



Akademia Górniczo-Hutnicza w Krakowie

Wydział Fizyki i Informatyki Stosowanej

Bartosz Rogowski, III rok, 303793

24 stycznia 2021

Bazy Danych

Dokumentacja projektu

1. Projekt koncepcji, założenia

1.1. Temat projektu

Projekt skupia się na stworzeniu bazy danych osób powiązanych z podejrzeniem lub wykryciem wirusa Covid-19 wraz z ewidencją ich choroby, stanu zdrowia, pobytu oraz innych niezbędnych informacji, które pomagają w przygotowywaniu końcowych raportów epidemicznych.

Cele i zadania:

- zautomatyzowanie wpisywania osób z pozytywnym wynikiem na listę zakażonych, a także późniejszego wpisu na listę osób ozdrowiałych lub zmarłych
- wprowadzenie uporządkowania i lepszej kontroli w systemie ewidencji osób zakażonych, np. ich pobytu i stanu zdrowia
- ułatwienie dostępu części danych osoby dla osób kontaktowych (np. członków rodziny, najbliższych – zdefiniowanych przez użytkownika).

1.2. Analiza wymagań użytkownika

Z uwagi na różne kategorie użytkowników, mają oni różne wymagania:

➤ Osoba

Użytkownik powinien móc przeglądać swoje podstawowe dane, stan, miejsce pobytu oraz wskazywać osobę kontaktową, czyli taką, którą uwierzytelnia do przeglądania tych informacji o sobie. Dodatkowo może złożyć prośbę o wykonanie testu w laboratorium.

➤ Lekarz

Użytkownik powinien mieć dostęp do podstawowych danych pacjentów znajdujących się w danym szpitalu. W bazie pełni on rolę zarządzania stanem pacjenta, kierowaniem na izolację lub na badania.

➤ Pracownik laboratorium

Użytkownik powinien mieć dostęp do zleceń na testy oraz ma możliwość tworzenia testów (ich wyników). Jeśli wynik jest pozytywny, to automatycznie dane osoby z takim wynikiem zostaną zapisane do odpowiedniej tabeli, aby usprawnić pracę pracownika laboratorium.

➤ Pracownik wojewódzkiej stacji sanepidu

Użytkownik powinien mieć dostęp do baz, w której znajdują się dane osób zakażonych, ozdrowiałych, zmarłych lub podejrzanych (stan „*kwarantanna*”). Na ich podstawie może tworzyć raporty oraz kierować i nadzorować izolacje.

1.3. Określenie podstawowych funkcji realizowanych w bazie danych

- Wprowadzanie osób do systemu za pomocą prostego formularza
- Logowanie do systemu z podziałem na odpowiednią rolę (osoba, pracownik laboratorium, lekarz, osoba pracująca w wojewódzkim oddziale sanepidu)
- Umożliwienie przeglądania podstawowych informacji o danej osobie (także przez osobę do tego upoważnioną, jeśli została przypisana)
- Stworzenie mechanizmu automatycznej kwalifikacji pacjentów do odpowiednich tabel/kategorii (zakażony, ozdrowiały, zmarły) w zależności od wyniku testu
- Wprowadzanie danych testu przez pracowników laboratoriów
- Zarządzanie stanem, pobytem pacjenta np. przez lekarzy, a także kierowaniem na badania

2. Konceptualny projekt diagramów

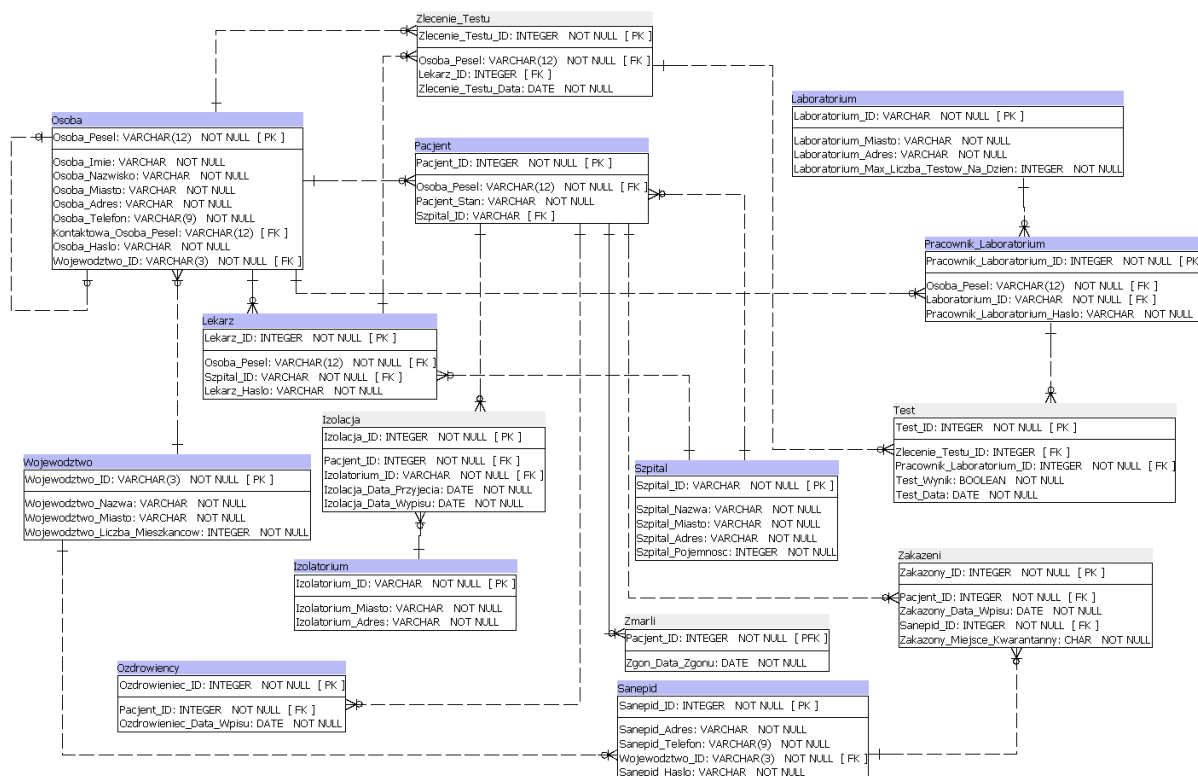
2.1. Zdefiniowanie encji oraz ich atrybutów

- **Osoba** – encja opisująca osobę i jej wszystkie podstawowe informacje: unikalny numer pesel, imię, nazwisko, adres, numer telefonu, województwo, w którym jest zameldowana, hasło do serwisu oraz pesel jednej osoby, którą upoważnia do informowania o swoim stanie (jeśli zostanie podana).

