Wydział: Fizyki i Informatyki Stosowanej
Kierunek: Informatyka Stosowana
Rok: 2021/22
Semsetr: zimowy
Typ: stacjonarne
Nr albumu: 401984
Data: 25.01.2022

AGH

Bazy danych I
Dokumentacja projektu

Spis treści

opracował:
Tomasz Szkaradek

1 Projekt koncepcji, założenia

1.1 Zdefiniowanie tematu projektu

W ramach laboratorium z przedmiotu Bazy Danych I zaprojektowałem system zarządzenia szpitalem. System umożliwia nam funkcje zalogowania się jako lekarz bądź członek personelu szpitala. Celem tej bazy jest rozkład obowiązków jak i wymagań miedzy lekarzy a personel oraz odpowiednia obsługa pacjentów. Przykładowo możemy zarezerwować przeprowadzenie operacji.

1.2 Analiza wymagań użytkownika

Baza danych ma za zadanie przede wszystkim :

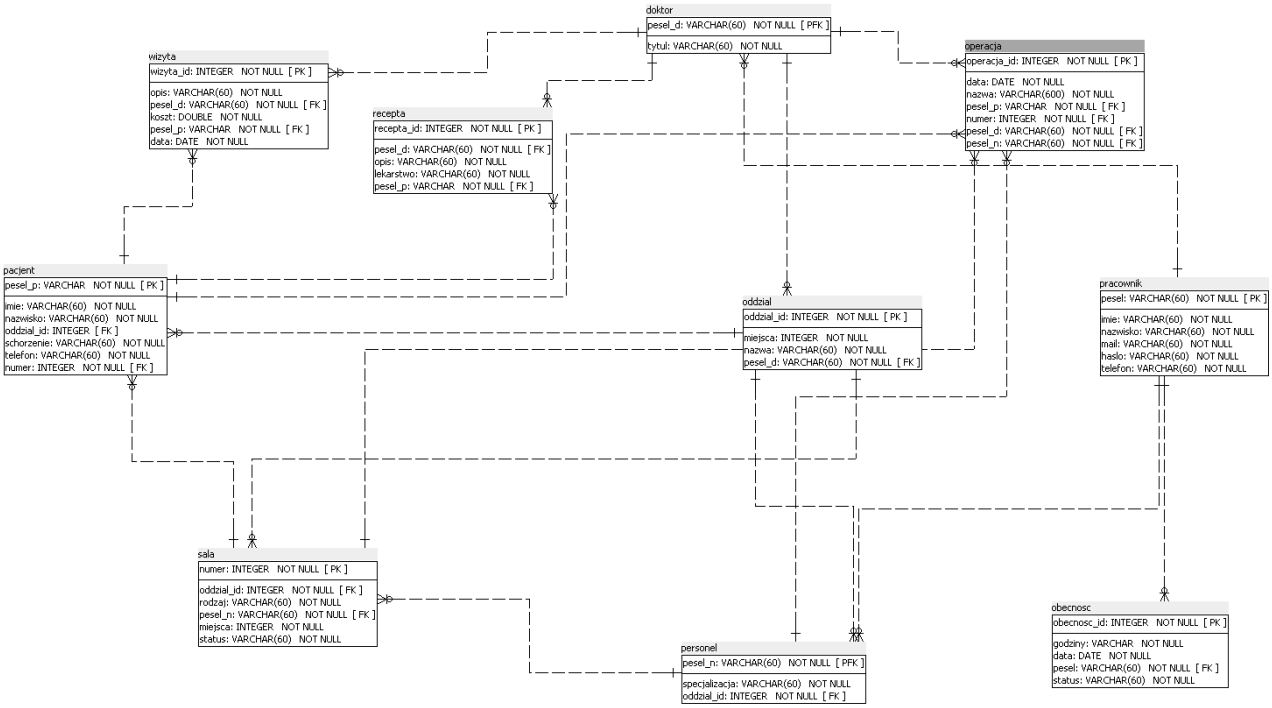
- Rezerwacja operacji
- Wystawianie recept przez lekarza
- Rejestrowanie pacjentów w szpitalu
- Dodawanie nowej sali według wydziału
- Rejestracja obecności
- Rejestracja przeprowadzonej wizyty
- Tworzenie nowych oddziałów jako wydzielonych terytorialnie jednostek (mogą być 2 oddziały o tej samej nazwie jednak nie o tym samym id)
- Wyświetlanie wszelkich powyższych baz z możliwością wybrania encji po której ma znaleźć.Możemy tez wybrać jedną z przygotowanych wcześniej opcji. Dostęp do wyświetlonych baz jest rozróżnialny pomiędzy lekarzem a personelem
- Sprawdzanie dostęp danego użytkownika, waliduje dane oraz odpowiednio nim zarządza.

