**WOJSKOWA AKADEMIA TECHNICZNA**

**IM. JAROSŁAWA DĄBROWSKIEGO**

**W WARSZAWIE**



Systemy Baz Danych

Sprawozdanie Lab3

Temat: Obiektowa baza danych dla szpitala (WakandaDB)

**Prowadzący:**  
mgr inż. Maciej Szymczyk

**Wykonali:**Marcin Borowski

Maksym Bugaj  
**Grupa:**I4B4S1

# Identyfikacja klas, metod i związków.

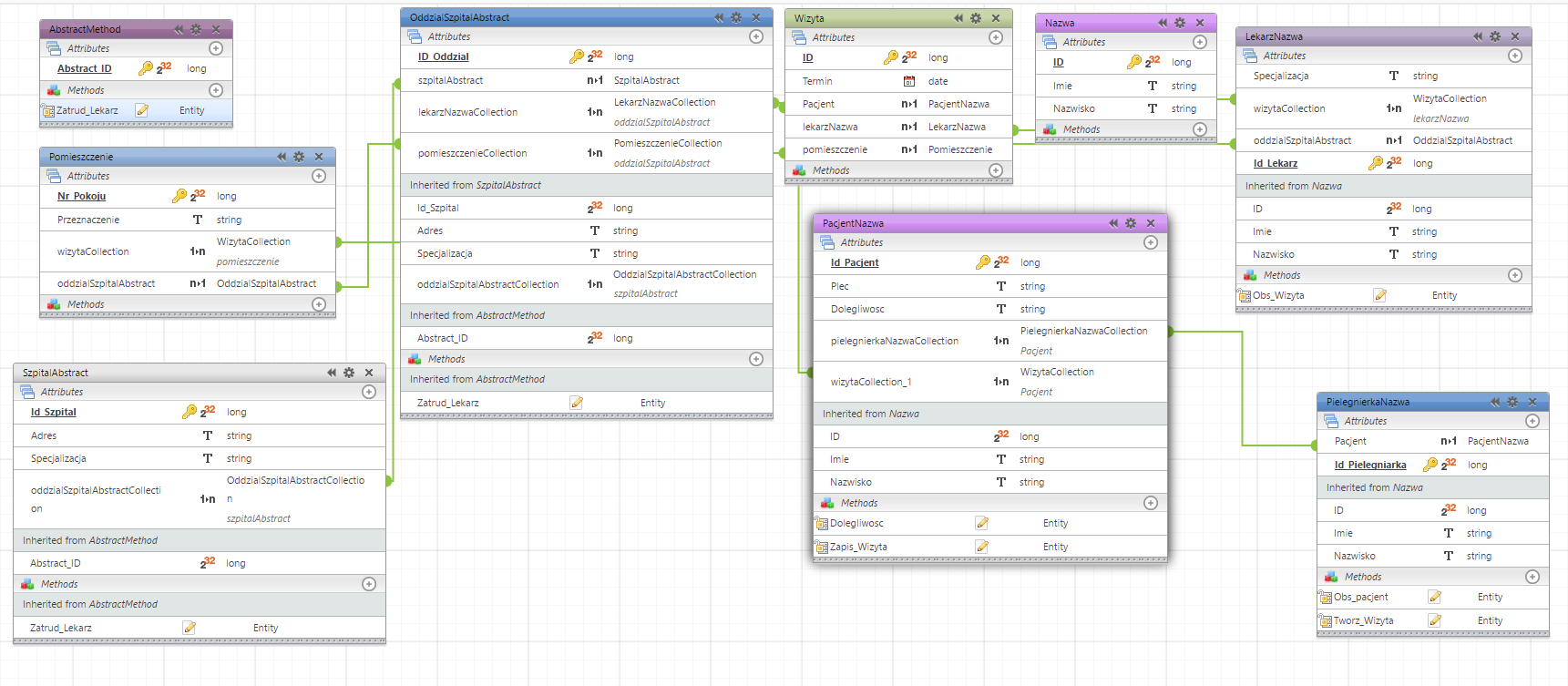
|  |  |
| --- | --- |
| **Nazwa Klasy** | **Opis Klasy** |
| Szpital | Podzielony na oddziały, w nim pracują lekarze i pielęgniarki oraz odbywają się wizyty |
| Oddzial | Posiada odpowiednie specjalizacje i posiada pomieszczenia |
| Pacjent | Osoba fizyczna obsługiwana przez pielęgniarkę oraz umówiona na wizytę |
| Pielegniarka | Osoba fizyczna pracująca w szpitalu mająca za zadanie obsługę pacjenta |
| Lekarz | Osoba fizyczna pracująca w szpitalu mająca za zadanie obsługę wizyt |
| Pomieszczenie | Miejsce w którym odbywają się wizyty |
| Wizyta | Termin spotkania pacjenta z lekarzem umówiony przez pielęgniarkę |
| AbstractMethod | Klasa abstrakcyjna z której inne klasy dziedziczą metody zatrudniania |
| Nazwa | Klasa będąca typem danych z którego korzystają osoby fizyczne |

