



**AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE**

**WYDZIAŁ INFORMATYKI, ELEKTRONIKI I TELEKOMUNIKACJI**

**INSTYTUT ELEKTRONIKI**

**PRACA DYPLOMOWA INŻYNIERSKA**

*System zabezpieczeń teleinformatycznych dla placówki szpitalnej –  
oprogramowanie systemowe i aplikacja zarządzająca danymi*

*IT security system for a hospital facility – system software and data management  
application*

Autor:

Kierunek studiów:

Typ studiów:

Opiekun pracy:

Aleksander Brachman

Nowoczesne Technologie w Kryminalistyce

Stacjonarne

dr inż. Jacek Stępień

Kraków, 2024

## Spis treści

|  |    |
|--|----|
| 1. Wstęp .....   | 5  |
| 1.1 Cel i zakres pracy .....   | 5  |
| 2. Oprogramowanie serwera szpitalnego.....                               | 7  |
| 2.1 Przegląd rozwiązań.....  | 7  |
| 2.2 Zestaw oprogramowania LAMP .....                                     | 9  |
| 2.2.1 System operacyjny Debian.....                                      | 10 |
| 2.2.2 Serwer HTTP Apache.....  | 11 |
| 2.2.3 System zarządzania bazą danych MariaDB .....                       | 12 |
| 2.2.4 Język programowania PHP .....                                      | 14 |
| 3. Projekt bazy danych i aplikacji webowej .....                         | 15 |
| 3.1 Struktura bazy danych „Szpital” .....                                | 15 |
| 3.1.1 Tabele i relacje pomiędzy tabelami.....                            | 15 |
| 3.1.2 Użytkownicy MariaDB z dostępem do bazy danych „Szpital”.....       | 16 |
| 3.2 Wymagania funkcjonalne i нефункционалне aplikacji webowej .....      | 17 |
| 3.2.1 Wymagania funkcjonalne z perspektywy wszystkich użytkowników ..... | 18 |
| 3.2.2 Wymagania funkcjonalne z perspektywy lekarza.....                  | 19 |
| 3.2.3 Wymagania funkcjonalne z perspektywy rejestratora .....            | 19 |
| 3.2.4 Wymagania нефункционалне .....                                     | 20 |
| 3.3 Przypadki użycia .....   | 20 |
| 3.3.1 Diagramy przypadków użycia .....                                   | 21 |

|  |    |
|--|----|
| 3.3.2 Opis przypadków użycia .....   | 22 |
| 4. Bezpieczeństwo serwera szpitalnego .....                                  | 36 |
| 4.1 Bezpieczeństwo systemu operacyjnego Debian.....                          | 36 |
| 4.1.1 Użytkownicy systemu operacyjnego i rodzaje uprawnień do plików .....   | 36 |
| 4.1.2 Firewall.....  | 37 |
| 4.2 Bezpieczeństwo serwera HTTP Apache .....                                 | 39 |
| 4.2.1 Protokół HTTPS .....   | 39 |
| 4.2.2 Blokada dostępu do folderów .....                                      | 41 |
| 4.2.3 Automatyczna kopia zapasowa serwera oraz logi Apache.....              | 43 |
| 4.3 Bezpieczeństwo systemu zarządzania bazą danych MariaDB .....             | 44 |
| 4.3.1 Uprawnienia przyznane użytkownikom systemu MariaDB.....                | 45 |
| 4.3.2 Zaszyfrowane dane w tabelach.....                                      | 47 |
| 4.3.3 Automatyczna kopia zapasowa bazy danych „Szpital” oraz logi MariaDB .. | 48 |
| 5. Implementacja aplikacji webowej oraz testy funkcjonalne aplikacji .....   | 51 |
| 5.1 Wykorzystane technologie programowania .....                             | 51 |
| 5.2 Łączenie aplikacji z bazą danych .....                                   | 52 |
| 5.3 Ochrona przed atakami SQL Injection .....                                | 53 |
| 5.4 Szyfrowanie i deszyfrowanie danych .....                                 | 54 |
| 5.5 Testy funkcjonalne aplikacji webowej.....                                | 55 |
| 5.5.1 Testy funkcjonalne aplikacji z perspektywy rejestratora .....          | 62 |
| 5.5.2 Testy funkcjonalne aplikacji z perspektywy lekarza.....                | 65 |

|                       |    |
|-----------------------|----|
| 6. Podsumowanie ..... | 74 |
| Bibliografia .....    | 76 |

## 1. Wstęp

W ostatnich kilku latach jesteśmy świadkami wielu zmian dotyczących sposobu postępowania z danymi osobowymi, w tym z danymi wrażliwymi. Do danych wrażliwych zaliczamy m.in. dane medyczne pacjentów przechowywane w różnych placówkach udzielających świadczeń medycznych. Katalizatorem zmian stało się unijne *Ogólne Rozporządzenie o Ochronie Danych Osobowych* (tzw. *RODO*) oraz polska *Ustawa o ochronie danych osobowych z dnia 10 maja 2018 roku*. Wejście w życie *RODO* i nowych regulacji krajowych rozpoczęło ożywioną dyskusję publiczną nt. ochrony i przetwarzania danych osobowych, co przyczyniło się do zwiększenia świadomości społeczeństwa w tym temacie.

Jedną z instytucji, w której codziennie gromadzone i przetwarzane są duże ilości danych osobowych, w tym danych wrażliwych, jest placówka szpitalna. W związku z przechowywaniem danych osobowych i medycznych pacjentów, placówka szpitalna powinna szczególnie zadbać o bezpieczeństwo tych danych. Zgodnie z wynikami kontroli Najwyższej Izby Kontroli (NIK) z 2019 roku [1], w ponad 50% skontrolowanych szpitali doszło do uchybień w sprawie ochrony danych osobowych. Część odnotowanych naruszeń zasad bezpieczeństwa danych osobowych było związanych z przechowywaniem dokumentacji medycznej pacjentów w formie papierowej, co ułatwiało nieuprawnionym osobom dostęp do tych danych. Istotnym problemem, na który także wskazuje raport z kontroli NIK, było przyznanie dostępu do danych medycznych pacjentów nieuprawnionym do tego typu danych pracownikom szpitala, jak salowe, czy sanitariusze.

### 1.1 Cel i zakres pracy

Celem niniejszej pracy jest utworzenie bezpiecznego środowiska serwera szpitalnego oraz zaprojektowanie i zaimplementowanie aplikacji webowej, która pozwoli na zarządzanie danymi zgromadzonymi w serwerze szpitalnym. Odpowiednio zabezpieczone oprogramowanie serwera oraz kod aplikacji webowej mają zapewnić bezpieczeństwo danych osobowych i medycznych pacjentów placówki szpitalnej, które będą gromadzone i przetwarzane w formie elektronicznej.

