

Dokumentacja

Projekt Bazy Danych 1

Szpital

Krzysztof Łakomy
Informatyka Stosowana
Wydział Fizyki i Informatyki Stosowanej

Spis treści

Projekt koncepcji, założenia	2
1.1. Zdefiniowanie tematu projektu	2
1.2. Analiza wymagań użytkownika	2
1.3. Zaprojektowanie funkcji	2
Projekt diagramów	3
2.1. Budowa i analiza diagramu przepływu danych (DFD)	3
2.2. Zdefiniowanie encji (obiektów) oraz ich atrybutów	4
2.3. Zaprojektowanie relacji pomiędzy encjami	6
Projekt logiczny	7
3.1. Projektowanie tabel, kluczy, indeksów	7
3.2. Słowniki danych	10
3.3. Analiza zależności funkcyjnych i normalizacja tabel	10
3.4. Denormalizacja struktury tabel	10
3.5. Zaprojektowanie operacji na danych	10
Projekt funkcjonalny	11
4.1. Interfejsy do prezentacji, edycji, i obsługi danych	11
4.2. Wizualizacja danych	13
4.3. Zdefiniowanie panelu sterowania aplikacji	14
4.4. Makropolecenia	14
Dokumentacja	14
5.1. Wprowadzanie danych	14
5.2. Dokumentacja użytkownika	14
5.3. Opracowanie dokumentacji technicznej	14
5.4. Wykaz literatury	14

1. Projekt koncepcji, założenia

1.1. Zdefiniowanie tematu projektu

Zaprojektowanie aplikacji korzystającej z bazy danych, która odpowiada funkcjonowaniu szpitala. Szpital z założenia podzielony jest na dwie części tj. poradnie dla pacjentów oraz oddziały, na których pacjent może leżeć.

Celem jest możliwość łatwego zapisu pacjenta na wizytę u lekarza, możliwość sprawdzenia terminów wizyt, okresów pobytu na oddziale, wykonanych operacji, informacji o chorobach, a także zarządzanie wszystkimi zgromadzonymi danymi.

1.2. Analiza wymagań użytkownika

Baza danych posiada funkcjonalności:

- dodawanie nowego pacjenta
- dodawanie nowo zatrudnionego lekarza
- dodawanie oddziałów
- dodawanie poradni
- dodawanie operacji
- tworzenie wizyt
- możliwość przyjęcia pacjenta na dany oddział
- możliwość zapisania pacjenta na wizytę
- możliwość wyświetlenia listy chorób wraz z objawami i powikłaniami
- możliwość sprawdzenia listy chorób pacjenta oraz wykonanych na nim operacji
- możliwość sprawdzania terminów wizyt oraz pobytów w na oddziale
- możliwość dodania oceny lekarza
- usuwanie lekarzy
- usuwanie poradni
- usuwanie pacjentów

1.3 Zaprojektowanie funkcji

Funkcjami realizowanymi w bazie danych są:

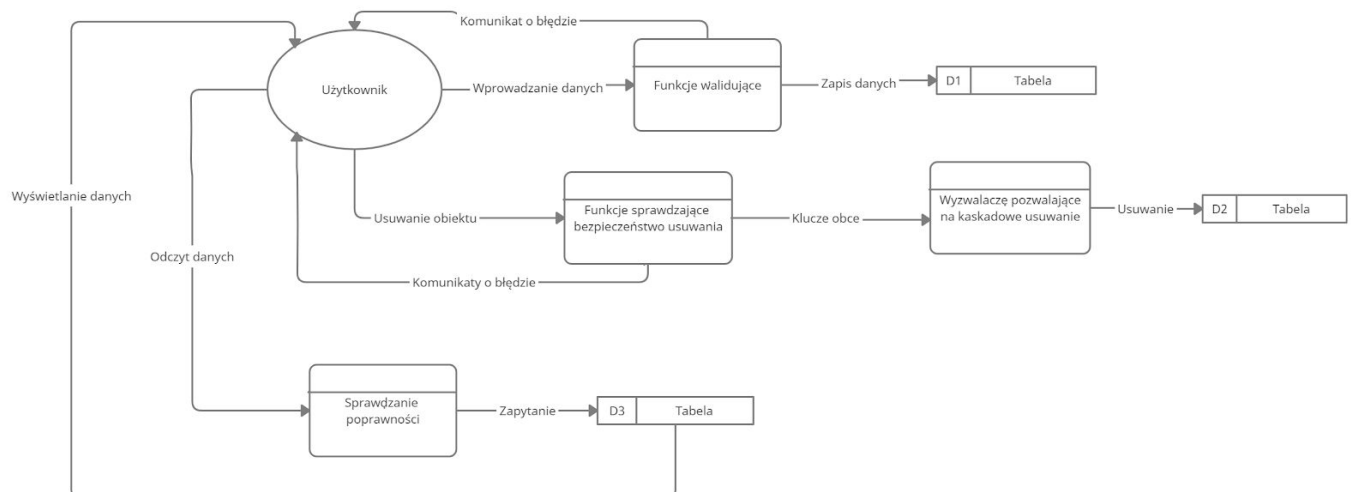
- Kontrola aktualności danych znajdujących się w bazie
- zapobieganie nieprawidłowym akcjom takim jak zapisywanie się na zajęty termin, lub brak wyboru terminu przy zapisie
- zapobieganie np. wykonywania operacji przez lekarzy nie będących na oddziale

