

**UNIVERSITATEA ROMÂNO-AMERICANĂ
FACULTATEA DE INFORMATICĂ MANAGERIALĂ**



LUCRARE DE LICENȚĂ

Coordonator științific:

lect.univ.dr. ENĂCEANU ALEXANDRU ȘERBAN

Absolvent:
CIUPERCĂ RADU-MIHAIL

București 2022

UNIVERSITATEA ROMÂNO-AMERICANĂ

FACULTATEA DE INFORMATICĂ MANAGERIALĂ



Proiectarea unei aplicații de gestionare a programărilor de pacienți în cadrul unității medicale particulare SC ROUTINE MED S.R.L

**Coordonator științific:
lect.univ.dr. ENĂCEANU ALEXANDRU ȘERBAN**

**Absolvent:
CIUPERCĂ RADU-MIHAIL**

Cuprins

1.	STUDIUL SI ANALIZA SISTEMULUI EXISTENT	3
1.1.	Prezentarea succinta a unității economico-sociale	3
1.2.	Activitățile desfășurate in unitatea economica	4
1.3.	Studiul sistemului de conducere	10
1.4.	Studiul sistemului condus	15
1.5.	Studiul sistemului informațional	16
1.5.1.	Schema fluxului informațional.....	16
1.5.2.	Descrierea circuitelor informaționale	18
1.5.3.	Documente utilizate conform Modelului Conceptual al Prelucrărilor.....	19
1.5.4.	Proceduri ale sistemului informațional în raport cu pacienții	22
1.5.5.	Analiza critică a sistemului actual și identificarea neajunsurilor existente in funcționarea sistemului existent	26
1.5.6.	Direcții de perfecționare a sistemului actual.....	26
2.	PROIECTAREA DETALIU A NOULUI SISTEM INFORMATIC	27
2.1.	Definirea obiectivelor aplicației	27
2.2.	Proiectarea logica si fizica a ieșirilor	28
2.2.1.	Lista rapoartelor de ieșire.....	28
2.2.2.	Machete ale documentelor de ieșire	30
2.3.	Proiectarea logica si fizica a intrărilor	35
2.3.1.	Lista documentelor si video formatelor de intrare	35
2.3.2.	Macheta video formatelor pentru preluare	36
2.4.	Proiectarea sistemului de codificare a datelor	39
2.5.	Proiectarea logica si fizica a bazei de date	40
2.6.	Schema de sistem a aplicației.....	48
2.7.	Proiectarea interfeței aplicației	49

2.7.1	Interfața Personal Conducere	50
2.7.2	Interfața Medic	60
2.7.3	Interfață recepționist	64
2.7.4.	Interfață Pacient	66
2.8.	Alegerea tehnologiei de prelucrare și a infrastructurii necesare pentru funcționarea optimă a aplicației.....	67
2.9.	Estimarea calendarului de resurse și a calendarului de realizare	68
3.	PREZENTAREA APLICATIEI SOFTWARE	70
3.1.	Cerințele platformei hardware și software ale produsului realizat.....	70
3.2.	Descrierea funcțiilor aplicației	71
3.2.1.	Implementarea programatică a sistemului de migrații.....	71
3.2.2.	Implementarea sub-sistemului de autentificare și securitate	73
3.2.3.	Implementarea sub-sistemului de programării.....	75
3.2.4.	Implementarea sub-sistemului de facturare	76
3.2.5.	Alte sub-sisteme ale aplicației	77
4.	EFICIENȚA ȘI UTILITATEA APLICAȚIEI.....	78
4.1.	Condiții privind implementarea aplicației.....	78
4.2.	Exploatarea curentă a aplicație.....	79
4.3.	Considerații privind eficiența aplicației	80
5.	BIBLIOGRAFIE.....	81
6.	ANEXE	82
6.1.	Anexe referitoare la documente și machete ale acestora.....	82
6.2.	Anexe referitoare la documentele generate de aplicația Informatică	87
6.3.	Anexe referitoare la codul sursă al aplicației	94

1. STUDIUL SI ANALIZA SISTEMULUI EXISTENT

1.1. Prezentarea succinta a unității economico-sociale

Societatea Comercială Routine Med S.R.L. este o clinică medicală din orașul Tulcea, cu o gamă largă de servicii din domeniul asistenței medicale de specialitate, ce ocupă o poziție de lider pe piața locală a serviciilor de asistență medicală și dorește extinderea și în alte județe.

Ca formă de organizare aceasta este o societate cu răspundere limitată, înființată în anul 2010 în localitatea Sulina, din județul Tulcea, având codul CAEN¹ 8621 – “Activități de asistență medicală generală”, deservind astfel comunitatea locală a orașului și totodată având șansa de a se dezvolta datorită poziționării geografice ce îngreunează accesul la alte instituții similare, în prezent având deschise 8 puncte de lucru plasate în tot județul Tulcea.

S.C. Routine Med S.R.L. are sediul în localitatea Sulina, județ Tulcea, în str. A II – A numărul 236A, și are numărul J36/202/2010 în cadrul registrului comerțului, având codul de identificare fiscală, 27028852, aceasta fiind plătită de TVA².

Deși are o gamă foarte largă de specialități medicale în cadrul firmei, aceasta și-a creat o imagine remarcabilă datorită laboratorului de analize medicale dotat cu tehnologie de ultimă generație, fapt ce a deschis o colaborare cu alte unități sanitare din regiunea sud-estică și centrală a țării. În contextul pandemiei de COVID-19, aceasta a avut unul din singurele laboratoare din județul Tulcea aprobate pentru testare de tip RT-PCR, fapt ce a dus la o creștere substanțială a încasărilor firmei.

În prezent conducerea Routine Med S.R.L. face eforturi mari pentru construirea unei unități medicale de tip spital pentru a putea trata și pacienții ce au nevoie de internare și intervenții chirurgicale.

¹ CAEN – Clasificarea Activităților în Economia Națională

² TVA – Taxă pe valoare adăugată

1.2. Activitățile desfășurate în unitatea economică

Routine Med S.R.L. are o gamă diversificată de servicii medicale, adresându-se atât persoanelor fizice pentru consultații, teste și analize medicale dar oferă și soluții pentru companii cum ar fi realizarea activităților aferente medicinei muncii, medicinei legale și activităților aferente asigurărilor de viață.

Firma Routine Med S.R.L. își propune satisfacerea nevoilor clientilor de servicii de asistență medicală generală din județul Tulcea, cu un grad ridicat de profesionalism și etică medicală prin următoarele obiective și anume:

- Desfășurare actului medical către pacienți în conformitate cu legislația în vigoare
- Educarea pacienților în domeniul sănătății de bază
- Flexibilitatea investigațiilor medicale prin selecție realizată de către pacient
- Accesibilitatea serviciilor prin pachete medicale generale și prin contracte de colaborare cu CNAS³

Obiectivele sunt atinse cu succes prin activități realizate de toate departamentele reușind astfel să ofere cea mai bună calitate a actului medical în raza de acțiune a județului Tulcea. Câteva din aceste activități sunt:

- Obținerea și menținerea gradului de acreditare ANMCS⁴ nivel 2 prin oferirea serviciilor medicale de înaltă calitate
- Efectuarea campaniilor de marketing privind educarea populației despre diverse afecțiuni reprezentative în perioada de timp
- Oferirea posibilității pacientului să își personalizeze serviciile medicale dorite în cadrul recepției unității.
- Realizarea de parteneriate cu instituțiile statului pentru decontarea parțială sau totală a serviciilor medicale.
- Realizarea de parteneriate cu companiile de asigurări private pentru decontarea cheltuielilor de către pacienți.

În figura cu numărul 1, putem vedea evoluția firmei Routine Med S.R.L. de la înființarea acesteia în anul 2010 până în prezent, observând astfel o evoluție continuă în

³ CNAS – Casa Națională de Asigurări de Sănătate

⁴ ANMCS - Autoritatea Națională de Management al Calității în Sănătate

ceea ce privește calitatea actului medical prin nivelul investițiilor efectuate în cadrul acesteia.

Figura 1 – S.C. ROUTINE MED S.R.L. – Evoluția firmei d.p.d.v. financiar

<i>Exercițiu Financiar</i>	<i>Cifra de afaceri</i>	<i>Profit Net</i>	<i>Datorii</i>	<i>Active Imobilizate</i>	<i>Active Circulante</i>
2010	30 284	430	7 562	0	8 192
2011	117 680	849	19 018	0	20 519
2012	201 594	1 007	97 091	0	99 599
2013	257 998	-	203 117	0	153 899
		50 59600			
2014	440 668	18 920	213 243	0	182 944
2015	643 537	9 189	221 050	0	199 940
2016	1 250 017	82 746	236 415	0	298 051
2017	2 252 050	149 180	294 902	0	445 717
2018	3 153 655	342 713	401 865	25 847	729 546
2019	3 710 864	36 076	659 743	223 165	526 182
2020	15 050 504	2 829 749	5 102 558	152 714	7 869 197
2021	20 087 675	7 228 162	1 796 029	1 150 727	8 275 092

Figura 2 – Evoluția profitului net comparată cu cea a cifrei de afaceri



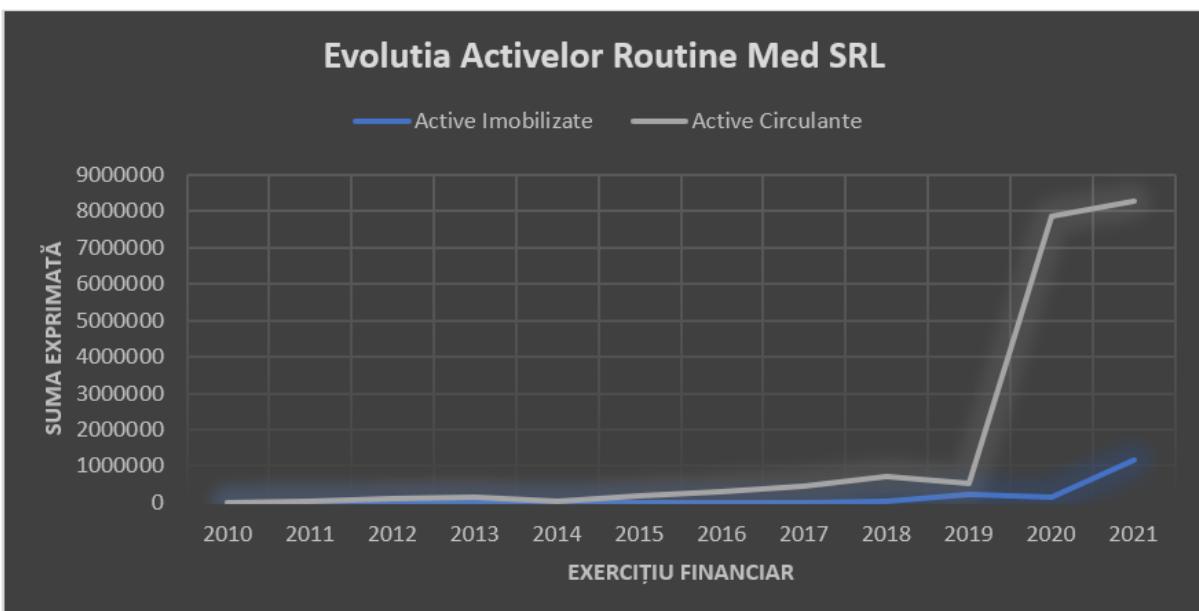
În tabelul din figura 1 am extras datele și am generat graficul din figura 2, unde observăm că de-a lungul existenței sale, Routine Med S.R.L. a avut o evoluție constantă în primii ani de activitate, și observăm inclusiv o pierdere de 50.596 RON în anul 2013, ceea ce ne spune că această societate comercială a trecut și prin perioade mai grele din punct de vedere financiar, dar odată cu creșterea investițiilor prin credite pentru achiziția de aparatură necesară observăm și un ritm mai accelerat de creștere al cifrei de afaceri și inclusiv al profitului.

Figura 3 – Istoricul Datoriilor al firmei ROUTINE MED S.R.L.



În graficul reprezentat în figura 3, putem observa o creștere treptată a datoriilor firmei până în anul 2020 inclusiv, ceea ce în primă instanță ne duce la un grad scăzut de încredere, dar privită prin prisma profitului net prezentat în figura anterioară putem presupune că aceste datorii sunt acumulate în scopul modernizării întreprinderii cu aparatură și utilaje noi. Odată cu exercițiul finanțier 2021, firma a reușit să își mai stingă din datoriile acumulate datorită unei creșteri accelerate a indicatorului cifra de afaceri cu circa 25 de puncte procentuale

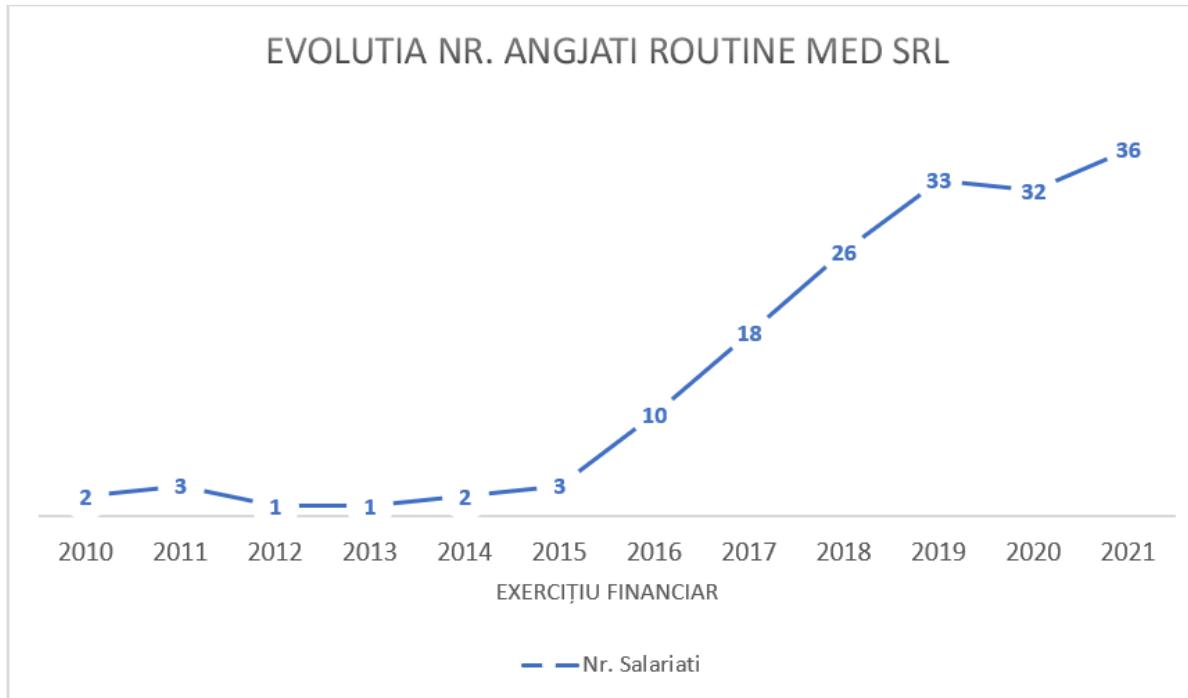
Figura 4 – Evoluția activelor din firma Routine Med S.R.L.



În figura 4, identificăm evoluția activelor firmei Routine Med S.R.L., observând astfel că aceasta nu a făcut investiții foarte mari cu amortizare de lungă durată, comparativ cu elementele de rulaj cum ar fi stocurile și materialele consumabile. Aceasta lucru ne indică un grad mare de lichiditate al firmei și o eficiență extrem de bună în utilizarea resurselor.

În anul 2021 observăm o creștere destul de alertă a activelor imobilizate, ceea ce înseamnă că, compania a făcut un set de investiții de lungă durată în bunuri cum ar fi proprietăți, terenuri, brevete, licențe și lista poate continua.

Figura 6 – Evoluția numărului mediu de angajați din firma Routine Med S.R.L.

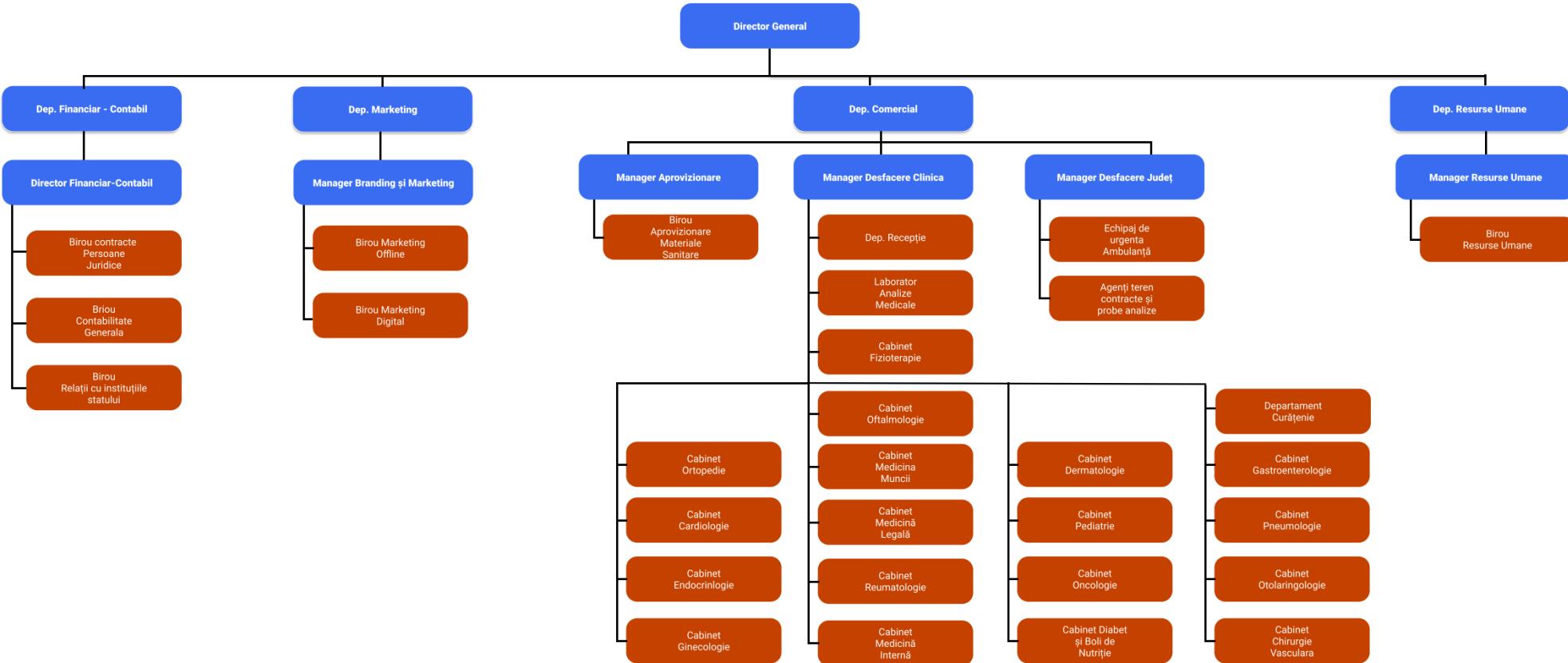


În figura 6 putem observa evoluția firmei Routine Med S.R.L. pe parcursul activității sale și observăm că aceasta și-a consolidat echipa în anul 2016 când, a trecut de la un număr de 3 salariați la 10 salariați. În prezent aceasta are un număr mediu de 36 de angajați, ceea ce ne indică că firma are o anume stabilitate și este în continuă creștere economică.

1.3. Studiul sistemului de conducere

Sistemul de conducere joacă un rol crucial în dezvoltarea unei firme, astfel o analiză amănunțită a acestuia ne poate ajuta să stabilim cu exactitate către ce părți componente din firmă trebuie să realizăm rapoartele generate de o aplicație economică. Pentru a identifica mai facil sistemul de conducere în figura 7 avem realizată organograma firmei Routine Med S.R.L.

Figura 7 – Organigrama firmei Routine Med S.R.L.



În fruntea organigramei din figura 7 este poziționat Directorul General care este persoana cu cea mai mare responsabilitate din firmă, fiind poziționat pe primul nivel ierarhic al organigramei. **Responsabilitățile Directorului General pot fi:**

- Relațiile dintre firma Routine Med S.R.L. și companiile terțe în negocierea contractelor
- Coordonează persoanele din funcțiile de conducere ale firmei.
- Se ocupă de luarea deciziilor firmei.
- Analizează împreună cu directorul de la departamentul finanțier contabil ofertele de colaborare pentru dezvoltarea firmei
- Reprezintă firma în relațiile cu statul asistat de Directorul departamentului Finanțier-Contabil
- Aprobă bugetele de marketing pentru dezvoltarea campaniilor de publicitate
- Se ocupă de prelucrarea feedback-ului împreună cu Directorul departamentului de Marketing
- Se ocupă de aplanarea situațiilor neclare sau tensionate referitor la pacienți.
- Participă la interviurile de angajare alături de Directorul departamentului de Resurse Umane

În cadrul departamentului finanțier contabil avem de a face cu **Directorul Finanțier – Contabil**, responsabilitățile acestuia fiind următoarele:

- Se asigură de realizarea la timp și fără erori a documentelor cu caracter fiscal în relația cu statul
- Se asigură că sunt respectate termenele de plată pentru serviciile din partea terților.
- Se asigură că contractele economice întocmite în numele organizației pentru clienți sunt corecte și complete.
- Ține evidență indicatorilor economici din firmă cum ar fi, Cifra de afaceri, Profit etc.
- Coordonează activitatea în departamentul pe care îl conduce.

În cadrul departamentului de marketing avem de a face cu **Manager-ul de Branding și Marketing**, responsabilitățile acestuia fiind următoarele:

- Se asigură că brand-ul Routine Med nu este folosit de terțe persoane fără acordul firmei
- Coordonează activitatea în departamentul pe care îl conduce.
- Verifică campaniile de marketing realizate de angajați și aprobă bugetele pentru acestea.

- Analizează piața de profil și întocmește rapoarte cu metode de perfecționare a activității.
- Se asigură de prelucrarea corectă și completă a feedback-ului clientilor.
- În cazul unui feedback negativ din partea clientilor demarează o strategie de perfecționare.

În cadrul departamentului de comercial avem de a face cu **Manager-ul de aprovisionare**, responsabilitățile acestuia fiind următoarele:

- Se asigură că există stocuri de materii prime și materiale consumabile pentru orice activitate din firmă.
- Să caute furnizori pentru materiile prime necesare firmei
- Să negocieze contractele de aprovisionare într-un mod avantajos firmei
- Să se ocupe de logistica materiilor prime și a materialelor consumabile.

Tot în departamentul comercial întâlnim funcția de **Managere Desfacere Clinică** care se ocupă de toate cabinetele ce sunt în cadrul clinici din Str. Mahmudie Nr 49-51, din localitatea Tulcea, județul Tulcea. Printre atribuțiile acestuia se numără următoarele

- Să asigure oricărui medic și asistent materialele necesare desfășurării activității.
- Să se asigure că personalul din departamentului *Recepție* se descurcă cu fluxul de clienți și să se suplimeze numărul de oameni pentru perioade scurte de timp dacă este necesar.
- Să se asigure că departamentul *Curățenie* își duce la bun sfârșit sarcinile în conformitate cu normele sanitare impuse pentru instituție
- Să se asigure că Laboratorul de analize medicale primește mostrele la timp și în stare intactă pentru prelucrare
- Să se asigure că angajații Laboratorului de analize nu suferă de vreo boală care ar putea altera indirect rezultatul analizelor.
- Să se asigure că, consultațiile medicilor coincid cu condica de programări
- Să verifice starea cabinetelor după terminarea programului de consultații și să cheme echipele de tehnic partenere pentru rezolvarea acestora în timp util.
- Să stea la dispoziția organelor de control și să ofere informații despre secția pe care o conduce

Ultima funcție de conducere din departamentul comercial este cea a **Managerului de Desfacere pentru activitățile din județ**. Printre responsabilitățile acestuia se numără:

- Să pună la dispoziția conducerilor auto mașinile în stare bună de funcționare

- Să verifice starea autoturismelor și a autoutilitarelor din firmă atât la începerea programului cât și la terminarea acestuia.
- Să preia și să verifice corectitudinea foilor de parcurs ale angajaților și să le predea biroului de contabilitate generală.
- Să întocmească referat de necesitate în cazul avariilor unui vehicul
- Să previzioneze și să optimizeze rutele de transport
- Să previzioneze și să optimizeze consumul de carburant
- Să se asigure că vehiculele firmei sunt în regulă din punct de vedere legal (Inspecție tehnică periodică, Rovignetă, Asigurări)
- Să ofere echipajului de ambulanță toate dotările necesare pentru a putea pleca în misiune (Butelii oxigen, Încărcare defibrilator, asigurarea unui stoc minim de medicamente)
- Să execute controale prin intermediul aplicațiilor de monitorizare a vehiculelor prin tehnologia GPS

Departamentul de resurse umane este condus de un **Manager de resurse umane** care are ca sarcini următoarele repere:

- Să se asigure că volumul de muncă este acoperit în totalitate de personalul necesar
- Să previzioneze nevoia de personal și să o acopere în timp util
- Să comunice cu celelalte departamente pentru a afla nevoia de personal
- Să realizeze campanii de recrutare când firma are un deficit de resurse umane
- Să se realizeze campanii de formare profesională în domeniu pentru angajații firmei
- Să ofere departamentului finanțier contabil informațiile necesare pentru operarea acestora în software-ul REVISAL pentru gestiunea salariaților.

1.4. Studiul sistemului condus

Orice firmă are nevoie de angajați pentru efectuarea activităților de zi cu zi. Odată analizat sistemul de conducere al firmei S.C. Routine Med S.R.L., observăm că acesta este foarte stufoș din punct de vedere al comportamentelor, și pentru că ne propunem proiectarea unui sistem informatic pentru departamentul recepție și pentru cabinetele de oftalmologie și fizioterapie, le vom analiza doar pe acestea la nivelul al sistemului condus.

Departamentul **Recepție** este unul cheie în realizarea activității firmei, deoarece reprezintă punctul de intrare al datelor în sistemul informațional și ne propunem prelucrarea acestor date printr-un sistem informatic. În prezent, atribuțiile personalului din cadrul acestui departament sunt următoarele

- Să ofere informații pacienților despre programări, medicii existenți, locația cabinetelor, atât în clinică cât și prin apel telefonic
- Să se asigure că pacientul are o programare sau să efectueze una dacă este cazul
- Să acționeze butonul de panică dacă există o situație ce nu se poate rezolva pașnic
- Să introducă datele în sistemul informatic actual și să verifice corectitudinea acestora
- Să verifice și să încaseze sumele de bani aferente serviciilor medicale cu taxă
- Să emită factură la cerere cu ajutorul unui sistem informatic deja existent
- Să anunțe pacienții viitori despre programări și să le confirme prin apel telefon

Cabinetele de **fizioterapie** și **oftalmologie** sunt formate similar și au aceleași responsabilități atât față de pacienți cât și față de sistemul de conducere al firme și anume

- Să ofere cel mai bun diagnostic posibil pentru simptomele prezентate de pacienți
- Să ofere cel mai bun tratament bazat pe diagnosticul oferit
- Să întocmească documentele necesare derulării activității împreună cu scrisoarea medicală
- La cerere să explice pacientului termenii necesari înțelegerei diagnosticului
- Să păstreze confidențialitatea medic-pacient
- Să anunțe departamentul recepție dacă s-au efectuat și alte operațiuni pe lângă cele din programare

1.5. Studiul sistemului informațional

Dat fiind faptul că activitatea a firmei noastre se fundamentează pe sistemul informațional, este bine să efectuăm un studiu detaliat al acestuia, un sistem bine pus la punct oferindu-ne avantaje în etapa de proiectare și dezvoltare a acestuia, și putem eficientiza la maxim procesele de prelucrare a datelor și relațiile atât din punctul sistemului condus către cel de conducere cât și cele dintre pacienți și clinică.

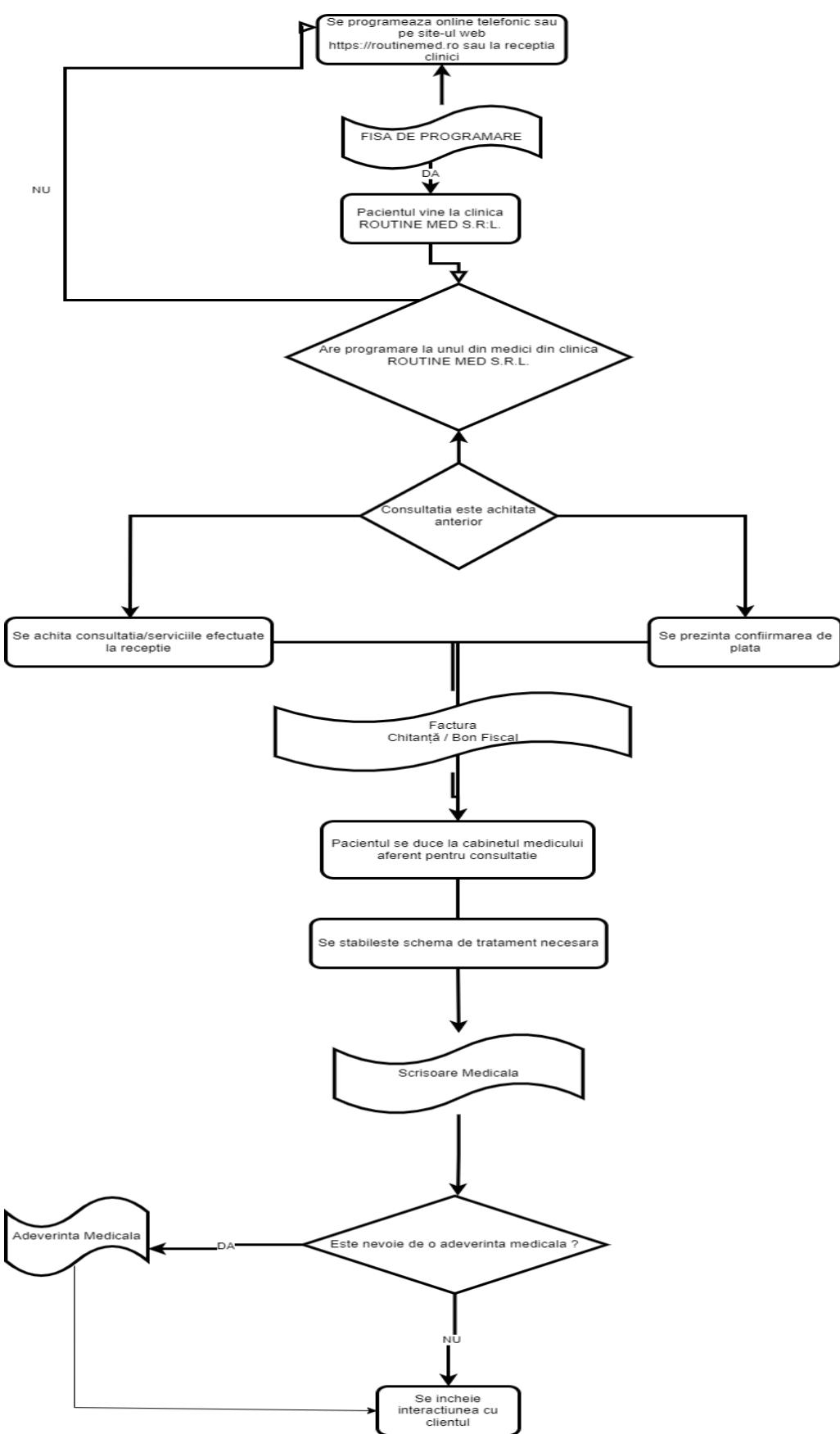
Pentru a efectua acest studiu al sistemului informațional, este nevoie să observăm acțiunile și deciziile care se iau atât în interiorul firmei cât și în relația cu clientul din punct de vedere obiectiv, și să observăm cum sunt prelucrate datele de pe diferiți suporti de memorie fie ieși digitali sau clasici.

1.5.1. Schema fluxului informațional

Cu ajutorul schemei fluxului informațional putem identifica mult mai ușor modalitatea prin care clinica va interacționa cu pacienții din momentul programării telefonice până la finalizarea consultului, identificând astfel și documentele emise pentru fiecare situație în parte.

În Figura 8 avem prezentată schema fluxului informațional.

Figura 8 – Schema fluxului informațional



1.5.2. Descrierea circuitelor informaționale

În figura 9 avem prezentat circuit informațional al firmei referitor la parcursul documentelor. Putem observa că procesul de programare începe cu o fisă de programare după care continuăm cu emiterea documentelor medicale și anume, scrisoarea medicală și adeverința medicală (dacă este cazul), iar în final, în baza onorariului medicului se stabilește valoarea facturii și modalitatea de plată a acesteia.

