

Platformă de check-in pentru bagaje

Documentație proiect PPAW

Profesor coordonator:

s.l. dr. ing. Gîză-Belciug Felicia

Student: Rebeca-Mihaela Polocoșer

Specializarea SIC anul 1

Cuprins

I. PROIECTARE APLICAȚIE	3
1. Paradigme utilizate – MVC și ORM Code First	3
2. De ce au fost alese?	3
3. Arhitectura aplicație	4
a) Modelele utilizate	4
b) Funcționalitatea per ansamblu a aplicației	4
II. IMPLEMENTARE.....	5
1. Business layer	5
2. Librării suplimentare utilizate	5
3. Secțiuni de cod sau abordări deosebite	5
III. MOD DE UTILIZARE	7
1. Pașii de instalare	7
a) Instalare si configurare pentru programator	7
b) Instalare și configurare a beneficiarului	7
2. Mod de utilizare	8
IV. CONCLUZII	11

I. PROIECTARE APLICAȚIE

1. Paradigme utilizate – MVC și ORM Code First

MVC vine de la Model – View – Controller și este un tipar de design arhitectural utilizat pe scară largă în dezvoltarea aplicațiilor software, în special a aplicațiilor web. Acesta organizează codul în trei componente interconectate, fiecare având responsabilități distincte, pentru a separa logica aplicației de interfața cu utilizatorul. Această separare îmbunătățește structura codului, facilitează testarea și permite scalabilitatea.

ORM (Object – Relational Mapping) este o tehnică folosită pentru conversirea datelor dintre un sistem relațional (baze de date SQL) și obiecte dintr-un limbaj de programare orientat pe obiecte. Câteva exemple considerabile sunt Entity Framework (.NET), Hibernate (Java), SQLAlchemy (Python), prima dintre ele fiind folosită în proiectul meu.

Cu alte cuvinte, ORM Code First este o abordare în dezvoltarea aplicațiilor utilizând un ORM care implică scrierea întâi a codului pentru clasele de model (obiecte din cod), iar baza de date este generată ulterior pe baza acestora. Este o metodă preferată atunci când se dorește un control mai mare asupra structurii codului, iar design-ul bazei de date este derivat direct din modelele aplicației.

Împreună, ORM Code First și MVC sunt folosite complementar, mai ales în dezvoltarea aplicațiilor moderne, aplicații web bine structurate și scalabile. Împreună sunt structurate și folosite astfel: *Modelul* conține datele și logica aplicației, *View* este interfața utilizatorului, *Controller* gestionează interacțiunile cu utilizatorul, lucrează cu modelul și returnează un view, iar *ORM Code First* manageriază partea de Model, generând baza de date pe baza claselor definite în cod.

Pentru baza de date am folosit MySQL Workbench unde datele au fost gestionate mai ușor după ce au fost create prin ORM Code First.

2. De ce au fost alese?

Motivația cea mai importantă pentru care am ales această combinație este faptul că există o separare clară a responsabilităților și eficiență în dezvoltare și flexibilitate. Baza de date este automatizată și modificările sunt gestionate prin migrări, ceea ce accelerează dezvoltarea.

Am ales MVC pentru avantajele pe care acesta le are, și anume, separarea responsabilităților, făcând codul mai ușor de întreținut, permite dezvoltarea și testarea componentelor independent și simplifică colaborarea între dezvoltatori (frontend și backend). De asemenea, acesta folosește diverse framework-uri populare cu care am putut rezona mai bine.

Avantajele principale pe care Code First le are sunt flexibilitatea, codul fiind un punct central, controlul asupra modelelor, evoluție rapidă și menținere simplificată, modelele fiind programate în baza de date prin migrări. Acestea au fost esențiale pentru mine în alegerea acestei tehnici și acestor tehnologii.

