





**Facultatea de Automatică și Calculatoare Programul de Licență: Informatică**

**LUCRARE DE LICENȚĂ**

**BeeConnected**

**Aplicație web pentru gestionarea apiculturii**

|  |  |
| --- | --- |
| Coordonatori  **Șl.dr.inf. Oana-Sorina CHIRILĂ** | Absolvent  **Gabriela Marcu** |

Timișoara

2024

CUPRINS

[Capitolul 1. Introducere 3](#_TOC_250026)

* 1. [Context 3](#_TOC_250025)
  2. [Tema proiectului 5](#_TOC_250024)

[Capitolul 2. Studiu bibliografic în sistemele medicale din cabinete stomatologice 6](#_TOC_250023)

* + 1. [DENTUNO 6](#_TOC_250022)
    2. [icMED – soft stomatologic 8](#_TOC_250021)
    3. [XDENT 8](#_TOC_250020)
  1. [Concluzii 9](#_TOC_250019)

[Capitolul 3. Bazele teoretice 10](#_TOC_250018)

* 1. [Microsoft Visual Studio 10](#_TOC_250017)
  2. [ASP.Net 10](#_TOC_250016)
  3. [Limbajul C# 13](#_TOC_250015)
  4. [Limbajul SQL 17](#_TOC_250014)
  5. [Bootstrap 19](#_TOC_250013)
  6. [AJAX Control Toolkit 19](#_TOC_250012)

[Capitolul 4. Specificațiile aplicației 22](#_TOC_250011)

* 1. [Descrierea aplicației 22](#_TOC_250010)
  2. [Funcțiile aplicației 23](#_TOC_250009)
  3. [Interfața cu utilizatorul 29](#_TOC_250008)
  4. [Baza de date 58](#_TOC_250007)
  5. [Comunicarea cu alte sisteme 60](#_TOC_250006)
  6. [Tipărirea la imprimantă 60](#_TOC_250005)

Capitolul 5. Proiectarea aplicației de gestionare a pacienților dintr-un cabinet stomatologic .62

* 1. [Implementarea aplicației 62](#_TOC_250004)
  2. [Stocarea informației 74](#_TOC_250003)

[Capitolul 6. Concluzii și perspective 80](#_TOC_250002)

[Bibliografie 81](#_TOC_250001)

[Index figuri 82](#_TOC_250000)

# Capitolul 1. Introducere

## Context

Tehnologia a schimbat modul în care comunicăm, ascultăm muzică, jucăm jocuri și multe altele. Nu este surprinzător că vedem tehnologia cum își extinde funcționalitatea în industria medicală. Sistemele informatice medicale ajută oamenii să trăiască mai mult, reducând timpul de așteptare pentru chestiuni formale și ajută la accelerarea procesului de diagnosticare.

Un sistem informatic medical se referă la un sistem conceput pentru a înregistra, analiza și împărtăși datele despre sănătatea pacienților. Exemple de sisteme informatice medicale: [1]

* + - Registrul medical electronic (EMR) și fișa medicala electronică (EHR) – fișa medicală electronică înlocuiește versiunea sa de hârtie din istoricul medical al pacientului. Registrul medical electronic include mai multe date legate de sănătatea pacientului, rezultatele testelor și tratamentelor;
    - Software pentru managementul programărilor si a facturilor;
    - Indicele pacientului (MPI) – conectează înregistrările separate ale pacienților din bazele de date. MPI-urile sunt utilizate pentru a reduce duplicarea înregistrărilor ale pacienților;
    - Portalurile pentru pacienți – permit pacienților să acceseze datele lor personale de sănătate, cum ar fi informații despre programării, medicamente și rezultatele analizelor medicale de laborator printr-o conexiune la internet. Unele portaluri permit pacienților comunicarea activă cu medicii lor, cereri de prescripții și posibilitatea de a reprograma programare;
    - Monitorizarea pacientului de la distanța (RPM) – monitorizarea la distanță a pacienților permite senzorilor medicali să trimită datele pacienților către instituțiile medicale care îi monitorizează. Cel mai des se monitorizează nivelul glicemiei și tensiunea arterială pentru pacienții cu afecțiuni cronice. Datele sunt utilizare pentru a detecta evenimente medicale care necesită intervenție și pot deveni posibil parte dintr-un studiu mai mare asupra sănătății populației;
    - Sistemele de suport pentru deciziile clinice (CDS) – analizează datele din diferite sisteme clinice și administrative pentru a ajuta personalul medical să ia decizii. Datele pot ajuta la identificarea diagnosticelor sau la prevederea unor evenimente medicale, cum ar fi reacția pacientului la diferite medicamente.

Sistemele informatice medicale tind să vizeze eficiența și gestionarea datelor. Principalii factori sunt: [1]

* + - Analiza datelor – industria medicala produce constant date. Sistemele informatice medicale ajută la colectarea, compilarea și analizarea datelor pentru a ajuta la gestionarea sănătății populației și la reducerea costurilor de asistența medicală;
    - Îngrijire colaborată – pacienți au adesea nevoie de tratamente de la diferite instituții medicale. Sistemele informatice medicale oferă posibilitatea ca instituțiile medicale să aibă acces comun la fișa medicala electronică (EHR) a pacientului;
    - Controlul costurilor – utilizarea rețelelor digitale pentru schimbul de date medicale sporește eficiență și micșorează costurile.
    - Managementul sănătății populației – sistemele informatice medicale pot colecta date despre pacienți, le pot analiza și identifica tendințele populație. Tehnologia funcționează și invers (Sistemele de suport pentru deciziile clinice (CDS)).

Una dintre principalele preocupări ale sistemului informatic medical este securitatea datelor. Toate rețelele sunt vulnerabile, dar sistemele medicale sunt ținte preferate pentru infractorii cibernetici.

Majoritatea sistemelor informatice medicale transmit informațiile între utilizatori pe baza unui proces numit „Ciclul elementar al informației medicale” (figura 1.1) [2].

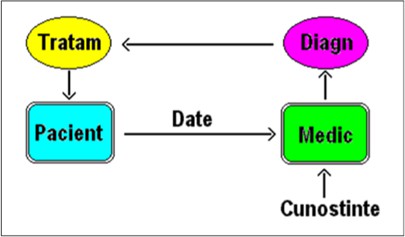


Figura 1.1 Ciclul elementar al informației medicale

Pacientul oferă medicului informații necesare pentru ca medicul pe baza cunoștințelor sale și datelor primite să poate identifica diagnosticul. Pe baza diagnosticului și datelor acumulate pe parcurs medicul poate stabili un tratament, care se potrivește cel mai bine pentru îmbunătățirea stării pacientului. După ce pacientul și-a primit tratamentul, el va fi monitorizat pentru colectarea noilor informații care pot fi de ajutor la continuarea tratamentului sau pentru înlocuirea sa.

## Tema proiectului

Realizarea unui sistem informatic medical pentru gestionarea pacienților dintr-un cabinet stomatologic. Pentru fiecare pacient se va realiza o fișa care va conține datele personale, datele de contact și domiciliul acestuia. Fiecare pacient înainte de a se prezenta la o consultație este nevoit să- își facă o programare. Pentru programare este destul să specifice motivul programări, data, ora și medicul solicitat, iar restul datelor vor fi completate automat datorita faptului că înainte de programare pacientul trebuie să fie înregistrat. În momentul când pacientul se prezintă la o consultație se va realiza o foaie de consultație care va conține pe lângă datele personale ale pacientului și diagnosticele, intervențiile efectuate și costul total. Pacientul va putea achita contravaloarea consultațiilor în aceeași zi sau în rate, caz in care i se va elibera mai multe chitanțe pentru aceeași consultație. Pa lângă cele scrise mai sus sistemul informatic medical mai are și următoarele funcții:

* + - Catalogul cu intervențiile ce se pot realiza în cabinet este disponibil tuturor persoanelor înregistrate și tot online pacienții pot verifica statusul programării solicitate;
    - Recepția se ocupă de realizarea fișei pacientului, tipărirea de chitanțe, confirmarea programărilor și prezența pacienților la cabinet;
    - La orice moment recepția poate obține situații cu pacienții care mai au de achitat consultațiile neachitate;
    - Medici se ocupă de realizare consultațiilor, încărcarea de radiografii și au posibilitatea de a vizualiza fișa pacientului, consultațiile, radiografiile, programările;
    - Sistemul trimite email-uri pentru confirmarea programărilor sau la cererea utilizatorilor informațiile necesare pentru accesarea sistemului informatic.

# Capitolul 2. Studiu bibliografic în sistemele medicale din cabinete stomatologice

Pentru a finaliza cu succes tema de licența am studiat mai multe sisteme informatice medicale produse de anumite companii, precum și din mai multe pagini de internet.

Aplicațiile pe care le-am studiat sunt: DENTUNO, icMED – soft stomatologic, XDENT.

## DENTUNO

Dentuno reprezintă un software stomatologic în cloud care respecta legea GDPR (Regulament General privind Protecția Datelor) a Uniunii Europene. Este o aplicație stomatologica pentru managementul programărilor și datelor pacienților. Unul dintre factorii principali ai Dentuno este protecția datelor despre pacienți:

* + - * Aplicația trimite automat mesaj pacienților ca datele au fost procesate de Dentuno;
      * Oferă o interfața pentru pacient care permite vizualizarea fișierelor, radiografiile proprii și schimbarea datelor personale de contact;
      * Permite descărcarea datelor în format electronic și transferul către alte aplicații;
      * Datele sunt criptate și stocate pe server;
      * Notifică pacientul daca și-a schimbat datele personale din fișa pacientului;
      * Se păstrează istoricul modificărilor efectuate în dosarul medical, cu ajutorul tehnologiei blockchain;
      * În cazul schimbării adresei de e-mail sau a numărului de telefon, pacientul va fi informat de această schimbare.

Blockchain – este o tehnologie avansată care permite distribuirea descentralizată a datelor și are următoarele caracteristici:

* + - * Permite vizualizarea istoricului tuturor modificărilor legate de pacienți;
      * Blockchain este nemodificabil, nu se pot modifica înregistrările din trecut;
      * Comunicarea între mai multe instituții medicale care se folosesc de aceeași fișa a pacientului;
      * Datele sunt criptate, astfel încât nimeni nu le poate accesa fără permisiunea pacientului sau a cabinetului.

Accesul la aplicație se face printr-o pagina de autentificare (figura 2.1.1) [3].

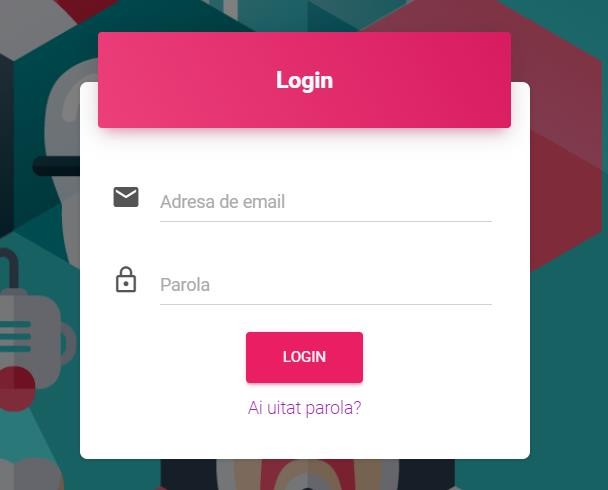


Figura 2.1.1 Pagina de autentificare

Opțiunile din meniu sunt:

* + - * Încasări – plățile ce trebuie încasate pe aceea zi;
      * Programări (figura 2.1.2) [3] – evidența programărilor;
      * Fișe pacienți – vizualizarea, modificarea, arhivarea fișei pacientului;
      * Adaugă pacient – informațiile cerute pentru fiecare persoană sunt: prenume, nume, CNP, data nașterii, telefon, observați asupra numărului de telefon, adresa email, adresa, profesia, numărul fișei, observații medicale și istoric medical;
      * Fișe incomplete – pacienții care nu au fișa completa;
      * Arhiva – fișele pacienților arhivate;
      * Rapoarte – încasările pentru o zi sau lunare, datornici, productivitatea medicului;
      * Unit-uri – alte cabinete, daca medicul este angajat la mai multe cabinete;
      * Proceduri – adăugare și modificare de proceduri;
      * Istoric medical – adăugarea de afecțiuni;
      * Program concediu – perioada de concediu selectată de către medic;
      * Formulare – formulare pe care pacienți trebuie sa le completeze la cabinet;

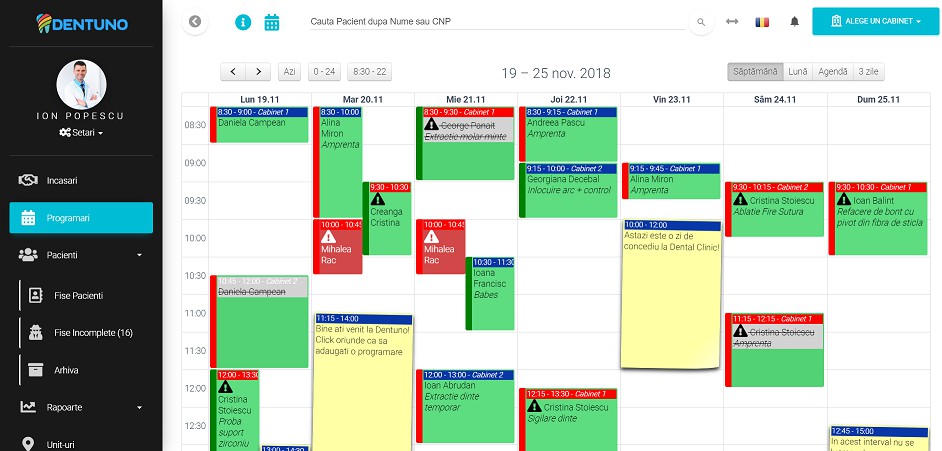


Figura 2.1.2 Evidența programărilor

## icMED – soft stomatologic

Este o aplicație care gestionează pacienții dintr-un cabinet stomatologic. Ea se ocupă de programări, fișe medicale, manoperă și servici și eliberare de documente.

Funcționalitatea aplicației:

* + - * Conform fișei pacientului din sistemul național icMED, medici vor primi avertismente cu privire la riscurile din istoricul medical al pacientului;
      * Fiecare intervenție, manopera și radiografie vor fi stocate în centrul lor de date, securizat și privat, pe care îl pot accesa oricând prin intermediul unui browser de internet de pe un laptop sau tabletă;
      * Aplicația icMed lucrează în contract cu Casa de Asigurări și nu mai e nevoie de accesarea SIUI (Sistemul Informatic Unic Integrat al Asigurărilor Sociale de Sănătate);
      * Oferă posibilitatea programărilor online de pe orice dispozitiv care are acces la browser de internet, avertizează pacienți prin SMS înaintea programării și trimite invitați pentru control periodic.

## XDENT

XDENT este un software pentru gestionarea programărilor dedicat cabinetelor stomatologice. Aplicația are următoarele funcționalități: gestionarea fișei medicale a pacientului, programarea pacienților, evidența cheltuieli, generări acorduri pacient, rapoarte de activitate, evidența încasări si plați.

Caracteristici:

* + - * Aplicația oferă panou de administrare pentru medici, recepționeri, manageri și interfață pentru utilizatori;
      * Calendarul poate fi integrat în site-ul web pentru a fi mai ușor accesibil pacienților și utilizatorilor;
      * Pentru clienți fideli aplicația oferă vouchere, reducerii si avantaje;
      * Pacienții au posibilitatea de a se înregistra pentru a face programări sau pentru a modifica programările existente;
      * Aplicația permite gestionarea unui număr nelimitat de cabinete, medici stomatologi, specializări, departamente [4];
      * Sistem de gestionare a încasărilor, emiterea de bonuri fiscale;
      * Afișare interactivă a calendarului unde este prezentat planul de lucru pe viitor, în mod general sau la nivelul fiecărui medic;
      * Notificarea pacienților prin SMS despre programările viitoare, oferte, reduceri;
      * Generarea si tipărirea documentelor pentru obținerea acordului clienților cu privire la anumite proceduri;

## Concluzii

În urma analizări sistemelor prezentate mai sus am observat că fiecare sistem încearcă sa ofere cat mai multe facilitați pentru a ușura munca cabinetelor medicale. În continuare voi prezenta un tabel comparativ care conține unele caracteristici ale fiecărei aplicați studiate:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Caracteristici** | **DENTUNO** | **icMED** | **XDENT** | **Denis - Dent** |
| Aplicația accesibila de oriunde prin intermediul unui browser de internet de pe un laptop sau tabletă | X | X | X | X |
| Interfețe pentru mai multe tipuri de utilizatori | X | X | X | X |
| Vizualizarea istoricului tuturor modificărilor legate de pacienții | X |  |  |  |
| Posibilitatea programărilor online |  | X | X | X |
| Notificării prin SMS, email legate de programări |  | X | X | X |
| Anunțare venire pacient în sala de așteptare |  |  |  | X |
| Transferul datelor către alte aplicați | X | X |  |  |
| Posibilitatea de tipărire a chitanțelor |  |  | X | X |
| Posibilitatea de tipărire a fișei pacientului, consultațiilor | X | X |  | X |
| Catalogul intervențiilor (serviciilor) disponibil tuturor persoanelor înregistrate |  | X |  | X |
| Validarea CNP-ului pacienților |  | X |  | X |

# Capitolul 3. Bazele teoretice

Pentru realizarea sistemului informatic de gestionarea a pacienților dintr-un cabinet stomatologic am folosit mediul de dezvoltare Microsoft Visual Studio. Acesta include un set complet de instrumente de dezvoltare pentru crearea de aplicații ASP.NET, Servicii Web XML, aplicații desktop și aplicați mobile [5].

## Microsoft Visual Studio

Visual Studio folosește platforme de dezvoltare Microsoft, cum ar fi Windows API, Windows Forms, Windows Presentation Foundation, Windows Store și Microsoft Silverlight. Include un editor de cod care acceptă IntelliSense (autocompletarea codului scris de programator) precum și refactorizarea codului. Depanatorul integrat funcționează și ca depanator la nivel de cod sursă și ca depanator la nivel de cod mașina. Alte instrumente încorporate includ un profilator de cod (exemplu: de câte ori este apelată fiecare metoda din codul meu? și cât durează fiecare dintre aceste metode?), designer pentru dezvoltarea aplicațiilor cu interfața grafică, designer web, designer de clase și proiectare de scheme de baze de date. La fel acceptă plug-in-uri care extind funcționalitatea la aproape toate nivelurile și diferite unelte ce vă pot face viața mai ușoara (exemplu în cazul meu AJAX Control Toolkit despre care o să vă prezint mai târziu).

Visual Studio acceptă 36 de limbaje de programare diferite, unele dintre ele fiind următoarele: C, C++, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML și CSS [5]. În trecut Visual Studio oferea suport și pentru Java și J# [5].

## ASP.Net

ASP.Net este o tehnologie Microsoft lansat în anul 2002, pentru dezvoltarea aplicațiilor web și serviciilor web care beneficiază de puterea platformei de dezvoltare .NET. ASP.Net poate rula pe platforme diferite precum Windows, Linux, macOS și Docker [6].

.NET este o platformă pentru dezvoltarea de diferite aplicații, formată din diferite unelte, limbaje de programare și biblioteci.

Câteva lucruri pe care ASP.Net le adaugă platformei .NET:

* ASP.Net reprezintă baza pentru procesarea cererilor web în C# sau F#;
* Sintaxa de șablonare a paginilor web, cunoscută sub numele de Razor, folosită pentru construirea paginilor web dinamice în C#;
* Biblioteci pentru Model View Controller (MVC);
* Sistem de autentificare care include biblioteci, bază de date și șabloane pentru pagini web de autentificare, inclusiv autentificarea cu mai mulți factori și autentificarea externă cu Google, Twitter și altele;
* Extensii pentru editor pentru a oferi evidențierea sintaxei, autocompletarea codului și alte funcționalități specifice dezvoltării paginilor web.

Caracteristici ASP.Net:

* Separă codul de design. Prin această separare, dezvoltarea aplicației ASP.Net devine mult mai ușoara. Tipul fișierelor ASP.Net este aspx. De exemplu dacă avem o pagină web numită VizualizareRadiografii.aspx pe lângă ea va mai exista și un alt fișier VizualizareRadiografii.aspx.cs care va conține partea de cod (C#) din spatele pagini. Așadar, Visual Studio creează fișiere separate pentru fiecare pagină web, unul pentru partea de design (front-end) și celălalt pentru cod (back-end);
* ASP.Net are facilitatea de a controla utilizarea informațiilor de stare (informațiile cererilor vechi). La HTTP se pierd informațiile cererilor vechi [7];
* ASP.Net poate implementa conceptul de Caching. Acest lucru îmbunătățește performanța aplicației. Paginile care sunt folosite des de utilizator pot fi stocate în memoria cache. Aceste pagini pot fi încărcate mai rapid și pot răspunde mai rapid comenzilor utilizatorului. Deci cache-ul poate îmbunătăți semnificativ performanța unei aplicații;
* ASP.Net este proiectat să funcționeze cu protocolul HTTP.

Atunci când o aplicație ASP.Net este lansată, există o serie de etape care sunt efectuate. Aceste etape alcătuiesc ciclul de viață al aplicației.

* + 1. Pornirea aplicației – ciclul de viața al aplicației ASP.Net începe atunci când o solicitare este făcuta de către utilizator. Această solicitare este adresată serverului Web pentru aplicația ASP.Net după care serverul Web va executa o metodă numită Application\_start. De obicei, în această metodă, toate variabilele globale sunt setate la valorile lor implicite.
    2. Crearea obiectelor – următoarea etapă este crearea obiectelor de tipul HttpContext, HttpRequest și HttpResponse de către serverul Web. HttpContext reprezintă doar containerul pentru obiectele HttpRequest și HttpResponse. Obiectul de tipul HttpRequest conține informații despre solicitarea curentă, inclusiv cookie-uri și informații despre browser. Obiectul de tipul HttpResponse conține răspunsul care este transmis utilizatorului.
    3. Crearea aplicației HTTP – obiectul de tipul HttpApplication este creat de serverul Web. Acest obiect este folosit pentru a procesa fiecare solicitare ulterioară care va fi trimisă aplicației.
    4. Dispose (eliberarea memoriei) – această metodă este apelată înainte de distrugerea instanței aplicației. Ea se poate folosi manual pentru a elibera orice resursă de memorie de care nu mai este nevoie.
    5. Încheierea aplicației – aceasta este ultima etapă a aplicației, etapa în care aplicația nu mai dispune de resurse de memorie.

La fel, când se apelează o pagină ASP.Net, ea trece printr-un anumit ciclu de viața. Aceasta se face înainte ca răspunsul să fie trimis utilizatorului. Există o serie de pași care sunt urmați pentru procesarea unei pagini ASP.Net.

1. Solicitarea paginii – această etapa se întâmplă atunci când pagina este solicitată pentru prima dată de la server. Când pagina este solicitată, serverul verifică dacă este solicitată pentru prima dată sau nu. Dacă pagina este solicitată pentru prima dată, atunci serverul Web trebuie să compileze pagina, pregătească răspunsul și să o trimită utilizatorului. Dacă nu este prima dată când se solicită pagina, se va verifica daca in cache deja pagina există și dacă există ea va fi trimisă ca răspuns utilizatorului.
2. Pornirea paginii – în această etapă două obiecte sunt create, sub numele de Request (Solicitare, Cerere) și Response (Răspuns). Obiectul Request este folosit pentru a reține toate informațiile care au fost trimise la solicitarea paginii. Obiectul Response este utilizat pentru a reține informațiile care sunt trimise înapoi utilizatorului.
3. Inițializarea paginii – în această etapă toate controalele care aparțin de pagina web sunt inițializate. Deci, aici se inițializează etichetele (Labels), casetele text (TextBoxes) sau orice alte controale de pe paginile web.
4. Încărcarea pagini – în această etapa pagina este încărcată efectiv cu toate valorile implicite. De exemplu, când se întocmește o consultație, data consultări va avea valoarea zilei în care consultația a fost efectuată și atunci caseta de text (Text Box) pentru dată va avea valoare implicită, care la rândul ei va fi încărcată în timpul încărcării paginii.
5. Validare – uneori, pot exista seturi de validare pe paginile web. De exemplu, poate exista o validare care spune că o casetă de listă ar trebui să aibă un anumit set de valori. Dacă condiția nu este îndeplinită, ar trebui să existe o eroare la încărcarea pagini.
6. Gestionarea evenimentelor de tip Post Back – acest eveniment este declanșat dacă aceeași pagină este din nou încărcată. Acest lucru se întâmplă ca răspuns la un eveniment anterior. De exemplu, când utilizatorul face clic pe un buton de trimitere din pagină.
7. Redarea paginii – această etapă se desfășoară chiar înainte ca toate informațiile de răspuns să fie transmise utilizatorului. Toate informațiile din formular (<form>) sunt salvate și rezultatul este trimis utilizatorului ca o pagină web completă.
8. Eliberarea memoriei – odată ce rezultatul paginii este trimis utilizatorului, nu mai este necesar păstrarea în memorie a obiectelor din formularul web ASP.Net, ca atare ele vor fi eliminate din memorie.

## Limbajul C#

Limbajul C# este un limbaj de programare dezvoltat de Microsoft prin ani 2000 - 2001, care rulează pe platforma .NET. C# este utilizat pe scară largă pentru dezvoltarea de aplicații mobile, jocuri, aplicații Windows, aplicații ASP.Net, etc. El este un limbaj de programare de nivel înalt, orientat pe obiecte. Caracteristicile principale ale limbajului C# sunt:

* Modern și ușor – scopul C#-ului a fost dezvoltarea unui limbaj de programare care să nu fie numai ușor de învățat, dar să și suporte funcționalitatea modernă pentru toate tipurile de dezvoltare software. Limbajul C# a fost proiectat pentru a ține cont de nevoile afacerilor și întreprinderilor. El a fost dezvoltat pentru întreprinderi, pentru crearea a tot felul de software- urii utilizând un singur limbaj de programare. Unele dintre caracteristicile moderne ale limbajului de programare C# sunt: tipuri var, inițializarea automată a tipurilor și colecțiilor, expresii lambda, programare dinamică, programare asincronă, depanare avansată, tratare de excepții, etc. Sintaxa limbajului C# este influențată de C++, Java, Pascal și altele. C# evită complexitatea și caracteristicile limbajului nestructurat.
* Rapid și „open source” - C# este „open source” în cadrul .NET Foundation, care este condusă și rulată independent de Microsoft. Specificațiile limbajului C#, compilatoare și diferite unelte le puteți găsi pe Github. În timp ce designul caracteristicilor de limbaj C# este condus de Microsoft, comunitatea „open source” este foarte activă în dezvoltarea și îmbunătățirea limbajului. Limbajul C#, comparativ cu alte limbaje de nivel înalt este mai rapid.
* Poate rula pe platforme diferite – putem crea aplicații .NET care pot fi implementate pe platformele Windows, Linux și Mac. Aplicațiile C# de asemenea pot fi implementate și in cloud.
* Sigur și eficient – C# nu permite conversii de tip care pod duce la pierderi de date sau alte probleme.
* Poate crea aplicații de tip diferit – C# poate fi utilizat pentru a dezvolta tot felul de aplicații, aplicații client Windows, componente și biblioteci, servicii și API-uri, aplicații Web, aplicații mobile, aplicații cloud și jocuri video.
* C# tot mai evoluează – C# 8.0 este ultima versiune de C#. Dacă ne uităm la istoricul limbajului de programare C#, C# evoluează mai repede decât orice alte limbaje. C# inițial a fost dezvoltat

pentru a crea aplicații client Windows, dar astăzi, C# poate face apropare orice, de la aplicații în consolă, aplicații cloud pană la softuri de învățare automată.

