



**UNIVERSITATEA TEHNICĂ „GHEORGHE ASACHI” IAȘI
FACULTATEA AUTOMATICĂ ȘI CALCULATOARE
SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA
INFORMAȚIEI
DISCIPLINA: INGINERIA PROGRAMĂRII**



**APLICAȚIE PENTRU GESTIONAREA
UNUI HOTEL ȘI A PERSONALULUI ACESTUIA**

**Coordonator,
Prof. Corina Cîmpanu**

Studenți,

**Bîzgan Alina, 1306B
Iftime Adrian Dumitru, 1306B
Modiga Camelia Maria, 1306B
Ștefan Alexandra, 1306B**

IAȘI, 2021

Documentul specificațiilor cerințelor software

1. Introducere

1.1 Scopul

Acest SRS are ca scop descrierea funcționalității aplicației pentru gestionarea unui hotel și a personalului acestuia. Funcționalitatea aplicației constă în afișarea, inserarea și ștergerea datelor în funcție de drepturile angajatului care se loghează.

1.2 Motivul

Motivul din spatele proiectului este de a crea o aplicație care ușurează munca personalului unui hotel. Astfel angajatul se conectează cu numele de utilizator și parola specifică postului său și poate accesa cu ușurință date referitoare la persoanele cazate, ceilalți angajați sau starea camerelor (ocupate sau rezervate).

1.3 Definiții, acronime și abrevieri

Șablonul Singleton implică existența unei singure clase care este responsabilă cu crearea unui obiect în timp ce se asigură că doar un singur obiect este creat (doar o singură instanță a clasei poate exista).

Șablonul Proxy asigură un înlocuitor pentru alt obiect pentru a controla accesul la acesta.

1.4 Referințe

- https://www.youtube.com/watch?v=ga8q90XFqDk&t=167s&ab_channel=ChituOkoliChituOkoli
- <https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database>
- Ingineria programării - Laborator 8 (Șablonul Singleton)
- Ingineria programării - Laborator 9 (Șablonul Proxy)

1.5 Prezentare generală

Secțiunea 2 prezintă mai în profunzime cerințele acestei aplicații. Această secțiune este destinată clientului sau persoanelor care nu au o pregătire tehnică în acest domeniu.

Secțiunea 3 va conține toate cerințele tehnice ale aplicației incluzând o listă de cerințe funcționale și non-funcționale. Această secțiune este destinată dezvoltatorilor de software pentru a putea crea acest program.

2. Descriere

Acest proiect reprezintă o aplicație pentru gestionarea unui hotel și a personalului acestuia. Aplicația constă în evidența rapidă și ușoară a rezervărilor. Funcționalitatea

aplicației constă în afișarea, inserarea și ștergerea datelor în funcție de drepturile angajatului care se loghează. Pentru stocarea informațiilor privind persoanele cazate, tarifele camerelor, clasificarea camerelor, tarifele serviciilor auxiliare (internet, parcare, telefon), conținutul minibarului, starea camerelor (ocupate sau rezervate) s-a folosit o baza de date.

2.1 Perspectiva aplicației

Proiectul oferă utilizatorului o variantă simplă de a accesa informațiile referitoare la rezervările unor camere de hotel stocate într-o bază de date. Nu este singurul și nici primul produs de acest gen de pe piață dar poate aduce un aspect nou și mai simplu de utilizat.

2.2 Funcțiile aplicației

Aplicația dă posibilitatea următoarelor acțiuni realizabile de către personalul hotelului în funcție de angajatul care se conectează:

- Dacă angajatul se conectează cu numele de utilizator *admin* el va avea rolul de administrator și va putea vizualiza situația camerelor, fie ele ocupate sau rezervate, va putea adăuga noi camere care pot fi ulterior închiriate, va putea adăuga noi angajați, el fiind responsabil cu gestionarea întregului personal. De asemenea el poate șterge camere și angajați din baza de date.
- Dacă angajatul se conectează cu numele de utilizator *rec* el va avea rolul de recepționar și va putea vizualiza situația camerelor, fie ele ocupate sau rezervate, va putea crea o nouă rezervare pentru client. De asemenea va putea șterge și vizualiza rezervări.
- Dacă angajatul se conectează cu numele de utilizator *cam* el va avea rolul de camerist și va putea vizualiza situația camerelor, fie ele ocupate sau rezervate pentru a cunoaște momentul în care trebuie să actualizeze conținutul minibarului din camera respectivă.

2.3 Caracteristicile utilizatorului

Aplicația poate fi utilizată de oricine în orice domeniu fără a fi nevoie de o pregătire tehnică anticipată. Aplicația vizează persoanele din domeniul hotelier care vor utiliza aplicația pentru a gestiona rezervarea camerelor.

3. Cerințe specifice

3.1 Cerințe externe

3.1.1 Interfața cu utilizatorul

- Front-end software: Windows Forms App (.NET Framework)

- Back-end software: C#, Oracle

3.1.2 Componente hardware

- Windows

3.1.3 Componente software

Drept sistem de operare am folosit Windows datorită utilizării sale în masă și a ușurinței în utilizare. Drept mediu de dezvoltare am folosit Microsoft Visual Studio 2019 Community utilizând în crearea aplicației limbajul C#.

Interfața cu utilizatorul este realizată cu Windows Forms App (.NET Framework) iar conectarea cu baza de date s-a realizat cu Oracle Database 11gR2 Express Editions for Windows x32.

3.2 Cerințe de performanță

Fiind o aplicație simplă, ea poate fi rulată pe o platformă care nu necesită o performanță ridicată. Astfel avem nevoie de un procesor de 1,8 GHz, 58 MB memorie RAM și spațiu liber de 300 KB.

3.3 Constrângeri de design

Pentru că interfața a fost realizată cu Windows Forms App (.NET Framework) aceasta este una simplistă dar care conține suficiente detalii astfel încât poate să ofere utilizatorului acces la toate operațiile de care are nevoie.