Figura 9 – Descrierea circuitelor informaționale pentru cabinetele din cadrul firmei

Număr Etapă	Denumire Etapă	Document emis
1	Pacientul se programează pentru o consultăție	Fișă de programare
2	Pacientul este examinat de un specialist	Scrisoare Medicală
3	Pacientul solicită adeverință medicală	Adeverință medicală
4	Pacientul se prezintă la clinică și achită consultația	Factură Fiscală Chitanță/Bon Fiscal

1.5.3. Documente utilizate conform Modelului Conceptual al Prelucrărilor

Toată activitatea unei firme se poate identifica prin documentele pe care le emite, ca urmare a proceselor din acestea. În cazul de față, Routine Med S.R.L., pentru secțiile recepție, oftalmologie și fizioterapie, folosește atât documente cu caracter fiscal cum ar fi Factura Fiscală, Chitanța, cât și documente cu caracter medical.

O să analizăm pe rând fiecare document cu care avem de a face pentru a putea trasa o schemă a procedurilor utilizate și să efectuăm o analiză a sistemului actual.

Primul document cu care avem de a face este fișa de programare a pacientului. Aceasta se întocmește de către un operator din departamentul recepție, și se realizează pentru a rezerva un loc la clinică la un moment dat. Documentul se întocmește cu ajutorul aplicația ICMed, un soft dedicat management-ului programări pacienților.

În cadrul fișei de programare identificăm numele pacientului, specializarea la care se programează, data și ora la care se programează pacientului, medicul specialist, canal de comunicare cu clientul și statusul de plată al rezervări. Identificăm fișa de programare pe baza unei serii și un număr de identificare. În Anexa 1, fișa de programare are seria MAH iar numărul 89329.

Un element central pentru orice societate comercială este factura fiscală. Pachetul software ICMed menționat anterior generează și factura fiscală aferentă programărilor.

În cadrul documentului identificăm datele fiscale ale firmei RoutineMed și ale clientului său, împreună cu produsele oferite sau serviciile prestate de firmă. Factura fiscală se identifică printr-o serie și un număr, combinația acestora fiind cheia de identificare a documentului. La final observăm că avem și un responsabil de facturare, de obicei acesta fiind angajatul din departamentul recepție care a emis documentul.

Acest document se emite doar către persoanele juridice. În Anexa numărul 2 putem identifica o factură fiscală a firmei RoutineMed S.R.L.

Chitanța este un document ce vine în complementarea facturii fiscale în cazurile în care o persoană juridică dorește să achite serviciile prestate de clinică în numerar.

Pe chitanță avem datele de identificare ale firmei Routine Med S.R.L. și ale clientului acesta, împreună cu motivul pentru care se emite aceasta cum ar fi plata serviciilor sau restituirea unei sume de bani.

Scrisoarea medicală este unul din documentele centrale din activitatea unei instituții cu caracter medical, deoarece în ea sunt trecute datele consultului, împreună cu tratamentele prescrise. Aceasta cuprinde elementele de identificare ale pacientului, cum ar fi nume, CNP, data nașterii, cat și elemente specializate cum ar fi, diagnosticul, anamneza⁵ examene medicale efectuate dar și tratamentul prescris de acesta.

Se întocmește la finalizarea consultației în dublu exemplar, unul către client iar celălalt pentru arhiva medicului, pentru efectuarea unui dosar cu afecțiunile pacientului. În figura 10 putem identifica o scrisoare medicală emisă la secția oftalmologie.

Figura 10– Scrisoare medicală

Denumire Furnizor		
Medic		
Contract /convenție nr.		
CAS		
ANEXA 43		
SCRISOARE MEDICALĂ *		
<p>Stimătore(ă) coleg(a), vă informăm că CNP/cod unic de asigurare nr. F.O./nr. din Registrul de consultații Motivele prezintării</p> <p>născut la data de ... / /aux, a fost consultat în serviciul nostru la data de : ...</p>		
<p>Diagnosticul:</p> <p>WOD - 09 f. 1st / 05.08.10</p> <p>Anamneza:</p> <p>- factori de risc</p> <p>Examen clinic:</p> <p>- general</p> <p>- local</p> <p>Examene de laborator:</p> <p>- cu valori normale</p> <p>- cu valori patologice</p> <p>Examene paraclinice</p> <p>EKG</p> <p>ECG</p> <p>Rx</p> <p>Altele</p> <p>Tratament efectuat</p> <p>Alte informații referitoare la starea de sănătate a asiguratului:</p> <p>Corecte păravînto. Control pînă 6 luni</p>		
<p>Tratament recomandat</p> <p>S-a eliberat prescripție medicală, ca în care se va înscrive seria și numărul acesteia</p> <p>S-a eliberat prescripție medicală deoarece nu a fost necesar</p> <p>S-a eliberat prescripție medicală</p> <p>S-a eliberat concediu medical la extemare, ca în care se va înscrive seria și numărul acestuia</p> <p>S-a eliberat concediu medical la extemare deoarece nu a fost necesar</p> <p>S-a eliberat concediu medical la extemare</p> <p>S-a eliberat recomandare pentru îngrijiri medicale la domiciliu/paliative la domiciliu</p> <p>S-a eliberat recomandare pentru îngrijiri medicale la domiciliu/paliative la domiciliu, deoarece nu a fost necesar</p> <p>S-a eliberat prescripție medicală pentru dispozitive medicale în ambulatoriu</p> <p>S-a eliberat prescripție medicală pentru dispozitive medicale în ambulatoriu deoarece nu a fost necesar</p> <p>(cu viză Unității Județene de Implementare a programului, pentru diabet)</p> <p>Unitate județeană de diabet zaharat:</p> <p>Nr. înregistrare a asiguratului:</p> <p>Data ... 29.07.2010</p> <p>Calea de transmisie: <input type="checkbox"/> prin asigurat <input type="checkbox"/> prin postă</p> <p>Semnătura și parafă medicului</p> <p>Dr. M. ASĂU med. clin. SOU</p>		

*Scrisoarea medicală se întocmește în două exemplare, din care un exemplar rămâne la medicul care a efectuat consultația/serviciul în ambulatoriu de specialitate, iar un exemplar este trimis medicului de familie/medicului de specialitate din ambulatoriu de specialitate, direct ori prin intermediu asiguratului.

⁵ Anamneza – Simptomele resimțite de pacient și transmise doctorului pentru a putea formula un diagnostic în cadrul examinării

Figura 11 – Adeverința medicală

Județul <u>TL</u>	Nr. fișă carnet de sănătate
Localitatea <u>TL</u>	<u>824/24.09.2018</u>
Unitatea sanitată <u>Rovimé MED</u>	
ADEVERINȚĂ MEDICALĂ	
Se adeverește că: _____	sexul <u>M</u>
Data nașterii: <u>1982</u> luna <u>12</u> ziua <u>12</u> cu domiciliul în	<u>F</u>
județul <u>TL</u> localitatea <u>Ciovorlengi str. Principala</u>	
nr. <u>81</u> având ocupația de: <u>Economiști la S.C. BACIU S.R.L.</u>	
Este suferind de: <u>Epilepsie și ghem medular</u>	
Se recomandă <u>Rezpiratori fizioterapie</u>	
S-a eliberat pentru a-i servi la: <u>Locul de muncă - C. M.</u>	
Data eliberării:	Semnătura și parafa medicului:
<u>29</u> luna <u>11</u> ziua <u>09</u> L.S. <u>J</u>	
Tipografie GRAPHOTEK EXPRESS, TECUCL, 0236.812.144	

Adeverința medicală este un alt document foarte utilizat în domeniul medical deoarece permite comunicarea medicului cu diverse instituții publice sau private în interesul pacientului.

1.5.4. Proceduri ale sistemului informațional în raport cu pacienții

Deoarece, Routine Med este o firmă ce are o puternică interacțiune cu clienții, dat fiind domeniul de activitate în care se situează, aceasta are un set de proceduri foarte bine puse la punct în ceea ce privește fluxul informațional, astfel se încearcă pe cât posibil eficientizarea folosirii resurselor și reducerea timpului de așteptare pentru clienți.

Procedurile firmei în ceea ce privește sistemul informațional în raport cu pacienții, pornesc de la faza de programare a clientului și se finalizează în faza de fidelizare, plecând de la nevoia clientului pentru o consultație medicală, acestea fiind structurate astfel:

Eveniment 1. Pacientul se programează la clinică prin departamentul recepție

Eveniment 2. Firma înregistrează programarea și o confirmă

Eveniment 3. Firma notifică clientul cu 24 de ore în prealabil referitor la programare

Eveniment 4. Pacientul se prezintă la clinică și confirmă rezervarea cu un act de identitate

Eveniment 5. Se verifică dacă programarea este plătită anterior

Eveniment 6. Se plătește programarea

Eveniment 7. Firma comunica pacientului datele acestuia pentru verificare și confirmare

Eveniment 8. Firma solicită acceptul pacientului referitor la campaniile de fidelizare

Eveniment 9. Pacientul acceptă să primească campanii de fidelizare

Eveniment 10. Pacientul refuză să primească campanii de fidelizare

Eveniment 11. Pacientul este direcționat către cabinetul medicului curant

Eveniment 12. Medicul își prestează actul medical de consultăție și completează documentele aferente actului medical

Eveniment 13. Medicul recomandă data următorului consult medical

Eveniment 14. Pacientul acceptă programarea cu data stabilită și este direcționat către recepție pentru o nouă programare

Eveniment 15. Pacientul nu acceptă data stabilită

Eveniment 16. Se finalizează interacțiunea cu clientul.

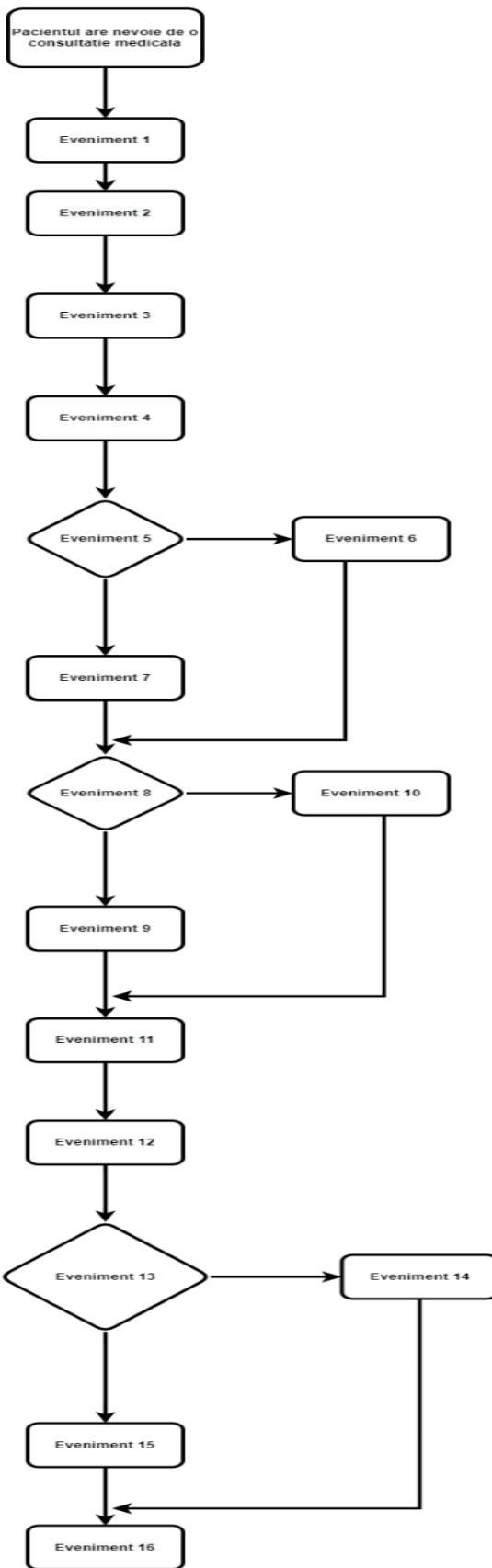
Pentru a înțelege mai bine aceste noțiuni ne propunem să efectuăm și un tablou de evenimente prin care putem identifica mai bine evenimentele declanșatoare ale acțiunilor și consecințele acestora. Acesta este prezentat în figura 12 iar în figura 13 avem modelul prelucrărilor conceptuale realizat sub formă de schemă logică.

Figura 12 – Tabel al evenimentelor din cadrul MCP

Eveniment Declanșator	Evenimentul curent	Consecința Evenimentului
Nevoia clientului pentru o consultație medicală	Eveniment 1 Pacientul se programează la clinică prin departamentul recepție	Eveniment 2
Eveniment 1	Eveniment 2 Firma înregistrează programarea și o confirmă	Eveniment 3
Eveniment 2	Eveniment 3 Firma notifică clientul cu 24 de ore în prealabil referitor la programare	Eveniment 4
Eveniment 3	Eveniment 4 Pacientul se prezintă la clinică și confirmă rezervarea cu un act de identitate	Eveniment 5
Eveniment 4	Eveniment 5 Se verifică dacă programarea este plătită anterior	Eveniment 6 sau Eveniment 7
Eveniment 5	Eveniment 6 Se plătește programarea	Eveniment 7
Eveniment 5	Eveniment 7 Firma comunica pacientului datele acestuia pentru verificare și confirmare	Eveniment 8
Eveniment 7	Eveniment 8 Firma solicită acceptul pacientului referitor la campanile de fidelizare	Eveniment 9 sau Eveniment 10
Eveniment 8	Eveniment 9 Pacientul acceptă să primească campanii de fidelizare	Eveniment 11
Eveniment 8	Eveniment 10 Pacientul refuză să primească campanii de fidelizare	Eveniment 11

Eveniment 9 sau Eveniment 10	Eveniment 11 Pacientul este direcționat către cabinetul medicului curant	Eveniment 12
Eveniment 11	Eveniment 12 Medicul își prestează actul medical de consultație și completează documentele aferente actului medical	Eveniment 13
Eveniment 12	Eveniment 13 Medicul recomandă data următorului consult medical	Eveniment 14 sau Eveniment 15
Eveniment 13	Eveniment 14 Pacientul acceptă data stabilită și este direcționat către recepție pentru o nouă programare	Eveniment 16
Eveniment 13	Eveniment 15 Pacientul nu acceptă data stabilită	Eveniment 16
Eveniment 14 sau Eveniment 15	Eveniment 16 Se finalizează interacțiunea cu clientul.	-

Figura 13 – Modelul prelucrărilor conceptuale



1.5.5. Analiza critică a sistemului actual și identificarea neajunsurilor existente în funcționarea sistemului existent

Deoarece nu există sistem informatic perfect ci doar perfectibil, nici sistem folosit de Routine Med S.R.L. nu este lipsit de neajunsuri, acestea fiind explicate pe larg în cele ce urmează.

O primă problemă identificată este realizarea și operarea documentelor medicale pe suporti de hârtie, pe baza şablonelor oferite de către instituțiile publice aferente sectorului de activitate fără a exista un sistem informatic pentru procesarea acestora în viitor, atât pentru a oferi pacientului un act medical mai calitativ, având disponibil cu lejeritate un istoric medical cât și operarea cu costuri mai mici permitând chiar trimiterea documentelor în format digital pentru clienți cu competențe în acest domeniu.

Altă problemă identificată este utilizarea mai multor pachete software cu caracteristici similare pentru desfășurarea activității în clinica medicală, existând un pachet software dedicat programărilor medicilor, acesta având inclusiv funcțiile de facturare, un pachet software dedicat analizelor de laborator care are și modul de programare a pacienților și a mostrelor, și lista poate continua.

Istoricul medical al unui pacient este greu de accesat, durând până la 2-3 zile pentru prelucrarea unei cererii pentru eliberarea unui document de la arhivă.

1.5.6. Direcții de perfecționare a sistemului actual

Bazat pe analiza critică a sistemului actual am ajuns la următoarele elemente ce pot fi îmbunătățite prin realizarea unui sistem informatic sau îmbunătățirea celui actual și anume, realizarea unei aplicații pentru cabinete individuale și clinici cu următoarele caracteristici

- Modul finanțiar cu posibilitate de dezactivare pentru lucru cu alte sisteme informaticice de gestiune finanțiară
- Implementarea unui sistem de programări dinamic, bazat pe orele alocate de medic conform contract de colaborare.
- Implementarea unui sistem prin care pacienți să se poată programa online și să își poată descărca mereu documentele emise de medicul curant
- Îmbinarea site-ului web și a rutinelor de verificare a pacienților
- Implementarea unei interfețe dedicat nivelelor ierarhice superioare pentru generarea de rapoarte

2. PROIECTAREA DETALIU A NOULUI SISTEM INFORMATIC

2.1. Definirea obiectivelor aplicației

Ne propunem ca, noul sistem informatic să fie unul care să pună în valoare capabilitățile tehnologice din prezent, aplicația fiind construită în jurul interacțiunilor cu pacienții, oferindu-le cele mai bune servicii medicale într-un timp optim. Totodată nu trebuie să neglijăm nivelele ierarhice superioare care vor putea genera rapoarte în orice moment și pot vedea evoluția în timp real a situații din clinică.

Aplicația informatică își propune oferirea unei experiențe ușoare de utilizare și are următoarele obiective pe care își propune să le atingă:

- Posibilitatea pacienților de ași putea accesa documentele emise de specialiști prin intermediul unei platforme web
- Eficientizarea procesului de introducere a datelor prin implementarea de echipamente hardware în departamentul recepție
- Realizarea unei interfețe facile pentru utilizatorii aplicației
- Posibilitatea folosirii unui modul terț pentru acțiunile economice
- Posibilitatea de a evalua actual medical în baza unor criterii bine stabilite

Aplicația informatică este gândită să ruleze pe o arhitectură de tip server-client conectat la o bază de date, astfel fiind diminuat necesarul de resurse utilizate de sistemul de calcul al clientului.

2.2. Proiectarea logica si fizica a ieșirilor

2.2.1. Lista documentelor de tip raport dedicate ieșirilor din aplicație

În cadrul aplicație noastre avem următoarele seturi de machete realizate pentru rapoarte de ieșire și anume:

Figura 14 – Lista rapoartelor de ieșire

Raport de ieșire	Descriere raport de ieșire	Figură Raport Ieșire
Fisă de programare	Acest raport este generat în urma confirmării programării în sistemul RoutineMed și conține informații despre aceasta.	Anexa 4
Chitanță	Document fiscal utilizat pentru a efectua operații cu numerar	Anexa 5
Factura Fiscală	Instrument finanțier dedicat persoanelor juridice pentru tranzacționarea de bunuri și servicii	Anexa 6
Scrisoarea Medicală	Document cu specific medical ce are scopul de a oferi informații către un alt specialist în domeniu despre starea pacientului. Se emite în dublu exemplar și se oferă clientului după fiecare consultație	Figura 15
Adeverința Medicală	Acest document se folosește pentru justificarea diferitelor probleme de natură medicală față de o instituție terță, publică sau privată.	Figura 16
Raport pacienți programati / Medic	Raport specializat creat pentru nivele ierarhice superioare dar și pentru medic-ul curent pentru a-și putea gestiona mai ușor pacienții pe o perioadă de timp selectată.	Figura 17
Raport pacienți programati / Cabinet	Raport specializat creat pentru nivele ierarhice superioare. Deoarece într-un cabinet pot fi mai mulți medici,	Figura 18

	putem observa performata acestora pe o perioadă de timp selectată.	
Raport pacienți programați / Clinică	Similar cu rapoartele anterioare, acesta cuprinde toți medicii de la toate specializările pe o perioadă de timp selectată.	Figura 19
Raport încasări persoane juridice	Raport dedicat departamentului finanțier contabil și pentru sistemul de conducere, care reflectă totalitatea încasărilor din partea persoanelor juridice pe o perioadă de timp selectată.	Figura 20
Raport prelucrare date în context de marketing	Prezenta machetă prezintă o parte din acordul privind prelucrarea datelor cu caracter personal în scop de marketing pentru oferirea unor servicii mai bune clienților	Figura 21

2.2.2. Machete ale documentelor de ieșire

Machetele aferente documentelor, fisă de programare, chitanță, factură fiscală sunt incluse în secțiunea de anexe a prezentei lucrări, corespondente acestora fiind anexele 4,5,6

Figura 15 – Machetă Scrisoare medicală

ROUTINE MED S.R.L. Medic Nume Medic	Cabinet: Oftalmologie
SCRISOARE MEDICALĂ	
Stimate(ă) coleg(ă), vă informăm că , născut la data de , CNP/cod unic de asigurare , a fost consultat în serviciul nostru la data de	
Motivele prezentării	
Diagnosticul	
Anamneză	
-factori de risc	
Examen clinic	
Examene de laborator	
-cu valori normale	
-cu valori patologice	
Examene Paraclinice	
EKG ECO Rx Altele	
Tratament efectuat	
Data 04.01.2022	Semnătura și parafa medicului

Figura 16 – Machetă Adeverință Medicală

Județul:
Localitatea:
Unitatea sanitară:

ADEVERINȚĂ MEDICALĂ

Se adeverește că:

Data nașterii: _____ luna _____ ziua _____ sexul _____ cu domiciliul în _____

județul _____ localitatea _____

adresa de: _____

având ocupația de: _____ la _____

Este suferind de: _____

Se recomandă _____

S-a eliberat pentru a-i servi la: _____

Data eliberării

Semnătura și parafa medicului:

L.S. _____

Figura 17 – Machetă Raport pacienți programăți / Medic

Nr Crt	Denumire Pacient	Data nașterii	Varsta	CNP	Data Programare	Ora Programare	Medic: Nume Medic
							Status Programare
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							
48							
49							
50							
51							
52							
53							
54							
55							
56							
57							
58							
59							
60							
61							
62							
63							
64							
65							
66							
67							
68							
69							
70							
71							
72							
73							
74							
75							
76							
77							
78							
79							
80							
81							
82							
83							
84							
85							
86							
87							
88							
89							
90							
91							
92							
93							
94							
95							
96							
97							
98							
99							
100							



Figura 18 – Machetă Raport pacienți programăți / Cabinet

Figura 19 – Machetă Raport pacienți programati / Clinică



Figura 20 – Machetă Raport încasări persoane juridice (Pe baza de FF)

Figura 21 – Machetă Raport acord prelucrare date în context de marketing

FORMULAR ACORD PRELUCRARE DATE ÎN CONTEXT DE MARKETING PENTRU CAMPANII DE FIDELIZARE

Subsemnatul **CLIENT** sunt de acord să mi se proceseze datele în scop de marketing în cadrul Routine MED S.R.L.

DATA

SEMNAȚURA

2.3. Proiectarea logica si fizica a intrarilor

2.3.1. Lista documentelor si video formatelor de intrare

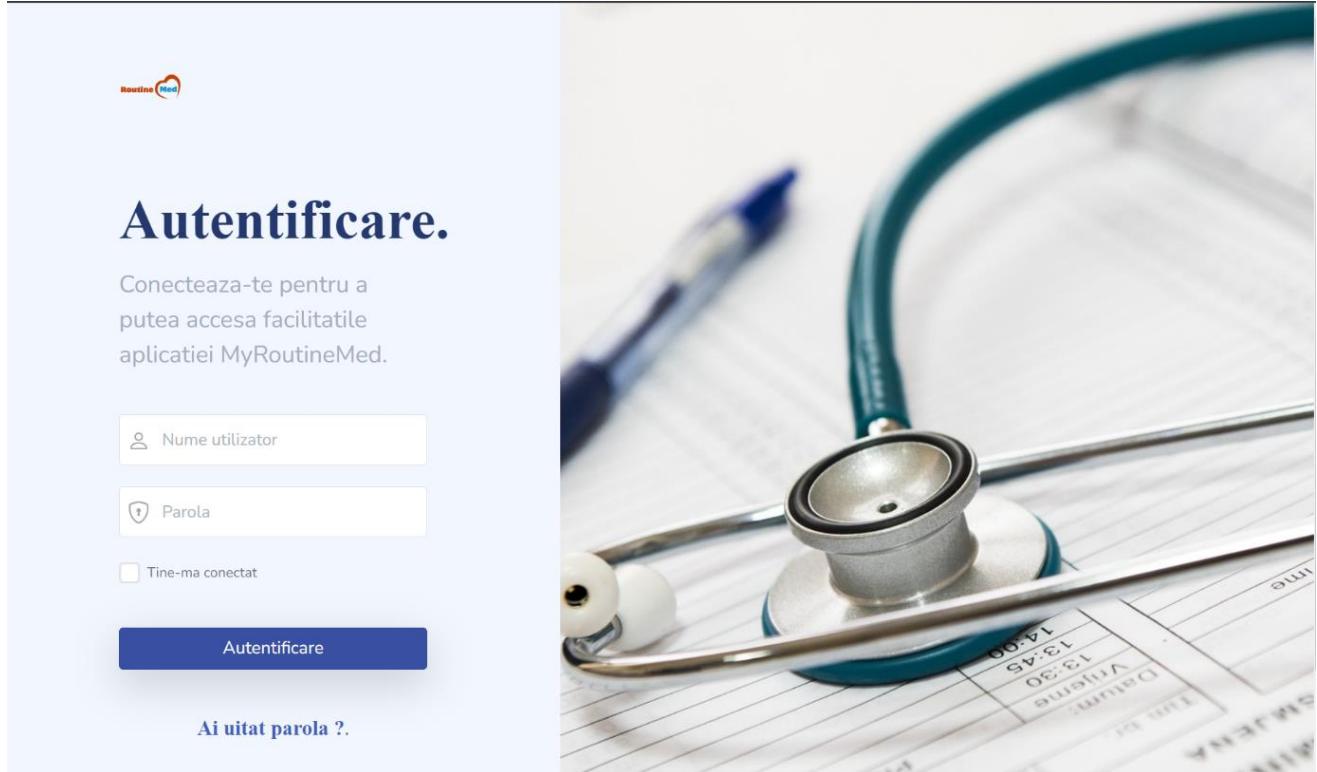
Video formatele de intrare sunt esențiale pentru desfășurarea activității noastre, astfel am realizat un set de machete pentru principalele funcții ale aplicației

Figura 22 – Lista machetelor pentru video formatele de intrare

Video format de intrare	Descriere raport de ieșire	Figură Video Format Intrare
Interfață autentificare	Facilitează accesul restricționat în aplicație	Figura 23
Înregistrare Utilizator faza 1	Video format dedicat introducerii datelor esențiale de conectare ale utilizatorului	Figura 24
Înregistrare Utilizator faza 2	Video format dedicat introducerii datelor utilizatorilor în baza de date. Folosim date cu caracter personal cum ar fi CNP-ul sau adresă deci trebuie să solicităm acordul pentru prelucrarea datelor	Figura 25
Înregistrare Factură	Este un video format complex format din două părți, un selector din care alegem datele de facturare și o căsuță de pre vizualizare a elementelor din factura, unde le introducem pe acestea	Figura 26
Înregistrare Programare	Fiind activitatea principală din cadrul clinicii acest video format este alcătuit din elemente de legătură cu entitățile bazei de date, fapt ce permite o completare mai rapidă a acestui	Figura 27

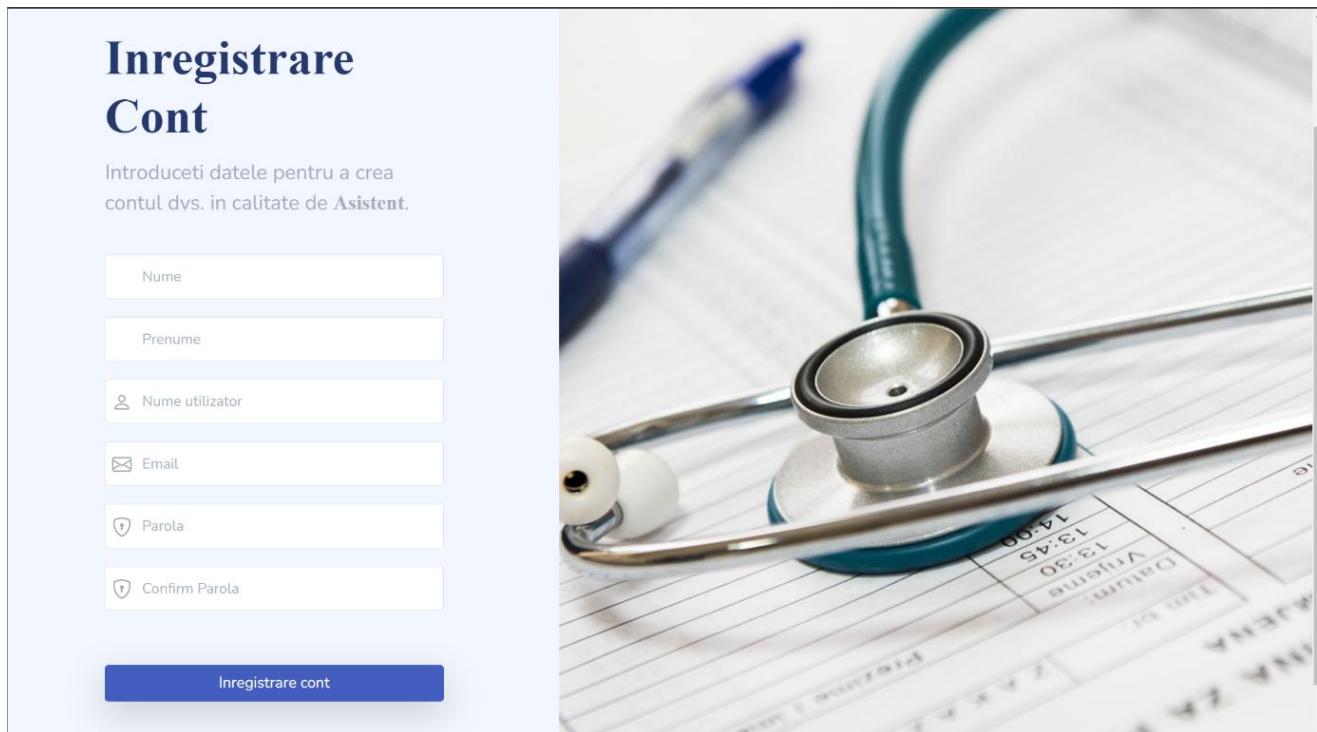
2.3.2. Macheta video formatelor pentru preluare

Figura 23 –Interfață Autentificare



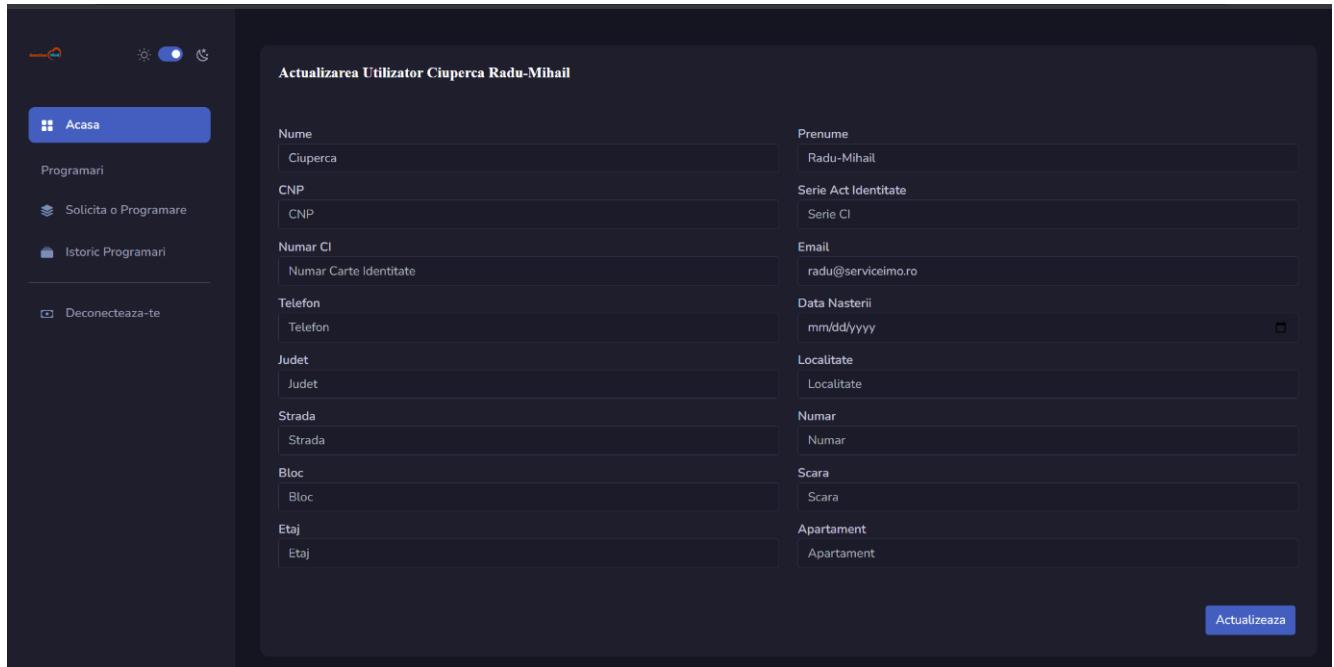
The image shows a login interface for the MyRoutineMed app. At the top left is the RoutineMed logo. The main title "Autentificare." is in large blue font. Below it is a subtitle: "Conecteaza-te pentru a putea accesa facilitatile aplicatiei MyRoutineMed." There are two input fields: "Nume utilizator" and "Parola", each with a corresponding icon (user and lock). Below these is a checkbox labeled "Tine-ma conectat". A blue button labeled "Autentificare" is at the bottom. A link "Ai uitat parola ?." is also present. To the right of the form is a photograph of a stethoscope lying on a medical chart.