### Controale

Controalele sunt obiecte care au componente grafice și ele interacționează cu utilizatorul pe o pagină web. Utilizând controale, utilizatorii pot introduce date, pot face selecții și pot indica preferințele utilizatorului. Controalele sunt, de asemenea, utilizate pentru validare, accesul la date, securitate, crearea de pagini master și manipularea datelor.

ASP.Net folosește cinci tipuri de controale web:

* Controale HTML;
* Controale Server HTML;
* Controale Server ASP.Net;
* Controale Ajax Server ASP.Net;
* Controale utilizator și alte controale personalizate.

Controalele serverului ASP.Net sunt cele mai utilizate în ASP.Net. Ele pot fi grupate în următoarele categorii:

* Controale de validare – acestea sunt utilizate pentru validarea datelor introduse de utilizator și funcționează rulând script-uri din partea clientului;
* Controale sursă de date – acestea controale asigură legătura la diferite surse de date;
* Controale vizualizare date – acestea sunt diverse liste și tabele, care pot face legătura la date din surse diferite pentru a fi afișate;
* Controale de personalizare – acestea sunt utilizate pentru personalizarea pagini în funcție de preferințele utilizatorului, pe baza informațiilor acumulate de la utilizator;
* Controale de conectare și securitate – aceste controale sunt folosite pentru autentificarea utilizatorului;
* Pagini master – aceste controale asigură interfața consistentă în întreaga aplicație;
* Controale de navigare – ajută la navigare;
* Controale bogate (Rich controls) – aceste controale implementează funcții speciale. De exemplu, AdRorator, FileUpload și Calendar.

Controalele care au fost folosite cel mai mult în realizarea sistemului informatic pentru gestiunea pacienților dintr-un cabinet stomatologic sunt următoarele:

* Label (eticheta), TextBox (casetă text) – aceste controale sunt folosite pentru a oferii informații utilizatorului sau pentru a colecta informații de la utilizator (casetele text). Label în principal se utilizează pentru a crea o legendă pentru celelalte controale, cum ar fi de exemplu pentru TextBox. Pentru a crea un Label sau un TextBox putem scrie cod sau putem folosi funcția de „drag and drop” din Visual Studio 2017 (valabil pentru toate controalele);
* Panel (panou) – aceste controale sunt folosite pentru organizarea informației de pe pagina web. De asemenea, permite generarea de controale dinamic;
* CheckBox (caseta de bifat) – permite selectarea de opțiuni;
* Button (buton), ImageButton (imagine care are aceeași funcție ca și butonul simplu) – aceste controale controlează aplicația. Button poate vi un buton de trimitere (submit) sau un buton de comandă. În mod implicit, acest control este un buton de trimitere. Butonul de trimitere nu are un nume de comandă și trimite cerere spre server atunci când este dat clic pe el. Este posibil crearea de un gestionar de evenimente pentru a controla acțiunile efectuate la apăsarea butonului.
* FileUpload (încărcarea de fișiere) – acest control interacționează cu obiecte din afara aplicației. Ne permite să încărcăm fișiere pe un server Web sau stocarea într-un formular Web;
* HyperLink (hiperlink sau link) – acest control reprezintă o referință, legătură sau element de navigare într-o aplicație. Răspunde la un eveniment de clic. Îl putem folosi pentru a trimite utilizatorul la orice pagină web de pe server;
* DropDownList (lisată de selecție) – acest control permite selectarea de opțiuni dintr-o listă derulantă și ea poate conține oricâte opțiuni, nu are limită.
* GridView – acest control are mai multe funcționalități, el poate accepta, afișa și edita date pe o pagină web. Fiecare coloană din GridView reprezintă un câmp, în timp ce fiecare rând reprezintă o înregistrare.

### Lucru cu fișiere

Un fișier este o colecție de date stocate pe un disc cu un nume specific și o cale de director. Când un fișier este deschis pentru citire sau scriere, acesta devine un flux de date. Fluxul de date este practic secvența de octeți care trece prin canalul de comunicare. Există două fluxuri de date principale: fluxul de intrare și fluxul de ieșire. Fluxul de intrare este utilizat pentru citirea datelor din fișier (operația de citire), iar fluxul de ieșire este utilizat pentru scrierea în fișier (operația de scriere).

Spațiul de nume System.IO are diverse clase care sunt utilizare pentru a efectua numeroase operații cu fișiere, precum crearea și ștergerea fișierelor, citirea sau scrierea unui fișier, închiderea de fișier,

etc. Clasele utilizate cel mai frecvent din spațiul de nume System.IO sunt următoarele: FileStream, StreamReader, StreamWriter, Directory, DirectoryInfo, File, BinaryReader, BinaryWriter, Path, etc.

### Tratarea excepțiilor în C#

Tratarea excepțiilor în C# ne ajută să facem față oricăror situații neașteptate sau excepționale care apar atunci când un program este executat, rulat. Pentru tratarea excepțiilor folosim cuvintele cheie try, catch și finally. Excepțiile pot fi generate în timpul execuției, de bibliotecile .NET sau altele sau excepții care apar din codul aplicației. Excepțiile se creează cu ajutorul cuvântului cheie throw.

În multe cazuri, excepția poate fi aruncată nu numai de o metodă pe care codul aplicației o apelează, ci și de o altă metodă care se află mai jos în stiva de apeluri. Când se întâmplă acest lucru, CLR (Common Language Runtime) va goli stiva, căutând o metodă cu un bloc de catch pentru tipul de excepție specific și va executa primul bloc catch corespunzător pe care îl găsește. Dacă nu găsește nici un bloc catch adecvat în stiva de apeluri, CLR va încheia aplicația (procesul) și va afișa un mesaj utilizatorului.

### Blocul finally

În programare, uneori o excepție poate provoca o eroare care pune capăt metodei curente. Totuși, acea metodă ar fi putut deschide un fișier sau o rețea care trebuie închisă. Așadar, pentru a depăși astfel de tipuri de probleme, C# conține un cuvânt cheie special numit finally. Blocul finally se va executa atunci când blocul try/catch părăsește execuția, indiferent de starea care o determină. Se execută întotdeauna dacă blocul try se încheie normal sau se încheie din cauza unei excepții.

Scopul principal al blocului finally este eliberarea resurselor din memorie. Blocul finally se află întotdeauna după blocul try/catch.

### Instrucțiunea using

C# și .NET asigură gestionarea resurselor pentru obiecte prin intermediul lui „garbage collector” – memoria nu trebuie alocată și eliberată explicit. Pentru a permite programatorului să efectueze în mod explicit operațiile de curățare, obiectul poate furniza o metodă numită Dispose care poate fi apelată atunci când obiectul nu mai este necesar.

Instrucțiunea using din C# definește o limită pentru obiect în afara căruia, obiectul este distrus automat. Din instrucțiunea using se iasă atunci când se ajunge la sfârșitul blocului using sau când se aruncă o excepție.

## Limbajul SQL

SQL (Structured Query Language) se folosește pentru a comunica cu baza de date. Instrucțiunile SQL sunt utilizate pentru a efectua sarcini precum actualizarea datelor dintr-o bază de date sau pentru a prelua date dintr-o bază de date. Unele sistem comune de gestionare a bazelor de date relaționale care folosesc SQL sunt: Microsoft SQL Server, Oracle, etc. Deși majoritatea sistemelor de baze de date utilizează SQL, cele mai multe dintre ele au, de asemenea, propriile lor modificări care sunt de obicei utilizate doar în sistemul lor. Comenzile standard SQL, cum ar fi Select, Insert, Update, Delete pot fi utilizare pentru a realiza aproape tot ce e nevoie pentru a lucra cu o bază de date.

*Select* – instrucțiunea select este utilizată pentru interogarea bazei de date și pentru a prelua datele selectate care corespund criteriilor stabilite de noi. Forma unei declarații simple de selectare:

*select Judet from Judete order by Judet;*

order by Judet – utilizat pentru a sorta rezultatul (județele) în ordine crescătoare, pentru ordinea descrescătoare trebuie adăugat cuvântul cheie DESC (order by Judet DESC).

Numele coloanelor ce se află după cuvântul cheie select, determină ce coloane vor fi returnate în rezultat. Putem selecta câte numiri de colane dorim sau putem folosi \* (Asterix) pentru a selecta toate colanele. Numele tabelei care se află după cuvântul cheie from specifică tabelul cate va fi solicitat pentru a prelua rezultatele dorite.

*Insert* – instrucțiunea insert este utilizată pentru a insera (adăuga) o înregistrare nouă în baza de date.

*insert into ContPacienti ([CNP], [Nume], [Prenume], [Sex], [DataNasterii], [Telefon], [Email], [Parola]) values (‘1234567891234’, ‘Avram’, ‘Denis’, ‘Masculin’, ‘06/20/2020’, ‘0712312311’,* [*‘test@mail.ocm*](mailto:test@mail.ocm)*’ ‘test’);*

Numele tabelei care se află după cuvântul cheie into specifică tabelul în care se vor insera datele. Numele coloanelor care se află între parantezele rotunde, după numele tabelei, reprezint coloanele care vor fi populate cu date (parantezele drepte [ ] sunt opționale). După cuvântul cheie values

urmează intre parantezele rotunde valorile care vor fi inserate (apostrofele le folosim pentru a insera șiruri de date, iar pentru valori numerice nu avem nevoie de ele).

*Update* – instrucțiunea update este utilizată pentru a actualiza sau schimba înregistrările care corespund criteriilor specificate. Acest lucru se realizează prin folosirea cuvântului cheie where.

*update Pacienti set Prenume=’Todor’ where CNP=’1234567891234’;* Pacientii – numele tabelului unde va fi efectuată modificare (actualizarea); Prenume – colana care va fi modificată (actualizată);

CNP – criteriul care trebuie îndeplinit pentru ca modificarea (actualizarea) să aibă loc.

Cuvântul cheie where specifică ce valori sau rânduri de date vor fi modificate (actualizate) pe baza criteriilor descrise după where.

*Delete* – instrucțiunea delete este utilizată pentru a șterge înregistrări sau rânduri din tabel.

*delte from NumeTabel where NumeColoana operator valoare;* NumeTabel – reprezintă numele tabelului unde va avea loc ștergerea; NumeColana – reprezintă coloana care va fi afectată;

operator - =, >, <, >=, <=;

valoare – valoarea criteriului care trebuie să fie îndeplinită, pentru ca ștergerea să aibă succes.

### ADO.NET

ADO.NET (ADO – ActiveX Data Object) este o parte importantă a platformei .NET care unifică accesul la baze de date, fișiere XML și alte date de aplicație. MySql.Data este o implementare a ADO.NET pentru baza de date MySQL. Este un driver scris în limbajul C# și este disponibil pentru toate limbajele de programare .NET.

MySqlConnection, MySqlCommand, MySqlDataReader, DataSet și MySqlDataProvired sunt elementele de bază ale modelului de furnizor de date .NET. MySqlConnection creează conexiunea la baza de date. Obiectul de tipul MySqlCommand execută o instrucțiune SQL asupra unei baze de date. MySqlDataReader citește fluxul de date dintr-o bază de date.

Obiectul de tip DataSet este utilizat pentru lucru cu baza de date care conține foarte multe date. DataSet este populat de SqlDataAdapter. Atât MySqlDataReader, cât și DataSet sunt utilizate pentru lucru cu date, ele sunt utilizate în situații diferite. Dacă trebuie doar să citim rezultatele unei interogări, MySqlDataReader este alegerea mai bună. Dacă avem nevoie de o prelucrare mai complexă a datelor sau dorim să legăm un control (GridView) ASP.Net la o tabelă a bazei de date, DataSet este opțiunea mai bună.

## Bootstrap

Bootstrap este o colecție uriașă de chestii folositoare, cod reutilizabil scris în HTML, CSS și JavaScript. Este o platformă de dezvoltare front-end, care permite dezvoltatorilor și proiectanților să creeze rapid pagini web responsive.

Bootstrap-ul poate fi redus la trei fișiere principale:

* bootstrap.css – partea CSS;
* bootstrap.js – partea JavaScript/jQuery;
* glyphicons – font.

Bootstrap necesită jQuery pentru a funcționa. jQuery este o bibliotecă JavaScript extrem de populară și utilizată pe scară largă, care simplifică si adaugă compatibilitatea browser-ului cu JavaScript.

Scopul principal al adăugării la o aplicație web este de a aplica opțiunile Bootstrap de culoare, dimensiune, font și aspect la acel proiect. Odată adăugat la un proiect, Bootstrap oferă definiții de bază ale stilului (style) pentru toate elementele HTML. Rezultatul este un aspect uniform pentru scris, tabele și controale care apar în browser-ul web. În plus, dezvoltatorii pot beneficia de clasele CSS care sunt deja definite în Bootstrap pentru a personaliza în continuare aspectul conținutului de pe pagina web.

Bootstrap vine cu mai multe componente JavaScript sub formă de plugin-uri jQuery care oferă elemente suplimentare pentru interfața cu utilizatorul, cum ar fi casetele de dialog, „tooltips” și carusele.

## AJAX Control Toolkit

Ajax Control Toolkit este o bibliotecă „open source” pentru dezvoltarea aplicațiilor web. El conține controale web extrem de bogate pentru crearea de aplicații web responsive și interactive. Are peste

40 de controale care sunt ușor de utilizat pentru a spori productivitatea. Controalele le putem găsi în caseta de instrumente Visual Studio, iar pentru a le integra în aplicația noastră putem folosi „drag and drop”. Unele dintre controale pe care le-am folosit și eu sunt: CalendarExtender, CollapsiblePanelExtende și ModalPopupExtender.

### CalendarExtender

CalendarExtender (figura 3.6.1) este un extensor ASP.Net care poate fi atașat la orice control TextBox ASP.Net. Oferă funcționalitatea de selectare a datelor pe partea clientrului cu format de date personalizabil și interfață de utilizator într-un control pop-up. Putem interacționa cu calendarul facem clic pe o zi pentru a seta data sau link-ul „Today” pentru a seta data curentă. Săgețile din stânga și din dreapta pot fi folosite pentru a merge înainte sau înapoi o lună. Făcând clic pe titlu calendarului puteți schimba vizualizarea de la zilele din luna curentă, la luna din anul curent. Dacă mai dam o dată clic tot acolo ne va trece la anii din deceniul curent. Această acțiune de permite să sărim cu ușurința la datele din trecut sau viitor din interiorul controlului.

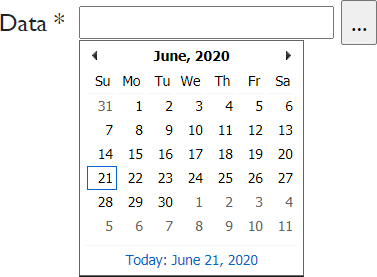


Figura 3.6.1 CalendarExtender

### CollaspiblePanelExtender

CollapsiblePanelExtender (figura 3.6.2) este un extensor foarte flexibil care ne permite să adăugăm cu ușurință secțiuni pliabile pe pagina web. Programatorul specifică ce control din pagina ar trebui să fie controlerul deschi/închis pentru panou, sau să se deschidă automat atunci când cursorul se deplasează în sau din acesta. Panoul este, de asemenea, conștient de evenimentul „post-back”.





Figura 3.6.2 CollapsiblePanelExtender

### ModalPopupExtender

ModalPopupExtender (figura 3.6.3) este un extensor care permite unei pagini să afișeze conținutul pagini într-o manieră „modală” care împiedică utilizatorul să interacționeze cu restul pagini. Conținutul modal poate avea orice controale și este afișat deasupra unui fundal care poate avea un stil personalizat. Când este afișat, numai conținutul modal poate fi interacționat, iar clic pe restul pagini nu va face nimic. Când utilizatorul finalizează interacțiunea cu conținutul modal, un clic al unui control (buton) OK/Cancel/Inchide anulează conținutul modal.

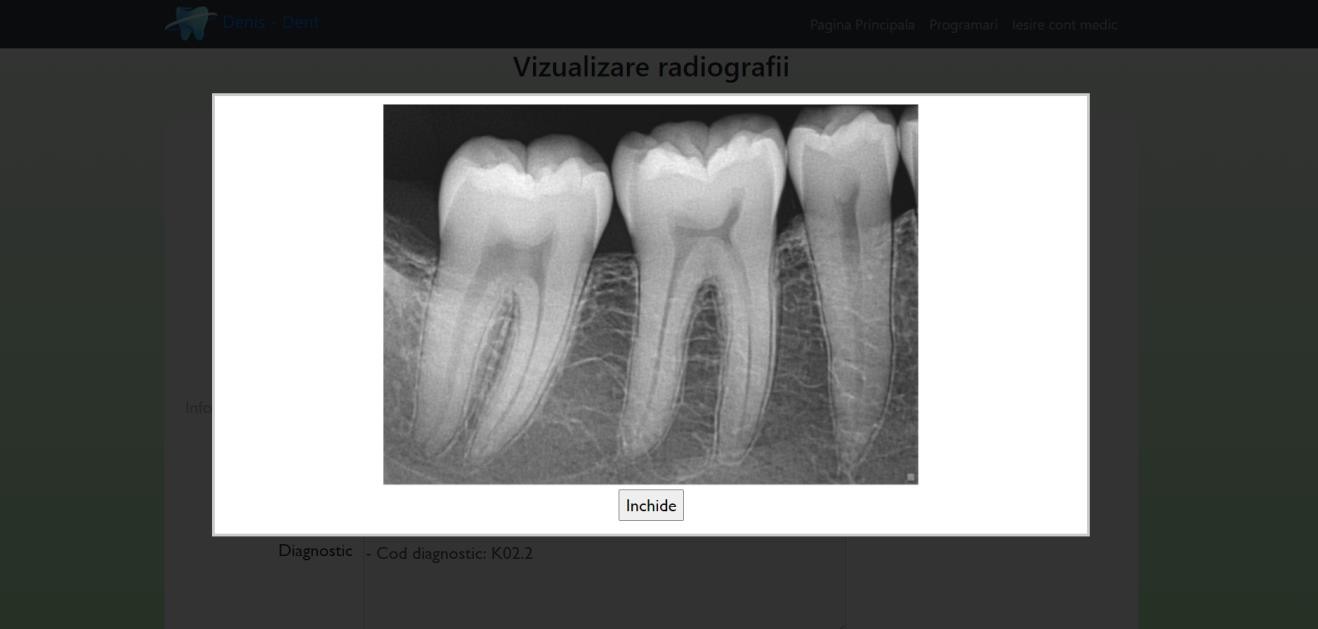
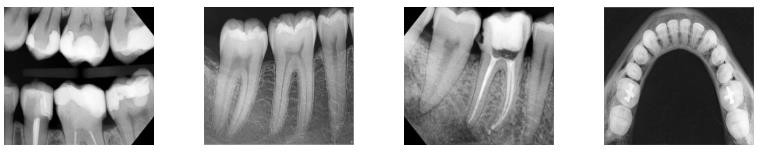


Figura 3.6.3 ModalPopupExtender

# Capitolul 4. Specificațiile aplicației

## Descrierea aplicației

Solicitarea/vizualizare stare programare

Utilizator

(pacient, recepție, medic)

Tastatura

Gestionarea pacienților în cabinetul stomatologic

Figura 4.1 Schema bloc a sistemului

Efectuarea consultațiilor

Periferic de ieșire

- Imprimantă

Periferic de ieșire

- Monitor

Unității auxiliare de memorie

Aplicația reprezintă un sistem informatic care este personalizat pentru trei tipuri de utilizatori, pacient, recepție, medic. Pacientul trebuie să își facă singur cont pentru a putea accesa sistemul (partea care ține de pacient), iar pentru recepție și medici conturile sunt deja pregătite.

După autentificare:

* Pacientul are acces la catalogul de intervenții, solicitarea programări și verificarea stări programărilor solicitate (confirmate, neconfirmate, expirate);
* Recepția are acces la gestionarea pacienților din cabinetul stomatologic;
* Medicul are acces la tot ce ține de efectuarea consultațiilor într-un cabinet stomatologic.

Recepția are posibilitatea de a vizualiza lista pacienților din cabinet și la selectarea unui anumit pacient din tabel (GridView), se vor activa și mai multe posibilități precum, vizualizarea/editarea fișei

pacientului, tipărirea de chitanțe pentru consultațiile ce iau mai rămas de achitat și anunțarea medicului că pacientul se află în sala de așteptare. Tot recepția se ocupă și de gestiunea programărilor solicitate din partea pacienților, adăugarea de pacienți noi și evidențierea consultațiilor neachitate.

Medicul are posibilitatea de a vizualiza informații despre pacientul care este prezent în sala de așteptare. Poate vizualiza fișa pacientului, consultațiile precedente și radiografiile pacientului. În momentul când pacientul este prezent la medic, medicul mai are și posibilitatea de a adăuga consultații noi și radiografii pe care medicul lea solicitat.

Toate aceste informații vor fi stocate într-o bază de date și clasificate în tabelele corespunzătoare.

## Funcțiile aplicației

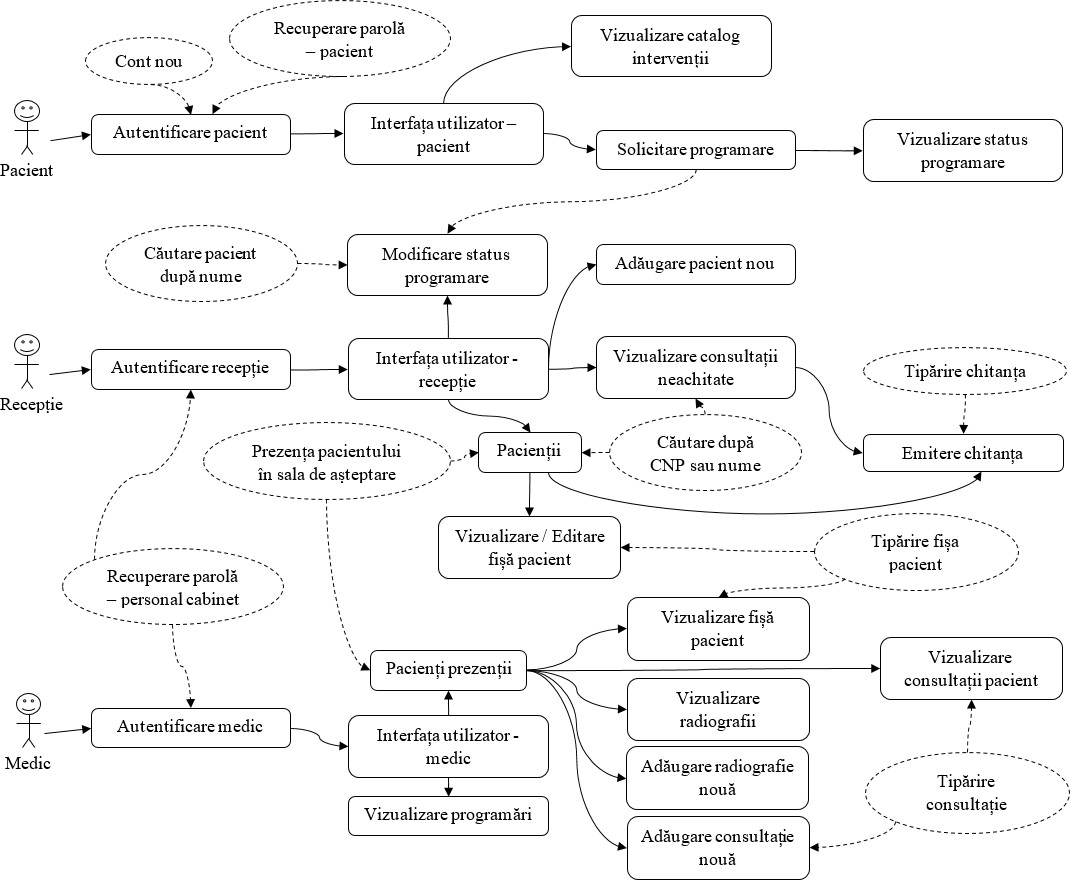


Figura 4.2.1 Funcțiile aplicației (schema bloc)

Cum aplicația este personalizată pentru trei tipuri de utilizatori, pentru fiecare tip avem diferite funcții care corespund utilizatorului.

Pentru pacient:

* Înregistrare;
* Autentificare;
* Uitat parola;
* Afișarea catalogului de intervenții;
* Solicitarea programări;
* Verificarea statusului programărilor solicitate (confirmate, neconfirmate, expirate); Pentru recepție:
* Autentificare;
* Uitat parola;
* Afișarea tuturor pacienților din cabinet;
* Căutare pacient după CNP sau nume;
* Selectarea pacientului din tabel;
* Anunțarea medicului că pacientul se află în sala de așteptare;
* Adăugare pacient nou;
* Vizualizare/editare fișă pacient;
* Vizualizare consultații neachitate;
* Căutarea consultațiilor neachitate după CNP sau numele pacientului;
* Selectarea consultației neachitate din tabel;
* Tipărire chitanța;
* Tipărire fișă pacient;
* Afișarea tuturor programărilor solicitate;
* Selectarea programări;
* Modificarea statusului programări (confirmată, neconfirmată, expirată) și informarea pacientului prin adresa email când programare este confirmată;
* Afișarea programărilor în funcție de statusul programării ales;
* Căutarea programărilor după numele pacientului; Pentru medic:
* Autentificare;
* Uitat parola;
* Afișarea pacienților care sunt prezenți în sala de așteptare;
* Selectarea pacientului din tabel;
* Adăugare consultație nouă;
* Adăugare radiografie nouă;
* Vizualizare fișă pacient;
* Vizualizare consultații pacient;
* Vizualizare radiografii;
* Tipărire fișă pacient;
* Tipărire consultație;
* Afișare programări pentru ziua curentă, săptămâna curentă sau pentru o dată selectată manual de către medic.

Funcțiile precum **autentificare** și **uitat parola** au aceeași funcționalitate pentru toate cele 3 tipuri de utilizator, diferența fiind informațiile de care au nevoie pentru ași îndeplini sarcina. Autentificarea este folosită pentru a primi acces la aplicație, iar funcția uitat parola ajută utilizatorul să își recupereze parola uitată.

### Pacient

**Înregistrare** – permite înregistrarea utilizatorilor noi (pacienți) care doresc să utilizeze aplicația. Înregistrarea se face pe baza datelor personale și parola care va fi ulterior folosită pentru autentificare, împreună cu CNP-ul utilizatorului. După înregistrare, utilizatorul (pacientul) primește acces pentru vizualizarea catalogului de intervenții, solicitarea de programării și verificarea stări în care se află programarea (confirmată, neconfirmată, expirată).

**Afișarea catalogului de intervenții** – întregul catalog de intervenții este stocat în baza de date și în momentul accesării pagini catalogul va fii încărcat automat în mai multe GirdView-uri pentru a fi afișat. La încărcarea pagini doar primul catalog este deschis, iar restul sunt închise și pentru a le deschide este de ajuns să dați un clic pe numele catalogului (CollapsiblePanelExtender).

**Solicitarea programări** – utilizatorul (pacientul) poate solicita programarea online. Pentru a solicita o programare este de ajuns să selecteze medicul dorit, motivul programări și să aleagă perioada dorita (data, ora), deoarece restul câmpurilor sunt completate automat cu datele introduse de utilizator la înregistrare. În cazul în care perioada solicitată este deja ocupată de un alt pacient, solicitarea va fi respinsă și utilizatorul va fi rugat să aleagă o altă oră, respectiv o altă dată.

