile unei lucrări de licență. Datele sunt simulate.

Coordonator științific:

Conf. univ. dr. Mihai Stanciu (nume fictiv)

Absolvent:

Radu Georgescu (nume fictiv)

CLUJ-NAPOCA, 2026

Exemplu fictiv cu date simulate · **Ai nevoie de ajutor cu lucrarea ta?** licenta-disertatie-romania.com/contact

DECLARAȚIE DE AUTENTICITATE

Subsemnatul(a) **Radu Georgescu**, absolvent(ă) al(a) programului de studii Informatică, declar pe propria răspundere că lucrarea de licență cu titlul „Aplicație web pentru programări online într-un cabinet medical” a fost elaborată de mine, sub îndrumarea coordonatorului științific, nu a mai fost prezentată la o altă facultate sau instituție de învățământ superior și că toate sursele folosite, inclusiv cele de pe internet, sunt citate în lucrare, cu respectarea regulilor privind drepturile de autor.

Data: [data]

Semnătura: [semnătura]

Majoritatea facultăților au un model oficial de declarație de autenticitate. Copiază-l exact din ghidul facultății și semnează-l; textul de aici este doar orientativ.
Sfat
20/05/2026 10:00

CUPRINS

Cuprinsul arată din prima pagină logica lucrării: introducere, capitole teoretice, cercetarea proprie, concluzii. În Word îl generezi automat din stilurile de titlu (Referințe → Cuprins), ca să se actualizeze singur.
Ghid: licenta-disertatie-romania.co
Rolul secțiunii
20/05/2026 10:00

Lista figurilor și a tabelelor.....4
Rezumat.....5
Introducere.....6
Capitolul 1. Studiu bibliografic și tehnologii.....7
1.1. Soluții existente.....7
1.2. Tehnologii folosite.....7
Capitolul 2. Analiză și proiectare.....8
2.1. Cerințe funcționale.....8
2.2. Arhitectura aplicației.....8
2.3. Baza de date.....9
Capitolul 3. Implementare.....10
3.1. Fluxul unei programări.....10
3.2. Prevenirea programărilor duplicate.....10
Capitolul 4. Testare și evaluare.....11
4.1. Teste automate.....11
4.2. Performanță.....12
4.3. Utilizabilitate.....13
4.4. Evaluarea autorului.....14
Concluzii și dezvoltări ulterioare.....15
Bibliografie.....16
Anexa 1. Capturi de ecran și codul sursă.....17
Bonus: același paragraf, înainte și după corectare.....18

Lista figurilor şi a tabelelor

Listele de figuri şi tabele sunt cerute de multe facultăţi. În Word le generezi automat dacă pui titlurile cu Referinţe → Inserare legendă, apoi Referinţe → Inserare tabel de figuri. **Ghid:** licenta-disertatie-romania.co
Rolul secţiunii
20/05/2026 10:00

Lista figurilor

Figura 1. Arhitectura aplicaţiei pe trei niveluri....8
Figura 2. Modulele aplicaţiei....9
Figura 3. Fluxul unei programări....10
Figura 4. Fluxul de integrare şi livrare continuă (CI/CD) folosit în proiect....11
Figura 5. Testele automate, pe tipuri....12
Figura 6. Timpul mediu de răspuns sub încărcare....13
Figura 7. Scorul SUS mediu, pe grupuri de participanţi....14

Lista tabelelor

Tabelul 1. Comparaţie între aplicaţii existente de programări online....7
Tabelul 2. Tehnologiile alese şi motivul alegerii....7
Tabelul 3. Cerinţele funcţionale ale aplicaţiei....8
Tabelul 4. Tabelele principale din baza de date....9
Tabelul 5. Rezultatele testelor automate....11
Tabelul 6. Timpul mediu de răspuns în funcţie de numărul de utilizatori simultani (ms)....12
Tabelul 7. Evaluarea utilizabilităţii cu chestionarul SUS (10 participanţi)....13
Tabelul 8. Evaluarea autorului: obiectivele şi cerinţele nefuncţionale faţă de rezultate....14