- **Pacjent** – encja opisująca każdą osobę, której stan uległ zmianie (tzn. taka, która kontaktowała się z lekarzem i jej stan jest inny niż „zdrowy”, zawiera unikalne ID pacjenta (może dojść do sytuacji, w której osoba ozdrowiała ponownie zachoruje lub gdy lekarz stanie się pacjentem), numer pesel, stan (możliwe są następujące opcje: „chory”, „ozdrowiały”, „nie żyje”, „kwarantanna”) oraz ID szpitala (może nie być konieczny pobyt w szpitalu, np. w sytuacji domowej kwarantanny lub izolacji, gdy stan pacjenta nie wymusza hospitalizacji, stąd nie każdy pacjent musi być przypisany do konkretnej jednostki).
- **Lekarz** – encja opisująca lekarza (dla uproszczenia bazy nie wyróżnia się podziału na lekarza pierwszego kontaktu i pracownika szpitala), zawiera unikalne ID lekarza, numer pesel, ID szpitala, do którego jest przypisany oraz hasło do serwisu.
- **Pracownik_Laboratorium** – encja opisująca pracownika laboratorium, zawiera unikalne ID, numer pesel, ID laboratorium , do którego jest przypisany oraz hasło do serwisu.
- **Szpital** – encja opisująca podstawowe informacje o szpitalu – unikalne ID, nazwę, adres oraz pojemność (wielkość opisująca maksymalną liczbę pacjentów, którą może przyjąć).
- **Laboratorium** – encja opisująca podstawowe informacje o jednostce laboratoryjnej – unikalne ID, adres oraz maksymalną liczbę testów, którą można przeprowadzić każdego dnia.
- **Izolatorium** – encja opisująca dane izolatorium – unikalne ID, miasto oraz adres.
- **Województwo** – encja opisująca podstawowe informacje o danym województwie (potrzebna w celu ułatwienia przypisania osób do odpowiedniej jednostki sanepidu oraz do wykonania raportów) – zawiera unikalne ID, nazwę, miasto wojewódzkie oraz (przybliżoną) liczbę mieszkańców.

- **Sanepid** – encja opisująca wojewódzki punkt stacji sanitarno-epidemiologicznej – zawiera unikalne ID, adres siedziby, numer telefonu, ID województwa, w którym się znajduje oraz hasło potrzebne do zalogowania się do systemu.
- **Ozdrowiency** – encja zawierająca dane osób, które zostały uznane za uzdrowieńców (np. po minięciu 10 dni od wpisania na listę zakażonych, jeżeli w tym czasie pacjent nie umrze).
- ❖ **Izolacja** – encja asocjacyjna zawierająca informacje o umieszczeniu osób w danym izolatorium (UWAGA: nie są w niej zapisywane dane osób, które przechodzą izolację w swoich adresach zamieszkania lub szpitalach), zawiera unikalne ID, ID pacjenta, datę przyjęcia oraz datę wypisu.
- ❖ **Zlecenie_Testu** – encja asocjacyjna opisująca zlecenie testu przez lekarza lub osobiście przez daną osobę; zawiera unikalne ID zlecenia, pesel osoby, która ma być badana pod kątem obecności wirusa, ID lekarza, który zalecił test (nieobowiązkowe pole, jeśli osoba samodzielnie zapisuje się na test) oraz data zlecenia testu.
- ❖ **Test** – encja asocjacyjna opisująca test; zawiera unikalne ID testu, ID zlecenia testu (w którym zawarte są informacje na temat osoby badanej oraz lekarza zlecającego), ID pracownika laboratorium przeprowadzającego i stwierdzającego wynik testu, wynik testu (typu logicznego: *true*, jeśli wynik jest pozytywny na obecność wirusa) oraz datę wpisania wyniku do bazy.
- ❖ **Zakazeni** – encja asocjacyjna, w której zawarte są niezbędne dla sanepidu informacje o osobach zakażonych; zawiera unikalny numer ID zakażonego, ID pacjenta, datę wpisu na listę, ID sanepidu oraz miejsce kwarantanny (możliwe są następujące opcje: „d” – oznacza miejsce zamieszkania).
- ❖ **Zmarli** – encja słownikowa, w której znajdują się informacje o dacie zgonu osoby o podanym peselu (UWAGA: służy głównie celom statycznym).

2.2. Relacje pomiędzy encjami



Rysunek 1. Diagram ERD projektu wykonany w programie SQL Power Architect.

3. Projekt logiczny

3.1. Projektowanie tabel, kluczy, indeksów

Struktura zgodna z opisem i diagramem ERD z wcześniejszych podpunktów została wygenerowana i zapisana do skryptu *create_covid.sql* z pomocą programu SQL Power Architect. Wszelkie modyfikacje (np. uwzględnienie własnych domen) wraz z zestawem funkcji oraz wyzwalaczy zostały zapisane w skrypcie *func_covid.sql*.

Klucze obce zostały dodane za pomocą mechanizmu ADD CONSTRAINT oraz REFERENCES. Więcej o szczegółach technicznych tabel można znaleźć w podpunkcie 3.2. *Słowniki danych*.

