

# Emejzon

## Aplikacja do zarządzania magazynem

### Opis projektu

Celem projektu było stworzenie prostej aplikacji konsolowej, posiadającej wszystkie niezbędne funkcje do zarządzania magazynem, czyli zarządzanie produktami znajdującymi się w magazynie, pracownikami i zleceniami. Dodatkowo zapewniliśmy interfejs dla klientów, umożliwiający im składanie zamówień. Oznacza to, że nasza aplikacja będzie zarówno dla pracowników, menadżerów i administratorów magazynu, ale także dla klientów.

### Technologie

- Język programowania: C#, obsługa bazy danych przy pomocy SQL
- Całość tworzona w .NET 9.0
- IDE: Visual studio 2022, Heidi
- SQL connector

### Struktura plików

- Emejzon - główny folder zawierający projekt, znajduje się w nim też baza danych
- Emejzon/Documentation - Folder zawierający dokumentację projektu oraz schemat UML
- Emejzon/Emejzon - Zawiera główną część projektu, program.cs oraz pliki niezbędne do odpalenia programu w visual studio 2022
- Emejzon/Emejzon/Interfaces - Folder zawiera pliki .cs deklarujące interfejsy używane w programie
- Emejzon/Emejzon/Services - Folder zawiera pliki .cs, służące do obsługi logów, bazy danych i hasła, niezbędne do działania programu
- Emejzon/Emejzon/Users - Folder zawiera pliki .cs, deklarujące klasy użytkowników: User, Client, Worker

## Instrukcja instalacji i uruchomienia

1. Aby uruchomić program, trzeba posiadać urządzenie zdolne obsłużyć Visual Studio 2022, zainstalowany MySQL connector oraz dowolną bazę MySQL. W instrukcji będzie mowa o programie XAMPP, bo to przy jego pomocy budowaliśmy aplikację.
2. Mając zainstalowane Visual Studio 2022, w konsoli trzeba wpisać "dotnet add package MySqlConnector", to powinno automatycznie rozszerzyć Visual Studio o MySQL Connector. Program XAMPP można z kolei pobrać z tego linku: [XAMPP](#)
3. Aby sklonować repozytorium należy skopiować link z serwisu GitHub i wkleić go do Visual Studio 2022 w zakładce "Klonuj Repozytorium" widocznej tuż po odpaleniu IDE
4. W lokalizacji Emejzon\Emejzon\bin\Debug\net9.0 trzeba stworzyć plik database.txt z zawartością:  
server=<lokalizacja serwera>; (domyślnie 127.0.0.1)  
user=<nazwa użytkownika>; (domyślnie "root")  
password=<hasło>;  
(jeżeli użytkownik nie ma ustawionego hasła, można pominąć)  
database=Emejzon;  
Wszystko powinno znaleźć się w jednej linijce, jedno po drugim, użyłem enterów dla czytelności. Przykładowo:  
Server=127.0.0.1;user="root";database=Emejzon;  
Jest to działanie jednorazowe, wymagane po zainstalowaniu aplikacji.
5. W programie XAMPP control panel należy włączyć funkcje: Apache i MySQL
6. Po wykonaniu powyższych kroków można włączyć Emejzon/Emejzon/Emejzon.sln i odpalić program (domyślnie F5). Powinno pojawić się okno logowania

## Logowanie

Aby zalogować się od strony pracowników stworzyliśmy przykładowe konto admina o mailu [admin@example.com](mailto:admin@example.com) i hasło admin.

## Opis działania aplikacji

Po włączeniu aplikacji mamy do wyboru zalogowanie się, utworzenie nowego konta klienta, lub wyjście. Utworzenie nowego konta wymaga podania: imienia, nazwiska, adresu e-mail, hasła, numeru telefonu, miasta oraz adresu zamieszkania. Po zalogowaniu się na konto przy pomocy podanego wcześniej e-maila i hasła mamy do wyboru zobaczyć status swoich zamówień, możliwość wycofania zamówienia oraz złożenia nowego.

Jeśli jesteśmy pracownikami magazynu nasze konto zostało dodane ręcznie do bazy danych, więc wystarczy się zalogować.

Jako **pracownik Worker** możemy wyświetlić oczekujące zamówienia, wybrać zamówienie do zfinalizowania, lub wysłać zamówienie.

Jako **pracownik Menager** możemy dodać produkty do magazynu, usunąć produkty, uzupełnić zapasy produktów, wyświetlić wszystkie produkty, wyświetlić wszystkie przypisane oraz nieprzypisane produkty oraz przypisać zamówienie do pracownika.

Jako **pracownik Admin** możemy wykonać wszystkie funkcje dostępne dla Menagera, a do tego dodawać, modyfikować i usuwać użytkowników, a także zajrzeć w logi aplikacji.