Rezumat

Lucrarea prezintă proiectarea, implementarea și testarea unei aplicații web pentru programări online într-un cabinet medical mic. Aplicația permite pacienților să vadă intervalele libere și să își facă, mute sau anuleze singuri programarea, iar medicilor și recepției să gestioneze agenda și o listă de așteptare. Soluția folosește React și TypeScript pe client, Node.js și Express pe server și o bază de date PostgreSQL; programările duplicate sunt prevenite printr-o tranzacție cu blocarea rândului. Din cele 184 de teste automate, 183 au trecut, cu o acoperire a codului de 91% pe server. Timpul mediu de răspuns rămâne sub 200 ms până la 200 de utilizatori simultani, iar evaluarea cu 10 participanți a dat un scor SUS mediu de 80,0. Lucrarea se încheie cu o evaluare a cerințelor și recomandă un test pilot înainte de folosirea în practică.

Rezumatul are de obicei 150–300 de cuvinte și spune pe scurt tema, obiectivul, metoda, rezultatele și concluzia. Multe facultăți îl cer și în limba engleză, cu 3–5 cuvinte-cheie.
Rolul secțiunii
20/05/2026 10:00

Cuvinte-cheie: aplicație web, programări online, React, Node.js, PostgreSQL, testare software

Abstract

This thesis presents the design, implementation and testing of a web application for online appointments in a small medical practice. Patients can view free time slots and book, move or cancel their own appointments, while doctors and reception staff manage the schedule and a waiting list. The solution uses React and TypeScript on the client, Node.js and Express on the server and a PostgreSQL database; duplicate bookings are prevented by a transaction with row locking. Of the 184 automated tests, 183 passed, with 91% code coverage on the server. The average response time stays below 200 ms for up to 200 concurrent users, and an evaluation with 10 participants gave a mean SUS score of 80.0. The thesis ends with an assessment against the requirements and recommends a pilot test before real-world use.

Keywords: web application, online appointments, React, Node.js, PostgreSQL, software testing

Introducere

La Informatică introducerea spune ce problemă rezolvă aplicația, pentru cine și ce aduce nou față de soluțiile existente. Contribuția personală trebuie să fie clară de la început.
Ghid: licenta-disertatie-romania.co
Rolul secțiunii
20/05/2026 10:00

Multe cabinete medicale mici fac programările încă prin telefon. Pacienții sună în timpul programului, recepția notează manual, iar programările anulate rămân deseori neocupate.

Obiectivul lucrării este proiectarea și implementarea unei aplicații web prin care pacienții își fac singuri programarea, iar cabinetul își gestionează agenda și lista de așteptare.

Contribuţiile personale sunt: analiza cerinţelor, proiectarea bazei de date, implementarea aplicației complete (frontend și backend), mecanismul care împiedică programările duplicate și testarea aplicației.

Scrie obiectivul astfel încât să poată fi verificat la final: aici, cerințele din Tabelul 3 sunt reluate în evaluarea din Capitolul 4.
Ghid: licenta-disertatie-romania.co
Sfat
20/05/2026 10:00

Capitolul 1. Studiu bibliografic și tehnologii

Capitolul compară soluțiile existente și justifică alegerea tehnologiilor. Nu explica ce este React în general; explică de ce l-ai ales pentru proiectul tău.
Ghid: licenta-disertatie-romania.co
Rolul secțiunii
20/05/2026 10:00

1.1. Soluții existente

Am analizat două aplicații de programări folosite de cabinete din România [1], [2]. Tabelul 1 arată funcționalitățile lor în comparație cu aplicația propusă.

Tabelul 1. Comparație între aplicații existente de programări online

Funcționalitate	Aplicația A	Aplicația B	Aplicația propusă
Programare online 24/7	Da	Da	Da
Confirmare prin e-mail și SMS	Doar e-mail	Da	Da
Calendar pentru fiecare medic	Nu	Da	Da
Lista de așteptare automată	Nu	Nu	Da
Cost pentru cabinet	Abonament	Abonament	Open-source

Sursa: Analiză proprie; aplicațiile A și B sunt generice.

1.2. Tehnologii folosite

Alegerea tehnologiilor a urmărit trei criterii: siguranța datelor, ușurința dezvoltării și costul de găzduire [3], [4].