**Verificarea statusului programărilor solicitate** – în momentul accesării pagini „Programari”, toate programările utilizatorului respectiv vor fi afișate într-un tabel, ordonate după statusul programări (confirmate, neconfirmate, expirate). Pentru a găsi o programare mai rapid, utilizatorul poate filtra datele din tabel în funcție de statusul programări.

### Recepție

**Afișarea tuturor pacienților din cabinet** – după autentificarea recepției, din baza de date vor fi afișații toți pacienții din cabinet. Pacienții se încarcă automat în momentul încărcării pagini web, iar afișarea se va face într-un GridView și dedesubt vor fi butoanele pentru gestionarea datelor.

**Căutare pacient după CNP sau nume** – pentru a găsi mai ușor un anumit pacient, deasupra GridView-ului, se află două casete de text, una pentru căutarea pacientului după CNP și cealaltă pentru căutarea pacientului după nume. CNP-ul sau numele pacientului de care recepția are nevoie se va introduce în caseta de text corespunzătoare, după care se apasă butonul de căutare aferent casetei de text și căutarea în baza de date începe. Dacă pacientul căutat a fost găsit, el va fi afișat în tabel și automat și selectat, iar daca nu a fost găsit se va afișa un mesaj corespunzător.

**Selectarea pacientului din tabel** (GridView) – selectarea pacientului din tabel se face cu un singur clic pe rândul tabelului care conține informațiile despre pacientul dorit. După selectarea pacientului, recepția va putea modifică prezența pacientului în sala de așteptare și va primi acces la butoanele pentru gestionarea datelor pacientului.

**Anunțarea medicului că pacientul se află în sala de așteptare** – pentru ca medicul să fie anunțat că pacientul se află în sala de așteptare, pacientul mai întâi trebuie selectat și pe urmă modificată prezența din tabelul cu pacienții afișați. Pentru a efectua modificarea, sub tabel după selectarea pacientului apare o etichetă „Prezent” și un DropDownList unde trebuie selectată opțiunea „Da” și medicul va fi anunțat.

**Adăugare pacient nou** – pentru adăugarea pacientului în baza de date trebuie completate toate câmpurile obligatorii din formularul de adăugare pacient nou. De exemplu: nume, prenume, CNP, data nașterii, sex, etc.

**Vizualizare/editare fișă pacient** – pentru ca această funcție să fie disponibilă recepției, pacientul trebuie să fie selectat. Funcția permite vizualizarea și editarea fișei pacientului. Fișa pacientului conține date personale, datele de contact, domiciliu pacientului și alte informații legate de pacient.

**Vizualizare consultații neachitate** – în momentul accesări pagini „Consultatii neachitate”, din baza de date vor fi încărcate toate consultațiile care nu au fost achitate și ele vor fi afișate într-un tabel. În tabel vor fi afișate doar informațiile care ne vor fi de folos pentru tipărirea chitanței (număr consultație, CNP-ul pacientului, nume pacient, data și ora consultației, intervențiile efectuate și prețul pentru fiecare intervenție, numele medicului și costul total).

**Căutarea consultațiilor neachitate după CNP sau numele pacientului** – pentru afișarea consultațiilor neachitate ale unui anumit pacient, deasupra tabelului, la fel ca la căutarea pacientului

după CNP sau nume prezentată mai sus, se pot găsi două casete de text, câte una pentru fiecare posibilitate de căutare. Introducând CNP-ul sau numele pacientului în una dintre casetele de text corespunzătoare și după apăsarea butonului de căutare aferent casetei de text, începe căutarea în baza de date. Dacă pacientul are consultații neachitate ele vor fi afișate, iar dacă nu are, se va afișa un mesaj corespunzător.

**Selectarea consultației neachitate din tabel** – selectarea consultației din tabelul de consultații neachitate se face cu un singur clic pe rândul tabelului care conține informații despre acea consultație. După selectarea consultației, recepția va primi acces la butonul de „Tiparire chitnta” care se află sub tabel.

**Tipărire chitanță** – funcția se folosește pentru generarea de chitanțe și tipărirea lor la imprimantă, poate fi accesată din două pagini ale aplicației. Dacă butonul de tipărire chitanță este accesat din pagina principală, pacientul trebuie să fie selectat pentru ca funcția să fie disponibilă și ulterior trebuie selectată consultația pentru care se dorește achitarea unei sume de bani. Dacă butonul este accesat din pagina „Consultatii neachitate”, consultația trebuie să fie selectată în tabelul de consultații neachitate, ulterior mai trebuie introdusă suma de bani care se dorește achita și consultația ii gata de tipărit, pentru că datele legate de consultație deja se află în controalele corespunzătoare (datele despre consultație sunt transmise prin sesiuni din pagina „Consultatii neachitate”). În ambele cazuri datele pacientului sunt transmise (din pagina principală sau din pagina „Consultații neachitate) prin utilizarea de sesiuni și afișate în casete de text, care sunt neaccesibile pentru editare.

**Tipărire fișă pacient** – accesând pagina „Vizualizare/Editare fisa pacient”, la sfârșitul pagini se află un buton „Printare” care tipărește la imprimantă fișa pacientului.

**Afișarea tuturor programărilor solicitate** – în momentul accesării pagini „Programari”, din baza de date automat vor fi încărcate într-un tabel (GridView) toate programările solicitate, ordonate după statusul programări.

**Selectarea programări** – selectarea programări din tabelul de programări se face cu un singur clic pe rândul tabelului care conține informații asociate programări căutate. După selectarea programări, recepția va putea modifica statusul programării respective.

**Modificarea statusului programări și informarea pacientului prin adresa email când programarea este confirmată** – pentru ca statusul programări să fie modificat, programarea mai întâi trebuie selectată. Pentru a efectua modificarea, sub tabel după selectarea programări apare o etichetă „Modificare status programare” și un DorpDownList care oferă trei opțiuni („Confirmata”,

„Neconfirmata”, „Expirata”) și în funcție de opțiunea aleasă statusul programări va fi modificat. Dacă

programarea este confirmată, pacientul respectiv va primi mail precum că programarea lui din data cu tare și ora cu tare este confirmată.

**Afișarea programărilor în funcție de statusul programări ales** – dacă recepția dorește să își micșoreze timpul de căutare pentru anumite programări, ea poate filtra datele din tabel în funcție de statusul programări selectat dintre opțiunile (Toate, Confirmata, Neconfirmata, Expirata) oferite în DropDownList-ul „Status Programare”.

**Căutarea programărilor după numele pacientului** – pentru a găsi mai ușor programările unui pacient anume, deasupra tabelului cu programări se află o casetă de text unde putem introduce numele pacientului dorit și un DropDownList în care putem să selectăm diferite opțiuni (Toate, Confirmata, Neconfirmata, Expirata) pentru a filtra rezultatul căutări. Căutarea funcționează în felul următor: se introduce numele pacientului ale cărui programări ne interesează în caseta de text, după care se poate selecta statusul de programare dorit din DropDownList și apoi apăsăm butonul de căutare. Dacă căutarea a avut succes programările sunt afișate, daca nu, se va afișa un mesaj corespunzător.

### Medic

**Afișarea pacienților care sunt prezenți în sala de așteptarea** – după autentificarea medicului, din baza de date vor fi afișații toți pacienții prezenți în sala de așteptare. Pacienții se încarcă automat în momentul încărcării pagini web, iar afișarea se va face într-un tabel (GridView) și dedesubt vor fi butoanele pentru gestionarea datelor. Dacă între timp mai apar pacienți în sala de așteptare și în tabel nu sunt afișați, medicul poate să apese butonul „Reimprospatare” și tabelul va fi reîmprospătat.

**Selectarea pacientului din tabel** – selectarea pacientului din tabel se face la fel ca și în selecțiile prezentate mai sus, cu un singur clic pe rândul tabelului care conține informații despre pacientul care ne interesează. După selectarea pacientului, medicul va primi acces la butoanele pentru gestionarea datelor pacientului.

**Adăugare consultație nouă** - în momentul când pacientul se prezintă la medic, medicul va crea o consultație noua (salvată în baza de date), unde se va specifica dintele sau dinți la care s-a lucrat, diagnosticele, intervențiile efectuate, planul de tratament, iar restul datelor vor fi completate automat (datele personale ale pacientului, domiciliu, data și ora consultației, costul total, numele medicului).

**Adăugare radiografie nouă** – pentru adăugarea unei radiografii în baza de date trebuie încărcată radiografia pe pagina web și adăugat diagnosticul/diagnosticele, iar restul informaților vor fi încărcate automat pe pagina de adăugare radiografii.

**Vizualizare fișă pacient –** funcția permite vizualizarea fișei pacientului selectat. Fișa pacientului conține date personale, datele de contact, domiciliu pacientului și alte informații legate de pacient.

**Vizualizare consultații pacient** – funcția permite vizualizarea consultațiilor pacientului selectat (consultațiile efectuate la medicul autentificat). În momentul accesări pagini „Vizualizare consultatii pacient”, din baza de date vor fi încărcate toate consultațiile în două DropDownList-uri (DDL). Într- un DDL vor fi încărcate numărul consultațiilor și în celălalt, data consultaților, oferind posibilitatea selectării consultației care va fi afișată, ori după numărul consultației, ori după data consultației.

**Vizualizare radiografii** – funcția permite vizualizarea radiografiilor pacientului selectat. În momentul accesări pagini „Vizualizare radiografii”, din baza de date vor fi încărcate toate radiografiile și afișate ca ImageButton (buton imagine). La clic o data pe radiografia care se dorește a fi vizualizată, mai jos se vor afișa informații legate de aceea radiografie (nume imagine, data, diagnostic, observați), dacă se mai face încă o dată clic pe aceeași radiografie, radiografia va fi afișată în format mai mare – dimensiunea originala (ModalPopupExtender).

**Tipărire fișă pacient** - accesând pagina „Vizualizare fisa pacient”, la sfârșitul pagini se află un buton

„Printare” care tipărește la imprimantă fișa pacientului.

**Tipărire consultație** – accesând pagina „Vizualizare consultatii pacient”, la sfârșitul pagini se află un buton „Printare” care tipărește la imprimantă consultația selectată mai sus în pagina Web. Dacă consultația nu este selectată și se încearcă apăsarea butonului „Printare” se va afișa un mesaj de eroare.

**Afișare programări pentru ziua curentă, săptămâna curentă sau pentru o dată selectată manual de către medic** – în momentul accesării pagini „Programari”, din baza de date automat vor fi încărcate toate programările confirmate și afișate într-un tabel în funcție de medicul autentificat, ordonate după data și ora programării. Funcția oferă posibilitatea filtrării programărilor din tabel pentru ziua curentă, săptămâna curentă sau pentru o dată selectată manual de către medic.

## Interfața cu utilizatorul

În momentul pornirii aplicației va apărea pagina „Acasa” (figura 4.3.1) care pune la dispoziție opțiunile pentru accesarea sistemului informatic pentru gestionarea pacienților dintr-un cabinet stomatologic. În această pagină avem prezentat logo-ul cabinetului stomatologic împreună cu mesajul

„Bine ati venit” și „Va rugam să alegeti un cont potrivit voua”. Pe lângă cele două mesaje pagina mai conține și trei butoane:

* + - „CONT PACIENT” – acest buton ne redirecționează spre pagina de autentificare a pacientului;
    - „CONT RECEPTIE” – acest buton ne redirecționează spre pagina de autentificare a recepției;
    - „CONT MEDIC” – acest buton ne redirecționează spre pagina de autentificare a medicului.



Figura 4.3.1 Pagina „Acasa”

### Interfața aplicației pentru pacient

Pagina „Autentificare pacient” (figura 4.3.2) este pagina de autentificare a pacientului. Pentru ca pacientul să aibă acces la aplicație, dacă are cont este de ajuns să se autentifice, iar dacă nu are cont trebuie să își facă unul. Autentificarea și-o face introducând CNP-ul și parola, folosite anterior pentru înregistrare, după care este nevoie de acționarea butonului „Autentificare”. La acționarea butonului de autentificare pot exista mai multe scenarii: (mesajele de eroare vor fi afișate sub casta de text corespunzătoare)

1. Erori la introducerea CNP-ului:
   1. Nu s-a introdus CNP-ul – „Nu ati introdus CNP-ul.” (figura 4.3.3);
   2. CNP-ul conține litere – „CNP-ul nu are voie sa contina litere.”;
   3. CNP-ul nu conține 13 cifre – „CNP-ul trebuie sa contina 13 cifre.”;
   4. CNP-ul nu este înregistrat – „CNP-ul introdus nu are asociat nici un cont.”;
2. Erori la introducerea parolei:
   1. Nu s-a introdus parola – „Nu ati introdus parola.”;
   2. Parola introdusă conține mai puțin de 8 caractere sau mai mult de 15 caractere –

„Parola poate sa contina minim 8 caractere si maxim 15 caractere.”;

* 1. Parola introdusă nu corespunde CNP-ului introdus mai sus – „Parola introdusa este gresita!” (figura 4.3.4);

1. CNP-ul corespunde parolei introduse și pacientul va fi redirecționat la pagina principala a pacientului (autentificarea efectuată cu succes).

Dacă pacientul și-a uitat parola, poate da clic pe „Ati uitat parola?” și se va deschide mai jos un panou (CollapsiblePanelExtender) în care trebuie introdus CNP-ul pacientului și adresa email care a fost utilizată de pacient la crearea contului. Parola va fi trimisă pe adresa email, după introducerea datelor necesare și acționarea butonului „Trimite”. La acționarea butonului „Trimite” pot exista mai multe scenarii:

1. Erori la introducerea CNP-ului – aceleași cauze și mesaje ca și la acționarea butonului de autentificare;
2. Erori la introducerea adresei email:
   1. Nu s-a introdus adresa email – „Nu ati introdus adresa e-mail.”;
   2. Adresa email introdusă este invalidă – „E-mail-ul poate sa contina litere, cifre,-, \_. Obligatoriu trebuie sa contina @nume provider.nume domain” (figura 4.3.5);
   3. Adresa email introdusă nu este asociată CNP-ului introdus mai sus – „Adresa e-mail nu corespunde cu CNP-ul `CNP utilizator`”.
3. CNP-ul și email-ul introdus sunt corecte (pacientul va primi parola pe email).

Dacă pacientul nu are cont, poate accesa link-ul „Cont nou” care îl va redirecționa la pagina de înregistrare.

Pe lângă toate cele spuse, pagina de autentificare mai dispune și de un meniu și un buton „Anulare”. La acționarea butonului de anulare, toate controalele din pagina respectivă vor fi resetate la valorile inițiale și pe pagină se va afișa următorul mesaj „Anulare: toate campurile sau fost resetate la valorile initiale.”.

În continuare toate paginile (inclusiv interferențele pentru recepție și medic) car vor fi prezentate vor conține un meniu corespunzător pagini și posibil, dacă nu sigur, un buton de anulare, care are aceeași funcție de resetare în toate paginile (funcția descrisă mai sus).



Figura 4.3.2 Pagina „Autentificare pacient”

|  |  |
| --- | --- |
| Figura 4.3.3 Nu s-a introdus CNP-ul | Figura 4.3.4 Parola introdusă nu corespunde cu CNP-ul introdus mai sus |

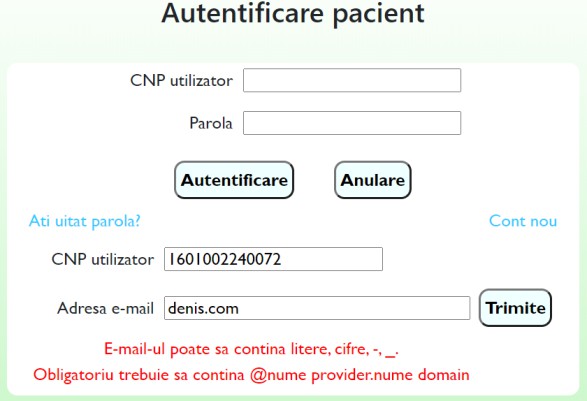


Figura 4.3.5 Adresa email introdusă este invalidă

Pagina „Inregistrare cont nou” (figura 4.3.6) este pagina pentru înregistrarea utilizatorilor noi care doresc să utilizeze aplicația. Pentru înregistrare trebuie completate toate câmpurile obligatorii (marcate cu „\*” Asterix) cu datele personale ale pacientului, datele de contact și parola care o va utiliza pentru autentificare. Dacă utilizatorul nu își introduce adresa email nu va putea folosi funcția

„Ati uitat parola?”. Pentru a se înregistra, după ce sau completat toate câmpurile obligatori, trebuie acționat butonul de înregistrare și pot rezulta mai multe scenarii:

1. Erori la introducerea numelui:
   1. Nu s-a introdus numele – „Nu ati introdus numele.”;
   2. Numele nu are o formă corespunzătoare – „Numele nu are voie sa contina altceva in afara de litere.” (figura 4.3.7);
2. Erori la introducerea prenumelui – aceleași cauze și mesaje ca la introducerea numelui;
3. Erori la introducerea CNP-ului – pe lângă erorile legate de introducerea CNP-ului amintite mai sus, la autentificare, în această pagină mai avem o verificare în plus:
   1. Dacă CNP-ul introdus este invalid (neadevărat) – „Nu exista CNP-ul introdus. CNP- ul este gresit.”;
4. Erori la introducerea numărului de telefon:
   1. Nu s-a introdus numărul de telefon – „Nu ati introdus numarul de telefon.”;
   2. Numărul de telefon conține litere – „Numarul de telefeon poate sa contina numai cifre.”;
5. Eroare la introducerea adresei email:
   1. Adresa email introdusă este invalidă – „E-mail-ul poate sa contina litere, cifre,-, \_. Obligatoriu trebuie sa contina @nume provider.nume domain”;
6. Erori la introducerea parolei:
   1. Nu s-a introdus parola – „Nu ati introdus parola.”;
   2. Parola introdusă conține mai puțin de 8 caractere sau mai mult de 15 caractere –

„Parola poate sa contina minim 8 caractere si maxim 15 caractere.”;

1. Erori la confirmarea parolei:
   1. Nu s-a confirmat parola – „Nu ati confirmat parola introdusa mai sus.”;
   2. Parola și confirmarea parolei nu coincid – „Parola introdusa nu coincide cu parola introdusa mai sus.” (figura 4.3.8);
2. Eroare la înregistrarea contului:
   1. CNP-ul este deja folosit la înregistrarea unui alt utilizator – „CNP-ului `CNP pacient` ii este asociat un cont si nu se poate reutiliza pentru a crea un alt cont.”;
3. Înregistrarea a avut succes și în câteva momente (timpul de așteptare 5 secunde) va fi redirecționat automat pe pagina de autentificare
   1. Mesajul afișat – „Inregistrare a fost efectuata cu succes. In 5 secunde veti fi redirectionati pe pagina de Autentificare.” (figura 4.3.9);

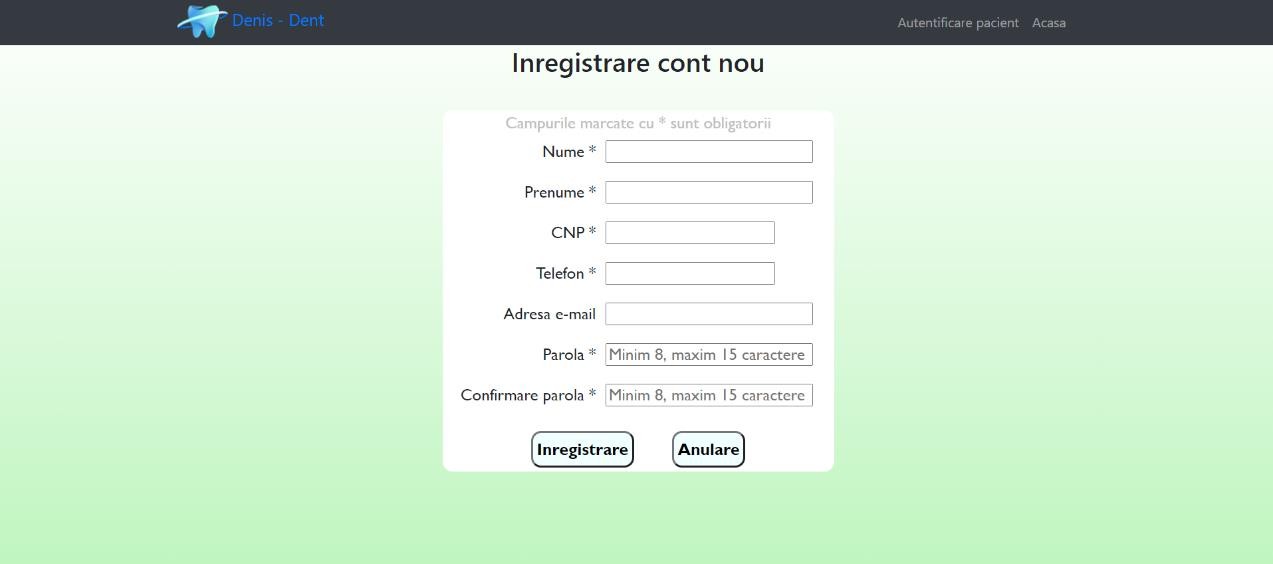


Figura 4.3.6 Pagina „Inregistrare cont nou”

|  |  |  |
| --- | --- | --- |
| Figura 4.3.7 Numele nu are o formă corespunzătoare | Figura 4.3.8 Parola și confirmarea parolei nu coincid | Figura 4.3.9 Înregistrarea a avut succes |

După autentificare, se ajunge la pagina „Pagina Principala” (figura 4.3.10). Aceasta conține mai multe tabele cu catalogul de intervenții și la sfârșitul pagini se află și un formulare pentru solicitarea programări (figura 4.3.11). La încărcarea pagini doar primul catalog va fi deschis, restul vor fi închise, iar pentru deschiderea lor doar trebuie dat clic pe numele catalogului (CollaspiblePanelExtender).

Pentru a solicita o programare, pacientul nu are de cât să selecteze medicul, motivul programări, data, ora programări și să acționeze butonul „Solicitati programarea”. Restul datelor, cum ar fi datele personale, datele de contact vor fi completate automat cu datele folosite anterior pentru înregistrare.

După acționarea butonului „Solicitati programarea” pot exista mai multe scenarii:

1. Erori la introducerea numelui – aceleași cauze și mesaje ca pentru fiecare câmp unde trebuie introdus numele;
2. Erori la introducerea prenumelui - aceleași cauze și mesaje ca pentru fiecare câmp unde trebuie introdus prenumele;
3. Erori la introducerea numărului de telefon - aceleași cauze și mesaje ca pentru fiecare câmp unde trebuie introdus numărul de telefon;
4. Eroare la introducerea adresei email - aceleași cauze și mesaje ca pentru fiecare câmp unde trebuie introdusă adresa email;
5. Eroare la selectarea medicului:
   1. Nu s-a selectat medicul – „Nu ati selectat medicul stomatolog dorit.”;
6. Eroare la selectarea motivului de programare:
   1. Nu s-a selectat motivul programări – „Nu ati selectat motivul programari.”;
7. Erori la selectarea datei programări:
   1. Nu s-a selectat data – „Nu ati ales data dorita.”;
   2. Data aleasă este expirată – „Data aleasă este indisponibila. Nu se poate realiza programarea pentru data de azi sau zilele trecute deja.”;
   3. Data selectată nu reprezintă o zi lucrătoare pentru cabinet – „Sambata și duminica cabinetul stomatologic Denis – Dent nu lucreaza.” (figura 4.3.12);
8. Erori la selectarea orei programări:
   1. Nu s-a selectat ora – „Nu ati selectat ora dorita.”;
   2. Ora selectată este deja ocupată – „Incercati sa alegeti o alta ora pentru programarea dumneavoastra. Daca nici o ora nu este disponibila pentru data selectata, va rugam sa incercati pentru o data diferita.” (figura 4.3.13);
9. Solicitarea programări a avut succes
   1. Mesaj afișat – „Programarea a fost solicitatt cu succes. Statusul programari il puteti verifica accesand pagina Programari din meniu”.

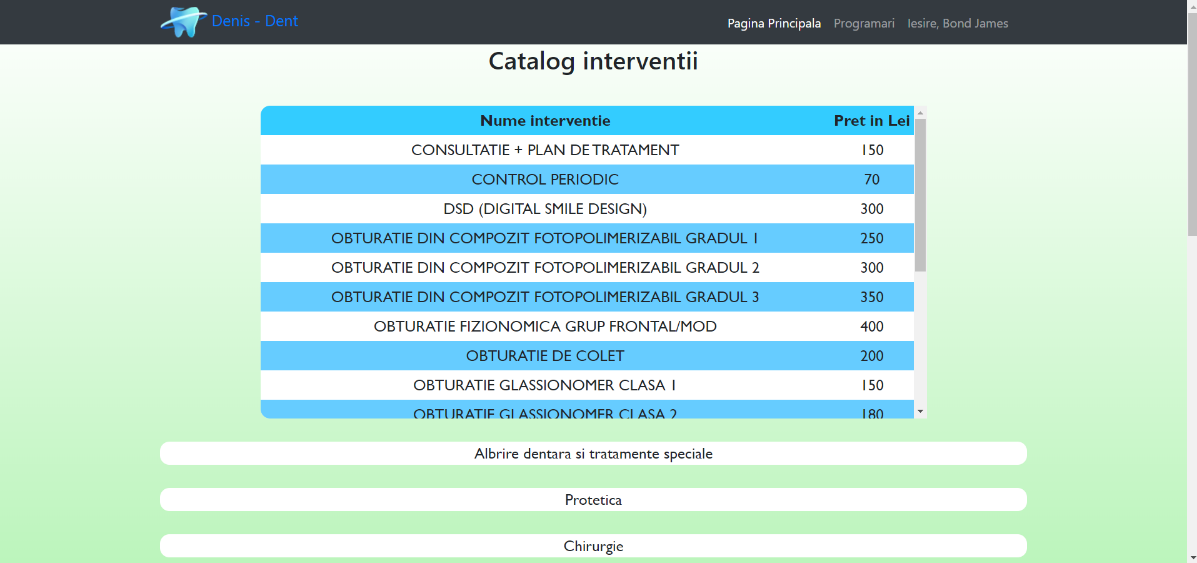


Figura 4.3.10 Pagina „Pagina Principala”

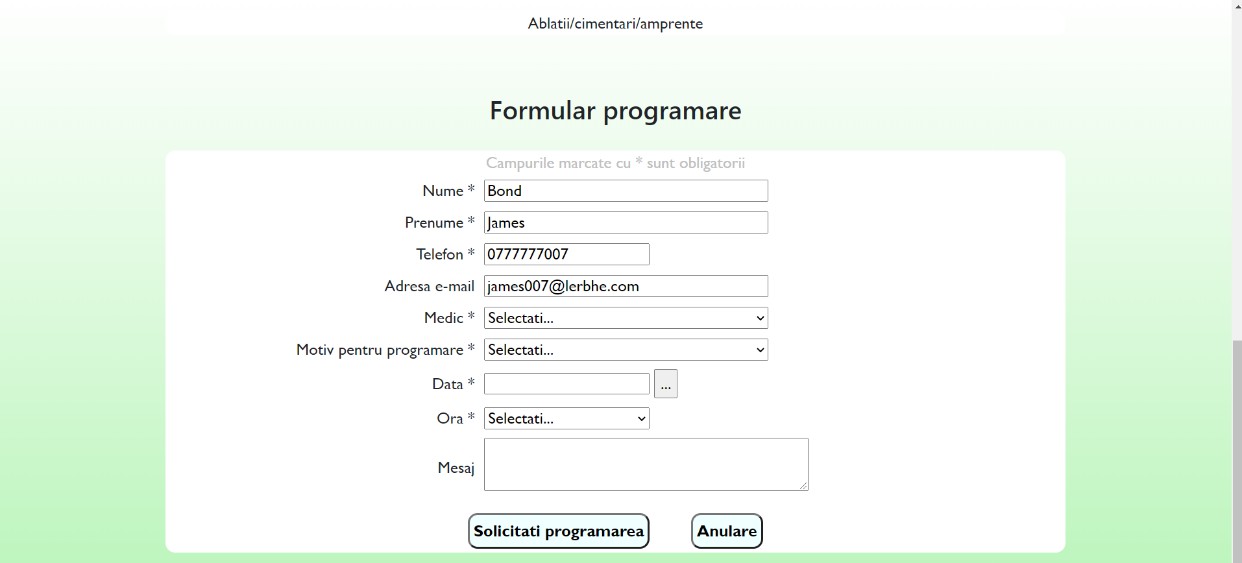


Figura 4.3.11 Formular programare

|  |  |
| --- | --- |
| Figura 4.3.12 Data selectată nu reprezintă o zi lucrătoare pentru cabinet | Figura 4.3.13 Ora selectată este deja ocupată |

Pe lângă vizualizarea catalogului de intervenții, solicitarea de programări, pacientul mai poate și să vizualizeze statusul programări accesând pagina „Programari” (figura 4.3.14) din meniu.

