



Wydział Automatyki, Elektroniki i Informatyki

# Projekt z Baz Danych 2

System obsługi przewoźnika komunikacji miejskiej.

Autorzy:

Mateusz Chłopek  
Anna Deja  
Anna Mach  
Artur Marbach  
Sebastian Przyszlak

# 1. Założenia projektu.

Czołowym celem tego projektu było zaprojektowanie oraz zaimplementowanie aplikacji okienkowej umożliwiającej obsługę oraz zarządzanie systemem przewoźnika komunikacji miejskiej przy zastosowaniu łączności z bazą danych. Dodatkowo, system przewiduje trzy możliwe role jakie może objąć w nim użytkownik: pasażer, kierowca oraz zarządca. Fakt ten oznacza, że projekt podzielono na trzy moduły mające na celu obsługę tych trzech funkcjonalności, których szczegółowe założenia przedstawione są poniżej.

W roli pasażera użytkownik powinien mieć możliwość dokonywania czynności charakterystycznych dla rzeczywistych systemów komunikacji miejskiej, tj. przeglądania rozkładów jazdy dla danych linii lub przystanków, a także wyszukiwania dostępnych połączeń poprzez podawanie przystanku początkowego oraz tego docelowego.

Moduł kierowcy ma w założeniu umożliwić użytkownikowi przeglądanie swojego grafiku pracy. Dostęp do tego modułu jest chroniony hasłem, przypisanemu indywidualnie do konta danego kierowcy. W ten sposób pracownik ma wgląd jedynie w swój obowiązujący rozkład, który dodatkowo podzielony jest na tygodnie występujące w danym roku.

Również chroniona hasłem, sekcja zarządcy umożliwia niemal pełną kontrolę nad organizacją systemu oraz jego bazą danych. Zarządca może edytować i usuwać elementy, a także je sortować, wyszukiwać oraz filtrować. Dodatkowo posiada on możliwość automatycznej generacji rekordów opisujących kursy odbyte w danym okresie czasu.

Na koniec, zdecydowano się na implementację powyżej opisanego oprogramowania za pomocą języka C#, który nie tylko umożliwia prosty i wydajny proces projektowania interfejsu graficznego aplikacji, ale także dzięki architekturze .NET posiada wbudowane funkcje i inne narzędzia do obsługi i komunikowania się z bazą danych. Sama baza została oparta na technologii SQL Server, która to jest przystosowana do współpracy właśnie z architekturą .NET. Idąc już w duchu dopasowywania naszych rozwiązań postanowiliśmy wybrać Microsoft Visual Studio jako nasze środowisko programistyczne, będąc natywnym dla C# i .NET narzędziem.

## 2. Zmiany dokonane wobec początkowych założeń.

Wśród początkowych założeń naszego projektu nie zabrakło takich, które musiały zostać zmodyfikowane względem swoich pierwotnych wersji, z czego większość zmian uzasadniają testy oprogramowania. Jako przykład, pierwsze okno programu u zarządcy miało w założeniu umożliwiać przejście do poszczególnych sekcji odpowiedzialnych za informacje przechowywane w bazie danych. Oznaczało to, że na ilość tabel reprezentujących podstawowe pojęcia w zakresie komunikacji miejskiej (i zarządzania nią) miała przypadać taka sama liczba sekcji u zarządcy. Okazało się jednak, że takie rozwiązanie prowadzi do lekkiego chaosu, czy też niewygody podczas korzystaniu z programu. Przykładowo, scalenie możliwości zarządzania trasami wraz z ich wyświetlaniem tylko poprzez sekcję "trasy" okazało się nie być najlepszym rozwiązaniem. Wynika to z tego, iż trasa pozostawiona samej sobie nie ma żadnego sensu - nabiera dopiero znaczenia przy zestawieniu z innymi danymi. Z tego powodu zdecydowaliśmy, że dostęp do przejazdów będzie umożliwiany dopiero po wybraniu konkretnej linii, a do tras - poprzez przejazdy.

Początkowo również planowaliśmy zaimplementować w sekcji “kierowcy” modułu zarządcy funkcję dodawania wybranemu kierowcy przejazdu wybraną linią w dany dzień. Jednakże, w miarę prac nad programem zdecydowaliśmy się na automatyczne generowanie przejazdów, wraz z kursami. Jako efekt tego zmieniła się rola tej zakładki. Ostatecznie, poza przeglądaniem przejazdów, umożliwia ona jedynie dodawanie dni wolnych danemu kierowcy. Wraz ze zmianą naszych planów co do tej zakładki, dodana została także nowa tabela do bazy danych - Absences. Zawiera ona informacje o tym jaki kierowca będzie nieobecny w danym dniu, co umożliwia wykluczenie go z potencjalnego automatycznego dopasowania do kursu podczas jego generacji.

### **3. Wnioski.**

Pomimo naszych największych starań by wykonać diagram klas odpowiedni dla naszego programu już na samym początku, to jednak to zadanie okazało się nie być takie łatwe. Możliwe jest, iż większe doświadczenie pozwoliłoby nam uniknąć konieczności wprowadzania zmian w bazie, jednak na naszym obecnym poziomie to założenie okazało się być nierealne.

Praca nad programem pokazała, że samo konstruowanie wyobrażeń na temat tego jak powinien wyglądać program nie zawsze stanowi najlepsze możliwe rozwiązanie. Testowanie aplikacji wskazało nam jak wiele pozornie logicznych rozwiązań w praktyce okazują się nieintuicyjne. Dłuższe użytkowanie programu ujawniało rzeszę irytujących i marnotrawiących czas drobiazgów: nawet coś tak pozornie niewinnego jak choćby konieczność naciśnięcia przycisku powrotu po każdorazowym wprowadzeniu nowego pracownika. Mimo naszych wielokrotnych prób eliminacji wszelkich napotkanych problemów, to nasz projekt zapewne wciąż posiada liczne niewykryte usterki, których ujawnienie zapewniłoby dopiero skrupulatne testowanie przez potencjalnego użytkownika.

Opisane powyżej doświadczenia pokazały nam wagę dostosowywania naszej aplikacji do pozytywnego odbioru przez grupę docelową oraz przybliżyły nam tematykę zapewniania intuicyjności i łatwości użytkowania zamiast brania pod uwagę jedynie wykonywania przez program założonych działań.