Figura 24 – Interfață înregistrare faza 1



The image shows a registration form titled "Inregistrare Cont". The subtitle says "Introduceti datele pentru a crea contul dvs. in calitate de Asistent.". The form contains six input fields: "Nume", "Prenume", "Nume utilizator", "Email", "Parola", and "Confirm Parola". Each field has its respective icon (person, person, user, envelope, lock, lock). A blue button labeled "Inregistrare cont" is at the bottom. To the right of the form is a photograph of a stethoscope lying on a medical chart.

Figura 25 –Interfața înregistrare faza 2

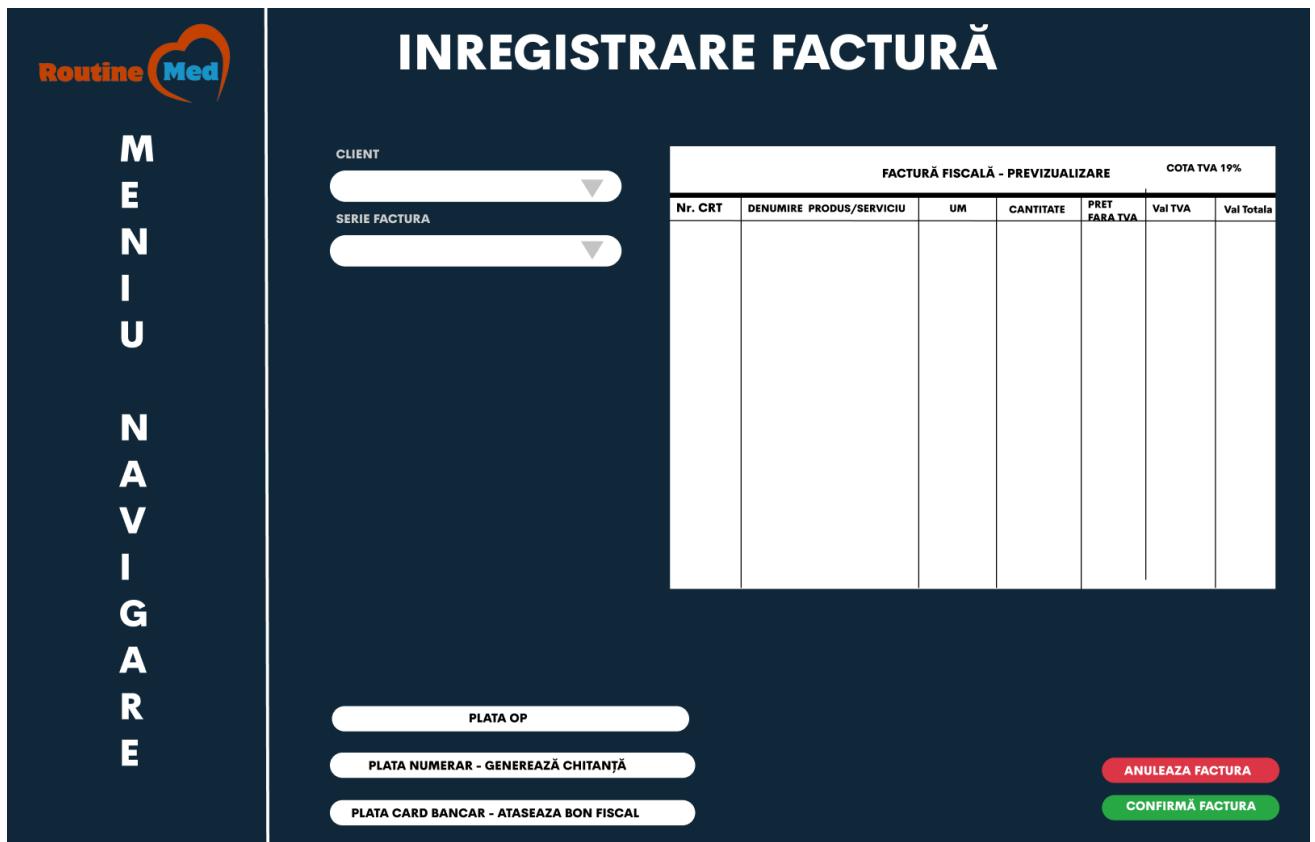


The screenshot shows a user profile update form titled "Actualizarea Utilizator Ciuperca Radu-Mihail". The form fields include:

- Name: Ciuperca, Surname: Radu-Mihail
- CNP: CNP
- Numar CI: Numar Carte Identitate
- Email: radu@serviceimo.ro
- Telefon: Telefon
- Data Nasterii: mm/dd/yyyy
- Judet: Judet
- Localitate: Localitate
- Strada: Strada
- Numar: Numar
- Bloc: Bloc
- Scara: Scara
- Etaj: Etaj
- Apartament: Apartament

At the bottom right is a blue "Actualizeaza" button.

Figura 26 – Macheta interfață emitere factură



The interface is titled "INREGISTRARE FACTURĂ". On the left, there is a vertical menu labeled "MENIU NAVIGARE" and a logo for "Routine Med". The main area contains the following elements:

- CLIENT dropdown menu
- SERIE FACTURA dropdown menu
- FACTURĂ FISCALĂ - PREVIZUALIZARE table header with columns: Nr. CRT, DENUMIRE PRODUS/SERVICIU, UM, CANTITATE, PRET FARA TVA, Val TVA, Val Totala, and COTA TVA 19%
- PLATA OP button
- PLATA NUMERAR - GENEREAZĂ CHITANȚĂ button
- PLATA CARD BANCAR - ATASEAZĂ BON FISCAL button
- ANULEAZA FACTURA button
- CONFIRMĂ FACTURA button

Figura 27 – Machetă interfață înregistrare programare



2.4. Proiectarea sistemului de codificare a datelor

Aplicația folosește un sistem de codificare simplu bazat pe coduri unice pentru cheile primare. Le putem recunoaște după formularea cod + denumirea tabelei în baza de date aferentă programului nostru. Acest sistem de codificare al cheilor primare ajută în primul rând la creșterea eficienței aplicației informaticе, prin eficientizarea interogărilor, cât și oferă bazei de date o structură mult mai rezistentă.

Pentru evidența utilizatorilor am conceput un sistem de codificare simplist bazat pe asocierea dintre o valoare numerică și o funcție, după cum urmează: Pacient – Valoarea 1, Recepționist – Valoarea 2, Medic – Valoarea 3, Personal conducere – Valoarea 4.

Un alt sistem de codificare al datelor se regăsește în cadrul tabelei dedicată orelor de muncă. Acest sistem de codificare are ca rol reglementarea zilei din săptămână la nivelul bazei de date pentru a ușura etapa de programare. Astfel, zilele săptămânii primesc un cod de la 1 la 7, unde 1 reprezintă ziua de luni, iar 7 ziua de duminică.

2.5. Proiectarea logica si fizica a bazei de date

Aplicația are în spate o bază de date destul de generoasă din punct de vedere al dimensiunilor, deoarece când discutăm de activitatea medicală nu putem să dăm deoparte activitățile economice, astfel încât prin combinarea celor două domenii de activitate am obținut un conglomerat de tabele într-o bază de date.

Aceste tabele nu au nici o semnificație dacă nu sunt fundamentate corect, astfel ne propunem să analizăm structura bazei de date pe baza structurilor tabelare fundamentate în cadrul acesteia.

Figura 28 Structura tabelară aferentă tăblei Utilizatori (Users)

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
<i>username</i>	Şir de caractere(191)
<i>nume</i>	Şir de caractere(191)
<i>prenume</i>	Şir de caractere(191)
<i>CNP</i>	Şir de caractere(191)
<i>serieCI</i>	Şir de caractere(191)
<i>nrCI</i>	Şir de caractere(191)
<i>telefon</i>	Şir de caractere(191)
<i>DOB</i>	date
<i>Judet</i>	Şir de caractere(191)
<i>localitate</i>	Şir de caractere(191)
<i>strada</i>	Şir de caractere(191)
<i>numar</i>	Şir de caractere(191)
<i>bloc</i>	Şir de caractere(191)
<i>scara</i>	Şir de caractere(191)
<i>etaj</i>	Şir de caractere(191)
<i>apartament</i>	Şir de caractere(191)
<i>role_id</i>	Număr Întreg(20)
<i>email</i>	Şir de caractere(191)
<i>password</i>	Şir de caractere(191)

Tabela utilizator este tabela cea mai utilizată a bazei de date ea fiind totodată și punctul de pornire al pacienților. Considerăm echivalarea utilizator -> pacient ca regulă globală de gestiune.

Figura 29 Structura tabelară aferentă tabelei User_roles

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
rol	Şir de caractere(191)

Această tabelă o folosim pentru a stoca rolul utilizatorului sub formă de șir de caractere la nivelul bazei de date, prevenind astfel hard codare acestora la nivelul aplicației informaticе.

Figura 30 Structura tabelară aferentă tabelei administrativ

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
codUser	Număr Întreg(20)
activ	Valoare Booleană (1)

Tabela administrativ este alocată pentru a ține evidența persoanelor cu rol administrativ la nivelul aplicației informaticе.

Figura 31 Structura tabelară aferentă tabelei medic

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
codUser	Număr Întreg(20)
numarParafa	Şir de caractere(191)
codCuloare	Şir de caractere(191)
codSpecializare	Şir de caractere(191)
activ	Valoare Booleană(1)

Tabela medic este gândită ca fiind o extensie a tabelei utilizator, permitând adăugarea informațiilor suplimentare fără a altera într-un fel entitatea de bază.

Figura 32 Structura tabelară aferentă tabelei receptionist

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
codUser	Număr Întreg(20)
activ	Valoare Booleană(1)

Similar cu tabela medic avem și tabela receptionist , în aceasta stocăm doar starea de activitate a contului, încrucită receptionistul nu are informații specifice momentan, acesta fiind și o oportunitate de actualizare în viitor.

Figura 33 Structura tabelară aferentă tabelei Persoane Juridice

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
Denumire	Șir de caractere(191)
CUI	Șir de caractere(191)
nrRegCom	Șir de caractere(191)
telefon	Șir de caractere(191)
email	Șir de caractere(191)
Judet	Șir de caractere(191)
localitate	Șir de caractere(191)
strada	Șir de caractere(191)
numar	Șir de caractere(191)
bloc	Șir de caractere(191)
scara	Șir de caractere(191)
etaj	Șir de caractere(191)
apartament	Șir de caractere(191)
contBancar	Șir de caractere(191)
bancaEmitenta	Șir de caractere(191)

O altă entitate a bazei noastre de date este reprezentată de Persoana Juridică, aceasta fiind nucleul activităților economice. Ca și regulă de gestiune la nivelul aplicației se consideră

că primul tuplu de valori al tabeliei persoane juridice este firma curentă și anume S.C. Routine Med S.R.L.

Figura 34 Structura tabelară a tabelei specializări

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
specializare	Şir de caractere(191)

Tabela specializări are rolul de a reține în ea valorile specializărilor fixe și pentru a evita scrierea acestora în moduri diferite de-a lungul dezvoltării aplicație.

Figura 35 Structura tabelară aferentă tablei Cabinet

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
codSpecializare	Număr Întreg(20)
denumireCabinet	Şir de caractere(191)

Entitatea Cabinet este una foarte importantă în structura aplicației noastre deoarece ne permite o mai bună organizare a activităților medicale. Folosindu-ne de tabela specializări menționată creăm o legătură referențiară de tipul "1 la M" între cele 2 tabele menționate anterior.

Figura 36 Structura tabelară aferentă tablei Working Hours (Ore de muncă)

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
codMedic	Număr Întreg(20)
codCabinet	Număr Întreg(20)
codZi	Număr (6)
startConsultatii	Dată de tip oră
stopConsultatii	Dată de tip oră

Această tabelă reprezintă un punct de legătură între cabinete și medici, realizând astfel programul de lucru al acestora. Folosim un sistem de codificare bazat pe numere crescătoare ce a fost prezentat în cadrul capitolului 2.4. *Proiectarea sistemului de codificare a datelor*.

Câmpurile startConsultații și stopConsultații reprezintă capetele intervalor de timp în care un medic își va desfășura activitatea la nivelul cabinetului în care a fost repartizat.

Figura 37 Structura tabelară aferentă tabelei status programări

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
status	Şir de caractere(191)

Tabela status programări, nu are nici un sens dacă este luată singură, dar aceasta face parte dintr-un set de 3 entități care să ajută să reglementăm activitatea de evidență a programărilor.

Figura 38 Structura tabelară aferentă tabelei tip_programare

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
scopProgramare	Şir de caractere(191)
durata	Număr Intreg (11)

Similar cu tabela anterioară, dar are un atribut mai deosebit și anume "*durata*", exprimată în minute, care reprezintă perioada de timp predefinită pentru un anumit tip de programare. La nivelul produsului software myRoutineMed s-au definit 2 tupluri de date pentru a putea lucra și anume "*Consultul medical de specialitate*" care are o durată de 30 de minute și "*Eliberare adeverință medicală*" ce are o durată de 10 minute.

Figura 39 Structura tabelară aferentă tabelei Programări

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
idPacient	Număr Întreg(20)
idMedic	Număr Întreg(20)
idCabinet	Număr Întreg(20)
DataProgramarii	Dată
OraProgramarii	Dată de tip oră
tipProgramare	Număr Întreg (1)

statusProgramare	Şir de caractere(191)
codDocumentFinanciar	Şir de caractere(191)

Entitatea Programării entitatea principală pe care se bazează realizarea aplicației propuse, deoarece ea se ocupă cu stocarea și gestionarea sistemului de programări conceput în etapa curentă de proiectare. Această entitate are legături cu entitățile cheie ale proiectului pe care îl realizăm, realizând astfel un cumul de relații de tipul "M la M", cu tabelele "Utilizatori", "Medic", "Cabinete", "Tip Programare" și "statusProgramare".

Figura 40 Structura tabelară aferentă tabelei Scrisoare Medicală

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
idProgramare	Număr Întreg(20)
Motivele_Prezentarii	text
Diagnosticul	text
Anamneza	text
Factori_Risc_Anamneza	text
ExamenClinic	text
ExameneLab_ValNormale	text
ExameneLab_ValPatologice	text
Examene_EKG	text
Examene_ECO	text
Examene_RX	text
Examene_AlteExamene	text
Tratament_Efectuat	text
onorariuMedic	Număr Întreg(11)

Scrisoarea medicală ca și entitatea a bazei de date, ne ajută să stocăm informațiile reieșite în urma unui consult medical, ea fiind formată din atribute reglementate la nivel național prin diverse acte normative din domeniul medical, o excepție de la regulă fiind onorariul medicului, acest câmp reprezentând o punte de legătură între activitatea medicală și cea economică.

Figura 41 Structura tabelară aferentă tabelei Adeverință Medicală

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
idProgramare	Număr Întreg(20)
Ocupatia	Şir de caractere(191)
LocDeMunca	Şir de caractere(191)
SuferindDe	Şir de caractere(191)
Recomandare	Şir de caractere(191)
Destinatie	Şir de caractere(191)
onorariuMedic	Număr Întreg (11)

Adeverința medicală este un document care se realizează pentru a servii drept dovadă referitor la diverse afecțiuni medicale atât în instituțiile publice cât și în cele private. Din punct de vedere structural, adeverința medicală se asemănă foarte mult cu scrisoarea medicală deoarece aceleași elemente specifice și anume, cheie străină, reprezentată de atributul ”*idProgramare*”, elementele reglementate la nivel de normativ medical și onorariul medicului.

Figura 42 Structura tabelară aferentă tabelei Facturi Fiscale

Coloană	Tip de date
<i>numarFactura</i>	Număr Întreg(20)
codClient	int(11)
codResponabilIntocmire	int(11)
dataFacturii	Date
dataScadenta	date

Entitatea factură fiscală este cea mai importantă în relațiile cu persoanele juridice, valoarea fiscală a documentului tipizat în sistemele economice fiind foarte mare. În structura tabelară avem doar elementele de identificare ale documentului finanțier și anume, referințele către client și persoana care a întocmit documentul finanțier, dar și data emiterii și scadenței documentului.

Figura 43 Structura tabelară aferentă tabelei Facturi Detaliat

Coloană	Tip de date
<i>id</i>	Număr Întreg(20)
numarFactura	Număr Întreg (11)
DenumireArticol	Şir de caractere(191)
UnitateMasura	Şir de caractere(191)
Cantitate	Număr Întreg (11)
PretUnitarFaraTVA	Şir de caractere(191)
cotaTVA	Număr Întreg (11)

Tabela Facturi Detaliat, vine în completarea tabelei Facturi Fiscale, aceasta având o referință prin cheia străină reprezentată de numărul de factură. Din punct de vedere structural atributele acestei tabele reprezintă elementele centrale ale facturii și anume Denumirea Produsului, unitatea de măsură, cantitatea, prețul fără tva unitar și cota aferentă taxei pe valoarea adăugată (TVA).1

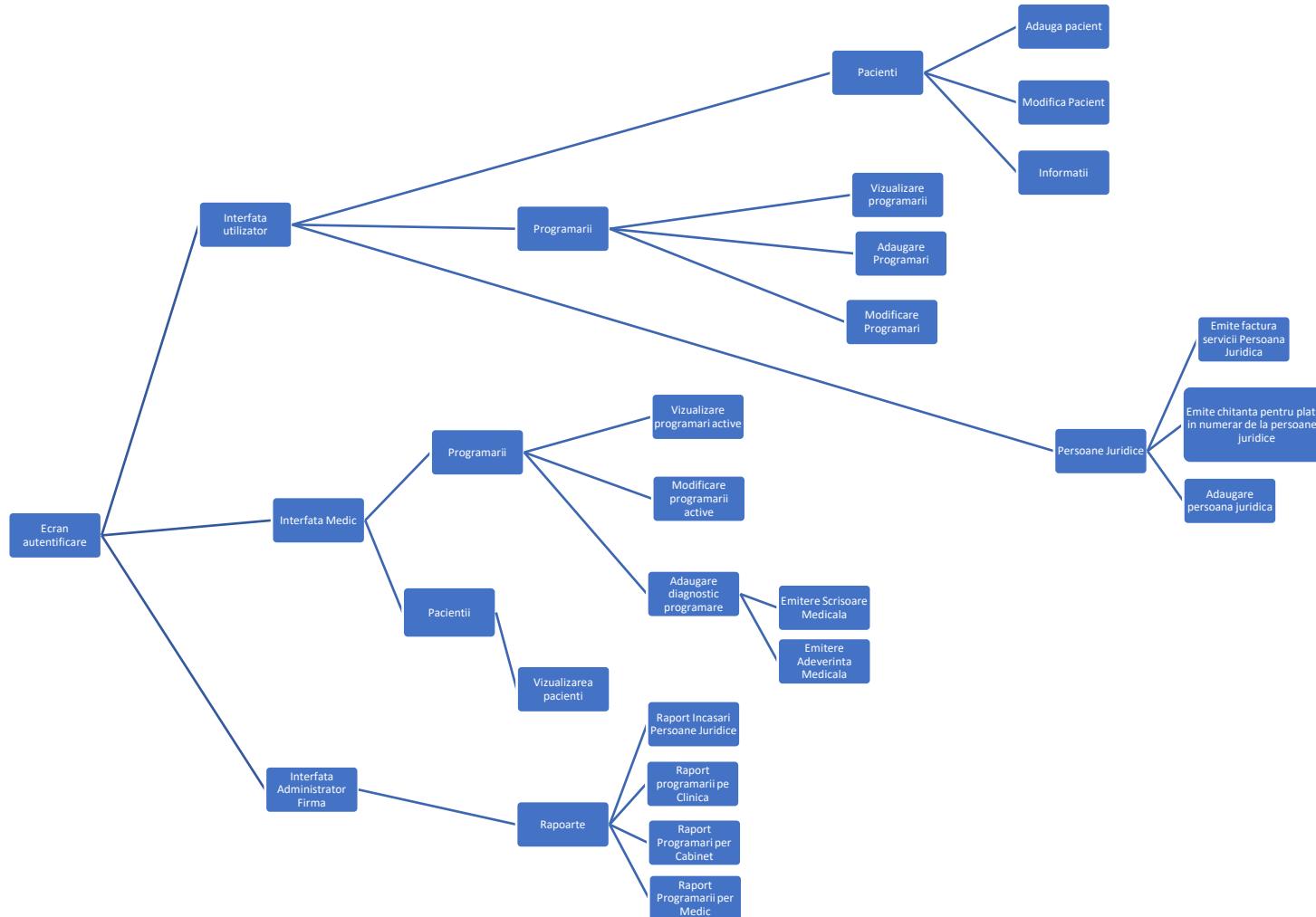
Figura 44 Structura tabelară aferentă tabelei Chitanțe

Coloană	Tip de date
<i>numarChitante</i>	Număr Întreg(20)
dataChitanta	date
numarFactura	Număr Întreg (11)
suma	Număr Întreg (8,2)
sumaInLitere	Şir de caractere(191)

Ultima tabelă din structura bazei de date a aplicației software propuse, myRoutineMed este ceea în care stocăm informațiile referitoare la plățile efectuate în Numerar.

Figura 45 – Schema de sistem a aplicației

2.6. Schema de sistem a aplicației



2.7. Proiectarea interfeței aplicației

Pentru a putea proiecta interfața aplicației ne vom folosi de un şablon gratuit pus la dispoziție pe platforma GitHub. Acesta se află sub o licență de tip MIT și poate fi folosit în diferite proiecte în mod gratuit. Această alegere ne permite să ne concentram mai mult pe funcționalitățile aplicației permitând o dezvoltare mult mai eficientă a acesteia.

Design-ul este inspirat de panourile de administrare ale soluțiilor software de tip ERP și are un meniu lateral în care sunt trecute legăturile către diferite module ale aplicației dezvoltate conform figurii numărul 46 și vine și cu un buton pentru a schimba schema de culoare în funcție de preferințele utilizatorului final.

Figura 46 – Meniul lateral pentru un utilizator cu funcția de medic



2.7.1 Interfața Personal Conducere

Deoarece folosim un sistem de tip (RBAC) "Role Based Access Control", pentru fiecare utilizator avem o interfață diferită, orientată spre nevoile acestora. În Figura 47 avem interfață pentru personalul de conducere, Figura 48, reprezintă interfață pentru medic, Figura 49 interfață pentru recepționist, iar Figura 50 reprezintă interfață pentru pacienți.

Pentru a configura aplicația ca la carte trebuie să ne conectăm din contul ce are rol de personal conducere și să schimbăm rolurile diversilor utilizatori pentru a pute avea acces la rolurile anterior menționate. Acest lucru este posibil accesând sub meniul „Listează Utilizatori” din fila Utilizatori. Odată deschisă interfață ne vom întâlnii cu fereastra din figura 51. Unde avem selectate 3 căsuțe. Prima din ele ne va oferi informațiile utilizatorului conform figurii 52, a doua ne va duce la interfață de actualizare a utilizatorului conform figurii 53 iar a 3-a ne va genera raportul privind protecția datelor cu caracter personal care se poate găsi la Anexa 6.2.1.

Dacă dorim să adăugăm un pacient nou, dar fără să îi mai facem noi cont, putem face acest lucru prin opțiunea adăugă utilizator, din meniul utilizator. Aceasta va deschide o fereastră similară cu cea din figura 53 și ne va genera un document cu datele noului cont care se poate vizualiza la Anexa 6.2.2..

Deoarece am terminat cu modulul de persoane fizice este timpul să trecem la modulul de persoane juridice. Acesta are 3 opțiuni și anume: Listare, Adăugare și Documente Fiscale. Primele 2 sunt prezentate în Figurile 53 și 54, iar cea de a treia este un raport ce va ajuta în momentul în care o să avem și activitate la nivelul firmei.

Sectiunea cabinete este crucială pentru activitatea noastră, aceasta având 3 sub meniuri esențiale și anume: "Adaugă Cabinet", "Lista Medici", "Listează Cabinete". În figura 55 vom adăuga cabinetele „Oftalmo 1”, „Oftalmo 2,” și ”Fizioterapie 1,,,” în figurile 56 și 57 vom aloca unui medic un program de lucru într-un cabinet, iar în figurile 58 și 59 o să vedem exact repartitia medicilor pe cabinete.

Figura 47– Interfața personal conducere

Acasa

Programari

Solicita o Programare

Istoric Programari

Administrativ

- Cabinete
 - Adauga Cabinet
 - Lista Medici
 - Listeaza Cabinete
- Rapoarte
 - Raport Programari Clinica
 - Cabinet
- Financiar
 - Lista Persoane Juridice
 - Adauga Persoane Juridice
 - Documente fiscale
- Utilizatori
 - Adauga utilizator
 - Listeaza utilizatori

Deconecteaza-te

Informatiile Manager-ului

Programari Totale 0

Programari in curs 0

Programari Finalizate 0

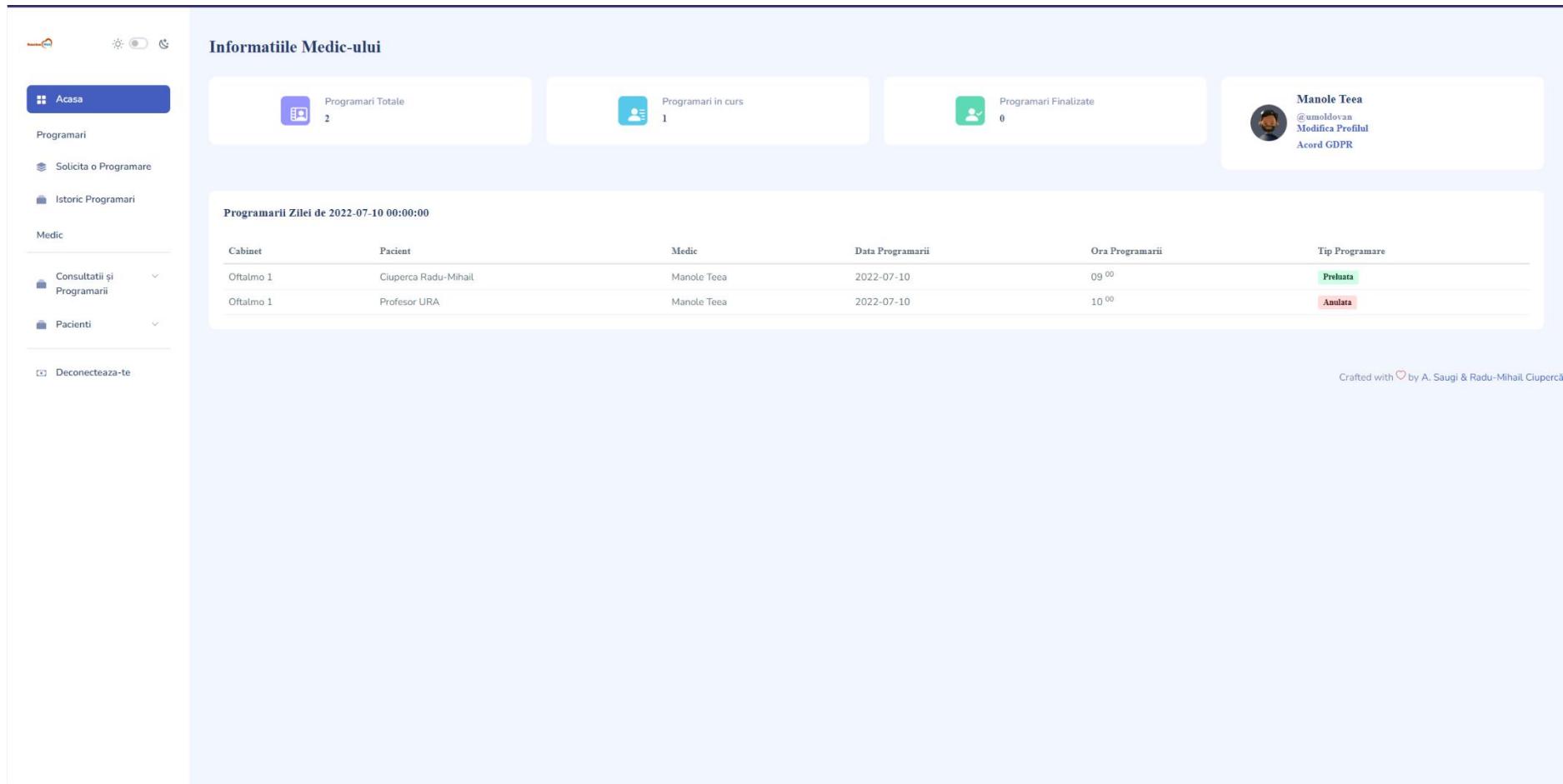
Profesor URA
 @prof.rau
[Modifica Profilul](#)
[Acord GDPR](#)

Incasari Curente

Cabinet	Data Programarii	Ora Programarii	Medic	Status
Oftalmologie	23-05-2022	09:30	Nasiri Alah	Acceptata
Fizioterapie	21-05-2022	11:30	Iordanescu Pavel	Finalizata
Fizioterapie	15-05-2022	18:30	Stanescu Stan	Anulat
Oftalmologie	01-05-2022	09:30	Nasiri Alah	Neprezentat
Oftalmologie	23-01-2022	09:30	Nasiri Alah	Acceptata

Crafted with  by A. Saugi & Radu-Mihail Ciupercă

Figura 48– Interfața personal medic



Programari
Solicita o Programare
Istoric Programari
Medic
Consultatii și Programarii
Pacienti
Deconecteaza-te

Informatiile Medic-ului

Programari Totale
2
Programari in curs
1
Programari Finalizate
0

Programarile Zilei de 2022-07-10 00:00:00

Cabinet	Pacient	Medic	Data Programarii	Ora Programarii	Tip Programare
Oftalmo 1	Ciuperca Radu-Mihail	Manole Teea	2022-07-10	09:00	Preluita
Oftalmo 1	Profesor URA	Manole Teea	2022-07-10	10:00	Anulata



Manole Teea
@umoldovan
[Modifica Profilul](#)
[Acord GDPR](#)

Crafted with ❤ by A. Saugi & Radu-Mihail Ciupercă

Figura 49– Interfața personal conducere

Acasa

Programari

Solicita o Programare

Istoric Programari

Administrativ

- Cabinete
 - Adauga Cabinet
 - Lista Medici
 - Listeaza Cabinete
- Rapoarte
 - Raport Programari Clinica
 - Cabinet
- Financiar
 - Lista Persoane Juridice
 - Adauga Persoane Juridice
 - Documente fiscale
- Utilizatori
 - Adauga utilizator
 - Listeaza utilizatori