W ramach osiągnięcia celu pracy dyplomowej skupiono się na wyborze i zaimplementowaniu zestawu oprogramowania dla serwera szpitalnego oraz napisaniu aplikacji webowej, przeznaczonej do pracy z danymi osobowymi i medycznymi pacjentów placówki szpitalnej. Najpierw w rozdziale 2. przedstawiony został przegląd ogólnodostępnych rozwiązań tworzących środowisko serwera. Następnie w rozdziale 3. zaprezentowano projekt szpitalnej bazy danych oraz wymagania funkcjonalne i niefunkcjonalne aplikacji webowej, wraz z przypadkami użycia. Rozdział 4. dotyczy zaimplementowanych rozwiązań zwiększających bezpieczeństwo serwera szpitalnego. W rozdziale 5. przedstawiono implementację aplikacji webowej, w tym wykorzystane do napisania aplikacji technologie programowania, oraz testy funkcjonalne aplikacji webowej. Rozdział 6. stanowi podsumowanie rozważań podjętych w pracy.

*Niniejsza praca domyślnie miała stanowić jedną z dwóch części pracy dyplomowej o temacie „System zabezpieczeń teleinformatycznych dla placówki szpitalnej” autorstwa Aleksandra Brachmana i Jakuba Słoty. Z powodu opóźnienia w realizacji drugiej części pracy (dotyczącej projektu infrastruktury sieciowej dla placówki szpitalnej) przez drugiego z autorów – Jakuba Słotę – praca została rozdzielona na dwie osobne prace o tematach: „System zabezpieczeń teleinformatycznych dla placówki szpitalnej – oprogramowanie systemowe i aplikacja zarządzająca danymi” autorstwa Aleksandra Brachmana oraz „System zabezpieczeń teleinformatycznych dla placówki szpitalnej - sprzętowa struktura sieci” autorstwa Jakuba Słoty. Oprogramowanie serwera szpitalnego opisane w niniejszej pracy dyplomowej ma współpracować z systemem sieciowym placówki szpitalnej, który opisany został w pracy dyplomowej „System zabezpieczeń teleinformatycznych dla placówki szpitalnej - sprzętowa struktura sieci” autorstwa Jakuba Słoty. Nawiązania merytoryczne do systemu sieciowego placówki szpitalnej zostaną wskazane w niniejszej pracy.*

## 2. Oprogramowanie serwera szpitalnego

Istotną częścią infrastruktury teleinformatycznej placówki szpitalnej jest serwer. Urządzenie to służy jako miejsce przechowywania różnego rodzaju danych, jednocześnie umożliwiając uprawnionemu użytkownikowi uzyskanie dostępu do tych danych, z wykorzystaniem sieci teleinformatycznej. W celu zapewnienia prawidłowego działania różnorodnych funkcji, jakie powinno gwarantować oprogramowanie serwera, konieczny jest wybór takiego rozwiązania, którego poszczególne elementy tworzą integralny, niezawodny zestaw oprogramowania serwerowego. Niniejszy rozdział składa się z przeglądu ogólnodostępnych, darmowych rozwiązań tworzących oprogramowanie (środowisko) serwera oraz omówienia wybranego rozwiązania.

### 2.1 Przegląd rozwiązań

Ogólnodostępne rozwiązania najczęściej przyjmują postać zestawów oprogramowania, składających się z następujących komponentów [2]:

- systemu operacyjnego, który stanowi podstawę do pracy pozostałych elementów,
- serwera web (serwera HTTP) umożliwiającego dostęp z sieci do stron internetowych i aplikacji webowych,
- systemu zarządzania bazą danych (dalej jako SZBD), w której przechowuje się informacje,
- języka programowania, który umożliwia stworzenie strony internetowej lub aplikacji webowej.

Cechami wspólnymi wyżej wspomnianych komponentów jest ich otwartoźródłowość oraz możliwość bezpłatnego użytkowania, co czyni je rozsądnymi alternatywami dla gotowych, płatnych rozwiązań komercyjnych. Poniżej znajduje się lista wybranych, najpopularniejszych zestawów oprogramowania typu *open source*, przeznaczonych do stworzenia środowiska serwera. Nazwy na liście składają się z pierwszych liter elementów, które wchodzi w skład danego zestawu.

- LAMP
  - System operacyjny: Linux,
  - Serwer web: Apache HTTP Server,
  - SZBD: MySQL lub MariaDB,
  - Język programowania: PHP/Perl/Python.
- LEMP
  - System operacyjny: Linux,
  - Serwer web: Nginx,
  - SZBD: MySQL lub MariaDB,
  - Język programowania: PHP/Perl/Python.
- LAPP
  - System operacyjny: Linux,
  - Serwer web: Apache,
  - SZBD: PostgreSQL,
  - Język programowania: PHP.
- LLMP
  - System operacyjny: Linux,
  - Serwer web: Lighttpd,
  - SZBD: MySQL/MariaDB,
  - Język programowania: PHP/Perl/Python.

Jak można zauważyć, podstawą wyżej wymienionych zestawów jest system operacyjny oparty na Linuxie oraz relacyjne SZBD takie jak MySQL, PostgreSQL i MariaDB. Wcześniej wspomniana bezpłatność użytkowania i otwartoźródłowość są niewątpliwymi zaletami takich zestawów, których mocnymi stronami są równocześnie: szerokie wsparcie, wynikające z rozbudowanej społeczności Linuxa, Apache czy baz danych typu SQL oraz łatwa konfiguracja. Do wad należy zaliczyć konieczność opierania się na jednym systemie operacyjnym, czy mniejszą wydajność relacyjnych SZBD w porównaniu do ich nierelacyjnych odpowiedników.