Tabelul 2. Tehnologiile alese și motivul alegerii

Componenta	Tehnologia	De ce
Interfața (frontend)	React + TypeScript	Componente reutilizabile, verificarea tipurilor la compilare
Server (backend)	Node.js + Express	Același limbaj pe client și pe server, API REST simplu
Baza de date	PostgreSQL	Tranzacții și constrângeri care previn programările duplicate
Autentificare	JWT	Sesiuni fără stare pe server, ușor de scalat
Găzduire	Docker	Același mediu la dezvoltare și în producție

Sursa: Prelucrare proprie.

La IEEE, sursele se numerotează în ordinea în care apar în text: [1], [2], [3]. Documentațiile oficiale ale tehnologiilor sunt surse acceptate.
Sfat
20/05/2026 10:00

Capitolul 2. Analiză și proiectare

Proiectarea arată deciziile luate înainte de cod: cerințe, arhitectură, baza de date. Diagramele UML (cazuri de utilizare, clase, secvență) sunt cerute la multe facultăți.
Rolul secțiunii
20/05/2026 10:00

2.1. Cerințe funcționale

Tabelul 3. Cerințele funcționale ale aplicației

ID	Cerința	Actor	Prioritate
CF1	Pacientul vede intervalele libere ale unui medic	Pacient	Ridicată
CF2	Pacientul face, mută sau anulează o programare	Pacient	Ridicată
CF3	Aplicația trimite confirmare și reamintire	Sistem	Ridicată
CF4	Medicul își definește programul de lucru	Medic	Medie
CF5	Recepția vede programările zilei și le modifică	Recepție	Medie
CF6	Un loc eliberat este oferit primului din lista de așteptare	Sistem	Scăzută

Sursa: Analiza cerințelor.

Cerințele nefuncționale sunt: timp de răspuns sub 200 ms pentru 100 de utilizatori simultani, protecția datelor personale conform GDPR și funcționarea pe telefon.

2.2. Arhitectura aplicației

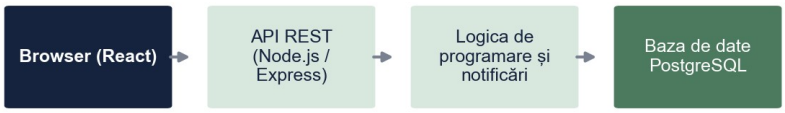


Figura 1. Arhitectura aplicației pe trei niveluri
Sursa: Prelucrare proprie.

Aplicația are o arhitectură pe trei niveluri (Figura 1). Modulele sunt grupate după tipul de utilizator (Figura 2).

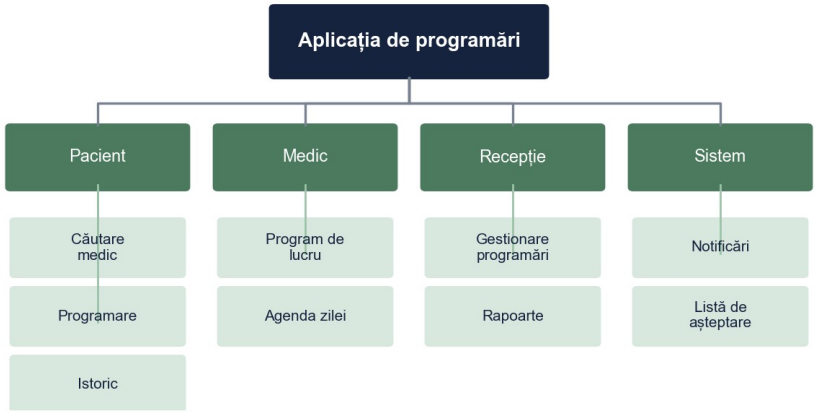


Figura 2. Modulele aplicației
Sursa: Prelucrare proprie.

2.3. Baza de date

Baza de date are patru tabele principale (Tabelul 4). Fiecare interval poate avea cel mult o programare, iar o constrângere de unicitate pe interval_id împiedică dublarea lui.