3.2. Słowniki danych

Nazwa tabeli			
nazwa kolumny	typ	informacje	uwagi/objaśnienia
Wojewodztwo			
Wojewodztwo_ID	varchar(3)	PRIMARY KEY	Identyfikator województwa
Wojewodztwo_Nazwa	varchar	NOT NULL	
Wojewodztwo_Miasto	varchar	NOT NULL	Stolica województwa
Wojewodztwo_Liczba_Mieszkancow	integer	NOT NULL	

Laboratorium			
Laboratorium_ID	varchar	PRIMARY KEY	Identyfikator laboratorium
Laboratorium_Miasto	varchar	NOT NULL	
Laboratorium_Adres	varchar	NOT NULL	

Sanepid			
Sanepid_ID	INTEGER	PRIMARY KEY	Klucz główny generowany automatycznie
Sanepid_Adres	varchar	NOT NULL	
Sanepid_Telefon	varchar	NOT NULL	
Wojewodztwo_ID	varchar(3)	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Wojewodztwo</i>
Sanepid_Haslo	varchar	NOT NULL	

Izolatorium			
Izolatorium_ID	varchar	PRIMARY KEY	Identyfikator izolatorium
Izolatorium_Miasto	varchar	NOT NULL	
Izolatorium_Adres	varchar	NOT NULL	

Szpital			
Szpital_ID	varchar	PRIMARY KEY	Identyfikator szpitala
Szpital_Nazwa	varchar	NOT NULL	
Szpital_Miasto	varchar	NOT NULL	
Szpital_Adres	varchar	NOT NULL	
Szpital_Pojemnosc	integer	NOT NULL	

Osoba			
Osoba_Pesel	varchar(12)	PRIMARY KEY	Identyfikator osoby – pesel
Osoba_Imie	varchar	NOT NULL	
osoba_Nazwisko	varchar	NOT NULL	
Osoba_Miasto	varchar	NOT NULL	
Osoba_Adres	varchar	NOT NULL	
Osoba_Telefon	telefon	NOT NULL	Domena – tekst składający się z 9 cyfr
Kontaktowa_Osoba_Pesel	varchar(12)	FOREIGN KEY	Klucz obcy do tabeli <i>Osoba</i> (może być NULL)
Osoba_Haslo	varchar	NOT NULL	
Wojewodztwo_ID	varchar(3)	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Wojewodztwo</i>

Pracownik_Laboratorium			
Pracownik_Laboratorium_ID	INTEGER	PRIMARY KEY	Klucz główny generowany automatycznie
Osoba_Pesel	varchar(12)	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Osoba</i>

Laboratorium_ID	varchar	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Laboratorium</i>
Pracownik_Laboratorium_Haslo	varchar	NOT NULL	

Lekarz			
Lekarz_ID	INTEGER	PRIMARY KEY	Klucz główny generowany automatycznie
Osoba_Pesel	varchar(12)	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Osoba</i>
Szpital_ID	varchar	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Szpital</i>
Lekarz_Haslo	varchar	NOT NULL	

Zlecenie_Testu			
Zlecenie_Testu_ID	INTEGER	PRIMARY KEY	Klucz główny generowany automatycznie
Osoba_Pesel	varchar(12)	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Osoba</i>
Lekarz_ID	INTEGER	FOREIGN KEY	Klucz obcy do tabeli <i>Lekarz</i> (może być NULL)
Zlecenie_Testu_Data	date	NOT NULL	

Test			
Test_ID	INTEGER	PRIMARY KEY	Klucz główny generowany automatycznie
Zlecenie_Testu_ID	INTEGER	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Zlecenie_Testu</i>
Pracownik_Laboratorium_ID	INTEGER	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Pracownik_Laboratorium</i>
Test_Wynik	boolean	NOT NULL	
Test_Data	date	NOT NULL	

Pacjent			
Pacjent_ID	INTEGER	PRIMARY KEY	Klucz główny generowany automatycznie
Osoba_Pesel	varchar(12)	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Osoba</i>
Pacjent_Stan	varchar	NOT NULL	
Szpital_ID	varchar	FOREIGN KEY	Klucz obcy do tabeli <i>Szpital</i> (może być NULL)

Zmarli			
Pacjent_ID	integer	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Pacjent</i>
Zgon_Data_Zgonu	date	NOT NULL	

Ozdrowienci			
Ozdrowieniec_ID	INTEGER	PRIMARY KEY	Klucz główny generowany automatycznie
Pacjent_ID	integer	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Pacjent</i>
Ozdrowieniec_Data_Wpisu	date	NOT NULL	

Zakazeni			
Zakazony_ID	INTEGER	PRIMARY KEY	Klucz główny generowany automatycznie
Pacjent_ID	integer	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Pacjent</i>
Zakazony_Data_Wpisu	date	NOT NULL	
Sanepid_ID	integer	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Sanepid</i>
Zakazony_Miejsce_Kwarantanny	char	NOT NULL	

Izolacja			
Izolacja_ID	INTEGER	PRIMARY KEY	Klucz główny generowany automatycznie
Pacjent_ID	integer	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Pacjent</i>
Izolatorium_ID	varchar	FOREIGN KEY NOT NULL	Klucz obcy do tabeli <i>Izolatorium</i>
Izolacja_Data_Przyjecia	date	NOT NULL	
Izolacja_Data_Wypisu	date	NOT NULL	

Tabela 1. Objasnienie atrybutów w poszczególnych encjach.

3.3. Analiza zależności funkcyjnych i normalizacja tabel (dekompozycja do 3NF)

Nazwa encji	Analiza
Wojewodztwo	F: {Wojewodztwo_ID} → {Wojewodztwo_Nazwa, Wojewodztwo_Miasto, Wojewodztwo_Liczba_Mieszkancow} Relacja ta spełnia I, II oraz III postać normalną.
Laboratorium	F: {Laboratorium_ID} → {Laboratorium_Miasto, Laboratorium_Adres} ale laboratorium_Adres nie jest atomowa Gdyby kolumna z adresem została podzielona na dwie – ulica, numer mieszkania, to relacja ta spełniałaby I, II oraz III postać normalną.
Sanepid	F: {Sanepid_ID} → {Sanepid_Adres, Sanepid_Telefon, Wojewodztwo_ID, Sanepid_Haslo} ale Sanepid_Adres nie jest atomowa