W przypadku nierelacyjnych SZBD ogólnodostępną propozycją jest MEAN [3], który w części „bazodanowej” opiera się na nierelacyjnym SZBD MongoDB, a pozostałe komponenty stworzone zostały na podstawie języka programowania JavaScript – Express.js, Angular.js oraz Node.js. Pozwala to deweloperowi na używanie tego samego

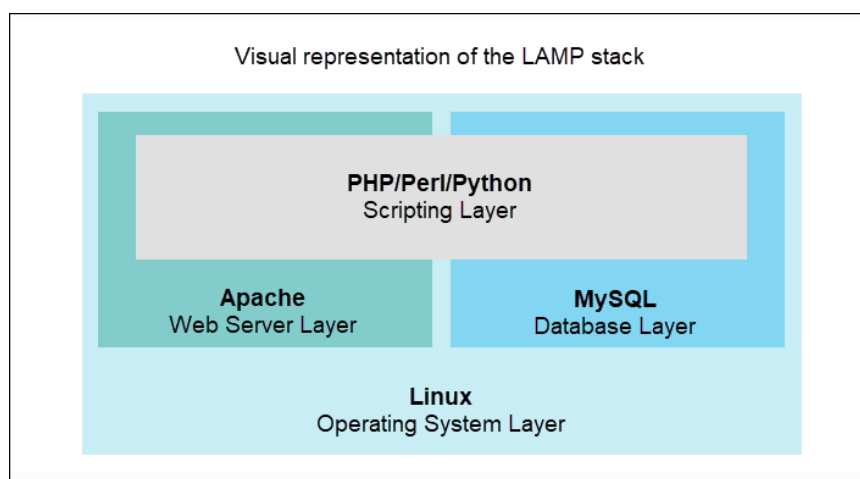


języka zarówno po stronie backendowej, jak i frontendowej aplikacji, przyspieszając tym samym jej tworzenie oraz działanie, także dzięki szybszej, nierelacyjnej bazie danych. Co ważne, podstawą zestawu MEAN może być każdy system operacyjny, który wspiera środowisko Node.js. Do minusów MEAN należy zaliczyć fakt, że rozwiązanie to nie jest zalecane do budowy aplikacji dużej wielkości oraz, że tylko doświadczeni twórcy aplikacji będą w stanie wykorzystać cały jego potencjał.

## 2.2 Zestaw oprogramowania LAMP

Jednym z przedstawionych w poprzednim podrozdziale rozwiązań jest zestaw oprogramowania LAMP. Biorąc pod uwagę wcześniej wspomniane zalety zestawu LAMP, a zwłaszcza jego darmowość, łatwą konfigurację oraz możliwość znalezienia rozwiązania potencjalnego problemu w szybki sposób (dzięki szerokiemu wsparciu, idącemu za popularnością dystrybucji Linuxa, Apache czy MariaDB wśród społeczności zajmującej się tworzeniem i hostowaniem aplikacji webowych), zdecydowano się na wybór właśnie tego zestawu oprogramowania do stworzenia środowiska serwera szpitalnego.

LAMP składa się z czterech warstw oprogramowania [2]: warstwy systemu operacyjnego, warstwy bazy danych, warstwy serwera web (serwera HTTP) oraz warstwy skryptowej. Warstwy te, wraz z nazwami komponentów mogących wchodzić w ich skład, zostały zwizualizowane na rys. 1.



Rys. 1. Wizualizacja zestawu LAMP za pomocą warstw oprogramowania, (źródło: [2])

W implementacji zestawu oprogramowania LAMP wykorzystanej na potrzeby pracy dyplomowej, w skład warstwy bazy danych wchodzi MariaDB, a nie MySQL. Nie jest to jednak istotna zmiana, ponieważ system MariaDB powstał na bazie systemu MySQL [4].

MariaDB, jak i pozostałe komponenty tworzące całość oprogramowania serwera szpitalnego, zostaną przybliżone w dalszej części rozdziału.

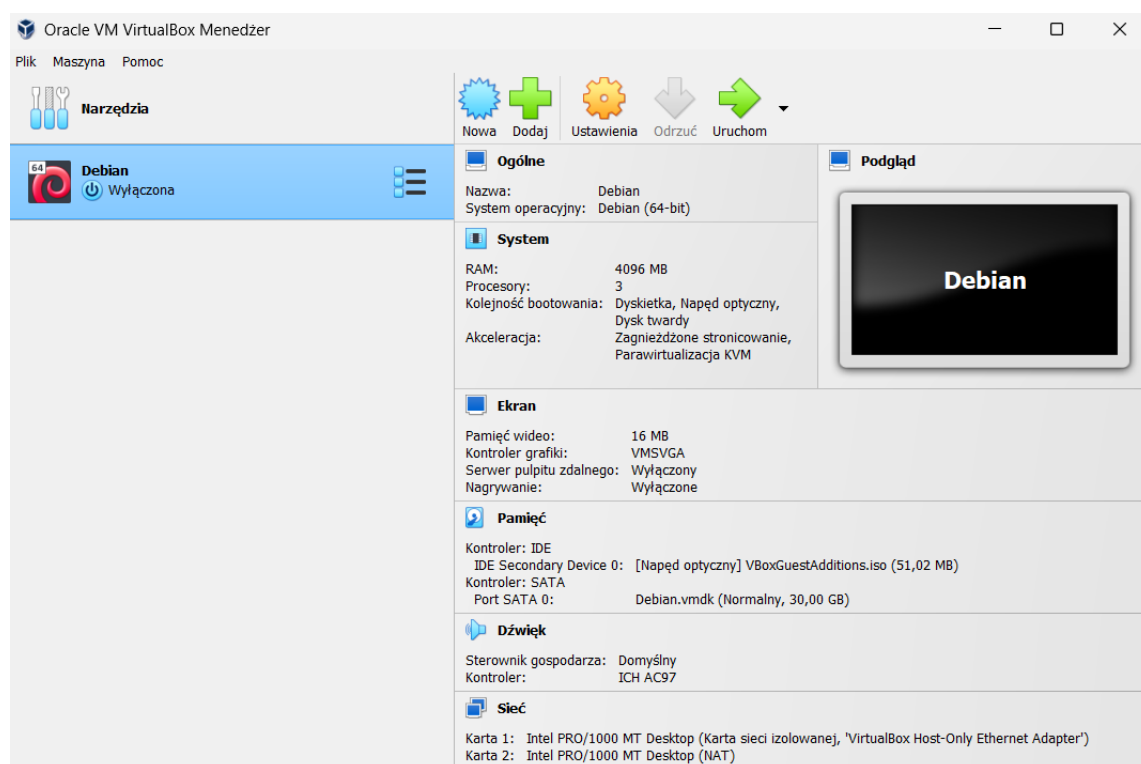
### 2.2.1 System operacyjny Debian

Linux [5] to rodzina systemów operacyjnych opartych na wspólnym jądrze, których główną cechą jest otwartość kodu źródłowego. W skład tej rodziny wchodzi wiele różnych dystrybucji Linuxa, m.in. Ubuntu, Fedora czy Debian. Systemy oparte na jądrze Linux są na ogół wykorzystywane jako systemy operacyjne serwerów, firewalli, bądź routerów. Ze względu na możliwość modyfikacji kodu źródłowego, niezawodność, czy bezpieczeństwo, systemy operacyjne Linux są wykorzystywane w ważnych instytucjach finansowych, np. na giełdzie londyńskiej [6], w diagnostyce medycznej [7] oraz przez siły zbrojne państw, np. Marynarkę Wojenną USA [8].

