

Bazy Danych I – Projekt
System zarządzania przychodnią

Kinga Pyrek
Numer albumu: 30390

I. Projekt koncepcji, założenia

1. Zdefiniowanie tematu projektu

Tematem projektu jest stworzenie bazy danej obsługującej działanie przychodni lekarskiej. Celem aplikacji jest zapewnienie funkcjonalności dla trzech typów użytkownika – lekarza, pacjenta i admina.

2. Analiza wymagań użytkownika

Użytkownicy logując się na dany typ konta mają określone funkcjonalności.

Pacjent ma możliwość umówienia wizyty wybierając placówkę, a w niej poradnię, a następnie wybierając wizytę u danego lekarza w dostępnej godzinie. Pacjent ma również możliwość wyszukania lekarza po nazwisku i/lub specjalizacji i przeglądania profilu lekarza oraz wyświetlenia swojego profilu, e-recept do zrealizowania oraz nadchodzących wizyt.

Lekarz ma opcję wyświetlenia swojego profilu – swoich danych, specjalizacji oraz dyżurów, profilu swoich pacjentów (którzy mają umówioną wizytę u niego), wypisania e-recept pacjentowi, dodania dolegliwości pacjenta.

Admin może dodawać placówki, przypisywać do nich poradnie. Może zarządzać lekarzami – dodawać nowych/edytować oraz przypisywać ich do danych poradni, dodawać dyżury oraz gabinety w poradni, wyświetlać statystyki o dyżurach, wyświetlać/ edytować pacjentów oraz wyświetlać listę leków.

3. Zaprojektowanie funkcji

Baza danych ma za zadanie ułatwić użytkownikom korzystanie z usług przychodni za pomocą funkcji

- Dodawanie do bazy placówek, przychodni, gabinetów
- Dodawanie lekarzy i przypisywanie ich do przychodni
- Dodawanie dyżurów lekarzy
- Umówienie wizyty (30 – minutowej) w czasie dyżuru w godzinie, która nie została jeszcze zajęta przez innego pacjenta
- Wypisywanie e-recept oraz przypisywanie pacjentom chorób
- Wyszukiwanie lekarzy

II. Projekt diagramów

1. Encje oraz ich atrybuty

- Placówka – zawiera informacje o placówce
 - Id_placowka – klucz główny
 - Nazwa – nazwa placówki
 - Id_adres – klucz obcy, adres pod jakim znajduje się placówka
- Poradnia_typ – definiuje rodzaj poradni
 - Id_typ – klucz główny

- Nazwa – nazwa typu poradni
- Opis – opis oferty poradni
- Adres – zawiera informacje o adresie
 - Id_adres – klucz główny
 - Miasto
 - Ulica
 - Numer
 - Kod_pocztowy
- Poradnia – zawiera informacje o poradni
 - Id_poradnia – klucz główny
 - Id_placowka – klucz obcy – placówki, do jakiej należy poradnia
 - Budynek – budynek poradni
 - Piętro – piętro, na którym znajduje się poradnia
 - Id_typ – klucz obcy – typu poradni
- Lekarz – zawiera informacje o lekarzu
 - Id_lekarz – klucz główny
 - Imie – imię lekarza
 - Nazwisko – nazwisko lekarza
 - Email – email lekarza
 - Telefon – numer telefonu lekarza
 - Pesel – pesel lekarza
 - Login – login lekarza do serwisu
 - Haslo – hasło lekarza do serwisu
- Specjalizacja – opisuje specjalizacje lekarskie
 - Id_specjalizacja – klucz główny
 - Nazwa – nazwa specjalizacji
 - Uczelnia – uczelnia, na której skończył studia lekarz
 - Rok_otrymania – rok ukończenia specjalizacji, przez lekarza
- Lekarz_poradnia – informacje na temat lekarzy w poradni
 - Id_lekarz – klucz obcy, identyfikator lekarza
 - Id_poradnia – klucz obcy, identyfikator poradni
- Lekarz_specjalizacja – zawiera informacje o specjalizacjach lekarza
 - Id_lekarz – klucz obcy, identyfikator lekarza
 - Id_specjalizacja – klucz obcy, identyfikator specjalizacji
- Dyżur – zawiera informacje o dyżurze
 - Id_dyżur – klucz główny
 - Id_lekarz – klucz obcy, identyfikator lekarza, który posiada ten dyżur
 - Dzień – data dyżuru
 - Początek – godzina rozpoczęcia dyżuru
 - Koniec – godzina zakończenia dyżuru
 - Id_gabinet – klucz obcy, identyfikator gabinetu, w którym odbywa się dyżur
- Gabinet – informacje o gabinecie
 - Id_gabinet – klucz główny
 - Numer – numer gabinetu
 - Id_poradnia – klucz obcy, identyfikator poradni, w której znajduje się gabinet
- Pacjent – zawiera informacje o pacjencie
 - Id_pacjent – klucz główny
 - Imie – imię pacjenta