	Gdyby kolumna z adresem została podzielona na dwie – ulica, numer mieszkania, to relacja ta spełniałaby I, II oraz III postać normalną.
Izolatorium	<p>F: {Izolatorium_ID} → {Izolatorium_Miasto, Izolatorium_Adres} ale Izolatorium_Adres nie jest atomowa</p> <p>Gdyby kolumna z adresem została podzielona na dwie – ulica, numer mieszkania, to relacja ta spełniałaby I, II oraz III postać normalną.</p>
Szpital	<p>F: {Szpital_ID} → {Szpital_Nazwa, Szpital_Miasto, Szpital_Adres, Szpital_Pojemnosc} ale {Szpital_Miasto} → {Szpital_Nazwa} oraz Szpital_Adres nie jest atomowa</p> <p>Gdyby kolumna z adresem została podzielona na dwie – ulica, numer mieszkania, to relacja ta spełniałaby I oraz II postać normalną.</p>
Osoba	<p>F: {Osoba_Pesel} → {Osoba_Imie, Osoba_Nazwisko, Osoba_Miasto, Osoba_Adres, Osoba_Telefon, Kontaktowa_Osoba_Pesel, Osoba_Haslo, Wojewodztwo_ID} ale Osoba_Adres nie jest atomowa</p> <p>Gdyby kolumna z adresem została podzielona na dwie – ulica, numer mieszkania, to relacja ta spełniałaby I, II oraz III postać normalną.</p> <p>(Uwaga: Może się wydawać, że nie jest spełniona III postać normalna, ponieważ uzupełniane hasło zależy od imienia i nazwiska, jednak zostało to wprowadzone jedynie na potrzeby prostego generowania dużej liczby danych przykładowych. Hasło można zmienić w serwisie na dowolne, które nie zależy od żadnych kolumn.)</p>
Pracownik_Laboratorium	<p>F: {Pracownik_Laboratorium_ID} → {Osoba_Pesel, Laboratorium_ID, Pracownik_Laboratorium_Haslo}</p> <p>Relacja ta spełnia I, II oraz III postać normalną.</p>
Lekarz	<p>F: {Lekarz_ID} → {Osoba_Pesel, Szpital_ID, Lekarz_Haslo}</p> <p>Relacja ta spełnia I, II oraz III postać normalną.</p>
Zlecenie_Testu	<p>F: {Zlecenie_Testu_ID} → {Osoba_Pesel, Lekarz_ID, Zlecenie_Testu_Data}</p> <p>Relacja ta spełnia I, II oraz III postać normalną.</p>
Test	F: {Test_ID} → {Zlecenie_Testu_ID, Pracownik_Laboratorium_ID, Test_Wynik, Test_Data}

	Relacja ta spełnia I, II oraz III postać normalną.
Pacjent	F: {Pacjent_ID} → {Osoba_Pesel, Pacjent_Stan, Szpital_ID} Relacja ta spełnia I, II oraz III postać normalną.
Ozdrowienci	F: {Ozdrowieniec_ID} → {Pacjent_ID, Ozdrowieniec_Data_Wpisu} Relacja ta spełnia I, II oraz III postać normalną.
Zakazeni	F: {Zakazeni_ID} → {Pacjent_ID, Zakazony_Data_Wpisu, Sanepid_ID, Zakazony_Miejsce_Kwarantanny} Relacja ta spełnia I, II oraz III postać normalną.
Izolacja	F: {Izolacja_ID} → {Pacjent_ID, Izolatorium_ID, Izolacja_Data_Przyjścia, Izolacja_Data_Wypisu} W obecnej formie relacja ta spełnia I oraz II postać normalną, jednak, aby spełniała III, należałoby zmienić sposób ustawiania daty wypisu izolacji tak, aby nie zależała od daty przyjęcia (obecnie jest to 10 dni od daty przyjęcia).

Tabela 2. Analiza postaci normalnych poszczególnych relacji w projekcie.

3.4. Zaprojektowanie operacji na danych

W celu przeprowadzania operacji na danych, zastosowano szereg funkcji oraz wyzwalaczy, które usprawniają i zabezpieczają działanie bazy. Ich definicja znajduje się w pliku *func_covid.sql*. Są one opisane w poniższej tabeli.

Nazwa	Typ	Opis
wpisz_na_liste_zakazonych	funkcja	Funkcja ta zwraca wyzwalacz. Jeśli wynik testu jest pozytywny, to następuje sprawdzenie, czy osoba, która go uzyskała znajduje się już w tabeli <i>Pacjent</i> , jeśli nie, to tworzy nowego pacjenta na bazie informacji o osobie testowanej, jeśli się znajduje, to wyszukiwany jest jego numer ID oraz następuje zmiana stanu na „chory”. Następnie szukany jest numer ID sanepidu, do którego należy dana osoba oraz zostaje ona wpisana do tabeli <i>Zakażeni</i> . Jeśli wynik testu jest negatywny, to funkcja nie modyfikuje żadnych elementów.
test_trigger	wyzwalacz	Jest wywoływany po wstawieniu danych do tabeli <i>Test</i> i wykonywana jest funkcja <i>wpisz_na_liste_zakazonych</i> (odpowiada za

		wypełnienie innych tabeli, jeśli wynik testu jest pozytywny).
wpisz_na_liste_ozdrowialych_lub_zmarlych	funkcja	Zwraca wyzwalacz. Jeśli nowy stan pacjenta to „ozdrowiały”, jego dane zapisywane są do tabeli <i>Ozdrowienci</i> , jeśli „nie żyje” – do tabeli <i>Zmarli</i> .
pacjent_stan_trigger	wyzwalacz	Jest wywoływany po zmodyfikowaniu danych w tabeli <i>Pacjent</i> i wykonywana jest funkcja <i>wpisz_na_liste_ozdrowialych_lub_zmarlych</i> .
edytuj_pesel_kontaktowej	funkcja	Wymaga dwóch argumentów varchar (pesel osoby oraz pesel osoby kontaktowej), zwraca informację tekstową. Sprawdzane jest, czy pesel osoby kontaktowej istnieje w bazie. Jeśli tak, to zmienia pole <i>Kontaktowa_Osoba_Pesel</i> z tabeli <i>Osoba</i> na drugi argument u osoby z polem <i>Osoba_Pesel</i> zgodnej z pierwszym argumentem funkcji.
dodaj_izolacje	funkcja	Wymaga trzech argumentów: varchar, varchar, date – pesel osoby, ID izolatorium oraz datę wpisu. Jeśli osoba o zadanym peselu nie znajduje się w tabeli <i>Pacjent</i> , to jest tam dodawana ze stanem „kwarantanna”. Wyszukiwany jest numer ID pacjenta o zadanym peselu, a następnie na jego podstawie wypełnia się tabelę <i>Izolacja</i> podając ID pacjenta, ID izolatorium, datę wpisu (data wypisu uzupełniana jest automatycznie – 10 dni od daty wpisu). Zwraca prawdę, jeśli operacja została wykonana prawidłowo.