1.3 Zaprojektowanie funkcji

Funkcjami realizowanymi w bazie danych są miedzy innymi dodawanie i odpowiednia modyfikacja odpowiednich tabel, wyświetlanie wprowadzonych wcześniej danych oraz zapobieganie wprowadzaniu danych, które są nieprawidłowe, takich jak dodanie pacjenta z nieprawidłowym peselem czy umieszczenie pacjenta w złej sali oraz takich, które jeszcze nie istnieją – jeśli nie powinny.

2 Projekt diagramów (konceptualny)

2.1 Budowa i analiza diagramu



Rysunek 1: Diagram ERD bazy szpitala

2.2 Zdefiniowanie encji

Lista encji znajdujących się w bazie danych:

- personel
- doktor
- pacjent
- pracownik
- sala
- operacja
- wizyta
- recepta
- oddzial
- obecnosc

2.3 Zaprojektowanie relacji pomiędzy encjami

Większość relacji pomiędzy tabelami to relacje 1:N.

3 Projekt logiczny

3.1 Projektowanie tabel, kluczy, indeksów

Encja `oddzial` reprezentuje oddział w szpitalu do którego należą np. sale czy personel. Składa się z atrybutów:

- `oddzial_id` - klucz główny INTEGER
- `miejsca` - liczba wolnych etatów dla pracowników INTEGER
- `nazwa` - nazwa oddziału VARCHAR
- `pesel_d` - pesel doktora który założył oddzial klucz obcy VARCHAR

Encja `pracownik` reprezentuje pracownika szpitala. Składa się z atrybutów:

- `pesel` - klucz główny VARCHAR
- `imie` - imie pracownika VARCHAR
- `nazwisko` - nazwisko pracownika VARCHAR
- `mail` - mail konta pracowniczego VARCHAR
- `haslo` - haslo do konta pracowniczego VARCHAR
- `telefon` - telefon pracownika VARCHAR

Encja `doktor` reprezentuje lekarzy w szpitalu. Składa się z atrybutów:

- `pesel_d` - klucz główny, klucz obcy VARCHAR
- `tytul` - nazwa tytułu danego lekarza VARCHAR

Encja `personel` reprezentuje pracownika szpitala. Składa się z atrybutów:

- `pesel_n` - klucz główny,klucz obcy VARCHAR
- `specjalizacja` - nazwa specjalizacji VARCHAR
- `oddzial_id` - numer wydziału na którym pracuje pracownik klucz obcy INTEGER

Encja `pacjent` reprezentuje pacjenta przyjętego do szpitala. Składa się z atrybutów:

- `pesel_p` - klucz główny VARCHAR
- `imie` - imie pacjenta VARCHAR
- `nazwisko` - nazwisko pacjenta VARCHAR
- `oddzial_id` - numer wydziału na którym przebywa pacjent,klucz obcy INTEGER
- `schorzenie` - choroba którą zgłosił pacjent VARCHAR
- `telefon` - telefon pacjenta VARCHAR
- `numer` - sala na której leczy/leczył się pacjent jeśli doktor tak postanowił,klucz obcy INTEGER

Encja `operacja` operacje która ma sie odbyć/odbyła sie. Składa się z atrybutów:

- `operacja_id` - klucz główny INTEGER
- `data` - data przeprowadzenia operacji DATA
- `nazwa` - nazwa operacji VARCHAR
- `pesel_p` - klucz główny VARCHAR
- `numer` - numer sali w której odbyła/odbędzie sie operacja klucz obcy INTEGER
- `pesel_d` - pesel doktora który przeprowadza operacje klucz obcy VARCHAR
- `pesel_n` - pesel personelu który przeprowadza operacje klucz obcy VARCHAR