- Nazwisko – nazwisko pacjenta
- Data_urodzenia – data urodzenia pacjenta
- Email – email pacjenta
- Pesel – pesel pacjenta
- Id_adres – klucz obcy, identyfikator adresu zamieszkania pacjenta
- Login – login pacjenta do serwisu
- Haslo – hasło pacjenta do serwisu
- Dolegliwosc – zawiera informacje o dolegliwości
 - Id_dolegliwosc – klucz główny
 - Nazwa – nazwa dolegliwości
 - Opis – opis dolegliwości
- Pacjent_dolegliwosc – zawiera informacje o dolegliwościach pacjenta
 - Id_pacjent – klucz obcy, identyfikator pacjenta
 - Id_dolegliwosc – klucz obcy, identyfikator dolegliwości
- Wizyta – zawiera informację o wizycie w przychodni
 - Id_wizyta – klucz główny
 - Data – data wizyty
 - Godzina – godzina wizyty
 - Id_pacjent – klucz obcy, identyfikator pacjenta, który jest umówiony na wizytę
 - Id_dyzur – klucz obcy, identyfikator dyżuru lekarza, podczas którego odbędzie się wizyta
- E_recepta – zawiera informacje o e-receptce
 - Id_e_recepta – klucz główny
 - Id_pacjent – klucz obcy, identyfikator pacjenta, dla którego zapisana jest recepta
 - Id_lekarz – klucz obcy, identyfikator lekarza, który wypisuje receptę
- Lek – zawiera informacje o leku
 - Id_lek – klucz główny
 - Nazwa – nazwa leku
 - Substancja – substancja, którą zawiera lek
- E_recepta_lek – zawiera informacje o lekach na e_receptce
 - Id_e_recepta – klucz obcy, identyfikator e_recepty
 - Id_lek – klucz obcy, identyfikator leku
 - Dawkowanie – zalecane dawkowanie leku

The diagram illustrates the relationships between various entities in a medical system. The entities and their attributes are as follows:

- lekarz_specjalizacja**: id_lekarz, id_specjalizacja
- specjalizacja**: id_specjalizacja, nazwa, rok_otrzymania, uczelnia
- lekarz**: email, id_lekarz, haslo, imie, login, nazwisko, pesel, telefon
- lekarz_poradnia**: id_lekarz, id_poradnia
- poradnia_typ**: id_typ, nazwa, opis
- poradnia**: id_placowka, id_poradnia, id_typ, budynek, pietro
- placowka**: id_adres, id_placowka, nazwa
- adres**: id_adres, kod_pocztowy, miasto, numer, ulica
- dyzur**: id_dyzur, id_gabinet, id_lekarz, dzien, koniec, poczatek
- wizyta**: id_dyzur, id_pacjent, +3 hidden fields
- pacjent**: email, id_adres, id_pacjent, +6 hidden fields
- pacjent_dolegliwosc**: id_dolegliwosc, id_pacjent
- dolegliwosc**: id_dolegliwosc, nazwa, opis
- e_recepta**: id_e_recepta, id_lekarz, id_pacjent, kod
- e_recepta_lek**: id_e_recepta, id_lek, dawkowanie
- lek**: id_lek, nazwa, substancja
- gabinet**: id_gabinet, id_poradnia, numer

The relationships between the entities are indicated by arrows:

- lekarz_specjalizacja** points to **lekarz**.
- specjalizacja** points to **lekarz_specjalizacja**.
- lekarz** points to **lekarz_poradnia**.
- lekarz_poradnia** points to **poradnia**.
- poradnia_typ** points to **poradnia**.
- placowka** points to **poradnia**.
- adres** points to **placowka**.
- dyzur** points to **lekarz**.
- dyzur** points to **gabinet**.
- wizyta** points to **dyzur**.
- wizyta** points to **pacjent**.
- pacjent** points to **pacjent_dolegliwosc**.
- pacjent** points to **e_recepta**.
- pacjent_dolegliwosc** points to **dolegliwosc**.
- e_recepta** points to **e_recepta_lek**.
- e_recepta_lek** points to **lek**.
- gabinet** points to **lekarz_poradnia**.