Tabela 3. Opis funkcji oraz wyzwalaczy zawartych w bazie.

4. Projekt funkcjonalny

4.1. Interfejsy do prezentacji, edycji i obsługi danych

➤ Prezentacja danych

Prezentacja danych skupia się na wyświetleniu ich w formie tabeli (np. dane osobowe osoby).

Twoje dane osobowe:

Pesel	Imie	Nazwisko	Miasto	Adres	Telefon	Województwo	Stan	Szpital	Izolatorium - miasto	Izolatorium - adres
890725468720	Kacper	Górny	Gdynia	Lapisowa 22	839343978	pomorskie	-	-	-	-

Dane osób, które wyraziły zgodę na udzielanie informacji Tobie:

Pesel	Imie	Nazwisko	Miasto	Adres	Telefon	Województwo	Stan	Szpital	Izolatorium - miasto	Izolatorium - adres
920428864337	Stefan	Cabacki	Białystok	Żwirowa 69	813299491	podlaskie	-	-	-	-
690316543562	Karolina	Babacka	Poznań	Gajowa 113	819044727	wielkopolskie	-	-	-	-

Rysunek 2. Fragment strony, na której użytkownik zalogowany jako Osoba wyświetla dane swoje oraz osób, które zezwoliły mu na wgląd do ich danych.

Pozostała prezentacja skupia się na raportach, o których więcej w podpunkcie 4.2.

➤ Edycja danych

Głównym elementem, który podlega zmianie jest hasło, które każdy typ użytkownika może zmienić po zalogowaniu się do serwisu. Ponadto osoba może dokonać zmiany osoby kontaktowej (dodanie bądź edycja).

Zmień swoje dane kontaktowe

Wybierz typ danych, które chcesz zmienić, a następnie przypisz nową wartość w polu poniżej.

☒ Zmień swój numer telefonu
☐ Dodaj lub zmień osobę kontaktową*

Nowe dane:

Dalej

* - Osoba kontaktowa, to taka, która ma dostęp do wglądu podstawowych informacji o Tobie.

Rysunek 3. Fragment strony, na której użytkownik zalogowany jako Osoba wyświetla formularz do zmiany danych kontaktowych.

Kolejnym elementem jest zmiana stanu pacjenta przez lekarza, wówczas edytowana jest kolumna odpowiadająca za tę informację u odpowiedniego pacjenta.

Rysunek 4. Fragment strony, na której użytkownik zalogowany jako Lekarz wyświetla formularz do zmiany stanu pacjenta.

➤ Obsługa danych

Punkt ten odzwierciedlony jest w formularzach do logowania (każdy typ konta), w których użytkownik musi wypełnić odpowiednio login (ciąg cyfr o zadanej długości) oraz hasła (odpowiedni długi ciąg dowolnych znaków); oraz w formularzach takich jak funkcjonalność wyszukiwania użytkownika (Sanepid), nakładanie izolacji (Sanepid), czy kwarantanny (Lekarz), a także wpisywania pacjenta na odpowiednią listę (zakażeni, ozdrowiali, zmarli). Baza została napisana tak, aby użytkownik miał do wypełnienia jak najmniej danych, a system funkcji i wyzwalaczy automatycznie wypełniał pozostałe pola.

4.2. Wizualizacja danych – raporty

Raporty są wyświetlane głównie przez Sanepid, dlatego też logując się na ten typ konta można stworzyć proste podsumowanie informacji na wybrany temat. Ponieważ baza odzwierciedla ewidencje osób zakażonych wirusem Covid-19, dostępnych jest kilka rodzajów raportów – wszystkie przedstawione są jako lista w postaci tabeli z konkretnymi informacjami – m. in. raport o zakażonych z danego województwa (tego, na które aktualnie użytkownik – Sanepid – jest zalogowany), raport o zakażonych z całego kraju, raport o osobach zmarłych, a także o tych, które wyzdrowiały.

Raporty

Wybierz przedział czasowy dla raportu:

☐ Raport z dziś
☐ Raport z ostatnich 30 dni
☒ Raport z całego okresu

Pokaż raport

Pesel	Imię	Nazwisko	Telefon	Miasto	Adres	Województwo	Data wpisu	Miejsce kwarantanny	Szpital
700811815637	Danuta	Dadacka	454604769	Katowice	Szkołna 71	śląskie	2020-12-23	D	

Rysunek 5. Fragment strony wyświetlającej raport o osobach zakażonych dla województwa śląskiego.

Raporty

Wybierz przedział czasowy dla raportu:

☐ Raport z dziś
☐ Raport z ostatnich 30 dni
☒ Raport z całego okresu

Pokaż raport

Województwo	Łączna liczba zakażonych
śląskie	1
świętokrzyskie	1

Rysunek 6. Fragment strony wyświetlającej raport o osobach zakażonych dla całego kraju.

Raport o osobach zmarłych

Wybierz przedział czasowy dla raportu:

☐ Raport zmarłych z tego województwa
☒ Raport zmarłych z całego kraju

Pokaż raport

Pesel	Imię	Nazwisko	Miasto	Skrót województwa	Wiek
800503342945	Cecylia	Cabacka	Kraków	MAL	41
781130183642	Stefan	Abacki	Toruń	KPM	43

Łącznie osób: 2

Rysunek 7. Fragment strony wyświetlającej raport o osobach zmarłych z całego kraju.