Pagina „Programari” conține un tabel unde sunt afișate toate programările solicitate de pacientul autentificat. Dacă pacientul până acuma nu are solicitată nici o programare, în loc de tabelul cu programări va avea afișat mesajul „Nu aveti solicitata nici o programare”, iar dacă pacientul are programări solicitate la cabinet, de deasupra tabelului va fi afișat mesajul „Au fost gasite urmatoarele programari”.

Pentru a vizualiza doar programările care au un status de programare anume, utilizatorul poate filtra informațiile din tabel acționând următoarele butoane:

* Confirmate – va afișa în tabel programările confirmate sau un mesaj dacă nu sau găsit programări confirmate („Nu aveti nici o programare confirmata”);
* Neconfirmate – va afișa în tabel programările neconfirmate sau un mesaj dacă nu sau găsit programări neconfirmate („Nu aveti nici o programare neconfirmata”);
* Expirate – va afișa în tabel programările expirate și un mesaj dacă nu sau găsit programări expirate („Nu aveti nici o programare expirata”) (figura 4.3.15).

Dacă utilizatorul dorește să vadă toate programările după acționarea unuia dintre butoanele prezentate mai sus, poate acționa butonul „Toate” care va afișa toate programările pacientului.

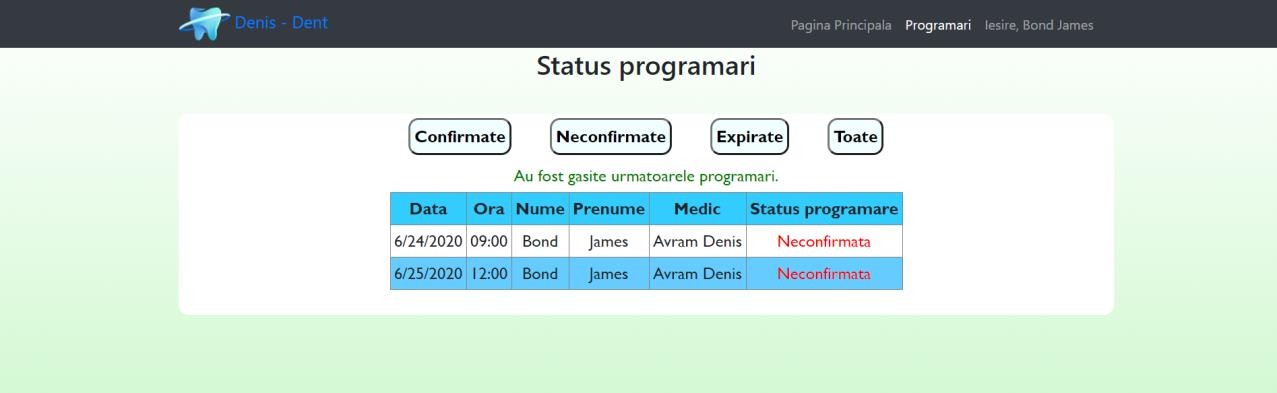


Figura 4.3.14 Pagina „Programari”

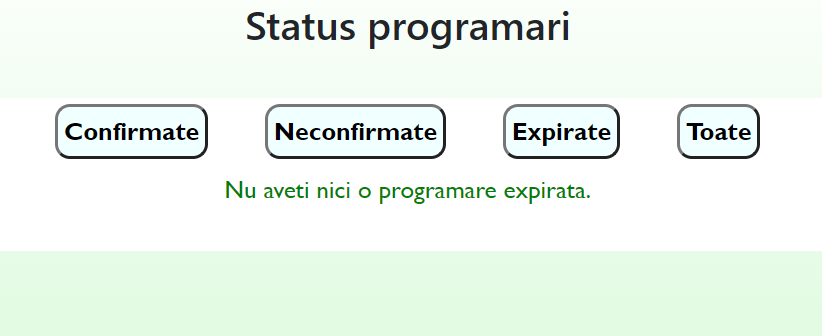


Figura 4.3.15 Acționarea butonului „Expirate”

În continuare unde apar aceleași scenarii de erori care deja au fost prezentate, vor fi doar enumerate, iar scenariile (erorile) ce apar de nou vor fi explicate ca și până acuma.

### Interfața aplicației pentru recepție

Pagina „Autentificare receptie” (figura 4.3.16) reprezintă pagina pentru autentificarea recepției. În cazul recepției avem un cont comun pentru toți recepționiști și nu conturi separate cum avem pentru medici sau pacienți. Pentru ca recepția să aibă acces la aplicație, doar trebuie să se autentifice, din cauza faptului că contul este deja creat. Autentificarea se va face introducând parola și apoi acționarea butonului de autentificare. Ca și la pacient, la acționarea butonului „Autentificare” pot rezulta mai multe scenarii, în cazul ăsta două:

1. Erori la introducerea parolei:
   1. Nu s-a introdus parola – „Nu ati introdus parola.” (figura 4.3.17);
   2. Parola introdusă este greșită – „Parola introdusă este gresita!”;
2. Parola introdusă este corectă și recepția va fi redirecționat la pagina principala a recepției (autentificarea efectuată cu succes).

În cazul care recepția și-a uitat parola, poate da clic pe „Ati uitat parola?” și se va deschide mai jos un panou (CollapsiblePanelExtender) cu o casetă de text și un buton. Pentru a recupera parola, în

caseta de text trebuie introdus email-ul cabinetului și pe urmă trebuie apăsat butonul „Trimite”. La apăsarea butonului „Trimite” pot rezulta două scenarii:

1. Erori la introducerea adresei email:
   1. Nu s-a introdus adresa email – „Nu ati introdsu adresa e-mail.”;
   2. Adresa email introdusă este invalidă – „E-mail-ul poate sa contina litere, cifre,-, \_. Obligatoriu trebuie sa contina @nume provider.nume domain”;
   3. Adresa email introdusă nu reprezintă email-ul cabinetului – „Adresa e-mail nu corespunde pentru contul de recepție.” (figura 4.3.18);
2. Adresa email este corectă, recepția va primi parola pe email-ul cabinetului.



Figura 4.3.16 Pagina „Autentificare receptie”

|  |  |
| --- | --- |
| Figura 4.3.17 Nu s-a introdus parola | Figura 4.3.18 Adresa email introdusă nu reprezintă email-ul cabinetului |

După autentificare, se ajunge la pagina „Pagina Principala” (figura 4.3.19). Aceasta conține un tabel cu toți pacienți din cabinet, casete text și butoane pentru căutare, iar mai jos (sub tabelul cu pacienți) se află butoanele pentru gestionarea pacienților. Cele două butoane „Adauga pacient nou” și

„Consultatii neachitate” vor fi activate tot timpul, iar butoanele „Vizualizare / Editare fisa pacient”,

„Tiparire chitanta” și DropDownLis-ul pentru modificarea prezentei pacientului în sala de așteptare vor fi activate doar când pacientul este selectat. Deoarece la acționarea acestor două butoane („Vizualizare / Editare fisa pacient”, „Tiparire chitanta”) se vor încărca datele pacientului selectat în sesiuni și transmise în pagina aferentă butonului acționat.

La acționarea butonului de căutare a pacientului după CNP (sau nume) pot rezulta mai multe scenarii:

1. Erori la introducerea CNP-ului ( sau numelui) (figura 4.3.20);
2. CNP-ul (sau numele) introdus nu există în baza de date a cabinetului:
   1. Mesajul afișat – „Pacientul cu CNP-ul `CNP cautat` nu a fost gasit.” („Pacientul cu numele `nume cautat` nu a fost gasit” (figura 4.3.21));
3. Căutarea a avut succes și pacientul va fi afișat în tabel, împreună cu mesajul „Pacientul cautat a fost gasit.” (figura 4.3.22).

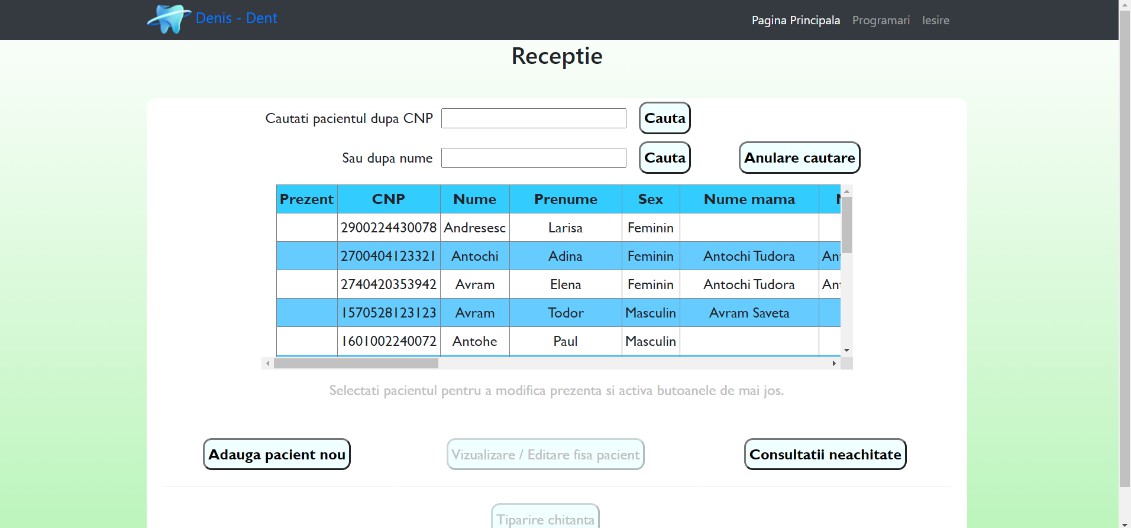


Figura 4.3.19 Pagina „Pagina Principala”

|  |  |
| --- | --- |
| Figura 4.3.20 CNP-ul nu conține 13 cifre | Figura 4.3.21 Numele introdus nu există în baza de date a cabinetului |

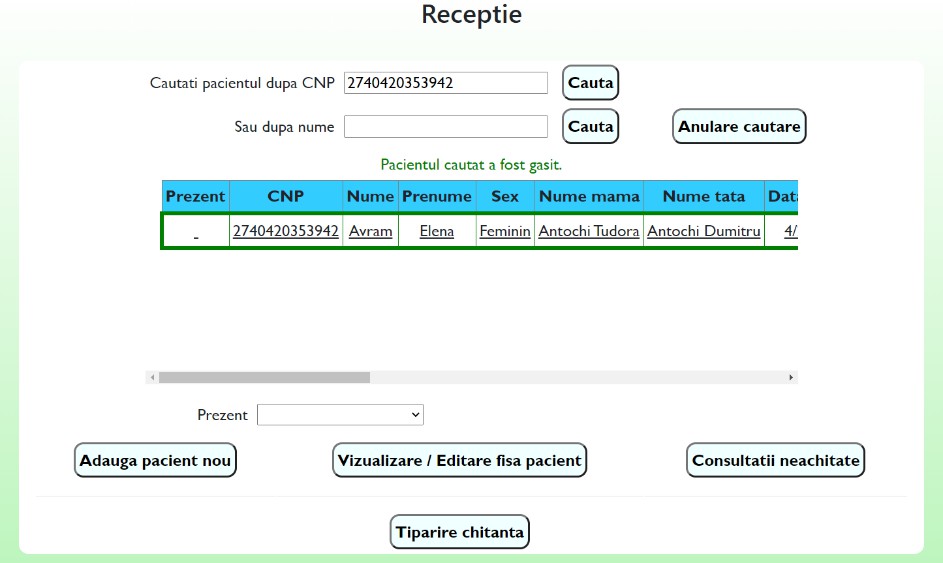


Figura 4.3.22 Căutarea a avut succes

Pagina „Adaugare pacient nou” (figura 4.3.23) este pagina pentru adăugarea pacienților noi care sau prezentat la cabinet. Pentru a adăuga pacientul trebuie completate toate câmpurile obligatorii (marcate cu „\*” Asterix) cu datele personale, datele de contact, domiciliul pacientului și pe urmă acționat butonul „Adauga pacient nou”. Dacă pacientul înainte să se prezinte la cabinet și-a făcut cont de pacient, procedura de adăugare va fi mult mai rapidă (introducând CNP-ul pacientului, datele personale vor fi automat completate). În cazul în care pacientul nu are cont de pacient, iarăși nu va fii totul așa de negru, introducând CNP-ul, data nașterii și sexul vor fi completate automat. DropDwonList-ul care conține județele, se încarcă automat la accesarea acestei pagini (evenimentul

„Page\_Load”), iar DDL-ul care conține orașele se încarcă în momentul selectări județului. Scenariile ce pot să apară după acționarea butonului de adăugare pacient sunt următoarele:

1. Erori la introducerea numelui;
2. Erori la introducerea prenumelui;
3. Erori la introducerea CNP-ului;
4. Eroare la introducerea datei de naștere:
   1. Nu s-a introdus data nașterii – „Nu ati introdus data nasterii.”;
5. Eroare la selectarea sexului:
   1. Nu s-a selectat sexul – „Nu ați selectat sexul.”;
6. Erori la introducerea localități nașteri:
   1. Nu s-a introdus localitatea nașterii – „Nu ati introdus loclitatea nașterii.”;
   2. Localitatea nașterii introdusă nu are o formă corespunzătoare – „Localitatea nasterii are voie sa contina: litere, spatii, `-`.”;
7. Erori la introducerea numărului de telefon;
8. Eroare la introducerea adresei email;
9. Eroare la selectarea județului (domiciliu):
   1. Nu s-a selectat județul – „Nu ati selectat judetul.” (figura 4.3.24);
10. Eroare la selectarea orașului (domiciliu):
    1. Nu s-a selectat orașul – „Nu ati selectat orasul.”;
11. Eroare la introducerea străzi (domiciliu):
    1. Nu s-a introdus strada – „Nu ati introdus strada domiciliului.”;
12. Erori la introducerea valorilor numerice:
    1. Numărul introdus are valoare negativă – „Numarul poate avea numai valori pozitive. (>=0)” (figura 4.3.25);
    2. Blocul introdus are valoare negativă – „Blocul poate avea numai valori pozitive. (>=0)”;
    3. Scara introdusa are valori negativă – „Scara poate avea numai valori pozitive. (>=0)”;
    4. Numărul apartamentului introdus are valoare negativă – „Numarul apartamentului poate avea numai valori pozitive. (>=0)”;
13. Eroare la introducerea numelui medicului de familie;
    1. Numele medicului de familie introdus nu are o formă corespunzătoare – „Numele medicului de familie are voie sa contina numai litere si spatii libere.”;
14. Eroare la introducerea numelui mamei:
    1. Numele mamei nu are o formă corespunzătoare – „Numele mamei are voie sa contina numai litere si spatii libere.”;
15. Eroare la introducerea numelui tatălui:
    1. Numele tatălui nu are o formă corespunzătoare – „Numele tatalui are voie sa contina numai litere si spatii libere.” (figura 4.3.26);
16. Adăugarea pacientului a avut succes și în câteva momente (timpul de așteptare 5 secunde) recepția va fi redirecționată înapoi pe pagina principală
    1. Mesajul afișat – „Pacientul a fost adugat cu succes. In 5 secunde veti fi redirectionati pe Pagina Principala.”;

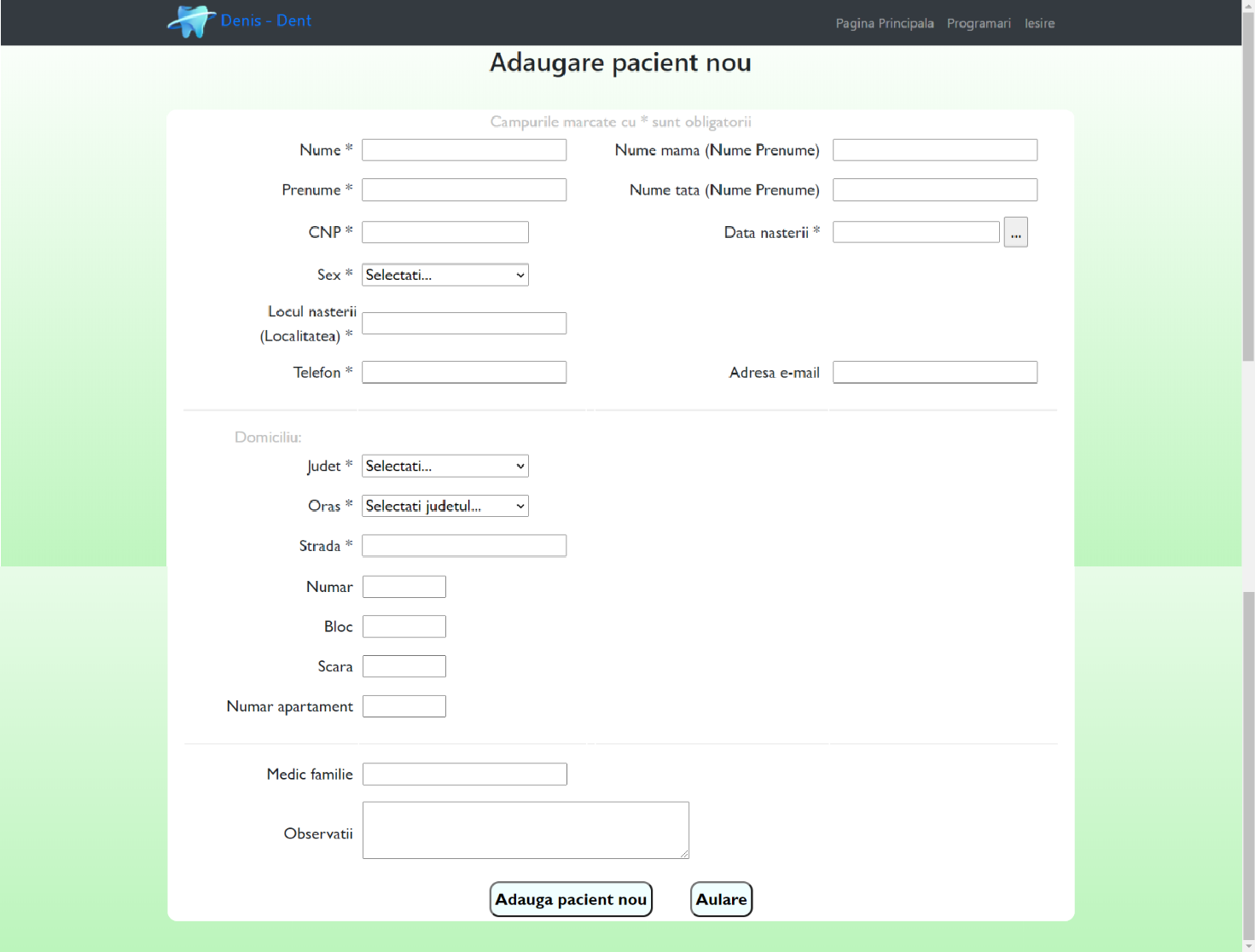


Figura 4.3.23 Pagina „Adaugare pacient nou”

|  |  |
| --- | --- |
| Figura 4.3.24 Nu s-a selectat județul | Figura 4.3.25 Numărul introdus are valoare negativă |

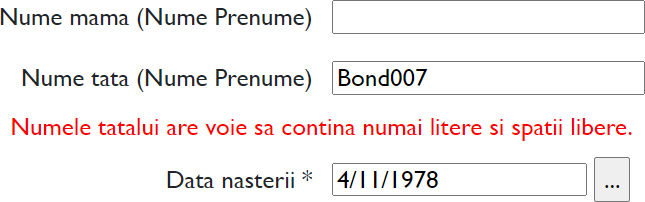


Figura 4.3.26 Numele tatălui nu are o formă corespunzătoare

Pagina „Vizualizare / Editare fisa pacient” (figura 4.3.27) poate fi accesată doar dacă pacientul înainte a fost selectat și oferă posibilitatea de a modifica, vizualiza datele pacientului. Această pagină arată identic cu pagina de adăugare pacient nou, dar în plus lângă fiecare control mai avem 3 butoane (ImageButton) – la accesarea pagini doar un singur buton pentru fiecare control este vizibil, restul vor fi vizibili după acționarea unuia dintre butoanele afișate. Cele trei butoane oferă posibilitatea de a modifica fiecare câmp în parte (TextBox, DropDownList). Toate informațiile vizibile în această pagină au fost transmise, utilizând sesiuni, din pagina principala.

Dacă dorim să modificăm un anumit câmp, de exemplu numele, se va apasă butonul „Editare.” care se află lângă caseta de text pentru nume. La acționarea butonului de editare el va dispărea și celelalte două butoane vor fi vizibile, butoanele „Salvare.” și „Anulare.”. Dacă se acționează butonul de anulare câmpul va rămâne neschimbat, iar dacă se acționează butonul de salvare, mai întâi se va verifica dacă ce s-a introdus nu generează erori specifice câmpului modificat (tot același erori care pot să apară la adăugarea pacientului, prezentate anterior) și dacă erori nu apar, modificarea va fi salvată (figura 4.3.28), altfel mesajul de eroare va fi afișat si procesul de salvare va fi întrerupt (figura 4.3.29). Acest proces se repetă pentru oricare dintre câmpurile care se doresc a fi modificate.

Tot in această pagini există și un buton pentru tipărirea fișei pacientului (butonul „Printare”).

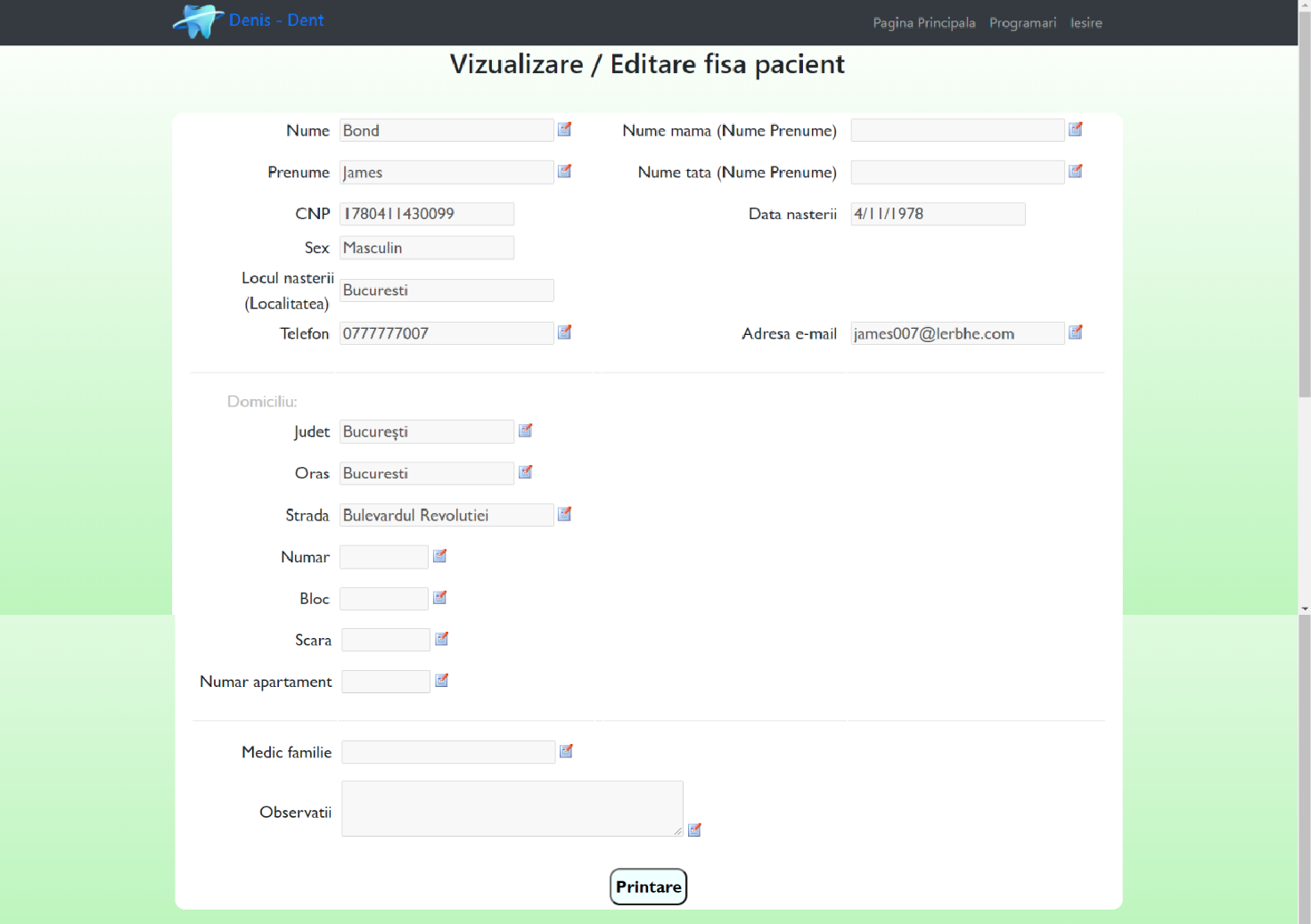


Figura 4.3.27 Pagina „Vizualizare / Editare fisa pacient”

|  |  |
| --- | --- |
| Figura 4.3.28 Salvarea modificări | Figura 4.3.29 Eroare la selectarea orașului (domiciliu) |

Pagina „Consultatii neachitate” (figura 4.3.28) este compusă dintr-un tabel în care sunt afișate toate consultațiile care au rămas neachitate, un buton pentru tipărirea de chitanțe și două casete de text cu două butoane de căutare. La accesarea pagini, butonul „Tiparire chitanta” va rămâne inactiv, iar după selectarea unei dintre consultațiile afișate el va putea fi acționat. La fel ca și la pagina principală unde recepția pentru a găsi mai rapid si mai ușor un pacient putea folosi posibilitatea de căutare a pacientului după CNP sau nume și în această pagină se poate regăsi aceeași posibilitate, dar în cazul ăsta se vor căuta consultațiile neachitate ale pacientului după CNP-ul și numele său.