- walidacja wprowadzanych danych
- odpowiednie formatowanie wprowadzanych danych
- możliwość bezpiecznego usuwania pacjentów z bazy przy jednoczesnym zwolnieniu terminu np. wizyty u lekarza
- bezpieczne usuwanie lekarzy
- bezpieczne usuwanie poradni

2. Projekt diagramów

2.1. Budowa i analiza diagramu przepływu danych (DFD)

W projekcie występują trzy drogi przepływu, wprowadzanie danych, usuwanie danych oraz pobieranie danych.



2.2. Zdefiniowanie encji (obiektów) oraz ich atrybutów

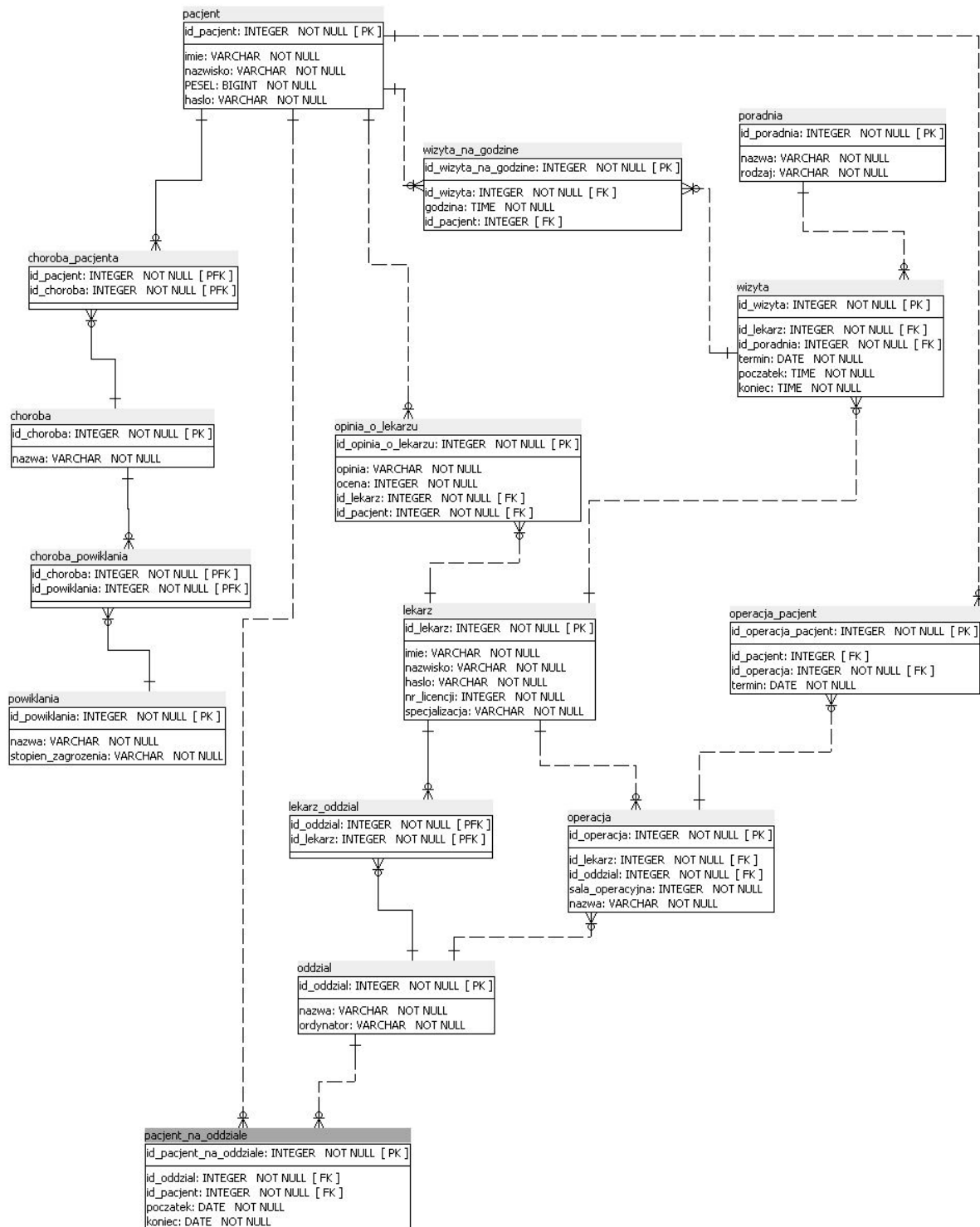
W bazie danych definiujemy następujące encje:

- pacjent
 - id_pacjent
 - imię
 - nazwisko
 - PESEL
- lekarz
 - id_lekarz
 - imię
 - nazwisko
 - hasło

- nr_licencji
 - specjalizacja
- oddział
 - id_oddział
 - nazwa
 - ordynator
- operacja
 - id_lekarz
 - id_oddział
 - sala_operacyjna
 - nazwa
- poradnia
 - id_poradnia
 - nazwa
 - rodzaj
- wizyta
 - id_wizyta
 - id_lekarz
 - id_poradnia
 - termin
 - początek
 - koniec
- wizyta_na_godzinę
 - id_wizyta_na_godzinę
 - id_wizyta
 - godzina
 - id_pacjent
- opinia_o_lekarzu
 - id_opinia_o_lekarzu
 - opinia
 - ocena
 - id_lekarz
 - id_pacjent
- choroba
 - id_choroba
 - nazwa
- powiklania
 - id_powiklania
 - nazwa
 - stopień_zagrozenia

- choroba_powiklania
 - id_choroba
 - id_powiklania
- choroba_pacjenta
 - id_pacjent
 - id_choroba
- pacjent_na_oddziale
 - id_pacjent_na_oddziale
 - id_oddzial
 - id_pacjent
 - początek
 - koniec
- lekarz_oddzial
 - id_oddzial
 - id_lekarz
- operacja_pacjent
 - id_operacja_pacjent
 - id_pacjent
 - id_operacja
 - termin

2.3. Zaprojektowanie relacji pomiędzy encjami



Najczęstszym typem relacji jest relacja 1:N. Występuje ona dla większości z tabel.

Każdą relację N:M rozwiązujemy stosując tablice asocjacyjne. Występują one między tabelami:

- lekarz i oddział
- wizyta i pacjent
- lekarz i pacjent

- operacja i pacjent na oddziale
- pacjent i choroba
- powikłania
- pacjent i oddział

Używamy w tym celu odpowiednio następujących encji asocjacyjnych, niektórych prostych a innych rozszerzonych o dodatkowe kolumny:

- lekarz_oddzial oraz operacja
- wizyta_na_godzinie
- opinia o lekarzu
- operacja_pacjent
- choroba_pacjenta
- choroba_powiklania
- pacjent_na_oddziale

3. Projekt logiczny

3.1. Projektowanie tabel, kluczy, indeksów

Tabela *pacjent* reprezentuje wszystkich pacjentów szpitala przebywających na oddziałach jak i korzystających z wizyt składa się z następujących atrybutów:

- id_pacjent - klucz główny, automatycznie generowany numer typu BIGSERIES, zaczyna się od 1 i wzrasta o 1 przy każdej próbie wpisania nowego rekordu
- imie - atrybut odpowiadający imieniu pacjenta, jest typu VARCHAR
- nazwisko - atrybut reprezentujący nazwisko danego pacjenta, jest typu VARCHAR
- PESEL - atrybut przechowujący numer PESEL pacjenta, jego typ to BIGINT, jest liczbą jedenastocyfrową
- haslo - atrybut przechowujący ustalone hasło pacjenta, jest typu VARCHAR

Tabela *lekarz* reprezentuje wszystkich lekarzy pracujących w szpitalu, na oddziałach i w poradniach, posiada następujące atrybuty:

- id_lekarz - klucz główny, automatycznie generowany numer typu BIGSERIES, zaczyna się od 1 i wzrasta o 1 przy każdej próbie wpisania nowego rekordu
- imie - atrybut reprezentujący imię lekarza, jego typ VARCHAR
- nazwisko - atrybut typu VARCHAR reprezentujący nazwisko lekarza
- haslo - atrybut przechowujący hasło, jest typu VARCHAR
- nr_licencji - atrybut przechowujący numer licencji lekarskiej danego lekarza, jest typu INTEGER oraz jest liczbą minimum dwucyfrową
- specjalizacja - atrybut przechowujący specjalizację lekarza, jest typu VARCHAR

Tabela *oddzial* reprezentuje oddziały występujące w szpitalu, składa się z następujących atrybutów:

- id_oddzial - klucz główny, automatycznie generowany numer typu BIGSERIES, zaczyna się od 1 i wzrasta o 1 przy każdej próbie wpisania nowego rekordu
- nazwa - atrybut typu VARCHAR reprezentujący nazwę danego oddziału
- ordynator - atrybut typu VARCHAR przechowujący informacje jak nazywa się ordynator oddziału

Tabela *operacja* przechowuje operacje wykonywane w szpitalu, posiada następujące atrybuty:

- id_operacja - klucz główny, automatycznie generowany numer typu BIGSERIES, zaczyna się od 1 i wzrasta o 1 przy każdej próbie wpisania nowego rekordu
- id_lekarz - klucz obcy odnoszący się do tabeli lekarz, jest typu INTEGER
- id_oddzial - klucz obcy nawiązujący do tabeli oddzial, jego typ to INTEGER
- sala_operacyjna - atrybut typu INTEGER reprezentujący numer sali, w której przeprowadzana jest operacja
- nazwa - atrybut typu VARCHAR reprezentujący nazwę operacji

Tabela *poradnia* reprezentuje poradnie występujące w szpitalu, jej atrybuty to:

- id_poradnia - klucz główny, automatycznie generowany numer typu BIGSERIES, zaczyna się od 1 i wzrasta o 1 przy każdej próbie wpisania nowego rekordu
- nazwa - atrybut typu VARCHAR reprezentujący nazwę danej poradni
- rodzaj - atrybut przechowujący rodzaj poradni, jego typ to VARCHAR

Tabela *wizyta* reprezentuje wizytę prowadzoną przez doktora w danym terminie, posiada następujące atrybuty:

- id_wizyta - klucz główny, automatycznie generowany numer typu BIGSERIES, zaczyna się od 1 i wzrasta o 1 przy każdej próbie wpisania nowego rekordu
- id_lekarz - klucz obcy odnoszący się do tabeli lekarz, jest typu INTEGER
- id_poradnia - klucz obcy nawiązujący do tabeli poradnia, jest typu INTEGER
- termin - atrybut typu DATE reprezentujący termin, w którym wizyta się odbywa
- poczatek - atrybut przechowujący godzinę rozpoczęcia wizyty, jest typu TIME
- koniec - atrybut typu TIME przechowujący godzinę zakończenia wizyty, posiada wartość zawsze większą od atrybutu początek

Tabla *wizyta_na_godzinie* reprezentuje powiązanie pacjenta z wizytą o określonej godzinie ustaloną przez konkretnego lekarza, jej atrybuty to:

- id_wizyta_na_godzinie - klucz główny, automatycznie generowany numer typu BIGSERIES, zaczyna się od 1 i wzrasta o 1 przy każdej próbie wpisania nowego rekordu
- id_wizyta - klucz obcy nawiązujący do tabeli wizyta, jest typu INTEGER
- id_pacjent - klucz obcy odnoszący się do tabeli pacjent, jest typu INTEGER
- godzina - atrybut typu TIME reprezentujący daną godzinę określonej wizyty

Tabela *operacja_pacjent* reprezentuje powiązanie pacjenta z operacją na daną ustaloną godzinę, składa się z następujących atrybutów:

- id_operacja_pacjent klucz główny, automatycznie generowany numer typu BIGSERIES, zaczyna się od 1 i wzrasta o 1 przy każdej próbie wpisania nowego rekordu
- id_pacjent - klucz obcy odnoszący się do tabeli pacjent, jest typu INTEGER
- id_operacja - klucz obcy nawiązujący do tabeli operacja, jest typu INTEGER
- termin - atrybut przechowujący termin, w którym odbywa się operacja danego pacjenta, jego typ to DATE

Tabela opinia_o_lekarzu reprezentuje powiązanie między pacjentem i lekarzem w formie opinii danego pacjenta o lekarzu, posiada następujące atrybuty:

- id_opinia_o_lekarzu - klucz główny, automatycznie generowany numer typu BIGSERIES, zaczyna się od 1 i wzrasta o 1 przy każdej próbie wpisania nowego rekordu
- opinia - atrybut przechowujący opinię pacjenta o lekarzu, jest typu VARCHAR
- ocena - atrybut przechowujący ocenę w skali od 1 - 5 pacjenta o lekarzu, jest typu INTEGER
- id_pacjent - klucz obcy odnoszący się do tabeli pacjent, jest typu INTEGER
- id_lekarz - klucz obcy nawiązujący do tabeli lekarz, jego typ to INTEGER

Tabela *pacjent_na_oddziale* reprezentuje pacjenta przebywającego na oddziale w danym okresie, jej atrybuty to:

- id_pacjent_na_oddziale - klucz główny, automatycznie generowany numer typu BIGSERIES, zaczyna się od 1 i wzrasta o 1 przy każdej próbie wpisania nowego rekordu
- id_oddzial - klucz obcy nawiązujący do tabeli oddzial, jest typu INTEGER
- id_pacjent - klucz obcy odwołujący się do tabeli pacjent, jest typu INTEGER
- poczatek - atrybut typu DATE przechowujący datę początku pobytu pacjenta na oddziale
- koniec - atrybut typu DATE przechowujący końcową datę pobytu pacjenta na oddziale

Tabela *choroba* reprezentuje bazę chorób znanych w szpitalu, jej atrybuty to:

- id_choroba - klucz główny, automatycznie generowany numer typu BIGSERIES, zaczyna się od 1 i wzrasta o 1 przy każdej próbie wpisania nowego rekordu
- nazwa - atrybut typu VARCHAR reprezentujący nazwę określonej choroby

Tabela *powiklania* reprezentuje powikłania oraz objawy mogące wystąpić w chorobach, posiada następujące atrybuty:

- id_powiklania - klucz główny, automatycznie generowany numer typu BIGSERIES, zaczyna się od 1 i wzrasta o 1 przy każdej próbie wpisania nowego rekordu
- nazwa - atrybut typu VARCHAR reprezentujący nazwę danego objawu/powikłania
- stopien_zagrozenia - atrybut typu VARCHAR przechowujący informacje jak bardzo niebezpieczny jest dany objaw/powikłanie

Tabela *choroba_powiklania* reprezentuje powiązanie tabeli choroba oraz powiklanie, w taki sposób, że można jedno powiklanie przypisać do wielu chorób oraz wiele powikłań do jednej choroby, posiada następujące atrybuty:

- *id_choroba* - klucz obcy typu INTEGER nawiązujący do tabeli choroba
- *id_powiklanie* - klucz obcy typu INTEGER odnoszący się do tabeli powiklanie

Tabela *choroba_pacjenta* łączy tabele pacjent oraz choroba rozwiązując relację N:M występującą między nimi, jej atrybuty to:

- *id_choroba* - klucz obcy typu INTEGER nawiązujący do tabeli choroba
- *id_pacjent* - klucz obcy typu INTEGER nawiązujący do tabeli pacjent

Tabela *lekarz_oddzial* łączy tabele lekarz oraz oddzial w taki sposób aby móc przypisać do lekarza wiele oddziałów oraz do oddziału wiele lekarzy, posiada następujące atrybuty:

- *id_lekarz* - klucz obcy typu INTEGER nawiązujący do tabeli lekarz
- *id_oddzial* - klucz obcy typu INTEGER nawiązujący do tabeli oddzial

3.2. Słowniki danych

W projekcie nie zostały wykorzystane żadne tabele słownikowe.