Diagrame UML

Diagrama de cazuri de utilizare

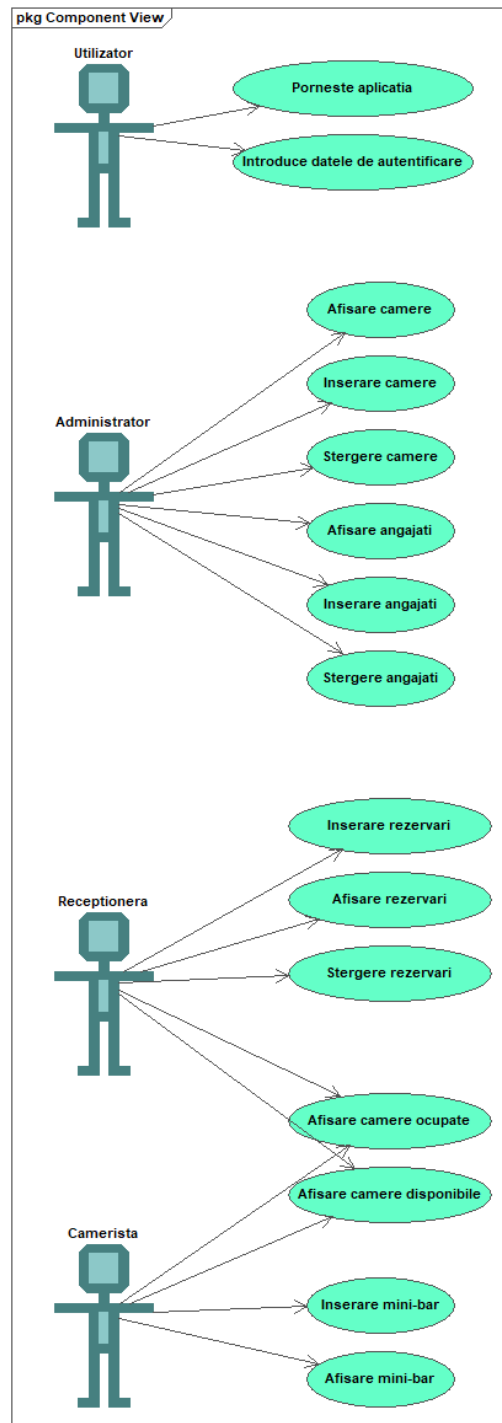


Diagrama de clase

```

<<partial>>
Admin
(from ProjectP)

_acasa:Acasa
_angajatProxy:AngajatProxy
_idAngajat:string=""
_idCamera:string=""
_idMinibar:string=""
_numaAngajat:string=""
_parolaAngajat:string=""
_pasaAngajat:string=""
_ipCamera:string=""
_pretCamera:string=""
components:System.ComponentModel.IContainer=null
tabControl:AdminSystem.Windows.Forms.TabControl
tabPageMficareCamera:System.Windows.Forms.TabPage
buttonMficareCamera:System.Windows.Forms.Button
dataGridViewCamera:System.Windows.Forms.DataGridView
tabPageInserareCamera:System.Windows.Forms.TabPage
tabPageStergereCamera:System.Windows.Forms.TabPage
tabPageMficareAngajati:System.Windows.Forms.TabPage
buttonMficareAngajati:System.Windows.Forms.Button
dataGridViewAngajati:System.Windows.Forms.DataGridView
tabPageInserareAngajati:System.Windows.Forms.TabPage
label5:System.Windows.Forms.Label
textBoxPostAngajat:System.Windows.Forms.RichTextBox
label4:System.Windows.Forms.Label
textBoxNumaAngajat:System.Windows.Forms.RichTextBox
buttonInserareAngajat:System.Windows.Forms.Button
tabPageStergereAngajati:System.Windows.Forms.TabPage
groupBox1:System.Windows.Forms.GroupBox
buttonInpoi:System.Windows.Forms.Button
buttonInserareCamera:System.Windows.Forms.Button
comboBoxIdMinibar:System.Windows.Forms.ComboBox
label10:System.Windows.Forms.Label
richTextBoxPretCamera:System.Windows.Forms.RichTextBox
richTextBoxTipCamera:System.Windows.Forms.RichTextBox
label8:System.Windows.Forms.Label
label2:System.Windows.Forms.Label
buttonStergereCamera:System.Windows.Forms.Button
labelStergereCamera:System.Windows.Forms.Label
comboBoxIdStergereCamera:System.Windows.Forms.ComboBox
label9:System.Windows.Forms.Label
comboBoxIdStergereAngajati:System.Windows.Forms.ComboBox
buttonStergereAngajati:System.Windows.Forms.Button
textBoxParolaAngajat:System.Windows.Forms.RichTextBox
label1:System.Windows.Forms.Label
buttonExit:System.Windows.Forms.Button

<<GetAccessor, SetAccessor, property>> IdAngajat():string
<<GetAccessor, SetAccessor, property>> IdCamera():string
<<GetAccessor, SetAccessor, property>> IdMinibar():string
<<GetAccessor, SetAccessor, property>> numaAngajat():string
<<GetAccessor, SetAccessor, property>> parolaAngajat():string
<<GetAccessor, SetAccessor, property>> pasaAngajat():string
<<GetAccessor, SetAccessor, property>> ipCamera():string
<<GetAccessor, SetAccessor, property>> pretCamera():string
<<constructor>> Admin()
InserareCamera():bool
StergereAngajat():bool
StergereCamera():bool
refreshAllComboBoxes():void
buttonInpoi_Click(in sender:object, in e:EventArgs):void
buttonMficareCamera_Click(in sender:object, in e:EventArgs):void
buttonMficareAngajati_Click(in sender:object, in e:EventArgs):void
buttonInserareCamera_Click(in sender:object, in e:EventArgs):void
buttonStergereAngajati_Click(in sender:object, in e:EventArgs):void
buttonStergereCamera_Click(in sender:object, in e:EventArgs):void
InserareAngajati():void
buttonInserareAngajati_Click(in sender:object, in e:EventArgs):void
buttonExit_Click(in sender:object, in e:EventArgs):void
<<override>> Dispose(in disposing:bool):void
InitializeComponent():void