3. Arhitectura aplicație

a) Modelele utilizate

Entitățile principale folosite în aplicația de față sunt următoarele:

- Zbor: isZbor, NumeCompanie, Îmbarcare, Destinație, DataPlecare, GreutateBagaj, LocuriDisponibile, Status, Preț, TaxăSuplimentară
- Utilizator: idUtilizator, Nume, Email, Parola, Pasager
- Checkin: idCheckin, idUtilizator, idZbor, GreutateBagaj, DataCheckin, LocRezervat, PrețFinal, isValid

Pe lângă acestea au fost create și câteva ViewModel pentru fiecare: AddCheckinViewModel.cs, AddUtilizatorVieewModel.cs, AddZborViewModel.cs

b) Funcționalitatea per ansamblu a aplicației

Aplicația pare a fi destinată gestionării zborurilor, cu următoarele funcționalități principale:

1. Gestionarea zborurilor

- Afișarea unei liste de zboruri disponibile:
 - Utilizatorii pot vizualiza detalii precum compania aeriană, locația de îmbarcare, destinația, data plecării, prețul și statusul fiecărui zbor.
- Navigarea prin funcționalitatea aplicației:
 - Linkuri și butoane pentru a merge în alte secțiuni (ex. "Înapoi la listă").

2. Check-in

- Utilizatorii pot realiza check-in pentru zboruri.
- Starea zborurilor (Status) influențează disponibilitatea acestei opțiuni. De exemplu:
 - Zborurile "Planificate" permit check-in, celelalte nu.

3. Interfața utilizatorului

- Interfața este construită folosind Bootstrap, ceea ce asigură:
 - Design responsive (adaptabil pentru diverse dispozitive),
 - Stiluri predefinite pentru formulare, butoane, tabele, etc.

4. Navigația între secțiuni

- Secțiuni principale:
 - **Home**: Pagina principală.
 - **Utilizatori**: Posibilă gestiune a utilizatorilor aplicației.
 - **Zboruri**: Gestionarea zborurilor.
 - **Check-in**: Gestionarea procesului de check-in pentru pasageri.
- Layout-ul (_Layout.cshtml) definește o structură comună pentru întreaga aplicație.

II. IMPLEMENTARE

1. Business layer

În proiectul meu, business logic layer-ul (BLL) reprezintă stratul care conține logica de afaceri, adică regulile și operațiunile care transformă datele și le oferă aplicației o funcționalitate concretă. În principal au fost folosite serviciile și anumite reguli specifice.

Serviciile care au fost implementate în proiect sunt CheckinService, MemoryCacheService, UtilizatorService și ZboerService. Pe lângă acestea am implementat și câteva View-uri special

a) **Procesul de check-in și validarea greutății bagajului:**

- La efectuarea check-in-ului, greutatea bagajului pasagerului este comparată cu greutatea maximă permisă (GreutateMaximaBagaj) pentru acel zbor.
- Dacă greutatea este depășită, se aplică o **taxă suplimentară** pentru fiecare kilogram în exces. Aceasta este calculată prin formula:

$$\text{Taxa Suplimentara} = (\text{Greutate Depășită}) \times (\text{Tarif per kilogram})$$

- Se comunică pasagerului taxa suplimentară prin interfața utilizatorului.

b) **Verificarea statusului zborului:**

- Check-in-ul este permis doar pentru zborurile cu statusul „Planificat”.
- Dacă zborul este „Anulat” sau deja finalizat, check-in-ul nu poate fi efectuat.

2. Librării suplimentare utilizate

Librăriile suplimentare pe care le-am folosit sunt pentru aplicația .NET pentru a gestiona dependențele, pentru lucrul cu baza de date și optimizarea performanțelor prin mecanisme de caching sunt:

- Autofac: ajută la configurarea și gestionarea serviciilor prin injecție de dependențe. Urmărește ciclul de viață al obiectelor pentru rezolvarea dependențelor. Ușor de configurat și flexibil.
- Autofac.Extensions.DependencyInjection: Permite înlocuirea containerului DI implicit din ASP.NET Core cu Autofac.
- Microsoft.EntityFrameworkCore.SqlServer: Conectează și manipulează datele între aplicațiile .NET și SQL Server.
- Microsoft.EntityFrameworkCore.Tools: Executarea comenzilor pentru gestionarea migrărilor și a bazei de date.
- Pomelo.EntityFrameworkCore.MySql: Conectează aplicațiile .NET la baze de date MySQL/MariaDB și utilizează funcționalități EF Core pentru manipularea datelor.
- System.Runtime.Caching: Stochează temporar datele (cache) pentru îmbunătățirea performanțelor aplicațiilor.

3. Secțiuni de cod sau abordări deosebite

O practică bună pe care am abordat-o în aplicația mea este faptul că folosesc condiții pentru a adapta interfața utilizatorului în funcție de starea datelor, îmbunătățind experiența utilizatorului. De exemplu butonul Detalii sau Check-in este afișat doar pentru statusul zborului care este setat ca “Planificat”. Tabelele de date sunt personalizate, afișându-se doar informațiile

relevante. Astfel, interfața este simplificată și utilizatorii se pot concentra pe datele esențiale, fără a-i copleși.

