APLICAŢIE BACK END C# CU BAZA DE DATE PENTRU GESTIONAREA IMPORTURILOR, COSTURILOR ŞI STOCK-ULUI CU REST API, MS SQL SERVER, ASP.NET CORE WEB API (C#)

Cristi PLĂMĂDEALĂ

CZU: 004.42 chris.plamadeala@gmail.com

This thesis develops a C# back end application for managing imports, costs, and stock. The application automates data import from CSV or JSON files, updates inventory, calculates costs, and ensures efficient data management using ASP.NET Core and SQL Server. Key features include automated data import, stock updates, cost calculation, and a user-friendly interface. The application improves efficiency and accuracy in managing online store inventories.

În era digitală actuală, dezvoltarea aplicațiilor web și mobile este în plină expansiune, cu aplicațiile back-end jucând un rol crucial în procesarea datelor și interacțiunea cu bazele de date. Această teză explorează dezvoltarea unei aplicații back-end folosind limbajul de programare C# și o bază de date relațională. Aplicația are scopul de a gestiona importurile, costurile și stocul pentru un magazin online, permițând importul de fișiere CSV sau JSON, adăugând automat produsele în stoc și calculând costurile aferente.

Obiectivele acestei teze sunt:

1. Înțelegerea rolului și importanței aplicațiilor back end în dezvoltarea aplicațiilor web și mobile.
2. Explorarea limbajului de programare C# și framework-ului ASP.NET Core MVC.
3. Învățarea și utilizarea Entity Framework Core pentru interacțiunea cu o bază de date relațională.
4. Proiectarea și implementarea logicii aplicației folosind C# și ASP.NET Core MVC.
5. Gestionarea relațiilor între entitățile aplicației prin intermediul claselor asociative și configurarea restricțiilor de integritate referențială.
6. Dezvoltarea unei interfețe grafice utilizând ASP.NET Core MVC pentru a afișa și manipula datele din aplicație.

Dezvoltarea Aplicației și Tehnologii Utilizate. Caracteristici și Funcționalități Cheie

Aplicația dezvoltată servește la automatizarea și eficientizarea gestionării inventarului și a costurilor asociate pentru un magazin online sau orice altă afacere care necesită monitorizarea atentă a stocurilor și a prețurilor de achiziție. Prin importul de

date din fișiere CSV sau JSON, aplicația elimină necesitatea introducerii manuale a datelor, reducând astfel timpul necesar și posibilitatea de erori umane.

Obiectivele principale ale aplicației includ:

1. **Import automatizat de date:** Aplicația poate prelua automat informațiile despre produse din fișiere CSV sau JSON, asigurându-se că toate datele necesare sunt importate corect în sistem.
2. **Actualizarea stocului**: După import, aplicația va actualiza cantitățile de produse în sistemul de inventar, permițând utilizatorilor să vadă rapid nivelurile actuale de stoc.
3. **Calculul costurilor:** Aplicația va calcula costurile asociate cu fiecare produs, posibil incluzând prețul de achiziție, taxele, costurile de transport și orice alte cheltuieli relevante, astfel încât utilizatorii să poată determina prețul de vânzare sau să mențină un control al marjei de profit.
4. **Gestionarea eficientă a datelor:** Prin utilizarea unei baze de date (cum ar fi SQL Server) și a unui API REST, aplicația va oferi o metodă organizată și securizată de stocare și recuperare a datelor despre produse și stocuri.
5. **Interfață utilizator prietenoasă:** O interfață de utilizator intuitivă care facilitează importul de fișiere, vizualizarea datelor importate și generarea de rapoarte de stoc și costuri [1].

Tehnologii Utilizate în Dezvoltarea Aplicației Back-End ASP.NET Core și C#

ASP.NET Core este un framework open-source care rulează pe multiple sisteme de operare, inclusiv Windows, Linux, macOS și Docker. Pune accent pe performanță, caracteristici moderne și suport extins. Framework-ul permite gestionarea eficientă, stocarea și recuperarea datelor, suportând procesele și aplicațiile de afaceri.

Caracteristici cheie ale ASP.NET Core:

1. Performanță excepțională, procesând milioane de cereri pe secundă.
2. Modern și inovator, suportând evoluția rapidă a componentelor de runtime, API-urilor, compilatoarelor și limbajelor.
3. Suport extins, oferind opțiuni flexibile de ciclu de viață pentru asigurarea compatibilității și actualizărilor constante [2].

