

Documentația de aplicare a agenției de închiriere auto

Intrare

Această documentație explică utilizarea și caracteristicile cheie ale unei aplicații simple de agenție de închiriere de mașini. Această aplicație este dezvoltată pentru a gestiona închirierea mașinilor clienților.

Interfața cu utilizatorul

Aplicația include interfața cu utilizatorul cu următoarele caracteristici cheie:

1. Acasă:

- Oferă un formular de conectare pentru înregistrarea clientului sau autentificare.
- Oferă informații despre vehiculele disponibile.

2. Înregistrarea clientului:

- Oferă un formular de înregistrare pentru noi clienți să se înscrie.
- Poate fi necesar să introduceți în formular informații precum numele utilizatorului, adresa de e-mail, numărul de telefon.

3. Introducere:

- Oferă un formular de conectare pentru ca clienții înregistrați să se autentifice.
- Conectarea poate necesita acreditări, cum ar fi numele de utilizator sau e-mailul și parola.

4. Pagina Instrumente:

- Oferă utilizatorilor o listă pentru a vedea vehiculele disponibile.
- Pentru fiecare vehicul, pot fi afișate informații precum marca, modelul, anul, tipul de combustibil, taxa zilnică de închiriere.
- Clienții pot face clic pe un link pentru a vedea detaliile vehiculelor și pot trimite o cerere pentru procesul de închiriere.

5. Pagina cu detaliile vehiculului:

- Afișează detaliile unui vehicul selectat.
- Detaliile pot include fotografii ale vehiculului, specificații tehnice și taxa zilnică de închiriere.
- Clienții pot completa un formular pentru a trimite o cerere de închiriere a vehiculului. În formular, pot fi solicitate informații precum data și durata închirierii.

6. Pagina de cereri de închiriere:

- Enumeră cererile de închiriere trimise de clienți.
- Pentru fiecare cerere, pot fi afișate detaliile vehiculului și starea solicitării (aprobată, respinsă sau în așteptare).

Structura bazei de date

Aplicația include următoarele structuri de bază de baze de date:

1. Tabel clienți:

- Nume
- E-mail
- Telefon

2. Diagrama vehiculului:

- Marca
- model
- Caracteristici
- Taxa zilnică de închiriere

3. Tabel cu cereri de închiriere:

- ID client (link către Tabelul clienți)
- ID vehicul (link către diagrama vehiculului)
- Data închirierii

- Perioada de inchiriere

Formular1

Această clasă este derivată din clasa `Form` și reprezintă forma principală a aplicației. Proprietățile și evenimentele sunt definite pentru a gestiona elementele de interfață găsite în formular.

Variabile statice

- `_obj` : O variabilă statică folosită pentru a stoca o instanță a clasei `Form1`. Este folosit pentru a implementa modelul de design Singleton. Permite crearea unei singure instanțe de `Form1`.

Caracteristici

- `Instanță` : O proprietate care returnează o singură instanță a clasei `Form1`. Dacă instanța `_obj` este nulă, o nouă instanță `Form1` este creată și atribuită variabilei `_obj`. Apoi `_obj` este returnat.
- `PnlContainer` : O proprietate de tip `Panel`. Acesta reprezintă panoul care găzduiește conținutul formularului.
- `ButtonAcasa` : O proprietate de tip `Button`. Acesta reprezintă butonul „Acasa”.

Variabile

- `buttonApasat` : Un steag folosit pentru a determina dacă formularul este tras din bara de titlu.
- `movX` și `movY` : Variabile utilizate pentru a urmări poziția mouse-ului în timpul operațiunii de glisare.
- `DECALAGE` : O valoare constantă reprezentând cantitatea de abatere de la poziția de pornire la tragerea formularului.

Evenimente

- `pictureBox2_Click` : eveniment declanșat atunci când se face clic pe butonul de ieșire. Formularul este închis prin apelarea metodei `Close`.
- `panel2_MouseDown` : eveniment declanșat când se face clic pe bara de titlu. Pornește procesul de glisare și salvează poziția mouse-ului.
- `panel2_MouseMove` : eveniment declanșat când mouse-ul este tras. Actualizează poziția formularului dacă a început operația de tragere.

- `panel2_MouseUp` : eveniment declanșat când se eliberează clic pe mouse. Încheie operația de tragere.

Metode

- `Form1` : Metoda constructorului clasei. Metoda `InitializeComponent` din ea încarcă designul formularului și stabilește alte setări inițiale.
- `Form1_Load` : Eveniment declanșat când formularul este încărcat. Codul din acesta efectuează acțiunile care trebuie făcute în timpul încărcării formularului. Pornește `timer1` , actualizează etichetele de oră și dată și încarcă pagina „Acasa” în mod prestabilit.
- `btnAcasa_Click` , `btnCienti_Click` , `btnAngajati_Click` , `btnMasini_Click` , `btnInchirieri_Click` : Evenimente declanșate atunci când se face clic pe butoanele de comutare a paginii. Aduce pagina relevantă în panoul `PnlContainer` și comută activitatea butonului corespunzător.
- `timer1_Tick` : eveniment care rulează de fiecare dată când `timer1` este declanșat. Actualizează eticheta ceasului.