Validările (și mesajele de eroare) câmpurilor CNP și nume sunt aceeași ca și la pagina principală, doar diferă mesajele care vor fi afișate în următoarele cazuri:

* Nu au fost găsite consultați pentru CNP-ul (numele) introdus – „Pacientul cu CNP-ul `CNP pacient` nu are nici o consultatie.” („Pacientul cu numele `nume cautat` nu are nici o consultatie.” (figura 4.3.29));
* Pacientul nu are consultații neachitate – „Pacientul cu CNP-ul `CNP pacient` nu are nici o consultatie neachitata.”(figura 4.3.30) („Pacientul cu numele `nume cautat` nu are nici o consultatie neachitata.”);
* Pentru CNP-ul (numele) introdus au fost găsite consultații neachitate – „Pentru pacientul cu CNP-ul `CNP pacient` au fost gasite urmatoarele consultatii neachitate.” („Pentru pacientul cu numele `nume cautat` au fost gasite urmatoarele consultatii neachitate.” (figura 4.3.31)).

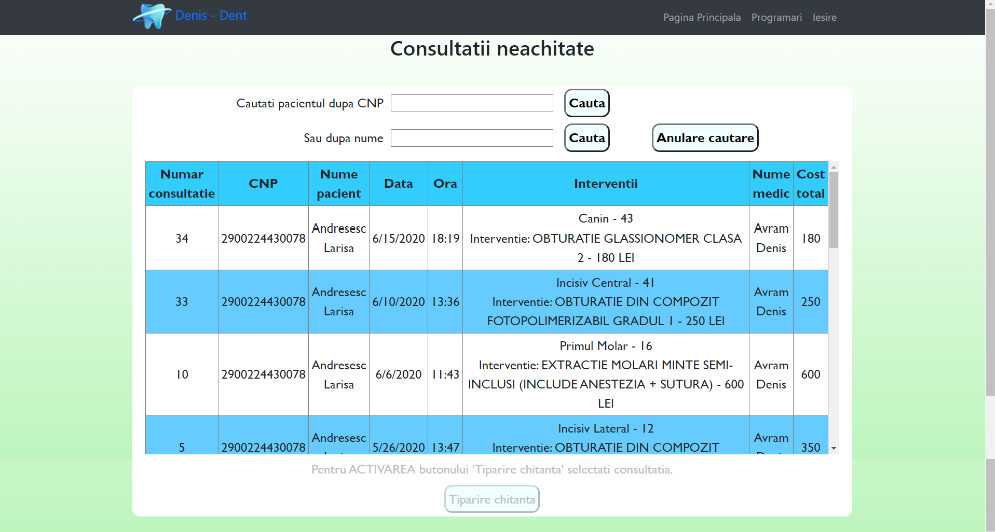


Figura 4.3.28 Pagina „Consultatii neachitate”

|  |  |
| --- | --- |
| Figura 4.3.29 Nu au fost găsite consultați pentru numele introdus | Figura 4.3.30 Pacientul nu are consultații neachitate |

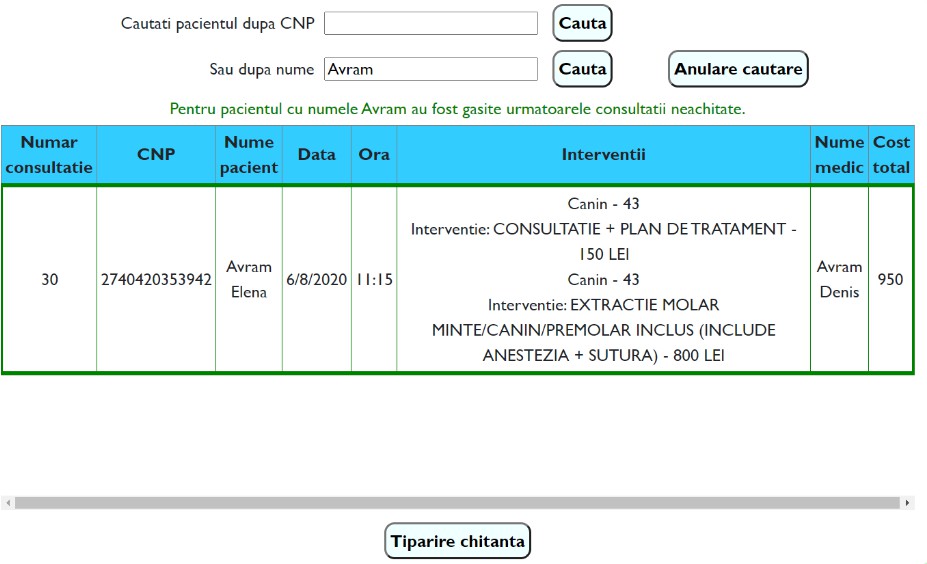


Figura 4.3.31 Pentru numele introdus au fost găsite consultații neachitate

Pagina „Tiparire chitanta” (figura 4.3.32) poate fi accesată din două pagini ale aplicației („Pagina Principala” și „Consultatii neachitate”). Ea are rolul generări și tipăriri chitanțelor pentru serviciile oferite de cabinet. Pentru a genera (tipării) chitanța, ea trebuie să fie selectată (din DropDonwList-ul

„Numar consultatie”), apoi introdusă suma care se dorește a fi achitată și la sfârșit acționat butonul

„Tipareste chitanta”. La acționarea butonului de tipărire chitanță pot rezulta trei scenarii:

1. Eroare la selectarea consultației:
   1. Nu s-a selectat consultația – „Nu ati selectat numarul consultatiei.”;
2. Eroare la introducerea sumei achitată azi:
   1. Se introduce suma de achitat, dar consultația care se dorește achita nu este selectată –

„Nu ati selectat numarul consultatie.”;

* 1. Suma introdusă este mai mare de cât suma ce trebuie achitată – „Suma introdusa este mai mare de cat este nevoie de achitat.” (figura 4.3.33);
  2. Nu s-a introdus suma achitată azi – „Nu ati introdus suma care urmeaza sa fie achitata.” (figura 4.3.34);

1. Chitanța a fost generată cu succes și se va deschide o fereastră nouă pentru tipărire ei
   1. Mesajul afișat – „Chitanta a fost tiparita. Pentru a revenii la pagina principala, dati clic pe Pagina Principala din meniu.”.

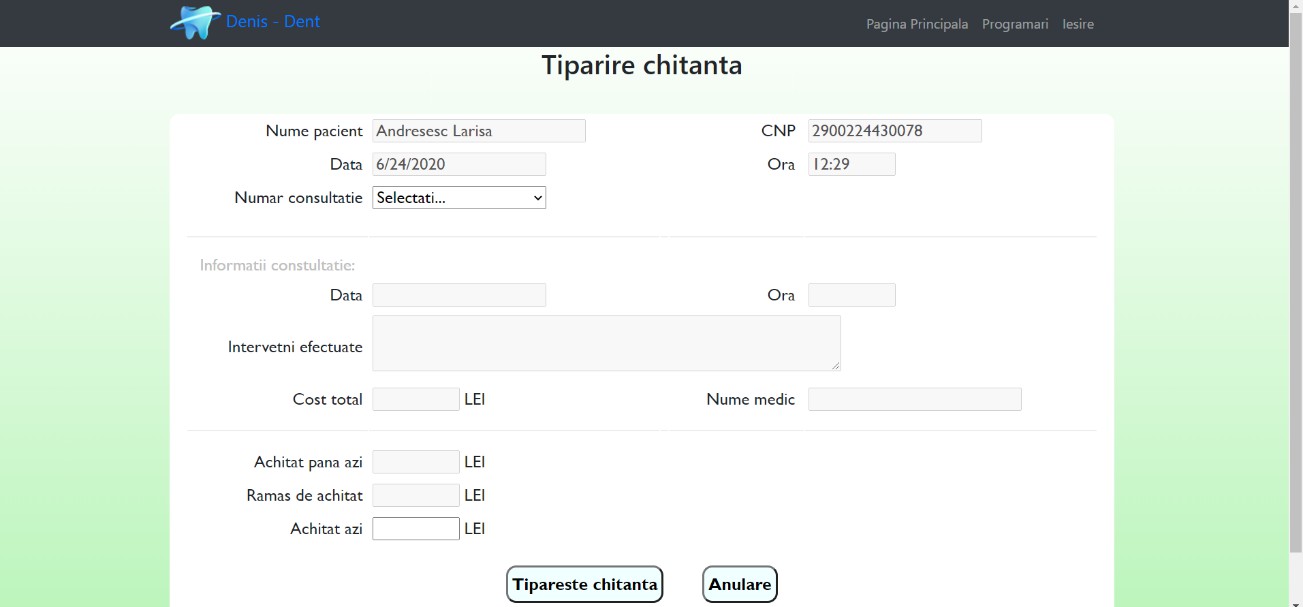


Figura 4.3.32 Pagina „Tiparire chitanta”

|  |  |
| --- | --- |
| Figura 4.3.33 Suma introdusă este mai mare de cât suma ce trebuie achitată | Figura 4.3.34 Nu s-a introdus suma achitată azi |

Pagina „Programari” (figura 4.3.35) conține un tabel unde sunt afișate toate programările solicitate de pacienți. În această pagină recepția poate modifica statusul programărilor solicitate. Pentru a efectua această modificare, programarea trebuie să fie selectată. Dacă până în momentul de față nu a fost solicitată nici o programare, în loc de tabelul cu programări va fi afișat mesajul „Nu aveti solicitata nici o programare.”, iar dacă cabinetul are deja programări solicitate deasupra tabelului va fi afișat mesajul „Au fost gasite urmatoarele programari.” (figura 4.3.35).

Modificarea statusului programări poate declanșa diferite scenarii:

* Dacă se încearcă confirmarea programări, pe data de azi, dar ora programări a trecut deja se va declanșa o eroare și programarea indiferent de statusul programării pe care îl avea, va fi pusă ca expirată:
  + Mesajul afișat – „Programarea selectata este expirata. (Data programari si ora a expirat)” (figura 4.3.36);
* Dacă se încearcă confirmarea programări, a cărei dată a trecut deja se va declanșa o eroare și programarea indiferent de statusul programării pe care îl avea, va fi pusă ca expirată:
  + Mesajul afișat – „Programarea selectata este expirata. (Data programari a expirat)”;
* Confirmarea programării a avut succes și pe email, pacientul va fi anunțat că programarea ia fost confirmată
  + Mesajul afișat – „Confirmarea programari a fost trimisa pe email-ul: `email pacient`” (figura 4.3.37).

Pentru ca recepția să își micșoreze timpul de căutare pentru anumite programări, poate filtra datele din tabel în funcție de statusul programări selectat în DropDownLis-ul „Status programare” sau/și folosirea posibilități de căutare a programărilor după numele pacientului.

Validările (și mesajele de eroare) câmpului de căutare după nume sunt aceeași ca și la pagina principală, doar diferă mesajele care vor fi afișate în următoarele cazuri:

* Nu au fost găsite programări pentru numele introdus – „Nu s-a gasit nici o programare pentru pacientul cu numele: `nume cautat`”;
* Pentru numele introdus au fost găsite programări – „Pentru pacientul cu numele `nume cautat` au fost găsite urmatoarele programari.”;

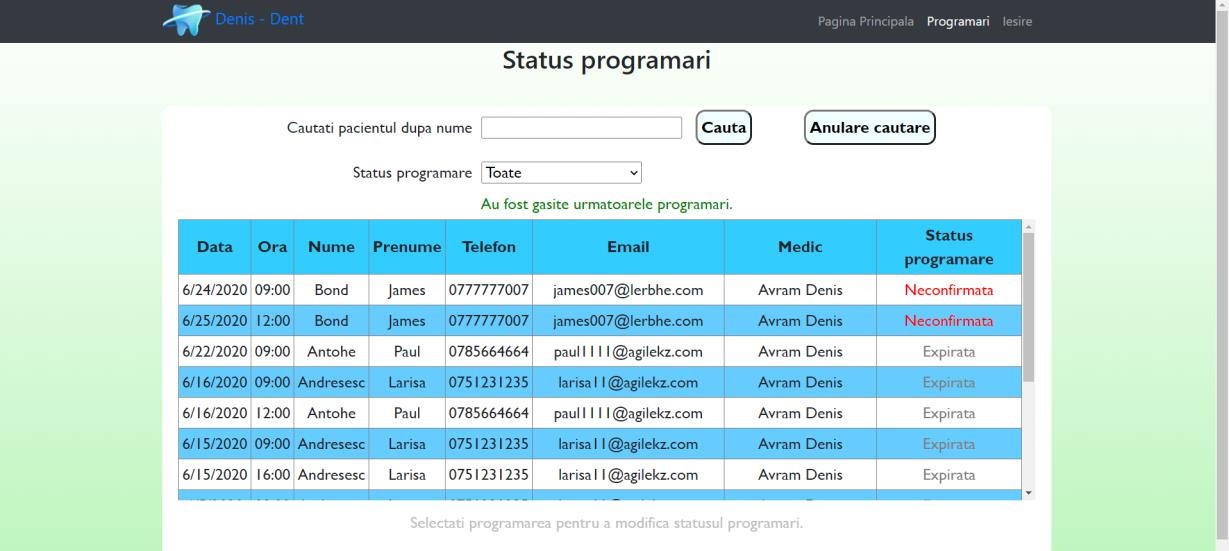


Figura 4.3.35 Pagina „Programari” (recepție)



Figura 4.3.36 Eroare la confirmarea programării deja expirate



Figura 4.3.37 Confirmarea programării a avut succes

### Interfața aplicației pentru medic

Pagina „Autentificare medic” (figura 4.3.38) reprezintă pagina de autentificare a medicului stomatolog. Pentru ca medicul să aibă acces la aplicație doar trebuie să se autentifice, nu mai trebuie să își creeze cont. Medici care sunt deja înregistrați în baza de date a cabinetului, vor fi încărcați automat, la accesarea pagini, în DropDownList-ul „Utilizator”. Cu ajutorul DropDownList-ului, se va putea selecta medicul care dorește să se autentifice, după care, în caseta de text pentru parolă se va introduce parola și la sfârșit se acționează butonul „Autentificare”. La acționarea butonului de autentificare pot exista mai multe scenarii:

1. Eroare la selectarea medicului:
   1. Nu s-a selectat utilizatorul – „Nu ati selectat utilizatorul.” (figura 4.3.39);
2. Erori la introducerea parolei;
3. Parola introdusă corespunde utilizatorului selectat și medicul va fi redirecționat la pagina principala a medicului (autentificarea efectuata cu succes).

În cazul în care medicul și-a uitat parola, poate da clic pe „Ati uitat parola?” și se va deschide mai jos un panou (CollapsiblePanelExtender) cu o casetă de text pentru email și un buton pentru trimiterea parolei. Pentru a recupera parola, în caseta de text trebuie introdus email-ul medicului, care s-a folosit pentru înregistrarea lui în cabinet și pe urmă trebuie apăsat butonul „Trimite”. La apăsarea butonului de trimitere pot rezulta două scenarii:

1. Erori la introducerea adresei email:
   1. Nu s-a introdus adresa email – „Nu ati introdus adresa e-mail.”;
   2. Adresa email introdusă este invalidă – „E-mail-ul poate sa contina litere, cifre, -, \_. Obligatoriu trebuie sa contina @nume provider.nume domain”;
   3. Adresa email introdusă nu este înregistrată, pentru nici un medic – „Adresa e-mail nu corespunde nici unui cont registrat la noi.”;
2. Adresa email este corectă, medicul va primi parola pe email
   1. Mesajul afișat – „Parola dumneavoastra a fost trimisa cu succes, pe e-mail-ul: `email- ul introdus`” (figura 4.3.40).



Figura 4.3.38 Pagina „Autentificare medic”

|  |  |
| --- | --- |
| Figura 4.3.39 Nu s-a selectat utilizatorul | Figura 4.3.40 Adresa email este corectă |

După autentificare, se ajunge la pagina „Pagina Principala” (figura 4.3.41). Dacă în sala de așteptare nu avem nici un pacient prezent se va afișa mesajul „Nici un pacient nu este prezent.” (figura 4.3.41), iar dacă în sala de așteptare avem pacienți se va afișa un tabel cu pacienți prezenți (figura 4.3.42). Pe lângă tabelul ce poate, sau nu, exista, pagina mai conține un buton de reîmprospătare și butoanele

„Adauga consultatie noua”, „Adauga radiografie noua”, „Vizualizare fisa pacient”, „Vizualizare consultatii pacient” și „Vizualizare radiografii”. Pentru activarea butoanelor prezentate anterior, pacientul trebuie să fie selectat (figura 4.3.42), deoarece la acționarea acestor butoane se vor încărca datele pacientului în sesiuni și se vor transmite în pagina aferentă butonului acționat.

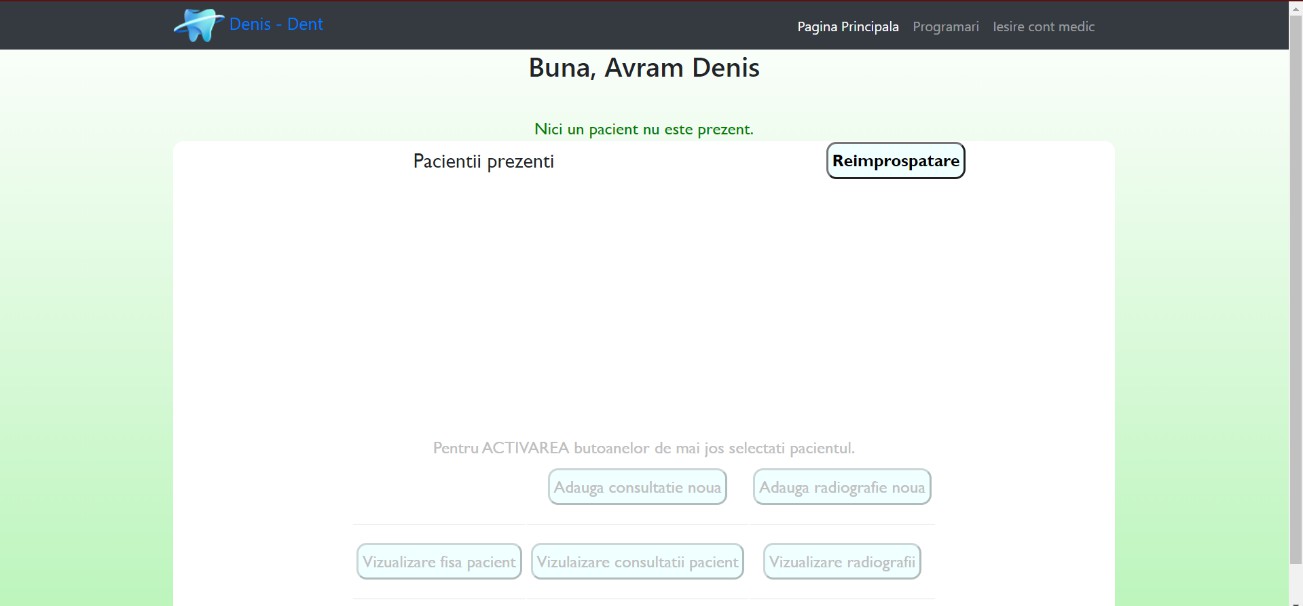


Figura 4.3.41 Pagina „Pagina Principala” (medic)

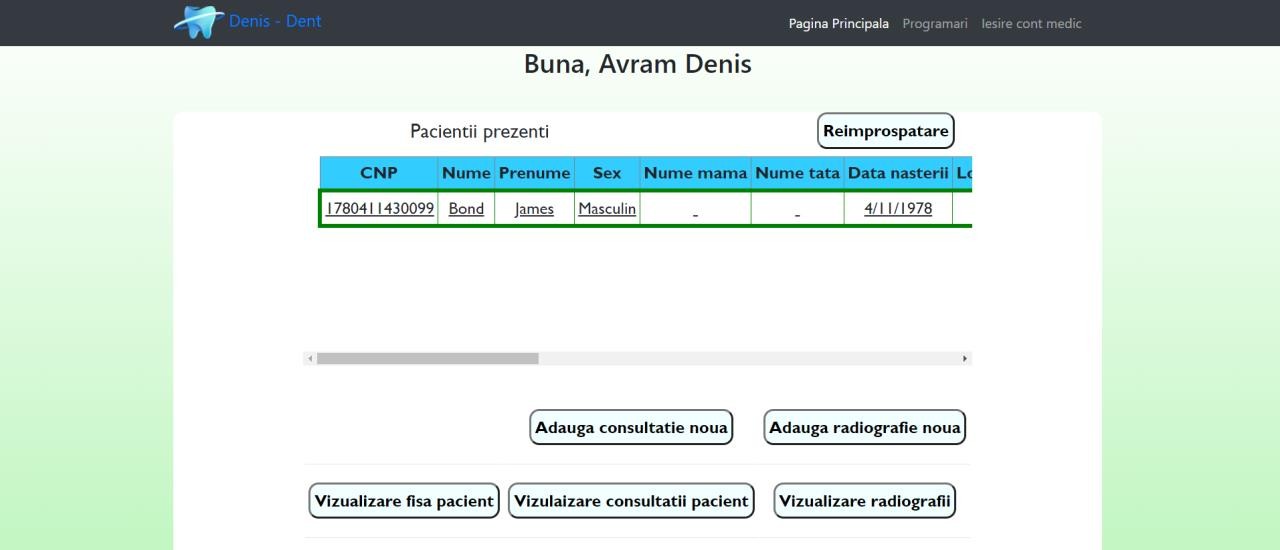


Figura 4.3.42 Pacientul prezent este selectat (butoanele active)

Pagina „Adaugare consultatie noua” (figura 4.3.43) este cea mai interactivă pagină din toată aplicația. Pentru ca medicul să creeze o consultație nouă doar trebuie să selecteze dintele sau dinți la care s-a lucrat, diagnosticele, intervențiile efectuate, planul de tratament și să acționeze butonul „Adauga consultatie noua”, iar restul datelor de care medicul are nevoie pentru a întocmi o consultație se vor completa automat. DropDwonList-ele care conțin clasele de diagnostice și clasele de intervenții, se vor încărca automat la accesarea acestei pagini, iar DDL-urile care conțin diagnosticele și intervențiile efectuate se încarcă în momentul selectării clasei de diagnostice, respectiv clasei de intervenții.

Pentru ca un diagnostic să fie adăugat la consultație trebuie acționat butonul „Adauga diagnostic”, iar după care se pot întâmpla următoarele scenarii:

* Nu s-a selectat dintele pentru care se adaugă diagnosticul – „Nu ati selectat dintele.” (figura 4.3.44);
* Nu s-a selectat clasa diagnosticului – „Nu ati selectat clasa diagnostice.”;
* Nu s-a selectat diagnosticul – „Nu ati selectat diagnosticul.”.;
* Diagnosticul a fost adăugat cu succes.

Pentru ca intervențiile să fie adăugate la consultație, se va acționa butonul „Adauga interventie” care poate declanșa diferite erori:

* Nu s-a selectat dintele;
* Nu s-a adăugat nici un diagnostic pentru dintele selectat – „Nu ati adaugat nici un diagnostic.” (figura 4.3.45);
* Nu s-a selectat clasa intervenției – „Nu ati selectat clasa intervenții.”;
* Nu s-a selectat intervenția efectuată – „Nu ati selectat intervenția.”.

Scenariile ce pot să apară după acționarea butonului de adăugare consultație sunt următoarele:

1. Eroare la adăugarea diagnosticului:
   1. Nu s-a adăugat diagnosticul – „Nu ati adaugat nici un diagnostic.”;
2. Eroare la adăugarea intervenției:
   1. Nu s-a adăugat intervenția efectuată – „Nu ati adaugat nici o interventie.” (figura 4.3.46);
3. Consultația a fost adăugată cu succes
   1. Mesajul afișat – „Consultatia a fost adugata cu succes. Pentru a revenii la pagina principala, dati click pe Pagina Principala din meniu.”;

Dacă medicul, după ce a completat consultația, dorește să vizualizeze cum va arăta consultația tipărită, poate bifa „Vizualizare format pentru printare.” înainte să acționeze butonul de adăugare a consultației.

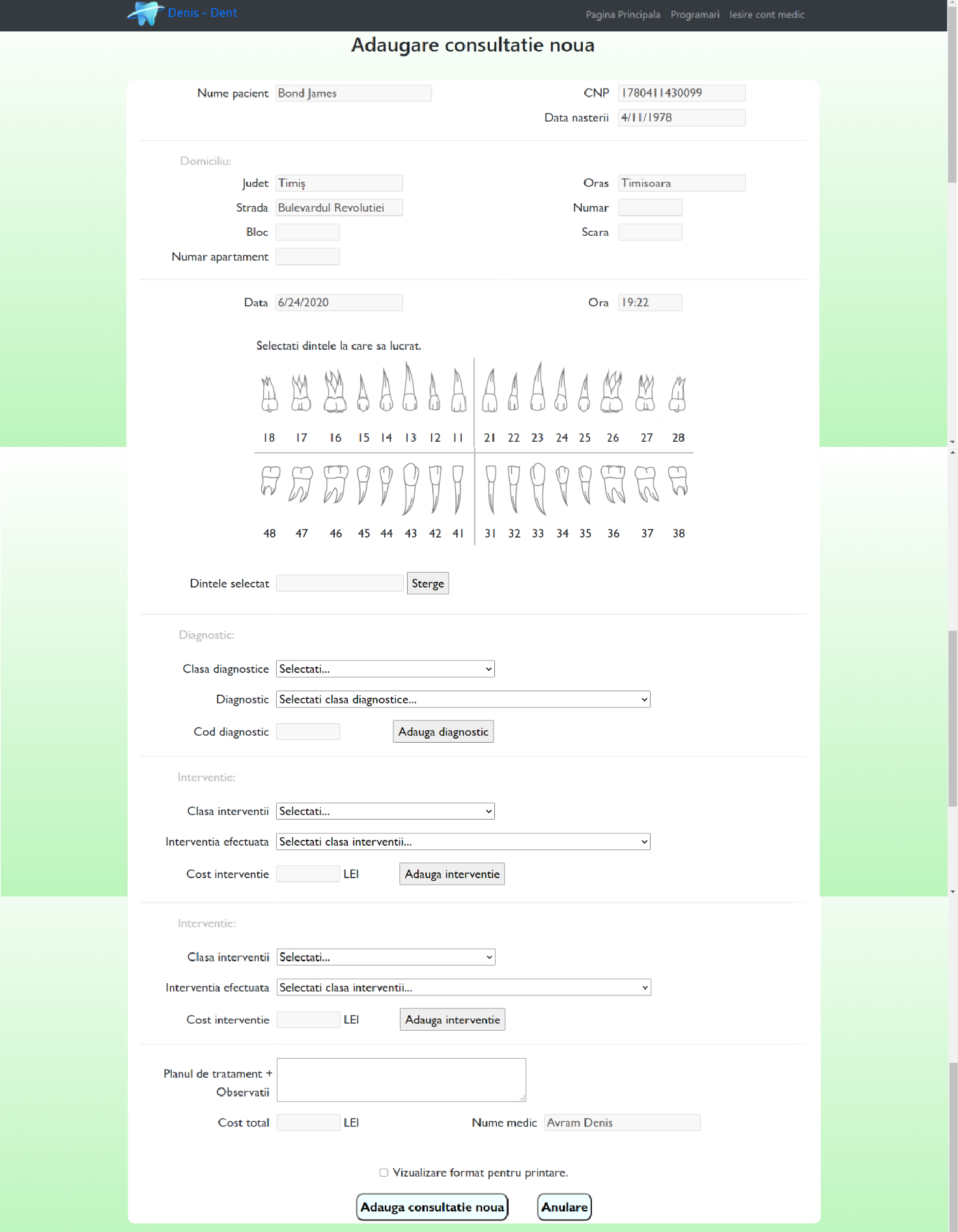


Figura 4.3.43 Pagina „Adaugare consultatie noua”

|  |  |
| --- | --- |
| Figura 4.3.44 Nu s-a selectat dintele pentru care se adaugă diagnosticul | Figura 4.3.45 Nu s-a adăugat nici un diagnostic pentru dintele selectat |

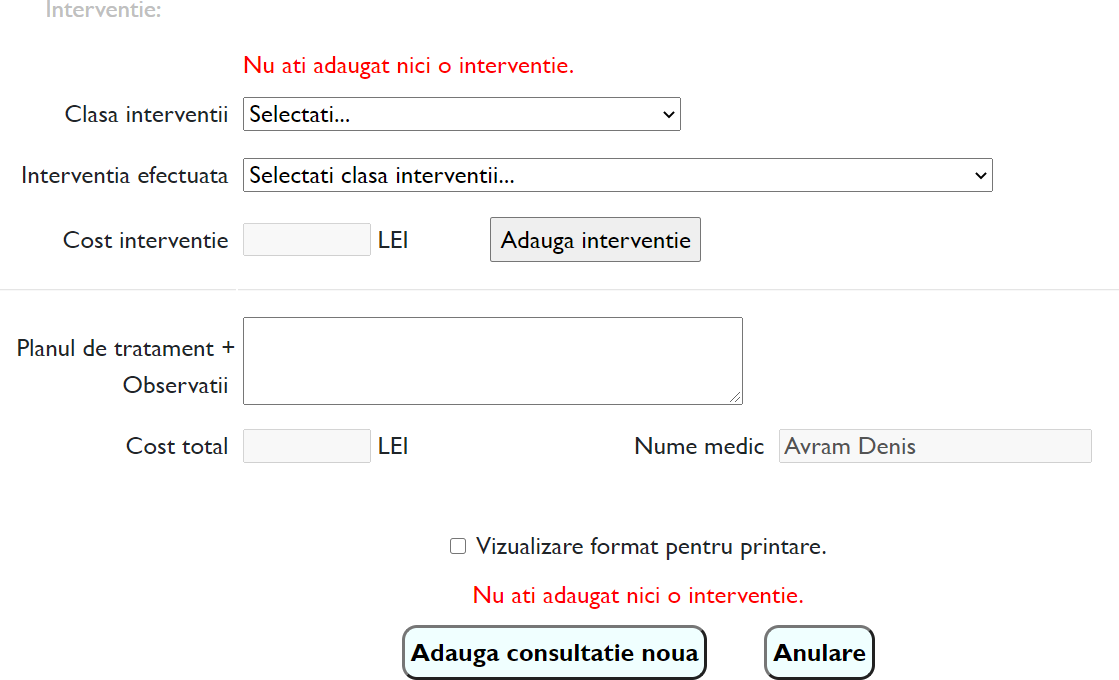


Figura 4.3.46 Nu s-a adăugat intervenția efectuată

Pagina „Adugare radiografie noua” (figura 4.3.47) reprezintă pagina pentru adăugarea radiografiilor pacientului, anterior solicitate de către medic. Ca medicul să încarce o radiografie, mai întâi trebuie selectată radiografia acționând butonul „Choose File”, apoi dat clic pe butonul „Incarca radiografia” si radiografia va fi încărcată in serverul Web. Dacă radiografia nu a fost selectată, dar butonul de încărcare a fost acționat va rezulta o eroare – „Nu ati selectat radiografia pentru a fi incarcata.”. După ce s-a încărcat radiografia, medicul o poate vizualiza cu un singur clic pe poza afișată într-un buton imagine (ImageButton). Pentru a adăuga radiografia în baza de date, nu e destul doar poza să fie încărcată, mai trebuie adăugat și un diagnostic. La fel ca și la consultații, DropDownList-ul care conține clasa de diagnostice, se încarcă automat la accesarea acestei pagini, iar DDL-ul care conține diagnosticele se încarcă în momentul selectări clasei de diagnostice.