|  |  |
| --- | --- |
| **Nazwa Metody** | **Opis Metody** |
| Zatrud\_Lekarz | Metoda dziedziczona z klasy abstrakcyjnej mająca na celu dodanie nowego lekarza do szpitala |
| Zatrud\_Pielegniarke | Metoda dziedziczona z klasy abstrakcyjnej mająca na celu dodanie nowej pielęgniarki do szpitala |
| Obs\_Pacjent | Metoda wywoływana przez pielęgniarkę stwierdzająca dolegliwość pacjenta |
| Tworz\_Wizyta | Metoda pozwalająca pielęgniarce na utworzenie nowej wizyty |
| Zapis\_Wizyta | Metoda pozwalająca pacjentowi na zapis na konkretną wizytę |
| Obs\_Wizyta | Metoda pozwalająca lekarzowi na obsługę wizyty |

|  |  |
| --- | --- |
| **Związek** | **Opis związku** |
| Szpital – Oddział | Związek określający do jakiego szpitala należy oddział |
| Oddział – Pomieszczenie | Związek określający do jakiego oddziału należy pomieszczenie |
| Pacjent – Pielęgniarka | Związek określający którym pacjentem zajmowała się która pielęgniarka |
| Szpital – Pielęgniarka | Związek określający w jakim szpitalu pracuje pielęgniarka |
| Szpital – Lekarz | Związek określający w jakim szpitalu pracuje lekarz |
| Pacjent – Wizyta | Związek określający który pacjent umówiony jest na wizytę |
| Lekarz – Wizyta | Związek określający który lekarz obsługuje wizytę |
| Pomieszczenie – Wizyta | Związek określający w jakim pomieszczeniu odbywa się wizyta |

# Implementacja

Do wykonania naszego projektu wykorzystaliśmy bazę danych open source WakandaDB. Jest to baza opierająca się o język JavaScript i współpracująca w sposób intuicyjny z aplikacjami webowymi oraz mobilnymi

Implementacja modelu bazy danych:

# Wypełnienie bazy danych danymi

Bazę danych uzupełnialiśmy według wzoru zaprezentowanego na stronie wakanda.github.io. Nowe dane dodawane są za pomocą skryptu JavaScript w następujący sposób:

ds.Superhero.remove(); *//clean the database from all former Superhero generated*

**new** ds.Superhero({name:"Batman"}).save();

**new** ds.Superhero({name:"Wolverine"}).save();

**var** favoriteSupehero **=** **new** ds.Superhero({name:"Wonder Woman"});

favoriteSupehero.save();

ds.Superhero.all();

Za pomocą tego kodu do przykładowej bazy danych dodawani są super-bohaterowie.

W naszej bazie danych utworzyliśmy odpowiedni plik skryptowy za pomocą którego uzupełnialiśmy bazę danymi o szpitalu jego pracownikach oraz pacjentach. Przykładowe skrypty:

ds.SzpitalAbstract.remove();

new ds.SzpitalAbstract({ID:'1',Adres:"Kaszubska 3, Warszawa",Specjalizacja:"Ogolna"}).save();

ds.SzpitalAbstract.all();

ds.LekarzNazwa.remove();

new ds.LekarzNazwa({Imie:"Kamil",Nazwisko:"Wozniak",Specjalizacja:"Endokrynolog", Id\_Lekarz:'1'}).save();

new ds.LekarzNazwa({Imie:"Krzysztof",Nazwisko:"Sobuszczyk",Specjalizacja:"Internista", Id\_Lekarz:'2'}).save();

new ds.LekarzNazwa({Imie:"Marcin",Nazwisko:"Miziok",Specjalizacja:"Neurolog", Id\_Lekarz:'3'}).save();

new ds.LekarzNazwa({Imie:"Jagoda",Nazwisko:"Strojanowska",Specjalizacja:"Chirurg", Id\_Lekarz:'4'}).save();

ds.LekarzNazwa.all();