```














































```
























<<partial>>
Receptioners
(from ProjectP)

_acasa:Acasa
_telefonFix:string="0"
_parcare:string="0"
_uscatorPar:string="0"
_internet:string="0"
_idRezervare:string=""
_idCamera:string=""
_numestring=""
_prenumestring=""
_cnp:string=""
_dataCazare:string=""
_dataDecazare:string=""
components:System.ComponentModel.IContainer=null
tabControlReceptionera:System.Windows.Forms.TabControl
tabPageCameraOcupate:System.Windows.Forms.TabPage
buttonMficareOcupate:System.Windows.Forms.Button
dataGridViewCameraOcupate:System.Windows.Forms.DataGridView
tabPageCameraDisponibile:System.Windows.Forms.TabPage
buttonMficareDisponibile:System.Windows.Forms.Button
dataGridViewCameraDisponibile:System.Windows.Forms.DataGridView
tabPageInserareRezervare:System.Windows.Forms.TabPage
label5:System.Windows.Forms.Label
textBoxPrenume:System.Windows.Forms.RichTextBox
textBoxCNP:System.Windows.Forms.RichTextBox
label4:System.Windows.Forms.Label
textBoxNuma:System.Windows.Forms.RichTextBox
label1:System.Windows.Forms.Label
buttonInserare:System.Windows.Forms.Button
tabPageMficareRezervari:System.Windows.Forms.TabPage
buttonMficareRezervari:System.Windows.Forms.Button
dataGridViewRezervari:System.Windows.Forms.DataGridView
groupBox1:System.Windows.Forms.GroupBox
buttonInpoi:System.Windows.Forms.Button
tabPageStergere:System.Windows.Forms.TabPage
buttonStergereRezervare:System.Windows.Forms.Button
label8:System.Windows.Forms.Label
comboBoxIdCamera:System.Windows.Forms.ComboBox
checkboxBoxTelefonFix:System.Windows.Forms.CheckBox
checkboxBoxParcare:System.Windows.Forms.CheckBox
checkboxBoxUscatorPar:System.Windows.Forms.CheckBox
checkboxBoxInternet:System.Windows.Forms.CheckBox
label7:System.Windows.Forms.Label
label6:System.Windows.Forms.Label
richTextBoxDataDecazare:System.Windows.Forms.RichTextBox
richTextBoxDataCazare:System.Windows.Forms.RichTextBox
comboBoxStergereIdRezervare:System.Windows.Forms.ComboBox
label9:System.Windows.Forms.Label
buttonExit:System.Windows.Forms.Button

<<GetAccessor, SetAccessor, property>> TelefonFix():string
<<GetAccessor, SetAccessor, property>> Parcare():string
<<GetAccessor, SetAccessor, property>> UscatorPar():string
<<GetAccessor, SetAccessor, property>> Internet():string
<<GetAccessor, SetAccessor, property>> IdRezervare():string
<<GetAccessor, SetAccessor, property>> IdCamera():string
<<GetAccessor, SetAccessor, property>> Nume():string
<<GetAccessor, SetAccessor, property>> Prenume():string
<<GetAccessor, SetAccessor, property>> Cnp():string
<<GetAccessor, SetAccessor, property>> DataCazare():string
<<GetAccessor, SetAccessor, property>> DataDecazare():string
<<constructor>> Receptionera()
InserareRezervare():bool
StergereRezervare():bool
refreshAllComboBoxes():void
buttonInpoi_Click(in sender:object, in e:EventArgs):void
buttonMficareOcupate_Click(in sender:object, in e:EventArgs):void
buttonMficareDisponibile_Click(in sender:object, in e:EventArgs):void
buttonInserareRezervare_Click(in sender:object, in e:EventArgs):void
buttonMficare_Click(in sender:object, in e:EventArgs):void
buttonStergereRezervare_Click(in sender:object, in e:EventArgs):void
buttonExit_Click(in sender:object, in e:EventArgs):void
<<override>> Dispose(in disposing:bool):void
InitializeComponent():void

```

<<partial>> Camerista	
(from ProjectIP)	
	_acasa:Acasa
	_idMinibar:string=""
	_bauturiCarbogazoase:string=""
	_apaPlata:string=""
	_bere:string=""
	_vin:string=""
	components:System.ComponentModel.IContainer=null
	groupBox1:System.Windows.Forms.GroupBox
	buttonInapoi:System.Windows.Forms.Button
	tabControlCamerista:System.Windows.Forms.TabControl
	tabPageCamereOcupate:System.Windows.Forms.TabPage
	buttonAfisareC:System.Windows.Forms.Button
	dataGridViewCamereOSystem.Windows.Forms.DataGridView
	tabPageCamereDisponibile:System.Windows.Forms.TabPage
	buttonAfisareD:System.Windows.Forms.Button
	dataGridViewCamereDSystem.Windows.Forms.DataGridView
	tabPageInserareMinibar:System.Windows.Forms.TabPage
	textBoxApaSystem.Windows.Forms.RichTextBox
	textBoxBereSystem.Windows.Forms.RichTextBox
	textBoxSucSystem.Windows.Forms.RichTextBox
	label3:System.Windows.Forms.Label
	label2:System.Windows.Forms.Label
	label4:System.Windows.Forms.Label
	textBoxVinSystem.Windows.Forms.RichTextBox
	label1:System.Windows.Forms.Label
	buttonInserare:System.Windows.Forms.Button
	tabPageAfisareMinibar:System.Windows.Forms.TabPage
	buttonAfisareM:System.Windows.Forms.Button
	dataGridViewMinibarSystem.Windows.Forms.DataGridView
	buttonExitSystem.Windows.Forms.Button
	<<GetAccessor, SetAccessor, property>> IdMinibar():string
	<<GetAccessor, SetAccessor, property>> BauturiCarbogazoase():string
	<<GetAccessor, SetAccessor, property>> ApaPlata():string
	<<GetAccessor, SetAccessor, property>> Bere():string
	<<GetAccessor, SetAccessor, property>> Vin():string
	<<constructor>> Camerista()
	InserareMinibar():bool
	buttonInapoi_Click(in sender:object, in e:EventArgs):void
	buttonAfisareC_Click(in sender:object, in e:EventArgs):void
	buttonAfisareD_Click(in sender:object, in e:EventArgs):void
	buttonInserare_Click(in sender:object, in e:EventArgs):void
	buttonAfisareM_Click(in sender:object, in e:EventArgs):void
	buttonExit_Click(in sender:object, in e:EventArgs):void
	<<override>> Dispose(in disposing:bool):void
	InitializeComponent():void

<<partial>> Acasa	
(from ProjectIP)	
	_angajatProxy:AngajatProxy
	instantaAcasa
	components:System.ComponentModel.IContainer=null
	buttonLogin:System.Windows.Forms.Button
	user:System.Windows.Forms.Label
	password:System.Windows.Forms.Label
	groupBox1:System.Windows.Forms.GroupBox
	buttonAjutor:System.Windows.Forms.Button
	buttonContact:System.Windows.Forms.Button
	buttonDespre:System.Windows.Forms.Button
	titlu:System.Windows.Forms.Label
	buttonExitSystem.Windows.Forms.Button
	textBoxUtilizatorSystem.Windows.Forms.TextBox
	textBoxParolaSystem.Windows.Forms.TextBox
	<<constructor>> Acasa()
	Instance():Acasa
	buttonLogin_Click(in sender:object, in e:EventArgs):void
	buttonExit_Click(in sender:object, in e:EventArgs):void
	buttonAjutor_Click(in sender:object, in e:EventArgs):void
	buttonDespre_Click(in sender:object, in e:EventArgs):void
	buttonContact_Click(in sender:object, in e:EventArgs):void
	<<override>> Dispose(in disposing:bool):void
	InitializeComponent():void

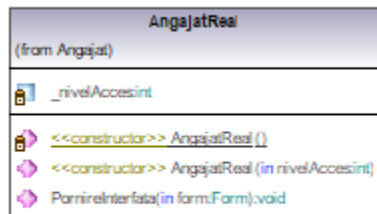
Generated by UModel

www



Generated by UModel

www.altova.com



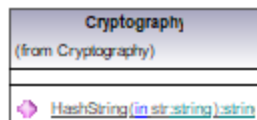
Generated by UModel

www.altova.com



Generated by UModel

www.altova.com



Generated by UModel

www.altova.com

FunctiIDII
(from FunctiIDII)
<pre> cstring="DATA SOURCE = XE;" + "PERSIST SECURITY INFO = True; USER ID = root; password = root; Pooling = False'; oracleConnection:OracleConnection=new OracleConnection(); selectFromDatabase(in viewDataGridView, in selectCommandstring);List<T1-> DataRow> refreshCombaBox(in combaCombaBox, in selectCommandstring);void </pre>