Pentru ca diagnosticul să fie adăugat se va acționa butonul „Adauga diagnostic”, care poate activa diferite erori:

* Nu s-a selectat clasa diagnosticului – „Nu ati selectat clasa diagnostice.”;
* Nu s-a selectat diagnosticul – „Nu ati selectat diagnosticul.”;

Cele patru scenarii ce pot să apară după acționarea butonului de adăugare radiografie sunt următoarele:

1. Eroare la încărcarea radiografie:
   1. Nu s-a încărcat radiografia – „Nu ati incarcat radiografia.” (figura 4.3.48);
2. Eroare la adăugarea diagnosticului;
3. Eroare, radiografia se repetă:
   1. Radiografia încărcată a mai fost adăugată – „Radiografia a fost deja adaugata la persoana respectiva. O puteti vizualiza in sectiunea de `Vizualizare radiografii` din

`Pagina Principala`.” (figura 4.3.49);

1. Radiografia a fost adăugată cu succes
   1. Mesaj afișat – „Radiografia a fost adaugata cu succes. Pentru a revenii la pagina principala, dati click pe Pagina Principala din meniu.”;

Dacă medicul, după ce radiografia este pregătită de adăugat, dorește să se întoarcă automat (după adăugarea radiografie) înapoi pe pagina principală, poate bifa „Intoarcere automata la Pagina Principala. (dupa adugarea radiografiei)” înainte să acționeze butonul de adăugare radiografie.

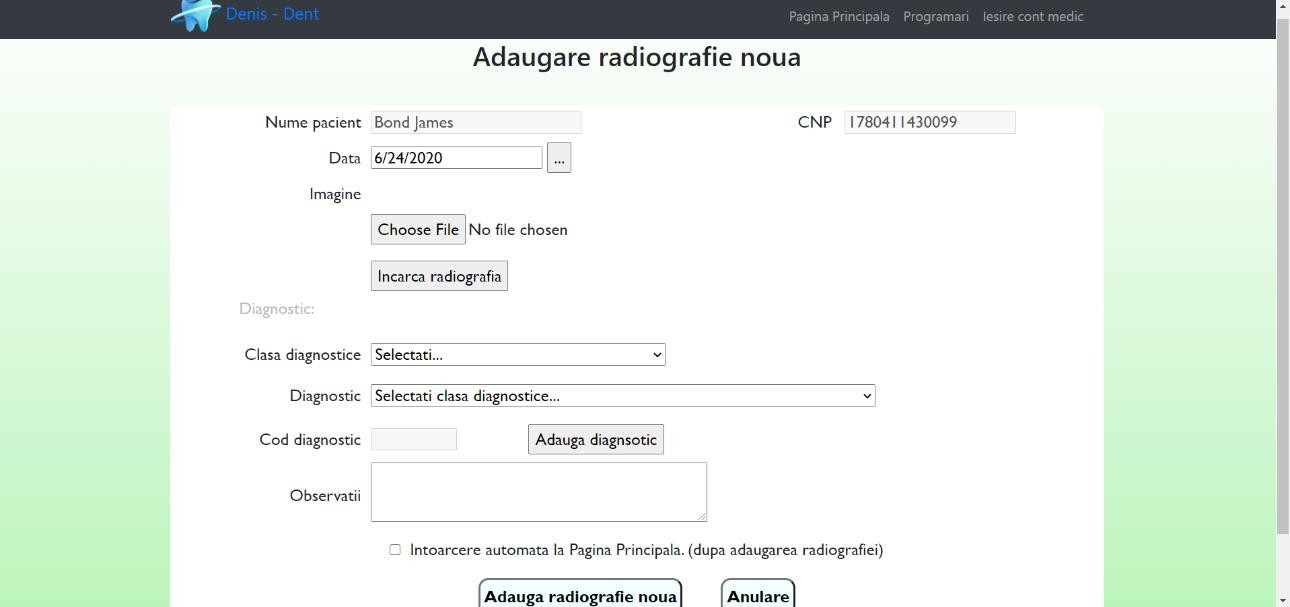


Figura 4.3.47 Pagina „Adugare radiografie noua”

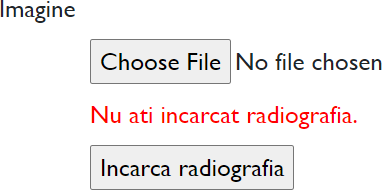


Figura 4.3.48 Nu s-a încărcat radiografia

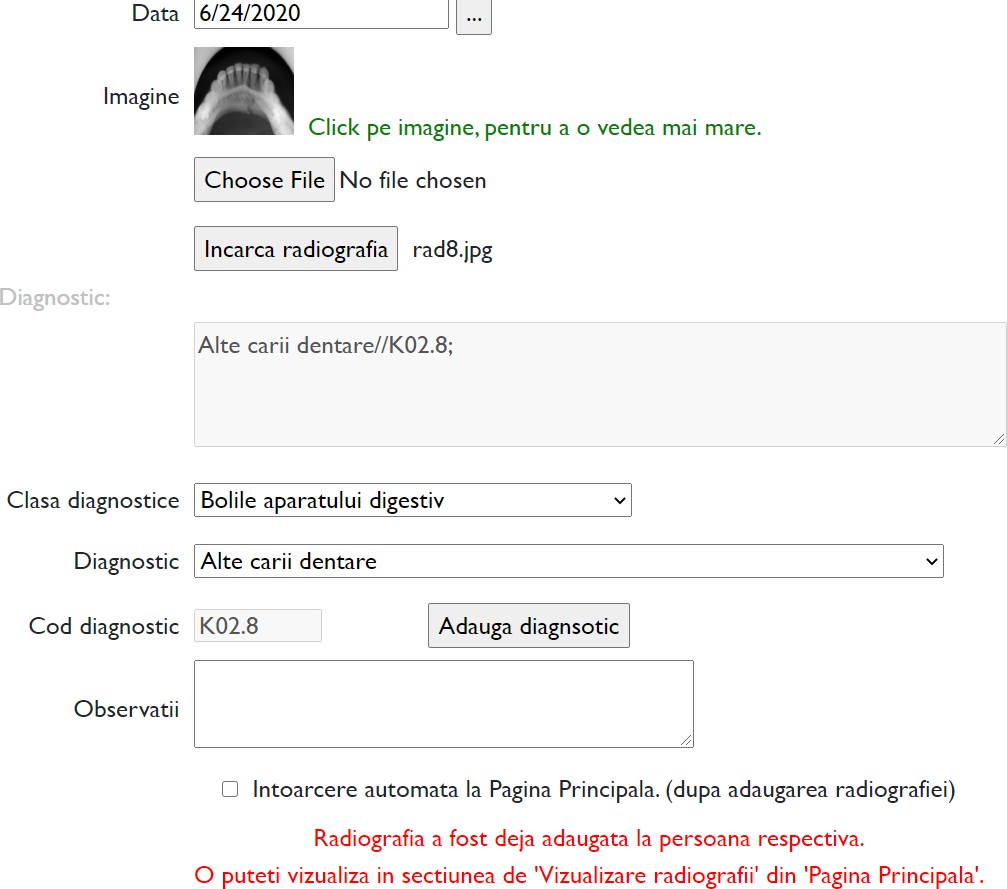


Figura 4.3.49 Radiografia încărcată a mai fost adăugată

Pagina „Vizualizare fisa pacient” (figura 4.3.50) conține toate datele legate de pacient, precum datele personale, datele de contact, domiciliu, medic familie, observații. Pe lângă faptul că medicul poate vizualiza mai ușor datele pacientului, dacă e nevoie poate și să le scoată la imprimantă, acționând butonul „Printare” care se află la sfârșitul pagini. Tot jos, în capeți din stânga și dreapta, avem și două HyperLink-uri, unul „Vizualizare consultatii” pentru accesul rapid la pagina de vizualizare a consultaților și celălalt „Vizualizare radiografii” pentru accesul rapid la pagina de vizualizare radiografii. Această pagină arată identic cu pagina „Vizualizare / Editare fisa pacient” din cadrul interfeței recepției, dar nu conține butoanele de editare a fișei.

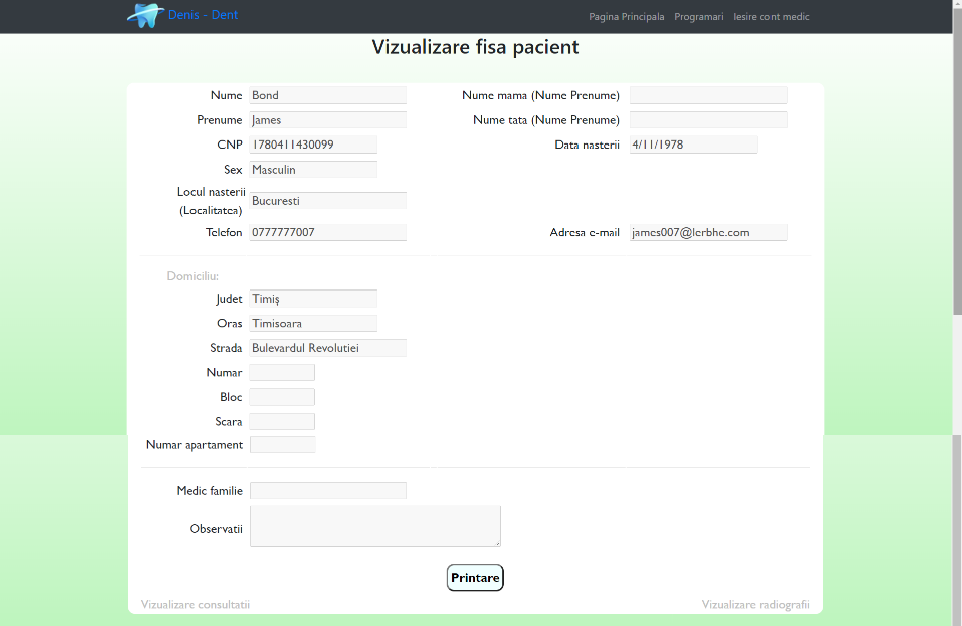


Figura 4.3.50 Pagina „Vizualizare fisa pacient”

Pagina „Vizualizare consultatii pacient” (figura 4.3.51) oferă posibilitatea medicului autentificat să își vadă toate consultațiile efectuate pentru pacientul selectat. Parte legată de date personale și domiciliu pacientului vor fi încărcate automat la accesare pagini, iar consultația care se dorește a vizualiza, va trebui selectată, ori după numarul ei, ori după data în care a fost efectuată. DropDownList-urile care conțin numărul consultațiilor și data consultațiilor, se încarcă automat la accesarea acestei pagini. Dacă se dorește tipărirea consultației, consultația mai întâi trebuie să fie selectată și apoi se acționează butonul „Printare”, dacă butonul a fost acționat, iar consultația nu a fost selectată se va afișa un mesaj de eroare – „Nu ati ales consultatia.”. Această pagină mai oferă și un HyperLink „Vizualizare radiografii” pentru accesul rapid la pagina de vizualizare radiografii.

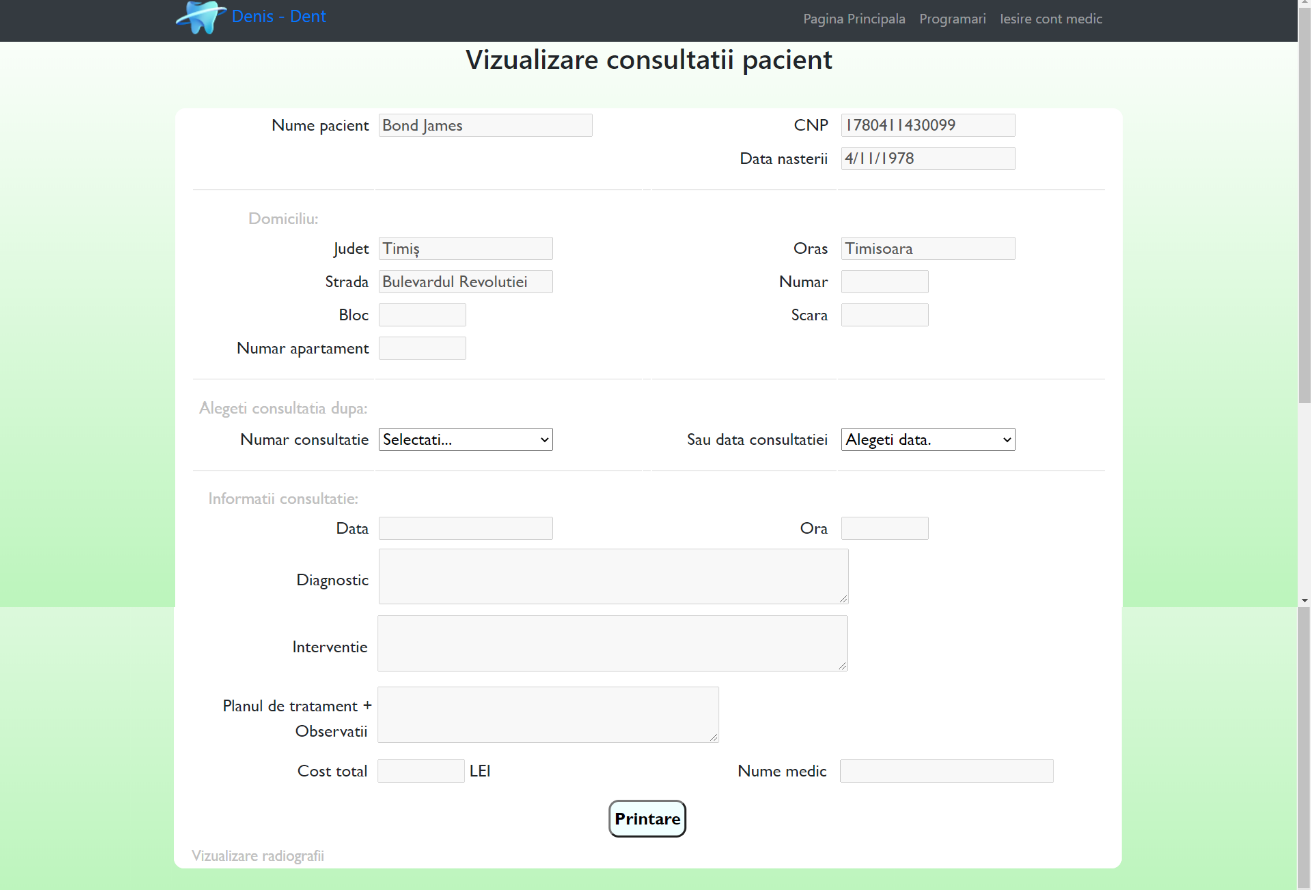


Figura 4.3.51 Pagina „Vizualizare consultatii pacient”

Pagina „Vizualizare radiografii” (figura 4.3.52) reprezintă pagina de vizualizare a radiografiilor adăugate pe parcurs, pentru pacientul respectiv. Dacă pacientul selectat nu are radiografii adăugate, se va afișa mesajul „Pacientul cu CNP-ul `CNP pacient` nu are nici o radiografie.”, iar dacă pacientul are radiografii adăugate ele vor fi afișate împreună cu mesajul „Pentru pacientul cu CNP-ul `CNP pacient` au fost gasite urmatoarele radiografii.” (figura 4.3.52). Radiografiile încărcate vor fi afișate ca ImageButton (butoane imagine), în format mai mic. La un singur clic asupra imagini, se vor afișa informațiile legate de acea radiografie, iar după al doilea clic pe aceeași imagine se va afișa imaginea în formatul original (ModalPopupExtender).

La fel ca și pagina prezentată anterior și această pagină oferă un HyperLink „Adauga radiografie noua” pentru accesul rapid la pagina de adăugare radiografii.

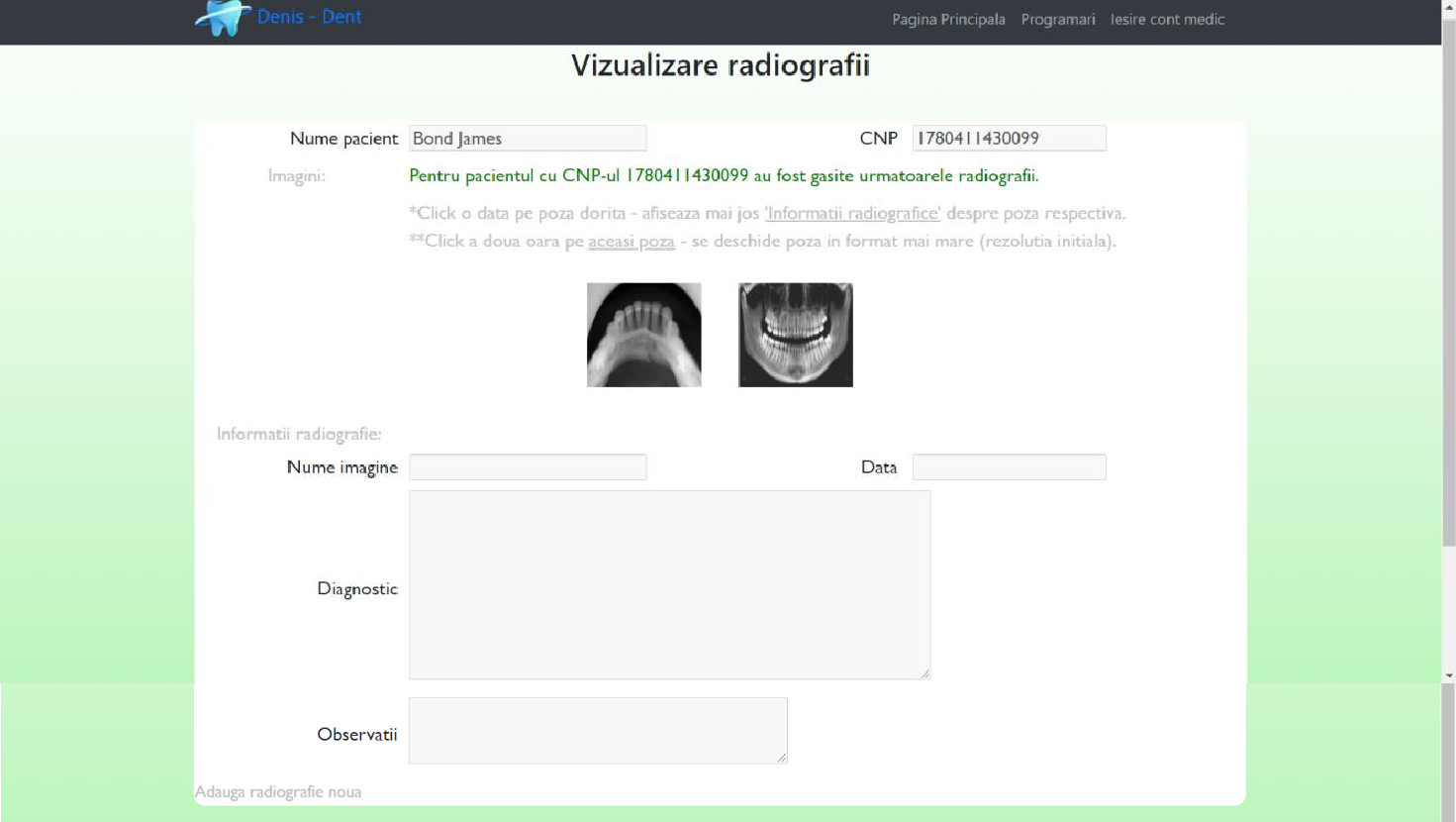


Figura 4.3.52 Pagina „Vizualizare radiografii”

Pagina „Programari” (figura 4.3.53) conține un tabel unde sunt afișate toate programările confirmate pentru medicul autentificat. Dacă medicul pentru ziua curentă nu are nici o programare, în loc de tabelul cu programări va fi afișat mesajul – „`nume medic`, pentru data de `data curenta` nu aveti nici o programare.”, iar dacă medicul are programări confirmate ele vor fi afișate împreună cu mesajul –

„`nume medic`, pentru data de `data curenta` aveti urmatoarele programari.” (figura 4.3.53). Această pagină pe lângă afișarea programărilor pentru ziua curentă, oferă si posibilitatea de a vizualiza programările pentru săptămâna curentă sau pentru o dată selectată manual de către medic (figura 4.3.54).

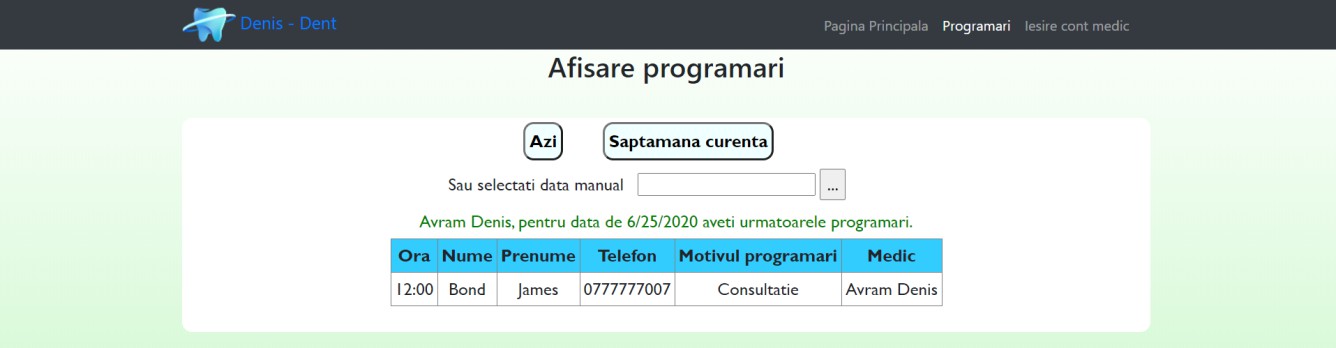


Figura 4.3.53 Pagina „Programari” (medic)



Figura 4.3.54 Data selectată manual de către medic

## Baza de date

Aplicația folosește o bază de date numită „CabinetStomatologic”, pentru stocarea și accesare informațiilor. Baza de date folosită este de tipul Microsoft SQL server și este compusă din mai multe tabele. Tabele cele mai importante care alcătuiesc această bază de date sunt următoarele:

* + - Tabela „Pacienti” conține datele personale ale pacientului, datele de contact, domiciliul și alte informații nu atât de importante. Cele mai importante câmpuri sunt:

|  |  |  |  |
| --- | --- | --- | --- |
| CNP | Nume | Prenume | Sex |
| Data nașterii | Localitatea nașterii | Județul (domiciliu) | Orașul (domiciliu) |
| Strada (domiciliu) | Telefon | Email |  |

* + - Tabela „Medici” conține datele personale ale medicului, împreună cu parola necesară pentru autentificarea sa și email-ul necesar pentru recuperarea parolei. Cele mai importante câmpuri sunt:

|  |  |  |  |
| --- | --- | --- | --- |
| IdMedic | Nume | Prenume | Parolă |
| Email |  |  |  |

* + - Tabela „ContPacineti”, conține datele necesare pentru înregistrarea utilizatorului (pacientului). Cele mai importante câmpuri sunt:

|  |  |  |  |
| --- | --- | --- | --- |
| CNP | Nume | Prenume | Telefon |
| Email | Parolă |  |  |

* + - Tabela „Programari”, aici sunt stocate toate programările solicitate, confirmate, neconfirmate, expirate. Cele mai importante câmpuri sunt:

|  |  |  |  |
| --- | --- | --- | --- |
| IdProgramare | CNP pacient | Nume pacient | Prenume pacient |

|  |  |  |  |
| --- | --- | --- | --- |
| Telefon | Email | Nume medic | Motivul programări |
| Data programări | Ora programări | Statusul programări |  |

* + - Tabela „Consultatii”, toate consultațiile efectuate vor fi stocate în această tabelă. Cele mai importante câmpuri sunt:

|  |  |  |  |
| --- | --- | --- | --- |
| IdConsultatie | CNP pacient | Nume pacient | Data nașterii |
| Data | Ora | Diagnostic | Intervenții |
| Nume medic | Costul total |  |  |

* + - Tabela „Radiografii” conține toate radiografiile pacienților care au fost solicitate de medicii stomatologi din cabinet. Cele mai importante câmpuri sunt:

|  |  |  |  |
| --- | --- | --- | --- |
| Id | CNP | Nume pacient | Data |
| Nume radiografie | Diagnostic |  |  |

* + - Tabela „Chitanțe” conține toate chitanțele care sau emis pacienților pentru serviciile oferite în cabinet. Cele mai importante câmpuri sunt:

|  |  |  |  |
| --- | --- | --- | --- |
| IdChitanta | IdConsultatie | CNP pacient | Nume pacient |
| Data | Ora | Suma achitată pana în data de azi | Suma achitată azi |
| Costul total |  |  |  |

* + - Tabela „ClasaDiagnostice” aici sunt stocate toate clasele diagnosticelor de care medici stomatologi au nevoie. Cele mai importante câmpuri sunt:

|  |  |
| --- | --- |
| Grupa | Numele clasei |

* + - Pentru fiecare clasă de diagnostic din „ClasaDiagnostice”, există câte un tabel pentru diagnosticele clasei, denumit în funcție de câmpul „Grupa” dint tabelul părinte. Cele mai importante câmpuri sunt:

|  |  |
| --- | --- |
| Cod diagnostic | Nume diagnostic |

## Comunicarea cu alte sisteme

Ca aplicația să își îndeplinească funcția pentru care e creată, are nevoie de o unitate centrală de prelucrare și stocare a datelor acumulate pe parcurs, de legături de tip Ethernet și acces la rețeaua Internet.