Encja `sala` reprezentuje sale/pokój przyjęć do szpitala. Składa się z atrybutów:

- `numer` - klucz główny INTEGER
- `oddzial_id` - identyfikator oddziału, klucz obcy INTEGER
- `rodzaj` - rodzaj sali np. operacyjna VARCHAR
- `pesel_n` - pesel pracownika odpowiedzialnego za sale,klucz obcy VARCHAR
- `miejsca` - liczba wolnego miejsca INTEAGER
- `status` - status VARCHAR

Encja `wizyta` reprezentuje wizytę odbytą pomiędzy pacjentem a lekarzem. Składa się z atrybutów:

- `wizyta_id` - klucz główny INTEGER
- `opis` - przebieg wizyty VARCHAR
- `pesel_d` - pesel doktora,klucz obcy VARCHAR
- `koszt` - zapłata dla wizytę INTEAGER
- `pesel_p` - pesel pacjenta,klucz obcy VARCHAR
- `data` - data odbycia wizyty DATE

Encja `recepta` reprezentuje wydaną receptę przez lekarza. Składa się z atrybutów:

- `recepta_id` - klucz główny INTEGER
- `pesel_d` - pesel doktora który wystawił recepte,klucz obcy VARCHAR
- `opis` - opis stosowania lekarstwa VARCHAR
- `lekarstwo` - przypisany medykament VARCHAR
- `pesel_p` - pesel pacjenta,klucz obcy VARCHAR

Encja `obecnosc` reprezentuje obecność pracownika szpitala w danym dniu. Składa się z atrybutów:

- `obecnosc_id` - klucz główny VARCHAR
- `godziny` - liczba przepracowanych godzin a VARCHAR
- `data` - data zaznaczonej obecnosci DATA
- `pesel` - pesel pracownika,klucz obcy VARCHAR
- `status` - status określający obecność np."spoźniony" VARCHAR

3.2 Słowniki danych

oddzial_id	INTEGER	NOT NULL	PRIMARY KEY	klucz główny
miejsca	INTEGER	NOT NULL		pole niepuste
nazwa	VARCHAR	NOT NULL		pole niepuste
pesel_d	VARCHAR	NOT NULL	FOREIGN KEY	pole niepuste

Tabela 1: Encja oddzial

pesel	VARCHAR	NOT NULL	PRIMARY KEY	klucz główny
imie	VARCHAR	NOT NULL		pole niepuste
nazwisko	VARCHAR	NOT NULL		pole niepuste
mail	VARCHAR	NOT NULL		pole niepuste
haslo	VARCHAR	NOT NULL		pole niepuste
telefon	VARCHAR	NOT NULL		pole niepuste

Tabela 2: Encja pracownik

pesel_d	VARCHAR	NOT NULL	PRIMARY FOREIGN KEY	klucz główny
tytul	VARCHAR	NOT NULL		pole niepuste

Tabela 3: Encja doktor

pesel_n	VARCHAR	NOT NULL	PRIMARY FOREIGN KEY	klucz główny
specjalizacja	VARCHAR	NOT NULL		pole niepuste
oddzial_id	INTEGER	NOT NULL	FOREIGN KEY	pole niepuste

Tabela 4: Encja personel

pesel_p	VARCHAR	NOT NULL	PRIMARY KEY	klucz główny
imie	VARCHAR	NOT NULL		pole niepuste
nazwisko	VARCHAR	NOT NULL		pole niepuste
oddzial_id	INTEGER		FOREIGN KEY	pole (nie)puste
schorzenie	VARCHAR	NOT NULL		pole niepuste
telefon	VARCHAR	NOT NULL		pole niepuste
numer	INTEGER	NOT NULL		pole niepuste