Generated by UModel

www.altova.com

<<attributes>> AdminTesting (from ComponentaTesting)
<pre> _idFrigider :string="" _idCamera:string="" </pre>
<pre> <<attributes>> Test0InsertCameraValid():void <<attributes>> Test1FrigiderOccupat():void <<attributes>> Test2AfisareIdCameraValid():void <<attributes>> Test3DeleteCameraValid():void <<attributes>> Test4AfisareIdCameraInvalid():void <<attributes>> Test5CameraLibera():void <<attributes>> Test6InsertFrigiderInvalid():void <<attributes>> Test7DeleteCameraInvalid():void </pre>

<<attributes>> ReceptioneraTesting (from ComponentaTesting)
<pre> _idRezervare:string="" </pre>
<pre> <<attributes>> Test1InsertRezervareValid():void <<attributes>> Test2CameraOccupata():void <<attributes>> Test3AfisareIdRezervareValid():void <<attributes>> Test4DeleteRezervareValid():void <<attributes>> Test5AfisareIdRezervareInvalid():void <<attributes>> Test6CameraLibera():void <<attributes>> Test7InsertRezervareInvalid():void <<attributes>> Test8DeleteRezervareInvalid():void </pre>

Generated by UModel

www.altova.com

<<attributes>> CameristaTesting (from ComponentaTesting)
<pre> <<attributes>> Test1NumarCamera():void <<attributes>> Test2InsertFrigiderValid():void <<attributes>> Test3InsertFrigiderInvalid1():void <<attributes>> Test4InsertFrigiderInvalid2():void </pre>