Biblioteci Utilizate în Dezvoltarea Aplicației

Aplicația utilizează diverse biblioteci, inclusiv Entity Framework Core pentru Object-Relational Mapping (ORM) și Microsoft.AspNetCore.Mvc pentru construirea aplicațiilor web folosind framework-ul ASP.NET Core. Aceste biblioteci oferă funcționalități predefinite, reducând timpul și efortul necesare pentru dezvoltarea unor aspecte specifice ale aplicației de la zero [3].

Proiectarea și Implementarea Aplicației. Proiectarea și Implementarea Bazei de Date

Design-ul bazei de date include tabele precum Item, Code, Import, ItemImport și Purchases. Fiecare tabel stochează informații specifice, menținând relații între iteme, importuri și achiziții. Structura permite urmărirea complexității stocurilor și vânzărilor într-un magazin, de la import până la vânzarea finală [4].

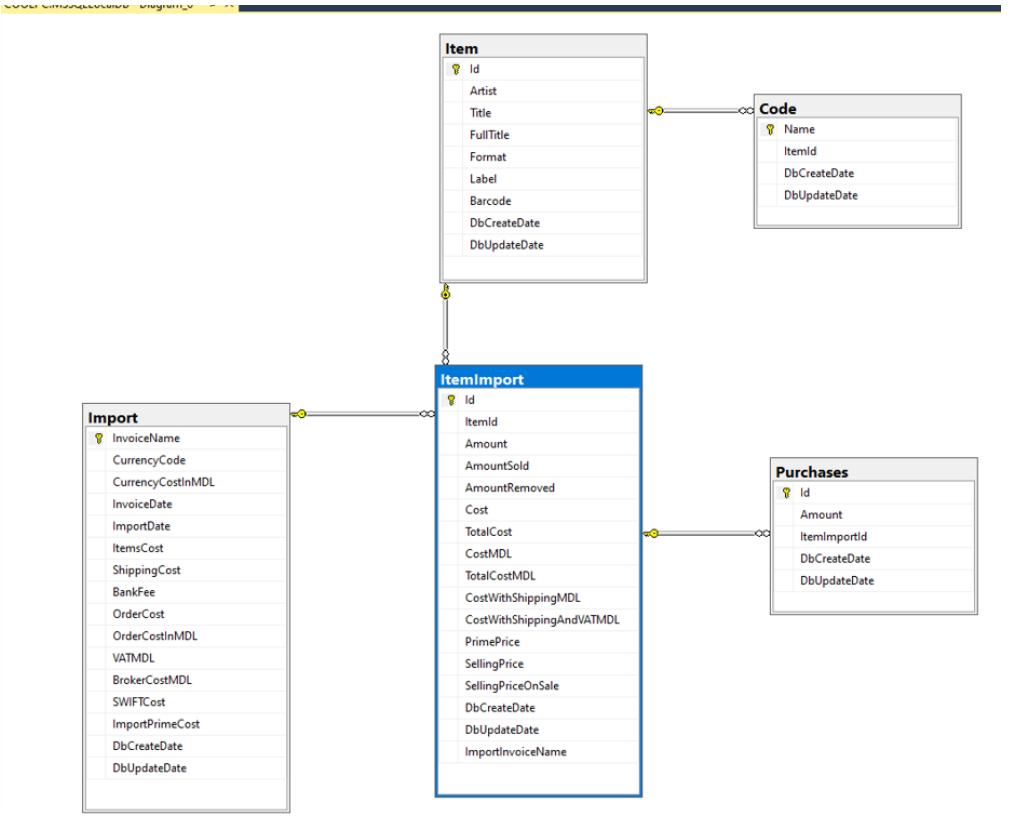


Fig. 1.1. Diagrama bazei de date create pentru aplicația elaborata

Concluzie. Dezvoltarea unei aplicații back-end utilizând C# și o bază de date relațională oferă beneficii semnificative în gestionarea datelor pentru un magazin online. Aplicația automatizează importul datelor, actualizează nivelurile de stoc, calculează costurile și asigură o gestionare eficientă a datelor printr-o interfață ușor de utilizat. Prin atingerea acestor obiective, teza oferă o perspectivă cuprinzătoare asupra dezvoltării unei aplicații back-end C# cu bază de date.

Referințe:

1. https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet-core (30.04.2024)
2. https://dotnet.microsoft.com/en-us/languages/csharp (12.03.2024)
3. https://learn.microsoft.com/en-us/ef/core/ (27.03.2024)
4. https://learn.microsoft.com/en-us/sql/relational-databases/databases/databases?view=sql-server-ver16 (14.04.2024)

Recomandat
***Eleonora SEICIUC,** dr., conf. univ.*