**Proiect Hotel**

**Tehnici Avansate de Programare**

**Boboc Adrian-Valentin**

**1. Introducere**

**Scop:**  
Aplicatia pentru hotel are ca scop gestionarea eficienta a rezervarilor, cazarii, istoricului cazarilor, camerelor si clientilor, automatizand procesele administrative. Aceasta permite introducerea, modificarea si stergerea rezervarilor, urmarirea statusului camerelor si calcularea preturilor in functie de numarul de zile si tipul camerei.

**Tehnologii utilizate:**

* **Limbaj de programare:** C#
* **IDE:** Visual Studio
* **Controale UI:** DateTimePicker, ComboBox, TextBox, Button
* **Baza de date:** Microsoft Access
* **Conectare:** OleDB.12
* **Validari:** Campuri obligatorii, format numeric, integritate referentiala, format text, format data

**2. Structura aplicatiei**

**Entitati si relatii:**

* **Clienti** (IdClient, NumeClient, NrTelefon)
* **Camere** (IdCamera, NrCamera, NrLocuri, Etaj, PretZi, SpImagine)
* **Rezervari** (IdRezervare, DataRezervarii, IdClient)
* **RezervariContinut** (IdRezervare, Nrc, IdCamera, DataCazarii, PretZi, NrZile)
* **CazariContinut** (IdCazare, Nrcrt, IdRezervare, DataCazare, DataPlecare, IdCamera, PretZi, NumarCamera)
* **IstoricClienti** (IdTranzactie, IdClient, IdCamera, PretZi, DataCazarii, DataPlecarii, Total)
* **Users** (Id, Username, Password)

**Relatii:**

1. **Clienti ↔ Rezervari**  
   Fiecare client poate face mai multe rezervari.  
   Relatia este stabilita prin **IdClient** in tabela **Rezervari**.
2. **Rezervari ↔ RezervariContinut**  
   O rezervare poate include mai multe camere si informatii despre cazare.  
   Relatia este stabilita prin **IdRezervare** in tabela **RezervariContinut**.
3. **Camere ↔ RezervariContinut**  
   Fiecare camera poate fi rezervata in mai multe rezervari.  
   Relatia este stabilita prin **IdCamera** in tabela **RezervariContinut**.
4. **Rezervari ↔ CazariContinut**  
   O rezervare poate genera o cazare, cu detalii despre data cazarii si plecarii.  
   Relatia este stabilita prin **IdRezervare** in tabela **CazariContinut**.
5. **Camere ↔ CazariContinut**  
   O camera poate fi asociata cu mai multe cazari.  
   Relatia este stabilita prin **IdCamera** in tabela **CazariContinut**.
6. **Clienti ↔ IstoricClienti**  
   Fiecare client poate avea un istoric de tranzactii (cazari anterioare).  
   Relatia este stabilita prin **IdClient** in tabela **IstoricClienti**.
7. **Camere ↔ IstoricClienti**  
   Fiecare camera poate aparea in istoricul mai multor clienti.  
   Relatia este stabilita prin **IdCamera** in tabela **IstoricClienti**.
8. **Users ↔ Clienti**  
   Fiecare utilizator (admin) poate gestiona datele clientilor hotelului.  
   Relatia poate fi indirecta, iar utilizatorii pot avea permisiuni asociate cu gestionarea clientilor, dar nu este o relatie directa in modelul de date.

**2. Interfete si formulare**

**Formular Start - Descriere:**

Acest formular este utilizat pentru a permite unui utilizator să se autentifice într-o aplicație de gestionare a hotelului. Acesta folosește un username și o parolă pentru a valida accesul. Dacă autentificarea este validă, utilizatorul este redirecționat către un alt formular pentru gestionarea datelor. Formularul utilizează o bază de date Access pentru a valida informațiile de conectare.

Descrierea elementelor UI:

1. TextBox pentru username (txtUsername)
   * Permite utilizatorului să introducă numele de utilizator.
2. TextBox pentru parolă (txtPassword)
   * Permite utilizatorului să introducă parola. Aceasta poate fi setată cu PasswordChar pentru a ascunde caracterele.
3. Button pentru conectare (btnConnect)
   * Acesta este butonul care declanșează procesul de verificare a autentificării atunci când este apăsat.
4. Label pentru titlu (Title)
   * Un label care poate conține un titlu precum "EHotel".
5. Label pentru autor (lblAutor)
   * O etichetă care afiseaza numele autorului.
6. MenuStrip sau alte controale pentru navigare (opțional)
   * Poate conține opțiuni de navigare sau pentru gestionarea altor funcționalități, cum ar fi "Log in".