→ Projektowanie tabel, kluczy, indeksów

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_adres	Int	PK NOT NULL	Klucz główny generowany automatycznie
Miasto	Varchar	NOT NULL	
Ulica	Varchar	NOT NULL	
Numer	Varchar	NOT NULL	
Kod_pocztowy	varchar	NOT NULL	

Poradnia_typ

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_typ	Int	PK NOT NULL	Klucz główny generowany automatycznie
Nazwa	Varchar	NOT NULL	
Opis	Varchar	NOT NULL	

Poradnia

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_poradnia	Int	PK NOT NULL	Klucz główny generowany automatycznie
Id_placowka	Int	FK NOT NULL	Klucz obcy do tabeli placowka
Budynek	Varchar(4)	NOT NULL	
Pietro	Int	NOT NULL	
Id_typ	int	FK NOT NULL	Klucz obcy do tabeli poradnia_typ

Lekarz

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_lekarz	Int	PK NOT NULL	Klucz główny generowany automatycznie
Imie	Varchar(30)	NOT NULL	
Nazwisko	Varchar(30)	NOT NULL	
Email	Varchar(40)	NOT NULL	
Telefon	Varchar	NOT NULL	
pesel	Varchar	NOT NULL	
Login	Varchar	NOT NULL	
Haslo	Varchar	NOT NULL	

Specjalizacja

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_specjalizacja	Int	PK NOT NULL	Klucz główny generowany automatycznie

Nazwa	Varchar	NOT NULL	
Uczelnia	Varchar	NOT NULL	
Rok_otrzymania	Int	NOT NULL	

Lekarz_specjalizacja

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_specjalizacja	Int	PFK NOT NULL	Klucz obcy do tabeli specjalizacja
Id_lekarz	INT	PFK NOT NULL	Klucz obcy do tabeli lekarz

Lekarz_poradnia

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_poradnia	Int	PFK NOT NULL	Klucz obcy do tabeli poradnia
Id_lekarz	INT	PFK NOT NULL	Klucz obcy do tabeli lekarz

Dyżur

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_dyżur	Int	PK NOT NULL	Klucz główny generowany automatycznie
Id_lekarz	int	NOT NULL	Klucz obcy do tabeli lekarz
Dzień	Date	NOT NULL	
Początek	Time	NOT NULL	
Koniec	Time	NOT NULL	
Id_gabinet	Int	FK NOT NULL	Klucz obcy do tabeli gabinet

Gabinet

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_gabinet	Int	PK NOT NULL	Klucz główny generowany automatycznie
Numer	Int	NOT NULL	
Id_poradnia	Int	FK NOT NULL	Klucz obcy do tabeli poradnia

Pacjent

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_pacjent	Int	PK NOT NULL	Klucz główny generowany automatycznie
imie	Varchar(30)	NOT NULL	
Uczelnia	Varchar(30)	NOT NULL	

Data_urodzenia	Date	NOT NULL	
Email	Varchar(40)	NOT NULL	
Pesel	Varchar	NOT NULL	
Id_adres	Int	FK NOT NULL	Klucz obcy do tabeli adres
Login	Varchar	NOT NULL	
Haslo	Varchar	NOT NULL	

Dolegliwosc

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_dolegliwosc	Int	PK NOT NULL	Klucz główny generowany automatycznie
Nazwa	Varchar	NOT NULL	
Opis	Varchar	NOT NULL	

Pacjent_dolegliwosc

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_pacjent	Int	PFK NOT NULL	Klucz główny do tabeli pacjent
Id_dolegliwosc	Int	PFK NOT NULL	Klucz główny do tabeli dolegliwosc

E_recepta

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_e_recepta	Int	PK NOT NULL	Klucz główny generowany automatycznie
Id_pacjent	Int	FK NOT NULL	Klucz obcy do tabeli pacjent
Id_lekarz	int	FK NOT NULL	Klucz obcy do tabeli lekarz

Lek

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_lek	Int	PK NOT NULL	Klucz główny generowany automatycznie
Nazwa	Varchar	NOT NULL	
Substancja	Varchar	NOT NULL	

E_recepta_lek

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_e_recepta	Int	PFK NOT NULL	Klucz obcy do tabeli e_recepta
Id_lek	Int	PFK NOT NULL	Klucz obcy do tabeli lek
Dawkowanie	Varchar	NOT NULL	