Raport o ozdrowieńcach

Wybierz przedział czasowy dla raportu:

☐ Raport ozdrowieńców z tego województwa
☒ Raport ozdrowieńców z całego kraju

Pokaż raport

Pesel	Imię	Nazwisko	Miasto	Skrót województwa	Data wpisu
490623512649	Weronika	Babacka	Opole	OPO	2021-01-24
520531521183	Anna	Szymańska	Sopot	POM	2021-01-24

Łącznie osób: 2

Rysunek 8. Fragment strony wyświetlającej raport o osobach ozdrowiałych z całego kraju.

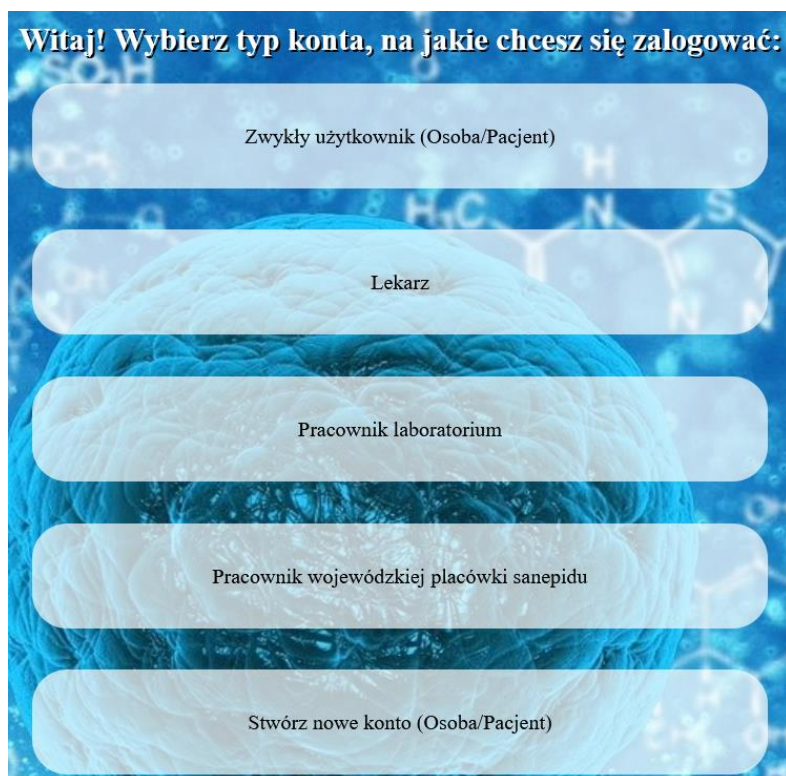
Uwaga: Zaprezentowane zrzuty pochodzą z bazy, w której zedytowano część informacji za pomocą aplikacji bazodanowej (wyniki tych samych operacji dla danych tylko i wyłącznie ze skryptu wypełniającego bazę będą inne).

Do przygotowania raportów użyto widoków – tak, by zabezpieczyć dane i skrócić wywołanie kwerendy; oraz funkcji wbudowanych pozwalających na obliczenie wieku z dwóch pierwszych cyfr peselu osoby (przycinanie stringów, rzutowanie, pobranie aktualnej daty).

Bardziej szczegółowe wyjaśnienie jak wyświetlić dane raporty i jakie warunki posiadają znajduje się w podpunkcie 5.2. *Dokumentacja użytkownika*.

4.3. Zdefiniowanie panelu sterowania aplikacji

Panel sterowania jest bardzo prosty i intuicyjny. Aplikacja została zrealizowana w postaci prostego serwisu internetowego z przyciskami (linkami) do poszczególnych funkcjonalności. Wdrożono prosty mechanizm autoryzacji i autentyfikacji użytkowników. Wszystkie akcje skupiają się na „przeklikaniu” się na odpowiednią witrynę oraz wypełnienie danych.



Rysunek 9. Fragment zrzutu ekranu ze strony startowej serwisu, gdzie użytkownik może wybrać typ konta, na które chce się zalogować lub stworzyć nowe konto.

5. Dokumentacja

5.1. Wprowadzanie danych

Z racji ogromu bazy danych i wielu relacji pomiędzy poszczególnymi tabelami powstał skrypt, który wypełnia bazę odpowiednimi danymi. Są one zawarte w pliku *pop_covid.sql* i, jeżeli baza jest pusta (ważne w razie ewentualnego zresetowania bazy danych), pozwalają na dalszy rozwój bazy po ich wprowadzeniu.

UWAGA: Przykładowe dane zostały wygenerowane przeze mnie z pomocą skryptu w języku Python. Wszelka zbieżność danych osobowych jest przypadkowa.
Uwaga: W celach bezpieczeństwa w projekcie zakłada się, że pesel ma 12 cyfr.

Dalsze wprowadzanie danych jest możliwe dzięki formularzowi, który pozwala tworzyć nowe osoby. Został on wyposażony w podstawowy system sprawdzania poprawności danych do wpisywanych pól (na poziomie HTML – pattern).

Podaj swój PESEL:

Podaj swoje imię:

Podaj swoje nazwisko:

Podaj swoje miasto, w którym jesteś zameldowany/a:

Podaj nazwę ulicy oraz numer mieszkania:

Podaj swój numer telefonu:

Podaj swoje hasło do serwisu:

Wybierz z listy swoje województwo:

zachodniopomorskie

Stwórz konto

Rysunek 10. Fragment zrzutu ekranu przedstawiający formularz do dodania nowej osoby.

5.2. Dokumentacja użytkownika

Po wejściu na stronę pascal.fis.agh.edu.pl/~8rogowski/bazydanych/index.html (**UWAGA: dostęp do strony jest możliwy wyłącznie z sieci wydziałowej, ponadto połączenie z bazą danych realizowane jest również z localhosta, dlatego koniecznym jest tunelowanie na serwer Pascal**) użytkownik może wybrać typ konta, na które chce się zalogować, bądź utworzyć nowe konto osoby.

Każdy typ ma swoje własne unikalne funkcjonalności.

Przed wejściem na dane konto potrzebne jest zalogowanie się na nie odpowiednim loginem (ciąg cyfr o zadanej długości) oraz hasłem.

➤ Osoba



Rysunek 11. Fragment zrzutu ekranu ze strony startowej po zalogowaniu jako Osoba.