Zboruri disponibile

Nume Companie	Locul de Îmbarcare	Destinație	Data Plecare	Pret	Status	Acțiuni
Wizz Air	Suceava	Roma	12/12/2024 14:30:00	25.00€	Planificat	Detalii
Ryanair	Roma	Paris	10/10/2024 14:00:00	100.00€	Anulat	Detalii indisponibile
Wizz Air	Suceava	Paris	21/12/2024 10:00:00	40.00€	Planificat	Detalii
Blue Air	Roma	Suceava	10/01/2025 21:00:00	60.00€	Planificat	Detalii
Vueling	Londra	Luxemburg	12/01/2025 22:00:00	120.00€	Planificat	Detalii
Wizz Air	Suceava	Londra	01/12/2024 14:00:00	40.00€	Finalizat	Detalii indisponibile

Codul folosește principii care permit reutilizarea, având metode bine structurate cum ar fi Zbor și funcționalitățile sunt organizate în metode distincte pentru a separa logica aplicației.

Am introdus câteva condiții de validare a datelor și server-site pentru gestionarea datelor introduse. Aceasta este o abordare robustă, asigurându-te să aplicația este mai puțin predispusă la erori cauzate de introducerea unor date incorecte sau nevalide.

Există o flexibilitate pentru editare și detalii bine structurate astfel încât să includă câmpuri configurabile, ceea ce permite extinderea facilă. De asemenea, am inclus afișare condiționată a unor elemente (disabled pentru câmpuri doar de citire) și formulare care păstrează funcționalitatea și design-ul consistent.

Detalii Zbor

Informații Zbor

ID Zbor

4

Locul de Îmbarcare

Roma

Data Plecare

10/01/2025 21:00

Taxa Suplimentară Bagaj

30.00

Status

Planificat

Înapoi la listă

Nume Companie

Blue Air

Destinație

Suceava

Greutate Maximă Bagaj (kg)

40.00

Locuri Disponibile

7

Mergi la Check-in

Mesajele de confirmare pentru ștergerea unui zbor (onsubmit="return confirm('Ești sigur că vrei să ștergi acest zbor?');") și recomandările pentru utilizarea mediului Development sunt indicii că am avut în vedere securitatea și experiența utilizatorului.

localhost:5113 says

Acasă Utilizatori Zboruri

Esti sigur ca vrei sa stergi acest utilizator?

OK Cancel

Id	Nume	Email	Pasager	Actiuni
1	Admin	admin@gmail.com	✓ Da	<div>Edit</div> <div>Șterge</div>
3	Maria Curelusa	maria@gmail.com	✓ Da	<div>Edit</div> <div>Șterge</div>

Concluzionând, am reușit să implementez câteva detalii importante pe lângă conținutul deosebit de important. Aceste lucruri mai mici dau o altă valoare proiectului, ajutând totodată și utilizatorul.

III. MOD DE UTILIZARE

1. Pașii de instalare

Câteva recomandări pe care aș putea să le am sunt ca să se folosească HTTPS în producție ca este mai sigur. Să se seteze un firewall pentru protejarea serverului și să nu se utilizeze conturi de administrație pentru conexiuni la baza de date. Pentru backup, ar fi bine ca să se configureze automat pe baza de date și aplicație. Opțional, să se creeze o documentație sau un ghid de utilizare pentru utilizator, însoțit de capturi de ecran.

a) Instalare și configurare pentru programator

Există câteva cerințe preliminare precum sistemul de operare poate fi Windows, macOS sau Linux. Pe lângă aceasta, câteva aplicații necesare sunt:

- Visual Studio pentru ASP.NET și web development
- .NET SDK, o versiune compatibilă
- MySQL Workbench ca server pentru baza de date
- GitHub, dacă proiectul este versionat

Pași de instalare:

- În MySQL Workbench creează un server pentru aplicația pe care o vei crea
- Deschide proiectul Visual Studio și alege fișierul .sln al proiectului
- Configurează baza de date modificând fișierul *appsettings.json* și incluzând detaliile bazei de date create în MySQL Workbench
- aplică migrațiile pentru baza de date și testează lansarea aplicației apăsând **CTRL + F5** sau chiar de butonul specific din Visual Studio

b) Instalare și configurare a beneficiarului

Configurarea pentru mediul de producție are, de asemenea, câteva cerințe de preeliminare precum