Generated by UModel

www.altova.com

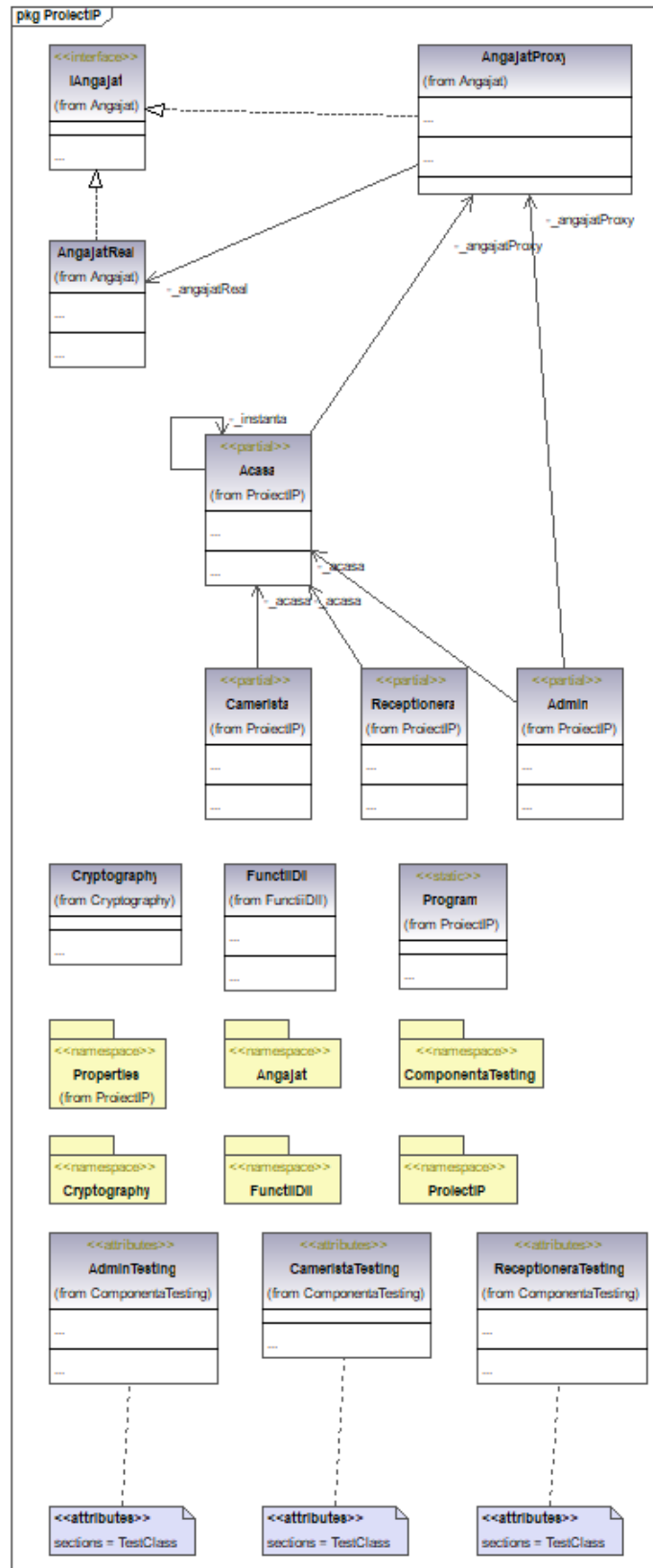


Diagrama de activități

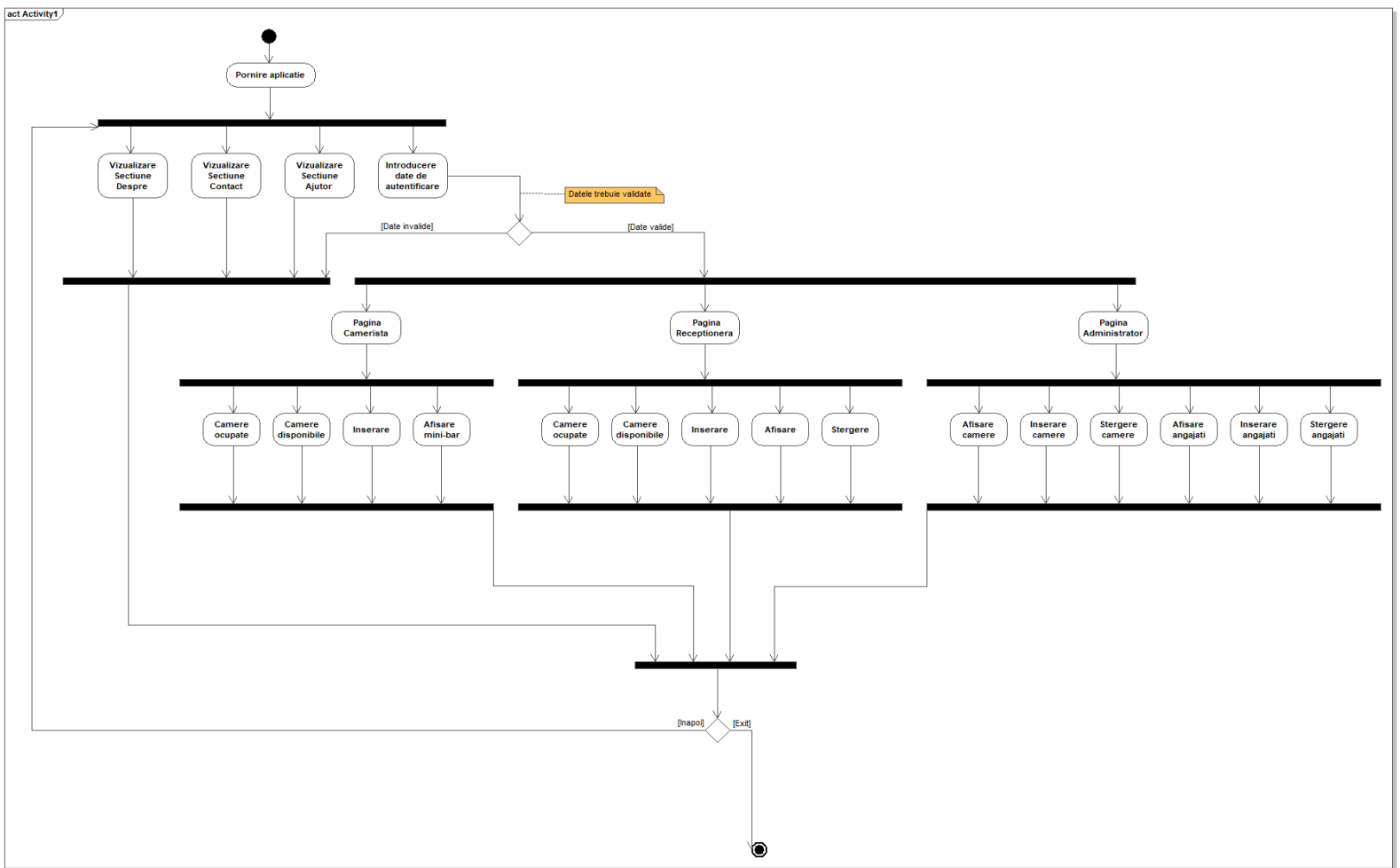


Diagrama de secvențe pentru utilizator

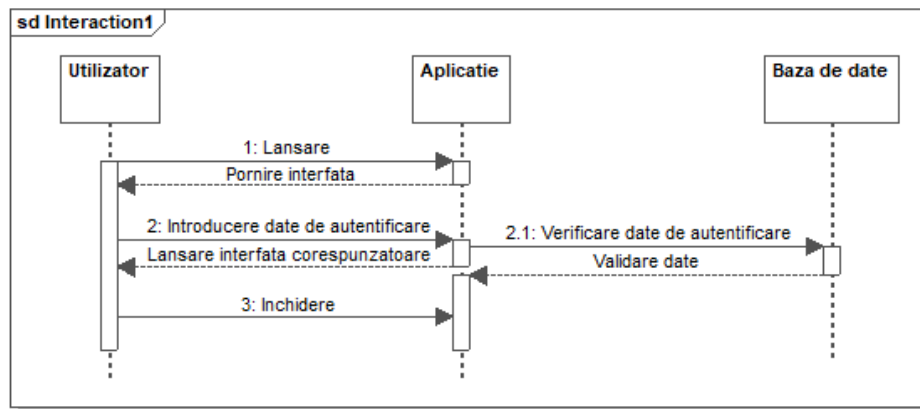