PaginaAngajati

Variabile

- `LUNGIME_CNP` : O valoare întreagă fixă. Specifică faptul că CNP (Personal Identification Number) trebuie să aibă o lungime validă.
- `INDEX_CMB` : O valoare întreagă fixă. Reprezintă că un ComboBox nu este un element selectat.
- `adminAngajati` : Un exemplu de obiect care implementează interfața `IStocareAngajati` . Acest obiect este folosit pentru a adăuga, prelua, actualiza și șterge interacțiuni din sursa de date.
- `angajati` : O listă din clasa `Angajat` . Această listă este folosită pentru a organiza angajamente.

Evenimente

- `btnAdaugare_Click` : Evenimentul declanșat când se face clic pe butonul „Adaugare”. Verifică dacă informațiile valide sunt introduse în câmpurile de intrare și creează un nou obiect `Angajat` și îl adaugă la lista de angajamente.

- **btnAfisare_Click** : Eveniment declanșat când se face clic pe butonul „Afisare”. Obține lista de implicare din sursa de date și o afișează în controlul DataGridView.
- **btnCautare_Click** : Evenimentul declanșat când se face clic pe butonul „Cautare”. Caută implicarea din sursa de date în funcție de criteriile de căutare și afișează rezultatul în controlul DataGridView.
- **btnModificare_Click** : Eveniment declanșat când se face clic pe butonul „Modificare”. Găsește implicarea în sursa de date pe baza criteriilor de căutare și actualizează anumite câmpuri.
- **btnStergere_Click** : Evenimentul declanșat când se face clic pe butonul „Stergere”. Șterge o implicare selectată din sursa de date.

Metode

- **ResetareMessage** : O metodă care resetează etichetele mesajelor și câmpurile de intrare.
- **ResetareMessage2** : O metodă care resetează etichetele de mesaje din al doilea grup și câmpurile de intrare.
- **DateValid** : O metodă pentru a verifica dacă câmpurile de intrare conțin date valide.
- **DateValid1** : O metodă pentru a verifica dacă câmpurile de căutare conțin date valide.

Acest cod oferă o interfață pentru a efectua operațiuni asupra angajamentelor și salvează și preia datele de implicare într-o bază de date. De asemenea, afișează mesaje de feedback pentru utilizator și afișează angajamente în controlul DataGridView.

PaginaClient

Variabile

- **LUNGIME_CNP** : O valoare întreagă fixă. Specifică faptul că CNP (Personal Identification Number) trebuie să aibă o lungime validă.
- **CNP_** : O valoare întreagă fixă. Reprezintă indexul coloanei pentru afișarea CNP în controlul DataGridView.
- **adminClienti** : Un exemplu de obiect care implementează interfața **IStocareClienti**. Acest obiect este utilizat pentru a adăuga, prelua, actualiza și șterge clienți din

sursa de date.

- **client** : O listă cu clasa **Client** . Această listă este folosită pentru a păstra clienții.

Evenimente

- **btnAdaugare_Click** : Evenimentul declanșat când se face clic pe butonul „Adaugare”. Verifică dacă informațiile valide sunt introduse în câmpurile de intrare și creează un nou obiect **Client** și îl adaugă la lista de clienți.
- **btnAfisare_Click** : Eveniment declanșat când se face clic pe butonul „Afisare”. Obține lista de clienți din sursa de date și o afișează în controlul DataGridView.
- **btnCautare_Click** : Evenimentul declanșat când se face clic pe butonul „Cautare”. Caută clientul din sursa de date conform criteriilor de căutare și afișează rezultatul în controlul DataGridView.
- **btnModificare_Click** : Eveniment declanșat când se face clic pe butonul „Modificare”. Găsește clientul în sursa de date pe baza criteriilor de căutare și actualizează anumite câmpuri.
- **btnStergere_Click** : Evenimentul declanșat când se face clic pe butonul „Stergere”. Șterge un client selectat din sursa de date.

Metode

- **ResetareMessage** : O metodă care resetează etichetele mesajelor și câmpurile de intrare.
- **ResetareMessage2** : O metodă care resetează etichetele de mesaje din al doilea grup și câmpurile de intrare.
- **DateValide** : O metodă pentru a verifica dacă câmpurile de intrare conțin date valide.
- **DateValide1** : O metodă pentru a verifica dacă câmpurile de căutare conțin date valide.

Acest cod oferă o interfață utilizată pentru a efectua tranzacții legate de clienți și salvează și preia datele clienților într-o bază de date. De asemenea, utilizează controlul DataGridView pentru a afișa și edita datele clienților pentru utilizator.

Client

Variabile:

- `LUNGIME_CNP` și `CNP_`: numere întregi fixe. Reprezintă lungimea CNP (Personal Identification Number) și indexul coloanei DataGridView.
- `adminClienti`: Un exemplu de obiect care implementează interfața `IStocareClienti`. Acest obiect este utilizat pentru stocarea și accesarea datelor clienților.
- `client`: O listă cu clasa `client`. Lista în care sunt stocați și afișați clienții.