Po zalogowaniu się jako osoba u góry widnieje imię osoby oraz panel różnych dostępnych akcji.

- Wyświetl informacje – wyświetla dane osobowych użytkownika oraz użytkowników, którzy podali daną osobę do wglądu w informacje o nich.
- Zmień hasło – wyświetla formularz, w którym znajduje się pole do podania nowego hasła. Po kliknięciu przycisku „Zmień hasło” zapytanie jest przetwarzane, a użytkownik zostanie powiadomiony odpowiednim alertem o statusie zmiany.
- Edytuj dane kontaktowe – wyświetla formularz, w którym znajduje się przycisk, gdzie użytkownik podaje rodzaj zmienianej informacji (osoby kontaktowej albo numeru telefonu) oraz pole do podania nowej wartości uprzednio wybranej informacji. Po kliknięciu przycisku „Dalej” zapytanie jest przetwarzane, a użytkownik zostanie powiadomiony odpowiednim alertem o statusie zmiany.
- Zgłoś zapis na test – wyświetla komunikat, w którym użytkownik potwierdza, czy chce zlecić wykonanie testu. Po jego akceptacji do bazy zostaje wysłane odpowiednie zapytanie, które nie wymaga od użytkownika wprowadzania żadnych dodatkowych informacji.

➤ Lekarz



Rysunek 12. Fragment zrzutu ekranu ze strony startowej po zalogowaniu jako Lekarz.

- Zmień stan pacjenta – wyświetlenie formularz, w którym użytkownik wybiera pacjenta, a następnie przypisuje mu nowy stan. Po kliknięciu przycisku „Zmień stan” zapytanie jest przetwarzane, a użytkownik zostanie powiadomiony odpowiednim alertem o statusie zmiany.
- Zmień hasło – wyświetla formularz, w którym znajduje się pole do podania nowego hasła. Po kliknięciu przycisku „Zmień hasło” zapytanie jest przetwarzane, a użytkownik zostanie powiadomiony odpowiednim alertem o statusie zmiany.
- Nałóż kwarantannę – wyświetla formularz, w którym użytkownik wybiera osobę, której chce nałożyć kwarantannę. Po kliknięciu przycisku „Zmień stan” użytkownik jest proszony o potwierdzenie akcji, a następnie zapytanie jest przetwarzane (baza

automatycznie wstawi nowy rekord do odpowiedniej tabeli) i użytkownik zostanie powiadomiony odpowiednim alertem o statusie zmiany.

➤ Pracownik laboratorium



Rysunek 13. Fragment zrzutu ekranu ze strony startowej po zalogowaniu jako Pracownik laboratorium.

- Uzupełnij wyniki testu – wyświetla formularz, w którym użytkownik wybiera zlecenie testu, które nie zostało jeszcze wykonane, a następnie wpisuje wynik na obecność wirusa u osoby. Po kliknięciu przycisku „Zapisz wynik testu” zapytanie jest przetwarzane, a użytkownik zostanie powiadomiony odpowiednim alertem o statusie zmiany.
- Zmień hasło – wyświetla formularz, w którym znajduje się pole do podania nowego hasła. Po kliknięciu przycisku „Zmień hasło” zapytanie jest przetwarzane, a użytkownik zostanie powiadomiony odpowiednim alertem o statusie zmiany.

➤ Sanepid



Rysunek 14. Fragment zrzutu ekranu ze strony startowej po zalogowaniu jako Sanepid.

- Zakażeni (województwo) – wyświetla formularz, w którym użytkownik wybiera przedział czasowy, dla którego chce otrzymać raport o osobach zakażonych z jego

województwa. Po kliknięciu przycisku „Pokaż raport” zapytanie jest przetwarzane, a użytkownik dostaje listę w postaci tabeli o osobach zakażonych (wszelkie niezbędne informacje).

- Zakażeni (cały kraj) – działanie analogiczne jak wyżej, jednak raport zawiera dane o osobach z całego kraju i wyświetla je zbiorczo (dla każdego województwa liczona jest suma zakażeń spełniających wybrany warunek czasowy).
- Raport o zmarłych – wyświetla formularz, w którym użytkownik wybiera warunek, według którego zostaną wyświetlone dane (pojedyncze województwo lub cały kraj). Po kliknięciu przycisku „Pokaż raport” zapytanie jest przetwarzane, a użytkownik dostaje listę w postaci tabeli o osobach zmarłych (podstawowe informacje).
- Raport o ozdrowieńcach – działanie analogiczne jak wyżej, jednak raport dotyczy informacji o osobach ozdrowiałych.
- Skieruj na izolację – wyświetla formularz, w którym użytkownik wybiera osobę, którą chce skierować na izolację. Po kliknięciu przycisku „Skieruj na izolację” użytkownik jest proszony o potwierdzenie akcji, a następnie zapytanie jest przetwarzane (baza automatycznie wstawi nowy rekord do odpowiedniej tabeli) i użytkownik zostanie powiadomiony odpowiednim alertem o statusie zmiany.
- Zmień hasło – wyświetla formularz, w którym znajduje się pole do podania nowego hasła. Po kliknięciu przycisku „Zmień hasło” zapytanie jest przetwarzane, a użytkownik zostanie powiadomiony odpowiednim alertem o statusie zmiany.
- Wyszukaj osobę – wyświetla formularz, w którym użytkownik proszony jest o podanie przynajmniej jednego z trzech pól – peselu, imienia lub nazwiska, które ma zostać znalezione w bazie danych. Po kliknięciu przycisku „Znajdź osobę” zapytanie jest przetwarzane, a użytkownik dostaje listę w postaci tabeli o osobach o danych spełniających nałożone warunki (podstawowe informacje).

5.3. Opracowanie dokumentacji technicznej

Projekt został zrealizowany głównie w języku PHP, HTML oraz elementów JavaScript zawartych w kodzie (głównie wyświetlanie okienek dialogowych). Z uwagi na wymaganie wielu rodzajów kont użytkowników oraz systemu autoryzacji dostępu do poszczególnych funkcjonalności konieczne było wprowadzenie sesji. Najważniejszym elementem aplikacji jest klasa *Register*, która zawiera wszelkie niezbędne metody do jej prawidłowego funkcjonowania. Opis metod w niej zawartych opisany jest poniżej.