Deconecteaza-te

Informatiile Manager-ului

Programari Totale 0

Programari in curs 0

Programari Finalizate 0

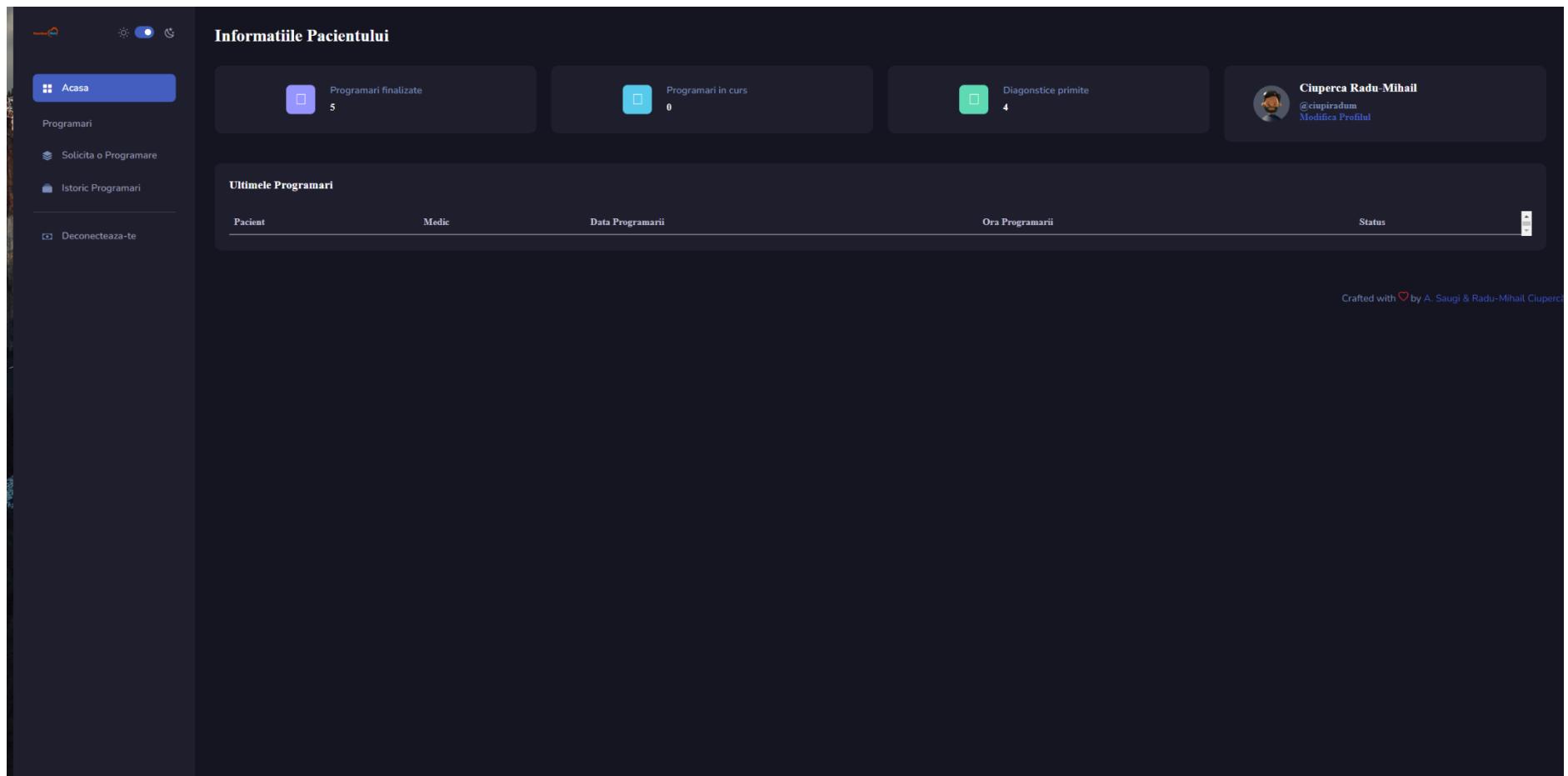
Profesor URA
 @prof.rau
[Modifica Profilul](#)
[Acord GDPR](#)

Incasari Curente

Cabinet	Data Programarii	Ora Programarii	Medic	Status
Oftalmologie	23-05-2022	09:30	Nasiri Alah	Acceptata
Fizioterapie	21-05-2022	11:30	Iordanescu Pavel	Finalizata
Fizioterapie	15-05-2022	18:30	Stanescu Stan	Anulat
Oftalmologie	01-05-2022	09:30	Nasiri Alah	Neprezentat
Oftalmologie	23-01-2022	09:30	Nasiri Alah	Acceptata

Crafted with  by A. Saugi & Radu-Mihail Ciupercă

Figura 50– Interfața pacient



Informatiile Pacientului

Programari finalizate 5

Programari in curs 0

Diagonstice primite 4

Ciuperca Radu-Mihail
@cinspiradum
Modifica Profilul

Ultimele Programari

Pacient	Medic	Data Programarii	Ora Programarii	Status

Crafted with ❤ by A. Saugi & Radu-Mihail Ciuperca

Figura 51– Interfața de listare a pacienților

Utilizatori ai platformei

Nume	Prenume	Email	Functie	Status	Operatii User	INFORMAȚII CONT	MODIFICARE ACORD GDPR
Profesor	URA	radu-mihail@serviceimo.ro	Administrativ	Activ			
Manole	Teea	ncazacu@example.net	Medic	Activ			
Niculae	Alex	mirona61@example.org	Receptionist	Activ			
Badea	Nicoleta	lazar.rus@example.org	Pacient	Activ			
Nemes	Pamela	lica.mihalcea@example.net	Pacient	Activ			
Roman	Nechifor	edgar18@example.org	Pacient	Activ			
Lazar	Amedeu	ianis.mirea@example.org	Pacient	Activ			
Mihai	Cristea	ngiurgiu@example.net	Pacient	Activ			

Figura 51– Vizualizare date pacient

Utilizatori ai platformei

Nume	Prenume	Email	Functie	Status	Operatii User
Profesor	URA	radu-mihail@serviceimo.ro	Administrativ	Activ	
Manole	Teea	ncazacu@example.net	Medic	Activ	
Niculae	Alex	mirona61@example.org	Receptionist	Activ	
Badea	Nicoleta	lazar.rus@example.org	Pacient	Activ	
Nemes	Pamela	lica.mihalcea@example.net	Pacient	Activ	
Roman	Nechifor	edgar18@example.org	Pacient	Activ	
Lazar	Amedeu	ianis.mirea@example.org	Pacient	Activ	
Mihai	Cristea	ngiurgiu@example.net	Pacient	Activ	
Miu	Ozana	Datele Utilizatorului prof.rau			
Biro	Olimpia	Nume: Profesor Prenume: URA Data Nasterii: 1949-09-27 CNP: 2880204294959 Carte Identitate: RM 827621 Telefon: 0313416648 Email: radu-mihail@serviceimo.ro Judet: Bucuresti Localitate: Bumbești-Jiu Adresa: Splaiul Zidarilor Nr. 69 BL. Sc. Et. Ap. Functie: Administrativ			
Draghici	Emil				
Stanescu	Luana				
Dinca	Andrei				
Alexe	Voicu				
Ilieșcu	Ianis				
Martin	Niculina				
Sarbu	Oliviu				
Suciul	Oana				
Ciobanu	Georgel				
Chiriac	Dumitra				
Nechita	Cristofor				

Figura 52– Editare informații cont

Actualizarea Utilizator Manole Teea

Nume	Prenume
Manole	Teea
CNP	Serie Act Identitate
3320219253472	RM
Numar CI	Email
962436	ncazacu@example.net
Telefon	Data Nasterii
0271711317	09/01/1923
Judet	Localitate
Bucuresti	Cluj-Napoca
Strada	Numar
B-dul.	73
Bloc	Scara
Bloc	Scara
Etaj	Apartament
Etaj	Apartament
Rol	
Medic	
Actualizeaza	

2021 © Mazer X RoutineMed Crafted with ❤ by A. Saugi & Radu-Mihail Ciupercă

Figura 53– Listare persoane juridice

Persoane Juridice

Cauta o firma dupa Denumire/ CUI / Nr. Reg Com							
Denumire	CUI	NrRegCom	Telefon	Email	Judet	Localitate	Operatii
S.C. ROUTINE MED S.R.L.	27028852	J36/202/2010	0240530751	contact@routinemed.ro	Tulcea	Sulina	
S.C. SERVICEIMO ELECTRONIC S.R.L.	39309134	J36/239/2018	0733956628	office@serviceimo.ro	Tulcea	Tulcea	

Figura 54– Adăugare persoane juridice

Adaugare Persoana Juridica

Denumire	CUI
S.C. SERVICEIMO ELECTRONIC S.R.L.	39309134
Nr. Reg.Com	Email
J36/239/2018	office@serviceimo.ro
Telefon	Judet
0733956628	Tulcea
Localitate	Strada
Tulcea	Dobroegeanu Gherea
Numar	Bloc
1	E2
Scara	Etaj
A	P
Apartament	Cont Bancar
2	RO73RZBR0000060020185555
Banca Emitenta	
Raiffaisen Bank Sucursala Tulcea	

Adauga Persoana Juridica

Figura 55– Adăugare cabinet Oftalmo 1

Adauga Cabinet

Denumire Cabinet	Oftalmo 1
Specializarea Cabinetului	Oftalmologie

Adauga Cabinet

Figura 56– Listare Medici

Medici inscrisi in platforma

Dashboard / DataTable

Simple Datatable

Nume	Email	Specializare	Cod Parafa	Operatii Medic
Manole Teea	ncazacu@example.net	Oftalmologie	85864812	Actualizeaza Program
Badea Nicoleta	lazar.rus@example.org	Oftalmologie	481521	Actualizeaza Program
Roman Nechifor	edgar18@example.org	Fizioterapie	9875614	Actualizeaza Program

Figura 57–Actualizare program medic

Programul de lucru al dr. Manole Teea

Alege-ți cabinetul pentru a putea edita programul

Oftalmo 1

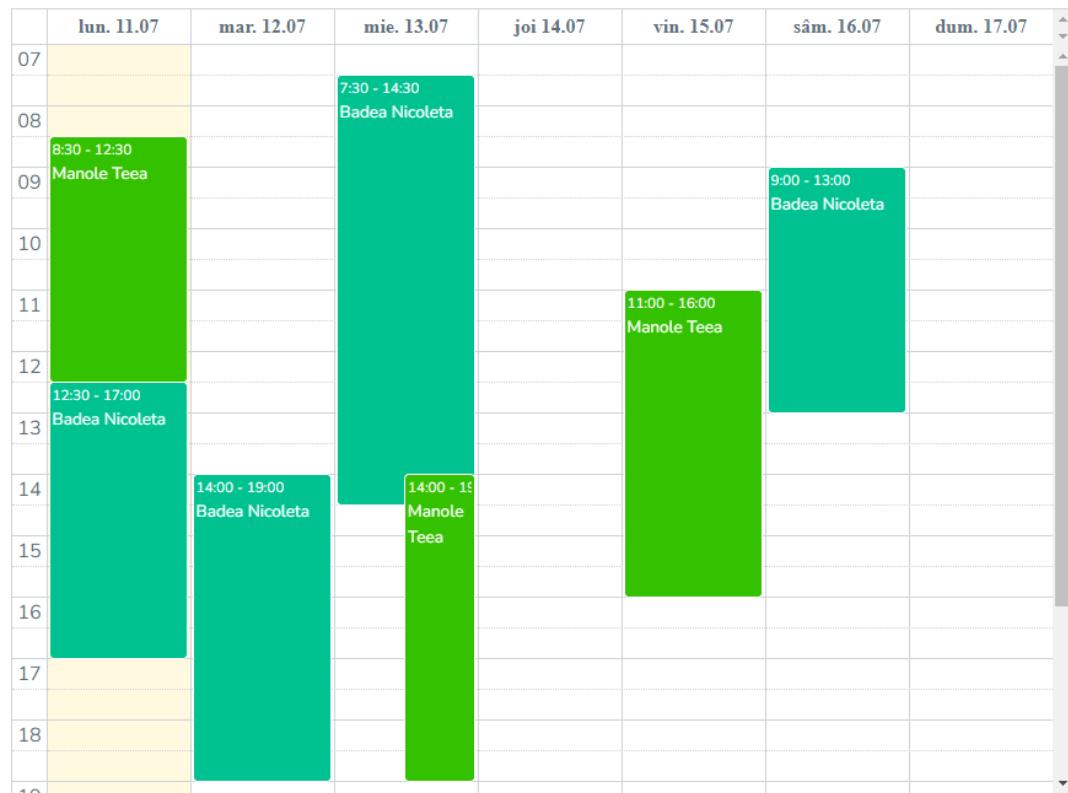
Introduceti programul pentru fiecare zi

Luni Start	08:30 AM	⌚	Luni Stop	12:30 PM	⌚
Marti Start	--:-- --	⌚	Marti Stop	--:-- --	⌚
Miercuri Start	02:00 PM	⌚	Miercuri Stop	07:00 PM	⌚
Joi Start	--:-- --	⌚	Joi Stop	--:-- --	⌚
Vineri Start	11:00 AM	⌚	Vineri Stop	04:00 PM	⌚
Sambata Start	--:-- --	⌚	Sambata Stop	--:-- --	⌚
Duminica Start	--:-- --	⌚	Duminica Stop	--:-- --	⌚

Actualizeaza Programul Medicului

Figura 58– Raport program pacient

Programul Cabinetului Oftalmo 1



2021 © Mazer X RoutineMed

Crafted with ❤ by A. Saugi & Radu-Mihail Ciuperca

Figura 59– Listare Cabinet

Cabinete Disponibile

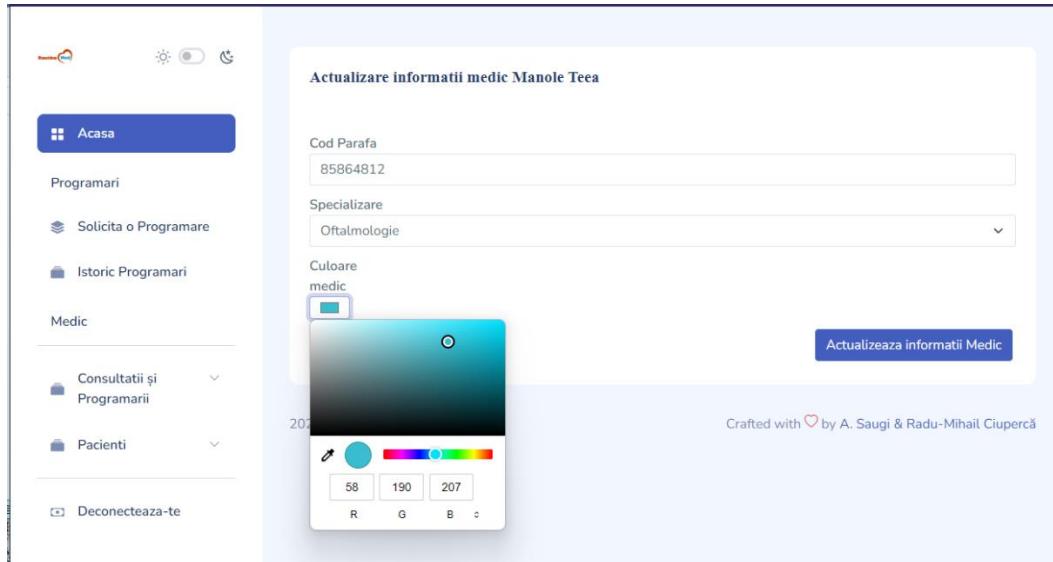
Simple Datatable

Denumire Cabinet	Specializare	Medici	Actiuni
Oftalmo 1	Oftalmologie	👤 Manole Teea 👤 Badea Nicoleta	Program Cabinet
Oftalmo 2	Oftalmologie		Program Cabinet
Fizioterapie 1	Fizioterapie	👤 Roman Nechifor	Program Cabinet

2.7.2 Interfața Medic

Odată ce ne conectăm pentru prima dată în contul de medic v-am fi obligați să ne introducem secția pe care suntem specializați, codul de parafă și o culoare pentru a putea să ne distingem mai ușor pe viitoarele grafice. În figura 60 putem observa acest lucru.

Figura 60 – Interfața conectare inițială pentru configurarea medicului

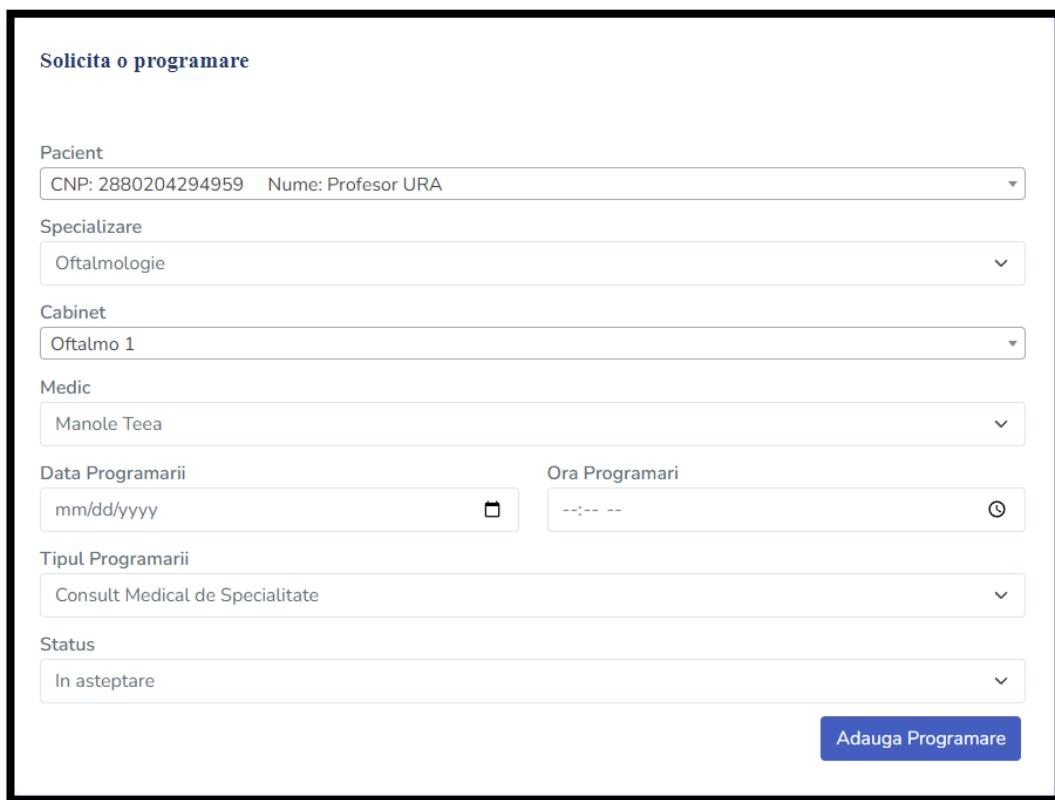


Putem observa câteva similarități ale interfeței de medic cu cea dedicată personalului de conducere.

În meniul lateral observăm 2 categorii noi, ”*Consultații și Programări*” ce conțin legăturile către raportul de consultații și adăugarea unei noi consultații și ”*Pacienți*”, această categorie conținând meniuri întâlnite anterior de adăugare și listare a pacienților.

În figura 61, putem observa interfața de adăugare a unei consultații, pre completată cu datele medicului curent. La pacient s-a implementat selector cu funcții de căutare ce este prezentat în figura 62.

Figura 61 – Interfața de adăugare a unei programări



Solicita o programare

Pacient
CNP: 2880204294959 Nume: Profesor URA

Specializare
Oftalmologie

Cabinet
Oftalmo 1

Medic
Manole Teea

Data Programarii
mm/dd/yyyy

Ora Programarii
--:-- --

Tipul Programarii
Consult Medical de Specialitate

Status
In asteptare

Adauga Programare

Figura 62 – Selectorul cu sistem de căutare



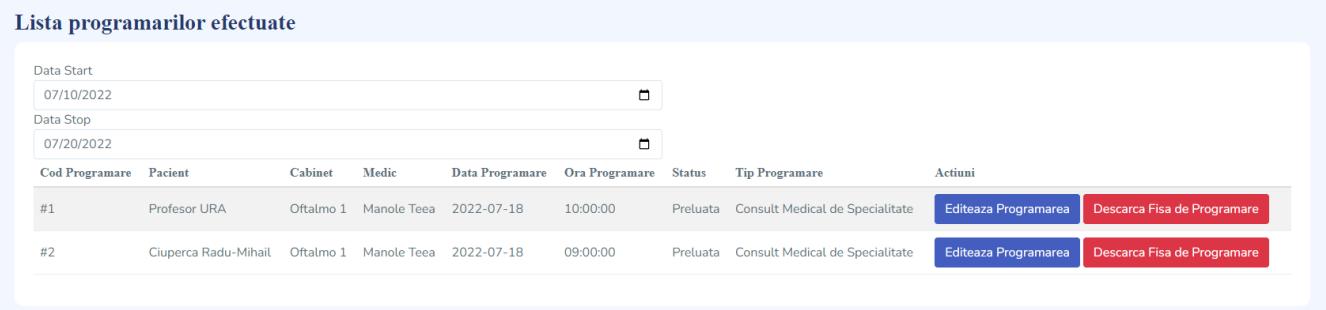
Pacient
CNP: 2880204294959 Nume: Profesor URA

Ciup

CNP: 5001111469521 Nume: Ciuperca Radu-Mihail

În figura 63 putem vedea interfața de listare a programărilor de tip raport ce ne permite să edităm programarea sau să descăr căm fișa de programare. Figura 64 ne va prezenta interfața pentru actualizarea programării iar în anexa 6.2.3. este atașată un model de fișă de programare.

Figura 63 – Interfața de listare a programărilor



Lista programarilor efectuate

Data Start	07/10/2022	Data Stop	07/20/2022					
Cod Programare	Pacient	Cabinet	Medic	Data Programare	Ora Programare	Status	Tip Programare	Acțiuni
#1	Profesor URA	Oftalmo 1	Manole Teea	2022-07-18	10:00:00	Preluata	Consult Medical de Specialitate	Editeaza Programarea Descarc Fisa de Programare
#2	Ciuperca Radu-Mihail	Oftalmo 1	Manole Teea	2022-07-18	09:00:00	Preluata	Consult Medical de Specialitate	Editeaza Programarea Descarc Fisa de Programare

Figura 64 – Interfața de actualizare a programări

PROGRAMARE

Pacient
CNP: 2880204294959 Nume: Profesor URA

Specializare
Oftalmologie

Cabinet
Oftalmo 1

Medic
Manole Teea

Data Programarii
07/18/2022

Ora Programarii
10:00 AM

Tipul Programării
Consult Medical de Specialitate

Status
Preluata

Documente Emise

Emite Adeverinta Medicala **Emite Scrisoare Medicala** **Actualizeaza**

În interfața din figura 64 putem observa 2 butoane, și anume „*Emite Adeverință Medicală*” și „*Emite Scrisoare Medicală*”, care vor deschide două noi interfețe de adăugare a documentelor cu caracter medical care sunt prezentate în Figurile 65 și 66

Figura 65 – Interfața de adăugarea a unei scrisori medicale

Scrisoare Medicală

Stimate(s) coleg(s), vă informăm că Profesor URA, născut la data de 1949-09-27 având CNP/Cod Unic de Asigurare 2880204294959, a fost consultat în serviciul nostru la data de 2022-07-18.

Motivele Prezentarii

Diagnosticul

Anamneza

Factori Risc Anamneza

Examen Clinic

Examene de Laborator cu Valori Normale

Examene de Laborator cu Valori patologice

Examinare EKG

Examinare Ecograf

Examinare Radiografie

Alte examinarii relevante

Tratament Efectuat

Onorariu Medic

Emite Scrisoare Medicala

Figura 66 – Interfața de adăugarea a unei adeverințe medicale

Adeverință Medicala

Se adeverește că **Profesor URA**, născut la data de 1949-09-27 cu domiciliul în județul **Bucuresti**, localitatea **Bumbești-Jiu**, adresa **Splaiul Zidarilor nr. 69** având ocupația de **Profesor** în cadrul **Universității Romano-Amc**.

Aceasta este suferind de:

Cornee Inflamata

Se Recomanda:

Mutarea acestuia într-un departament cu mai puțin praf

Prezenta adeverita a fost eliberata pentru a-i putea fi de folos la

Locul de munca

Onorariu Medic

10

Emite Adeverinta Medicala

2021 © Mazer X RoutineMed

Crafted with ❤ by A. Saugi & Radu-Mihail Ciuperca

Odată emise documentele medicale ele vor apărea pentru a fi accesate conform figurii 67. În cadrul anexelor 6.2.4, respectiv 6.2.5 vor fi reprezentate prezentele documente.

Figura 67 - Interfața de actualizare a programării cu documente generate în ea

PROGRAMARE

Pacient
CNP: 2880204294959 Nume: Profesor URA

Specializare
Oftalmologie

Cabinet
Oftalmo 1

Medic
Manole Teea

Data Programarii
07/18/2022

Ora Programarii
10:00 AM

Tipul Programarii
Consult Medical de Specialitate

Status
Preluata

Documente Emise Adeverinta Medicala #1 Scrisoare Medicala #1

Emite Adeverinta Medicala **Emite Scrisoare Medicala** **Actualizeaza**

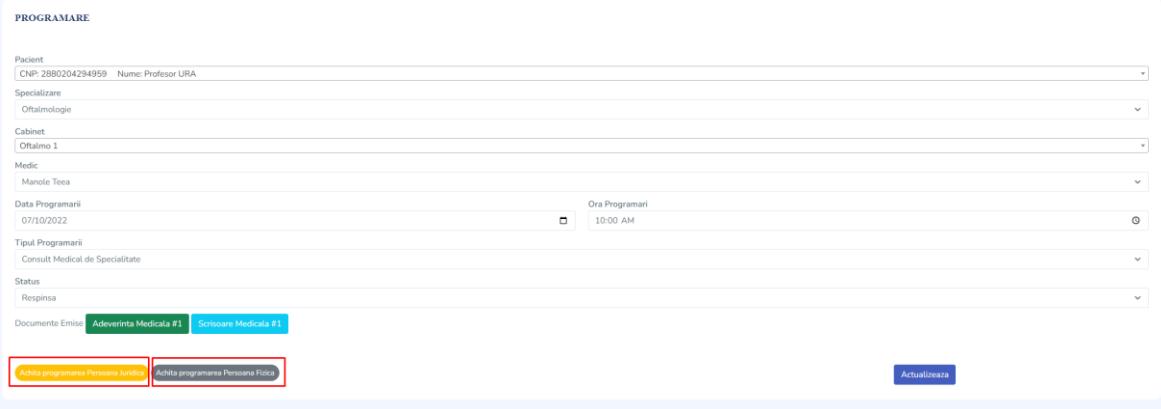
2.7.3 Interfață recepționist

În cadrul aplicației informative myRoutineMed, funcția de recepționist are aceleași privilegii cu cea dedicată personalului de conducere, diferențele apărând la zona de management al utilizatorilor și al cabinetelor, astfel , putem concluziona că mare parte din facilitățile de la poziția de administrare se regăsesc și aici.

Din punct de vedere al meniurilor avem de a face cu cele 3 mari categorii și anume „Programări”, „Pacienți”, „Angajați”. În cadrul acestora am mai intrat dar dorim să evidențiem finalitatea procesului de programare, astfel ne propunem să emitem o factură către unul din partenerii B2B în scop demonstrativ pentru a finaliza prezentarea interfeței.

În figura 68 putem observa că cele 2 butoane pentru emiterea de scrisoare medicală și adeverință medicală, au fost înlocuite cu „Achită programarea Persoană Juridică”, „Achită Programarea Persoană Fizică”. Acționarea butonului pentru persoane fizice nu a fost încă implementată, dar cea pentru persoane juridice ne va duce în interfață de alegere a clientului și a metodei de plată, prezentă în figura 69. După aceasta se va genera factura fiscală și după caz chitanță dacă s-a optat plata cu numerar, care vor fi atașate la anexe cu numerele 2.6.6 respectiv 2.6.7. Odată achitată programarea statusul ei se va schimba în finalizată iar butoanele din figura 68 vor dispărea complet, acest lucru fiind prezentat în figura 70.

Figura 68 - Interfața de actualizare a programări cu documente generate în ea II



PROGRAMARE

Pacient
CNP: 2880204294959 Nume: Profesor URA

Specializare
Oftalmologie

Cabinet
Oftalmo 1

Medic
Manole Teja

Data Programarii
07/10/2022

Ora Programarii
10:00 AM

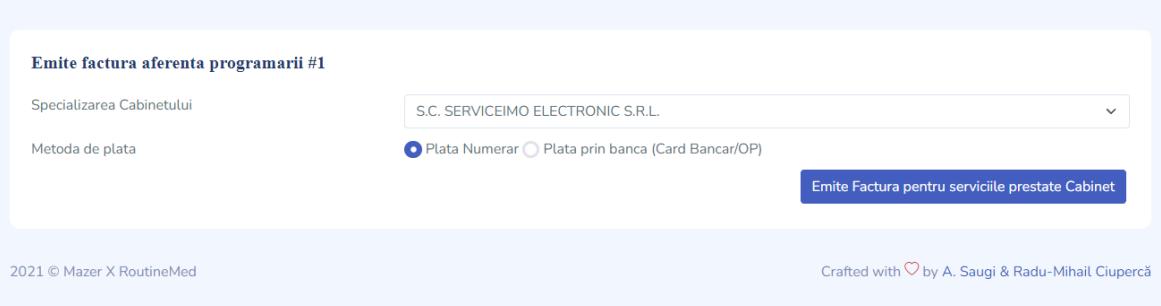
Tipul Programarii
Consult Medical de Specialitate

Status
Respsna

Documente Emise: **Adeverinta Medicala #1** **Scrisoare Medicala #1**

Actualizare Persoana Juridica **Actualizare Persoana Fizica** **Actualizare**

Figura 69 - Interfața de alegere a metodei de plată pentru persoană juridică



Emite factura aferenta programarii #1

Specializarea Cabinetului
S.C. SERVICEIMO ELECTRONIC S.R.L.

Metoda de plata
 Plata Numerar Plata prin banca (Card Bancar/OP)

Emite Factura pentru serviciile prestate Cabinet

2021 © Mazer X RoutineMed

Crafted with ❤ by A. Saugi & Radu-Mihail Ciupercă

Figura 70 - Interfața de actualizare a programări cu documente generate în ea III

PROGRAMARE

Pacient
CNP: 2880204294959 Nume: Profesor URA

Specializare
Oftalmologie

Cabinet
Oftalmo 1

Medic
Manole Teea

Data Programarii
07/10/2022

Ora Programarii
10:00 AM

Tipul Programarii
Consult Medical de Specialitate

Status
Respsnsa

Documente Emise Adeverinta Medicala #1 Scrisoare Medicala #1 Factura Fisicala #1 Chitanta #1

Actualizeaza

2.7.4. Interfață Pacient

Interfața pacientului este cea mai simplistă dintre toate rolurile, aceasta având acces doar la acordul de prelucrare a datelor cu caracter personal, istoricul programărilor și solicitarea unei noi programări.

Deoarece am prezentat atât acordul pentru prelucrarea datelor, efectuarea unei programări și actualizarea profilului vom vorbi despre istoricul personal.

Acesta poate fi accesat din orice pagină din rubrica „Istoric Programări”. În figura 71 avem această interfață.

Figura 71 - Istoric Programări

Lista programarilor efectuate de catre Ciuperca Radu-Mihail						
Cod Programare	Cabinet	Medic	Data Programare	Ora Programare	Status	Acțiuni
#2	Oftalmo 1	Manole Teea	2022-07-10	09:00:00	Preluata	Vizualizeaza Programarea Descarca Fisa de Programare
#3	Fizioterapie 1	Roman Nechifor	2022-07-13	09:00:00	In asteptare	Vizualizeaza Programarea Descarca Fisa de Programare

2.8. Alegerea tehnologiei de prelucrare și a infrastructurii necesare pentru funcționarea optimă a aplicației.

Nevoile identificate anterior, împreună cu modelul conceptual al bazei de date, nu sunt considerate o bază suficientă pentru realizarea unei aplicații informaticе, aceasta având nevoie de un limbaj de programare în care să fie realizată și de o infrastructură hardware și software pentru a putea fi executată în mod corespunzător.

Aplicația informatică este făcută în limbajul de programare PHP, versiunea 8, folosind cadrul de dezvoltare open source⁶ Laravel versiunea 9. Laravel are specific arhitectura de tip MVC⁷.

Pentru realizarea interfețelor se vor utiliza tehnologiile web HTML versiunea 5, CSS versiunea 3 și Java Script versiunea 6 împreună cu librării dedicate acestora cum ar fi Bootstrap și VueJS pentru o dezvoltare mai facilă a aplicației.

Pentru modulul de calendar am utilizat o librărie de Java Script și anume Full Calendar ce ne permite realizarea automată a unor interfețe de tip calendar pentru popularea lor cu informații din baza de date.

Din punct de vedere al bazelor de date aplicația informatică este proiectată pe un model de baze de date relaționale și va folosi SGBD⁸-ul MySQL și interfața de lucru phpMyAdmin, pentru a putea interacționa cu aceasta. Totodată vom folosi și ORM⁹-ul pus la dispoziție de cadrul de dezvoltare Laravel și anume Eloquent-ORM care ne permite definirea unor modele în cadrul bazelor de date și popularea acestora cu date de teste prin proceduri de tip seed¹⁰.

⁶ Open Source – Tipologie de proiecte (De regulă din domeniul programării) care este public și la care fiecare persoană poate aduce modificării în mod gratuit.

⁷ MVC – acronim în limba engleză (Mode-View-Controller), este un model arhitectural ce separă concepțele cheie ale unei aplicații între ele pentru a nu amesteca partea de interfață grafică cu logica de business.

⁸ SGBD - acronim (Sistem de Gestire a Bazelor de Date)

⁹ ORM – acronim în limba engleză (Object Relational Mapper), subprogram al unui cadru de dezvoltare ce facilitează interacțiunile cu bazele de date prin cod specific al limbajului prin operații denumite migrații.

¹⁰ Proceduri de tip seed – Sunt algoritmi speciali care pe baza limbajului de prelucrare RegEx pot genera prin intermediul unui Design Pattern de tip factory entități cu caracter aleator într-o bază de date pentru facilitarea etapei de dezvoltare.

2.9. Estimarea calendarului de resurse și a calendarului de realizare

Aplicația informatică este planificată pentru execuție pentru o perioadă de 6 luni începând cu data de 01 noiembrie 2021 conform calendarului de execuție, acesta a fost realizat prin programul software Microsoft Project în figura cu numărul 73 și are alocate ca și resurse următoarele elemente menționate în tabelul de mai jos.