Rolę systemu operacyjnego serwera szpitalnego pełni dystrybucja Linuxa o nazwie Debian [9]. Dystrybucja ta została po raz pierwszy zapowiedziana 16 sierpnia 1993 roku. Autorem pierwszej wersji Debiana, która została wydana 15 września 1993 roku, był Ian Murdock, ówczesny student uniwersytetu Purdue w stanie Indiana, USA. Głównym motto wielu tysięcy osób utrzymujących i rozwijających Debiana jest bezinteresowna chęć tworzenia wolnego systemu operacyjnego, dostępnego dla każdej osoby [10]. Słowo wolny ma zarazem oznaczać, że dostęp do systemu jest bezpłatny, a także, że każdy użytkownik Debiana posiada swobodę w użytkowaniu i modyfikowaniu systemu. Innymi cechami charakterystycznymi tej dystrybucji jest jej stabilność, wynikająca z ponad trzydziestoletniego rozwoju, zapewnianie regularnych aktualizacji systemu oraz bardzo duża liczba dostępnych do zainstalowania pakietów oprogramowania, które pozwalają na dostosowanie systemu do konkretnych potrzeb osób lub instytucji [11]. W kontekście zalet Debiana należy również wspomnieć o profesjonalnym podejściu jego twórców do kwestii bezpieczeństwa. Wszelkie zgłoszenia o problemach, bądź potencjalnych podatnościach pakietów oprogramowania

dostarczanych na Debian, są rozpatrywane na bieżąco i naprawiane w możliwie jak najszyszym czasie [12].

W ramach pracy dyplomowej system operacyjny Debian w wersji 12.1 został zainstalowany na maszynie wirtualnej, utworzonej w programie Oracle VM VirtualBox w wersji 7.0.10. Podstawowe ustawienia sprzętowe maszyny przedstawione są na rys. 2.



*Rys. 2. Podstawowe ustawienia sprzętowe maszyny wirtualnej z zainstalowanym systemem operacyjnym Debian (źródło własne)*

### 2.2.2 Serwer HTTP Apache

Rolę serwera HTTP w zestawie LAMP pełni serwer HTTP Apache [13], który po raz pierwszy został wydany w 1995 roku. Głównym założeniem osób opiekujących się projektem Apache jest utrzymywanie i rozwój ogólnodostępnego serwera HTTP w postaci *open source* dla systemów operacyjnych opartych na jądrze Linuxa oraz Windowsa. Serwer HTTP Apache ma zapewniać wysoką wydajność, bezpieczeństwo oraz możliwość poszerzania o nowe moduły.

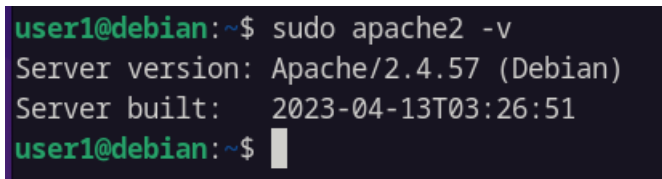
Podobnie jak w przypadku innych serwerów HTTP, głównym zadaniem serwera Apache jest obsługa zapytań wysyłanych przez użytkowników za pomocą przeglądarki internetowej. Zapytania te mogą przyjmować dwie formy [14]:

- statyczną – w postaci adresu URL. W odpowiedzi serwer Apache przesyła do przeglądarki plik odpowiadający żądaniu zawartemu w adresie URL.
- dynamiczną – w momencie gdy użytkownik, swoimi działaniami w przeglądarce, chce spowodować jakąś zmianę, np. struktury strony.

Serwer HTTP Apache jest jednym z najpopularniejszych serwerów HTTP wykorzystywanych do hostingu stron WWW [15]. Zgodnie z danymi serwisu *Netcraft* [15], w maju 2020 roku udział serwerów Apache wśród wszystkich serwerów HTTP wynosił 25,45%. Lepszym wynikiem może pochwalić się tylko serwer HTTP nginx – 36%. Na trzecim miejscu znalazło się rozwiązanie firmy Microsoft z wynikiem procentowym 12,52%.

Oprogramowanie Apache oferuje możliwość konfiguracji serwera za pomocą domyślnych plików konfiguracyjnych oraz specjalnych plików konfiguracyjnych typu *.htaccess* [14]. Udziela również wsparcia certyfikatom TLS/SSL, koniecznym do ustanowienia szyfrowanej komunikacji pomiędzy użytkownikiem (klientem), a serwerem za pomocą protokołu *HTTP over TLS*, czyli HTTPS [16].

Zgodnie z rys. 3, serwer HTTP Apache został zainstalowany na systemie operacyjnym Debian w wersji 2.4.57.



```
user1@debian:~$ sudo apache2 -v
Server version: Apache/2.4.57 (Debian)
Server built: 2023-04-13T03:26:51
user1@debian:~$
```

Rys. 3. Zainstalowana wersja serwera HTTP Apache (źródło własne)

### 2.2.3 System zarządzania bazą danych MariaDB

Kolejnym komponentem zestawu oprogramowania LAMP jest system zarządzania bazą danych. W zależności od preferencji osoby administrującej serwerem,

systemem tym może być MySQL lub MariaDB. W implementacji zestawu oprogramowania LAMP na potrzeby pracy dyplomowej wybrano system MariaDB.

Ważnym aspektem łączącym systemy MySQL i MariaDB jest ich relacyjny charakter [17]. Relacyjne bazy danych przechowują dane w uporządkowany sposób w tabelach, składających się z kolumn i wierszy [18]. Relacje pomiędzy tabelami w takiej bazie danych są możliwe dzięki unikalnym kluczom (kluczem jest jedna z zdefiniowanych w tabeli kolumn), które zawiera każda z tabel. Klucz jednej tabeli może zostać wykorzystany jako klucz obcy w drugiej tabeli, tworząc tym samym relację pomiędzy obiema tabelami i łącząc zawarte w nich informacje.

System zarządzania bazą danych MariaDB [19] został wydany po raz pierwszy 29 października 2009 roku, pod głównym nadzorem Michaela Wideniusa, współtwórcy MySQL. Nazwa MariaDB pochodzi od imienia córki Wideniusa, Marii. MariaDB, podobnie jak system operacyjny Debian, czy serwer HTTP Apache, jest oprogramowaniem typu *open source*.