Diagrama de secvențe pentru administrator

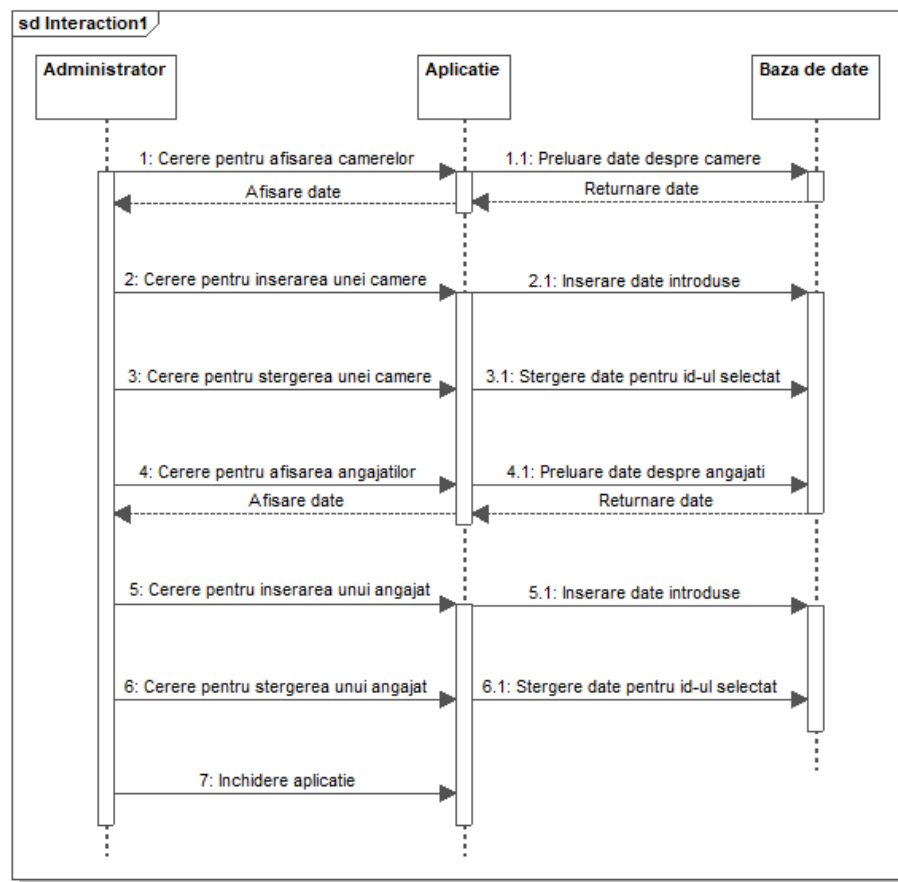


Diagrama de secvențe pentru recepționeră

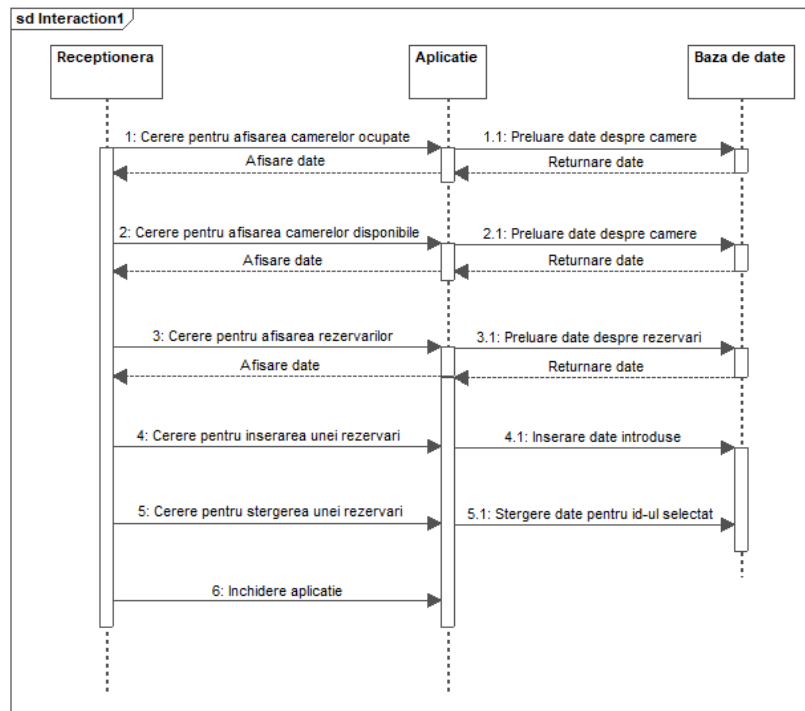
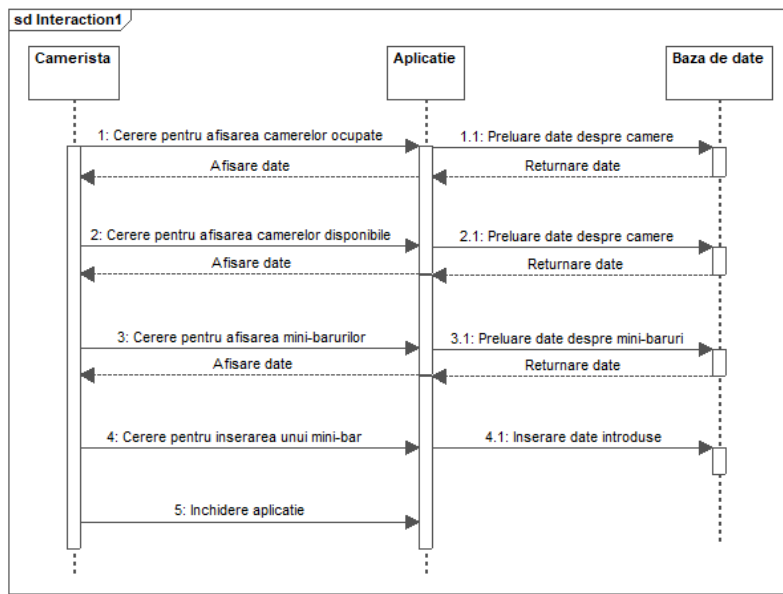
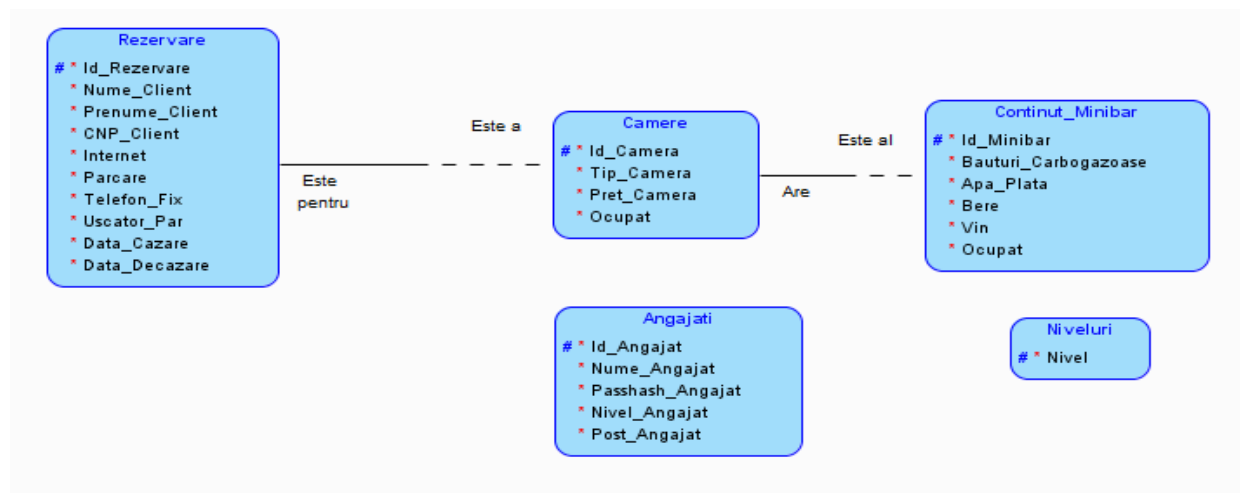