Am creat o legatura catre baza de date Access folosind tehnologia **OLE DB**. Aceasta legatura permite aplicatiei sa se conecteze la baza de date si sa efectueze interogari pentru a valida datele de autentificare ale utilizatorului.

In codul de mai jos, legatura catre baza de date este stabilita in metoda btnConnect\_Click folosind string-ul de conexiune:

private string connectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Users\win\source\repos\ProiectHotel\AplicatieHotel.accdb";

Acesta specifica:

* **Provider**: tipul de furnizor al bazei de date (in acest caz, Microsoft.ACE.OLEDB.12.0 pentru Access).
* **Data Source**: calea catre fisierul bazei de date .accdb.

Dupa stabilirea conexiunii prin OleDbConnection, aplicatia verifica autentificarea utilizatorului printr-o interogare SQL:

string query = "SELECT COUNT(\*) FROM users WHERE username = @username AND password = @password";

Aceasta interogare cauta in tabela users pentru a valida daca exista un utilizator cu username-ul si parola corespunzatoare. In cazul in care autentificarea este valida, utilizatorul este redirectionat catre urmatorul formular al aplicatiei.

**Formular Principal - Descriere:**

Formularul Form2 este utilizat pentru gestionarea si actualizarea rezervarilor clientilor intr-o aplicatie hoteliera. Acesta contine urmatoarele elemente principale:

1. **Vizualizarea datelor de rezervare**:
   * Un tabel (DataGridView) care afiseaza informatiile despre rezervarile curente, inclusiv numele clientului, numarul de telefon, data cazarii, data plecarii, pretul pe zi, numarul camerei si numarul de zile de cazare.
2. **Functionalitati de actualizare a datelor**:
   * Permite utilizatorului sa modifice datele (de exemplu, numele clientului, numarul de telefon, data plecarii, numarul camerei etc.) si sa salveze modificarile in baza de date.
3. **Validarea datelor**:
   * Asigura ca informatiile introduse sunt corecte, cum ar fi validarea numarului de telefon si a datelor, precum si verificarea disponibilitatii camerei pentru datele alese.
4. **Cautare si filtrare a datelor**:
   * Permite utilizatorilor sa caute rezervari pe baza numelui clientului sau a numarului de telefon, filtrand astfel rapid datele afisate in tabel.
5. **Navigare intre formulare**:
   * Ofera posibilitatea de a naviga la alte formulare, cum ar fi cele pentru check-in si check-out, pentru a continua gestionarea proceselor hotelului.
6. **Butoane de control**:
   * **Actualizare**: Permite trecerea la modul de editare.
   * **Salvare**: Salveaza modificarile facute de utilizator in baza de date.
   * **Renuntare**: Ofera optiunea de a anula modificarile efectuate.

**Funcțiile principale incluse:**

* LoadData: Încarcă datele inițiale din baza de date.
* ConfigureDataGridView: Configurează comportamentul DataGridView, făcându-l citibil sau editabil în funcție de acțiunea utilizatorului.
* SaveDataChanges: Salvează modificările efectuate de utilizator în baza de date, actualizând atât tabelul de clienți, cât și rezervările.
* IsRowModified: Verifică dacă un rând a fost modificat comparativ cu datele originale.
* IsPhoneNumberValid: Verifică dacă numărul de telefon introdus este valid.
* IsCameraAvailable: Verifică dacă o cameră este disponibilă pentru o anumită dată.

Am implementat acest formular conform modelului de la cursul 5 cu form vizualizare si actualizare date.In plus am adaugat functionalitati care faciliteaza munca utilizatorului putand naviga mai usor intre form uri si avand validari pentru toate campurile de modificare sau inserare.