Tabelul 4. Tabelele principale din baza de date

Tabela	Câmpuri principale	Relații
pacient	id, nume, email, telefon	1 : N cu programare
medic	id, nume, specialitate	1 : N cu interval
interval	id, medic_id, inceput, sfarsit	1 : 1 cu programare
programare	id, pacient_id, interval_id, stare, creat_la	N : 1 cu pacient și interval

Sursa: Proiectarea bazei de date

Nu pune în lucrare capturi de ecran cu cod sau cu diagrame greu de citit. Diagramele se refac clar, iar codul se pune ca text, doar fragmentele importante.
Ghid: licenta-disertatie-romania.co
Greșeală frecventă
20/05/2026 10:00

Capitolul 3. Implementare

Implementarea este contribuția ta personală, alături de testare. Nu pui tot codul: alegi 2–3 părți dificile, explici problema și soluția, iar codul complet merge într-un repository sau în anexă.
Rolul secțiunii
20/05/2026 10:00

3.1. Fluxul unei programări

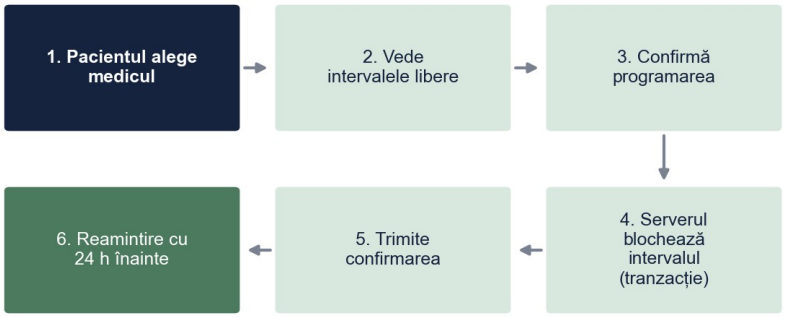


Figura 3. Fluxul unei programări
Sursa: Prelucrare proprie.

Figura 3 arată pașii unei programări, de la alegerea medicului la reamintirea trimisă cu 24 de ore înainte. Pasul critic este blocarea intervalului, descris mai jos.

3.2. Prevenirea programărilor duplicate

Dacă doi pacienți aleg același interval în același moment, doar unul trebuie să primească programarea. Problema se rezolvă cu o tranzacție și cu blocarea rândului în baza de date [5]:

```
async function creeazaProgramare(pacientId, intervalId) {
  return db.tx(async (t) => {
    const interval = await t.one(
      'SELECT * FROM interval WHERE id = $1 FOR UPDATE', [intervalId]);
    if (interval.ocupat) throw new Error('Interval ocupat');
    await t.none('UPDATE interval SET ocupat = true WHERE id = $1', [intervalId]);
    return t.one(
      'INSERT INTO programare (pacient_id, interval_id, stare)
       VALUES ($1, $2, $3) RETURNING id', [pacientId, intervalId, 'confirmata']);
  });
}
```

Instrucțiunea `FOR UPDATE` blochează rândul până la finalul tranzacției, astfel încât a doua cerere vede intervalul deja ocupat.

Pentru fiecare fragment de cod explică problema, soluția și de ce ai ales-o. Comisia întreabă des despre astfel de detalii la susținere.
Sfat
20/05/2026 10:00

Capitolul 4. Testare și evaluare

Testarea demonstrează că aplicația funcționează și cât de bine; împreună cu implementarea, este contribuția ta personală. Rezultatele măsurabile (teste trecute, timpi de răspuns, scor de utilizabilitate) sunt foarte apreciate. Încheie cu evaluarea ta față de cerințe.
Rolul secțiunii
20/05/2026 10:00

4.1. Teste automate

Testele automate rulează la fiecare commit, într-un flux de integrare și livrare continuă: dacă trec, se construiește o imagine Docker care ajunge întâi pe serverul de test și apoi în producție (Figura 4).

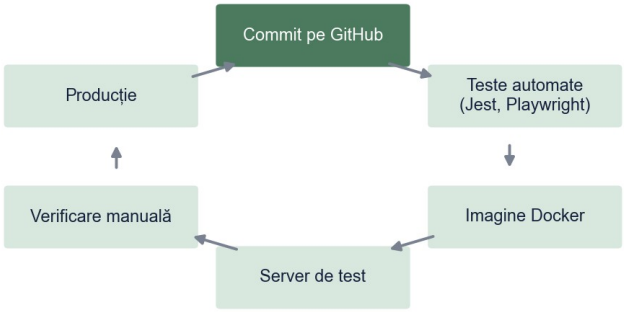


Figura 4. Fluxul de integrare și livrare continuă (CI/CD) folosit în proiect
Sursa: Prelucrare proprie.