- Server-ul web pentru Windows trebuie să fie configurat cu IIS (Internet Information Services), alternativ, pentru Linux se poate folosi Nginx sau Apache.
- .NET Runtime trebuie instalat corespunzător, în funcție de versiune

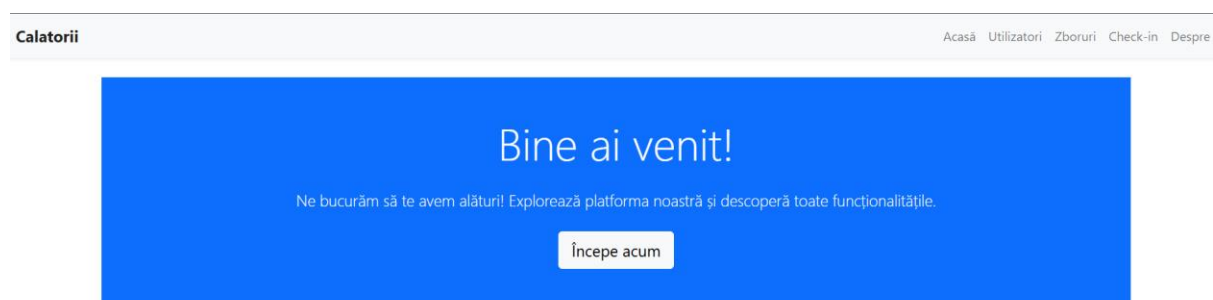
→ MySQL Workbench pentru baza de date, compatibil

Pași de instalare:

- Publicarea aplicației și transferul fișierelor pe server, după care configurarea IIS (pentru Windows) și a bazei de date, precum și a mediului de producție
- start și testare

2. Mod de utilizare

- Pentru început, când rulăm aplicația, ni se deschide pagina de Home (Acasă). De acolo selectăm **“Începe acum”** unde experiența utilizatorului începe.



- Pe pagina de Zboruri poți selecta dintr-o listă de zboruri disponibile să vezi mai multe detalii, cu scopul de a achiziționa transport pe acea rută. Poți totodată să ștergi rutele de zbor care nu te interesează.

Zboruri disponibile						
Nume Companie	Locul de îmbarcare	Destinație	Data Plecare	Pret	Status	Acțiuni
Wizz Air	Suceava	Roma	12/12/2024 14:30:00	25.00€	Planificat	Detalii Șterge
Ryanair	Roma	Paris	10/10/2024 14:00:00	100.00€	Anulat	Detalii indisponibile Șterge
Wizz Air	Suceava	Paris	21/12/2024 10:00:00	40.00€	Planificat	Detalii Șterge
Blue Air	Roma	Suceava	10/01/2025 21:00:00	60.00€	Planificat	Detalii Șterge
Vueling	Londra	Luxemburg	12/01/2025 22:00:00	120.00€	Planificat	Detalii Șterge
Wizz Air	Suceava	Londra	01/12/2024 14:00:00	40.00€	Finalizat	Detalii indisponibile Șterge

- Pe pagina detalii Zbor pot fi observate toate datele pe care le are acea cursă aeriană. Dacă utilizatorul dorește în continuare să achiziționeze acel traseu, atunci selectează **“Mergi la check-in”**, altfel, selectând butonul **“Înapoi la listă”** în va întoarce la pagina **“Zboruri disponibile”** unde va putea selecta un alt zbor care îl dorește.

Detalii Zbor

Informații Zbor

ID Zbor	Nume Companie
5	Vueling
Locul de Îmbarcare	Destinație
Londra	Luxemburg
Data Plecare	Greutate Maximă Bagaj (kg)
12/01/2025 22:00	80.00
Taxa Suplimentară Bagaj	Locuri Disponibile
15.00	10
Status	
Planificat	

[Înapoi la listă](#) [Mergi la Check-in](#)

- Pe pagina Adaugare Check-in se va selecta utilizatorul pentru zbor. Dacă acesta este deja în baza de date, va fi găsit în listă. Altfel va trebui să fie adăugat ca utilizator nou din tab-ul Utilizatori. Ca și date de completat, utilizatorul trebuie să pună greutatea Bagajului și locul din avion pe care îl ocupă. Data check-in se setează automat în funcție de ziua și ora curentă, iar preț Călătorie este setat în funcție chiar de zborul selectat și poate fi doar vizualizat.
- Ca și funcționalități, de aici se poate merge înapoi la lista de zboruri sau se adauga check-in.