**Formular 3- Descriere:**

Formularul **Form3** din aplicația hotelieră are scopul principal de a vizualiza, căuta, și gestiona rezervările existente, cu funcționalități suplimentare pentru finalizarea cazărilor și navigarea către alte formulare

**Functionalitati principale ale Form3**

1. **Vizualizarea datelor din baza de date**:
   * Metoda LoadData incarca informatiile despre rezervari din baza de date AplicatieHotel.accdb si le afiseaza intr-un tabel (DataGridView).
   * Datele afisate includ:
     + **Nume client**
     + **Numar de telefon**
     + **Data cazarii**
     + **Pret/zi**
     + **Numar zile**
     + **Numar camera**
     + **IdRezervare** (utilizat pentru identificare interna).
2. **Cautarea rezervarilor**:
   * Butonul **Cautare** permite utilizatorilor sa filtreze rezervarile pe baza numelui clientului.
   * Termenul de cautare este preluat dintr-un camp text (txtCautare), iar rezultatele sunt afisate in tabel dupa rularea unui query SQL cu filtrare.
3. **Finalizarea cazarii**:
   * Butonul **Finalizeaza cazare** deschide formularul **Form7** pentru gestionarea cazarii selectate.
   * Datele sunt preluate din randul selectat al tabelului:
     + **Nume client**, **Numar telefon**, **Data cazarii**, **Numar zile**, **Numar camera**, **IdRezervare**, si **Pret/zi**.
   * Calcularea **datei plecarii**:
     + Data plecarii se calculeaza pe baza datei cazarii si a numarului de zile (DateTime.AddDays).
   * Datele sunt transmise catre **Form7**, care poate fi utilizat pentru a finaliza rezervarea sau pentru a adauga detalii suplimentare.
4. **Navigare intre formulare**:
   * Meniul permite accesul rapid la alte sectiuni ale aplicatiei:
     + **Home** (Form2)
     + **Camere** (Form4)
     + **Rezervari** (Form6).
   * Navigarea inchide formularul curent si deschide formularul selectat.
5. **Gestionarea erorilor**:
   * Blocurile try-catch sunt utilizate pentru a preveni erorile in cazul in care conexiunea la baza de date sau query-ul SQL esueaza.

**Query-ul utilizat pentru incarcarea datelor**

**„SELECT**

**C.NumeClient AS [Nume],**

**C.NrTelefon AS [Telefon],**

**RC.DataCazarii AS [Data Cazarii],**

**RC.PretZi AS [Pret/zi],**

**RC.NrZile AS [Numar Zile],**

**Ca.NrCamera AS [Numar Camera],**

**RC.IdRezervare**

**FROM**

**((Clienti AS C**

**INNER JOIN**

**Rezervari AS R ON C.IdClient = R.IdClient)**

**INNER JOIN**

**RezervariContinut AS RC ON R.IdRezervare = RC.IdRezervare)**

**INNER JOIN**

**Camere AS Ca ON RC.IdCamera = Ca.IdCamera ”**

* **Selecteaza coloanele**:
  + C.NumeClient (numele clientului).
  + C.NrTelefon (numarul de telefon).
  + RC.DataCazarii, RC.PretZi, RC.NrZile (detalii despre rezervare).
  + Ca.NrCamera (numarul camerei).
  + RC.IdRezervare (id-ul rezervarii pentru referinte).
* **Tabele implicate**:
  + Clienti (informatii despre clienti).
  + Rezervari (legatura intre clienti si continutul rezervarilor).
  + RezervariContinut (detalii despre camere si cazare).
  + Camere (detalii despre camere).
* **Legaturi (JOIN-uri)**:
  + Se realizeaza legaturi intre tabele pe baza cheilor straine: IdClient, IdRezervare, si IdCamera.

Daca utilizatorul introduce un termen de cautare, query-ul este completat cu un filtru:

„WHERE C.NumeClient LIKE @searchTerm”

Acest filtru returneaza doar randurile unde NumeClient contine termenul introdus.

**Rolul Form3 in aplicatie**

Formularul **Form3** este un punct central pentru gestionarea rezervarilor:

* Permite vizualizarea rapida a tuturor rezervarilor curente.
* Ofera optiuni de cautare si filtrare pentru utilizator.
* Conecteaza utilizatorii la procesul de finalizare a cazarii prin deschiderea **Form7**.

**Formular 4- Descriere:**

Formularul **Form4** este responsabil pentru gestionarea camerelor din aplicatia hoteliera. Acesta ofera functionalitati de adaugare, modificare, stergere, si vizualizare a datelor despre camerele disponibile in hotel.