Tabelul 5. Rezultatele testelor automate

Tipul testelor	Număr de teste	Trecute	Acoperire cod
Teste unitare (backend)	86	86	91%
Teste unitare (frontend)	54	54	78%
Teste de integrare (API)	32	31	–
Teste end-to-end	12	12	–
Total	184	183	–

Sursa: Rularea testelor cu Jest și Playwright (valori simulate pentru exemplu).

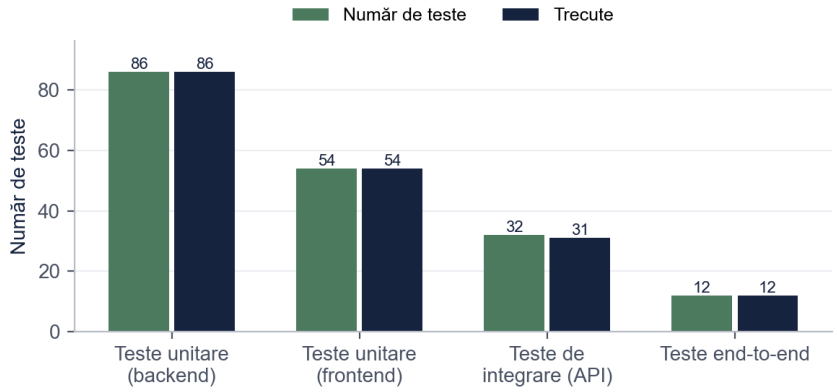


Figura 5. Testele automate, pe tipuri

Sursa: Prelucrare proprie.

Din cele 184 de teste, 183 au trecut (Tabelul 5, Figura 5). Singurul test de integrare eşuat a fost cauzat de un fus orar setat greşit pe serverul de test; după corectare, testul a trecut.

4.2. Performanţă

Tabelul 6. Timpul mediu de răspuns în funcție de numărul de utilizatori simultani (ms)

Utilizatori simultani	Căutare intervale	Creare programare
10	42	65
50	48	74
100	61	96
200	95	158
400	184	322

Sursa: Test de încărcare cu k6, server cu 2 vCPU și 4 GB RAM (valori simulate).

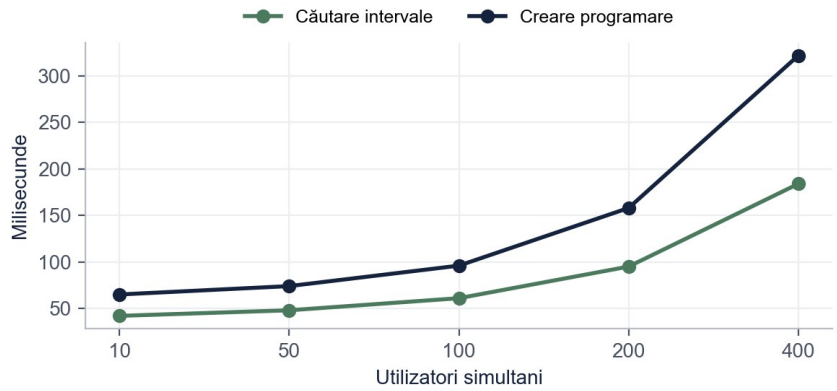


Figura 6. Timpul mediu de răspuns sub încărcare

Sursa: Prelucrare proprie.

Cerința nefuncțională, sub 200 ms pentru 100 de utilizatori simultani, este îndeplinită: 61 ms pentru căutare și 96 ms pentru crearea programării. Ambele operații rămân sub 200 ms și la 200 de utilizatori. La 400 de utilizatori, căutarea rămâne sub prag (184 ms), iar crearea programării îl depășește (322 ms), un trafic mult peste cel al unui cabinet mic (Figura 6).

4.3. Utilizabilitate

Aplicația a fost testată de 10 participanți (6 pacienți și 4 membri ai personalului medical), care au completat chestionarul SUS [6] după cinci sarcini: căutarea unui medic, programarea, mutarea, anularea și vizualizarea agendei. Scorul mediu, de 80,0, este peste media de referință de 68 (Tabelul 7, Figura 7).

