

System informacja na zawodach Formuły Student Polska

Adam Paleczny

Styczeń 2023

1 Założenia projektowe

Celem projektu jest aplikacja bazodanowa obsługująca członków biorących udział w zawodach Formuła Student Polska. Uczestnicy zawodów potrzebują informacji o zespołach biorących udział w zawodach, innych uczestnikach, klasyfikacji końcowej jak również bardziej indywidualnie gdzie jest ich nocleg oraz gdzie znajdują się ich pola namiotowe.

2 Koncepcja diagramów

2.1 Potrzebne tabele

Chcąc uzyskać niezbędne informacje istotne jest stworzenie kilku tabel. Niezbędne tabele to:

1. Członkowie - zawierająca informacje o uczestnikach zawodów
2. Bolidy - zawierająca informacje o samochodach biorących udział w zawodach
3. Zespoły - na zawodach członkowie muszą należeć do zespołów, które są rejestrowane bezpośrednio przez organizatorów
4. Nocleg - miejsca, gdzie członkowie mogą spać
5. Pola namiotowe - numerowane pola namiotowe, gdzie można rozbić namioty

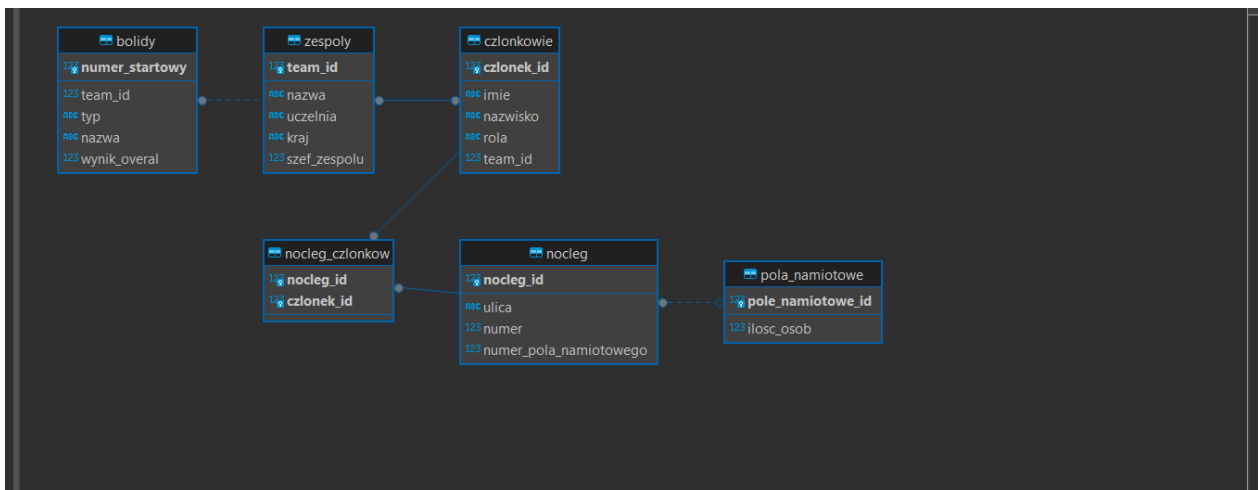
2.2 Relacje pomiędzy tabelami

1. Bolidy - Zespoły : relacja 1:N, gdyż każdy zespół może wziąć udział w zawodach tylko z jednym samochodem, który jest przypisany do zespołu

2. Zespoły - członkowie: Każdy uczestnik zawodów należy do jednego zespołu, a każdy zespół ma członka, który jest szefem zespołu
3. Członkowie - Nocleg : Relacja M:N, do jednego noclegu może być przydzielonych kilku uczestników zawodów
4. Pola namiotowe - Nocleg: Relacja 1:N. Do noclegu przypisane jest pole namiotowe.