**Functionalitati principale ale Form4**

1. **Vizualizarea datelor despre camere**:
   * Metoda LoadCamereData incarca si afiseaza toate camerele din baza de date AplicatieHotel.accdb.
   * Datele afisate in tabel includ informatii precum:
     + **Numarul camerei**
     + **Numarul de locuri**
     + **Etajul**
     + **Pretul pe zi**
     + **Calea imaginii camerei**
2. **Adaugarea unei camere noi**:
   * Utilizatorul poate completa detalii despre camera:
     + **Numar camera**, **Numar locuri**, **Etaj**, **Pret pe zi**, si **Imagine**.
   * Prin metoda InserareCamera, aceste informatii sunt salvate in baza de date.
3. **Modificarea unei camere existente**:
   * Selectand o camera din tabel, utilizatorul poate edita detaliile acesteia.
   * Metoda BtnConfirmare\_Modificare actualizeaza baza de date pe baza ID-ului camerei selectate.
   * Se efectueaza validari pentru a preveni erorile (ex. numar camera invalid sau imagine inexistenta).
4. **Stergerea unei camere**:
   * Selectand o camera din tabel, utilizatorul poate sterge aceasta in baza de date.
   * Operatiunea este gestionata de metoda btnStergere\_Click, cu o confirmare prealabila.
5. **Selectarea unei imagini pentru camera**:
   * Utilizatorul poate adauga o imagine camerei utilizand un dialog (OpenFileDialog).
   * Imaginea este afisata intr-un control PictureBox.
6. **Navigarea intre formulare**:
   * Formularul permite accesul rapid catre alte sectiuni ale aplicatiei:
     + **Home** (Form2).
     + **Rezervari** (Form6).
7. **Gestionarea erorilor**:
   * Se folosesc blocuri try-catch pentru a prinde erorile ce pot aparea in timpul interactiunii cu baza de date sau a altor operatiuni.

**Structura formularului si fluxul operatiunilor**

1. **Initializare**:
   * ComboBox-ul pentru etaje este populat cu valori predefinite (1-4).
   * Tabelul (DataGridView) este initializat cu datele din baza de date.
2. **Adaugare**:
   * Se apasa butonul **Adaugare**, se completeaza detaliile camerei, iar datele sunt validate si inserate in tabela.
3. **Modificare**:
   * Se selecteaza o camera din tabel, detaliile sale sunt afisate in campuri, iar modificarile sunt salvate dupa confirmare.
4. **Stergere**:
   * Se selecteaza o camera din tabel, iar dupa confirmare, aceasta este stearsa din baza de date.
5. **Resetare campuri**:
   * Dupa fiecare operatie, campurile sunt resetate pentru a asigura o experienta curata pentru utilizator.

**Formular 5- Descriere:**

Formularul **Form5** este o componenta esentiala a aplicatiei, responsabila pentru gestionarea operatiunilor de cautare a rezervarilor si procesarea check-out-ului clientilor. Scopul principal al acestui formular este de a permite utilizatorului sa identifice rapid clientii cazati si sa finalizeze procesul de cazare, cu toate datele fiind gestionate corespunzator in baza de date.

**Functionalitati principale:**

1. **Cautarea clientilor si rezervarilor**
   * Utilizatorul poate cauta clienti sau rezervari utilizand un criteriu precum:
     + Numele clientului.
     + Numarul camerei.
     + Data cazarii sau data plecarii.
   * Rezultatele sunt afisate intr-un tabel (DataGridView) cu informatii detaliate despre:
     + Numele clientului.
     + Numarul camerei.
     + Pretul pe zi.
     + Datele de cazare si plecare.
     + Numarul de zile si totalul de plata.
   * Cautarea este flexibila, utilizand caractere wildcard (%) pentru a permite potrivirea partiala a criteriilor introduse.
2. **Procesarea Check-Out-ului**
   * Utilizatorul poate selecta un client din lista afisata pentru a realiza procesul de check-out.
   * Operatiunea de check-out implica:
     + **Transferarea datelor clientului in tabelul IstoricClienti, pastrand o evidenta a sederii.**
     + **Stergerea inregistrarilor asociate clientului din tabelele active**:
       - CazariContinut
       - Rezervari
       - Clienti
   * Dupa check-out, clientul este eliminat din tabelul afisat in interfata.
3. **Generarea bonului de check-out**
   * Dupa finalizarea procesului, se creeaza un bon de check-out care contine:
     + Numele clientului.
     + Numarul camerei.
     + Pretul pe zi, datele de cazare si plecare.
     + Totalul de plata.
   * Utilizatorul poate opta sa imprime bonul utilizand functionalitatea integrata de printare.
4. **Navigarea intre formulare**
   * Meniul permite navigarea rapida catre alte parti ale aplicatiei:
     + Home (Form2)
     + Camere (Form4)
     + Rezervari (Form6)