MariaDB opiera się na języku SQL [20], przez co dodawanie i modyfikacja danych w bazie może odbywać się w przystępny dla użytkownika sposób. W porównaniu do MySQL lub PostgreSQL, MariaDB cechuje się szybkim odczytem danych z bazy [17]. Zalety MariaDB zostały zauważone m.in. przez firmę Google, która wykorzystuje to oprogramowanie w swojej działalności [20]. Jeżeli chodzi o aspekt bezpieczeństwa, MariaDB oferuje możliwość ustanowienia szyfrowanego przesyłu danych, monitoringu dostępu do bazy i jej danych, czy zaawansowanych form uwierzytelniania, takich jak dwustopniowa autentyfikacja przy logowaniu się do systemu MariaDB [21].

System zarządzania bazą danych MariaDB w wersji 10.11.4 został zainstalowany na systemie operacyjnym Debian (patrz: rys. 4).

```
user1@debian:~$ mysql -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.11.4-MariaDB-1~deb12u1-log Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```

*Rys. 4. Zainstalowana wersja MariaDB (źródło własne)*

#### 2.2.4 Język programowania PHP

Ostatnią literą w wyrażeniu „LAMP” jest litera P, która odnosi się do jednego z trzech języków programowania wykorzystywanych przy tworzeniu aplikacji webowych: PHP, Python lub Perl [3]. W implementacji zestawu oprogramowania LAMP na potrzeby pracy dyplomowej wybrano język PHP.

Język programowania PHP jest powszechnie stosowany [22] jako narzędzie do tworzenia aplikacji webowych oraz stron WWW. Istotnymi zaletami PHP jest możliwość łączenia tego języka w trakcie programowania aplikacji webowej z językiem HTML lub JavaScript oraz możliwość ustanowienia za jego pomocą, w prosty sposób, połączenia pomiędzy aplikacją webową, a bazą danych. PHP zawiera również funkcje pozwalające na obsługę zapytań w języku SQL, tym samym umożliwiając m.in. wyświetlanie w aplikacji webowej różnych danych pobranych z bazy.

W pracy wykorzystano pakiet *php* w wersji 8.2.7 (patrz: rys. 5).