3.3. Analiza zależności funkcyjnych i normalizacja tabel

Tabele spełniają warunki trzech postaci normalnych. Dane są przechowywane w sposób atomowy, każde pole przechowuje jedną informację. Każda tabela posiada klucz główny pozwalający jednoznacznie zidentyfikować dany wiersz. W każdej z tabel przechowywane są dane dotyczące tylko jednej klasy obiektów, oznacza to że wszystkie atrybuty danej tabeli zawierają informacje tylko o tej konkretnej klasie, którą mają przechowywać. Dodatkowo każda z kolumn, która nie jest kluczem głównym nie jest zależna od innej kolumny nie będącej kluczem głównym w danej tabeli. Wszystkie kolumny zależą tylko i wyłącznie od klucza a nie od siebie nawzajem.

3.4. Denormalizacja struktury tabel

Działanie jest opcjonalne i nie zostało wykonane.

3.5. Zaprojektowanie operacji na danych

Do łatwego zarządzania danymi został dodany szereg funkcji oraz wyzwalaczy realizujących różne zadania:

create_wizyta_na_godzine - jest to wyzwalacz uruchamiany po utworzeniu wizyty u lekarza w danym terminie i jest odpowiedzialny za dodawanie rekordów do tabeli *wizyta_na_godzine*

w taki sposób aby w okresie między początkiem a końcem wizyty dodane zostały wolne do zapisania się wizyty w godzinnych odstępach czasowych.

usuwanie_lekarzy - to wyzwalacz, który uruchamia się przy próbie usunięcia danego lekarza, przed usunięciem rekordu z tabeli usuwa on wszystkie powiązania z innymi tabelami (te, w których lekarz pojawia się jako klucz obcy) tak aby móc bez błędu usunąć sam rekord z tabeli *lekarz*.

usuwanie_wizyty - to także wyzwalacz uruchamiany przed usunięciem rekordu z tabeli *wizyta*, usuwa on wszystkie powiązania z innymi tabelami, tak aby bezpiecznie usunąć rekord z danej tabeli.

usuwanie_pacjentow - kolejny wyzwalacz uruchamiany przed usunięciem danych z tabeli *pacjent*, który przed samym usunięciem pozbywa się także wszelkich powiązań z innymi tabelami danego rekordu.

usuwanie_oddzialow - wyzwalacz uruchamiany przed usunięciem rekordu z tabeli *oddzial* powodujący uprzednie usunięcie wszelkich wystąpień rekordu jako klucza głównego w innych tabelach.

usuwanie_operacji - ostatni z wyzwalaczy zapewniających bezpieczeństwo przy usuwaniu rekordu z tabeli *operacja*, pozbywając się wszystkich powiązań z innymi tabelami.

dodawanie_pacjenta - wyzwalacz ustawiający imię i nazwisko pacjenta na zaczynające się z wielkiej litery, odpowiada także za poprawność wprowadzonego numeru PESEL.

dodawanie_lekarza - jest to wyzwalacz walidujący poprawność wprowadzonych danych to tabeli *lekarz* zapewnia on, że numer licencji lekarza jest więcej niż jedno cyfrowy oraz, że wprowadzone imię i nazwisko będą z wielkiej litery.

dodawanie_wizyty - kolejny wyzwalacz walidujący zapewniający, iż w tabeli *wizyta* nie godzina rozpoczęcia nie będzie wcześniejsza niż godzina zakończenia wizyty.

dodawanie_pacjenta_na_oddzial - ostatni z wyzwalaczy sprawdzający czy wprowadzana początkowa data pobytu pacjenta na oddziale nie jest późniejsza niż końcowa.