**Formular 6- Descriere:**

Acest formular (**Form6**) face parte dintr-o aplicatie de gestionare a unui hotel si permite utilizatorilor sa adauge sau sa modifice rezervari de camere pentru clienti. Interfata include campuri pentru completarea detaliilor clientului (nume, numar de telefon), numarul de camere si datele rezervarii (data cazarii, numarul de zile).

**Functionalitati principale:**

1. **Adaugare rezervare:**
   * Permite adaugarea unei noi rezervari, inclusiv salvarea informatiilor clientului si detaliilor rezervarii.
   * Verifica disponibilitatea camerei inainte de a salva rezervarea.
2. **Modificare rezervare:**
   * Permite modificarea unei rezervari existente, inclusiv actualizarea datelor clientului si camerei.
   * Verifica disponibilitatea camerei pentru data selectata.
   * Actualizeaza toate tabelele relevante din baza de date (Clienti, Rezervari, RezervariContinut).
3. **Validari input:**
   * Verifica validitatea datelor introduse de utilizator, cum ar fi numarul de telefon, data cazarii, numarul de zile si numarul camerei.
   * Se asigura ca data cazarii nu este in trecut si ca numarul de zile si camerelor este valid.
4. **Interactiune cu baza de date:**
   * Conecteaza la o baza de date Access pentru a salva si actualiza datele in tabelele Clienti, Rezervari, Camere si RezervariContinut.
   * Foloseste comenzi SQL pentru a insera sau actualiza informatiile despre rezervari si clienti.
5. **UI dinamica:**
   * Permite utilizatorului sa comute intre modurile de adaugare si modificare a rezervarilor, ascunzand sau afisand butoanele corespunzatoare pentru fiecare actiune.
   * Afiseaza rezervarile curente intr-un DataGridView pentru a facilita selectarea unei rezervari pentru modificare.

**Formular 7- Descriere:**

**Form7** gestioneaza finalizarea unei cazari pentru un client, preluand datele dintr-o rezervare existenta sau lasandu-l pe utilizator sa introduca manual informatiile pentru a finaliza procesul de cazare.

 **Constructorul Form7 cu parametri**:

* Daca **rezervarea** exista, datele (nume, telefon, datele cazarii si plecarii) sunt preluate din **Form3**.
* Se valideaza si se afiseaza datele (nume, telefon, numar zile, numar camera, datele cazarii si plecarii).
* Daca datele de cazare si plecare sunt valide, acestea sunt setate in controalele DateTimePicker.

 **Confirmarea cazarii (btnConfirmare\_Click)**:

* Se verifica existenta clientului in baza de date. Daca nu exista, se afiseaza un mesaj de eroare.
* Daca clientul exista, se preiau si se valideaza datele, inclusiv pretul pe zi.
* Cazarea este adaugata in **CazariContinut**, iar rezervarea este stearsa din tabelele **Rezervari** si **RezervariContinut**.

 **Stergerea rezervarii (DeleteRezervari)**:

* Dupa confirmarea cazarii, rezervarea este stearsa din baza de date (din tabelele **Rezervari** si **RezervariContinut**).

 **Functii de acces la baza de date**:

* GetClientId: Gaseste ID-ul clientului pe baza numelui si telefonului.
* GetIdRezervare: Gaseste ID-ul rezervarii pe baza ID-ului clientului.
* GetNextNrCrt: Obține urmatorul numar de ordine pentru cazare.
* AddCazareContinut: Adauga cazarea finalizata in **CazariContinut**.

 **Renuntare (btnRenuntare\_Click)**:

* Daca utilizatorul doreste sa renunte, aplicatia revine la **Form3** fara a salva datele cazarii.