```
user1@debian:~$ php -i
phpinfo()
PHP Version => 8.2.7
```

*Rys. 5. Zainstalowana wersja języka programowania PHP (źródło własne)*

### 3. Projekt bazy danych i aplikacji webowej

Utworzenie środowiska serwera szpitalnego pozwala na przejście do etapu projektu bazy danych i aplikacji webowej. W niniejszym rozdziale przedstawiona zostanie struktura bazy danych „Szpital”, na której będzie opierać się działanie aplikacji webowej – tabele w niej zawarte, relacje pomiędzy tabelami oraz wykaz użytkowników uprawnionych do operacji na bazie danych. W kontekście aplikacji webowej, przedstawione zostaną grupy użytkowników, które będą korzystać z aplikacji, zaprezentowane zostaną wymagania funkcjonalne i нефункционалне aplikacji oraz przypadki użycia (w formie graficznej i opisowej).

#### 3.1 Struktura bazy danych „Szpital”

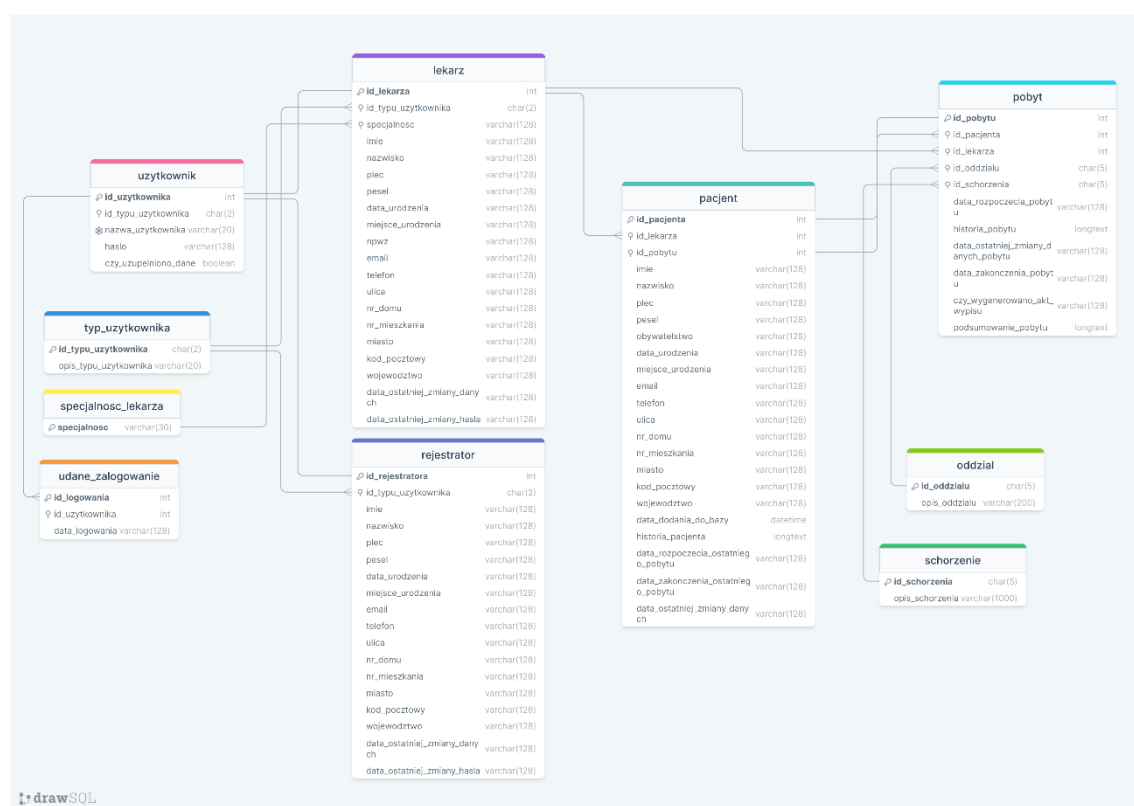
##### 3.1.1 Tabele i relacje pomiędzy tabelami

Szpitalna baza danych (dalej jako baza danych „Szpital”) składa się z dziesięciu tabel:

- *uzytkownik* – przechowuje login i hasło użytkownika, jego numer ID oraz informację o typie (roli) danego użytkownika,
- *typ\_uzytkownika* – przechowuje dostępne typy (role) użytkownika,
- *lekarz* – przechowuje numer ID lekarza, jego dane osobowe, zawodowe (specjalność lekarska, Numer Prawa Wykonywania Zawodu) i kontaktowe,
- *rejestrator* – przechowuje numer ID rejestratora, jego dane osobowe i kontaktowe,
- *pacjent* – przechowuje numer ID pacjenta, jego dane osobowe i kontaktowe, a także historię medyczną pacjenta,
- *pobyt* – przechowuje numer ID pobytu wraz z jego historią, numer ID pacjenta, którego dotyczy dany pobyt i numer ID lekarza prowadzącego pacjenta,
- *oddzial* – przechowuje ID danego oddziału szpitalnego oraz jego opis,
- *schorzenie* – przechowuje ID danego schorzenia (ID schorzenia na bazie ICD-10) oraz jego opis,
- *specjalnosc\_lekarza* – przechowuje specjalności lekarskie,

- *udane\_zalogowanie* – przechowuje daty udanych zalogowań użytkowników do aplikacji wraz z numerem ID danego logowania.

Rys. 6 zawiera diagram relacji pomiędzy tabelami bazy danych „Szpital”. Na diagramie każda tabela zawiera wszystkie zdefiniowane dla niej kolumny. Każda kolumna składa się z nazwy kolumny i jej typu. W tabelach dominują dwa typy kolumn: *varchar(N)*, gdzie *N* to liczba naturalna oraz *int* (skrót od *integer*). *varchar(N)* służy do przechowywania danych tekstowych (mogą składać się z liter i cyfr), gdzie *N* określa maksymalną liczbę znaków, które może przechowywać kolumna. *int* natomiast służy do przechowywania danych liczbowych.



Rys. 6. Diagram relacji tabel w bazie danych „Szpital” (diagram powstał za pomocą strony [www.drawsql.app](http://www.drawsql.app))

### 3.1.2 Użytkownicy MariaDB z dostępem do bazy danych „Szpital”

Dostęp do bazy danych „Szpital” przydzielony zostanie użytkownikom systemu MariaDB o nazwach:

- *lekarz*,



- *rejestracja*,
- *uwierzytelnienie*,
- *zmiana\_hasla*,
- *admin*.

Należy przy okazji wyróżnić cztery podstawowe operacje, które można wykonać na danych zawartych w bazie, należące do podzbioru DML (ang. *Data Modification Language*) języka SQL [18]:

- operacja SELECT – służąca pobieraniu informacji z bazy,
- operacja INSERT – służąca umieszczaniu w tabeli nowych rekordów (wierszy) z informacjami,
- operacja DELETE – służąca usuwaniu z tabeli istniejących rekordów,
- operacja UPDATE – służąca modyfikowaniu informacji zawartych w wierszach tabeli.

Każdy zdefiniowany użytkownik z dostępem do bazy danych „Szpital” (poza użytkownikiem *admin*) powinien mieć ograniczone możliwości korzystania z ww. operacji. Dla przykładu, użytkownicy *lekarz* lub *rejestracja* nie powinni posiadać uprawnień do wykonywania operacji na tabeli *uzytkownik*, związanych z pobieraniem z tabeli hasła lub jego modyfikacją. Za pobieranie hasła z tabeli *uzytkownik*, w trakcie próby uwierzytelnienia użytkownika, powinien odpowiadać tylko użytkownik *uwierzytelnienie*. Zaś za modyfikację hasła w tabeli powinien odpowiadać tylko użytkownik *zmien\_haslo*. Uprawnienia przyznane poszczególnym użytkownikom MariaDB zostaną przedstawione w rozdziale 4.3.1.

## 3.2 Wymagania funkcjonalne i нефункционалне aplikacji webowej

Przed rozpoczęciem rozważań nad wymaganiami funkcjonalnymi i нефункционалnymi aplikacji, należy określić grupy użytkowników, które będą korzystać z aplikacji. Pierwszą grupę użytkowników aplikacji będą stanowić lekarze, a drugą rejestratorzy. Są to dwie grupy pracowników szpitala, które muszą posiadać dostęp do danych osobowych i - w przypadku lekarzy – danych medycznych pacjentów. Zgodnie z celami pracy dyplomowej, dane te będą dostępne przez aplikację webową.