Wizyta

Nazwa atrybutu	Typ	Właściwość	Uwagi
Id_wizyta	Int	PK NOT NULL	Klucz główny generowany automatycznie
Godzina	Time	NOT NULL	
Data	Date	NOT NULL	
Id_pacjent	Int	NOT NULL	Klucz obcy do tabeli pacjent
Id_dyzur	Int	NOT NULL	Klucz obcy do tabeli dyzur

→ Zależności funkcyjnych i normalizacja tabel

Na etapie projektowania tabele zostały znormalizowane. Spełniają one wymagania dla postaci 1NF – dane są przechowywane w sposób atomowy w każdej tabeli, 2NF – każda tabela przechowuje dane dotyczące tylko konkretnej klasy obiektów i 3NF – kolumny nie należące do klucza nie są zależne od innych atrybutów niekluczowych.

→ Zaprojektowanie operacji na danych

Aby zrealizować założenia projektu wykorzystywane były widoki i funkcje.

Widoki:

- Lekarz_dyzury – wybiera informacje z tabel dotyczące dyżuru – dane lekarza, który pracuje w ten dyżur, placówki i poradni, w której odbywa się dyżur wykorzystywane w panelu pacjenta do wyboru wizyty w godzinach wolnych
- Lekarz_pacjenci – służy do wyświetlenia w panelu lekarza informacji o jego pacjentach
- Lekarz_poradnia_placowka – służy do wyświetlenia lekarzy w danej poradni w danej placówce
- Lekarz_recepty – służy do wyświetlenia informacji o receptach wypisanych przez danego lekarza na jego panelu
- Lekarz_wizyty – służy wyświetleniu informacji o wizytach danego lekarza na jego panelu
- Pacjent_lekarz_recepty – wyświetlanie informacji o przepisanych receptach pacjenta danego lekarza na jego panelu na profilu pacjenta
- Pacjent_wizyty- wyświetlania nadchodzących wizyt i informacji o nich na profilu pacjenta
- Wybor_wizyta – wyświetlenie szczegółowych informacji o wybranej przez pacjenta wizycie
- Wyszukaj_lekarz – używany do wyszukiwania lekarza po jego specjalizacji lub/i nazwisku w panelu pacjenta przy wyborze wizyty

Funkcję wspomagające działanie bazy:

- Dodaj_lekarz_poradnia – funkcja dodająca id_poradnia i id_lekarz do tabeli lekarz_poradnia
- Dodaj_pacjent_dolegliwosc – funkcja dodająca id_pacjent i id_dolegliwosc do tabeli pacjent_dolegliwosc
- Dostepnosc_dyzur – funkcja sprawdzająca, czy w danym czasie dyżuru jest już umówiona wizyta; wykorzystywana przy wyborze godziny wizyty przez pacjenta – wyświetlają się tylko wolne godziny do wyboru
- Lek_dodaj – funkcja sprawdza czy lek o danych parametrach znajduje się już w bazie – zwraca trigger
- Sprawdz_lekarz_poradnia – funkcja wykorzystywana przy przypisywaniu lekarza do konkretnej poradni – do wyboru nie są wyświetlani lekarze należący już do tej poradni
- Wyszwietl_lekarz_poradnia – wyświetlanie lekarzy w danej poradni
- Valid_dane – zwraca trigger, waliduje dane przy rejestracji pacjenta
- Usun_pacjent_dolegliwosc – zwraca trigger, usuwanie dolegliwości pacjenta

W bazie zostały także wykorzystane triggery:

- lek_nie_duplikuj – wykorzystywane przy dodawaniu leku do e-recepty, aby nie dodawać leku o tych samych danych
- pacjent_valid – walidacja formularza rejestracji pacjenta
- usun_dolegliwosc – usuwanie dolegliwości danego pacjenta
- usun_lekarz – usuwanie lekarza z pozostałych tabel, które są z nim powiązane

Kod sql znajduje się w dołączonym pliku baza.sql

IV. Projekt funkcjonalny

1. Interfejsy do prezentacji, edycji i obsługi danych

Na panelu startowym znajdują się odnośniki do trzech formularzy logowanie – dla lekarza, pacjenta, admina wymagające podania loginu i hasła. Dodatkowo przy formularzu logowania pacjenta znajduje się również odnośnik do formularza rejestracji pacjenta z danymi do wypełnienia w tabeli pacjent(imię, nazwisko, email, data urodzenia, pesel, login, hasło) oraz adres(miasto, ulica, numer, kod pocztowy).