2.3 ERD

Schemat ERD:



3 Projekt logiczny bazy danych

3.1 Tabele

3.1.1 Członkowie zawodów

Element	Opis
Członek_ID	ID członka zawodów - klucz główny
Imię	Imię członka
Nazwisko	Nazwisko członka
Rola	Opcjonalny atrybut. Członkowie mogą pełnić role w zespole jak i na zawodach. Są to np Project Managerzy czy szefowie techniczni (CTO)

Tabela 1: Członkowie

3.1.2 Zespoły

Element	Opis
TEAM_ID	ID zespołu - klucz główny
Nazwa	Nazwa zespołu
Uczelnia	Każdy zespół biorący udział w zawodach musi reprezentować jakiś uniwersytet
Kraj	Skrót kraju z którego pochodzi zespół
Szef_zespołu	Klucz obcy z tabeli członkowie

Tabela 2: Zespoły

3.1.3 Bolidy

Element	Opis
Numer_startowy	Każdy samochód biorący udział w zawodach musi mieć indywidualny i unikalny numer startowy - klucz główny
TEAM_ID	Klucz obcy z tabeli zespoły
Typ	Bolidy są elektryczne "EV" lub spalinowe "CV"
Nazwa	Samochody mogą mieć swoje nazwy
Wynik_overal	Wynik samochodu w zawodach

Tabela 3: Bolidy

3.1.4 Pola namiotowe

Element	Opis
Pole_namiotowe_ID	Numer pola namiotowego na którym można spać - klucz główny
Ilosc_osob	Ilość miejsc na danych polu namiotowym

Tabela 4: Pola namiotowe

3.1.5 Nocleg

Element	Opis
Nocleg_ID	ID noclegu - klucz główny
Ulica	Adres na którym znajdują się nocleg
Numer	Numer na ulicy
Numer_pola_namietowego	Klucz obcy do tabeli pola_namietowe

Tabela 5: Nocleg Table

3.1.6 Nocleg członków

Z racji, że w jednym noclegu może być więcej niż jeden członek potrzebna jest tabela asocjacyjna, która łączy zawodników i ich noclegi. Kluczem głównym w tabeli jest para (nocleg_id, czlonek_id).

Element	Opis
Nocleg_ID	Klucz obcy do tabeli nocleg
Czlonek_ID	Klucz obcy do tabeli czlonkowie

Tabela 6: Nocleg_czlonkow

3.1.7 SQL

Dokładny kod sql do stworzenia tabel można znaleźć w katalogu **sql** w pliku *'create_table.sql'*.

3.2 Triggery

3.2.1 Dodowanie do nocelgu

Istotnym elementem bazy danych jest to, żeby podczas wstawiania nowego uczestnika do bazy danych istniała opcja dodania go do noclegu, który już istnieje. Istotna jest również walidacja otrzymanych danych do tabeli nocleg-czlonkow w celu uniknięcia mieszania uczestników z innych zespołów do tego samego noclegu.

Równie istotne jest sprawdzanie czy nie dodajemy uczestnika do noclegu, na którym nie ma już dla niego miejsca.

Dzięki temu otrzymujemy 3 różne triggery, które dokonują walidacji czy nowy członek ma miejsce na zawodach, a także czy dane są wprowadzane poprawnie.

3.2.2 Usuwanie zespołów

W przypadku gdy zespół wyresztuje wszystkich swoich zawodników poza szefem zespołu, który ma służyć jako główna osoba do kontaktu dla organizatorów zespół automatycznie zostaje usunięty z bazy danych wraz z szefem zespołu. Wymaganiem uczestnictwa w zawodach jest posiadanie przynajmniej dwóch reprezentantów, gdyż nie ma możliwości podejścia do konkurencji z jednym zawodnikiem.

4 Interfejs użytkownika

The screenshot shows the 'Formula Student Polska System Informacji' web application. The interface is divided into two main sections: 'INFORMACJE O ZESPOŁACH I CZŁONKACH' and 'WYNIKI ZAWODÓW'.

INFORMACJE O ZESPOŁACH I CZŁONKACH

Buttons: **Załaduj przykładowe dane**, **ZESPOŁY**, **CZŁONKOWIE**

Imię	Nazwisko	TEAM	Rola
Jan	Kowalski	AGH Racing	CEO
Adam	Nowak	AGH Racing	CTO EV
Jan	Huniek	AGH Racing	CTO CV
Krzysztof	Ramania	AGH Racing	PM SUSPENSION
Cristian	Hund	TUfast Racing Team e-Techn	CTO CV
Mathias	Raus	TUfast Racing Team e-Techn	None
Franz	Schulz	TUfast Racing Team e-Techn	CEO
Andreas	Müller	TUfast Racing Team e-Techn	PM Chassis
Eva	Huniek	Ecurie Aix Formula Student T1	DPM Aerodynamics
Julian	Mark	Ecurie Aix Formula Student T1	Electronics Member

WYNIKI ZAWODÓW

Buttons: **CV**, **EV**, **OVERALL**

Wynik	Zespół	Numer startowy	Nazwa samochodu
1	AGH Racing	43	LEM
2	TUfast Racing Team e-Technology	8	ELE2.0
3	NTHU Racing	4	RTE2.0
4	Ecurie Aix Formula Student Team RWTH A	23	ELECTRIC_RUN
5	Dynamis PRC	128	Velocity Vortex
6	GreenTeam Uni Stuttgart	95	Turbo Titan
7	Fast Forest	140	Sonic Surge

Rejestracja nowych członków:

Form fields: Imię, Nazwisko, TEAM, Rola.