Tabelul 7. Evaluarea utilizabilității cu chestionarul SUS (10 participanți)

Grup	Participanți	Scor SUS mediu	Interpretare
Pacienți	6	82,5	Bun spre excelent
Personal medical	4	76,3	Bun
Total	10	80,0	Bun

Sursa: Testare cu utilizatori, chestionarul SUS [6] (valori simulate pentru exemplu).

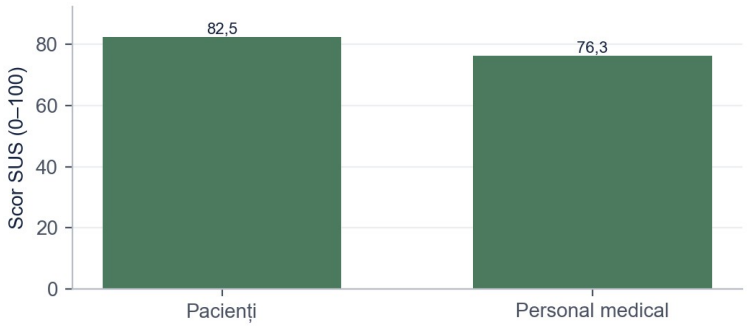


Figura 7. Scorul SUS mediu, pe grupuri de participanți

Sursa: Prelucrare proprie după Tabelul 7.

Pentru testarea cu utilizatori ai nevoie de acordul participanților și de un chestionar standard (SUS are 10 întrebări și un mod de calcul fix). Spune câți participanți ai avut și ce sarcini au făcut.
Ghid: licenta-disertatie-romania.co
Sfat
20/05/2026 10:00

4.4. Evaluarea autorului

Tabelul 8 reia cerințele din Capitolul 2 și le compară cu rezultatele. Cinci din șase cerințe sunt îndeplinite complet; protecția datelor este îndeplinită doar parțial, pentru că măsurile tehnice nu au fost verificate printr-un audit extern.

Tabelul 8. Evaluarea autorului: obiectivele și cerințele nefuncționale față de rezultate

Cerința	Criteriul	Rezultatul obținut	Evaluare
Cerințele funcționale CF1–CF6	Implementate și testate	Toate implementate; CF6 testată doar în scenarii simulate	Îndeplinită
Timp de răspuns	Sub 200 ms la 100 de utilizatori	61 ms (căutare), 96 ms (creare programare)	Îndeplinită
Fără programări duplicate	Test cu cereri simultane pe același interval	Testul de integrare a trecut; o singură programare creată	Îndeplinită
Funcționare pe telefon	Teste end-to-end pe ecran mic	12 din 12 teste trecute	Îndeplinită
Protecția datelor (GDPR)	Date minime, parole criptate, acces pe roluri	Implementate; fără audit extern	Parțial
Utilizabilitate	Scor SUS peste 68 (media de referință)	80,0	Îndeplinită

Sursa: Prelucrare proprie după Tabelele 3, 5, 6 și 7.

Considerăm că punctul cel mai solid al aplicației este mecanismul tranzacțional care împiedică programările duplicate, iar punctul cel mai slab este lista de așteptare (CF6), testată doar în scenarii simulate. Înainte de folosirea într-un cabinet real, recomandăm un test pilot de o lună și un audit de securitate.

Nu scrie „aplicația funcționează perfect”. Comisia apreciază mai mult o evaluare onestă, cu limitele spuse de tine, decât o listă de laude fără măsurători.
Greșeală frecventă
20/05/2026 10:00

Concluzii şi dezvoltări ulterioare

Concluziile arată ce ai realizat față de obiective, ce limite are aplicația și cum poate fi dezvoltată.
Ghid: licenta-disertatie-romania.co
Rolul secțiunii
20/05/2026 10:00

- Aplicația acoperă toate cele șase cerințe funcționale; lista de așteptare (CF6) a fost testată doar în scenarii simulate.
- Mecanismul cu tranzacții elimină programările duplicate, iar timpul de răspuns rămâne sub 200 ms până la 200 de utilizatori simultani.
- 183 din 184 de teste automate au trecut, iar scorul SUS mediu, de 80,0, arată o aplicație ușor de folosit.