W opcjach lekarza znajduje się:

- formularz do dodawania dolegliwości poszczególnym pacjentom - nazwa dolegliwości oraz jej opis przebiegu
- formularz wypełniania recept – nazwa leku, substancja oraz dawkowanie

W panelu pacjenta znajduje się:

- formularz edycji swoich danych osobowych(imię, nazwisko, email, pesel, data urodzenia, miasto, ulica, numer, kod pocztowy)
- formularz wyboru godziny wizyty
- formularz do wyszukiwania lekarza – nazwisko i/lub specjalizacja
W panelu admina znajduje się:
- formularz dodawania/edycji placówek – nazwa, adres
- formularz dodawania/edycji typów poradni – nazwa, opis
- formularz dodawania/edycji poradni – nazwa, typ budynek, piętro
- formularz dodawania/edycji lekarzy – imię, nazwisko, pesel, telefon, email
- formularz dodawania dyżurów – lekarz, dzień, początek, koniec, gabinet
- formularz dodawania/edycji gabinetów – poradnia, numer
- formularz edytowania pacjentów -imię, nazwisko, email, pesel, data urodzenia, misto, ulica, numer, kod pocztowy

2. Wizualizacja danych

Projekt napisany jest w języku PHP z wykorzystaniem HTML i JS. Wszelkie zbiorowe dane są wyciągane z bazy i przedstawiane jako tabelka z wykorzystaniem Bootstrapa. Przykładowe raporty to – spis placówek, poradni, gabinetów, lekarzy i ich dyżurów, specjalizacji, lista dostępnych wizyt, lista pacjentów danego lekarza, lista recept itd.

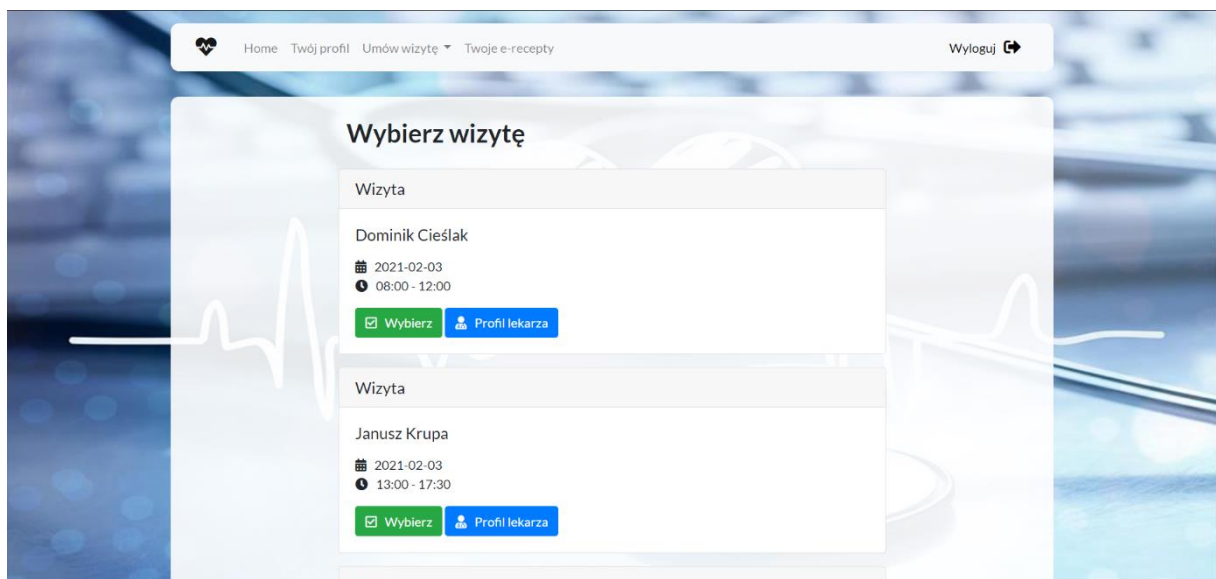
Ilość lekarzy posiadających dyżury: 33

Średnia ilość dyżurów lekarza: 5

Lekarze mający ponad 30 godzin dyżuru:

ID	Imię	Nazwisko	Suma godzin	Profil
21	Ludwik	Maciejewski	41	Profil
2	Janusz	Krupa	39	Profil
4	Anna	Lis	39	Profil
13	Lukasz	Pawlak	39	Profil
10	Izabela	Gajewska	36	Profil
9	Mikołaj	Jankowski	35	Profil
28	Weronika	Kalinowska	35	Profil
29	Marcin	Kołodziej	32	Profil
7	Barbara	Przybylska	31	Profil

Rysunek 2: Przykładowy raport



Rysunek 3: Przykładowe zaprezentowanie danych

The screenshot displays a web application with a sidebar menu on the left and a main content area. The sidebar menu includes 'Menu', 'Placówki', 'Lekarze', 'Dyżury', 'Pacjenci', and 'Wyloguj'. The main content area is titled 'Wszyscy zatrudnieni lekarze' and contains a table with 11 rows of doctor data. Each row includes an ID, first name, last name, email, and an 'Action' column with 'Profil' and 'Usuń' buttons.