Wymagania funkcjonalne projektowanej aplikacji webowej można podzielić na trzy kategorie: wymagania funkcjonalne aplikacji z perspektywy wszystkich jej użytkowników (czyli lekarzy i rejestratorów), wymagania funkcjonalne aplikacji z perspektywy lekarza oraz wymagania funkcjonalne aplikacji z perspektywy rejestratora.

### 3.2.1 Wymagania funkcjonalne z perspektywy wszystkich użytkowników

- Możliwość zalogowania się do aplikacji.
- Po pierwszym zalogowaniu się użytkownika w aplikacji, użytkownik musi zapisać w bazie danych, poprzez specjalny formularz, swoje dane osobowe, zawodowe (w przypadku lekarza) oraz kontaktowe.
  - Brak zapisu własnych danych przez użytkownika w bazie danych uniemożliwia przejście użytkownikowi do innych, przeznaczonych dla użytkownika, części aplikacji.
  - Po poprawnym zrealizowaniu zapisu danych użytkownika w bazie danych, użytkownik ma możliwość zmiany hasła przydzielonego przez Administratora.
- Wyświetlenie takiej strony głównej aplikacji po zalogowaniu się użytkownika, która jest adekwatna do roli (lekarz lub rejestrator) zalogowanego użytkownika.
- Możliwość zmiany hasła przez użytkownika.
- Możliwość wyświetlenia listy z datami ostatnich (dziesięciu, dwudziestu lub czterdziestu) zalogowań danego użytkownika.
- Możliwość wyświetlenia listy lekarzy znajdujących się w bazie danych.
- Możliwość wyświetlenia danych osobowych i kontaktowych użytkownika.
- Możliwość zmiany danych kontaktowych przez użytkownika.
- Wyświetlenie daty ostatniej zmiany hasła lub ostatniej zmiany danych użytkownika.
- Blokowanie dostępu (z poziomu kodu aplikacji) zalogowanego lekarza do części aplikacji przeznaczonej dla rejestratora i *vice versa*.

### 3.2.2 Wymagania funkcjonalne z perspektywy lekarza

- Możliwość dodania nowego pobytu, wraz z wskazaniem oddziału szpitalnego, w którym znajduje się prowadzony pacjent, rodzaju schorzenia pacjenta (wg ICD-10) oraz zapisem początkowych informacji nt. pobytu w formie tekstu.
- Wyświetlenie na stronie głównej listy aktualnie prowadzonych pacjentów przez lekarza.
- Możliwość wyświetlenia i zmiany danych kontaktowych prowadzonego pacjenta.
- Możliwość wyświetlenia historii medycznej prowadzonego pacjenta, jeżeli ten przebywał już w przeszłości w szpitalu.
- Możliwość wyświetlenia danych związanych z pobytem prowadzonego pacjenta, wraz z historią pobytu pacjenta.
- Możliwość aktualizacji danych pobytu prowadzonego pacjenta – możliwość aktualizacji oddziału, w którym znajduje się pacjent, aktualizacji rodzaju schorzenia oraz możliwość dodania nowych informacji w formie tekstu.
- Możliwość zakończenia pobytu prowadzonego pacjenta poprzez dokonanie wypisu pacjenta ze szpitala.
- Wygenerowanie aktu wypisu pacjenta, zawierającego dane osobowe pacjenta oraz informacje związane z pobytem prowadzonego pacjenta.
- Możliwość wyświetlania aktu wypisu pacjenta do 48 godzin od momentu zakończenia pobytu prowadzonego pacjenta.
- Wyświetlanie daty ostatniej zmiany danych prowadzonego pacjenta i danych pobytu.

### 3.2.3 Wymagania funkcjonalne z perspektywy rejestratora

- Możliwość zarejestrowania nowego pacjenta (dodania pacjenta do bazy danych), wraz z jego danymi osobowymi i kontaktowymi.
- Możliwość przeglądania listy zarejestrowanych pacjentów, wraz z opcją wyszukiwania pacjentów za pomocą numeru ID pacjenta, PESELu pacjenta lub nazwiska pacjenta.
- Wyświetlenie na liście pacjentów informacji czy dany pacjent przebywa aktualnie w szpitalu, jeżeli tak, to od kiedy i kto jest jego lekarzem prowadzącym.

- Możliwość wyświetlenia i zmiany danych kontaktowych zarejestrowanego pacjenta.
- Wyświetlenie daty ostatniej zmiany danych zarejestrowanego pacjenta.

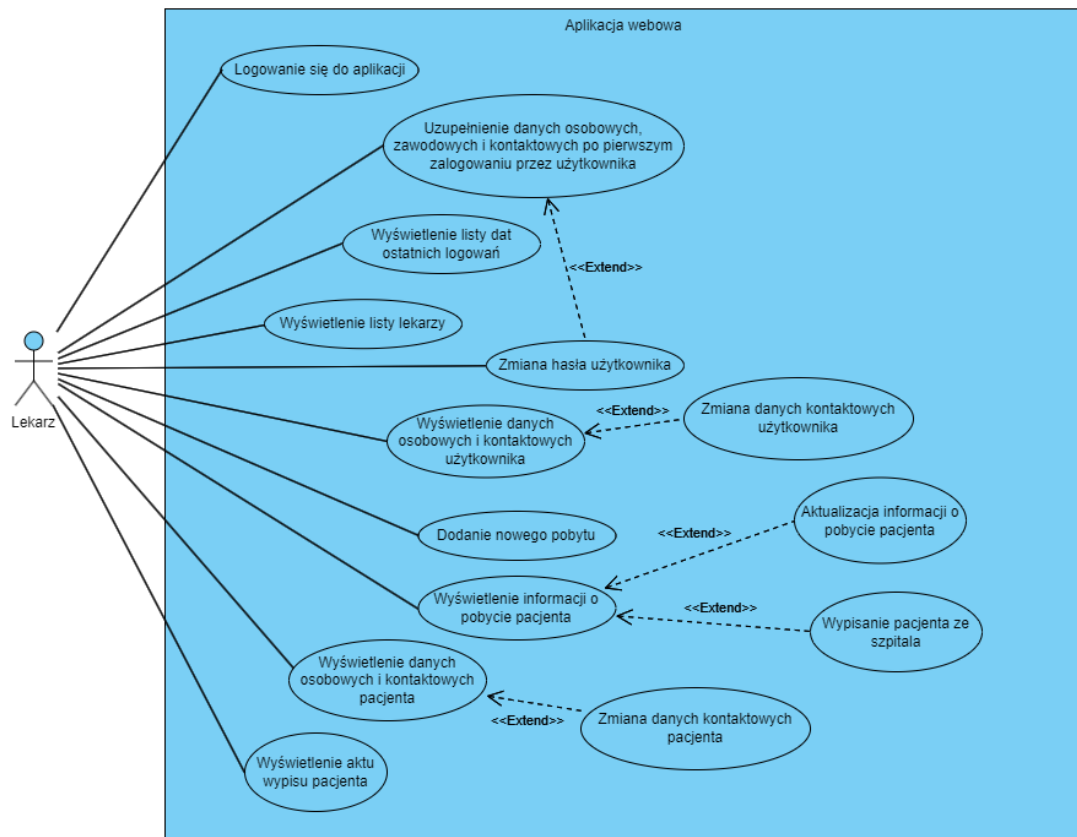
#### 3.2.4 Wymagania niefunkcjonalne

- Z aplikacji można korzystać wyłącznie na komputerach szpitalnych o określonych adresach IP (wyjątek dla lekarzy, którzy mogą korzystać z aplikacji poprzez tunel VPN do sieci szpitalnej) - *zgodnie z założeniami systemu sieciowego placówki szpitalnej, który opisany został w pracy dyplomowej „System zabezpieczeń teleinformatycznych dla placówki szpitalnej - sprzętowa struktura sieci” autorstwa Jakuba Słoty.*