## Tipărirea la imprimantă

La imprimantă pot fi tipărite fișa pacientului (figura 4.6.1), consultațiile (figura 4.6.2) și chitanțele (figura 4.6.3). Pentru ca fișa pacientului să fie tipărită, trebuie accesată pagina „Vizualizare / Editare fisa pacient” din cadrul interfeței recepției sau pagina „Vizualizare fisa pacient” din cadrul interfeței medicului și acționat butonul „Printare”.

Pentru tipărirea consultației se va accesa pagina „Vizualizare consultatii” din cadrul interfeței medicului, se va selecta consultația pentru care se dorește tipărirea și se va acționa butonul „Printare”. Tipărirea consultației se mai poate face și imediat după ce s-a completat, dacă înainte de acționarea butonul „Adauga consultație noua” s-a bifat opțiunea „Vizualizare format pentru printare”.

Pentru ca o chitanța să fie tipărită, se va accesa pagina „Tiparire chitanta” din cadrul interfeței recepției, se va selecta consultația care se dorește achita, se introduce suma achitată, în momentul de față, și se acționează butanol „Tipareste chitanta”.

|  |  |
| --- | --- |
| Figura 4.6.1 Fișa pacientului tipărită la imprimantă | Figura 4.6.2 Consultația tipărită la imprimantă |

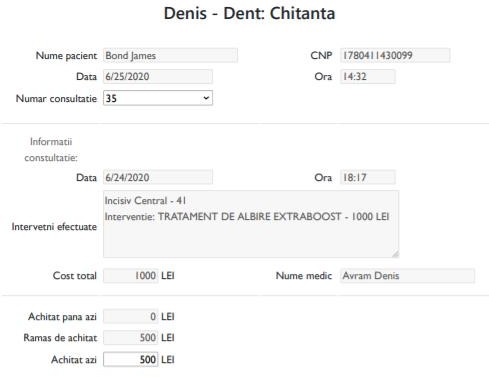


Figura 4.6.3 Chitanța tipărită la imprimantă

# Capitolul 5. Proiectarea aplicației de gestionare a pacienților dintr-un cabinet stomatologic

## Implementarea aplicației

Vor fi prezentate funcțiile (cod și explicații) care au o importanță semnificativă în dezvoltarea acestei aplicații.

### Recuperarea parolei - pacient

Această funcție este utilizată pentru interfața pacientului, în momentul autentificări. Dacă pacientul și-a uitat parola poate apela ajutorul acestei funcții (acționând pe „Ati uitat parola?”), care solicită introducerea CNP-ului și email-ul pacientului folosit la înregistrare. Funcția se va apela atunci când pacientul face clic pe butonul de trimitere. La acționarea acestui buton se va deschide baza de date și în tabelul „ContPacienti” se va căuta parola CNP-ului introdus, iar dacă CNP-ul nu a fost găsit se va afișa un mesaj de eroare (se iasă din funcție). După aceea, dacă CNP-ul s-a găsit în tabelul

„ContPacienti”, se face încă o verificare, dacă email-ul introdus corespunde CNP-ului. Dacă acesta nu corespunde, se va afișa un mesaj de eroare (se iasă din funcție), iar dacă email-ul corespunde CNP- ului se va continua cu pregătirea mail-ului ce va fi trimis utilizatorului.

Dacă înainte de acționarea butonului de trimitere, câmpurile CNP și adresa email nu sunt completate, funcția va afișa mesaj de eroare.

SqlConnection con;

string conString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=e:\An 3 - Sem 2\Licenta\CabStomatologic\CabStomatologic\App\_Data\CabinetStomatologic.mdf;Integrated Security=True";

Regex testNum = new Regex("^[0-9]+$");

Regex testEmail = new Regex("^[a-zA-Z\_.\'-'0-9]+[@][a-zA-Z\_\'-'0-9]+[.][a-zA-Z\_\'-'0-9]+$"); protected void trimiteEmailBtn\_Click(object sender, EventArgs e)

{

try

{

con = new SqlConnection(conString); con.Open();

if (cnpEmailTB.Text.Trim() != "")

{

if (testNum.IsMatch(cnpEmailTB.Text.Trim()))

{

if (cnpEmailTB.Text.Trim().Length == 13)

{

string stmt = "select Parola from ContPacienti where CNP='" + cnpEmailTB.Text.Trim() + "'";

SqlCommand sc = new SqlCommand(stmt, con); SqlDataReader dr = sc.ExecuteReader();

if (!dr.HasRows)

{

LabelCnpErr.ForeColor = Color.Red;

LabelCnpErr.Text = "CNP-ul introdus nu are asociat nici un cont.";

LabelParolaErr.Text = ""; LabelRezultatAutentificareErr.Text = "";

}

else

{

dr.Close();

sc.Dispose();

if (emailTB.Text.Trim() != "")

{

if (testEmail.IsMatch(emailTB.Text.Trim()))

{

stmt = "select Parola from ContPacienti where CNP='" + cnpEmailTB.Text.Trim() + "' and Email='" + emailTB.Text.Trim() + "'";

sc = new SqlCommand(stmt, con); dr = sc.ExecuteReader();

if (!dr.Read())

{

LabelTrimiteEmailErr.ForeColor = Color.Red;

LabelTrimiteEmailErr.Text = "Adresa e-mail nu corespunde cu CNP-ul " + cnpEmailTB.Text.Trim();

LabelCnpEmailErr.Text = "";

}

else

{

string parola = dr[0].ToString().Trim();

MailAddress to = new MailAddress(emailTB.Text.Trim()); MailAddress from = new

MailAddress("[denisavram121@gmail.com](mailto:denisavram121@gmail.com)", "Denis - Dent");

MailMessage message = new MailMessage(from, to); message.Subject = "Cerere parola";

message.Body = "CNP utilizator: " + cnpEmailTB.Text.Trim() + "\nParola d-voastra: " + parola;

SmtpClient client = new SmtpClient("smtp.gmail.com",

587)

{

Credentials = new

NetworkCredential("[denisavram121@gmail.com](mailto:denisavram121@gmail.com)", "uovcrgrikfmxzvws"),

EnableSsl = true

};

try

{

client.Send(message); LabelTrimiteEmailErr.ForeColor = Color.Green; LabelTrimiteEmailErr.Text = "Parola dumneavoastra a

fost trimisa cu succes, pe e-email-ul: " + emailTB.Text.Trim();

LabelCnpEmailErr.Text = "";

}

catch (SmtpException ex)

{

LabelTrimiteEmailErr.ForeColor = Color.Red;

LabelTrimiteEmailErr.Text = "Nu se poate realiza conexiunea la serverul SMTP: " + ex.Message;

LabelCnpEmailErr.Text = "";

}

}

dr.Close();

sc.Dispose();

}

else

{

LabelTrimiteEmailErr.ForeColor = Color.Red;

LabelTrimiteEmailErr.Text = "E-mail-ul poate sa contina litere, cifre, -, \_.<br/>Obligatoriu trebuie sa contina @nume provider.nume domain";

LabelCnpEmailErr.Text = "";

}

}

else

{

LabelTrimiteEmailErr.ForeColor = Color.Red; LabelTrimiteEmailErr.Text = "Nu ati introdus adresa e-mail."; LabelCnpEmailErr.Text = "";

}

}

dr.Close();

sc.Dispose();

}

else

{

}

}

else

{

LabelCnpEmailErr.ForeColor = Color.Red;

LabelCnpEmailErr.Text = "CNP-ul trebuie sa contina 13 cifre."; LabelTrimiteEmailErr.Text = "";

}

}

else

{

LabelCnpEmailErr.ForeColor = Color.Red;

LabelCnpEmailErr.Text = "CNP-ul nu are voie sa contina litere."; LabelTrimiteEmailErr.Text = "";

LabelCnpEmailErr.ForeColor = Color.Red; LabelCnpEmailErr.Text = "Nu ati introuds CNP-ul."; LabelTrimiteEmailErr.Text = "";

}

}

catch (Exception ex)

{

LabelTrimiteEmailErr.ForeColor = Color.Red;

LabelTrimiteEmailErr.Text = "Nu se poate realiza conexiunea la baza de date: " + ex.Message;

LabelCnpEmailErr.Text = "";

}

finally

{

con.Close();

}

}

### Autocompletarea datei de naștere și sexul după CNP-ul introdus la adăugarea de pacient nou (interfața recepției)

Această funcție demonstrează faptul că s-a încercat cât mai mult să se ușureze munca personalului cabinetului. Funcția operează în două locuri, într-un loc vizibil pe pagina de adăugare pacient nou, din interfața recepției și într-un loc invizibil pentru utilizator, în pagina de înregistrare a pacientului (interfața pacientului). În pagina de înregistrare cont nou pentru pacient, această funcție se folosește de CNP-ul pacientului să calculeze data nașterii și sexul, pentru a le introduce în baza de date, ca mai

târziu să accelereze procesul de adăugare (se introduce CNP-ul pacientului în pagina de adăugare pacient nou și toate datele de care dispune aplicația, acumulate de la înregistrarea pacientului, vor fi imediat completate în fișa pacientului – da, funcția pentru autocompletarea a două câmpuri, ajunge să completeze aproape toate datele personale și datele de contact ale pacientului).

În pagina de adăugare pacient nou, activitatea acestei funcții este vizibilă utilizatorului și ea poate acționa în două moduri diferite:

* + 1. Dacă pacientul înainte de prezența sa la cabinet și-a făcut cont ca utilizator pacient, această funcție pe lângă autocompletarea datei de naștere și sexului, va completa și numele, prenumele, numărul de telefon și email-ul;
    2. Dacă pacientul înainte de prezența sa la cabinet nu și-a făcut cont pentru utilizator pacient, această funcție va completa doar data nașterii și sexul.

Pentru ca această funcție să funcționeze s-a încercat improvizarea funcției „TextBox\_Leave” care nu există în ASP.Net, dar s-a ajuns la aceeași efect utilizând „TextBox\_TextChanged” împreună cu

„AutoPostBack=`true`” asupra casetei de text pentru CNP.

SqlConnection con;

string conString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=e:\An 3 - Sem 2\Licenta\CabStomatologic\CabStomatologic\App\_Data\CabinetStomatologic.mdf;Integrated Security=True";

static Boolean testCNP = false; //pentru validare inainte de apasarea butonului „Adauga pacient nou”

protected void cnpTB\_TextChanged(object sender, EventArgs e)

{

Regex testNum = new Regex("^[0-9]+$"); try

{

if (cnpTB.Text.Trim() != "")

{

if (testNum.IsMatch(cnpTB.Text.Trim()))

{

if (cnpTB.Text.Trim().Length == 13)

{

if (verificCNP(cnpTB.Text.Trim()) == true) //verificam daca CNP-ul este adevarat cu funtia ce va fi prezentata mai jos

{

null, final = null;

decembrie 1999

string a1 = null, a2 = null, l1 = null, l2 = null, z1 = null, z2 =

string cnp = cnpTB.Text.Trim(); string sex = cnp.Substring(0, 1);

if (sex == "1" || sex == "2") //nascuti intre 1 ianuarie 1900 si 31

{

a1 = cnp.Substring(1, 1); a2 = cnp.Substring(2, 1); l1 = cnp.Substring(3, 1); l2 = cnp.Substring(4, 1); z1 = cnp.Substring(5, 1); z2 = cnp.Substring(6, 1); if (l1 == "0" && z1 == "0")

{

final = l2 + "/" + z2 + "/19" + a1 + a2;

}

else if (l1 == "0" && Convert.ToInt16(z1) != 0)

{

final = l2 + "/" + z1 + z2 + "/19" + a1 + a2;

}

else if (l1 == "1" && z1 == "0")

{

final = l1 + l2 + "/" + z2 + "/19" + a1 + a2;

}

else if (l1 == "1" && Convert.ToInt16(z1) != 0)

{

final = l1 + l2 + "/" + z1 + z2 + "/19" + a1 + a2;

}

dataNasteriiTB.Text = final; if (sex == "1")

{

sexDDL.SelectedIndex = 1;

}

else if (sex == "2")

{

sexDDL.SelectedIndex = 2;

}

LabelDataNErr.Text = ""; LabelCnpErr.Text = ""; testCNP = true;

}

si 31 decembrie 2099

else if (sex == "5" || sex == "6") //nascuti intre 1 ianuarie 2000

{

a1 = cnp.Substring(1, 1); a2 = cnp.Substring(2, 1); l1 = cnp.Substring(3, 1); l2 = cnp.Substring(4, 1); z1 = cnp.Substring(5, 1); z2 = cnp.Substring(6, 1); if (l1 == "0" && z1 == "0")

{

final = l2 + "/" + z2 + "/20" + a1 + a2;

}

else if (l1 == "0" && Convert.ToInt16(z1) != 0)

{

final = l2 + "/" + z1 + z2 + "/20" + a1 + a2;

}

else if (l1 == "1" && z1 == "0")

{

final = l1 + l2 + "/" + z2 + "/20" + a1 + a2;

}

else if (l1 == "1" && Convert.ToInt16(z1) != 0)

{

final = l1 + l2 + "/" + z1 + z2 + "/20" + a1 + a2;

}

LabelDataNErr.Text = ""; LabelCnpErr.Text = "";

dataNasteriiTB.Text = final; if (sex == "5")

{

sexDDL.SelectedIndex = 1;

}

else if (sex == "6")

{

sexDDL.SelectedIndex = 2;

}

testCNP = true;

}

else //pentru rezidenti si ani 1800 si 1899

{

manual.";

LabelDataNErr.ForeColor = Color.Green;

LabelDataNErr.Text = "Va rugam sa introduceti data nasteri

LabelCnpErr.Text = "";

if (sex == "3" || sex == "7")

{

sexDDL.SelectedIndex = 1;

}

else if (sex == "4" || sex == "8")

{

sexDDL.SelectedIndex = 2;

}

testCNP = true;

}

//partea daca pacientul are cont se vor completa automat si restul campurilor

try

{

con = new SqlConnection(conString); con.Open();

if (!string.IsNullOrEmpty(receptie))

{

string stmt = "select Nume, Prenume, Telefon, Email from ContPacienti where CNP='" + cnpTB.Text.Trim() + "'";

SqlCommand sc = new SqlCommand(stmt, con); SqlDataReader dr = sc.ExecuteReader();

if (dr.Read())

{

numeTB.Text = dr["Nume"].ToString().Trim(); prenumeTB.Text = dr["Prenume"].ToString().Trim(); telefonTB.Text = dr["Telefon"].ToString().Trim(); emailTB.Text = dr["Email"].ToString().Trim();

}

dr.Close();

sc.Dispose(); Label22.Text = "";

}

else

{

Label22.ForeColor = Color.Red;

Label22.Text = "Eroare la transmiterea datelor utilizand sesiunin. Receptia nu este logata.";

}

}

catch (SqlException ex)

{

" + ex.Message;

Label22.ForeColor = Color.Red;

Label22.Text = "Nu se poate realiza conexiunea la baza de date:

}

finally

{

con.Close();

}

}

else

{

LabelCnpErr.ForeColor = Color.Red;

LabelCnpErr.Text = "CNP-ul introdus este inexistent.";

testCNP = false;

}

}

else

{

}

}

else

{

LabelCnpErr.ForeColor = Color.Red;

LabelCnpErr.Text = "CNP-ul trebuie sa contina 13 cifre."; testCNP = false;

}

}

else

{

LabelCnpErr.ForeColor = Color.Red;

LabelCnpErr.Text = "CNP-ul nu are voie sa contina litere."; testCNP = false;

LabelCnpErr.ForeColor = Color.Red; LabelCnpErr.Text = "Nu ati introuds CNP-ul."; testCNP = false;

}

}

catch (FormatException ex)

{

LabelCnpErr.ForeColor = Color.Red;

LabelCnpErr.Text = "Nu se poate calcula data nasterii din CNP-ul introdus: " + ex.Message;

}

}

### Validarea (verificarea) CNP-ului

Această funcție are scopul verificări dacă CNP-ul pacientului, care vine introdus în baza de date a aplicației este valid, există. Se ia CNP-ul introdus în caseta de text pentru CNP și se vor face următori pași:

1. Se descompune CNP-ul în 13 cifre;
2. Se calculează cifra de control după următorul șablon – fiecare cifră din CNP este înmulțită cu cifra de pe aceeași poziție din numărul 279146358279, rezultatele se adună, iar rezultatul final este împărțit cu rest la 11. Dacă restul este 10, atunci cifra de control este 1, altfel cifra de control este egală cu restul [8];
3. Se verifică dacă cifra de control calculată este egală cu cifra de control din CNP, dacă sunt egale, CNP-ul este valid, dacă nu sunt egale CNP-ul este greșit.

public static bool verificCNP(string cnp)

{

int s, a1, a2, l1, l2, z1, z2, j1, j2, n1, n2, n3, cifc, u; s = Convert.ToInt16(cnp.Substring(0, 1));

a1 = Convert.ToInt16(cnp.Substring(1, 1)); a2 = Convert.ToInt16(cnp.Substring(2, 1)); l1 = Convert.ToInt16(cnp.Substring(3, 1));

l2 = Convert.ToInt16(cnp.Substring(4, 1)); z1 = Convert.ToInt16(cnp.Substring(5, 1)); z2 = Convert.ToInt16(cnp.Substring(6, 1)); j1 = Convert.ToInt16(cnp.Substring(7, 1)); j2 = Convert.ToInt16(cnp.Substring(8, 1)); n1 = Convert.ToInt16(cnp.Substring(9, 1)); n2 = Convert.ToInt16(cnp.Substring(10, 1)); n3 = Convert.ToInt16(cnp.Substring(11, 1));

cifc = Convert.ToInt16(((s \* 2 + a1 \* 7 + a2 \* 9 + l1 \* 1 + l2 \* 4 + z1 \* 6 + z2 \* 3 + j1 \* 5 + j2 \* 8 + n1 \* 2 + n2 \* 7 + n3 \* 9) % 11));

if (cifc == 10)

{

cifc = 1;

}

u = Convert.ToInt16(cnp.Substring(12, 1)); if (cifc == u)

{

}

else

{

}

}

return true;

return false;

### Selectarea programări a cărei status a fost modificat

Această funcție se folosește pe pagina de „Programari” din cadrul interfeței recepției. Programările afișate sunt ordonate în ordinea statusului programări, confirmate, neconfirmate, expirate. După ce s- a modificat statusul unei programări, aceasta își va schimba poziția în tabelul afișat și ca selectul să nu rămână pe vechia poziție se apelează această funcție.

Funcția parcurge fiecare rând a tabelului, coloană cu id-ul programări (coloana invizibilă pentru utilizator) pentru a găsi programarea care a fost modificată.

protected void FunctieChangeSelect(string id) // functia selecteaza inregistrarea din gridview dupa ce sia schimbat pozitia dupa modificarea statusului

{

int index = -1;

foreach (GridViewRow row in GridView1.Rows)

{

if (row.Cells[8].Text.ToString().Trim() == id)

{

index = row.RowIndex; break;

}

}

GridView1.SelectedIndex = index;

}

### Afișarea consultațiilor neachitate

Această funcție este utilizată pentru încărcarea consultațiilor neachitate în tabelul afișat pe pagina

„Consultații neachitate” (interfața recepției). Funcția se va apela atunci când recepția accesează această pagină. Înainte ca funcția să se apeleze se va deschide baza de date. Prima dată se vor citi toate consultațiile din tabelul „Consultatii” și apoi se vor citi chitanțele din tabelul „Chitante”, pentru consultațiile care au fost achitate. Înainte să parcurgem ambele rezultate, vom crea un tabel de tip DataTable care va clona coloanele (numele coloanelor) din tabelul consultatii. Următorul pas este parcurgerea ambelor tabele, pentru a se găsi consultația achitată, care va fi ignorată și restul consultațiilor se vor adăuga în tabelul precedent creat. După apelul funcției baza de date se va închide.

protected void IncarcareGridView() //functia care incarca consultatiile neachitate in GridView

{

string stmt = "select IdConsultatie as [Numar consultatie], CNP, NumePacient as [Nume pacient], convert(varchar(20),Data,101) as [Data], Ora, Interventii, Medic as [Nume medic], CostTotal as [Cost total] from Consultatii order by NumePacient, Data DESC";

SqlDataAdapter da = new SqlDataAdapter(stmt, con); DataSet ds = new DataSet();

da.Fill(ds, "Consultatii");

DataRow[] consultatii = ds.Tables["Consultatii"].Select(); //toate consultatiile

stmt = "select IdConsultatie, SumaTotalAchitata, CostTotal from Chitante where CostTotal=SumaTotalAchitata";

SqlDataAdapter da2 = new SqlDataAdapter(stmt, con); DataSet ds2 = new DataSet();

da2.Fill(ds2, "Chitante");

DataRow[] achitat = ds2.Tables["Chitante"].Select(); //consultatile achitate

DataTable table = ds.Tables["Consultatii"].Clone(); string idConsAchitata = "";

foreach (DataRow cons in consultatii)

{

foreach (DataRow achit in achitat)

{

if (cons["Numar consultatie"].ToString().Trim() == achit["IdConsultatie"].ToString().Trim())

{

idConsAchitata = cons["Numar consultatie"].ToString().Trim(); //salvam consultatia achitata deja

}

}

if (cons["Numar consultatie"].ToString().Trim() != idConsAchitata) //daca consultatia nu este achitata o adaugam intrun tabel

{

table.Rows.Add(cons.ItemArray);

}

}

foreach (DataRow row in table.Rows) //cu asta am modificat cum sa arate coloana interventii

{

row.SetField<string>("Interventii",

row.Field<string>("Interventii").Replace(";", "\n").Replace("\_", "\nInterventie: ").Replace("//", " - ").Replace(".", " "));

}

GridView1.DataSource = table; GridView1.DataBind();

LabelPrincipalErr.Text = "";

table.Dispose(); ds2.Dispose(); da2.Dispose(); ds.Dispose();

da.Dispose();

}

### Transferul de date între pagini

Pentru transferul de date între pagini am folosit sesiuni, din cauza faptului că pot stoca valoarea primită, pe server, și datele sunt stocate separat pentru fiecare utilizator. De exemplu transferul de CNP și nume pacient din pagina principală a medicului, în pagina de adăugare radiografii.

Codul care încarcă datele în sesiuni:

Session["numePacient"] = GridView1.SelectedRow.Cells[1].Text.ToString().Trim() + " " + GridView1.SelectedRow.Cells[2].Text.ToString().Trim();

Session["cnp"] = GridView1.SelectedRow.Cells[0].Text.ToString().Trim();

Codul care preia datele stocate în sesiuni:

try

{

}

numePacientTB.Text = Session["numePacient"].ToString(); cnpTB.Text = Session["cnp"].ToString();

catch (NullReferenceException ex)

{

LabelPrincipalErr.ForeColor = Color.Red;

LabelPrincipalErr.Text = "Eroare la transmiterea datelor utilizand sesiunin: " + ex.Message;

}

### Afișarea radiografiilor și funcția cu două clicuri

Această funcție se utilizează în pagina de „Vizualizare radiografii”, pentru afișarea dinamică a radiografiilor și diferite acțiuni la clic pe ele. La accesarea pagini se deschide baza de date și se începe încărcarea radiografiilor din tabelul „Radiografii” într-un DataSet. Dacă tabelul radiografii din DataSet nu conține nici o radiografie se va afișa un mesaj corespunzător, altfel se parcurg radiografiile și pentru fiecare radiografie se va crea un buton imagine (ImageButton), căruia i se va asocia un gestionar pentru evenimentul clic.

Funcția care gestionează evenimentul clic asupra unei radiografii, va crea dinamic un ModalPopupExtender (control din AJAX Control Toolkit) care va afișa panelul ascuns cu imagina de pe buton, în format mai mare. Pentru că ModalPopupExtender se creează de abia la clic pe

radiografie, nici nu are cum să fie executat din prima și din cauza aceea se execută la al doilea clic, pe aceeași radiografie. Această , așa numită „eroare”, am folosit-o în avantajul meu și la primul clic se afișează informații despre radiografia selectată, iar la al doilea se va acționa ModalPopupExtender.

SqlConnection con;

string conString = @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=e:\An 3 - Sem 2\Licenta\CabStomatologic\CabStomatologic\App\_Data\CabinetStomatologic.mdf;Integrated Security=True";

string filePath = @"~\RadiografiiPoze\";

protected void Page\_Load(object sender, EventArgs e)

{

try

{

con = new SqlConnection(conString); con.Open();

if (!string.IsNullOrEmpty(numeM))

{

string stmt = "select Id, Data, NumeImagine, Diagnostic, Observatii from Radiografii where CNP='" + cnpTB.Text.Trim() + "'";

SqlDataAdapter da = new SqlDataAdapter(stmt, con); DataSet ds = new DataSet();

da.Fill(ds, "Radiografii");

DataRow[] radiografii = ds.Tables["Radiografii"].Select();

if (cnpTB.Text.Trim() != "")

{

if (ds.Tables["Radiografii"].Rows.Count == 0)

{

LabelRadiografiiErr.ForeColor = Color.Green;

LabelRadiografiiErr.Text = "Pacientul cu CNP-ul " + cnpTB.Text.Trim() + " nu are nici o radiografie.";

}

else

{

foreach (DataRow nume in radiografii)

{

ImageButton ibtn = new ImageButton();

ibtn.ImageUrl = filePath + nume["NumeImagine"].ToString().Trim(); ibtn.CommandName = nume["Id"].ToString().Trim();

ibtn.ID = nume["NumeImagine"].ToString().Trim(); ibtn.Height = 150;

ibtn.Width = 160;

ibtn.Attributes.Add("Style", "padding:20px;"); ibtn.Attributes.Add("title", nume["NumeImagine"].ToString().Trim()); Panel2.Controls.Add(ibtn);

ibtn.Click += new ImageClickEventHandler(ibtn\_Click);

}

LabelRadiografiiErr.ForeColor = Color.Green; LabelRadiografiiErr.Text = "Pentru pacientul cu CNP-ul " +

cnpTB.Text.Trim() + " au fost gasite urmatoarele radiografii.";

LabelPrincipalErr.Text = "";

}

}

}

else

{

da.Dispose();

ds.Dispose();

LabelPrincipalErr.ForeColor = Color.Red;

LabelPrincipalErr.Text = "Eroare la transmiterea datelor utilizand sesiunin.