Tabela 5: Encja pacjent

operacja_id	INTEGER	NOT NULL	PRIMARY KEY	klucz główny
data	DATE	NOT NULL		pole niepuste
nazwa	VARCHAR	NOT NULL		pole niepuste
pesel_p	VARCHAR	NOT NULL	FOREIGN KEY	pole niepuste
numer	INTEGER	NOT NULL	FOREIGN KEY	pole niepuste
pesel_d	VARCHAR	NOT NULL	FOREIGN KEY	pole niepuste
pesel_n	VARCHAR	NOT NULL	FOREIGN KEY	pole niepuste

Tabela 6: Encja operacja

numer	INTEGER	NOT NULL	PRIMARY KEY	klucz główny
oddzial_id	INTEAGER	NOT NULL	FOREIGN KEY	pole niepuste
rodzaj	VARCHAR	NOT NULL		pole niepuste
pesel_n	VARCHAR	NOT NULL	FOREIGN KEY	pole niepuste
miejsca	INTEGER	NOT NULL		pole niepuste
status	VARCHAR	NOT NULL		pole niepuste

Tabela 7: Encja sala

wizyta_id	INTEGER	NOT NULL	PRIMARY KEY	klucz główny
opis	VARCHAR	NOT NULL		pole niepuste
pesel_d	VARCHAR	NOT NULL		pole niepuste
koszt	INTEGER	NOT NULL		klucz główny
pesel_p	VARCHAR	NOT NULL	FOREIGN KEY	pole niepuste
data	DATE	NOT NULL		pole niepuste

Tabela 8: Encja wizyta

recepta_id	INTEGER	NOT NULL	PRIMARY KEY	klucz główny
pesel_d	VARCHAR	NOT NULL	FOREIGN KEY	pole niepuste
opis	VARCHAR	NOT NULL		pole niepuste
lekarstwo	VARCHAR	NOT NULL		pole niepuste
pesel_p	VARCHAR	NOT NULL		pole niepuste

Tabela 9: Encja recepta

obecnosc_id	INTEGER	NOT NULL	PRIMARY KEY	klucz główny
godziny	VARCHAR	NOT NULL		pole niepuste
data	DATE	NOT NULL		pole niepuste
pesel	VARCHAR	NOT NULL		pole niepuste
status	VARCHAR	NOT NULL		pole niepuste

Tabela 10: Encja obecnosc

3.3 Analiza zależności funkcyjnych i normalizacja tabel

Tabele spełniają założenia trzeciej postaci normalnej. Wartości niekluczowych kolumn zależą od kluczy głównych. Wzajemne zależności pomiędzy kolumnami nienależącymi do klucza nie występują.Każda wartość w bazie jest atomowa.

3.4 Zaprojektowanie operacji na danych

Użytkownik korzystający z aplikacji ma możliwość dodawania rekordów do każdej tabeli poprzez formularze, automatyczna aktualizację statusu sal,obecności wraz z dodaniem daty oraz wyświetlanie raportów dotyczących danych z każdej tabeli w bazie. Większość raportów oparta jest o widoki które były tworzone do bardziej skomplikowanych zapytań. Zostały one stworzone aby czytelniej przedstawić konkretne dane i w łatwiejszy sposób na nich operować natomiast reszta ma za zadanie wydobyć dane informacje za pomocą funkcji agregujących.
WIDOKI:

- `Etaty`
- `Liczba_godzin_według_wydzialow`
- `Średnia_frekfencja`
- `Zarobki_według_pacjentów`
- `Najczęstsze_schorzenia`
- `Oddział_z_największą_liczba_miejsc_w_salach`
- `Oddział_z_najbardziej_zapracowanym_personelem`
- `Doktor_z_największa_liczbą_operacji`
- `Pacjent_z_największą_liczbą_operacji`

Kod wszystkich widoków znajduje się w pliku 'VIEW.sql' w folderze 'SQL'. Zaprojektowane zostały także funkcje oraz wyzwalacze zapobiegające wprowadzeniu niepoprawnych danych do tabel lub do przeszukiwania bazy. FUNCKJE I WALIDATORY