Przykładowy wynik działania kodu:



# Pokaz wykonania metod obiektów w terminalu

Niestety w naszej bazie danych utworzenie metod wiązało się z problemami braku wiedzy na temat WakandyDB. Nie wyjaśniono w tutorialach istotnych rzeczy takich jak korzystanie metod używających parametrów podczas ich wykonywania. Dodatkowo nie zostało nigdzie pokazane jak metody wywoływać przez co nie byliśmy w stanie wykonać tego podpunktu sprawozdania

# Umieszczenie projektu w repozytorium git

Poniżej zamieszczam link do repozytorium:

<https://github.com/MarcinBorowski95/SBD_WakandaDB>

# Podsumowanie wykonanej pracy

* Napotkane problemy i ich rozwiązania
  + Brak znajomości środowiska
    - Szukanie informacji w dokumentacji
    - Czytanie bloga na temat środowiska
    - Zapoznanie się z podstawami nowego środowiska
  + Dokumentacja zawierająca małe ilości informacji
    - Szukanie ich w innych źródłach
  + Brak pełnego wytłumaczenia funkcjonalności środowiska
    - Brak rozwiązań
  + Niejasne i nieintuicyjne sposoby tworzenia dziedziczenia oraz związków między klasami
    - Wykorzystanie rozwiązań zaprezentowanych na stronie środowiska
  + Brak jakichkolwiek instrukcji tworzenia oraz wykorzystywania metod w środowisku
    - Brak rozwiązania
* Pomysły/idee
  + Wykorzystanie środowiska w celu utworzenia prostej bazy o szpitalu
  + Możliwość łatwej integracji backendu, frontendu strony internetowej a także aplikacji mobilnej
* Ocena środowiska
  + Przejrzystość aplikacji – Aplikacja jest przejrzysta a w miarę prosta do nawigacji. Niektóre funkcjonalności są jednak „ukryte”
  + Powiązanie z Git’em – Aplikacja sama w sobie jest w stanie bezproblemowo korzystać z repozytorium Git’owego. Ma problemy ze współpracą z innymi aplikacjami komunikującymi się z repozytorium
  + Tworzenie modelu bazy danych – Okno tworzenia modelu może być problematycznie małe. Czasami błędnie reaguje na kliknięcia. Niejasne tworzenie klas abstrakcyjnych, dziedziczenia oraz związków między klasami
  + Całokształt – całościowo aplikacja ma potencjał bycia dobrze rozwiniętym środowiskiem programowym jednak w obecnym stanie nie jest dostatecznie dobra by z niej korzystać. Liczne problemy pojawiają się przez prawie zupełny brak tutoriali do środowiska i rozwiązań błędów.

# Tabela realizacji zagadnień

|  |  |
| --- | --- |
| Jak środowisko realizuje zagadnienia związane z: | Odpowiedź |
| Obiektowością - Metodami | Bardzo kiepsko oraz nieintuicyjnie |
| Obiektowością – Dziedziczeniem, typami abstrakcyjnymi | Nieintuicyjnie |
| Obiektowością – Związkami między klasami (asocjacja, kompozycja) | Nieintuicyjnie ale po wykonaniu związków nie ma z nimi problemu |
| Obiektowością – Typy danych – proste, złożone | Bardzo dobrze, łatwe przypisywanie typów danych oraz tworzenie własnych typów danych |
| Obiektowością - Polimorfizmem | Nie znaleźliśmy możliwości korzystania z polimorfizmu |
| Obiektowością – Tożsamością danych | Dobrze |
| Obiektowością – Enkapsulacją | Tworzone klasy, ich atrybuty i metody nie mają możliwości ustalenia poziomu dostępu do nich |
| Obiektowością – Trwałością danych | Dane są trwałe |
| Administracja – Zarządzanie środowiskiem | Zarządzanie środowiskiem potrafi być problematyczne. |
| Interfejs – Czy narzędzie posiada API? Dla jakich języków? | Narzędzie korzysta z języków JavaScript oraz C++ |
| Środowisko – Czy narzędzie zawiera w sobie środowisko programistyczne? | Tak |
| Skalowalność – Czy narzędzie umożliwia horyzontalne skalowanie środowiska? (rozproszone przetwarzanie, magazynowanie, replikacja) | Nie |
| Multi-model – Czy narzędzie zapewnia inne rodzaje bazy danych? | Nie |