Medicul nu este logat.";

LabelInformatiiRErr.Text = ""; LabelRadiografiiErr.Text = "";

}

}

catch (Exception ex)

{

LabelPrincipalErr.ForeColor = Color.Red;

LabelPrincipalErr.Text = "Nu se poate realiza conexiunea la baza de date: " + ex.Message;

LabelRadiografiiErr.Text = ""; LabelInformatiiRErr.Text = ""; LabelRadiografiiErr.Text = "";

}

finally

{

con.Close();

}

}

private void ibtn\_Click(object sender, ImageClickEventArgs e)

{

try

{

//partea pentru crearea ModalPopupExtender ImageButton ibtn = (ImageButton)sender; largeImg.Src = filePath + ibtn.ID; ModalPopupExtender mpe = new ModalPopupExtender(); mpe.PopupControlID = "Panel3";

mpe.TargetControlID = ibtn.ID; mpe.CancelControlID = "closeBtn"; mpe.BackgroundCssClass = "modalBackground"; Panel4.Controls.Add(mpe);

//partea pentru afișarea infroamtiilor con = new SqlConnection(conString); con.Open();

string stmt = "select NumeImagine, Data, Diagnostic, Observatii from Radiografii where CNP='" + cnpTB.Text.Trim() + "' and NumeImagine='" + ibtn.ID + "' and Id='" + ibtn.CommandName + "'";

SqlCommand sc = new SqlCommand(stmt, con); SqlDataReader dr = sc.ExecuteReader();

if (dr.Read())

{

DateTime data = Convert.ToDateTime(dr["Data"].ToString().Trim()); dataTB.Text = data.ToString("M/d/yyyy");

string diagnostic = dr["Diagnostic"].ToString().Trim(); diagnostic = diagnostic.Replace(";", "\n");

diagnostic = diagnostic.Replace("//", "\n- Cod diagnostic: "); diagnosticTB.Text = diagnostic;

observatiiTB.Text = dr["Observatii"].ToString().Trim(); numeImagineTB.Text = dr["NumeImagine"].ToString().Trim();

}

LabelInformatiiRErr.Text = ""; dr.Close();

sc.Dispose();

}

catch (Exception ex)

{

LabelInformatiiRErr.ForeColor = Color.Red;

LabelInformatiiRErr.Text = "Nu se poate realiza conexiunea la baza de date: " + ex.Message;

}

finally

{

con.Close();}}

## Stocarea informației

Aplicația pentru stocarea informațiilor folosește baza de date „CabinetStomatologic” și pe lângă aceasta mai folosește și ceva fișiere pentru încărcarea orașelor din județe. Cele mai importante tabele din baza de date a aplicației sunt prezentate mai jos în Figura 5.2.1 (tabele și legăturile între ele).

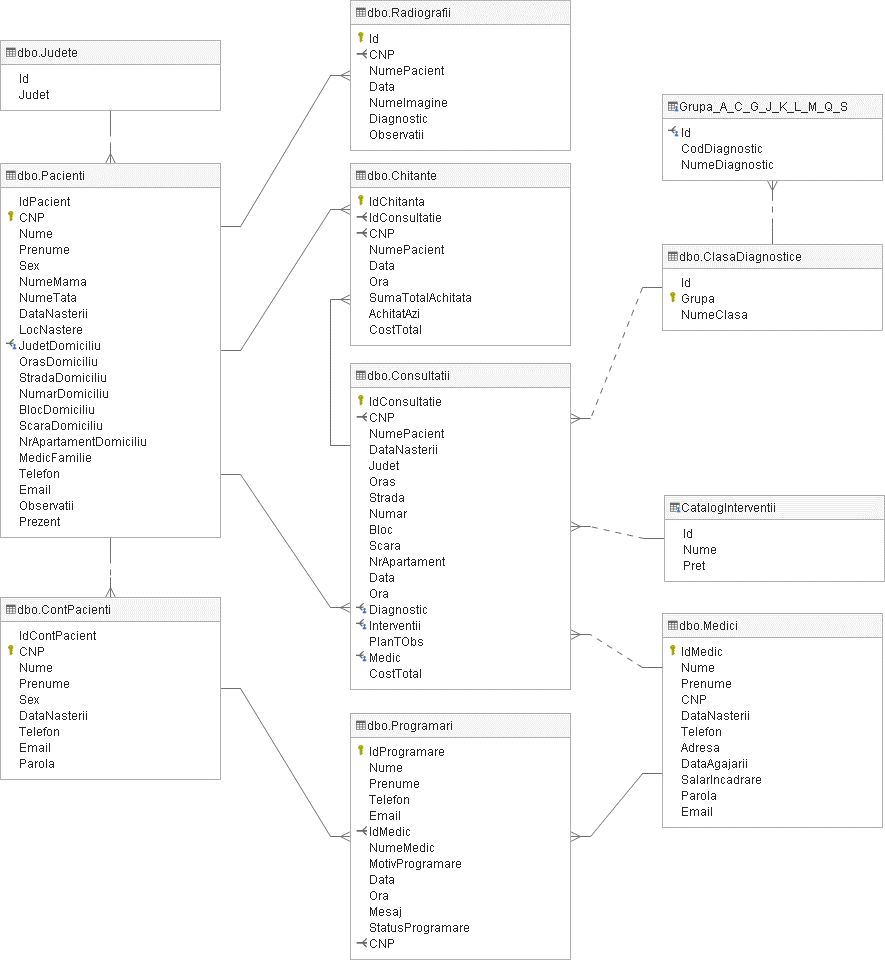


Figura 5.2.1 Baza de date a aplicației

Mai departe vor fi prezentate tabele cele mai importante din punct de vederea a pacientului, împreună cu structura câmpurilor sale.

Tabela „Pacienți” (figura 5.2.2) folosită pentru stocarea datelor legate de pacient, date personale, date de contact, domiciliu, etc.

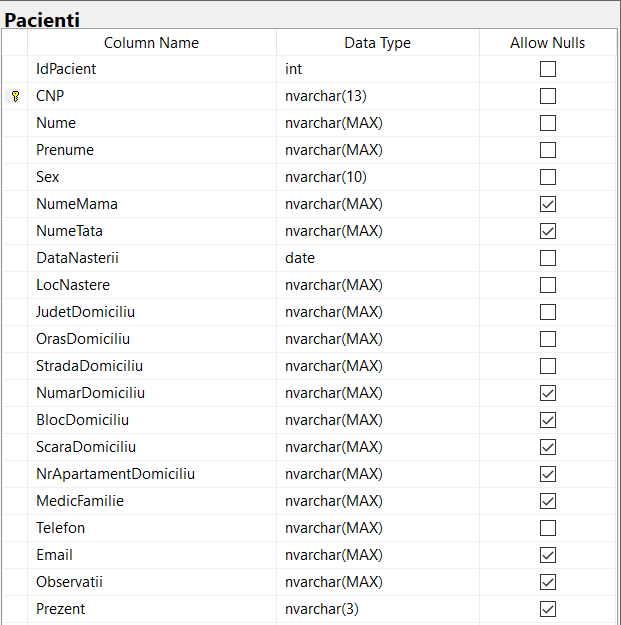


Figura 5.2.2 Tabela „Pacienți”

* *IdPacient* – câmp obligatoriu, reprezintă numărul pacientului în momentul înregistrării în baza de date, se auto incrementează la fiecare pacient nou;
* *CNP* – câmp obligatoriu, reprezintă CNP-ul pacientului, acest câmp face legătura între celelalte tabele și aceasta;
* Nume – câmp obligatoriu, reprezintă numele pacientului;
* Prenume – câmp obligatoriu, reprezintă prenumele pacientului;
* Sex – câmp obligatoriu, reprezintă sexul pacientului;
* NumeMama – câmp opțional, reprezintă numele și prenumele mamei;
* NumeTata – câmp opțional, reprezintă numele și prenumele tatălui;
* DataNasterii – câmp obligatoriu, reprezintă data nașterii a pacientului;
* LocNastere – câmp obligatoriu, reprezintă localitatea unde s-a născut pacientul;
* JudetDomiciliu – câmp obligatoriu, reprezintă județul domiciliului pacientului;
* OrasDomiciliu – câmp obligatoriu, reprezintă orașul domiciliului pacientului;
* StradaDomiciliu – câmp obligatoriu, reprezintă strada domiciliului pacientului;
* NumarDomiciliu – câmp opțional, reprezintă numarul străzi/casei domiciliului pacientului;
* BlocDomiciliu – câmp opțional, reprezintă blocul domiciliului pacientului;
* ScaraDomiciliu – câmp opțional, reprezintă scara blocului domiciliului pacientului;
* NrApartamentDomiciliu – câmp opțional, reprezintă numărul apartamentului domiciliului pacientului;
* MedicFamilie – câmp opțional, reprezintă numele și prenumele medicului de familie a pacientului;
* Telefon – câmp obligatoriu, reprezintă numărul de telefon a pacientului;
* Email – câmp opțional, reprezintă email-ul pacientului;
* Observatii – câmp opțional, reprezintă observații despre pacient (de exemplu, alergii);
* Prezent – câmp opțional, il modifică recepția dacă pacientul este prezent în sala de așteptare.

Tabela „ContPacienti” (figura 5.2.3) folosită la înregistrarea pacientului pentru a avea acces la aplicație.

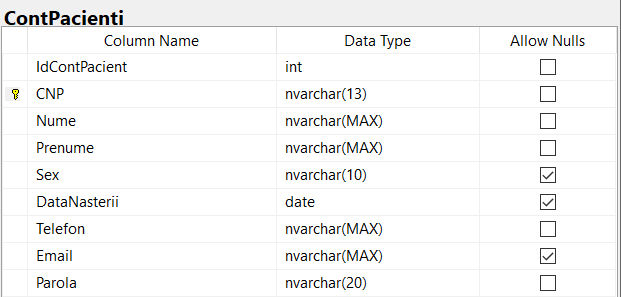


Figura 5.2.3 Tabela „ContPacienti”

* Tabela „ContPacienti” conține câmpuri asemănătoare cu tabela de pacienții, în plus ce are câmpul pentru parolă, care reprezintă parola folosită de pacient la autentificare în aplicație și câmpul IdContPacient, care reprezintă numărul de cont al pacientului (se auto incrementează).

Tabela „Programari” (figura 5.2.4), aici se încarcă toate programările solicitate de către pacienți.

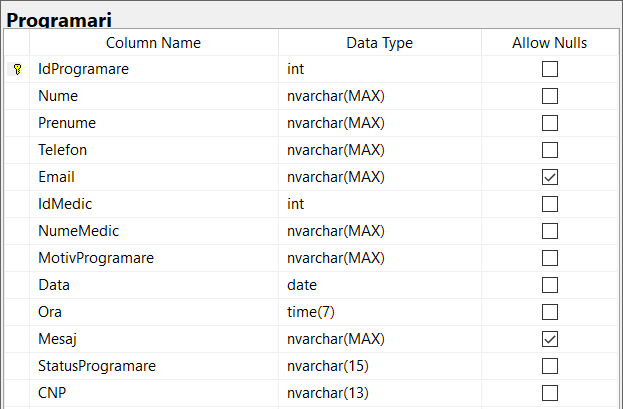


Figura 5.2.4 Tabela „Programari”

* IdProgramare – câmp obligatoriu, reprezintă numărulul după care se identifică programarea pacientului, se auto incrementează la fiecare programare;
* Nume – câmp obligatoriu, reprezintă numele pacientului;
* Prenume – câmp obligatoriu, reprezintă prenumele pacientului;
* Telefon – câmp obligatoriu, reprezintă numărul de telefon a pacientului;
* Email – câmp opțional, reprezintă email-ul pacientului;
* IdMedic – câmp obligatoriu, reprezintă numărul după care este medicul înregistrat în această aplicație, acest câmp face legătura între tabelul „Medici” și acest tabel;
* NumeMedic – câmp obligatoriu, reprezintă numele și prenumele medicului la care se face programarea;
* Data – câmp obligatoriu, reprezintă data programări;
* Ora – câmp obligatoriu, reprezintă ora programări;
* Mesaj – câmp opțional, reprezintă mesajul pacientului;
* StatusProgramare – câmp obligatoriu, reprezintă statusul programări (confirmată, neconfirmată, expirată);
* CNP – câmp obligatoriu, reprezintă CNP-ul pacientului.

Tabela „Consultatii” (figura 5.2.5) stochează toate consultațiile efectuate de acest cabinet stomatologic.

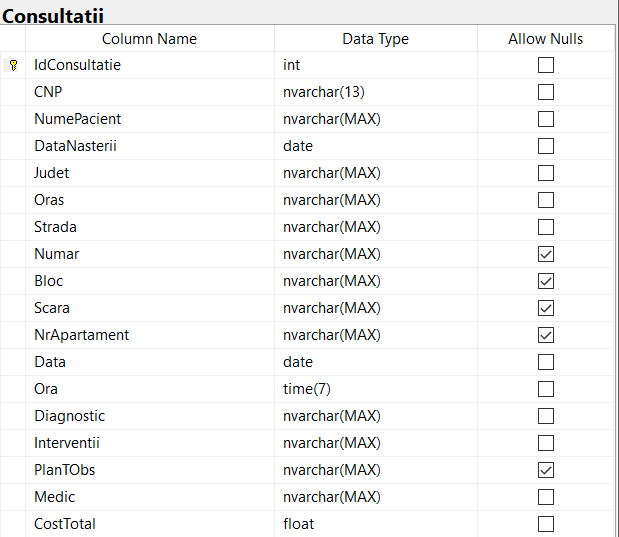


Figura 5.2.5 Tabela „Consultatii”

* IdConsultatie – câmp obligatoriu, reprezintă numărul după care se identifică consultația, se auto incrementează după fiacre consultație nouă și face legătura între tabelul „Chitante” și acest tabel;
* CNP – câmp obligatoriu, reprezintă CNP-ul pacientului, acest câmp face legătura între tabelul

„Pacienti” și acest tabel;

* NumePacient – câmp obligatoriu, reprezintă numele și prenumele pacientului;
* DataNasterii – câmp obligatoriu, reprezintă data nașterii a pacientului;
* Judet – câmp obligatoriu, reprezintă județul domiciliului pacientului;
* Oras – câmp obligatoriu, reprezintă orașul domiciliului pacientului;
* Strada – câmp obligatoriu, reprezintă strada domiciliului pacientului;
* Numar – câmp opțional, reprezintă numărul străzi/casei domiciliului pacientului;
* Bloc – câmp opțional, reprezintă blocul domiciliului pacientului;
* Scara – câmp opțional, reprezintă scara blocului domiciliului pacientului;
* NrApartament – câmp opțional, reprezintă numărul apartamentului domiciliului pacientului;
* Data – câmp obligatoriu, reprezintă data când s-a efectuat consultația;
* Ora – câmp obligatoriu, reprezintă ora la care s-a efectuat consultația;
* Diagnostic – câmp obligatoriu, reprezintă diagnosticul/diagnosticele care sau stabilit în timpul consultației;
* Intervenții – câmp obligatoriu, reprezintă intervențiile efectuate în timpul consultației;
* PlanTObs – câmp opțional, reprezintă planul de tratament stabilit după consultație și alte observații din timpul consultației;
* Medic – câmp obligatoriu, reprezintă numele și prenumele medicului care a întocmit consultația, acest câmp face legătura între tabelul „Medici” și acest tabel;
* CostTotal – câmp obligatoriu, reprezintă costul total al serviciilor oferite în timpul consultației.

Tabela „Chitante” (figura 5.2.6), toate chitanțele care sunt emise pacienților sunt stocate în această tabelă.

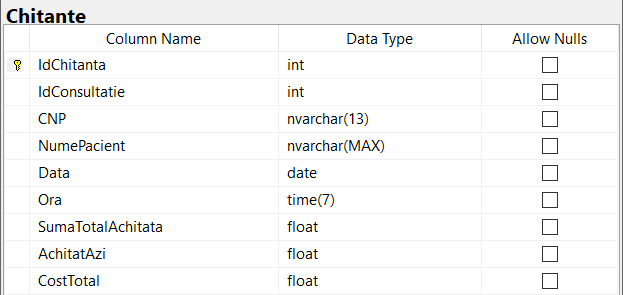


Figura 5.2.6 Tabela „Chitante”

* IdChitanta – câmp obligatoriu, reprezintă numărul fiecărei chitanțe emise, se auto incrementează după fiecare chitanță nouă;
* IdConsultație – câmp obligatoriu, reprezintă numărul după care se identifică consultația achitată, face legătura între tabelul „Consultatii” și acest tabel;
* CNP – câmp obligatoriu, reprezintă CNP-ul pacientului, acest câmp face legătura între tabelul

„Pacienti” și acest tabel;

* NumePacient – câmp obligatoriu, reprezintă numele și prenumele pacientului;
* Data – câmp obligatoriu, reprezintă data când s-a emis chitanța;
* Ora – câmp obligatoriu, reprezintă ora la care s-a emis chitanța;
* SumaTotalAchitata – câmp obligatoriu, reprezintă suma care a achitat-o pacientul pentru o consultație până în momentul de față (dacă se plătește în rate);
* AchitatAzi – câmp obligatoriu, reprezintă suma care se achită în momentul emiteri chitanței;
* CostTotal – câmp obligatoriu, reprezintă costul total al consultației pentru care se emite chitanța.

Tabela „Radiografii” (figura 5.2.7) se folosește pentru stocarea radiografiilor pacientului, anterior solicitate de către medici stomatologi.



Figura 5.2.7 Tabela „Radiografii”

* Id – câmp obligatoriu, reprezintă numărul după care se stochează radiografia, se auto incrementează după fiecare radiografie nouă;
* CNP – câmp obligatoriu, reprezintă CNP-ul pacientului, acest câmp face legătura între tabelul

„Pacienti” și acest tabel;

* NumePacient – câmp obligatoriu, reprezintă numele și prenumele pacientului;
* DataNasterii – câmp obligatoriu, reprezintă data nașterii a pacientului;
* Data – câmp obligatoriu, reprezintă data când s-a adăugat radiografia (sau data când radiografia a fost făcută);
* NumeImagine – câmp obligatoriu, reprezintă numele radiografiei;
* Diagnostic – câmp obligatoriu, reprezintă diagnosticul (diagnosticele) care s-a constatat pe baza radiografiei;
* Observatii – câmp opțional, reprezintă observațiile medicului în legătură cu radiografia.

# Capitolul 6. Concluzii și perspective

Am realizat un sistem informatic medical pentru gestionarea pacienților dintr-un cabinet stomatologic.

Aplicația este realizată în mediul de dezvoltare Microsoft Visual Studio 2017 (tehnologia ASP.Net), folosind baze de date Microsoft SQL Server 2017 și din cauza faptului că aplicația a fost dezvoltată pe platforma ASP.Net, poate fi accesibilă de oriunde și de pe orice, prind intermediul unui browser de internet.

Aplicația a fost proiectată pentru trei tipuri de utilizatori, pacient, recepție, medic. Pentru pacienții s- a încercat cât mai multă informație de care, posibil, va avea nevoie, să fie disponibilă cât se poate mai rapid prin intermediul internet-ului. La recepție și medic, s-a pus foarte mult accent pe accelerarea și ușurarea sarcinilor formale (adăugarea de pacienți noi, întocmirea consultațiilor, etc.).

Pentru viitor, aplicația poate fi dezvoltată în orice moment, adăugând-i-se tot mai multe facilitații. De exemplu s-ar putea introduce, pe lângă facilitățile pe care le are pacientul și posibilitatea de ași vizualiza fișa, consultațiile, radiografiile. De asemenea, se mai poate realiza și un modul pentru transferul datelor către alte aplicații.

# Bibliografie

1. <https://digitalguardian.com/blog/what-health-information-system>, Accesat în 16.06.2020.
2. <http://www.medinfo.umft.ro/dim/bioinformatica_files/master-poli/cursuri_pdf/bioinf-1.pdf>,

Accesat în 17.06.2020.

1. <https://www.dentuno.com/index.php>, Accesat în 17.06.2020.
2. [https://xfactorapp.com/soft-gestiune-cabinet-stomatologic/,](https://xfactorapp.com/soft-gestiune-cabinet-stomatologic/) Accesat în 17.06.2020
3. <https://ro.wikipedia.org/wiki/Microsoft_Visual_Studio>, Accesat în 18.06.2020
4. <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet>, Accesat în 18.06.2020
5. [https://ro.wikipedia.org/wiki/Hypertext\_Transfer\_Protocol,](https://ro.wikipedia.org/wiki/Hypertext_Transfer_Protocol) Accesat în 19.06.2020
6. [https://ro.wikipedia.org/wiki/Cod\_numeric\_personal#AA,](https://ro.wikipedia.org/wiki/Cod_numeric_personal#AA) Accesat în 25.06.2020

# Index figuri

[Figura 1.1 Ciclul elementar al informației medicale 4](#_bookmark0)

[Figura 2.1.1 Pagina de autentificare 7](#_bookmark1)

[Figura 2.1.2 Evidența programărilor 7](#_bookmark2)

[Figura 3.6.1 CalendarExtender 20](#_bookmark3)

[Figura 3.6.2 CollapsiblePanelExtender 21](#_bookmark4)

[Figura 3.6.3 ModalPopupExtender 21](#_bookmark5)

[Figura 4.1 Schema bloc a sistemului 22](#_bookmark6)

[Figura 4.2.1 Funcțiile aplicației (schema bloc) 23](#_bookmark7)

[Figura 4.3.1 Pagina „Acasa” 30](#_bookmark8)

[Figura 4.3.2 Pagina „Autentificare pacient” 31](#_bookmark9)

[Figura 4.3.3 Nu s-a introdus CNP-ul 32](#_bookmark10)

[Figura 4.3.4 Parola introdusă nu corespunde cu CNP-ul introdus mai sus 32](#_bookmark11)

[Figura 4.3.5 Adresa email introdusă este invalidă 32](#_bookmark12)

[Figura 4.3.6 Pagina „Inregistrare cont nou” 33](#_bookmark13)

[Figura 4.3.7 Numele nu are o formă corespunzătoare 34](#_bookmark14)

[Figura 4.3.8 Parola și confirmarea parolei nu coincid 34](#_bookmark15)

[Figura 4.3.9 Înregistrarea a avut succes 34](#_bookmark16)

[Figura 4.3.10 Pagina „Pagina Principala” 35](#_bookmark17)

[Figura 4.3.11 Formular programare 35](#_bookmark18)

[Figura 4.3.12 Data selectată nu reprezintă o zi lucrătoare pentru cabinet 36](#_bookmark19)

[Figura 4.3.13 Ora selectată este deja ocupată 36](#_bookmark20)

[Figura 4.3.14 Pagina „Programari” 37](#_bookmark21)

[Figura 4.3.15 Acționarea butonului „Expirate” 37](#_bookmark22)

[Figura 4.3.16 Pagina „Autentificare receptie” 38](#_bookmark23)

[Figura 4.3.17 Nu s-a introdus parola 38](#_bookmark24)

[Figura 4.3.18 Adresa email introdusă nu reprezintă email-ul cabinetului 38](#_bookmark25)

[Figura 4.3.19 Pagina „Pagina Principala” 39](#_bookmark26)

[Figura 4.3.20 CNP-ul nu conține 13 cifre 39](#_bookmark27)

[Figura 4.3.21 Numele introdus nu există în baza de date a cabinetului 39](#_bookmark28)

[Figura 4.3.22 Căutarea a avut succes 39](#_bookmark29)

[Figura 4.3.23 Pagina „Adaugare pacient nou” 41](#_bookmark30)

[Figura 4.3.24 Nu s-a selectat județul 42](#_bookmark31)

[Figura 4.3.25 Numărul introdus are valoare negativă 42](#_bookmark32)

[Figura 4.3.26 Numele tatălui nu are o formă corespunzătoare 42](#_bookmark33)

[Figura 4.3.27 Pagina „Vizualizare / Editare fisa pacient” 43](#_bookmark34)

[Figura 4.3.28 Salvarea modificări 43](#_bookmark35)

[Figura 4.3.29 Eroare la selectarea orașului (domiciliu) 43](#_bookmark36)

[Figura 4.3.28 Pagina „Consultatii neachitate” 44](#_bookmark37)

[Figura 4.3.29 Nu au fost găsite consultați pentru numele introdus 44](#_bookmark38)

[Figura 4.3.30 Pacientul nu are consultații neachitate 44](#_bookmark39)

[Figura 4.3.32 Pagina „Tiparire chitanta” 45](#_bookmark40)

[Figura 4.3.33 Suma introdusă este mai mare de cât suma ce trebuie achitată 46](#_bookmark41)

[Figura 4.3.34 Nu s-a introdus suma achitată azi 46](#_bookmark42)

[Figura 4.3.35 Pagina „Programari” (recepție) 47](#_bookmark43)

[Figura 4.3.36 Eroare la confirmarea programării deja expirate 47](#_bookmark44)

[Figura 4.3.37 Confirmarea programării a avut succes 47](#_bookmark45)

[Figura 4.3.38 Pagina „Autentificare medic” 49](#_bookmark46)

[Figura 4.3.39 Nu s-a selectat utilizatorul 49](#_bookmark47)

[Figura 4.3.40 Adresa email este corectă 49](#_bookmark48)

[Figura 4.3.41 Pagina „Pagina Principala” (medic) 50](#_bookmark49)

[Figura 4.3.42 Pacientul prezent este selectat (butoanele active) 50](#_bookmark50)

[Figura 4.3.43 Pagina „Adaugare consultatie noua” 52](#_bookmark51)

[Figura 4.3.44 Nu s-a selectat dintele pentru care se adaugă diagnosticul 53](#_bookmark52)

[Figura 4.3.45 Nu s-a adăugat nici un diagnostic pentru dintele selectat 53](#_bookmark53)

[Figura 4.3.46 Nu s-a adăugat intervenția efectuată 53](#_bookmark54)

[Figura 4.3.47 Pagina „Adugare radiografie noua” 54](#_bookmark55)

[Figura 4.3.48 Nu s-a încărcat radiografia 54](#_bookmark56)

[Figura 4.3.49 Radiografia încărcată a mai fost adăugată 55](#_bookmark57)

[Figura 4.3.50 Pagina „Vizualizare fisa pacient” 55](#_bookmark58)

[Figura 4.3.51 Pagina „Vizualizare consultatii pacient” 56](#_bookmark59)

[Figura 4.3.52 Pagina „Vizualizare radiografii” 57](#_bookmark60)

[Figura 4.3.53 Pagina „Programari” (medic) 57](#_bookmark61)

[Figura 4.3.54 Data selectată manual de către medic 58](#_bookmark62)

[Figura 4.6.1 Fișa pacientului tipărită la imprimantă 60](#_bookmark63)

[Figura 4.6.2 Consultația tipărită la imprimantă 60](#_bookmark64)

[Figura 4.6.3 Chitanța tipărită la imprimantă 61](#_bookmark65)

[Figura 5.2.1 Baza de date a aplicației 74](#_bookmark66)

[Figura 5.2.2 Tabela „Pacienți” 75](#_bookmark67)

[Figura 5.2.3 Tabela „ContPacienti” 76](#_bookmark68)

[Figura 5.2.4 Tabela „Programari” 76](#_bookmark69)

[Figura 5.2.5 Tabela „Consultatii” 77](#_bookmark70)

[Figura 5.2.6 Tabela „Chitante” 78](#_bookmark71)

[Figura 5.2.7 Tabela „Radiografii” 79](#_bookmark72)