4. Projekt funkcjonalny

4.1. Interfejsy do prezentacji, edycji, i obsługi danych

Na stronie istnieje wiele formularzy oraz przycisków obsługujących edycję danych. Każda z głównych tabel posiada formularze służące do wprowadzania danych do bazy.

W zakładce Lekarze widzimy 2 główne elementy:

- pierwszym z nich jest formularz pozwalający na dodawanie nowego lekarza do istniejącej listy.
- drugim natomiast jest już sama lista wypisująca wszystkich istniejących w szpitalu lekarzy.

Na liście lekarzy dostępny jest szereg przycisków oferujących następujące działania:

- samo kliknięcie na wiersz z wybranym lekarzem przenosi nas na stronę poświęconą dokładniejszą informacją o danym lekarzu
- przycisk Wizyta przenoszący na stronę z zapisami pacjentów na wizyty
- przycisk Edytuj pozwalający na edycję informacji o danym lekarzu
- przycisk Usun usuwający lekarza

W kolejnej z zakładek - Pacjenci ukazuje nam się tabela z wszystkimi pacjentami szpitala.

Dostępne jest:

- kliknięcie na pacjenta na liście powodujące przeniesienie na stronę z informacjami o wybranym pacjencie
- przycisk Usun usuwający pacjenta

Następną zakładką jest Baza chorób. wyświetla ona listę chorób w bazie pozwala ona na dodawanie nowej choroby oraz po uprzednim kliknięciu na pole z chorobą dodawanie objawów lub powikłań do danej choroby.

W zakładce Poradnie także widnieje formularz pozwalający na wprowadzanie nowej poradni do bazy oraz lista z poradniami. W każdym z wierszy dostępne są dwa przyciski:

- Lista wizyt pozwalający wyświetlić wizyty odbywające się w danej poradni
- Usun odpowiadający za usuwanie poradni

Ostatnią z zakładek jest Oddziały, odpowiada ona za wyświetlanie istniejących oddziałów w szpitalu. Na środku widoczna jest lista oddziałów, każdy z przyciskiem Delete pozwalającym na usuwanie. Dodatkowo dostępny jest formularz wprowadzania nowego oddziału oraz przycisk przenoszący na stronę z formularzem pozwalającym na dodawanie pacjenta do oddziału.

Kliknięcie na wybrany oddział powoduje przeniesienie do strony z detalami danego oddziału

Na pasku z zakładkami widnieją także 4 przyciski:

- Operacje jest to przycisk przenoszący nas do strony służącej zarządzaniem operacjami
- Wizyta przenosi na stronę z zapisem na wizyty
- Utworz Wizyte przycisk, który przenosi do strony z formularzem pozwalającym na dodanie nowej wizyty do bazy
- Rejestruj Pacjenta jest to przycisk odnoszący się do strony z formularzem wprowadzania nowego pacjenta do bazy danych

Strona z operacjami posiada listę operacji wraz z terminami oraz dwa przyciski:

- jeden przenosi do formularza wprowadzającego nową operację do bazy
- drugi pozwala na wpisanie pacjenta na daną operację

Na stronie z detalami wybranego lekarza znajdują się:

- dokładniejsze informacje o lekarzu
- oceny i opinie pacjentów
- formularz do dodawania opinii

Strona z dokładnymi informacjami o pacjencie zawiera listę posiadanych chorób przez pacjenta, listę wizyt pacjenta, pobyty pacjenta na oddziale oraz operacje wykonywane na pacjencie. Dodatkowo jest także formularz pozwalający przypisać chorobę z bazy chorób do pacjenta wpisując jej nazwę.

Ostatnią ze stron z detalami jest strona wybranego oddziału na której widnieją informacje o tym jakie operacje są przeprowadzane, jacy lekarze pracują na danym oddziale, oraz dane o pacjentach przebywających na oddziale.

Do dodawania lekarza oraz operacji na oddział służą odpowiednie dwa przyciski wyświetlające formularze wprowadzające.