Figura 72 - Tabel de resurse alocate

Tip resursa	
Resurse umane	1 Student cu atribuții echivalente ale unor persoane cu funcțiile: <ul style="list-style-type: none"> • Manager Proiect • Analist de sistem • Inginer Baze de date • Programator Web • Tester
Resurse de timp	6 luni
Resurse financiare	Nu s-au utilizat resurse financiare în cadrul acestui proiect
Resurse materiale	1 x Sistem de calcul cu rol de server având următoarele specificații cheie: Procesor: Intel Core i5 4460 3.2Ghz Memorie RAM: 8GB DDR3 1600Mhz Stocare: 1 X SSD Kingston 500GB SATA 1 x Sursă de alimentare neîntreruptibilă (UPS) 200VA pentru a asigura salvarea datelor în cazul întreruperilor de curent electric

Figura 73– Captura de ecran Microsoft Project – (De rectificat personalul și perioadele de timp)

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1	Normal	Project informatic myRoutineMed	422 days	Mon 03.01.22	Tue 15.08.23		
2	Normal	Sedinta de ofertare	1 day	Mon 03.01.22	Mon 03.01.22		
3	Normal	Descrierea nevoilor de informatizare	1 day	Mon 03.01.22	Mon 03.01.22		Manager Proiect[50%]
4	Normal	Descrierea sistemului software propus	1 day	Mon 03.01.22	Mon 03.01.22		Manager Proiect[50%]
5	Normal	Studiul și analiza sistemului existent	60 days	Tue 04.01.22	Mon 28.03.22	2	
6	Normal	Studiul sistemului existent	60 days	Tue 04.01.22	Mon 28.03.22		
7	Normal	Studiul statutului de infinitare și al structurii organizatorice	14 days	Tue 04.01.22	Fri 21.01.22		Analist de sistem[20%]
8	Normal	Studiul aplicațiilor informative existente în funcție în cadrul organizației și a domeniului de interes	24 days	Tue 04.01.22	Mon 28.03.22	7SS	Analist de sistem[40%]
9	Normal	Studiul de pregătire al personalului în utilizarea produselor software, etc	14 days	Tue 04.01.22	Fri 21.01.22	7SS	Analist de sistem[40%]
10	Normal	Studiul bazelor de date existente	7 days	Tue 04.01.22	Wed 12.01.22	7SS	Inginer Baze de Date
11	Normal	Studiul dotării cu tehnică de calcul existentă	7 days	Thu 13.01.22	Fri 21.01.22	10	Inginer Baze de Date
12	Normal	Analiza sistemului informational decizional	19 days	Mon 24.01.22	Thu 17.02.22	6	
13	Normal	Studiul sistemului decizional	5 days	Mon 24.01.22	Fri 28.01.22		Analist de sistem[50%]
14	Normal	Studiul sistemului de execuție	5 days	Mon 24.01.22	Fri 28.01.22	13SS	Analist de sistem[50%]
15	Normal	Studiul sistemului informational existent	14 days	Mon 31.01.22	Thu 17.02.22	14	Analist de sistem
16	Normal	Analiza critică a sistemului existent	3 days	Fri 18.02.22	Tue 22.02.22	15	Analist de sistem
17	Normal	Proiectarea de ansamblu	73 days	Wed 23.02.22	Fri 03.06.22	5	
18	Normal	Definirea obiectivelor sistemului;	5 days	Wed 23.02.22	Tue 01.03.22		Analist de sistem
19	Normal	Structurarea sistemului informatic pe subsisteme și componente	14 days	Wed 02.03.22	Mon 21.03.22	18	Analist de sistem
20	Normal	Definirea globală a ieșirilor	5 days	Tue 22.03.22	Mon 28.03.22	19	Analist de sistem
21	Normal	Definirea globală a intrarilor	5 days	Tue 29.03.22	Mon 04.04.22	20	Analist de sistem
22	Normal	Definirea colectoilor de date comune	10 days	Tue 05.04.22	Mon 18.04.22	21	Analist de sistem;Inginer Baze de Date
23	Normal	Alegerea soluțiilor tehnice	7 days	Tue 19.04.22	Wed 27.04.22	22	Analist de sistem;Inginer Baze de Date;Programator 1
24	Normal	Propunerea de dotare a organizației cu tehnica de calcul	5 days	Thu 28.04.22	Wed 04.05.22	23	Analist de sistem;Inginer Baze de Date
25	Normal	Estimarea necesarului de resurse	5 days	Thu 05.05.22	Wed 11.05.22	24	Analist de sistem;Manager Proiect
26	Normal	Estimarea eficienței economice	3 days	Thu 12.05.22	Mon 16.05.22	25	Analist de sistem;Manager Proiect
27	Normal	Planificarea realizării sistemului	7 days	Tue 17.05.22	Wed 25.05.22	26	Analist de sistem;Manager Proiect

Calendarul de realizare cuprinde cele 6 etape principale ale realizării unui produs software și anume:

1. Studiul de fezabilitate al aplicației
2. Studiul și analiza sistemului existent
3. Proiectarea de ansamblu
4. Proiectarea de detaliu
5. Programarea
6. Implementarea sistemului informatic

3. PREZENTAREA APLICATIEI SOFTWARE

3.1. Cerințele platformei hardware și software ale produsului realizat

Deoarece este o aplicație bazată pe arhitectură web și am ales realizarea acesteia cu limbaje de scripting cum ar fi HTML, CSS și Java Script dar și cu limbajul de programare PHP pentru partea de logică, instrumentat prin cadrul de dezvoltare Laravel, nu avem cerințe de sistem foarte ridicate ceea ce permit instalarea pachetelor software de tip dependință pe configurații cu capacitați mai restrânse din punct de vedere hardware.

Firma Routine Med S.R.L. a ales reutilizarea unui sistem desktop, menționat în capitolul 2.9, pentru a instala și configura aplicația.

Din punct de vedere software pentru a putea implementa aplicația vom avea nevoie de câteva pachete software ce se ocupă cu partea de administrarea a serverului. În limbaj de specialitate ele sunt cunoscute sub acronimul LAMP – “Linux-Apache-MySQL-PHP”. Alte dependințe software sunt realizate de instrumentarul composer care este un manager de pachete ce ne permite adăugarea de module extra în mod mult mai facil și server-ul de node.js, care se ocupă de gestionarea unor elemente server side din cadrul Laravel

3.2. Descrierea funcțiilor aplicației

Deoarece specificul activităților informaticice este bazat pe rezolvarea problemelor mari prin segmentarea acestora în probleme mai mici pe care le putem rezolva mult mai ușor am ales să structurăm aplicația informatică în 5 etape interconectate între ele după cum urmează:

- Implementarea programatică a sistemului de migrații
- Implementarea sub-sistemului de autentificare și securitate
- Implementarea sub-sistemului de programării
- Implementarea sub-sistemului de facturare
- Alte subsisteme ale aplicației

3.2.1. Implementarea programatică a sistemului de migrații

Cadrul de dezvoltare Laravel ne permite o abordare programatică pentru rezolvarea sarcinilor din domeniul bazelor de date prin intermediul instrumentului Eloquent-ORM care ne permite realizarea și manipularea tabelelor prin intermediul migrațiilor.

Migrațiile sunt proceduri realizate de către instrumentele de tip ORM prin care putem intervenii în structurile tabelare realizând operații mult mai ușor și mai sigur comparativ cu operarea datelor prin limbajul procedural SQL deoarece, prin clauzele specifice limbajului nu putem să securizăm conținutul interogării și un atacator poate profita de acest lucru realizând un atac de tip SQLInjection.

Fișierele de migrații sunt create automat utilizând comanda „*php artisan make:migration nume_migrație*“ sau „*php artisan make:model nume_model -m*“ unde acel “-m” reprezintă un tag pentru crearea unei migrații bazată pe model. În anexa 6.3.1 avem fișierele generate de comenziile de migrare menționate anterior.

Structura unei migrații de regulă se face în 2 etape și anume, etapa de crearea reprezentată de funcția *up()* și etapa de ștergere reprezentată de funcția *down()*. În anexa 6.3.2 avem cele funcții exemplificate pentru tabela *user_roles*.

Observăm că în funcția *up()* am creat o tabelă cu 2 atrbute și nume *id()* care este un generator automat de cheie primă oferit de Laravel, și *rol* ce este un atrbut de tip sir de caractere. Prin această cale programatică putem adăuga mult mai multe constrângeri ce se pot regăsi în documentația oficială Laravel.

Un alt lucru pe care îl observăm în cadrul funcției *up()* este referitor la inserarea unor valori în baza de date. Am ales să introducem cele 4 roluri menționate anterior la capitolul 2.4 al prezentei lucrări deoarece la trecerea de la o versiune a migrație la o altă versiune a

migrației se recomandă curățarea conținutului bazelor de date dacă s-a modificat structura curentă a tabelelor.

Funcția down() se ocupă cu ștergerea tabelelor și pregătirea pentru o nouă migrație. De regulă aici întâlnim instrucțiunea dropIfExists('numele_tabelei') care va șterge tabela și conținutul acesteia pentru a instala noua migrație.

Anexele 6.3.3, 6.3.4, 6.3.5, 6.3.6, 6.3.7, 6.3.8, 6.3.9 și 6.3.10 prezintă procesul de migrare realizat în mod programatic pentru celelalte tabele ale bazei de date.

Pentru a putea testa și simula aplicația informatică în condiții de lucru similare cu cele din viața reală ne putem folosi de facilitățile de factory ale framework-ului Laravel care ne permit să creăm modele complexe cu date simulate cu ajutorul extensie “faker-php”. În figura 6.3.11 putem observa o procedura de tipul factory pentru modelul utilizatorului care definește tipul de date și posibilitățile pe care acesta le poate lua. Pe lângă această metodă de tip factory, am adăugat prin intermediul Eloquent și o valoare inițială în baza de date pentru a ne facilita accesul pe poziția conturilor de administrare ori de câte ori refacem o migrație.

Odată realizată procedura de tip factory vom iniția acțiunea de „seeding”, aceasta fiind popularea propriu zisă a datelor în aplicația informatică. Această acțiune se realizează prin intermediul unui terminal rulând comanda prezentată în figura 6.3.12, iar rezultatele acesteia le putem vedea la nivelul interfeței “phpMyAdmin” în figura 6.3.13.

3.2.2. Implementarea sub-sistemului de autentificare și securitate

Cadrul de dezvoltare Laravel este cunoscut pentru gradul ridicat de securitate pe care îl oferă, fiind orientat către securitate din punct de vedere al programării, de aceea acesta are un modul de autentificare extrem de bine gândit ce implementează funcționalități avansate cum ar fi funcțiile de tip auth() sau token().

În cadrul prezentului proiect, am plecat de la modelul users care l-am asociat implicit cu funcția de pacient al clinici și care este prezentat din punct de vedere al structuri în anexa 6.3.3.

Modelul user se poate regăsi în anexa cu numărul 6.3.14, unde întâlnim vectorul \$fillable, Acesta ne arată valorile vectorului care pot fi populate de către utilizator prin diferite proceduri ale Eloquent ORM.

Pentru a putea realiza procedura de autentificare și înregistrare a fost nevoie să realizăm un set de rute ce se regăsesc în directorul cu același nume în fisierul web.php. Le avem prezentate și în prezenta lucrare la anexa 6.3.15.

În anexa 6.3.15 putem observa 2 tipuri de rute și anume rutele de tip GET și cele de tip POST. Rutele de tip GET se folosesc de regulă pentru a afișa o pagina, pe când ce rutele de tip POST se folosesc pentru validarea sau trimitera unor formulare și anume ruta login ne va afișa pagina de autentificare în timp ce ruta loginProcess se va ocupa de partea de validare și trimitere în aplicație. Același lucru se întâmplă și cu celelalte rute din cadrul anexei.

Mai putem observa existența unui middleware ce reprezintă un agent de validare pentru diferite funcționalități ale aplicației noastre, middleware-ul auth fiind implicit creat în cadrul Laravel și ne va permite trierea utilizatorilor autenicați de simplii vizitatori ai paginii prin redirecționarea acestora către pagina de autentificare a aplicației.

După realizarea rutelor a fost nevoie să realizam logica din spatele procesului de conectare. Aceasta s-a realizat în controller-ul UserController prin diferite funcții ce au fost apelate de către rute. Pentru realizarea rutelor de tip GET logica aplicației este foarte simplistă și este prezentată în anexele 6.3.16 – 6.3.19 unde la apelarea celor 2 funcții aferente rutelor doar oferim o legătură către aceea pagina.

Procesul de autentificare este mai complex deoarece pe deosebire avem etapa de validare care preia din pagina de login informațiile introduse de utilizator și le trece printr-un set de filtre dar se face și verificarea legată de faza a doua a procesului de înregistrare și anume prezența. Deoarece procesul de înregistrare se face în 2 etape, prima în care

utilizatorul își introduce datele esențiale pentru a crea un cont iar a doua în care introduce restul de date, este nevoie să validăm acest lucru prin constrângerea completării datelor. Pentru a face acest lucru am impus ca valoarea atributului CNP să nu fie de tip NULL pentru a putea continua.

Procesul de înregistrare este similar cu cel de autentificare cu aceleași mențiuni de mai sus referitor la validarea câmpurilor și a atributelor cheie necesare desfășurării activității.

Din punct de vedere al video format-ului utilizat din şablonul de design, singurele modificări pe care a trebuit să le aducem au fost modificarea identificatorilor de introducere a textului la nivel de limbaj HTML și adăugarea directivelor @CSRF pentru protecția la atacurile de tip Cross Site Request Forgery , acestea se pot identifica în anexa 6.3.16.

O altă specificație cu un rol decisiv în cadrul aplicației noastre informaticе este structurarea acesteia într-un sistem bazat pe roluri care permite diferențierea funcțiilor diversilor utilizatori pe baza unui rol bine stabilit. Din punct de vedere al implementării a fost adăugată coloana *role_id* la nivelul tablei *user* care poate lua valori de la 1 la 4, sistemul de codificare fiind explicitat pe larg în cadrul *Capitolului 2.4 Proiectarea sistemului de codificare al datelor*, iar pentru rolurile adiacente au fost adăugate tabele separate în care vom stoca informații referitoare la stadiul de activare al rolului ales. Gestionaarea acestor conturi se realizează din interfața de administrare accesibilă doar utilizatorilor cu rol 4 și este prezentată în cadrul capitolului 2.7 ”*Proiectarea interfeței aplicației*” iar codul sursă în anexele 6.3.20 – 6.3.22. Un alt avantaj al sistemului de acces bazat pe roluri este posibilitatea realizării interfețelor personalizate pentru fiecare rol în parte ele fiind prezentate în cadrul capitolului 2.7 ”*Proiectarea interfeței aplicației*”.

3.2.3. Implementarea sub-sistemului de programării

O altă componentă esențială a aplicației informaticice pe care o realizăm este reprezentată de sistemul de programării al pacienților, acesta fiind totodată și nucleul aplicației informaticice despre care discutăm astăzi.

Pentru a putea discuta despre sub-sistemul de programării este nevoie în primul rând să realizăm organizarea și fluxurile de date specifice acestuia, astfel ne propunem să realizăm o structură bazată pe cabinete medicale cu un program de lucru definit de către conducerea unității medicale și cu un sistem de generare automată a documentelor ce va fi un subiect de discuție pentru următorul capitol al prezentei lucrări.

În anexa 6.3.23 avem prezentate setul de rute pentru programări și cabinete care ne va ajuta să gestionăm cele 2 module. Începem cu rapoartele de vizualizare pentru cabinete și programări care sunt reprezentate vizual în capitolul 2.7, iar codul sursă în setul de anexe 6.3.24 – 6.3.29 avem porțiuni din codul sursă al paginilor care vor fi afișate în aplicație și fac parte din categoria vizualizărilor pentru cabinete, iar în setul de anexe cuprins între 6.3.30 – 6.3.43 prezintă interfețele de lucru dedicate modulului de programări.

Pentru modulul de cabinete ne folosim de un controller care se ocupă de operațiile cu baza de date și validarea formularelor. În cadrul anexei 6.3.44 avem o captură de ecran cu codul sursă aferent acestuia.

La nivelul modulului de programări controller-ul conține multe funcții de returnare a paginilor ce au un nivel de complexitate destul de redus. Totuși, cea mai interesantă funcție din punct de vedere al acțiunilor executate este funcția de validare a programărilor. Aceasta a fost scrisă într-un mod dinamic care să permită atât validările de introducere cât și pe cele de actualizare fără a mai fi nevoie de generarea unor noi paginii și funcții similare, respectând astfel principiul (DRY) „Don't repeat yourself”, referitor la scrierea codului curat ”Clean Code”. Putem vedea această funcție de validare la anexa 6.3.45. Procesul simplificat din această porțiune de cod se rezumă la validarea câmpurilor primite prin formular, verificarea existenței programărilor și a disponibilității medicului pentru programare, urmată de introducerea valorilor în baza de date pentru a putea fi puse în continuare în circuitul informatic.

3.2.4. Implementarea sub-sistemului de facturare

Pentru orice firmă sistemul de facturare este crucial pentru activitatea economică a acesteia, procesele de facturare digitală fiind o necesitatea a afacerilor din ziua de azi. Pentru a oferi o soluție software cât mai complexă ne-am propus să facem și un sistem de facturare prin care personalul clinici vor avea posibilitatea să-și genereze în mod automat documentele financiare către partenerii B2B fără a mai fi nevoie de intervenția nivelelor înalte de conducere, pentru programările standardizate, realizându-se astfel o conexiune între sub-sistemul de programări din capitolul 3.2.3 și sub-sistemul.

Din punct de vedere al tabelelor, avem de-a face cu 3 entități principale, la nivelul sistemului de gestiune al bazelor de date, și anume entitățile *"Facturi Fiscale"*, *"Facturi Detaliat"* și *"Chitanțe"*.

Procedura de emitere facturi este simplistă și este inițiată, la nivelul fiecărei programări de către persoanele ce au rolurile de *"Receptionist"* sau *"Personal Conducere"*, acestea fiind totodată și entitățile care sunt înregistrate ca și responsabili de facturare în raport cu autoritățile statului român.

Odată inițiată procedura de generare a documentelor se întreabă dacă plata serviciilor se face prin numerar, implicit dacă este nevoie de eliberarea unei chitanțe fiscale și se realizează un set de interogări la nivelul bazei de date pentru a căuta fiecare document medical emis și pentru a-l adăuga în cadrul unei facturi fiscale, care va fi generată în mod automat în format PDF (*"Portable Document Format"*) pentru a putea fi tipărită sau trimisă prin email partenerilor de afaceri.

În anexele 6.3.46 – 6.3.48 regăsim procedura de emitere a unei facturi pe baza unei programări.

3.2.5. Alte sub-sisteme ale aplicației

Odată prezentate sistemele mari ale firmei nu trebuie să neglijăm nici detaliile care fac diferență în ceea ce privește un produs software calitativ, detaliile ce cresc calitatea vieții utilizatorului software și care își propun să rezolve probleme ale utilizatorului de care acesta nici nu știa.

Printre aceste sub-sisteme avem și generatorul automat al formularului referitor la politica de confidențialitate privind stocarea și prelucrarea datelor cu caracter personal, sub-sistem al cărui rezultat poate fi văzut în cadrul anexei de rapoarte emise de către aplicația informatică.

Un alt subsistem este realizat prin implementarea unui convertor de variabile de tip numeric la variabile de tip sir de caractere preluat de pe forumul comunității "PHPRomânia", de la utilizatorul "dechim" și implementat în cadrul aplicației cu ajutorul şablonelor de tip Blade, ale cadrului de dezvoltare Laravel. Necesitatea acestui sistem a apărut odată cu problema generării de chitanțe fiscale în mod automat, deoarece acestea au secțiunea de control "suma în litere", acest sub-sistem reducând drastic timpii de lucru ai utilizatorilor și dimensiunile bazei de date.

Nu putem uita de sistemul de generare a unui document cu datele noilor utilizatori, adăugați de personalul clinicii, conceput în special pentru persoanele în etate care nu au cunoștințele necesare să opereze în mediul digital, dar care au persoane apropiate ce le pot ajuta cu integrare în lumea aplicațiile web. Acest sub sistem a fost conceput pentru a oferi o securitate sporită conturilor pacienților, setând un nume de utilizator și o parolă ce nu pot fi reținute foarte ușor.

În cadrul rapoartelor cu date de teste vor fi atașate câteva din documentele care utilizează aceste sisteme.

4. EFICIENȚA ȘI UTILITATEA APLICAȚIEI

4.1. Condiții privind implementarea aplicației

Deoarece discutăm despre o aplicație informatică bazată pe tehnologii web, implicit vorbim și despre o arhitectură de tip client-server de care aceasta are nevoie pentru a putea fi utilizată. După cum am menționat anterior în capitolul *2.9 Estimarea calendarului de resurse și a calendarului de realizare* firma Routine Med S.R.L. deja dispune de un sistem de calcul pentru a fi configurat pe post de server al aplicației acesteia, urmând ca pe viitor acesta să fie înlocuit când se ating valori ale utilizării acestuia ce depășesc pragul de 80 puncte procentuale din quantumul resurselor disponibile.

Odată stabilită infrastructura este nevoie să se instaleze aplicația propriu zisă și setul de dependințe aferent acesteia. Procedura de instalare se va efectua de către dezvoltatorul aplicației și se va face în perioada de 72 de ore de la data ieșirii din etapa incipientă de dezvoltare.

Deoarece aplicația este una realizată special pentru personalul clinicii Routine Med S.R.L., în cadrul etapei de implementare se vor realiza și ședințe de pregătire în care aceștia vor utiliza o variantă izolată a produsului software pentru a proba toate facilitățile în condiții de muncă simulate. Această etapă este esențială, deoarece în cadrul acestor ședințe, atât angajații clinicii beneficiază de o instruire de înaltă calitate, dar și dezvoltatorul de aplicație poate identifica posibile cazuri care nu au fost luate în calcul în fazele de proiectare și programare a produsului software, ceea ce poate duce la o dezvoltare armonioasă a aplicației verificând dacă problema a fost din culpa clientului, bazată pe exprimarea ambiguă asupra cerințelor aplicației informaticе, sau din cauza dezvoltatorului care va trebui să repare erorile stabilite.

În urma ședințelor de pregătire se va scrie o documentație simplificată pe baza cazurilor și nivelului de cunoștințe al personalul, documentație ce va fi pusă la dispoziție din prima zi a punerii aplicației în funcțiune. Această documentație are rolul de a amortiza nevoia de suport tehnic din partea dezvoltatorului, totodată permitând formarea mai facilă a noilor angajați.

4.2. Exploatarea curentă a aplicație

În prezent aplicația este exploatață cu succes de către clinica medicală la sediul din localitatea Sulina, acesta fiind un sediu cu un volum redus de muncă, comparativ cu cel din str. Mahmudiei, din localitatea Tulcea.

S-a făcut alegerea utilizării soluției software în cadrul acestui sediu deoarece aplicația nu se află încă în etapă finală, dar cu ajutorul utilizatorilor, aceasta va putea cu ușurință să ruleze la nivelul fiecărei clinici din rețeaua Routine Med S.R.L.

Deoarece prezintă facilității cu un puternic caracter interdisciplinar, în special pe zona economică, aplicația informatică poate satisface în prezent un nivel de muncă al unui economist, ajutând astfel la procesele de management ale firmei prin intermediul raporturilor pe care aceasta le pune la dispoziția utilizatorului

4.3. Considerații privind eficiența aplicației

Luând în calcul standardele cantitative și calitative ale aplicației informaticе realizată în prezența lucrare, putem observa că acestea sunt într-o relație de interdependență una față de cealaltă.

Din punct de vedere al eficienței cantitative considerăm eficiența ca fiind dată de timpii foarte buni de încărcare a paginilor și de viteza de prelucrare a datelor, acestea fiind date în special de infrastructura hardware și software ce se descurcă în prezent cu gestionarea cererilor de date la nivelul aplicației.

Totuși aplicația ar putea fi mai rapidă din punct de vedere a vitezei dacă se utiliza doar limbajul de programare fără a se adăuga cadrul de dezvoltare Laravel peste. Avantajul aplicației bazată pe acest cadru de dezvoltare este reprezentată în schimb de viteza de lucru crescută și nivelul ridicat de securitate pe care îl implică aceasta.

Din punct de vedere calitativ, aplicația nu se prezintă excelent având un număr relativ mic de rapoarte destinate utilizatorului final. Eficiența calitativă o putem afla prin consultarea utilizatorilor aplicației folosind diverse instrumente de marketing cum ar fi sondajul de opinie și chestionarul.

Legătura dintre cele două componente este una indirect proporțională și se evidențiază prin scăderea eficienței calitative (din punct de vedere al vitezei de lucru), cu cât adăugam mai multe funcționalități. Ca și dezvoltatorii de aplicații, noi trebuie să ne asigurăm că respectăm un echilibru între cele două aspecte, iar în cazul înclinării cu o pondere mai mare spre o anumită parte a eficienței, să încercăm să restabilim echilibrul prin măsuri externe cum ar fi extinderea platformelor hardware și software, sau eficientizarea algoritmilor de prelucrare în cadrul funcționalităților complexe.

5. BIBLIOGRAFIE

- @dechim. (2009, 08 14). Transformare numere în litere. *Forumul comunității de PHP din România - PHPRomania.net*.
- Achour, M., Betz, F., Dovgal, A., Lopes, N., Magnusson, H., Richter, G., . . . Cowburn, P. (2022, 05 28). *PHP Manual*. Retrieved from PHP.net: <https://www.php.net/manual/en/>
- Bierer, D. (2021). *PHP 8 Programming Tips, Tricks and Best Practices: A practical guide to PHP 8 features, usage changes, and advanced programming techniques*. Packt.
- Botezatu, C. (2007). *Proiectarea Sistemelor Informaticice*. Pro Universitaria.
- Botezatu, C. (n.d.). *Note de curs - Proiectarea Sistemelor Informaticice*.
- Botezatu, M., Botezatu, C., & Căruțășu, G. (2016). *Managementul proiectelor informaticice*. Editura Pro Universitaria.
- Cornelia , B., & Iacob, I. (2007). *Proiectarea sistemelor informaticice. Studii de caz pentru managementul activităților unei societăți*. Pro Universitaria.
- Crisan, D., & Tabusca, A. (2004). *Programarea aplicațiilor web, limbajul HTML, Elemente de programare JavaScript Editia a II a*. Pro Universitaria.
- Duckett, J. (2014). *HTML and CSS: Design and Build Website*.
- Duckett, J. (2014). *JavaScript and jQuery: Interactive Front-End Web Development*.
- Duckett, J. (2022). *PHP & MySQL: Server-side Web Development*.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J., & Grady, B. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional.
- Iacob, I. (2009). *Oracle 10g*. Pro Universitaria.
- Laravel LLC. (2022, 05 28). *Documentatie Laravel versiunea 9.x*. Retrieved from Laravel.com: <https://laravel.com/docs/9.x>
- Martin, R. C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship*. Pearson.
- Pirnau, M., Botezatu , C., & Botezatu, C. (2010). GENERAL INFORMATION ON BUSINESS INTELLIGENCE AND OLAP SYSTEMS ARCHITECTURE. *The 2nd International Conference on Computer and Automation Engineering* .
- Stanciu, V. (n.d.). *Proiectarea Sistemelor Informaticice*.
- Ullman, L. (2005). *PHP și MySQL pentru site-uri dinamice*. Teora.

6. ANEXE

6.1. Anexe referitoare la documente și machete ale acestora

Anexa 1 – Fisă de programare



S.C. ROUTINE MED S.R.L.
Str. Mahmudie 49-51, Tulcea, Judet Tulcea
Tel. Fix: 0240 530 751
Mobil: 0728 212 508

FISA DE PROGARAMARE

#MAH-89329

Pacientul [REDACTED] a solicitat o programare la cabinetul **Oftalmologie** în data de **10 noiembrie 2021** la medicul specialist [REDACTED].

Programarea a fost realizată prin **telefon** și nu a fost platită anterior.

Anexa 2 – Factură Fiscală

Routine Med		FACTURA			
		Seria MAH-F Numarul 83 Data (ZZ-LL-AAAA) 06 01 2020			
Furnizor		Client			
S.C. ROUTINE MED S.R.L. Nr. Reg. Com: J36/202/2010 C.U.I./C.I.F.: 27028852 Adresă: Str. a II-a, nr.236A, Sulina, Jud. Tulcea		Nr. Reg. Com: _____ C.U.I./C.I.F./ Serie C.I.: _____ Adresa: _____			
IBAN: RO73 INGB 0000 0600 2018 5555 Banca Emitentă: ING Bank		IBAN: Banca Emitentă:			
Nr. Crt	Denumire produs/Serviciu	U.M.	Cantitate	Pret Unitar (Lei)	Valoarea (Lei)
1	Consultatie oftalmologica	BUC	1	150	150 5=3x4
Factura este valabilă fără semnătură și stampilă, dacă conține informațiile minime obligatorii stabilite prin art. 155 (19) din Codul Fiscal.					
Întocmit de: _____ CNP: _____ Serie C.I.: _____			TOTAL DE PLATA (RON) 150.00		
Numele Delegatului: Mijlocul de transport: Semnătura, Stampila Furnizorului			Semnătură de primire		

Anexa 3 – Chitanță

Furnizor S.C. ROUTINE MED S.R.L.
Reg. Com J36/202/2010
C.U.I./C.I.F. 27028852
Cont IBAN: RO73 INGB 0000 0600 2018 5555
Banca Emitentă: ING Bank
Adresă Str. a II-a, nr.236A, Sulina, Jud. Tulcea

CHITANȚĂ

Seria: MAH-CH Nr. 32
Data (zi-lună-an) 6 ianuarie 2022

Am primit de la _____

C.U.I./C.I.F./Serie C.I. : _____

Adresă: _____

Suma de 150 adică O suta cincizeci lei
Reprezentând contravaloarea parțială a facturii MAH-F 83 din data de 06.01.2022

Anexa 4 Machetă Fișă de programare



S.C. ROUTINE MED S.R.L.
Str. Mahmudiei 49-51, Tulcea, Județ Tulcea
Tel. Fix: 0240 530 751
Mobil: 0728 212 508

FISA DE PROGARAMARE NUMAR

PROGRAMARE - 1

Pacientul **PACIENT** a solicitat o programare la cabinetul **Cabinet** în data de **DATA PROGRAMARE**, **ORA PROGRAMARE** la medicul specialist **Medic Specialist**.

Programarea a fost realizată prin **modalitate Rezervare** și **Status Plata** anterior

Anexa 5 – Machetă Chitanță

Furnizor S.C. ROUTINE MED S.R.L.
Reg. Com J36/202/2010
C.U.I./C.I.F. 27028852
Cont IBAN: RO73 INGB 0000 0600 2018 5555
Banca Emitentă: ING Bank
Adresă Str. a II-a, nr.236A, Sulina, Jud. Tulcea

CHITANȚĂ

Seria: SERIE + NUMAR 1
Data (zi-lună-an) 04-01-2022

Am primit de la FIRMA

C.U.I./C.I.F./Serie C.I. : CUI

Adresă: ADRESA

Suma de adică ics lei

Reprezentând motiv emitere chităntă



Anexa 6– Machetă Factură Fiscală



FACTURA

Seria SERIE Numarul 1
Data (ZZ-LL-AAAA) 04-01- 2022

Furnizor	Client
----------	--------

S.C. ROUTINE MED S.R.L.
Nr. Reg. Com: J36/202/2010
C.U.I./C.I.F.: 27028852
Adresă: Str. a II-a, nr.236A, Sulina, Jud. Tulcea

CLIENT
Nr. Reg. Com: 1234
C.U.I./C.I.F./ Serie C.I.: CUI FIRMA
Adresa: ADRESA FIRMA

IBAN: RO73 INGB 0000 0600 2018 5555
Banca Emitenta: ING Bank

IBAN:
Banca Emitenta:

Nr. Crt	Denumire produs/Serviciu	U.M.	Cantitate	Pret Unitar (Lei)	Valoarea (Lei)
		1	2	3	4 5=3x4

Factura este valabilă fără semnătură și stampilă, dacă conține informațiile minime obligatorii stabilite prin art. 155 (19) din Codul Fiscal.

Întocmit de: ANGAJAT CNP: CNP ANGAJAT Serie ACT IDENTITATE RESPONSABIL FACTURARE	TOTAL DE PLATA (RON)	
Numele Delegatului: Mijlocul de transport:	Semnătură de primire	
Semnătura, Stampila Furnizorului		

6.2. Anexe referitoare la documentele generate de aplicația Informatică

6.2.1. Raport privind acordul de prelucrare a datelor cu caracter personal



S.C. ROUTINE MED S.R.L.
Nr. Reg. Com: J36/202/2010
C.U.I.: 27028852
office@routinemed.ro

Acord privind prelucrarea datelor cu caracter personal

Subsemnatul/a **Manole Teea**, posesor al CI/Pasaport cu seria **RM**, numarul **962436**, avand CNP **3320219253472**, în calitate de utilizator al platformei software **myRoutineMed**, îmi exprim acordul cu privire la utilizarea și prelucrarea datelor mele personale de către operatorul privat S.C. Routine Med S.R.L. în scopul îndeplinirii activităților specifice privind operarea pacientilor dar și în scopuri de marketing, cu respectarea prevederilor legale.