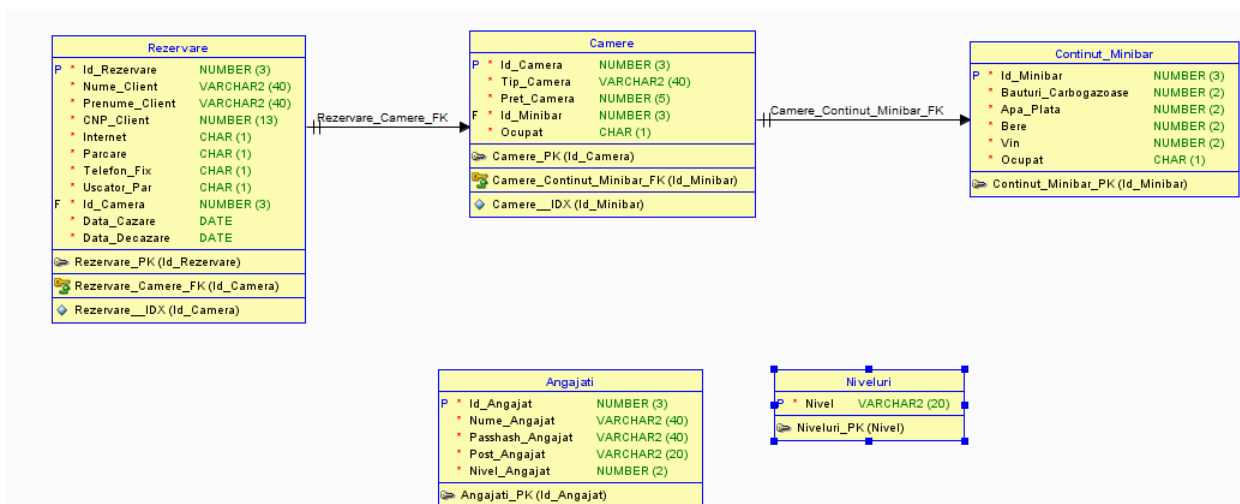
Diagrama de secvențe pentru cameristă



Modelul logic



Modelul relațional

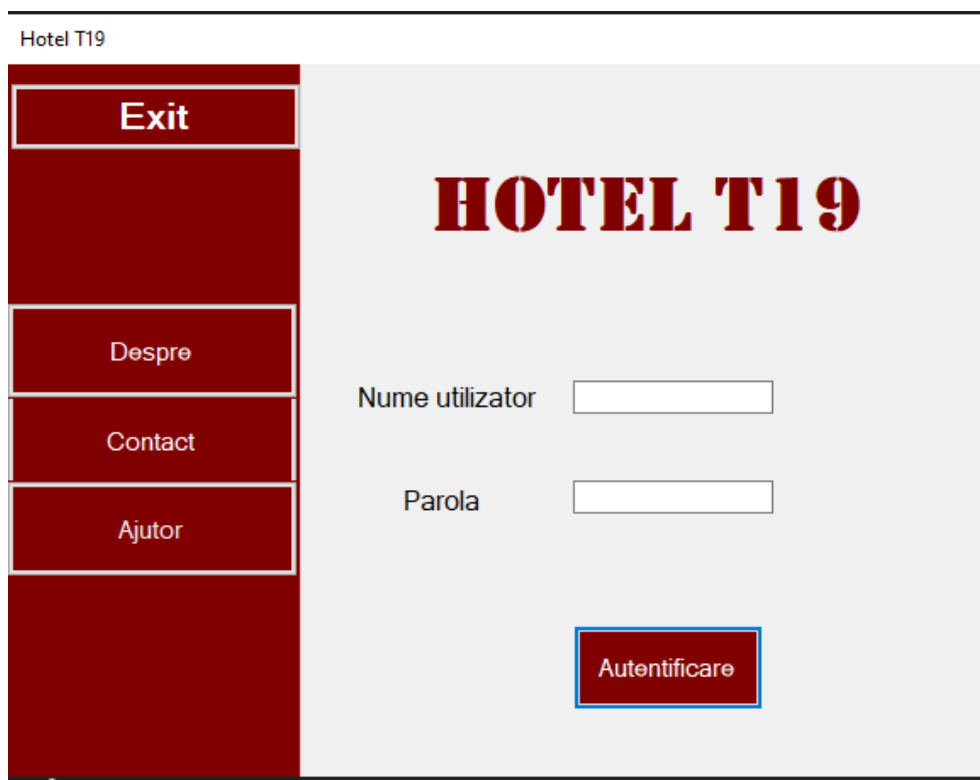


Modul de utilizare a programului

Aplicația își propune să ușureze munca angajaților unui hotel. Astfel totul va fi stocat în format electronic fără a exista posibilitatea de a încurca datele referitoare la rezervări, la camere sau la personalul hotelului.

Pagina Acasă

Când utilizatorul lansează în execuție aplicația acesta este direcționat pe pagina de *Acasă* unde are posibilitatea de a se conecta cu numele de utilizator și parola specifică postului pe care îl ocupă. Dacă nu se introduce un nume de utilizator sau o parolă validă acesta nu se va putea conecta și i se va cere să introducă un username sau o parolă corectă. Din pagina de *Acasă* utilizatorul poate vizualiza informații referitoare la funcționarea programului prin apăsarea butonului *Ajutor* care va deschide fișierul help asociat proiectului. Prin apăsarea butonului *Contact* se va deschide un Message Box care conține datele de contact ale hotelului iar prin apăsarea butonului *Despre* se va deschide un Message Box care conține informații referitoare la creatorii aplicației.



Hotel T19

Exit

HOTEL T19

Nume utilizator

Parola

Autentificare

Despre

Contact

Ajutor

Pagina Administrator

Dacă se realizează conectarea cu username-ul *admin* vom fi direcționați către o nouă pagină la care are acces doar administratorul hotelului. El poate vizualiza situația camerelor, fie

ele ocupate sau rezervate, poate gestiona situația camerelor adică poate adăuga noi camere care vor putea fi închiriate sau poate șterge camerele dacă nu dorește să le ofere spre închiriere. Administratorul se ocupă și cu gestionarea personalului hotelului. El poate adăuga noi angajați sau îi poate șterge pe cei vechi.

Administrator

Exit

Afișare camere
Inserare camere
Ștergere camere
Afișare angajați
Inserare angajați
Ștergere angajați

	ID_CAMERA	TIP_CAMERA	PRET_CAMERA	ID_MINIBAR	OCUPAT
▶	1	Single	150	1	0
	2	Single	150	2	0
	3	Single	150	3	0
	4	Single	150	4	0
	5	Single	150	5	0
	6	Single	150	6	0
	7	Doublex	250	7	0
	8	Doublex	250	8	0
	9	Doublex	250	9	0
	10	Doublex	250	10	0
	11	Doublex	250	11	0
	12	Doublex	250	12	0
*					

Afișare

Administrator

Exit

Afișare camere
Inserare camere
Ștergere camere
Afișare angajați
Inserare angajați
Ștergere angajați

	ID_ANGAJAT	NUME_ANGAJAT	PASSHASH_ANGAJAT	NIVEL_ANGAJAT	POST_ANGAJAT
▶	1	rec	t2maHK2YAY9g...	0	Receptionist
	2	cam	tBM8NtbYk9CHL...	1	Camerist
	3	admin	ODPiKuNlrVmD8...	-1	Administrator
*					

Afișare

Pagina Recepționeră

Dacă se realizează conectarea cu username-ul *rec* vom fi direcționați către pagina la care are acces recepționera hotelului. Funcția principală a recepționeriei este de a rezerva camere. Ea poate vedea situația camerelor înainte de a realiza rezervarea pentru a ști ce camere sunt disponibile și ce camere sunt deja rezervate. De asemenea, recepționera poate șterge rezervarea după ce se decazează clientul.

Recepționera

Exit

Camere ocupate

Camere disponibile

Inserare

Afișare

Ștergere

ID camera

1

CNP

2991212123123

Nume

Popa

Data cazare

21.05.2021

Prenume

Maria

Data decazare

25.05.2021

☐ Telefon fix

☒ Internet

☐ Uscător păr

☒ Parcare

Inserare

Recepționera

Exit

Camere ocupate

Camere disponibile

Inserare

Afișare

Ștergere

	ID_CAMERA	TIP_CAMERA	PRET_CAMERA	ID_MINIBAR
▶	1	Single	150	1
•				

Afișare

Pagina Cameristă

Dacă utilizatorul se conectează cu username-ul *cam* el va avea rolul de cameristă și va putea vedea ce camere sunt ocupate și ce camere sunt disponibile pentru a ști în ce camere trebuie să facă curat. După ce o cameră se eliberează, camerista actualizează conținutul minibarului și se asigură de faptul că totul este pregătit pentru a primi noi oaspeți în camera respectivă.