ID	Imię	Nazwisko	Email	Action
1	Dominik	Cieślak	dominik.cieslak@mail.com	Profil Usuń
2	Janusz	Krupa	krupaj@gmail.com	Profil Usuń
3	Bartosz	Kalinowski	barkai@mail.com	Profil Usuń
4	Anna	Lis	lis.anna@mail.com	Profil Usuń
5	Wanda	Laskowska	laskowskaw@mail.com	Profil Usuń
6	Józef	Kamiński	jozef.kam@mail.com	Profil Usuń
7	Barbara	Przybylska	barbara.przybylska@mail.com	Profil Usuń
8	Aleksy	Szymański	aleks.szymanski@mail.com	Profil Usuń
9	Mikołaj	Jankowski	milosc.jan@mail.com	Profil Usuń
10	Izabela	Gajewska	izagaj@mail.com	Profil Usuń
11	Dagmara	Jaworska	dag.jaw@mail.com	Profil Usuń

Rysunek 4: Raport wszystkich lekarzy

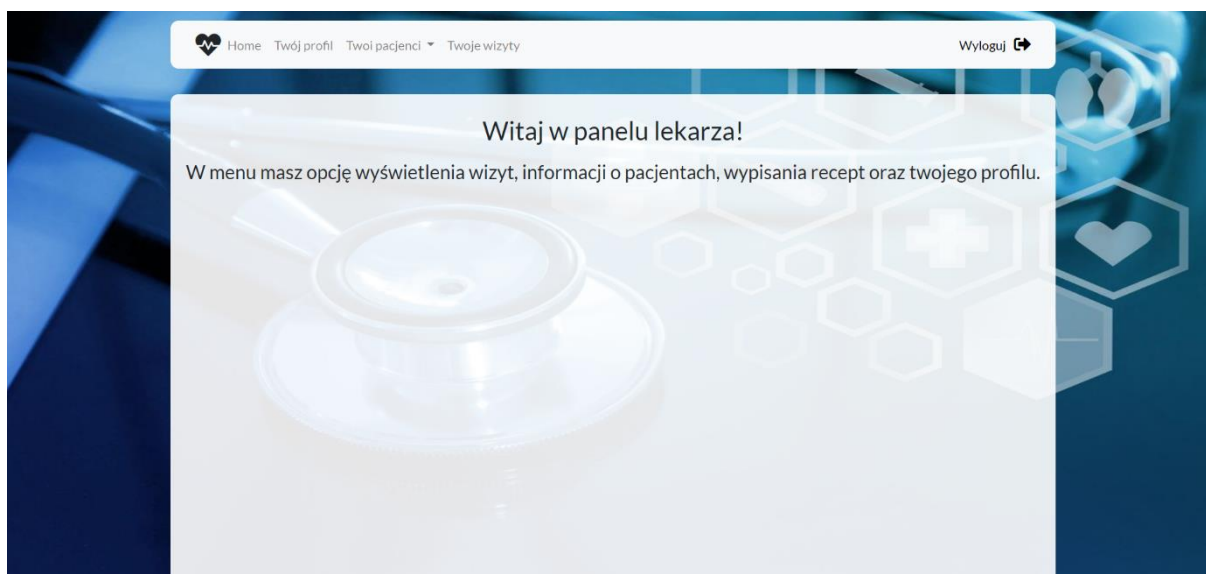
3. Zdefiniowanie panelu sterowania aplikacji

Użytkownik po wejściu na stronę startową ma do wyboru trzy opcje zalogowania:

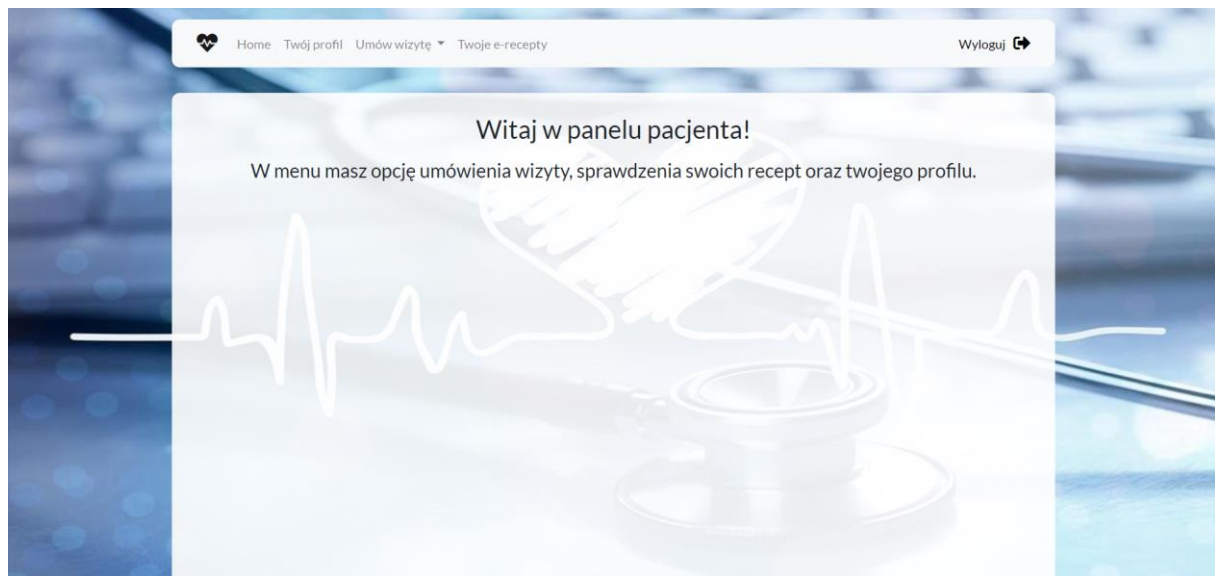


Rysunek 3: Panel startowy

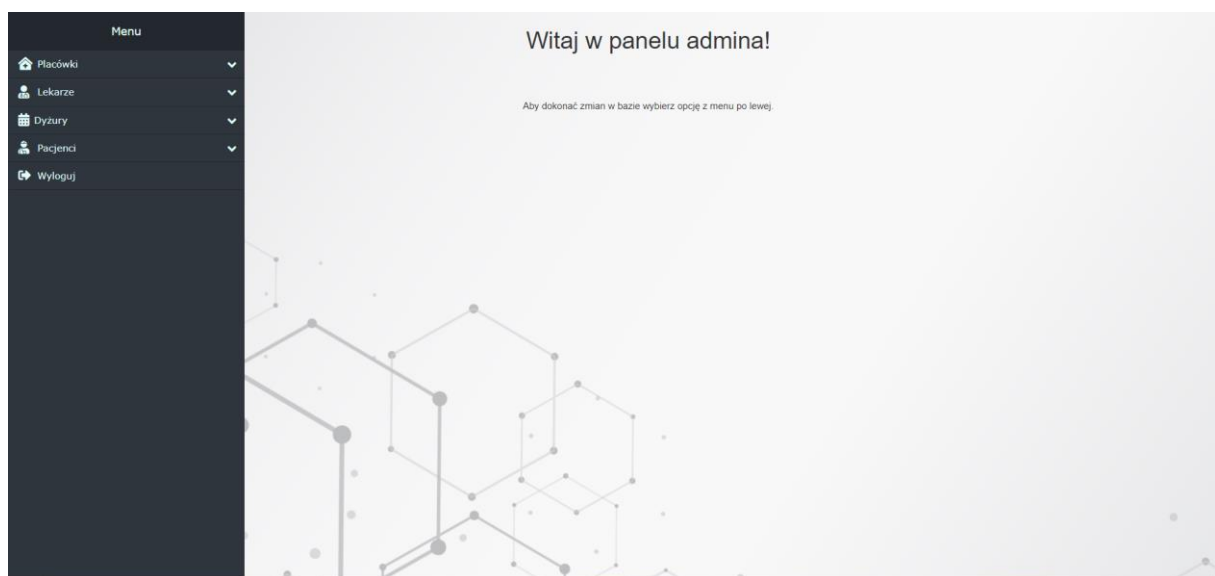
W zależności od typu konta na jakie się zalogujemy przenosimy się na panel z menu z odpowiednimi dla naszej roli funkcjonalnościami.