Am fost informat că datele furnizate vor fi tratate confidențial, în conformitate cu prevederile art. 18 din Legea nr. 544 din 2001 privind liberul acces la informațiile de interes public, cu respectarea prevederilor Regulamentului (UE) 2016/679 privind protecția persoanelor fizice în ceea ce privește prelucrarea datelor cu caracter personal și privind libera circulație a acestor date și de abrogare a Directivei 95/46/CE (Regulamentul general privind protecția datelor) și cu Legea nr. 190/2018 privind măsuri de punere în aplicare a Regulamentului (UE) 2016/679 al Parlamentului European și al Consiliului din 27 aprilie 2016 privind protecția persoanelor fizice în ceea ce privește prelucrarea datelor cu caracter personal și privind libera circulație a acestor date și de abrogare a Directivei 95/46/CE (Regulamentul general privind protecția datelor).

Data

07-07-2022

Semnatura Pacient

6.2.2. Raport generat automat cu datele noului pacient

	S.C. ROUTINE MED S.R.L. Nr. Reg. Com: J36/202/2010 C.U.I.: 27028852 office@routinemed.ro
Datele pacientului:	
Constantin Iordanescu	
Atribut	Valoare
Nume	Iordanescu
Prenume	Constantin
Data Nasterii	2000-03-14
CNP	5001010398721
Serie C.I.	BQ
Numar C.I.	498739
Judet	Buzau
Localitate	Buzau
Strada	Magnoliei
Numar	48
Bloc	
Scara	
Etaj	
Apartament	
Email	test@popessti.ro
Telefon	0748999540
Nume utilizator	test@popessti.ro
Parola	j3F8pPf8LF

6.2.3. Raport Fisă de programare



S.C. ROUTINE MED S.R.L.

Nr. Reg. Com: J36/202/2010

C.U.I.: 27028852

office@routinemed.ro

Fisa de programare

Stimate client, te anuntam cu respect ca programarea dumneavoastra in vederea Consult Medical de Specialitate în baza programării cu numărul #2

Sunteti asteptat la sediul nostru din str. Mahmudiei Nr. 49-51, Localitatea Tulcea, Judet Tulcea, la cabinetul Oftalmo 1, medicul Manole Teea fiind cel care te va ajuta cu acest caz medical.

Va asteptam in data de 2022-07-18 la ora 09:00:00 și vă recomandăm să veniti cu 15 minute mai devreme pentru evitarea unor posibile neplaceri.

La momentul generarii prezentei fise de programare, stadiul acesteia este de Preluata.

Iti multumim ca ai ales serviciile medicale oferite de noi !

Cu stima si respect !

Echipa Routine Med S.R.L

6.2.4. Raport adeverință medicală



S.C. ROUTINE MED S.R.L.

Nr. Reg. Com: J36/202/2010

C.U.I.: 27028852

office@routinemed.ro

Medic: Manole Teea

Cabinet: Oftalmo 1

Specializarea: Oftalmologie

Adeverinta Medicala

Se adeverește că **Profesor URA**, născut la data de **1949-09-27** cu domiciliul în județul **Bucuresti**, localitatea **Bumbești-Jiu**, adresa Splaiul Zidarilor nr. 69 având ocupatia de **Profesor** în cadrul **Universitatii Romano-Americane**,

Este suferind de: **Cornee Inflamata**

Se recomanda: **Mutarea acestuia într-un departament cu mai puțin praf**

S-a eliberat pentru a-i servii la **Locul de munca**

Data: **2022-07-18**

Cod Parafa: 85864812

6.2.5. Raport scrisoare medicală



S.C. ROUTINE MED S.R.L.
Nr. Reg. Com: J36/202/2010
C.U.I.: 27028852
office@routinemed.ro

Medic: Manole Teea
Cabinet: Oftalmo 1
Specializarea: Oftalmologie

Scrisoare Medicală

Stimate(ă) coleg(ă), vă informăm că **Profesor URA**, născut la data de **1949-09-27** având CNP/Cod Unic de Asigurare **2880204294959**, a fost consultat în serviciul nostru la data de **2022-07-18**.

Motivele Prezentarii:

Corp străin intrat în cavitatea optică

Diagnosticul:

Corp străin a infectat stratul inferior al corneei ochiului drept

Anamneza:

Un corp străin a intrat prin aruncare în cornea pacientului

Factori risc: Orbire

Examenul Clinic:

S-a încercat scoaterea corpului străin

Examene de laborator:

- cu valori normale:
- cu valori patologice:

Examene paraclinice

EKG:

ECO:

Rx:

Alte Examene: Tehniune oculară 17, peste medie

Tratament Efectuat

Vizik 2 Picături / 12 Ore

Data: 2022-07-18

Cod Parafă: 85864812

6.2.6. Raport Factură Fiscală

	<h2 style="margin: 0;">FACTURA FISCALĂ</h2> <p style="margin: 0;">Seria RM Numar: 1</p> <p style="margin: 0;">Data Emitterii: 2022-07-10</p> <p style="margin: 0;">Data Scadentei: 2022-07-25</p>																																																
FURNIZOR S.C. ROUTINE MED S.R.L. Nr. Reg. Com: J36/202/2010 C.U.I.: 27028852 Adresa: Str. a II - A, Nr. 236A Loc.: Sulina, Jud.: Tulcea Cont Bancar: RO00INGB852195729647256 Banca Emitenta: ING Bank Email: contact@routinemed.ro	CLIENT S.C. SERVICEIMO ELECTRONIC S.R.L. Nr. Reg. Com: J36/239/2018 C.U.I.: 39309134 Adresa: Str. Dobrogeanu Gherea, Nr. 1 Bl.: E2 Sc.: A Et.: E2 Ap.: 2 Loc.: Tulcea, Jud.: Tulcea Cont Bancar: RO73RZBR0000060020185555 Banca Emitenta: Raiffaisen Bank Sucursala Tulcea Email: office@serviceimo.ro																																																
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 5%;">Nr. Crt</th> <th style="width: 45%;">Denumire Produs/Serviciu</th> <th style="width: 10%;">Unitate Masura</th> <th style="width: 10%;">Cantitate</th> <th style="width: 10%;">Pret Unitar fara TVA</th> <th style="width: 10%;">Valoare Totala fara TVA</th> <th style="width: 10%;">Cota TVA</th> <th style="width: 10%;">Valoare totala cu TVA</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">1</td> <td>Prestarii servicii medicale conform programarii #1 Eliberare Adeverinta Medicala</td> <td style="text-align: center;">BUC</td> <td style="text-align: center;">1</td> <td style="text-align: center;">10</td> <td style="text-align: center;">10</td> <td style="text-align: center;">19 %</td> <td style="text-align: center;">11.9</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Prestarii servicii medicale conform programarii #1 Consult Medical de specialitate</td> <td style="text-align: center;">BUC</td> <td style="text-align: center;">1</td> <td style="text-align: center;">150</td> <td style="text-align: center;">150</td> <td style="text-align: center;">19 %</td> <td style="text-align: center;">178.5</td> </tr> <tr> <td colspan="6" style="text-align: center;">TOTAL FARA TVA</td> <td colspan="2" style="text-align: center;">160 RON</td> </tr> <tr> <td colspan="6" style="text-align: center;">VAL TVA 19%</td> <td colspan="2" style="text-align: center;">30.4 RON</td> </tr> <tr> <td colspan="6" style="text-align: center;">TOTAL</td> <td colspan="2" style="text-align: center;">190.4 RON</td> </tr> </tbody> </table>		Nr. Crt	Denumire Produs/Serviciu	Unitate Masura	Cantitate	Pret Unitar fara TVA	Valoare Totala fara TVA	Cota TVA	Valoare totala cu TVA	1	Prestarii servicii medicale conform programarii #1 Eliberare Adeverinta Medicala	BUC	1	10	10	19 %	11.9	2	Prestarii servicii medicale conform programarii #1 Consult Medical de specialitate	BUC	1	150	150	19 %	178.5	TOTAL FARA TVA						160 RON		VAL TVA 19%						30.4 RON		TOTAL						190.4 RON	
Nr. Crt	Denumire Produs/Serviciu	Unitate Masura	Cantitate	Pret Unitar fara TVA	Valoare Totala fara TVA	Cota TVA	Valoare totala cu TVA																																										
1	Prestarii servicii medicale conform programarii #1 Eliberare Adeverinta Medicala	BUC	1	10	10	19 %	11.9																																										
2	Prestarii servicii medicale conform programarii #1 Consult Medical de specialitate	BUC	1	150	150	19 %	178.5																																										
TOTAL FARA TVA						160 RON																																											
VAL TVA 19%						30.4 RON																																											
TOTAL						190.4 RON																																											
Responsabil Facturare: Nume: Niculae Alex CNP: 3411024141577 Act Identitate: RM 404145																																																	
Semnatura Furnizor <hr/>	Semnatura Client <hr/>																																																

6.2.7. Raport Chitanță



CHITANTA

Seria RM-C | Numar: 1
Data Emiterii: 2022-07-10

S.C. ROUTINE MED S.R.L.

Nr. Reg. Com: J36/202/2010

C.U.I.: 27028852

Adresa: Str. a II - A, Nr. 236A

Loc.: Sulina, Jud.: Tulcea

Am primit de la **S.C. SERVICEIMO ELECTRONIC S.R.L.** identificata prin C.U.I./C.I.F. **39309134** avand sediul în **Str. Dobrogeanu Gherea, Nr. 1 Bl.: E2 Sc.: A Et.: E2 Ap.: 2 Loc.: Tulcea, Jud.: Tulcea**

Suma de **190.4 LEI** adică **UnaSutaNouaZeci lei si PatruZeci bani**

Reprezentând contravaloarea Facturii Fiscale cu seria **RM-F** cu numărul #**1** emisă la data de **2022-07-10**



CHITANTA

Seria RM-C | Numar: 1
Data Emiterii: 2022-07-10

S.C. ROUTINE MED S.R.L.

Nr. Reg. Com: J36/202/2010

C.U.I.: 27028852

Adresa: Str. a II - A, Nr. 236A

Loc.: Sulina, Jud.: Tulcea

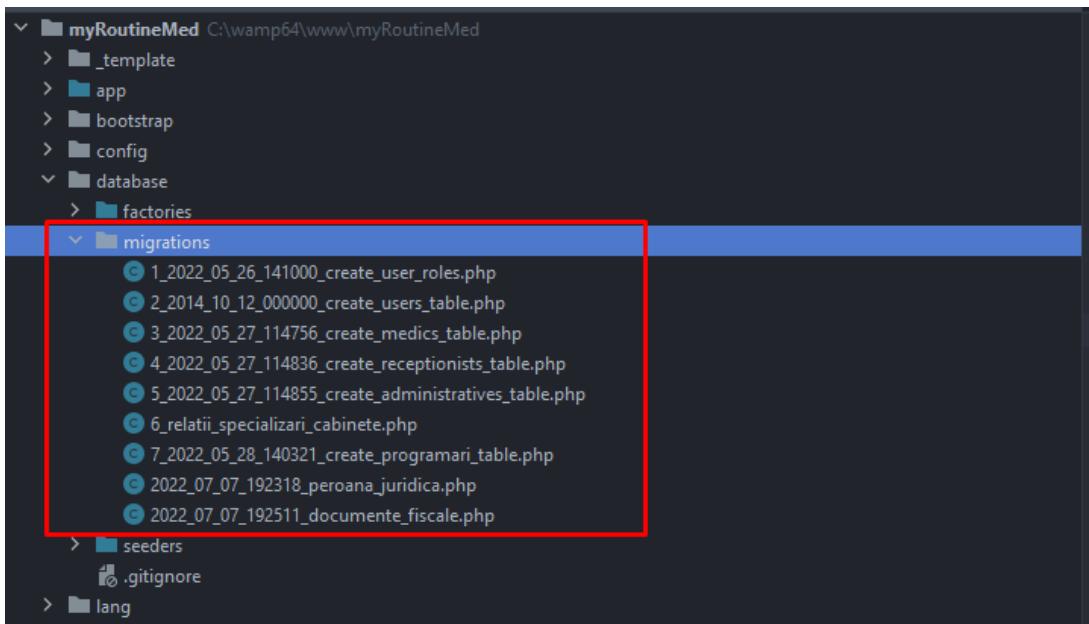
Am primit de la **S.C. SERVICEIMO ELECTRONIC S.R.L.** identificata prin C.U.I./C.I.F. **39309134** avand sediul în **Str. Dobrogeanu Gherea, Nr. 1 Bl.: E2 Sc.: A Et.: E2 Ap.: 2 Loc.: Tulcea, Jud.: Tulcea**

Suma de **190.4 LEI** adică **UnaSutaNouaZeci lei si PatruZeci bani**

Reprezentând contravaloarea Facturii Fiscale cu seria **RM-F** cu numărul #**1** emisă la data de **2022-07-10**

6.3. Anexe referitoare la codul sursă al aplicației

6.3.1. Fișierele generate de comanda de generare a migrațiilor



6.3.2 Structura unei migrații

```
public function up()
{
    Schema::create('user_roles', function (Blueprint $table) {
        $table->id();
        $table->string('rol');
    });

    //Inserarea valorilor de baza la migratie
    DB::table('user_roles')->insert(
        [
            'rol' => 'Pacient'
        ]);
    DB::table('user_roles')->insert(
        [
            'rol' => 'Recepție'
        ]);
    DB::table('user_roles')->insert(
        [
            'rol' => 'Medic'
        ]);
    DB::table('user_roles')->insert(
        [
            'rol' => 'Conducere'
        ]);
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('user_roles');
}
};
```

6.3.3 Migrăție pentru tabela users

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('username');
        $table->string('nume');
        $table->string('prenume');
        $table->string('CNP')->nullable();
        $table->string('serieCI')->nullable();
        $table->string('nrCI')->nullable();
        $table->string('telefon')->nullable();
        $table->date('DOB')->nullable();
        $table->string('Judet')->nullable();
        $table->string('localitate')->nullable();
        $table->string('strada')->nullable();
        $table->string('numar')->nullable();
        $table->string('bloc')->nullable();
        $table->string('scara')->nullable();
        $table->string('etaj')->nullable();
        $table->string('apartament')->nullable();
        $table->foreignId('role_id')->default(1)->constrained('user_roles','id')-
>onDelete('cascade');
        $table->string('email')->unique();
        $table->string('password')->default('');
    });
}
/** 
 * @return void
 */
public function down()
{
    Schema::dropIfExists('users');
}
};
```

6.3.4 Migrăție pentru tabela medics

```
public function up()
{
    Schema::create('medics', function (Blueprint $table) {
        $table->id();
        $table->foreignID('codUser')->constrained('users','id');
        $table->string('numarParafa')->nullable();
        $table->string('codCuloare')->default('C44100');
        $table->string('codSpecializare')->nullable();
        $table->boolean('activ');
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('medics');
}
```

6.3.5 Migrăție pentru tabela receptionists

```
public function up()
{
    Schema::create('receptionists', function (Blueprint $table) {
        $table->id();
        $table->foreignId('codUser')->constrained('users','id');
        $table->boolean('activ');
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('receptionists');
}
```

6.3.6 Migrăție parțială pentru tabelele programări, documente, documente_emise, servicii prestate și consultații – 1/3

```
public function up()
{
    Schema::create('tip_programare', function (Blueprint $table) {
        $table->id();
        $table->string('scopProgramare');
        $table->integer('durata');
    });

    DB::table('tip_programare')->insert(['scopProgramare' => 'Consult Medical de Specialitate',
'durata' => 30]);
    DB::table('tip_programare')->insert(['scopProgramare' => 'Eliberare adeverinta medicala',
'durata' => 10]);

    Schema::create('status_programari', function (Blueprint $table) {
        $table->id();
        $table->string('status');
    });

    DB::table('status_programari')->insert(['status' => 'In asteptare']);
    DB::table('status_programari')->insert(['status' => 'Preluata']);
    DB::table('status_programari')->insert(['status' => 'Finalizata']);
    DB::table('status_programari')->insert(['status' => 'Respinsa']);
    DB::table('status_programari')->insert(['status' => 'Anulata']);

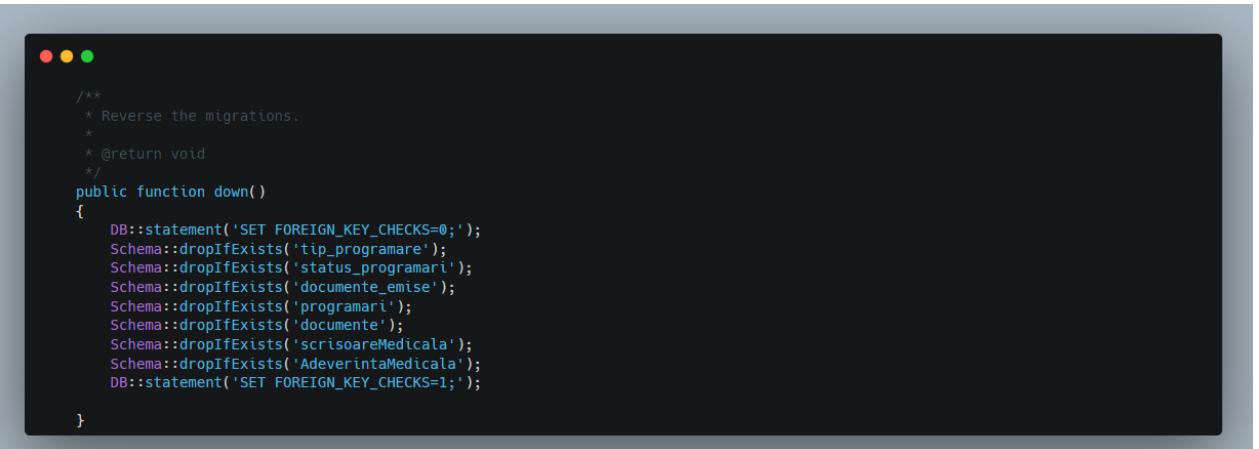
    Schema::create('programari', function (Blueprint $table) {
        $table->id();
        $table->foreignID('idPacient')->constrained('users', 'id')->onDelete('cascade');
        $table->foreignID('idMedic')->constrained('medics', 'id')->onDelete('cascade');
        $table->foreignID('idCabinet')->constrained('cabinete', 'id')->onDelete('cascade');
        $table->date('DataProgramarii');
        $table->time('OraProgramarii');
        $table->integer('tipProgramare');
        $table->string('statusProgramare');
        $table->string('codDocumentFinanciar')->nullable();
        $table->timestamps();
    });
}
```

6.3.7 Migrăție parțială pentru tabelele programări, documente, documente_emise, servicii prestate și consultații – 2/3

```
Schema::create('scrisoareMedicala', function (Blueprint $table) {
    $table->id();
    $table->foreignID('idProgramare')->constrained('consultatii','id')->onDelete('cascade');
    $table->text('Motivele_Prezentarii')->nullable();
    $table->text('Diagnosticul')->nullable();
    $table->text('Anamneza')->nullable();
    $table->text('Factori_Risc_Anamneza')->nullable();
    $table->text('ExamenClinic')->nullable();
    $table->text('ExameneLab_ValNormale')->nullable();
    $table->text('ExameneLab_ValPatologice')->nullable();
    $table->text('Examene_EKG')->nullable();
    $table->text('Examene_ECO')->nullable();
    $table->text('Examene_RX')->nullable();
    $table->text('Examene_AlteExamene')->nullable();
    $table->text('Tratament_Efectuat')->nullable();
    $table->integer('onorariuMedic')->nullable();
});

Schema::create('AdeverintaMedicala', function (Blueprint $table) {
    $table->id();
    $table->foreignID('idProgramare')->constrained('programarii','id')->onDelete('cascade');
    $table->string('Ocupatia');
    $table->string('LocDeMunca');
    $table->string('SuferindDe');
    $table->string('Recomandare');
    $table->string('Destinatie');
    $table->integer('onorariuMedic')->nullable();
});
}
```

6.3.8 Migrăție parțială pentru tabelele programări, documente, documente_emise, servicii prestate și consultații – 3/3



```
/*
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    DB::statement('SET FOREIGN_KEY_CHECKS=0;');
    Schema::dropIfExists('tip_programare');
    Schema::dropIfExists('status_programari');
    Schema::dropIfExists('documente_emise');
    Schema::dropIfExists('programari');
    Schema::dropIfExists('documente');
    Schema::dropIfExists('scrisoareMedicala');
    Schema::dropIfExists('AdeverintaMedicala');
    DB::statement('SET FOREIGN_KEY_CHECKS=1;');
}
```

6.3.9 Migrăție pentru tabela administratives



```
public function up()
{
    Schema::create('administratives', function (Blueprint $table) {
        $table->id();
        $table->foreignId('codUser')->constrained('users','id');
        $table->boolean('activ');
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('administratives');
}
```

6.3.10 Migrăție pentru tabelele specializari, cabinete, working_hours

```
public function up()
{
    Schema::create('specializaris', function (Blueprint $table) {
        $table->id();
        $table->string('specializare');
    });

    DB::table('specializaris')->insert(
        [
            'specializare' => 'Oftalmologie'
        ]);
    DB::table('specializaris')->insert(
        [
            'specializare' => 'Fizioterapie'
        ]);
}

Schema::create('cabinete', function (Blueprint $table) {
    $table->id();
    $table->foreignId('codSpecializare')->constrained('specializari','id')-
>onDelete('cascade');
    $table->string('denumireCabinet');
});

Schema::create('working_hours', function (Blueprint $table) {
    $table->id();
    $table->foreignId('codMedic')->constrained('medics','id')->onDelete('cascade');
    $table->foreignId('codCabinet')->constrained('cabinete','id')->onDelete('cascade');
    $table->smallInteger('codZi');
    $table->time('startConsultatii');
    $table->time('stopConsultatii');
});

/** 
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('working_hours');
    Schema::dropIfExists('cabinete');
    Schema::dropIfExists('specializaris');
}
};|
```

6.3.11 Procedură de tip factory pentru modelul user

```

class UserFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition()
    {
        return [
            'username' =>$this->faker->userName(),
            'nume' => $this->faker->lastName(),
            'prenume' => $this->faker->firstName(),
            'email' => $this->faker->unique()->safeEmail(),
            'CNP' => $this->faker->cnp(),
            'serieCI' => 'RM',
            'nrCI' => $this->faker->randomNumber(6),
            'telefon' =>$this->faker->phoneNumber(),
            'Judet' => 'Bucuresti',
            'DOB'=> $this->faker->dateTimeThisCentury,
            'localitate' => $this->faker->city(),
            'strada' => $this->faker->streetName(),
            'numar' =>$this->faker->randomNumber('2'),
            'password' => '$2y$10$92IXUNpkjO0rQ05byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', // password
        ];
    }
}

```

6.3.12 Comanda dedicată sistemului de migrații

```

C:\wamp64\www\myRoutineMed>php artisan migrate:refresh --seed
Rolling back: 7_2022_05_28_140321_create_programari_table
Rolled back: 7_2022_05_28_140321_create_programari_table (80.09ms)
Rolling back: 6_relatii_specializari_cabinete
Rolled back: 6_relatii_specializari_cabinete (14.22ms)
Rolling back: 5_2022_05_27_114855_create_administratives_table
Rolled back: 5_2022_05_27_114855_create_administratives_table (4.76ms)
Rolling back: 4_2022_05_27_114836_create_receptionists_table
Rolled back: 4_2022_05_27_114836_create_receptionists_table (4.84ms)
Rolling back: 3_2022_05_27_114756_create_medics_table
Rolled back: 3_2022_05_27_114756_create_medics_table (6.41ms)
Rolling back: 2022_07_07_192511_documente_fiscale
Rolled back: 2022_07_07_192511_documente_fiscale (17.19ms)
Rolling back: 2022_07_07_192318_peroana_juridica
Rolled back: 2022_07_07_192318_peroana_juridica (4.77ms)
Rolling back: 2019_12_14_000001_create_personal_access_tokens_table
Rolled back: 2019_12_14_000001_create_personal_access_tokens_table (4.73ms)
Rolling back: 2_2014_10_12_000000_create_users_table
Rolled back: 2_2014_10_12_000000_create_users_table (6.19ms)
Rolling back: 1_2022_05_26_141000_create_user_roles
Rolled back: 1_2022_05_26_141000_create_user_roles (6.94ms)
Migrating: 1_2022_05_26_141000_create_user_roles
Migrated: 1_2022_05_26_141000_create_user_roles (21.08ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (33.71ms)
Migrating: 2022_07_07_192318_peroana_juridica
Migrated: 2022_07_07_192318_peroana_juridica (10.02ms)
Migrating: 2022_07_07_192511_documente_fiscale
Migrated: 2022_07_07_192511_documente_fiscale (19.09ms)
Migrating: 2_2014_10_12_000000_create_users_table
Migrated: 2_2014_10_12_000000_create_users_table (50.62ms)
Migrating: 3_2022_05_27_114756_create_medics_table
Migrated: 3_2022_05_27_114756_create_medics_table (22.27ms)
Migrating: 4_2022_05_27_114836_create_receptionists_table
Migrated: 4_2022_05_27_114836_create_receptionists_table (20.10ms)
Migrating: 5_2022_05_27_114855_create_administratives_table
Migrated: 5_2022_05_27_114855_create_administratives_table (14.94ms)
Migrating: 6_relatii_specializari_cabinete
Migrated: 6_relatii_specializari_cabinete (62.46ms)
Migrating: 7_2022_05_28_140321_create_programari_table
Migrated: 7_2022_05_28_140321_create_programari_table (107.66ms)
Database seeding completed successfully.

```

C:\wamp64\www\myRoutineMed>

6.3.13 Rezultat output migrație phpMyAdmin

+ Options																							
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	id	username	nume	prenume	CNP	serieCI	nrCI	telefon	DOB	Judet	localitate	strada	numar	bloc	scara	etaj	apartament	role_id	email	password
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1	prof.rau	Profesor	URA	2880204294959	RM	827621	0313416648	1949-09-27	Bucuresti	Bumbești-Jiu	Splaiul Zidarilor	69	NULL	NULL	NULL	NULL	4	radu-mihai@serviceimo.ro	\$2a\$12\$GtF89We5GeFNH90RFfKZuEoCk2WucThG0kIx69DG...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2	umoldovan	Manole	Teea	3320219253472	RM	962436	0271711317	1923-09-01	Bucuresti	Cluj-Napoca	B-dul. Piersicului	73	NULL	NULL	NULL	NULL	1	ncazacu@example.net	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	dserban	Niculae	Alex	3411024141577	RM	404145	0727600974	2021-11-14	Bucuresti	Mun. Tulcea	Splaiul Mihai Eminescu	43	NULL	NULL	NULL	NULL	1	mirona61@example.org	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4	negrea.leonard	Badea	Nicoleta	1480623373952	RM	777650	0769485771	1925-07-11	Bucuresti	Salcea	Calea Croitorilor	84	NULL	NULL	NULL	NULL	1	lazar.rus@example.org	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	5	stanciu.ovidiu	Nemes	Pamela	3230801273780	RM	345788	0254714794	1951-09-19	Bucuresti	Scornicești	B-dul. Franklin Delano Roosevelt	64	NULL	NULL	NULL	NULL	1	licia.mihalcea@example.net	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	6	aleodor.apostol	Roman	Nechifor	2840824294027	RM	791026	0749506448	2015-02-12	Bucuresti	Vicovu de Sus	P-ța 1 Decembrie	29	NULL	NULL	NULL	NULL	1	edgar18@example.org	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	7	iacob.dimitrina	Lazar	Amedeu	4790701290789	RM	960009	0278457944	2017-09-14	Bucuresti	Mun. Alexandria	Str. Învățătorului	37	NULL	NULL	NULL	NULL	1	ianis.mirea@example.org	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	8	edgar.rosu	Mihai	Cristea	3511006422237	RM	404587	0753882120	1968-02-06	Bucuresti	Mun. Sărmanie	Aleea J.J. Rousseau	78	NULL	NULL	NULL	NULL	1	ngiurgiu@example.net	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	9	voicu.eusebia	Miu	Ozana	6090609361318	RM	704877	0702054832	1949-08-14	Bucuresti	Slănic-Moldova	Aleea Memorandumului	64	NULL	NULL	NULL	NULL	1	mariuca91@example.org	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	10	nicolescu.nechifor	Biro	Olimpian	1880529406714	RM	623826	0790002257	1935-01-16	Bucuresti	Mun. Târgu Jiu	P-ța Eroilor	75	NULL	NULL	NULL	NULL	1	jan.ursu@example.net	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	11	mdumitrascu	Draghici	Emil	6650118270714	RM	865180	0353106858	2006-12-26	Bucuresti	Mun. Nucet	P-ța Sinaia	28	NULL	NULL	NULL	NULL	1	ion.roxana@example.com	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12	cbucur	Stanescu	Luana	4980421171874	RM	394180	0377169782	1981-08-21	Bucuresti	Recaș	Aleea Cireșilor	3	NULL	NULL	NULL	NULL	1	umateescu@example.org	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	13	dina09	Dinca	Andrei	2390907227769	RM	231236	0791855231	1950-09-20	Bucuresti	Făsin	Splaiul Decebal	27	NULL	NULL	NULL	NULL	1	morar.gregorian@example.org	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	14	mitica.moise	Alexe	Voicu	1391227217985	RM	562924	0270302000	1938-02-25	Bucuresti	Mun. Oțelu Roșu	Str. Muncii	4	NULL	NULL	NULL	NULL	1	bsolomon@example.net	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	15	oana.paraschiv	Ilieșcu	Ianis	2930825290751	RM	619443	0775487416	1968-09-14	Bucuresti	Fetești	Splaiul Unirii	27	NULL	NULL	NULL	NULL	1	simonescu.simi@example.org	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	16	soancea	Martin	Niculina	5991006069771	RM	244952	0703248883	1977-03-15	Bucuresti	Mun. Sinaia	Splaiul Salcămilor	64	NULL	NULL	NULL	NULL	1	ivan.tudor@example.org	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	17	tudorita.peter	Sarbu	Oliviu	1100805136869	RM	18378	0332777690	1947-09-12	Bucuresti	Crăiova	Str. Salcămilor	77	NULL	NULL	NULL	NULL	1	qmarginean@example.net	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	18	nagy.alina	Suciul	Oana	6040716081028	RM	759731	0785259179	2016-06-21	Bucuresti	Mun. Găești	P-ța Cireșilor	93	NULL	NULL	NULL	NULL	1	micu.sidonia@example.org	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	19	caterina19	Clobanu	Georgel	4050520400081	RM	939387	0347818190	2017-11-27	Bucuresti	Mun. Găești	P-ța Sinaia	33	NULL	NULL	NULL	NULL	1	dumitru87@example.org	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	20	mitru20	Chiriac	Dumitra	5511026282991	RM	221819	0332230532	1951-12-22	Bucuresti	Bacău	Str. Mesterilor	96	NULL	NULL	NULL	NULL	1	maximilian.negrea@example.net	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	21	bucur.nistor	Nechita	Cristofor	6020709054765	RM	51778	0745245398	1984-11-09	Bucuresti	Mun. Beiuș	Str. Unirii	83	NULL	NULL	NULL	NULL	1	marga.lupu@example.com	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	22	savina41	Bota	Zeno	1881103461021	RM	705645	0231257264	1961-04-07	Bucuresti	Strehia	Calea Croitorilor	20	NULL	NULL	NULL	NULL	1	pistrate@example.net	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	23	sorin.sandu	Oprea	Iulian	6861209071349	RM	656484	0705962422	2007-10-22	Bucuresti	Mun. Pecești	Splaiul Vlad Tepes	42	NULL	NULL	NULL	NULL	1	kduta@example.net	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...
<input checked="" type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	24	ada61	Mihalache	Doru	1241026363543	RM	166128	0703459826	1975-09-28	Bucuresti	Ruzias	Aleea	56	NULL	NULL	NULL	NULL	1	ichiriacr@example.org	\$2y\$10\$92IXUNpkjO0rOQ5byMi.Ye4oKoEa3Ro9lIC/.og/at2...

6.3.14 Modelul User

```
use HasApiTokens, HasFactory, Notifiable;

/**
 * The attributes that are mass assignable.
 *
 * @var array<int, string>
 */
protected $fillable = [
    'nume',
    'prenume',
    'username',
    'CNP',
    'serieCI',
    'nrCI',
    'telefon',
    'dob',
    'judet',
    'localitate',
    'strada',
    'numar',
    'bloc',
    'scara',
    'etaj',
    'apartament',
    'email',
    'password',
    'role_id'
];