4.2. Wizualizacja danych

Z powodu potrzeby przejrzystości danych prezentowanych na stronie zostały utworzone następujące widoki:

lekarze - widok wyświetla wszystkie informacje o lekarzu wraz z jego ocenami i opiniami

opinie_o_wybranym_lekarzu - widok przedstawiający opinię o lekarzu wraz z informacją o pacjencie, który ją wystawił

poradnie_wybranego_lekarza - widok wyświetla informacje o lekarzu wraz z danymi o poradni, w której przyjmuje

terminy_wizyt_lekarza - wyświetla kompletne informacje o wizytach danego lekarza, ich terminie, godzinie przyjęcia, a także poradni w której się odbywają sortowane po nazwie poradni

wizyty_pacjenta - przedstawia szczegółowe informacje o wizytach wybranego pacjenta tj. godzinie wizyty, terminie, poradni, w której się odbywa oraz lekarzu u którego jest prowadzona.

oddzialy_pacjenta - wyświetla informacje o oddziale oraz okresie pobytu danego pacjenta na oddziale

operacje_pacjenta - przedstawia dane o operacjach na wybranym pacjencie, takie jak informacje o lekarzu przeprowadzającym operację, rodzaju operacji, terminie oraz sali operacyjnej

objawy_choroby - wyświetla informacje o chorobie wraz z danymi na temat ich objawów i powikłań

doktorzy_na_oddziale - posiada informacje o lekarzach pracujących na danym oddziale

operacje_na_oddziale - widok wyświetlający informacje o przeprowadzanych na oddziale operacjach oraz dane lekarzy prowadzących.

operacje_pacjentów - widok przedstawiający informacje o pacjentach oraz operacjach jakie są na nich wykonywane wraz z lekarzami przeprowadzającymi oraz wszystkimi terminami

4.3. Zdefiniowanie panelu sterowania aplikacji

Obsługa aplikacji jest możliwa z panelu z zakładkami znajdującego się w górnej części aplikacji. Mamy możliwość bezpośredniego dostępu do listy lekarzy, listy chorób wraz z objawami, list oddziałów i poradni oraz listy pacjentów. Wszystko jest łatwo dostępne i pozwala sprawnie poruszać się po aplikacji. Dodatkowo szereg przycisków funkcjonalnych daje dostęp do szybkich zapisów informacji do bazy oraz możliwość łatwego powiązania danych.



4.4. Makropolecenia

Makropolecenia nie zostały użyte w aplikacji, ponieważ wprowadzanie i wyświetlanie informacji jest bardzo proste.

5. Dokumentacja

5.1. Wprowadzanie danych

Dane w aplikacji wprowadzamy przy pomocy formularzy. Każdy element posiada dedykowane dla siebie formularze.

W folderze SQL zawarte są wszystkie polecenia tworzące oraz edytujące bazę danych:

- w pliku *baza.sql* znajduje się kod tworzący całą bazę danych
- w pliku *przykladowe_dane.sql* znalazły się przykładowe startowe dane wprowadzane do bazy
- w pliku *funkcje_i_wyzwalacze.sql* jest kod tworzący potrzebne do działania aplikacji funkcje i wyzwalacze
- w pliku *widoki.sql* zawarte został kod do tworzenia widoków poprawiających przejrzystość prezentowanych danych

5.2. Dokumentacja użytkownika

Działający projekt jest dostępny na stronie internetowej pod adresem:

<http://krzysztoflakomyprojektbd1.s3-website.eu-central-1.amazonaws.com/>

Korzysta on ze zdalnej bazy danych zawartej na serwerze ElephantSQL.

Wraz z dokumentacją dołączony jest kod źródłowy dzielący się na trzy katalogi:

- server - katalog zawierający aplikację klienta zrobioną z wykorzystaniem technologii ReactJS
- client - katalog zawierający część aplikacji po stronie serwera napisanej z wykorzystaniem NodeJS
- SQL - katalog zawierający kod w języku sql wykorzystywany do tworzenia i uzupełnienia bazy

5.3. Opracowanie dokumentacji technicznej

Kod został napisany zgodnie ze standardem. Nazwy funkcji i zmiennych są dobrane w taki sposób aby kod był samodokumentujący się.

5.4. Wykaz literatury

1. <https://pl.reactjs.org/docs/getting-started.html>
2. <https://nodejs.org/en/docs/>
3. <https://www.postgresql.org/docs/13/index.html>
4. Wykład Bazy Danych 1