## Przykłady użycia

1. Jako nowy klient – Otwieram program ([Instrukcja uruchomienia](#)) i tworzę konto. W tym celu podaję moje dane osobowe, a następnie loguję się na to konto. Z menu wybieram opcję, pokazującą wszystkie dostępne produkty w magazynie, wybieram ten, który mnie interesuje i zapisuję jego ID. Wciskam dowolny przycisk aby cofnąć się do menu i wybieram opcję złożenia zamówienia. Tam podaję ID produktu i ilość jaka mnie interesuje.
2. Jako pracownik Worker - Loguję się do systemu i sprawdzam oczekujące zamówienia. Finalizuję je i wysyłam.
3. Jako pracownik Manager - Loguję się do systemu i sprawdzam czy pojawiły się nowe zamówienia. Jeśli tak to przypisuję je pracownikom.

## Struktura danych i klasy

### Klasy:

Klasa User – Przypisana do każdego użytkownika systemu. Zawiera właściwości przechowujące dane osobowe.

Klasa Client – przypisana do każdego użytkownika, który jest klientem magazynu. Zawiera menu klienta.

Klasa Worker – przypisana do każdego użytkownika, który pracuje w magazynie. Zawiera menu wszystkich pozycji pracowników, to jest worker, manager, admin.

## Interfejsy:

IClient – zawiera metody dla klientów:

- PlaceOrder – Metoda obsługująca składanie zamówienia
- ShowOrderStatus – Pokazuje status wybranego zamówienia
- RemoveOrder – usuwa wybrane zamówienie

IWarehouseManagement – Zawiera metody dla workera z rolą Manager i Admin

- RefillProducts – Pozwala dodać określoną ilość sztuk do wybranego produktu w magazynie
- DeleteProduct – Pozwala usunąć produkt z magazynu po podaniu jego ID
- AddProduct – Pozwala dodać nowy produkt do magazynu
- ShowAllProducts - Wyświetla wszystkie produkty znajdujące się w magazynie (Używane również przez klasę Client)
- ShowAllOrders - Wyświetla wszystkie oczekujące zamówienia
- ShowAllAssignedOrders - Wyświetla wszystkie oczekujące zamówienia, które są przypisane do pracowników
- ShowAllUnassignedOrders - wyświetla wszystkie oczekujące zamówienia, które nie są przypisane do żadnego pracownika
- AssignOrder- Pozwala przypisać zamówienie do wybranego pracownika

ISystemManagement – Zawiera metody dostępne wyłącznie dla workera o roli Admin

- AddUser – Pozwala dodać do listy nowego użytkownika, wraz z jego pozycją (Client, Worker, Manager, Admin)
- ModifyUser – Pozwala zmienić dowolną informację o wybranym użytkowniku, na przykład imię, nazwisko, adres itd.
- DeleteUser – Pozwala usunąć wybranego użytkownika z listy
- ShowLogs - wyświetla logi programu

## Diagram UML

Diagram UML znajduje się w osobnym pliku Emejzon/Documentation/Diagram UML Emejzon.png. Stworzony był na początku projektu, a od tamtego czasu zaszły drobne zmiany w strukturze programu, jednak dobrze przedstawia ogólną strukturę programu.

## Obsługa błędów

Przewidujemy różne błędy od strony użytkownika, takie jak literówki, czy nie podanie danej w zapytaniu. Blokujemy możliwość dodania nowego konta, którego dane pokrywają się z istniejącymi (taki sam e-mail czy numer telefonu). Przewidujemy też błędy zapytań w bazie danych, informując użytkownika o, na przykład, nieistniejącym ID. Wszystkie błędy są komunikowane w konsoli.

## Testowanie

Nie posiadamy testów jednostkowych. Aplikacja testowana była ciągle w trakcie powstawania, a wszystkie błędy były poprawiane na bieżąco.

## Problemy i ograniczenia

Proces instalacji mógłby być łatwiejszy, aktualnie jest dosyć skomplikowany i wymaga instalacji kilku innych programów. Dodatkowo można by rozszerzyć możliwości edycji danych klienta. Obecnie po stworzeniu konta klienta, nie można zmienić jego danych, a jedyną opcją jest stworzenie nowego konta, co na pewno nie jest najlepszym rozwiązaniem.

## Plany rozwoju

W przyszłości warto by było przenieść aplikację na Entity Framework, dla prostszej obsługi. Warto by było też stworzyć własny interfejs dla programu, aby nie korzystać tylko z konsoli, a używanie systemu było prostsze

## Autorzy

Filip Radke kontakt: [filip.radke@uczen.zsk.poznan.pl](mailto:filip.radke@uczen.zsk.poznan.pl)

Wiktor Waligórski kontakt: [wiktor.waligorski@uczen.zsk.poznan.pl](mailto:wiktor.waligorski@uczen.zsk.poznan.pl)