/**
 * The attributes that should be hidden for serialization.
 *
 * @var array<int, string>
 */
protected $hidden = [
    'password',
    'remember_token',
];

/**
 * The attributes that should be cast.
 *
 * @var array<string, string>
 */
protected $casts = [
    'email_verified_at' => 'datetime',
];
```

6.3.15 Rutele dedicate procesului de autentificare

```
/* Rute Autentificare */
Route::get('/', function () {return view('dashboard');})->middleware('auth');

Route::get('/login', [UserController::class, 'loginPage'])->name('login');
Route::POST('/loginProcess',[UserController::class, 'loginProcces']);
Route::get('/register',[UserController::class, 'registerPage']);
Route::POST('/registerProcces',[UserController::class, 'registerProcces']);
Route::get('/register/{inviteCode}',[UserController::class, 'registerbyInvite']); // DE LUCRAT
ROUTE::GET('/logout', [UserController::class, 'logout']);
```

6.3.16 Prezentarea Controller-ului UserController 1/4

```
class UserController extends Controller
{
    public function loginPage()
    {
        return view('auth.login');
    }
    public function registerPage()
    {
        return view('auth.register');
    }

    public function loginProcces(Request $request)
    {
        $userValidator2 = request()->validate([
            'username' => 'required',
            'password' => ['required', 'min:6'], // Pentru a folosi confirmed trebuie sa folosim campul password_confirmation
        ]);
        if(auth()->attempt($userValidator2))
        {
            $request->session()->regenerate();
            $user = auth()->user();

            if($user->CNP == null){
                return redirect('/updatePatient/'.$user->id);
            }
            return redirect('/')->with("message", 'Te-ai conectat cu succes');
        }
        return back();
    }

    public function registerProcces(Request $request)
    {
        $validareCampuri = request()->validate(
            [
                'nume' => 'required',
                'prenume' => 'required',
                'username' => ['required', 'min:6', Rule::unique('users','username')],
                'email' => ['required', 'email', Rule::unique('users','email')],
                'password' => ['required','confirmed', 'min:8']
            ]
        );
        $validareCampuri['password'] = bcrypt($validareCampuri['password']);

        $user = User::Create($validareCampuri);
        auth()->login($user);
        if($user->CNP == null){
            return redirect('/updatePatient/'.$user->id);
        }
        return redirect('/');
    }
}
```

6.3.17 Prezentarea Controller-ului UserController 2/4

```
public function logout()
{
    auth()->logout(auth()->user());
    return redirect('/login');
}

public function returnUpdateView($id, Request $request)
{
    if($id == auth()->user()->id || auth()->user()->role_id > 1)
    {
        $selectedUser = User::find($id);
        if($selectedUser != null){
            return view('patient.update',[

                'selectedUser' => $selectedUser
            ]);
        }
        else
        {
            abort(404);
        }
    }else
    {
        abort(401);
    }
}
```

6.3.18 Prezentarea Controller-ului UserController 3/4

```

public function updateUserInDB(Request $request, $id)
{
    $selectedUser = User::find($id);
    $validareCampuriObligatorii = request()->validate(
        [
            'nume' => 'required',
            'prenume' => 'required',
            'CNP' => ['required', 'max:13'],
            'serieCI' => 'required',
            'nrCI' => 'required',
            'email' => ['required', 'email'],
            'telefon' => 'required',
            'dob' => 'required',
            'judet' => 'required',
            'localitate' => 'required',
            'strada' => 'required',
            'numar' => 'required',
            'role_id' => 'required',
        ]
    );

    $campuriObtionale = [
        'bloc' => $request->bloc,
        'scara' => $request->scara,
        'etaj' => $request->etaj,
        'apartament' => $request->apartament,
    ];
    $toInsertinDB = array_merge($validareCampuriObligatorii, $campuriObtionale);
    $functii_tabele = ([
        1 => 'users',
        2 => 'receptionists',
        3 => 'medics',
        4 => 'administratives'
    ]);

    if($toInsertinDB['role_id'] != $selectedUser->role_id)
    {
        $role_id = $toInsertinDB['role_id'];
        if ($selectedUser->role_id != 1) {

            //Schimbam valoarea curenta a utilizatorului la 0 pentru a nu pierde datele
            DB::update('UPDATE '.$functii_tabele[$selectedUser->role_id]. ' set activ = ? WHERE codUser = ?',[0,$selectedUser->id]);
            //Verificam daca utilizatorul a mai avut aceasta functie inainte pentru a nu suprascrie informatiile
            if($role_id != 1)
            {
                if(DB::table($functii_tabele[$role_id])->where('codUser',$selectedUser->id)->exists()){
                    DB::update('UPDATE '.$functii_tabele[$role_id]. ' set activ = ? WHERE codUser = ?',[1,$selectedUser->id]);
                }
                else{ // Adaugam direct noua linie de cod
                    DB::INSERT('INSERT INTO ' . $functii_tabele[$role_id]. ' (codUser, activ) VALUES (?, ?,)', [$selectedUser->id, 1]);
                }
            }
            else {
                if($role_id != 1) DB::INSERT('INSERT INTO ' . $functii_tabele[$role_id]. ' (codUser, activ) VALUES (?, ?,)', [$selectedUser->id, 1]);
            }
        }
        $selectedUser->update($toInsertinDB);

        if(auth()->user()->role_id >= 4){
            return redirect('/manageUsers');
        }else
        {
            return redirect('/');
        }
    }
}

```

6.3.19 Prezentarea Controller-ului UserController 4/4

```

public function returnCreateUserView(Request $request)
{
    return view('patient.add');
}

public function createNewUser(Request $request)
{
    $validareCampuriObligatorii = request()->validate(
        [
            'nume' => 'required',
            'prenume' => 'required',
            'CNP' => ['required', 'max:13'],
            'serieCI' => 'required',
            'nrCI' => 'required',
            'email' => ['required', 'email'],
            'telefon' => 'required',
            'dob' => 'required',
            'judet' => 'required',
            'localitate' => 'required',
            'strada' => 'required',
            'numar' => 'required',
        ]
    );

    $campuriObtionale = [
        'bloc' => $request->bloc,
        'scara' => $request->scara,
        'etaj' => $request->etaj,
        'apartament' => $request->apartament,
    ];
    $password = Str::random(10);
    $campuriAutentificare = [
        'password' => bcrypt($password),
        'username' => $validareCampuriObligatorii['email']
    ];
    $validareCampuri = array_merge($validareCampuriObligatorii, $campuriObtionale, $campuriAutentificare);
    $user = User::Create($validareCampuri);

    $date = [
        'user' =>$user,
        'password' => $password
    ];

    $pdf = app('dompdf.wrapper');
    $pdf->loadView('patient.printPatientData',$date);
    return $pdf->stream('Datele Utilizatorului'. '$user->prenume.' . '$user->nume.'.pdf');
}

public function downloadPDFPatient(User $user, $password){
    $date = [
        'user' =>$user,
        'password' => $password
    ];

    $pdf = app('dompdf.wrapper');
    $pdf->loadView('patient.printPatientData',$date);
    return $pdf->stream('info_pacient.pdf');

}

public function getTesteView()
{
    return view('teste');
}
}

```

6.3.20 Pagina de administrare a utilizatorilor 1/3

```

@extends('main')
@section('content')

@php
use App\Models\User;

$currUser = auth()->user();
$users = User::simplePaginate(30);
// dd($currUser->nume, '$currUser->prenume');
@endphp



<header class="mb-3">
        <a href="#" class="burger-btn d-block d-xl-none">
            <i class="bi bi-justify fs-3"></i>
        </a>
    </header>

    <div class="page-heading">
        <div class="page-title">
            <div class="row">
                <div class="col-12 col-md-6 order-md-1 order-last">
                    <h3>Utilizatori ai platformei</h3>
                </div>
                <div class="col-12 col-md-6 order-md-2 order-first">
                    <nav aria-label="breadcrumb" class="breadcrumb-header float-start float-lg-end">
                        <ol class="breadcrumb">
                            <li class="breadcrumb-item"><a href="index.html">Dashboard</a></li>
                            <li class="breadcrumb-item active" aria-current="page">DataTable</li>
                        </ol>
                    </nav>
                </div>
            </div>
        </div>
        <section class="section">
            <div class="card">
                <div class="card-header">
                    Simple Datatable
                </div>
                <div class="card-body">
                    <table class="table table-striped" id="table1">
                        <thead>
                            <tr>
                                <th>Nume</th>
                                <th>Prenume</th>
                                <th>Email</th>
                                <th>Funcție</th>
                                <th>Status</th>
                                <th>Operatii User</th>
                            </tr>
                        </thead>
                        <tbody>
                            @foreach ($users as $table_user)

                                @php
                                    $functie = '';
                                    switch ($table_user->getUserFunction()) {
                                        case '1':
                                            $functie = 'Pacient';
                                            break;
                                        case '2':
                                            $functie = 'Receptionist';
                                            break;
                                        case '3':
                                            $functie = 'Medic';
                                            break;
                                        case '4':
                                            $functie = 'Administrativ';
                                            break;
                                        default:
                                            # code...
                                            break;
                                    }
                                @endphp



Pag. 109


```

6.3.21 Pagina de administrare a utilizatorilor 2/3

```

@endphp

<tr>
    <td>{{ $table_user->nume }}</td>
    <td>{{ $table_user->prenume }}</td>
    <td>{{ $table_user->email }}</td>
    <td>{{ $functie }}</td>
@php
$isActive = false;
switch ($functie) {
    case 'Administrativ':
        if($table_user->administrative()->get('activ')[0]->activ == 1) $isActive = true;
        break;
    case 'Medic':
        if($table_user->medic()->get('activ')[0]->activ == 1) $isActive = true;
        break;
    case 'Receptionist':
        if($table_user->receptionist()->get('activ')[0]->activ == 1) $isActive = true;
        break;
    default:
        $isActive = true;
        break;
}
@endphp

    <td>
        @if ($isActive)
            <span class="badge bg-success">Activ</span>
        @else <span class="badge bg-danger">Dezactivat</span>
        @endif
    </td>
    <td>
        <i class="bi bi-person-square" data-bs-toggle="modal"
           data-bs-target="#userInfo.$table_user->id"></i>
        <a href="/updatePatient/$table_user->id"><i class="bi bi-person-rolodex"></i></a>
        <a href="..//raportGDPR/{{$table_user->id}}" target="_blank"><i class="bi bi-shield-lock-fill"></i>
    </td>
</tr>

{{-- Modal dedicat pentru aparitia informatiilor despre utilizator --}}
<div class="modal fade text-left" id="{{$userInfo.$table_user->id}}" tabindex="-1" role="dialog"
     aria-labelledby="myModalLabel160" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered modal-dialog-scrollable"
         role="document">
        <div class="modal-content">
            <div class="modal-header bg-primary">
                <h5 class="modal-title white" id="myModalLabel160">Datele Utilizatorului {{$table_user->username}}</h5>
            </div>
            <button type="button" class="close" data-bs-dismiss="modal"
                   aria-label="Close">
                <i data-feather="x"></i>
            </button>
        </div>
        <div class="modal-body">
            <p>
                <b>Nume:</b> {{$table_user->nume}}<br>
                <b>Prenume:</b> {{$table_user->prenume}}<br>
                <b>Data Nasterii:</b> {{$table_user->DOB}}<br>
                <b>CNP:</b> {{$table_user->CNP}}<br>
                <b>Carte Identitate:</b> {{$table_user->serieCI.' '.$table_user->nrCI}}<br>
                <b>Telefon:</b> {{$table_user->telefon}}<br>
                <b>Email:</b> {{$table_user->email}}<br>
                <b>Judet:</b> {{$table_user->judet}}<br>
                <b>Localitate:</b> {{$table_user->localitate}}<br>
                <b>Adresa:</b> {{$table_user->strada.' Nr. '.$table_user->numar.
                    ' Bl. '.$table_user->bloc.' Sc. '.$table_user->scara.' Et. '.$table_user->etaj.
                    ' Ap. '.$table_user->apartament}}<br>
                <b>Functie:</b> {{$functie}}<br>
            </p>
        </div>
    </div>
</div>

```

6.3.22 Pagina de administrare a utilizatorilor 3/3

```

</p>
</div>
{{-- <div class="modal-footer">
    <button type="button" class="btn btn-light-secondary"
        data-bs-dismiss="modal">
        <i class="bx bx-x d-block d-sm-none"></i>
        <span class="d-none d-sm-block">Close</span>
    </button>
    <button type="button" class="btn btn-primary ml-1"
        data-bs-dismiss="modal">
        <i class="bx bx-check d-block d-sm-none"></i>
        <span class="d-none d-sm-block">Accept</span>
    </button>
</div> --}}
</div>
</div>
@endforeach
</tbody>
</table>
</div>
</div>
{{{$users->links()}}}

<!--primary theme Modal -->
</div>
</section>
</div>

```

6.3.23 Rute pentru cabinete și programări

```

/*Rute pentru cabinete*/
ROUTE::GET('/addNewCabinet',[Cabinete::class, 'returnAddCabinetView'])->middleware('head.staff');
ROUTE::GET('/programCabinet/{id}', [Cabinete::class, 'returnManageCabinetView']);
ROUTE::GET('/listCabinete',[Cabinete::class, 'getCabineteForListing']);
ROUTE::PUT('/updateCabinetProces',[Cabinete::class, 'updateCabinetOnDB']);
ROUTE::GET('/getCabineteBySpecializare/{specializare}',[Cabinete::class, 'getCabinetBySpecializare']); // RUTA AJAX

/* Rute pentru programari */
ROUTE::GET('/adaugaProgramare', [ProgramariController::class, 'returnAddProgramareView']);
ROUTE::PUT('/validateProgramare', [ProgramariController::class,'validateProgramare']);

ROUTE::GET('/listaProgramari',[ProgramariController::class, 'getAllProgramariView']);
ROUTE::GET('/calendarProgramari',[ProgramariController::class, 'getAllProgramariCalendarView']); //DE INTEGRAT
ROUTE::GET('/istoricProgramari',[ProgramariController::class , 'getUserProgramari']);
ROUTE::GET('/Programare/{id}', [ProgramariController::class, 'returnUpdateProgramare']);

ROUTE::GET('/fisaProgramare/{idProgramare}',[ProgramariController::class, 'generateFisaProgramare']);

```

6.3.24 Codul sursă aferent pagini de adăugare a cabinetelor

```
● ● ●

@include('partials/_header')

@php
$listaSpecializari = \App\Models\specializari::all();
@endphp

<section id="basic-horizontal-layouts">
    <div class="row match-height">
        <div class="col-12">
            <div class="card">
                <h5 class="card-header">Adauga Cabinet</h5>
                <div class="card-body">
                    <form method="POST" action="/updateCabinetProces" class="form form-horizontal">
                        @csrf
                        @method('PUT');
                        <div class="form-body">
                            <div class="row">
                                <div class="col-md-4">
                                    <label>Denumire Cabinet</label>
                                </div>
                                <div class="col-md-8 form-group">
                                    <input type="text" id="denumireCabinet" class="form-control" name="denumireCabinet"
                                           placeholder="Denumire Cabinet">
                                </div>
                                <div class="col-md-4">
                                    <label>Specializarea Cabinetului</label>
                                </div>
                                <div class="col-md-8 form-group">
                                    <select name="codSpecializare" id="codSpecializare" class="form-select">
                                        @foreach($listaSpecializari as $specializare)
                                            <option value="{{ $specializare->id }}>{{ $specializare->specializare }}</option>
                                        @endforeach
                                    </select>
                                </div>
                                <div class="col-sm-12 d-flex justify-content-end">
                                    <button type="submit" class="btn btn-primary me-1 mb-1">Adauga Cabinet</button>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</section>
@include('partials/_footer')
```

6.3.25 Codul sursă aferent pagini de adăugare a cabinetelor

```

@extends('main')
@section('content')

@php
    use App\Models\User;
    use Illuminate\Support\Facades\DB;

    $currUser = auth()->user();
    $cabinete = \App\Models\cabinete::simplePaginate(10);
    $specializari = \App\Models\specializari::all();

@endphp



<header class="mb-3">
        <a href="#" class="burger-btn d-block d-xl-none">
            <i class="bi bi-justify fs-3"></i>
        </a>
    </header>

    <div class="page-heading">
        <div class="row">
            <div class="col-12 col-md-6 order-md-1 order-last">
                <h3>Cabinete Disponibile</h3>
            </div>
        </div>
    </div>
    <section class="section">
        <div class="card">
            <div class="card-header">
                Simple Datatable
            </div>
            <div class="card-body">
                <table class="table table-striped" id="table1">
                    <thead>
                        <tr>
                            <th>Denumire Cabinet</th>
                            <th>Specializare</th>
                            <th>Medici</th>
                            <th>Actiuni</th>
                        </tr>
                    </thead>
                    <tbody>
                        @foreach($cabinete as $cabinet)
                            @php
                                $denumire_specializare = \App\Models\specializari::find($cabinet->codSpecializare);
                                $medics = array();
                            @endphp
                            <tr>
                                <td>{{ $cabinet->denumireCabinet }}</td>
                                <td>{{ $denumire_specializare->specializare }}</td>
                                <td>
                                    <table>
                                        @php
                                            $workingHours = DB::table('working_hours')->where('codCabinet', $cabinet->id)->get();
                                            @endphp
                                        @foreach ($workingHours as $sora)
                                            @php
                                                $currentMedic = \App\Models\medic::find($sora->codMedic);
                                                $userOfMedic = User::find($currentMedic->codUser);
                                                if(in_array($currentMedic->id, $medics)){
                                                    $publish = false;
                                                }else{
                                                    array_push($medics, $currentMedic->id);
                                                    $publish = true;
                                                }
                                            @endphp
                                            @endif
                                        @endforeach
                                    </table>
                                </td>
                                <td>
                                    <a href=".programCabinet/{{ $cabinet->id }}" class="btn btn-primary">
                                        <i class="bi bi-calendar-date">Program Cabinet</i>
                                    </a>
                                </td>
                            </tr>
                        @endforeach
                    </tbody>
                </table>
            </div>
        </div>
        {{ $cabinete->links() }}
        <!--primary theme Model -->
    </div>
</div>


```

6.3.26 Codul sursă aferent modulului de generare program de lucru al medicilor pe bază de calendar

```

@include('partials/_header')
@php
@endphp
<script src="/js/full-calendar/main.js"></script>
<script>
    function padTo2Digits(num)
    {
        return num.toString().padStart(2, '0');
    }

    function formatDate(date) {
        return [
            date.getFullYear(),
            padTo2Digits(date.getMonth() + 1),
            padTo2Digits(date.getDate()),
        ].join('-');
    }

    var today = new Date (Date.now());
    var currentDate = new Date();
    var luni = new Date();
    var marti = new Date();
    var miercuri = new Date();
    var joi = new Date();
    var vineri = new Date();
    var sambata = new Date();
    var duminica = new Date();

    switch (currentDate.getDay())
    {
        case 0: // Duminica
            luni.setDate(currentDate.getDate() - 6);
            marti.setDate(currentDate.getDate() - 5);
            miercuri.setDate(currentDate.getDate() - 4);
            joi.setDate(currentDate.getDate() - 3);
            vineri.setDate(currentDate.getDate() - 2);
            sambata.setDate(currentDate.getDate() - 1);
            duminica.setDate(currentDate.getDate());
            break;
        case 1: // Luni
            luni.setDate(currentDate.getDate());
            marti.setDate(currentDate.getDate() +1);
            miercuri.setDate(currentDate.getDate() +2);
            joi.setDate(currentDate.getDate() +3);
            vineri.setDate(currentDate.getDate() +4);
            sambata.setDate(currentDate.getDate() +5);
            duminica.setDate(currentDate.getDate() + 6);
            break;
        case 2: // Marti
            luni.setDate(currentDate.getDate() -1);
            marti.setDate(currentDate.getDate());
            miercuri.setDate(currentDate.getDate() +1);
            joi.setDate(currentDate.getDate() +2);
            vineri.setDate(currentDate.getDate() +3);
            sambata.setDate(currentDate.getDate() +4);
            duminica.setDate(currentDate.getDate() + 5);
            break;
        case 3: // Miercuri
            luni.setDate(currentDate.getDate() -2);
            marti.setDate(currentDate.getDate() -1);
            miercuri.setDate(currentDate.getDate());
            joi.setDate(currentDate.getDate() +1);
            vineri.setDate(currentDate.getDate() +2);
            sambata.setDate(currentDate.getDate() +3);
            duminica.setDate(currentDate.getDate() + 4);
            break;
    }

```

6.3.27 Codul sursă aferent modulului de generare program de lucru al medicilor pe bază de calendar 2

```
case 4: //Joi
    luni.setDate(currentDate.getDate() -3);
    marti.setDate(currentDate.getDate() -2);
    miercuri.setDate(currentDate.getDate() -1);
    joi.setDate(currentDate.getDate());
    vineri.setDate(currentDate.getDate() +1);
    sambata.setDate(currentDate.getDate() +2);
    duminica.setDate(currentDate.getDate() + 3);
    break;
case 5: //Vineri
    luni.setDate(currentDate.getDate() - 4);
    marti.setDate(currentDate.getDate() - 3);
    miercuri.setDate(currentDate.getDate() - 2);
    joi.setDate(currentDate.getDate() - 1);
    vineri.setDate(currentDate.getDate());
    sambata.setDate(currentDate.getDate() + 1);
    duminica.setDate(currentDate.getDate() + 2);
    break;
case 6: //Sambata
    luni.setDate(currentDate.getDate() - 5);
    marti.setDate(currentDate.getDate() - 4);
    miercuri.setDate(currentDate.getDate() - 3);
    joi.setDate(currentDate.getDate() - 2);
    vineri.setDate(currentDate.getDate() -1);
    sambata.setDate(currentDate.getDate() );
    duminica.setDate(currentDate.getDate() + 1);
    break;
}
var weekDays =
[
    luni,marti,miercuri,joi,vineri,sambata,duminica
];
document.addEventListener('DOMContentLoaded', function() {
    var calendarEl = document.getElementById('calendar');
    var calendar = new FullCalendar.Calendar(calendarEl, {
        initialView: 'timeGridWeek',
        themeSystem: 'bootstrap5',
        locale: 'ro',
        firstDay: 1,
        allDaySlot: false,
        slotMinTime: '07:00',
        slotMaxTime: '23:00',
        headerToolbar: {
            start: '',
            center: '',
            end: ''
        },
        footerToolbar: {
            prevYear: true,
            nextYear: true,
            prevMonth: true,
            nextMonth: true,
            today: true,
            prevWeek: true,
            nextWeek: true
        }
    });
    calendar.render();
});
```

6.3.28 Codul sursă aferent modulului de generare program de lucru al medicilor pe bază de calendar 3

```

eventSources: [
    // your event source
    {
        events: [ // put the array in the `events` property
            @foreach($selectedCabinet as $cabinet)
            {
                //Denumire MEDIC
                @php
                $medic = \App\Models\medic::find($cabinet->codMedic);
                $user = \App\Models\User::find($medic->codUser);

                @endphp
                title : '{$user->nume} {$user->prenume}' ,
                color : '{$medic->codCuloare}' ,
                @switch($cabinet->codZi)
                @case(1)
                start : formatDate(weekDays[0]) + 'T{$cabinet->startConsultatii}' ,
                end : formatDate(weekDays[0]) + 'T{$cabinet->stopConsultatii}' ,
                @break
                @case(2)
                start : formatDate(weekDays[1]) + 'T{$cabinet->startConsultatii}' ,
                end : formatDate(weekDays[1]) + 'T{$cabinet->stopConsultatii}' ,
                @break
                @case(3)
                start : formatDate(weekDays[2]) + 'T{$cabinet->startConsultatii}' ,
                end : formatDate(weekDays[2]) + 'T{$cabinet->stopConsultatii}' ,
                @break
                @case(4)
                start : formatDate(weekDays[3]) + 'T{$cabinet->startConsultatii}' ,
                end : formatDate(weekDays[3]) + 'T{$cabinet->stopConsultatii}' ,
                @break
                @case(5)
                start : formatDate(weekDays[4]) + 'T{$cabinet->startConsultatii}' ,
                end : formatDate(weekDays[4]) + 'T{$cabinet->stopConsultatii}' ,
                @break
                @case(6)
                start : formatDate(weekDays[5]) + 'T{$cabinet->startConsultatii}' ,
                end : formatDate(weekDays[5]) + 'T{$cabinet->stopConsultatii}' ,
                @break
                @case(7)
                start : formatDate(weekDays[6]) + 'T{$cabinet->startConsultatii}' ,
                end : formatDate(weekDays[6]) + 'T{$cabinet->stopConsultatii}' ,
                @break
            @endswitch
        },
        @endforeach
    ],
    // any other event sources...
],
dateClick: function() {
    alert('a day has been clicked!');
}
};

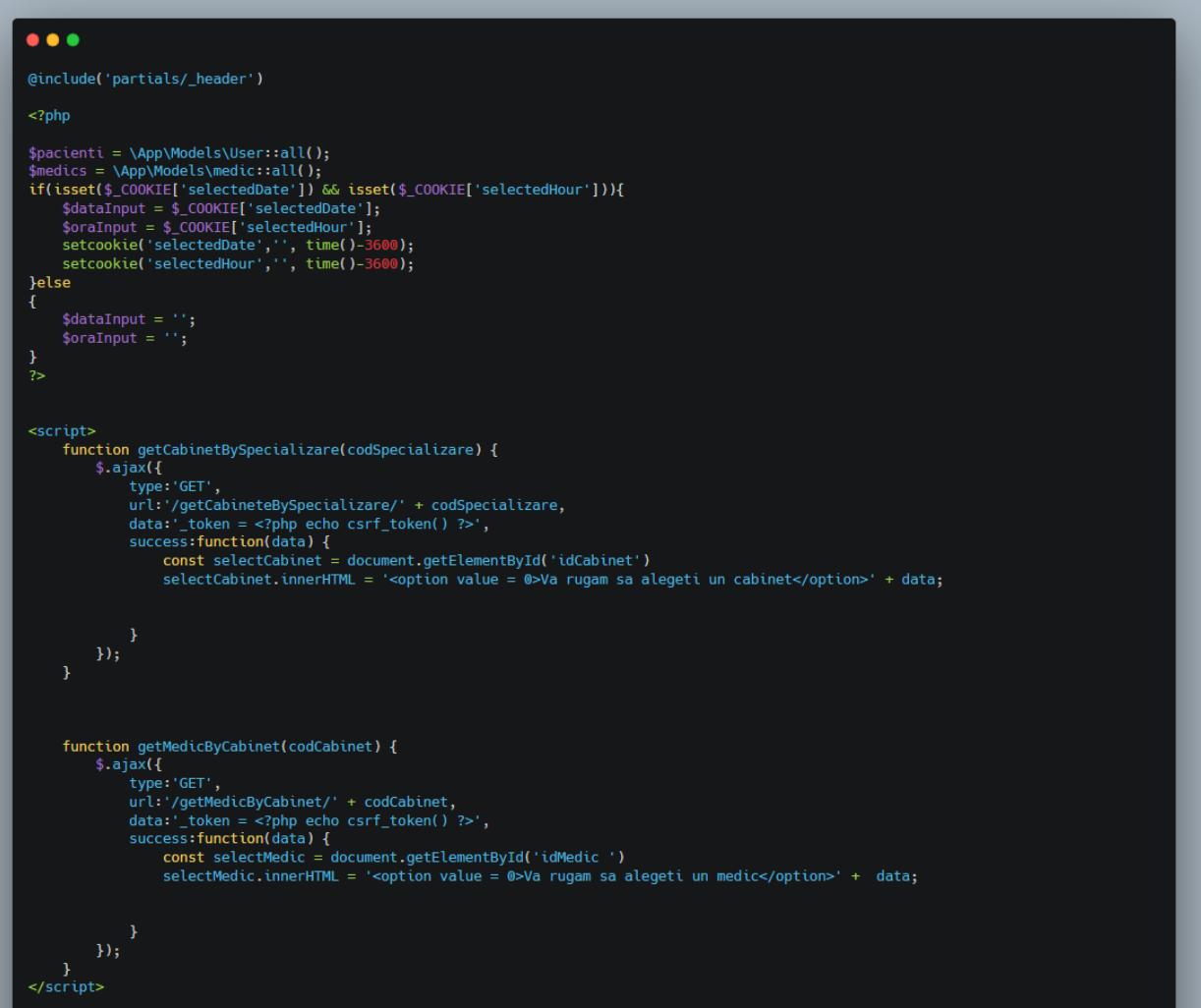
calendar.render();
});
</script>

```

6.3.29 Codul sursă aferent modulului de generare program de lucru al medicilor pe bază de calendar 4

```
@php
$cabinetCurrent = \App\Models\cabinete::find($codCabinet);
@endphp
<section id="basic-horizontal-layouts">
    <div class="row match-height">
        <div class="col-12">
            <div class="card">
                <h5 class="card-header">Programul Cabinetului {{ $cabinetCurrent->denumireCabinet }}</h5>
                <div class="card-body">
                    <form method="POST" action="" class="form form-horizontal">
                        @csrf
                        @method('PUT')
                    <div class="form-body">
                        <div id="calendar">
                            </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
</section>
@include('partials/_footer')
```

6.3.30 Codul sursă aferent view-ului de adaugare a unei programari -Scripturi AJAX



```
@include('partials/_header')

<?php

$pacienti = \App\Models\User::all();
$medics = \App\Models\medic::all();
if(isset($_COOKIE['selectedDate']) && isset($_COOKIE['selectedHour'])){
    $dataInput = $_COOKIE['selectedDate'];
    $oraInput = $_COOKIE['selectedHour'];
    setcookie('selectedDate','', time()-3600);
    setcookie('selectedHour','', time()-3600);
} else {
    $dataInput = '';
    $oraInput = '';
}
?>

<script>
    function getCabinetBySpecializare(codSpecializare) {
        $.ajax({
            type:'GET',
            url:'/getCabineteBySpecializare/' + codSpecializare,
            data:'_token = <?php echo csrf_token() ?>',
            success:function(data) {
                const selectCabinet = document.getElementById('idCabinet')
                selectCabinet.innerHTML = '<option value = 0>Va rugam sa alegeti un cabinet</option>' + data;
            }
        });
    }

    function getMedicByCabinet(codCabinet) {
        $.ajax({
            type:'GET',
            url:'/getMedicByCabinet/' + codCabinet,
            data:'_token = <?php echo csrf_token() ?>',
            success:function(data) {
                const selectMedic = document.getElementById('idMedic ')
                selectMedic.innerHTML = '<option value = 0>Va rugam sa alegeti un medic</option>' + data;
            }
        });
    }
</script>
```

6.3.31 Codul sursă aferent pagini de interfață pentru adăugare programare nou

```

<section id="multiple-column-form">
    <div class="row match-height">
        <div class="col-12">
            <div class="card">
                <div class="card-header">
                    <h4 class="card-title">Solicita o programare</h4>
                </div>
                <div class="card-content">
                    <div class="card-body">
                        <form METHOD="POST" class="form" action="../validateProgramare">
                            @csrf
                            @method('PUT')
                            <div class="row">

                                @if(auth()>user()>role_id > 1)
                                <div class="col-12">
                                    <div class="form-group">
                                        <label for="first-name-column">Pacient</label>
                                        @error('idPatient')
                                            {{$message}}
                                        @enderror
                                        <select name="idPatient" id="idPatient" class="form-select">
                                            @foreach($pacienti as $pacient)
                                                <option value="{{ $pacient->id }}>CNP: {{$pacient->CNP}} &nbsp; Nume: {{$pacient->nume}}
                                            {{ $pacient->prenume }}

```

6.3.32 Codul sursă aferent pagini de interfață pentru adăugare programare nou

```

        @endif
    <div class="col-12">
        <div class="form-group">
            <label for="Specializare">Specializare</label>
            @error('specializare')
            {{$message}}
            @enderror
            <select name="specializare" id="specializare" class="form-select"
onchange="getCabinetBySpecializare(this.value)">
                @php
                $listaSpecializari = \App\Models\specializari::all();
                @endphp
                <option value="0">Va rugam sa selectati o specializare</option>
                @foreach($listaSpecializari as $specializare)
                    <option value="{{$specializare->id}}>{{$specializare->specializare}}</option>
                @endforeach
            </select>
        </div>
    </div>

    <div class="col-12">
        <div class="form-group">
            <label for="last-name-column">Cabinet</label>
            @error('idCabinet')
            {{$message}}
            @enderror
            <select name="idCabinet" id="idCabinet" class="form-select"
onchange="getMedicByCabinet(this.value)">
                </select>
                <script>
                    $(document).ready(function() {
                        $('#idCabinet').select2();
                    });
                </script>
        </div>
    </div>

    <div class="col-12">
        <div class="form-group">
            <label for="last-name-column">Medic</label>
            @error('idMedic ')
            {{$message}}
            @enderror
            <select name="idMedic" id="idMedic " class="form-select">
                </select>
                <script>
                    $(document).ready(function() {
                        $('#idMedic').select2();
                    });
                </script>
        </div>
    </div>