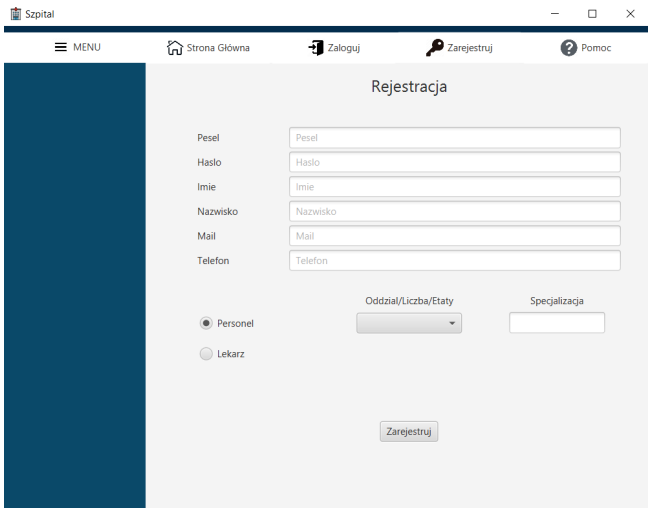
- `pracownik_validate()`
- `obecnosc_validate()`
- `pacjent_validate()`
- `operacja_validate()`
- `wizyta_validate()`
- `find_department(name varchar)`
- `ifPeselDoktor(pesel VARCHAR)`
- `ifPeselPersonel(pesel VARCHAR)`

Kod wszystkich funkcji i wyzwalaczy znajduje się w pliku 'FUNCTION.sql' w folderze 'SQL'. W języku Java zostały też zaimplementowane też co prostrze zapytania SQL, które zapewniają wprowadzanie do tabel poprawnych wartości dotyczących kluczy obcych np. dodanie sali pod warunkiem ze dany oddział istnieje. Walidacja danych takich jak np ten sam lekarz/pracownik personelu podczas operacji nie może być równocześnie pacjentem i pracownikiem wykonywając/asystującym zabieg.

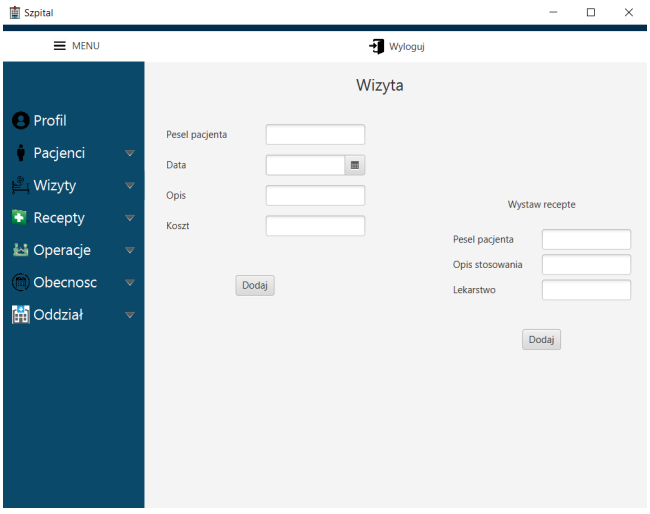
4 Projekt funkcjonalny

4.1 Interfejsy do prezentacji, edycji i obsługi danych

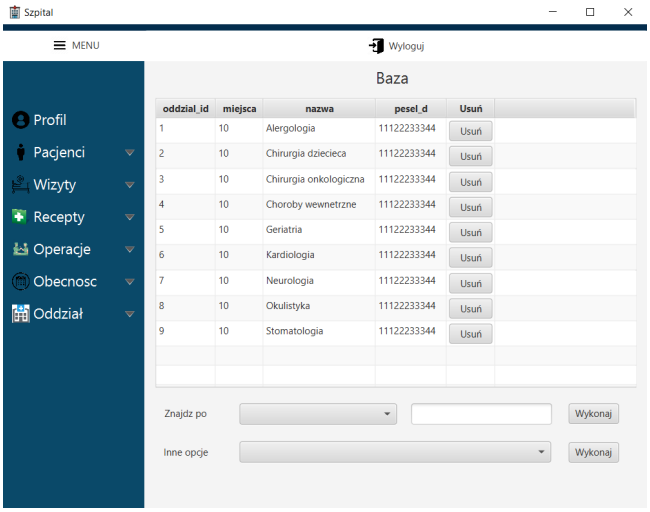
Aplikacja posiada stronę główną. Po uruchomieniu możemy sie zalogować lub zarejestrować wprowadzając dane do formularza. Następnie po zalogowaniu ukazuje się nam interfejs właściwy z formularzami do wprowadzania danych i tabelami do ich przeglądania i analizowania