Adăugare Check-in

Utilizator
Selectați un utilizator
Zbor
10 locuri disponibile - greutate bagaj: 80.00 kg
Greutatea Bagajului
0.00
Loc Rezervat
Introduceți locul rezervat
Data Check-in
11/01/2025 19:37:47.089
Preț Calatorie
120.00 EUR

[Înapoi la Lista de Zboruri](#) [Adauga Check-in](#)

- Dacă greutatea bagajului nu depășește greutatea care este specifică zborului, utilizatorul va putea vedea lista de check-in-uri.

Check-in-uri

Id Checkin	Utilizator ID	Zbor ID	Greutate Bagaj	Data Checkin	Loc Rezervat	Acțiuni
1	1	1	20.00	12/12/2024 14:30:00	12A	Edit Șterge
2	3	1	125.00	10/01/2025 11:53:35	C12	Edit Șterge
3	1	1	120.00	10/01/2025 11:54:52	15B	Edit Șterge
4	8	4	50.00	10/01/2025 15:05:29	20A	Edit Șterge
5	5	1	10.00	10/01/2025 15:35:46	B12	Edit Șterge
6	7	3	15.00	10/01/2025 15:38:51	B12	Edit Șterge
7	8	3	25.00	10/01/2025 16:06:52	17B	Edit Șterge
8	5	3	45.00	10/01/2025 23:36:07	12B	Edit Șterge
9	7	3	30.00	10/01/2025 23:39:15	12B	Edit Șterge
10	5	3	40.00	11/01/2025 19:49:17	12B	Edit Șterge

→ Dacă greutatea bagajului depășește, va apărea o pagină nouă unde ni se specifică informații despre taxa bagajului. Dacă dorește, acesta poate continua și finaliza cu succes check-in-ul sau poate merge la lista de zboruri.

Confirmare Taxă Suplimentară

O taxă suplimentară de 100.0000 EUR a fost calculată pentru bagajul în exces (exces: 20.00 kg).

Confirmă și continuă

Vezi zboruri

→ Celelalte tab-uri pe care le are proiectul sunt următoarele:

→ Pagina Utilizatori pot fi adaugati noi utilizatori care sunt după incluși în baza de date.

Utilizatori

Id	Nume	Email	Pasager	Acțiuni
1	Admin	admin@gmail.com	✓ Da	Edit Șterge
3	Maria Curelusa	maria@gmail.com	✓ Da	Edit Șterge
4	Gheorghe Vasile	gheorghe@gmail.com	✗ Nu	Edit Șterge
5	Melisa Polocoser	meli@gmail.com	✓ Da	Edit Șterge
7	Iemima Polocoser	mimi@gmail.com	✗ Nu	Edit Șterge
8	Rebeca Polocoser	Rebeca@gmail.com	✓ Da	Edit Șterge
9	Damaris Polocoser	damaris@gmail.com	✓ Da	Edit Șterge

Adaugă Utilizator

→ Pagina despre sunt detalii despre creatorul acestui proiect

Autor

Informații autor:

Nume: Polocoser Rebeca Mihaela

Specializare: SIC, an 1

Grupa: 3711B

IV. CONCLUZII

Acest proiect reprezintă o soluție modernă pentru gestionarea eficientă a zborurilor și rezervărilor într-o companie aeriană. Prin utilizarea arhitecturii MVC, aplicația separă clar logica de business, interfața de utilizator și gestionarea datelor, oferind un sistem scalabil și ușor de întreținut.

Funcționalitatea sa permite automatizarea proceselor esențiale, precum verificarea locurilor disponibile, calculul taxelor suplimentare pentru bagaje și validarea datelor introduse. Astfel, aplicația reduce erorile și optimizează timpul necesar pentru administrarea zborurilor.

Prin integrarea tehnologiilor moderne, aplicația oferă o interfață prietenoasă și accesibilă pe diverse dispozitive. Este un exemplu concret al modului în care dezvoltarea software poate rezolva probleme reale într-un mod eficient și sigur.

În viitor, aplicația poate fi extinsă cu funcționalități suplimentare, cum ar fi notificările automate sau raportarea avansată, pentru a satisface nevoile în continuă schimbare ale utilizatorilor.