Camerista

Exit

Camere ocupate

Camere disponibile

Inserare

Afișare minibar

	ID_MINIBAR	BAUTURI_CARBO	APA_PLATA	BERE	VIN
▶	1	5	3	2	1
	2	5	3	2	1
	3	5	3	2	1
	4	5	3	2	1
	5	5	3	2	1
	6	5	3	2	1
	7	7	4	3	2
	8	7	4	3	2
	9	7	4	3	2
	10	7	4	3	2
	11	7	4	3	2
	12	7	4	3	2
	13	7	4	3	2
	14	7	4	3	2

<

Afișare

Anexă

AngajatProxy.cs

```
public bool Login(string utilizator, string parola)
{
    int flag = 0;
    string hash = Cryptography.Cryptography.HashString(parola);
    for (int i = 0; i < _utilizatori.Count; i++)
    {
        if (_utilizatori[i].Nume == utilizator &&
            _utilizatori[i].PassHash == hash)
        {
            _angajatReal = new AngajatReal(_utilizatori[i].NivelAcces);
            _utilizatorCurent = new User(_utilizatori[i].Nume,
                _utilizatori[i].PassHash, _utilizatori[i].NivelAcces,
                _utilizatori[i].Post);
            flag = 1;
            break;
        }
    }
    if (flag == 1)
        return true;
    else
        return false;
}
```

Metoda Login() permite autentificarea utilizatorului. Se verifică dacă numele de utilizator și parola criptată sunt corecte apoi se realizează conectarea.

```
private void SalveazaUtilizator()
{
    FunctiiDll.FunctiiDll.oracleConnection.ConnectionString = FunctiiDll.FunctiiDll.c;
    FunctiiDll.FunctiiDll.oracleConnection.Open();
    try
    {
        OracleDataAdapter oracleDataAdapter = new OracleDataAdapter();
        string insert = "INSERT INTO ANGAJATI (NUME_ANGAJAT, PASSHASH_ANGAJAT,
            NIVEL_ANGAJAT, POST_ANGAJAT) VALUES('"
            + _utilizatori[_utilizatori.Count - 1].Nume + "', '"
            + _utilizatori[_utilizatori.Count - 1].PassHash + "', '"
            + _utilizatori[_utilizatori.Count - 1].NivelAcces + "', '"
            + _utilizatori[_utilizatori.Count - 1].Post + "')";

        oracleDataAdapter.InsertCommand = new OracleCommand(insert,
            FunctiiDll.FunctiiDll.oracleConnection);
        oracleDataAdapter.InsertCommand.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
    }
```

```

    {
        FunctiiDll.FunctiiDll.oracleConnection.Close();
    }
}

```

Metoda SalveazaUtilizator salvează în baza de date noul utilizator introdus. În primă fază se realizează conexiunea cu baza de date, apoi se verifică dacă s-a realizat conexiunea. Dacă a fost realizată cu succes conexiunea se introduc în baza de date noile valori (numele, parola, nivelul de acces și postul angajatului), altfel se generează o excepție.

Receptionera.cs

```

public bool InserareRezervare()
{
    FunctiiDll.FunctiiDll.oracleConnection.ConnectionString = FunctiiDll.FunctiiDll.c;
    FunctiiDll.FunctiiDll.oracleConnection.Open();
    try
    {
        OracleDataAdapter oracleDataAdapter = new OracleDataAdapter();
        string insert = "INSERT INTO REZERVARE(ID_CAMERA, NUME_CLIENT,
            PRENUME_CLIENT,CNP_CLIENT, DATA_CAZARE,
            DATA_DECAZARE, INTERNET,PARCARE,"
            + "TELEFON_FIX,USCATOR_PAR) " + "VALUES(' "
            + IdCamera.Trim().ToString() + "', '"
            + Nume.Trim().ToString() + "', '"
            + Prenume.Trim().ToString() + "', '"
            + Cnp.Trim().ToString() + "', '"
            + DataCazare.Trim().ToString() + "', '"
            + DataDecazare.Trim().ToString() + "', '"
            + Internet + "', '"
            + Parcare + "', '"
            + TelefonFix + "', '"
            + UscatorPar + "')";
        oracleDataAdapter.InsertCommand = new OracleCommand(insert,
            FunctiiDll.FunctiiDll.oracleConnection);

        oracleDataAdapter.InsertCommand.ExecuteNonQuery();

        string update = "UPDATE CAMERE " + "SET OCUPAT=1"
            + "WHERE ID_CAMERA=" + IdCamera;
        oracleDataAdapter.UpdateCommand = new OracleCommand(update,
            FunctiiDll.FunctiiDll.oracleConnection);

        oracleDataAdapter.UpdateCommand.ExecuteNonQuery();
        return true;
    }
    catch (Exception ex)
    {
        MessageBox.Show("Invalid data!");
        return false;
    }
    finally
    {
        FunctiiDll.FunctiiDll.oracleConnection.Close();
        refreshAllComboBoxes();
    }
}

```

```
}  
}
```

Metoda `InsereazaRezervare()` permite introducerea de noi rezervări în baza de date. În primă fază se realizează conexiunea cu baza de date, apoi se verifică dacă s-a realizat conexiunea. Dacă a fost realizată cu succes conexiunea se introduc în baza de date valorile pentru rezervare după care se actualizează valoarea variabilei "ocupat" (trece din 0 în 1) și se returnează valoarea "true", altfel se generează o excepție și se returnează "false".

Admin.cs

```
public bool StergeCamere()  
{  
    FunctiiDll.FunctiiDll.oracleConnection.ConnectionString = FunctiiDll.FunctiiDll.c;  
    FunctiiDll.FunctiiDll.oracleConnection.Open();  
    try  
    {  
        OracleDataAdapter oracleDataAdapter = new OracleDataAdapter();  
        string select = "SELECT ID_MINIBAR FROM CAMERE WHERE ID_CAMERA=" + IdCamera;  
        OracleCommand command = new OracleCommand(select,  
            FunctiiDll.FunctiiDll.oracleConnection);  
        string id_minibar = command.ExecuteScalar().ToString();  
        string update = "UPDATE CONTINUT_MINIBAR " +  
            "SET OCUPAT=0 " +  
            "WHERE ID_MINIBAR=" + id_minibar;  
        oracleDataAdapter.UpdateCommand = new OracleCommand(update,  
            FunctiiDll.FunctiiDll.oracleConnection);  
        oracleDataAdapter.UpdateCommand.ExecuteNonQuery();  
        string delete = "DELETE FROM CAMERE WHERE ID_CAMERA=" + IdCamera;  
        oracleDataAdapter.DeleteCommand = new OracleCommand(delete,  
            FunctiiDll.FunctiiDll.oracleConnection);  
        oracleDataAdapter.DeleteCommand.ExecuteNonQuery();  
        return true;  
    }  
    catch (Exception ex)  
    {  
        MessageBox.Show("Invalid id!");  
        return false;  
    }  
    finally  
    {  
        FunctiiDll.FunctiiDll.oracleConnection.Close();  
        refreshAllComboBoxes();  
    }  
}
```

Metoda `StergeCamere()` permite ștergerea camerelor disponibile din baza de date. În primă fază se realizează conexiunea cu baza de date, apoi se verifică dacă s-a realizat conexiunea. Dacă a fost realizată cu succes conexiunea se actualizează starea minibarului (variabila ocupat ia valoarea 0) apoi se șterge din baza de date camera și se returnează valoarea "true", altfel se generează o excepție și se returnează "false".