(a) Formularz do rejestracji



(b) Przykladowy formularz do wprowadzania danych



Rysunek 3: Przykładowa baza

4.2 Wizualizacja danych

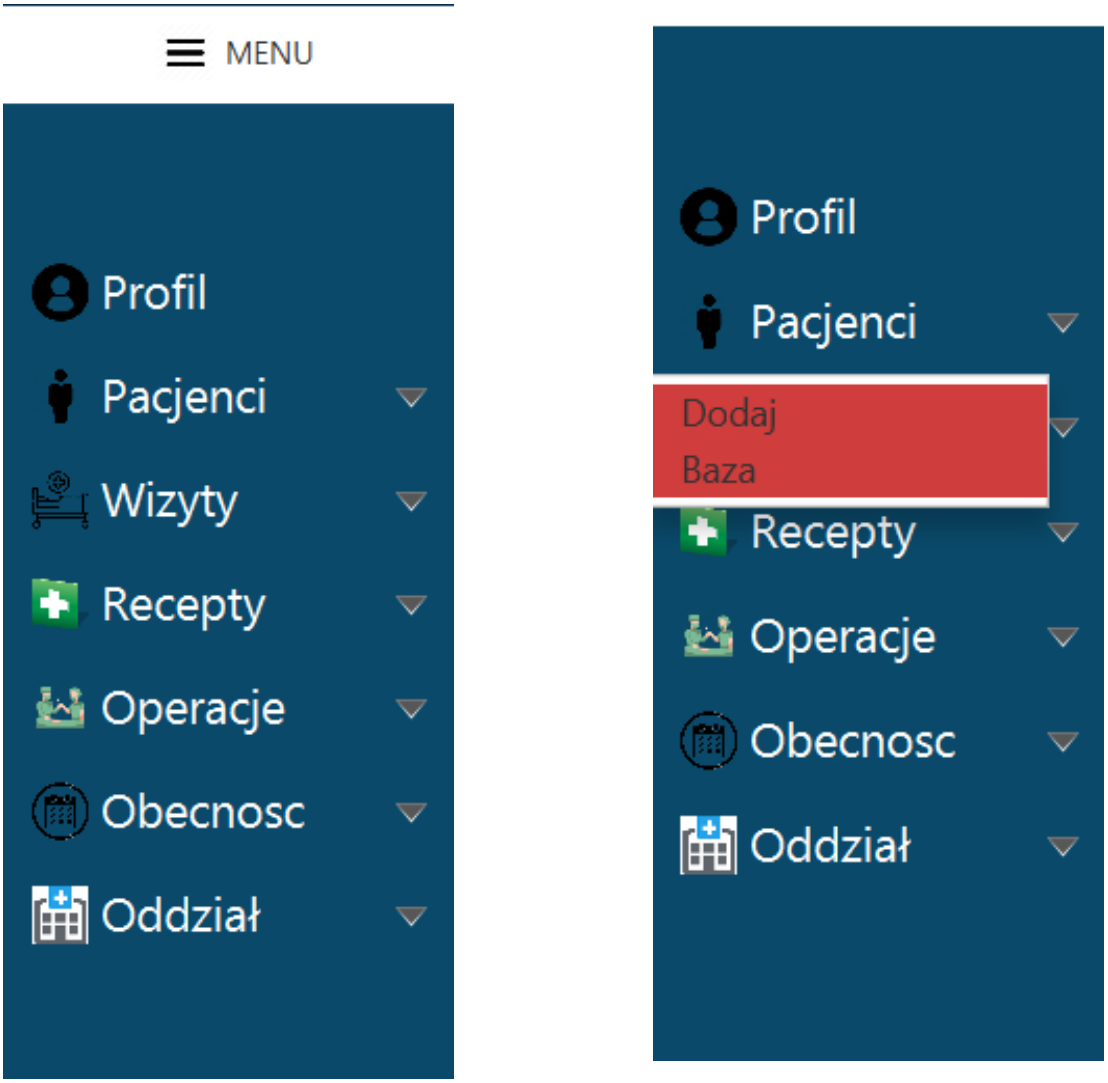
Aplikacja posiada także możliwość wyświetlania wszystkich rekordów wraz z możliwością do ich usuwania jak do ich odpowiedniego przeszukiwania predefiniowanymi opcjami na spodu aplikacji. Przykładowa tabela wyników ukazująca wszystkie oddziały szpitala