        <div class="col-md-6 col-12">
            <div class="form-group">
                @error('dataProgramarii')
                {{$message}}
                @enderror
                <label for="city-column">Data Programarii</label>
                <input type="date" id="dataProgramarii" class="form-control" placeholder="CNP"
name="dataProgramarii" min="{{$date('Y-m-d')}}" value="{{$dataInput}}">
            </div>
        </div>

        <div class="col-md-6 col-12">
            <div class="form-group">
                @error('oraProgramarii')
                {{$message}}
                @enderror
                <label for="country-floating">Ora Programarii</label>
                <input type="time" id="oraProgramarii" class="form-control"
name="oraProgramarii" value = "{$oraInput}" >
            </div>
        </div>
    
```

6.3.33 Codul sursă aferent pagini de interfață pentru adăugare programare nou

```

        name="tipProgramare" value="{$tip->id}" >
    </div>
</div>

<div class="col-12">
    <div class="form-group">
        @error('tipProgramare')
        {$message}
        @enderror
        <label for="country-floating">Tipul Programarii</label>
        <select name="tipProgramare" id="tipProgramare" class="form-select">
            @php
                $tipuriProgramare = DB::table('tip_programare')->get();
            @endphp
            @foreach($tipuriProgramare as $tip)
                <option value="{{ $tip->id }}>{{ $tip->scopProgramare }}</option>
            @endforeach
        </select>
    </div>
</div>

@if(auth()->user()->role_id > 1)
    <div class="col-12">
        @error('statusProgramare')
        {$message}
        @enderror
        <div class="form-group">
            <label for="first-name-column">Status</label>
            <select name="statusProgramare" id="statusProgramare" class="form-select">
                @php
                    $statusProgramari = DB::table('status_programari')->get();
                @endphp
                @foreach($statusProgramari as $statusProgramare)
                    <option value="{{ $statusProgramare->id }}>{{ $statusProgramare->status }}</option>
                @endforeach
            </select>
        </div>
    </div>
@else
    <div class="col-12" hidden>
        @error('statusProgramare')
        {$message}
        @enderror
        <div class="form-group">
            <label for="first-name-column">Status</label>
            <select name="statusProgramare" id="statusProgramare" class="form-select" >
                @php
                    $statusProgramari = DB::table('status_programari')->get();
                @endphp
                @foreach($statusProgramari as $statusProgramare)
                    <option value="{{ $statusProgramare->id }}>{{ $statusProgramare->status }}</option>
                @endforeach
            </select>
        </div>
    </div>
@endif

<div class="col-12 d-flex justify-content-end">
    <button type="submit" class="btn btn-primary me-1 mb-1">Adauga Programare</button>
</div>

<div class="modal fade text-left" id="modal-eroare" tabindex="-1" role="dialog"
aria-labelledby="modal-eroare" aria-hidden="true">
<div class="modal-dialog modal-dialog-centered modal-dialog-scrollable"
role="document">
    <div class="modal-content">
        <div class="modal-header bg-danger">
            <h5 class="modal-title white" id="myModalLabel160"> A aparut o problema
            </h5>
            <button type="button" class="close" data-bs-dismiss="modal"
aria-label="Close">
                <i data-feather="x"></i>
            </button>
        </div>
    </div>
</div>

```



6.3.34 Codul sursă aferent pagini de interfață pentru adăugare programare nou

```
        <i data-feather="x"></i>
    </button>
</div>
<div class="modal-body">
    <p>{{Session::get('errorMessage')}}</p>
</div>

</div>
</form>
@if(Session::get('errorMessage') != null)
<script>
    $(document).ready(function(){
        $("#modal-eroare").modal('show');
    });
</script>
@endif
</div>
</div>
</div>
</div>
</section>

@include('partials/_footer')
```

6.3.35 Codul sursă aferent View-ului de listare al programărilor

```

@extends('main')
@section('content')

@php
if(auth()->user()->role_id == 3)
{
    $cuurentMedic = \App\Models\medic::where('codUser', auth()->user()->id)->get();
    $cuurentMedic = $cuurentMedic[0];
    $programari = \App\Models\programari::where('idMedic', $cuurentMedic->id )->get();
}

else
{
    $programari = \App\Models\programari::all();
}

@endphp



<header class="mb-3">
        <a href="#" class="burger-btn d-block d-xl-none">
            <i class="bi bi-justify fs-3"></i>
        </a>
    </header>

    <div class="page-heading">
        <div class="page-title">
            <div class="row">
                <div class="col-12 col-md-6 order-md-1 order-last">
                    <h3>Lista programarilor efectuate</h3>
                </div>
            </div>
        </div>
        <section class="section">
            <div class="card">
                <div class="card-body">
                    <div class="col-md-6 flex-row" >
                        <label for="">Data Start</label>
                        <input type="date" class="form-control" />
                        <label for="">Data Stop</label>
                        <input type="date" class="form-control" />
                    </div>
                    <table class="table table-striped" id="table1">
                        <thead>
                            <tr>
                                <th>Cod Programare</th>
                                <th>Patient</th>
                                <th>Cabinet</th>
                                <th>Medic</th>
                                <th>Data Programare</th>
                                <th>Ora Programare</th>
                                <th>Status</th>
                                <th>Tip Programare</th>
                                <th>Actiuni</th>
                            </tr>
                        </thead>
                        <tbody>
                            @foreach($programari as $programare)
                            <tr>
                                @php
                                    $currentPatient = \App\Models\User::find($programare->idPatient);
                                    $currentMedic = \App\Models\medic::find($programare->idMedic);
                                    $userOfMedic = \App\Models\User::find($currentMedic->codUser);
                                    $currentCabinet = \App\Models\cabinete::find($programare->idCabinet);
                                    $currentStatus = DB::table('status_programari')->where('id',$programare->statusProgramare)->get();
                                    $tipProgramari = DB::table('tip_programare')->where('id',$programare->tipProgramare)->get();
                                @endphp

                                <td>#{$programare->id}</td>
                                <td>{{$currentPatient->nume}} {{$currentPatient->prenume}}</td>
                                <td>{{$currentCabinet->denumireCabinet}}</td>
                                <td>{{$userOfMedic->nume}} {{$userOfMedic->prenume}}</td>
                                <td>{{$programare->DataProgramarii}}</td>
                                <td>{{$programare->OraProgramarii}}</td>
                                <td>{{$currentStatus[0]->status}}</td>
                                <td>{{$tipProgramari[0]->scopProgramare}}</td>
                                <td><a href=".//Programare/{$programare->id}"><button class="btn btn-primary">Editeaza Programarea</a></td>
                                <td><a href=".//fisaProgramare/{$programare->id}"><button class="btn btn-danger">Descarca Fisa de Programare</a></td>



Pag. 122


```

6.3.36 Codul sursă aferent vizualizării programărilor bazate pe calendar 1/2

```
@include('partials/_header')
@php
$programari = \App\Models\programari::all();
//dd($programari);
@endphp
<script src="/js/full-calendar/main.js"></script>
<script>
    function padTo2Digits(num)
    {
        return num.toString().padStart(2, '0');
    }

    function formatDate(date) {
        return [
            date.getFullYear(),
            padTo2Digits(date.getMonth() + 1),
            padTo2Digits(date.getDate()),
        ].join('-');
    }

    function addMinutes(date, minutes) {
        return new Date(date.getTime() + minutes*60000);
    }

    document.addEventListener('DOMContentLoaded', function() {
        var calendarEl = document.getElementById('calendar');
        var calendar = new FullCalendar.Calendar(calendarEl, {
            initialView: 'timeGridWeek',
            themeSystem: 'bootstrap5',
            locale: 'ro',
            firstDay: 1,
            allDaySlot: false,
            slotMinTime: '07:00',
            slotMaxTime: '23:00',
            headerToolbar: {
                start: '',
                center: '',
                end: 'prev, next'
            },
            eventSources: [
                {
                    url: '/api/programari'
                }
            ]
        });
        calendar.render();
    });
</script>
```

6.3.37 Codul sursă aferent vizualizării programărilor bazate pe calendar 2/2

```

        events: [
            @foreach($programari as $programare)
            {
                //Denumire MEDIC
                @php
                    $medic = \App\Models\medic::find($programare->idMedic);
                    $user = \App\Models\User::find($medic->codUser);
                    $time = 100;
                    $endTime = date('H:i', strtotime('+'.$time.'minutes', strtotime($programare->OraProgramarii)));
                @endphp
                title : '{{ $user->nume}} {{ $user->prenume}}',
                color : '{{ $medic->codCuloare}}',
                start : '{{ $programare->DataProgramarii}}T{{ $programare->OraProgramarii}}',
                end : '{{ $programare->DataProgramarii}}T{{ $endTime}}',
                eventClick: function() {
                    window.location.replace("/Programare/"+{{ $programare->id}});
                }
            },
            @endforeach
        ],
    }

    dateClick: function(date) {
        var data = date.dateStr.slice(0,10);
        var ora = date.dateStr.slice(11,16);
        document.cookie = 'selectedDate=' + data;
        document.cookie = 'selectedHour=' + ora;

        window.location.replace("../adaugaProgramare");
    }
}

);
calendar.render();
});

</script>

<section id="basic-horizontal-layouts">
    <div class="row match-height">
        <div class="col-12">
            <div class="card">
                <h5 class="card-header">Programări</h5>
                <div class="card-body">
                    <form method="POST" action="" class="form form-horizontal">
                        @csrf
                        @method('PUT')
                        <div class="form-body">
                            <div id="calendar">
                                </div>
                            </div>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>
</section>
@include('partials/_footer')

```

6.3.38 Formular dinamic pentru vizualizare și modificarea unei programări 1/6

```

@include('partials/_header')

<?php
$pacienti = \App\Models\User::all();
$medics = \App\Models\medic::all();
$medicalCertificates = DB::table('adeverintamedicala')->where('idProgramare', $programare->id)->get();
$medicalLetters = DB::table('scrisoaremedicala')->where('idProgramare', $programare->id)->get();

?>

<script>
    function getCabinetBySpecializare(codSpecializare) {
        $.ajax({
            type:'GET',
            url:'/getCabineteBySpecializare/' + codSpecializare,
            data:'_token = <?php echo csrf_token() ?>',
            success:function(data) {
                const selectCabinet = document.getElementById('idCabinet')
                selectCabinet.innerHTML = '<option value = @>Va rugam sa alegeti un cabinet</option>' + data;
            }
        });
    }

    function getMedicByCabinet(codCabinet) {
        $.ajax({
            type:'GET',
            url:'/getMedicByCabinet/' + codCabinet,
            data:'_token = <?php echo csrf_token() ?>',
            success:function(data) {
                const selectMedic = document.getElementById('idMedic ')
                selectMedic.innerHTML = '<option value = @>Va rugam sa alegeti un medic</option>' + data;
            }
        });
    }
</script>

<section id="multiple-column-form">
    <div class="row match-height">
        <div class="col-12">
            <div class="card">
                <div class="card-header">
                    <h1 class="card-title">PROGRAMARE</h1>
                </div>
                <div class="card-content">
                    <div class="card-body">
                        <form METHOD="POST" class="form" action="../validateProgramare">
                            @csrf
                            @method('PUT')
                            <div class="row">

                                @if(auth()->user()->role_id > 1)
                                <div class="col-12">
                                    <div class="form-group">
                                        <label for="first-name-column">Pacient</label>
                                        @error('idPatient')
                                            {{$message}}
                                        @enderror

```

6.3.39 Formular dinamic pentru vizualizare și modificarea unei programări 2/6

```

<select name="idPacient" id="idPatient" class="form-select">
    @foreach($pacienti as $pacient)
        @if($programare->idPacient == $pacient->id)
            <option selected value="{{$pacient->id}}>CNP: {{$pacient->CNP}} &nbsp;Nume:
            {{$pacient->nume}} {{$pacient->prenume}}</option>
        @endif
    @endforeach
    </select>
    <script>
        $(document).ready(function() {
            $('#idPatient').select2();
        });
    </script>

    </div>
</div>
@else
    @php
        $pacient = auth()->user();
    @endphp
    <div class="col-12">
        <div class="form-group">
            <label for="first-name-column">Pacient</label>
            @error('idPatient')
                {{$message}}
            @enderror
            <select name="idPatient" id="idPatient" class="form-select" readonly>
                <option value="{{$pacient->id}}>CNP: {{$pacient->CNP}} &nbsp;Nume: {{$pacient->nume}} {{$pacient->prenume}}</option>
            </select>
        </div>
    </div>
    @endif
    <div class="col-12">
        <div class="form-group">
            <label for="Specializare">Specializare</label>
            @error('specializare')
                {{$message}}
            @enderror
            <select name="specializare" id="specializare" class="form-select"
onchange="getCabinetBySpecializare(this.value)">
                @php
                    $listaSpecializari = \App\Models\specializari::all();
                    $selectedCabinet = \App\Models\cabinete::find($programare->idCabinet);
                    $selectedSpecializare = \App\Models\specializari::find($selectedCabinet->codSpecializare)
                @endphp
                @foreach($listaSpecializari as $specializare)
                    @if($selectedSpecializare->id == $specializare->id)
                        <option value="{{$specializare->id}}>{{$specializare->specializare}}</option>
                    @endif

                    @endforeach
                </select>
            </div>
        </div>
        <div class="col-12">
            <div class="form-group">
                <label for="last-name-column">Cabinet</label>
                @error('idCabinet')
                    {{$message}}
                @enderror
                <select name="idCabinet" id="idCabinet" class="form-select"
onchange="getMedicByCabinet(this.value)">
                    <option value="{{$selectedCabinet->id}}>{{$selectedCabinet->denumireCabinet}}</option>

```

6.3.40 Formular dinamic pentru vizualizare și modificarea unei programări 3/6

```

        </select>
        <script>
            $(document).ready(function() {
                $('#idCabinet').select2();
            });
        </script>
    </div>
</div>

<div class="col-12">
    <div class="form-group">
        <label for="last-name-column">Medic</label>
        @error('idMedic')
        {$message}
        @enderror
        @php
            $selectedMedic = \App\Models\medic::find($programare->idMedic);
            $userOfMedic = \App\Models\User::find($selectedMedic->codUser)
        @endphp
        <select name="idMedic" id="idMedic" class="form-select">
            <option value="{{ $selectedMedic->id }}>{{ $userOfMedic->nume }} {{ $userOfMedic->prenume }}>
        </select>
        <script>
            $(document).ready(function() {
                $('#idMedic').select2();
            });
        </script>
    </div>
</div>

<div class="col-md-6 col-12">
    <div class="form-group">
        @error('dataProgramarii')
        {$message}
        @enderror
        <label for="city-column">Data Programării</label>
        <input type="date" id="dataProgramarii" class="form-control" placeholder="CNP"
               name="dataProgramarii" value="{{ $programare->DataProgramarii }}"
               @if(auth()->user()->role_id == 1)
                   readonly
               @endif
        </div>
    </div>
<div class="col-md-6 col-12">
    <div class="form-group">
        @error('oraProgramarii')
        {$message}
        @enderror
        <label for="country-floating">Ora Programării</label>
        <input type="time" id="oraProgramarii" class="form-control"
               name="oraProgramarii" value="{{ $programare->OraProgramarii }}" src=""
               @if(auth()->user()->role_id == 1)
                   readonly
               @endif
        </div>
    </div>
<div class="col-12">
    <div class="form-group">
        @error('tipProgramare')
        {$message}
        @enderror
        <label for="country-floating">Tipul Programării</label>
        <select name="tipProgramare" id="tipProgramare" class="form-select">
            @if(auth()->user()->role_id == 1)
                readonly
            @endif
        </select>
    </div>
</div>

```

6.3.40 Formular dinamic pentru vizualizare și modificarea unei programări 4/6

```

        @php
            $tipuriProgramare = DB::table('tip_programare')->get();
            $selectedTipProgramare = $programare->tipProgramare;
        @endphp
        @foreach($tipuriProgramare as $tip)
            @if($selectedTipProgramare == $tip->id)
                <option value="{{ $tip->id }}">{{ $tip->scopProgramare }}</option>
            @endif
        @endforeach
        </select>
    </div>
</div>

@if(auth()->user()->role_id > 1)
    <div class="col-12">
        @error('statusProgramare')
        {{ $message}}
        @enderror
        <div class="form-group">
            <label for="first-name-column">Status</label>
            <select name="statusProgramare" id="statusProgramare" class="form-select">
                @switch($programare->statusProgramare)
                    @case (1) //In asteptare
                        <option value="1">In Asteptare</option>
                        <option value="2">Preluata</option>
                        <option value="3">Finalizata</option>
                        <option value="4">Respinsa</option>
                        <option value="5">Anulata</option>
                    @break
                    @case (2) //Preluata
                        <option value="2">Preluata</option>
                        <option value="3">Finalizata</option>
                        <option value="4">Respinsa</option>
                        <option value="5">Anulata</option>
                    @break
                    @case (3) //Finalizata
                        <option value="3">Finalizata</option>
                    @break
                    @case (4) //Respinsa
                        <option value="4">Respinsa</option>
                        <option value="3">Finalizata</option>
                    @break
                    @case (5) //Anulata
                        <option value="5">Respinsa</option>
                    @break
                @endswitch
            </select>
        </div>
    </div>
@else
    <div class="col-12" hidden>
        @error('statusProgramare')
        {{ $message}}
        @enderror
        <div class="form-group">
            <label for="first-name-column">Status</label>
            <select name="statusProgramare" id="statusProgramare" class="form-select" >
                @php
                    $statusProgramari = DB::table('status_programari')->get();
                @endphp
                @foreach($statusProgramari as $statusProgramare)
                    <option value="{{ $statusProgramare->id }}">{{ $statusProgramare->status }}</option>
                @endforeach
            </select>
        </div>
    </div>
@endif

```

6.3.41 Formular dinamic pentru vizualizarea și modificarea unei programări 5/6

```

<div class="col-12">
    @php
    @endphp
    @if($programare->statusProgramare != 4 || auth()->user()->role_id != 1)
        <label for="DocumenteEmise">Documente Emise</label>
        @foreach($medicalCertificates as $certificate)
            <a target="_blank" href="/document/MedicalCertificate/{{$certificate->id}}"
class="btn btn-success">Adeverinta Medicala #{{ $certificate->id }}</a>
        @endforeach
        @foreach($medicalLetters as $letter)
            <a target="_blank" href="/document/ScrisoareMedicala/{{ $letter->id }}" class="btn btn-
info">Scrisoare Medicala #{{ $letter->id }}</a>
        @endforeach
        @endif
        @if($programare->codDocumentFinanciar != null)
            @if(substr($programare->codDocumentFinanciar, 0, 2) == 'FF')
                <a target="_blank" href="/facturaFiscală/{{ substr($programare->codDocumentFinanciar,
2)}}"
class="btn btn-danger">Factura Fiscală #{{ substr($programare->codDocumentFinanciar, 2 )}}</a>
            @php
                $checkChitanta = DB::table('chitante')->where('numarFactura', substr($programare-
codDocumentFinanciar, 2))->get();
            @endphp
            @if(count($checkChitanta) != 0)
                @foreach ($checkChitanta as $chitanta)
                    <a target="_blank" href="/chitanta/{{ $chitanta->numarChitante }}"
class="btn
btn-danger">Chitanta #{{ $chitanta->numarChitante }}</a>
                @endforeach
            @endif
        @endif
    </div>
    <br>
    <br>
    <br>
    <br>

    <div class="col-6 justify-content-start">
        @if(auth()->user()->role_id != 1)
            @if($programare->codDocumentFinanciar == NULL)
                @if(auth()->user()->role_id == 3)
                    <a href="..> /document/newMedicalCertificate/{{ $programare->id }}"
class="btn btn-
success btn-sm rounded-pill" >Emite Adeverinta Medicala</a>
                    <a href="..> /document/newScrisoareMedicala/{{ $programare->id }}"
class="btn btn-
info btn-sm rounded-pill" >Emite Scrisoare Medicala</a>
                @endif
                <a href="..> /emiteFacturaProgramare/{{ $programare->id }}"
class="btn btn-warning btn-
sm rounded-pill">Achita programarea Persoana Juridica</a>
                <script>
                    function inLucru()
                    {
                        alert('Functie momentan in lucru');
                    }
                </script>
                <a href="" class="btn btn-secondary btn-sm rounded-pill" onclick="inLucru()>Achita
programarea Persoana Fizica</a>
            @endif
        @endif
    </div>
    <div class="col-4 d-flex justify-content-end">
        <input name="idProgramare" type="text" hidden value="{{ $programare->id }}>
        <button type="submit" class="btn btn-primary me-1 mb-1">Actualizeaza </button>
    </div>
<div class="modal fade text-left" id="modal-eroare" tabindex="-1" role="dialog"
aria-labelledby="modal-eroare" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered modal-dialog-scrollable">

```

6.3.42 Formular dinamic pentru vizualizarea și modificarea unei programări 6/6

```
<div class="modal-header bg-danger">
    <h5 class="modal-title white" id="myModalLabel160"> A aparut o problema
    </h5>
    <button type="button" class="close" data-bs-dismiss="modal"
            aria-label="Close">
        <i data-feather="x"></i>
    </button>
</div>
<div class="modal-body">
    <p>{{Session::get('errorMessage')}}</p>
</div>

</div>
</form>
@if(Session::get('errorMessage') != null)
    <script>
        $(document).ready(function(){
            $("#modal-eroare").modal('show');
        });
    </script>
    @endif
</div>
</div>
</div>
</div>
</section>

@include('partials/_footer')
```

6.3.43 Formular de tip raport pentru vizualizarea programărilor proprii

```

@extends('main')
@section('content')

    @php
        $programari = DB::table('programari')->where('idPacient', auth()->user()->id)->get();
    @endphp

    <div id="main">
        <header class="mb-3">
            <a href="#" class="burger-btn d-block d-xl-none">
                <i class="bi bi-justify fs-3"></i>
            </a>
        </header>

        <div class="page-heading">
            <div class="page-title">
                <div class="row">
                    <div class="col-12 col-md-6 order-md-1 order-last">
                        <h3>Lista programarilor efectuate de catre {{auth()->user()->nume}} {{auth()->user()->prenume}}</h3>
                    </div>
                </div>
            </div>
            <section class="section">
                <div class="card">
                    <div class="card-body">
                        <table class="table table-striped" id="table1">
                            <thead>
                                <tr>
                                    <th>Cod Programare</th>
                                    <th>Cabinet</th>
                                    <th>Medic</th>
                                    <th>Data Programare</th>
                                    <th>Ora Programare</th>
                                    <th>Status</th>
                                    <th>Actiuni</th>
                                </tr>
                            </thead>
                            <tbody>
                                @foreach($programari as $programare)
                                    <tr>
                                        @php
                                            $currentPatient = \App\Models\User::find($programare->idPatient);
                                            $currentMedic = \App\Models\medici::find($programare->idMedic);
                                            $userOfMedic = \App\Models\User::find($currentMedic->codUser);
                                            $currentCabinet = \App\Models\cabinete::find($programare->idCabinet);
                                            $currentStatus = DB::table('status_programari')->where('id',$programare->statusProgramare)->get();
                                        @endphp
                                        <td>#{$programare->id}</td>
                                        <td>{{$currentCabinet->denumireCabinet}}</td>
                                        <td>{{$userOfMedic->nume}} {{$userOfMedic->prenume}}</td>
                                        <td>{{$programare->DataProgramarii}}</td>
                                        <td>{{$programare->OraProgramarii}}</td>
                                        <td>{{$currentStatus[0]->status}}</td>
                                        <td>
                                            <a href=".//Programare/{{$programare->id}}"><button class="btn btn-primary">Vizualizeaza Programarea</button></a>
                                            <a href=".//fisaProgramare/{{$programare->id}}"><button class="btn btn-danger">Descarca Fisa de Programare</button></a>
                                        </td>
                                    </tr>
                                @endforeach
                            </tbody>
                        </table>
                    </div>
                </div>
            <!--primary theme Modal -->
        </div>
    </div>

```

6.3.44 Captură de ecran a codului aferent controller-ului Cabinete

```

● ● ●

<?php

namespace App\Http\Controllers;

use App\Models\specializari;
use App\Models\workingHour;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class Cabinete extends Controller
{
    public function returnAddCabinetView(){
        return view('cabinete.addCabinet');
    }

    public function returnManageCabinetView($idCabinet)
    {
        $selectedCabinet = DB::table('working_hours')->where('codCabinet', $idCabinet)->get();
        return view('cabinete.viewCalendar')->with(['selectedCabinet' => $selectedCabinet, 'codCabinet' => $idCabinet]);
    }

    public function updateCabinetOnDB(Request $request)
    {
        $validare = request()->validate([
            'codSpecializare' => 'required',
            'denumireCabinet' => 'required',
        ]);
        $cabinet = \App\Models\cabinete::Create($validare);
        return redirect('/');
    }

    public function getCabineteForListing()
    {
        return view('cabinete.listCabinete');
    }

    public function getCabinetBySpecializare($codSpecializare)
    {
        $cabinete = DB::table('cabinete')->where('codSpecializare', $codSpecializare)->get();
        $inputString = '';
        foreach ($cabinete as $cabinet)
        {
            $inputString = $inputString.'<option value = "'.$cabinet->id.'">'.$cabinet->denumireCabinet.'</option>';
        }
        return $inputString;
    }
}

```

6.3.45 Extras din codul sursă al controller-ului de programări

```

public function validateProgramare(Request $request)
{
    $validareCampuri = $request->validate([
        'idPatient' => ['different:0', 'required'],
        'idMedic' => ['different:0', 'required'],
        'specializare' => ['different:0', 'required'],
        'idCabinet' => ['different:0', 'required'],
        'dataProgramarii' => 'required',
        'oraProgramarii' => 'required',
        'tipProgramare' => 'required',
        'statusProgramare' => 'required'
    ]);

    $programari = programari::where('dataProgramarii',$request['dataProgramarii'])
        ->where('oraProgramarii',$request['oraProgramarii'])
        ->where('idCabinet',$request['idCabinet'])
        ->get();

    $dayOfWeek = date('w',strtotime($request['dataProgramarii']));

    $disponibilitate = DB::table('working_hours')->where('codZi',$dayOfWeek)
        ->where('codMedic', $request['idMedic'])
        ->where('codCabinet',$request['idCabinet'])
        ->get();
    if(count($disponibilitate) != 0)
    {
        $Consultatie = strtotime($request['oraProgramarii']);
        $consultatieEND = date("H:i", strtotime('+10 minutes', $Consultatie));
        $consultatieEND = strtotime($consultatieEND);
        foreach ($disponibilitate as $dis)
        {
            $startTime = strtotime($dis->startConsultatii);
            $endTime = strtotime($dis->stopConsultatii);
            if(!$startTime < $Consultatie && $consultatieEND < $endTime){
                return redirect('../adaugaProgramare')->with(['errorMessage' => 'Ne pare rau dar programarea dvs. depaseste
programul de lucru al medicului nostru']);
            }
        }
    }else return redirect('../adaugaProgramare')->with(['errorMessage' => 'Ne pare rau dar programarea dvs. nu satisface
programul de lucru al medicului nostru ']);

    if(count($programari) != 0 && $request['idProgramare'] == null) {
        return redirect('/');
    }else if($request['idProgramare'] == null)
    {
        DB::table('programari')->insert([
            'idPatient' => $request['idPatient'],
            'idMedic' => $request['idMedic'],
            'idCabinet' => $request['idCabinet'],
            'dataProgramarii' => $request['dataProgramarii'],
            'oraProgramarii' => $request['oraProgramarii'],
            'tipProgramare' => $request['tipProgramare'],
            'statusProgramare' => $request['statusProgramare'],
        ]);
    }else
    {
        DB::table('programari')->where('id', $request['idProgramare'] )
            ->update([
            'idPatient' => $request['idPatient'],
            'idMedic' => $request['idMedic'],
            'idCabinet' => $request['idCabinet'],
            'dataProgramarii' => $request['dataProgramarii'],
            'oraProgramarii' => $request['oraProgramarii'],
            'statusProgramare' => $request['statusProgramare'],
        ]);
    }

    return redirect('/');
}

```

6.3.46 Emiterea documentelor financiare în raport cu programările - Controller Gestionare Surse de Date pentru a crea o factură fiscală

```

● ● ●

public function generateInvoicePDF($idFactura)
{
    $facturaCurenta = DB::table('facturafiscală')->where('numarFactura', $idFactura)->get();
    $facturaCurenta = $facturaCurenta[0];
    $furnizor = PersoanaJuridica::find(1);
    $client = PersoanaJuridica::find($facturaCurenta->codClient);
    $responsabilFacturare = User::find($facturaCurenta->codResponsabilIntocmire);
    $facturiDetaliate = DB::table('facturidetaliat')->where('numarFactura', $facturaCurenta->numarFactura)->get();

    $date = [
        'furnizor' => $furnizor,
        'client' => $client,
        'facturaCurenta' => $facturaCurenta,
        'facturaDetaliat' => $facturiDetaliate,
        'responsabil' => $responsabilFacturare
    ];

    $pdf = app('dompdf.wrapper');
    $pdf->loadView('invoice.invoicePDF', $date);
    return $pdf->stream('Factura Fiscală'.'.pdf');

}

```

6.3.47 Emiterea documentelor financiare în raport cu programările - Controller Gestionare Surse de Date pentru a crea o factură fiscală

```

● ● ●

public function generateReceiptPDF($idChitanta)
{
    $currentChitanta = DB::table('chitante')->where('numarChitanta', $idChitanta)->get();
    $currentChitanta = $currentChitanta[0];
    $facturaCurenta = DB::table('facturafiscală')->where('numarFactura', $currentChitanta->numarFactura)->get();
    $facturaCurenta = $facturaCurenta[0];
    $furnizor = PersoanaJuridica::find(1);
    $client = PersoanaJuridica::find($facturaCurenta->codClient);

    $date = [
        'furnizor' => $furnizor,
        'client' => $client,
        'factura' => $facturaCurenta,
        'chitanta' => $currentChitanta
    ];

    $pdf = app('dompdf.wrapper');
    $pdf->loadView('receipt.printReceipt', $date);
    return $pdf->stream('Chitanta'.'.pdf');
}

public function viewListaFacturi()
{
    return view('invoice.viewAllInvoices');
}

```

6.3.48 Procedură de adăugare și actualizare a documentelor financiare de tip factură și chitanță în baza de date

```

public function updateFacturaInDB(Request $request)
{
    $validareCampuri = $request->validate([
        'codFirma' => 'required',
        'metodaPlata' => 'required'
    ]);
}

$programare = programari::find($request['codProgramare']);
$adeverinteProgramare = MedicalCertificate::all()->where('idProgramare', $programare->id);
$scrisoriProgramare = ScrisoareMedicala::all()->where('idProgramare', $programare->id);

$todayDate = date("Y-m-d");
$scandeta = date('Y-m-d', strtotime($todayDate. ' + 15 days'));

DB::table('facturafiscală')->insert([
    'codClient' => $request['codFirma'],
    'codResponsabilIntocmire' => $request['codUser'],
    'dataFacturii' => $todayDate,
    'dataScadenta' => $scandeta,
]);

$numarFactura = DB::getPdo()->lastInsertId();
$sumaFactura = 0;
foreach ($adeverinteProgramare as $adeverinta) {
    DB::table('facturidetaliat')->insert([
        'numarFactura' => $numarFactura,
        'DenumireArticol' => 'Prestarii servicii medicale conform programarii #'. $programare->id . '<br>Eliberare Adeverinta
Medicală',
        'UnitateMasura' => 'BUC',
        'Cantitate' => 1,
        'PretUnitarFaraTVA' => $adeverinta->onorariuMedic,
        'cotaTVA' => 19
    ]);
    $sumaFactura = $sumaFactura + ($adeverinta->onorariuMedic * 1.19);
}
foreach ($scrisoriProgramare as $scrisoare) {
    DB::table('facturidetaliat')->insert([
        'numarFactura' => $numarFactura,
        'DenumireArticol' => 'Prestarii servicii medicale conform programarii #'. $programare->id . '<br>Consult Medical de
specialitate',
        'UnitateMasura' => 'BUC',
        'Cantitate' => 1,
        'PretUnitarFaraTVA' => $scrisoare->onorariuMedic,
        'cotaTVA' => 19
    ]);
    $sumaFactura = $sumaFactura + ($scrisoare->onorariuMedic * 1.19);
}

if($validareCampuri['metodaPlata'] == 1)
{
    DB::table('chitante')->insert([
        'dataChitanta' => $todayDate,
        'numarFactura' => $numarFactura,
        'suma' => $sumaFactura,
        'sumaInLiteră' => Chitanta::sumaInLiteră($sumaFactura),
    ]);
}

DB::table('programari')->where('id', $request['codProgramare'])
->update([
    'codDocumentFinanciar' => 'FF' . $numarFactura
]);

return redirect('../Programare/'. $programare->id);
}

```