- `__construct()` – konstruktor klasy, odpowiedzialny jest za rozpoczęcie i podtrzymanie sesji (konieczne w celu wdrożenia systemu logowania), realizowane za pomocą metody `session_start()`.
- `execute_sql($query)` – funkcja, która łączy się z bazą danych PostgreSQL na serwerze Pascal za pomocą prywatnie ustawionych zmiennych (jeśli nie uda się tego zrobić – wyświetla alert o stosownym komunikacie) oraz wykonuje dowolne zapytanie

z pomocą funkcji `pg_query` zapisane jako tekst w zmiennej `$query`, następnie zamyka dostęp do bazy oraz zwraca wynik kwerendy.

- `_is_logged_as($type)` – funkcja, której zadaniem jest stwierdzenie, czy użytkownik jest zalogowany i czy ma uprawnienia do wglądu w odpowiednią kategorię, zawartą w zmiennej `$type`, sprawdza, czy w zmiennej `$_SESSION` ustawione są odpowiednie pola, zwraca prawdę, jeśli tak. Np. `_is_logged_as('osoba')` sprawdza, czy użytkownik jest zalogowany jako osoba. (Uwaga: Mechanizm ten jest potrzebny, aby kontrolować dostęp do odpowiednich zasobów na stronie aplikacji, o czym więcej poniżej).
- `_login_as($type)` – metoda, która loguje użytkownika do jednej z 4 kategorii kont (decyduje o tym zmienna `$type`). Pobiera login oraz hasło z formularza do logowania przesłanego metodą POST, a następnie łączy się z bazą danych (jeśli napotka problem z połączeniem, to wyświetla stosowny komunikat) w celu weryfikacji wprowadzonych danych. Jeśli dane się zgadzają, to do zmiennej `$_SESSION` zapisywane są login oraz flaga autentyfikacji, a następnie kieruje użytkownika na stronę główną danego typu konta. Jeśli natomiast dane okażą się błędne, wyświetla alert o stosownym komunikacie, a następnie przekierowuje do strony z nieudanym logowaniem, z której użytkownik może wrócić ponownie do odpowiedniego formularza z logowaniem.
- `_logout()` – metoda, która służy do wylogowania każdego rodzaju użytkownika. Usuwa wszystkie dane, które są przechowywane w zmiennej `$_SESSION`, a następnie kończy sesję – używana jest metoda `session_destroy()`.

Metody te są używane we wszystkich plikach php, zwłaszcza jako mechanizm uwierzytelniający i autoryzujący. Ponieważ schemat kodu jest analogiczny w każdym przypadku, zostanie on omówiony na konkretnym przykładzie, jak pokazano poniżej.

```
include '../Register.php';

$user = new Register;

if ($user->_is_logged_as('sanepid')) {
    $user->
>execute_sql("CREATE OR REPLACE VIEW projekt.widok_ozdrowienicy AS SE
LECT o.osoba_pesel, o.osoba_imie, o.osoba_nazwisko, o.osoba_miasto,
o.wojewodztwo_id, oz.ozdrowieniec_data_wpisu FROM projekt.osoba o JO
IN projekt.pacjent p ON p.osoba_pesel = o.osoba_pesel JOIN projekt.o
zdrowienicy oz ON p.pacjent_id = oz.pacjent_id ORDER BY o.osoba_nazwi
sko, o.osoba_imie");
```

```

        /*...*/
    }
    else
    {
        header("Location: ../index.html");
    }

```

Listing 1. Fragment kodu php (źródło: sanepid_report_recovered.php).

Na początku dołączana jest klasa *Register*, która zawiera cały niezbędny interfejs; tworzony jest obiekt `$user`. Następnie wywoływana jest metoda, która sprawdza czy użytkownik jest zalogowany jako Sanepid, tylko wtedy bowiem mogą wykonywać się pozostałe operacje takie jak wykonywanie określonych zapytań, wyświetlanie formularzy, alertów lub przekierowywanie na dalsze strony (w tym konkretnym przypadku jest to stworzenie widoku przy pomocy funkcji, która obsługuje wszystkie zapytania do bazy, a następnie zapisuje je w zmiennej, którą później przekazuje się dalej w formie odpowiedniej do wyświetlenia (`pg_fetch_row()` bądź `pg_fetch_all()` oraz wyświetlenie elementów w tabeli)). W przeciwnym wypadku – jeśli stwierdzi się, że użytkownik nie ma dostępu do strony – jest on przekierowywany do strony powitalnej *index.html*.

Jak zostało to podkreślony wcześniej, schemat ten (sprawdzenie stanu zalogowania użytkownika, a następnie uruchamianie pozostałych funkcji) jest zastosowany dla niemal wszystkich plików i zapewnia on podstawową kontrolę w dostępie do funkcjonalności aplikacji.

6. Dalsze uwagi

6.1. Przykładowe dane logowania do serwisu

Login Hasło	Typ konta
890725468720 KacperGórny	Osoba
690316543562 KarolinaBabacka	
4 jestemLekarzem	Lekarz
1 covid2020	Pracownik laboratorium
2 Sanepid	Sanepid
14 Sanepid	

6.2. Uwagi dotyczące projektu i ewentualnej rozbudowy w przyszłości

Część kolumn w niektórych encjach nie została użyta (np. pojemność placówek, maksymalna liczba testów, czy wielkość populacji danego województwa). Nie została ona usunięta z uwagi na ogrom bazy i konieczność zmiany większości skryptu wypełniającego bazę, jednak może stanowić dobry początek w ewentualnej rozbudowie bazy.

Nie wszystkie funkcjonalności z początku planu koncepcyjnego zostały wdrożone z uwagi na wielkość bazy oraz powtarzalność niektórych elementów, niemniej jednak główny mechanizm został zaimplementowany.

7. Bibliografia

W przygotowaniu projektu posłużono się poniższymi pomocami:

- materiały dydaktyczne przygotowane przez dr Zimnocha
- wykłady dr Dydejczyka
- własne notatki.