**4.Functionalitati si Validari Obligatorii**

Aplicația pentru gestionarea cazarii intr-un hotel are mai multe formulare (Form1, Form2, Form3, Form4, Form5, Form6, Form7), fiecare avand functii specifice care ajuta utilizatorul sa interactioneze cu baza de date si sa administreze clientii, camerele, rezervarile si cazarea acestora. Mai jos sunt prezentate functionalitatile fiecarui formular si validările implementate.

**Controale specifice utilizate:**

* **DateTimePicker:** Folosit pentru a selecta datele de cazare si plecare ale clientului. Este validat pentru a asigura ca data de plecare nu este anterioara datei de cazare.
* **ComboBox:** Permite selectia cheilor externe, cum ar fi ID-ul camerei dintr-o lista de camere disponibile. Afiseaza informatii semantice (numele camerei, tipul acesteia, disponibilitatea) pentru a ajuta utilizatorul sa aleaga corect camera.
* **PictureBox:** Permite incarcare si vizualizarea imaginilor, cum ar fi poze ale camerelor sau ale hotelului.
* **TextBox:** Folosit pentru introducerea datelor de baza, precum numele clientului, numarul de telefon, numarul camerei, etc. Aceste campuri sunt validate pentru a asigura corectitudinea formatului (de exemplu, numerele trebuie sa aiba un format numeric corect).

**2. Validari obligatorii:**

* **Format numeric pentru campurile numerice:**
  + **Numar de telefon:** Verifica daca numarul introdus respecta formatul numeric corect.
  + **Numarul de zile:** Trebuie sa fie un numar intreg pozitiv**.**
  + **Pret pe zi:** Este validat pentru a fi un numar zecimal
* **Campuri obligatorii:** Campurile esentiale, cum ar fi numele clientului, telefonul, numarul camerei si data cazarii, sunt obligatorii. Daca utilizatorul nu le completeaza corect, aplicatia va afisa un mesaj de eroare si nu va permite completarea formularului.
* **Unicitatea cheii semantice:**
  + ID-ul clientului si ID-ul rezervarii sunt validate pentru a fi unice in baza de date, evitand astfel duplicarea acestora.
* **Integritate referentiala la stergere:** Daca utilizatorul incearca sa stearga un client sau o rezervare care este legata de alte inregistrari (de exemplu, o rezervare asociata cu o cazare), aplicatia va afisa un mesaj de avertisment pentru a preveni stergerea accidentala a datelor.
* **Confirmare la stergerea inregistrarilor:** Cand utilizatorul incearca sa stearga o inregistrare (de exemplu, un client sau o rezervare), aplicatia cere confirmare pentru a preveni stergerea accidentala**.**
* **Avertisment de eroare:** Orice eroare de completare a formularului (de exemplu, un numar de telefon invalid sau o data incorecta) va genera un mesaj de eroare pentru utilizator. De asemenea, orice eroare de conexiune la baza de date va fi afisata intr-un mesaj clar.

**3. Functionalitati implementate in fiecare forma:**

**Form1 - Logare:**

* Formul care permite autentificarea utilizatorului in aplicatie.
* Utilizatorul trebuie sa completeze un nume de utilizator si o parola.
* La apasarea butonului de logare, aplicatia verifica daca datele de autentificare sunt corecte comparand cu baza de date.
* Daca autentificarea este reusita, utilizatorul este redirectionat catre Form2, care reprezinta pagina principala a aplicatiei, unde sunt accesibile toate functionalitatile.

**Form2 - Gestionare cazari (pagina principala):**

* Este pagina principala din care utilizatorul poate accesa toate functionalitatile aplicatiei.
* Permite gestionarea cazarilor clientilor, adaugarea de clienti noi, vizualizarea rezervarilor si cazarii acestora.
* In acest formular, utilizatorul poate accesa sectiunile pentru Form4 (Adaugare client / vizualizare clienti), Form5 (Gestionare camere), Form6 (Gestionare rezervari), si Form7 (Finalizarea cazarii).
* Fiecare dintre aceste sub-formulare este accesibil din Form2, facand astfel gestionarea hotelului mai eficienta.
* Validarea rezervarilor si a datelor de cazare se face direct din aceasta pagina.

