

GSMTool - System do zarządzania serwisem GSM

Autorzy projektu:

- Hubert Szmyt (Nr. Albumu: 138428)
- Michał Szczombrowski (Nr. Albumu: 138425)

Krótki opis działania projektu

“GSMTool” to aplikacja webowa przeznaczona do zarządzania procesami serwisowymi w serwisie GSM. Umożliwia przyjmowanie urządzeń na serwis, generowanie dokumentów potwierdzających zdanie sprzętu oraz bieżące śledzenie statusu zgłoszeń serwisowych. Dodatkowo system zawiera moduł obsługi magazynu, pozwalający na dodawanie części/akcesoriów do listy. Moduł magazynu również posiada możliwość generowania indywidualnych QR kodów w celu łatwiejszego oznaczania części oraz akcesoriów.

Aplikacja usprawnia pracę serwisów GSM, zapewnia przejrzystość procesów oraz łatwość w dokumentacji i zarządzaniu zasobami. Aplikacja jest responsywna i przystosowana do pracy na urządzeniach mobilnych oraz desktopowych.

Specyfikacja technologii

Backend:

- C# .NET 8.0
- ASP.NET Core MVC

Baza danych:

- MySQL 8.0 (poprzez XAMPP)
- Entity Framework Core 7.0.14
- Pomelo.EntityFrameworkCore.MySql 7.0.0

Frontend:

- HTML5
- CSS3
- JavaScript (ES6+)
- Bootstrap 5.3
- jQuery 3.6
- Toastr.js 2.1.4
- QRCode.js

Instrukcja pierwszego uruchomienia

1. Wymagania wstępne:

- Visual Studio 2022
- .NET 8.0 SDK
- XAMPP z MySQL

2. Konfiguracja bazy danych:

- Uruchom XAMPP i włącz MySQL
- Otwórz phpMyAdmin
(<http://localhost/phpmyadmin>)
- Utwórz nową bazę danych o nazwie "gsm_tool"
- Zaimportuj plik gsm_tool.sql

3. Uruchomienie projektu:

- Otwórz plik `GSMTool.sln` w Visual Studio 2022
- Przywróć pakiety NuGet
- Skompiluj i uruchom projekt

4. Domyślne konto administratora:

- Login: admin
- Hasło: admin123

Struktura projektu

- Controllers/: Kontrolery MVC
- Models/: Modele danych i ViewModels
- Views/: Widoki MVC
- Data/: Kontekst bazy danych

- wwwroot/ Pliki statyczne (CSS, JS, obrazy)

Modele danych

User - Model odpowiedzialny za przechowywanie informacji o użytkownikach systemu.

Device - Model reprezentujący urządzenie przyjęte do serwisu.

Part - Model reprezentujący część w magazynie.

Kontrolery

AccountController

Odpowiada za autoryzację i uwierzytelnianie.

- Login (GET): Wyświetla formularz logowania
- Login (POST): Obsługuje proces logowania
- Logout (POST): Wylogowuje użytkownika

AdminController

Zarządzanie użytkownikami (dostępne tylko dla administratorów).

- Users (GET): Lista wszystkich użytkowników
- CreateUser (GET): Formularz tworzenia użytkownika
- CreateUser (POST): Tworzy nowego użytkownika
- DeleteUser (POST): Usuwa użytkownika

- UpdateRole (POST): Aktualizuje rolę użytkownika

DeviceController

Zarządzanie zleceniami serwisowymi.

- Index (GET): Lista urządzeń
- Create (GET/POST): Dodawanie nowego urządzenia
- UpdateStatus (POST): Aktualizacja statusu
- PrintOrder (GET): Generowanie wydruku zlecenia

PartsController

Zarządzanie magazynem części.

- Index (GET): Lista części
- Create (GET/POST): Dodawanie nowej części
- UpdateQuantity(POST): Aktualizacja ilości
- Delete (POST): Usuwanie części

System użytkowników

Role

1. Administrator:

- Pełny dostęp do systemu
- Zarządzanie użytkownikami
- Wszystkie operacje na zleceniach i magazynie

2. User:

- Obsługa zleceń serwisowych
- Zarządzanie częściami w magazynie
- Brak dostępu do zarządzania użytkownikami

Dostęp

- Goście: Mogą tylko sprawdzić status naprawy po numerze serwisowym
- Zalogowani użytkownicy: Dostęp do funkcji zależnie od roli

Najciekawsze funkcjonalności

1. System statusów napraw:

- Automatyczne powiadomienia o zmianach statusu
- Historia zmian statusów
- Publiczny podgląd statusu przez klienta

2. Responsywny interfejs:

- Dostosowany do urządzeń mobilnych i desktopowych
- Dynamiczne aktualizacje statusów bez przeładowania strony
- Tryb ciemny (dark mode)

3. Zarządzanie magazynem:

- Śledzenie ilości części