Rysunek 4: Panel lekarza



Rysunek 5: Panel pacjenta



Rysunek 6: Panel admina

V. Dokumentacja

1. Wprowadzanie danych

Dane są wprowadzane ręcznie, jedynie przy tabelach asocjacyjnych dane są wprowadzane automatycznie bez ingerencji użytkownika

Baza została wstępnie uzupełniona przykładowymi danymi z pliku inserts.sql.

2. Dokumentacja użytkownika

Użytkownik loguje się na odpowiedni typ konta wybierając formularz z panelu startowego. Następnie w menu ma dostępne opcje:

- Panel lekarza
 - Wyświetlenie swojego profilu z danymi osobowymi, dyżurami, specjalizacjami
 - Wyświetlenie swoich pacjentów (czyli tacy którzy mieli/będą mieć u niego wizytę) i wypisanie dla nich e-recept lub dodanie dolegliwości.
 - Wyświetlenia e-recept wypisanych przez siebie
 - Wyświetlenie wizyt i ich szczegółów wraz z odnośnikiem do profilu pacjenta
- Panel pacjenta
 - Wyświetlenie swojego profilu z danymi osobowymi, dolegliwościami z możliwością edycji danych osobowych
 - Umówienie wizyty wybierając placówkę, a następnie poradnię i wybór dyżuru o czasie nam odpowiadającym, po wybraniu dyżuru pojawia się opcja wyboru godziny – wyświetlają się tylko te godziny z dyżuru, podczas których nie odbywa się inna wizyta lekarza
 - Wyszukiwanie lekarza po jego nazwisku i/lub specjalizacji, wybór konkretnego dyżuru lub możliwość obejrzenia jego profilu wraz z jego dyżurami – możliwość umówienia wizyty również z profilu lekarza
 - Wyświetlenie swoich e-recept wraz z wygenerowanym automatycznie kodem do odbioru
- Panel admina
 - W zakładce placówki możliwość dodania placówki, wyświetlenia/ edycji placówek, dodawanie/edycja typów poradni, dodawanie/wyświetlenie/edycja poradni w danej placówce
 - W zakładce lekarz możliwość wyświetlenia/edycji wszystkich lekarzy lub lekarzy w wybranej poradni w danej placówce i możliwość przypisania wcześniej dodanego do bazy lekarza do danej poradni, dodanie lekarza do bazy
 - W zakładce dyżury dodawanie/edytowanie gabinetów w danej poradni, dodawanie dyżuru po wcześniejszym wyborze lekarza z poradni, wyświetlenie statystyk o dyżurach lekarzy
 - W zakładce pacjent wyświetlanie/edytowanie pacjentów, wyświetlenie wszystkich leków, które były przepisywane pacjentom

Najwięcej danych znajduje się w placówce „Medico 1” Kraków, najlepiej testować aplikację wybierając tę placówkę.

Aplikacja jest dostępna na Pascalu pod adresem pascal.fis.agh.edu.pl/~8pyrek/bd_project/.

Baza danych znajduje się na ElephantSQL, gdzie łączenie do niej jest w pliku connect.php. Można odtworzyć bazę danych na podstawie dołączonych plików sql, jednak najprościej podłączyć się do gotowej bazy na ElephantSQL.

Przykładowe dane do logowania:

- Lekarz - login : dcieslak hasło: password1
- Pacjent - login: Tadeusz hasło: testpassword
- Admin – login: admin hasło: password123

3. Opracowanie dokumentacji technicznej

Aplikacja została napisana w języku PHP z wykorzystaniem elementów HTML, CSS, JS i biblioteki Bootstrap. Łączenie się z bazą i wyciąganie z niej danych jest realizowane za pomocą funkcji dostępnych w PHP. Łączenie z bazą realizuje się przez funkcję **`pg_connect (string $connection_string , int $connect_type = ?)`**. Zarządzanie danymi realizowane jest głównie funkcją **`pg_query (resource $connection = ? , string $query)`**, która realizuje podane argumenty \$query zapytanie do bazy. Następnie wykorzystuję funkcję **`pg_fetch_assoc (resource $result , int $row = ?)`**, która umożliwia dostęp do każdego pola w rekordzie np. \$row['id'] zwraca pole id w rekordzie.

Aby uruchomić aplikację lokalnie należy

- zainstalować PHP `sudo apt install php libapache2-mod-php`
- następnie `sudo apt install php-pgsql`
- uruchomić server PHP będąc w katalogu głównym aplikacji – tam gdzie index.php `php -S localhost:8000`
- w przeglądarce wpisać adres localhost:8000

3. Wykaz literatury

<https://www.php.net/manual/en/book.pgsql.php>

trevor.io – użyte do zbudowania diagramu ERD