Buttons: **Zarejestruj uczestnika**, **Usuń uczestnika**

Wyszukaj informacja dla członka:

Form fields: Wprowadź imię uczestnika, Wprowadź nazwisko uczestnika.

Buttons: **Nocleg**

Ulica | **Numer** | **Pole namiotowe** | **Lokatorzy**

Empty table area for location data.

4.1 Opis interfacu

Obok banera informacyjnego dla pierwszej tabeli znajduje się przycisk "Załaduj przykładowe dane". Odpala on skrypty, które usuwają wszystkie istniejące dane i tworzą całą strukturę na nowo. Jest to przydatne w przypadku sprawdzenia funkcjonalności aplikacji.

W lewym rogu znajdują się dwa przyciski "ZESPOŁY" i "CZŁONKOWIE". Pierwszy z nich służy do wyświetlenia zespołów biorących udział w zawodach. Znajdują się tam informacje o nazwie zespołu, uczelni, z której pochodzą oraz ilości członków, którzy pojawili się na zawodach. Drugi przycisk służy do wyświetlania uczestników zawodów - ich imiona i nazwiska, zespołu oraz ich roli w zespole. Wszystkie dane znajdują się w tabeli poniżej przycisków.

Na prawo od informacji ogólnych znajduje się tabeli informująca o wynikach zawodów. Trzy przyciski wskazują wyniki do poszczególnych kategorii jak również dla wyników dla wszystkich bolidów. Przycisk "CV" wskazuje wyniki dla aut spalinowych, "EV" dla elektrycznych, a "OVERALL" dla wszystkich bolidów.

Panel do rejestracji członków służy do zapisania nowego członka na zawody. Wystarczy wpisać jego imię, nazwisko oraz zespół do którego należy. Pole 'Rola' jest mandatoryjne i nie wymaga uzupełnienia. Po naciśnięciu przycisku użytkownik automatycznie zostanie dodany do bazy danych, jak również zostanie mu przydzielone miejsce do spania.

W przypadku chęci wyrejestrowania uczestnika zawodów wystarczy wpisać jego imię i nazwisko oraz nacisnąć przycisk "Usuń uczestnika". Jeśli znajduje się on w bazie danych sprawdzane jest czy jest on szefem zespołu, do którego należy a potem usuwany jeśli nim nie jest i otrzymujemy informację o jego usunięciu. Jeżeli wpisane imię i nazwisko nie znajdują się w bazie danych wyświetla się odpowiedni komunikat.

Ostatni panel służy do informowania uczestników zawodów o ich noclegu. Wystarczy wpisać imię i nazwisko, a członek automatycznie otrzyma informację o miejscu spania jak również o współlokatorach.

5 Implementacja

5.1 Wykorzystane oprogramowanie

Do stworzenia interfejsu graficznego została wykorzystana biblioteka z języka Python *tkinter*, natomiast do obsługi bazy danych służy biblioteka *psycopg2*, która łączy się z bazą danych postgresSQL. Przykładowa baza danych wykorzystana w projekcie to baza danych z platformy ElephantSQL.

5.2 Struktura projektu

Cały interfejs bazy danych znajduje się w klasie *DatabaseViewer*. Zawiera ona metody do tworzenia interfejsu graficznego, metody wykonujących operacje po naciśnięciu przycisków jak również metody potrzebne do łączenia się z bazą danych.

5.3 Przykładowa konfiguracja bazy danych

Projekt bazowo jest umieszczony na bazie danych ElephantSQL, chociaż możliwa jest jego konfiguracja na innej bazie danych. W metodzie `connect_to_database()` klasy *Database-*

Viewer należy skonfigurować połączenie ze swoją bazą danych.

Sposób na konfigurowanie przykładowej bazy danych znajduje się w *README.md* repozytorium.