Limite şi dezvoltări ulterioare: aplicația nu a fost folosită încă într-un cabinet real și nu a trecut printr-un audit de securitate. Ca dezvoltări propunem plăți online, integrarea cu Google Calendar și o aplicație mobilă.

Bibliografie

La IEEE sursele sunt numerotate în ordinea citării, nu alfabetic. Documentația PostgreSQL [5] și capitolul despre SUS [6] sunt reale; sursele [1]–[4] sunt generice, inventate pentru acest exemplu.
Rolul secțiunii
20/05/2026 10:00

[1] Aplicația A, „Programări online pentru clinici”, 2025. [Online]. Disponibil: https://exemplu-a.ro. [Accesat: 10 mai 2026].

[2] Aplicația B, „Documentație pentru cabinete”, 2025. [Online]. Disponibil: https://exemplu-b.ro. [Accesat: 10 mai 2026].

[3] A. Popa și D. Neagu, „Arhitecturi web pentru aplicații medicale”, *Revista de Informatică Aplicată*, vol. 4, nr. 2, pp. 33–41, 2023.

[4] M. Ilie, *Dezvoltarea aplicațiilor web cu Node.js*. Cluj-Napoca: Editura Exemplu, 2022.

[5] The PostgreSQL Global Development Group, „Explicit Locking”, *PostgreSQL Documentation*. [Online]. Disponibil: https://www.postgresql.org/docs/current/explicit-locking.html. [Accesat: 10 mai 2026].

[6] J. Brooke, „SUS: A ‚quick and dirty’ usability scale”, în *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, B. A. Weerdmeester și I. L. McClelland, Eds. Londra: Taylor & Francis, 1996, pp. 189–194.

Anexa 1. Capturi de ecran și codul sursă

În anexe pui capturi de ecran ale interfeței, manualul de utilizare și linkul către codul sursă. Verifică dacă facultatea cere codul pe CD, pe o platformă proprie sau pe GitHub.
Ghid: licenta-disertatie-romania.co
Rolul secțiunii
20/05/2026 10:00

Codul sursă complet: https://github.com/exemplu/programari-cabinet (adresă fictivă).

Instalare locală:

```
git clone https://github.com/exemplu/programari-cabinet
cd programari-cabinet
docker compose up
```

Bonus: același paragraf, înainte și după corectare

Această pagină nu face parte din lucrare. Arată cum se schimbă un paragraf la o corectare atentă: aceeași idee, scrisă clar, cu sursă și în stil academic.
Ghid: licenta-disertatie-romania.co
Rolul secțiunii
20/05/2026 10:00

Fragmentul de mai jos este scris intenționat cu greșelile pe care le întâlnim des în lucrările de licență la Informatică. Dedesubt este aceeași idee după corectare.

VARIANTA INIȚIALĂ

> Aplicatia este foarte rapida si sigura, am folosit cele mai noi tehnologii (React, Node, Postgres) care sunt cele mai bune de pe piata. Am testat-o si merge fara probleme, chiar si cu multi utilizatori. Codul este in poza de mai jos.

VARIANTA CORECTATĂ

> Aplicația a fost implementată cu React și TypeScript pe client, Node.js și Express pe server și PostgreSQL, alese pentru verificarea tipurilor, pentru limbajul comun pe client și server și pentru tranzacțiile care previn programările duplicate (Tabelul 2). Din cele 184 de teste automate, 183 au trecut (Tabelul 5). La un test de încărcare, timpul mediu de răspuns a rămas sub 200 ms până la 200 de utilizatori simultani (Figura 6). Fragmentul de cod relevant este redat ca text în Capitolul 3, iar codul complet este disponibil în repository.

CE AM CORECTAT

1. Am pus diacriticele (ș, ț cu virgulă), obligatorii în lucrare.
2. „Cele mai bune de pe piață” a devenit o justificare a alegerii, legată de cerințele proiectului.
3. „Merge fără probleme” a devenit un rezultat măsurat: numărul de teste trecute.
4. „Chiar și cu mulți utilizatori” a devenit un test de încărcare cu un prag și un număr de utilizatori.
5. Captura de ecran cu cod a fost înlocuită cu cod redat ca text și trimitere la repository.

Vrei aceeași corectare pentru lucrarea ta? licenta-disertatie-romania.com/contact