Evenimente:

- `btnAdaugare_Click`: Apare când se face clic pe butonul „Adaugare”. Verifică câmpurile de intrare pentru date valide și creează un nou obiect `client` și îl adaugă la lista de clienți. De asemenea, adaugă clientul la sursa de date.
- `btnAfisare_Click`: Apare când se face clic pe butonul „Afisare”. Resetează mesajele, activează butonul „Stergere” și afișează lista de clienți în controlul DataGridView.
- `btnCautare_Click`: Apare când se face clic pe butonul „Cautare”. Caută clientul din sursa de date conform criteriilor de căutare și afișează rezultatul în controlul DataGridView. Dacă clientul nu este găsit, acesta afișează un mesaj corespunzător.
- `btnModificare_Click`: Apare când se face clic pe butonul „Modificare”. Găsește clientul în sursa de date pe baza criteriilor de căutare și actualizează anumite câmpuri. Aceasta va reseta mesajele după această operațiune.
- `btnStergere_Click`: Apare când se face clic pe butonul „Stergere”. Șterge clientul selectat din sursa de date și actualizează controlul DataGridView.

Metode:

- `ResetareMessage`: O metodă care resetează etichetele mesajelor și câmpurile de intrare.
- `ResetareMessage2`: O metodă care resetează etichetele de mesaje din al doilea grup și câmpurile de intrare.
- `DateValid`: O metodă de a verifica dacă câmpurile de intrare conțin date valide.
- `DateValid1`: O metodă de a verifica dacă câmpurile de căutare conțin date valide.

PaginaInchirieri

Variabile:

- `LUNGIME_CNP` și `LUNGIME_SERIE` : numere întregi fixe. Reprezintă lungimea CNP (Personal Identification Number) și seria vehiculului.
- `adminiInchirieri` , `adminiClienti` și `adminMasini` : implementarea interfețelor `IStocareInchirieri` , `IStocareClienti` și `IStocareMasini` instanțe de obiect. Aceste obiecte sunt folosite pentru a stoca și accesa tranzacțiile de închiriere, datele clienților și datele vehiculului.
- `inchirieri` , `clienti` și `masini` : format din `Inchiriere` , `Client` și `Masina` liste de clase. Închiririle, clienții și vehiculele sunt stocate în aceste liste.

Evenimente:

- `btnAdaugare_Click` : Apare când se face clic pe butonul „Adaugare”. Verifică câmpurile de intrare pentru date valide și creează un nou obiect `Inchiriere` și îl adaugă la lista de închiriere. De asemenea, adaugă contractul de închiriere la sursa de date.
- `btnAfisare_Click` : Apare când se face clic pe butonul „Afisare”. Resetează controalele, activează butonul „Stergere” și afișează lista de închiriere în controlul DataGridView.
- `btnStergere_Click` : Apare când se face clic pe butonul „Stergere”. Șterge operațiunea de închiriere selectată din sursa de date și actualizează controlul DataGridView.

Metode:

- `ExistaClient` : O metodă care verifică dacă CNP-ul specificat există în sursa de date.
- `ExistaMasina` : O metodă care verifică dacă seria de vehicule specificată există în sursa de date.
- `DateValide` : O metodă de a verifica dacă câmpurile de intrare conțin date valide.
- `ResetareControale` : O metodă care resetează controalele.

Despre PaginiMasina

Codul are două spații de nume separate numite „LibrarieModele” și „NivelAccesDate”. Spațiul de nume „LibrarieModele” conține modele de date și operațiuni legate de

instrumente, în timp ce spațiul de nume „NivelAccesDate” oferă operațiuni de bază de date și metode de acces la date.

Formularul are multe controale. Modelele de vehicule pot fi selectate cu ajutorul butoanelor radio, seria de vehicule și prețul pot fi introduse cu casete text, caracteristicile vehiculului pot fi selectate cu casete de selectare, iar vehiculele disponibile sunt afișate cu o casetă de listă. În plus, există butoane pentru adăugare, editare, căutare și ștergere.

În cod, lista de instrumente este păstrată într-o listă numită „masini” și această listă este salvată sau preluată din baza de date prin intermediul obiectului „adminMasini”. După ce sunt întreprinse acțiunile, etichetele mesajelor sunt actualizate și caseta cu listă este repopulată.

Codul poate fi folosit pentru a efectua operațiuni de bază, cum ar fi adăugarea, editarea, căutarea și ștergerea vehiculelor. Cu toate acestea, se observă că există unele componente care lipsesc pentru ca codul să funcționeze. De exemplu, conținutul claselor precum „StocareFactory” și „IStocareMasini” nu este specificat. În plus, alte componente ale designului interfeței nu sunt incluse în cod.

Prin urmare, este necesar să completați componentele lipsă și să furnizați în mod corespunzător alte dependențe. De asemenea, codul trebuie testat cu atenție pentru a-i asigura funcționalitatea și acuratețea.

Biblografie

<https://github.com/feliciacricri/AplicatieDemoPIU/tree/main>

<https://learn.microsoft.com/en-us/dotnet/csharp/>