**Form3 - Finalizarea cazarii:**

* Permite finalizarea cazarii unui client, adaugandu-l in tabelul CazariContinut.
* Daca clientul are o rezervare existenta, informatiile se preiau automat; daca nu are, utilizatorul trebuie sa completeze manual campurile.
* Este validat daca clientul exista deja in baza de date si daca rezervarea este corecta.
* Daca clientul nu exista in baza de date, aplicatia afiseaza un mesaj de eroare.
* La finalizarea cazarii, daca rezervarea a fost facuta, aceasta este stearsa din tabelul Rezervari si RezervariContinut pentru a marca procesul de cazare complet.

**Form4 - Adaugare client / vizualizare clienti:**

* Permite utilizatorilor sa adauge clienti noi in baza de date.
* Verifica unicitatea numarului de telefon pentru a preveni duplicarea.
* Permite cautarea si vizualizarea clientilor existenti in baza de date.
* Afiseaza un avertisment daca utilizatorul incearca sa stearga un client care are rezervari active.

**Form5 - Gestionare camere:**

* Permite gestionarea camerelor (adaugarea, actualizarea disponibilitatii acestora).
* Camerele sunt afisate intr-un ComboBox cu informatii detaliate (tipul camerei, pret, disponibilitate).
* Validarea pretului pe zi si a disponibilitatii camerei.
* Confirmare inainte de stergerea unei camere care poate fi deja rezervata.

**Form6 - Gestionare rezervari:**

* Permite crearea si gestionarea rezervarilor pentru clienti.
* Verifica disponibilitatea camerei selectate si calculeaza automat pretul total pe baza numarului de zile si a pretului pe zi.
* Permite confirmarea sau anularea rezervarii.
* Validare referentiala pentru a asigura corectitudinea datelor (de exemplu, clientul trebuie sa existe in baza de date pentru a putea face o rezervare).

**Form7 - Finalizarea cazarii:**

* Permite finalizarea cazarii, adaugand clientul in tabelul CazariContinut.
* Daca clientul are o rezervare, informatiile sunt preluate automat; daca nu, utilizatorul trebuie sa completeze campurile manual.
* Daca clientul nu exista in baza de date, aplicatia va afisa un mesaj de eroare.
* **Confirmarea si stergerea rezervarii din tabelele Rezervari si RezervariContinut.**

**Concluzie:**

Aceasta aplicatie de gestionare a cazarilor unui hotel integreaza mai multe forme si controale specifice, fiecare avand un rol bine definit in procesul de gestionare a clientilor, camerelor, rezervarilor si cazarii. Fiecare formular implementeaza validari pentru a asigura integritatea datelor si pentru a preveni erorile umane, iar interfata este conceputa pentru a oferi o experienta intuitiva utilizatorului.

**5.Implementare**

**Diagrame de stari:**

**1.**Pentru legarea form1 si form 2(logare) am folosit aceasta diagrama de stari din Cursul 3

:

A diagram of a software

Description automatically generated

**2.**Pentru form 2 am folosit aceasta diagrama de stari din Cursul 5

A diagram of a flowchart

Description automatically generated

Intitial grid-ul este doar pentru vizualizare.Utilizatorul un buton actualizare,iar daca il actioneaza vor fi vizibile butoanele salvare si renuntare.Am folosit acest model pentru form 2(principal) pentru afisarea tabelului persoanelor cazate pentru a fi mai usor de gestionat de utilizator si a permite actualizarea doar la actionarea butonului **actualizeaza.**

**3.**Pentru form 4 si form 6 am folosit diagrama de stari din Cursul 7

A diagram of a program

Description automatically generated

Aceasta cotine un grid si un panel si 3 butoane principale adaugare,modificare si stergere iar la actionarea acestora cand se va alege una dintre optiuni vor aparea butoane de confirmare sau renuntare pentru optiunea principala.Cand se va apasa renuntare se revine la inceput,butoanele principale devin vizibile,nefiind nicio optiune selectata.

Am folosit majoritatea modelelor implementate la curs si laborator deoarece am incercat sa combin cat mai multe tehnici si functii care pot usura munca utilizatorilor intr-o aplicatie de gestionare a rezervarilor. Aplicatia dezvoltata are rolul de a facilita rezervarea si gestionarea camerelor, inregistrarea clientilor, monitorizarea istoricului cazarii si a statelor acestora. Am integrat functionalitati care permit utilizatorilor sa vizualizeze disponibilitatea camerelor, sa efectueze rezervari, sa gestioneze datele clientilor si sa acceseze istoricul cazarii, iar acest lucru se face intr-un mod simplu si eficient.