Wyświetlanie danych obejmuje wyświetlenie:

- Pacjentów danego lekarza
- Oddziałów
- Wizyt przeprowadzonych przez danego lekarza
- Recept wystawianych pacjentowi
- Obecności danego pracownika
- Sal
- Etaty
- Liczba godzin według wydziałów
- Średnia frekwencja
- Zarobki według pacjentów
- Najczęstsze schorzenia
- Oddział z największą liczba miejsc w salach
- Oddział z najbardziej zapracowanym personelem
- Doktor największa liczbą operacji
- Pacjent z największą liczbą operacji

4.3 Zdefiniowanie panelu sterowania aplikacji

Aby umożliwić swobodne poruszanie się po aplikacji zapewnia ona odpowiednie menu do przechodzenia miedzy formularzami i tabelami. Po naciśnięciu danego przycisku na menu wyświetla się rozwijana lista z opcją przejścia do formularza lub do bazy



Rysunek 4: Wyglad menu

5 Dokumentacja

5.1 Wprowadzanie danych

Poza przedefiniowanymi danymi wprowadzonymi na początek do bazy danych (Pliku INSERT.sql znajduje się kod wprowadzający przykładowe dane do każdej tabeli informacje znajdujące się w każdym poleceniu INSERT są poprawne z założeniami), wszystkie nowe informacje wprowadzane są ręcznie za pomocą formularzy zawartych w aplikacji graficznej.Użytkownik powinien wpisywać dane ręcznie, w miejscach na to przeznaczonych. Formularze pozwalają dane wpisać do takich tabel jak Pacjent, Operacja,Wizyta,Recepta,Obecnosc,Sala,Odzial W folderze "SQL" znajduje się:

- `DROP` - plik do szybkiego usunięcia bazy
- `FUNCTION_TRIGGERS` - plik z funkcjami i trigerami
- `INSERT` - plik zawierający wszystkie komendy
- `INSERT_TABLES` - plik wprowadzający podstawowe dane
- `ROLES` - plik z rolami oraz przywilejami
- `TABLES` - plik ze strukturą bazy wygenerowany za pomocą PowerArchitecta z diagramu
- `VIEWS` - plik zawierający widoki

5.2 Dokumentacja użytkownika

Przed uruchomieniem programu należy stworzyć bazę danych wykonując skrypt INSERT.sql Program otwiera sie za na 2 sposoby:

- komendy: java -jar [ścieżka do pliku jar] [url do bazy]
  np: `java -jar C:\Users\tomek\Desktop\Projekt\out\artifacts\Projekt_jar\Projekt.jar "jdbc:postgresql://localhost:5432/cos"`

- Ustawiając url do połączenia sie z bazą danych w pliku tekstowym URL.txt Następnie uruchomić plik Projekt.jar znajdujący się w folderze Projekt\jar

5.3 Dokumentacja techniczna

Dokumentacja techniczna aplikacji klienta została wygenerowana za pomocą javadoc którego plik znajduje sie w Foldzerze JavaDoc pod nazwa "javadoc.pdf".Dokumentacja ta zawarta jest również w kodzie

5.4 Wykaz literatury

[1] Strona profesora A. Dydejczyka
https : *//newton.fis.agh.edu.pl/ antek/index.php?sub = dbase.*

[2] Dokumentacja Javy
https : *//docs.oracle.com/en/java/javase/*11*/docs/api/*

[3] Stackoverflow
https : *//stackoverflow.com*

[4] Dokumentacja biblioteki JavaFx
https : *//docs.oracle.com/javase/8/javase − clienttechnologies.htm*