- Działanie aplikacji na najpopularniejszych przeglądarkach internetowych: Google Chrome, Opera, Microsoft Edge, Mozilla Firefox i inne.
- Bezproblemowe łączenie się aplikacji z bazą danych.
- Kod aplikacji zmniejszający ryzyko ataku typu SQL Injection.
- Szyfrowanie danych, w szczególności danych medycznych pacjentów, przed ich zapisem w bazie danych i deszyfrowanie danych pobieranych z bazy danych.
- Przejrzysty interfejs graficzny.

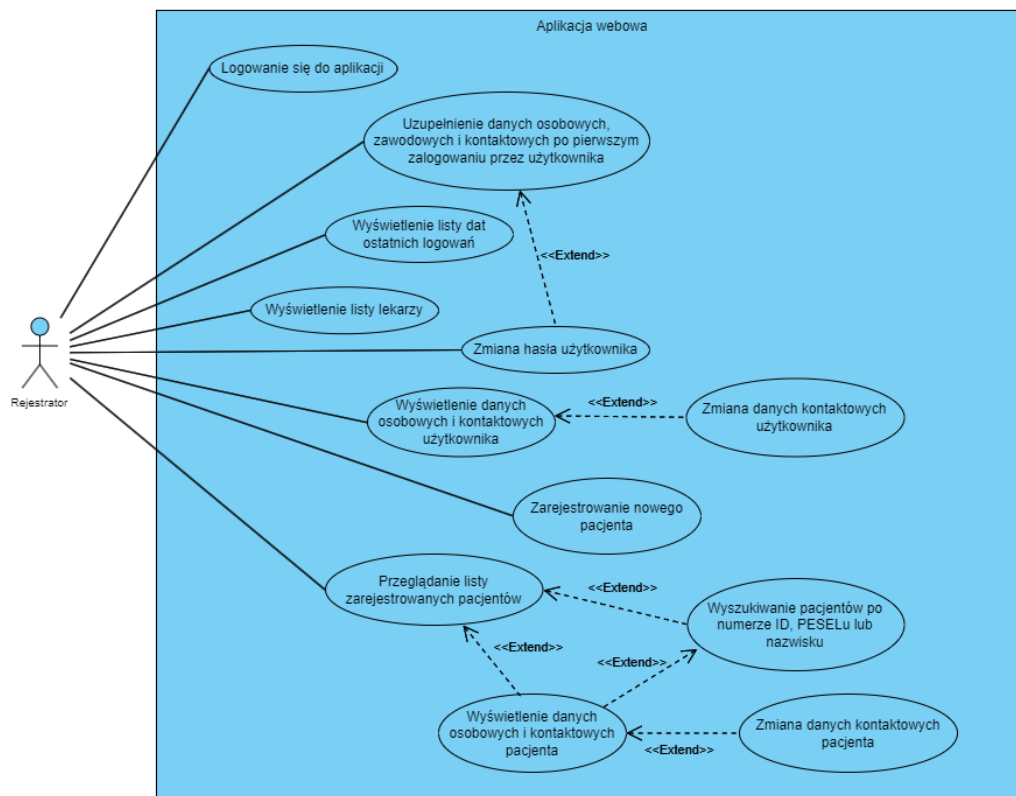
### 3.3 Przypadki użycia

W celu stworzenia diagramów przypadków użycia (patrz: rys. 7 oraz 8) oraz ich opisu w formie tabel (patrz: tab. 1 do 17), rolę aktorów w przypadkach użycia przyjęli Lekarz i Rejestrator.

### 3.3.1 Diagramy przypadków użycia



Rys. 7. Diagram przypadków użycia dla aktora Lekarz (diagram utworzono za pomocą strony: [www.online.visual-paradigm.com/pl/](http://www.online.visual-paradigm.com/pl/))



Rys. 8. Diagram przypadków użycia dla aktora Rejestrator (diagram utworzono za pomocą strony: [www.online.visual-paradigm.com/pl/](http://www.online.visual-paradigm.com/pl/))

### 3.3.2 Opis przypadków użycia

|                             |  |
|-----------------------------|--|
| <b>Kod przypadku użycia</b> | PU-1   |
| <b>Nazwa</b>                | Logowanie się do aplikacji   |
| <b>Aktor</b>                | Lekarz/Rejestrator   |
| <b>Warunki początkowe</b>   | Lekarz/Rejestrator posiadają login i hasło, konieczne do zalogowania się.  |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Lekarz/Rejestrator wpisuje swój login i hasło na stronie logowania.</li> <li>2. Lekarz/Rejestrator korzysta z przycisku „Zaloguj się”.</li> <li>3. Wyświetlona zostaje strona główna aplikacji, adekwatna dla Lekarza/Rejestratora.</li> </ol> |
| <b>Ścieżka alternatywna</b> | <ol style="list-style-type: none"> <li>3a. W przypadku wpisania błędnego loginu/hasła przez Lekarza/Rejestratora lub wystąpienia błędu w procesie uwierzytelnienia użytkownika, na stronie logowania pojawia się odpowiedni komunikat.</li> </ol>  |
| <b>Warunki końcowe</b>      | Lekarz/Rejestrator poprawnie logują się do aplikacji.  |

Tabela 1. PU-1 – Logowanie się do aplikacji

|                             |  |
|-----------------------------|--|
| <b>Kod przypadku użycia</b> | PU-2   |
| <b>Nazwa</b>                | Uzupełnienie danych osobowych i kontaktowych po pierwszym zalogowaniu przez użytkownika  |
| <b>Aktor</b>                | Lekarz/Rejestrator   |
| <b>Warunki początkowe</b>   | <ul style="list-style-type: none"> <li>• Lekarz/Rejestrator zalogował się do aplikacji,</li> <li>• Lekarz/Rejestrator nie zapisali danych w bazie danych.</li> </ul>   |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Po pierwszym zalogowaniu się Lekarza/Rejestratora w aplikacji, wyświetlona zostaje strona z formularzem, w której Lekarz/Rejestrator proszony jest o uzupełnienie swoich danych osobowych, zawodowych (w przypadku lekarza) oraz kontaktowych.</li> <li>2. Lekarz/Rejestrator potwierdza wypełnienie pól w formularzu, klikając przycisk „Uzupełnij dane”.</li> <li>3. Dane zostają zapisane w bazie danych, a Lekarz/Rejestrator zostaje przekierowany na stronę, w której może zmienić hasło ustanowione przez Administratora lub przejść dalej do strony głównej.</li> </ol>  |
| <b>Ścieżka alternatywna</b> | <ol style="list-style-type: none"> <li>1a. Strona z formularzem może zostać wyświetlona również po drugim, trzecim i kolejnych zalogowaniach się Lekarza/Rejestratora. Taki wariant możliwy jest dopóki Lekarz/Rejestrator nie zapiszą swoich danych w bazie danych.</li> <li>3a. W przypadku niewypełnienia wszystkich obowiązkowych pól w formularzu, bądź gdy pole zostało uzupełnione w błędnym formacie, po kliknięciu przycisku „Uzupełnij dane” pojawia się odpowiedni komunikat.</li> <li>3b. W przypadku niepowodzenia zapisania podanych danych w bazie danych, na stronie z formularzem pojawia się odpowiedni komunikat i Lekarz/Rejestrator wypełniają formularz ponownie.</li> </ol> |
| <b>Warunki końcowe</b>      | Dane osobowe, zawodowe (w przypadku Lekarza) oraz kontaktowe Lekarza/Rejestratora zostały poprawnie zapisane w bazie danych.   |

*Tab. 2. PU-2 – Uzupełnienie danych osobowych i kontaktowych przez użytkownika po pierwszym zalogowaniu*

|                             |   |
|-----------------------------|---|
| <b>Kod przypadku użycia</b> | PU-3  |
| <b>Nazwa</b>                | Wyświetlenie listy dat ostatnich logowań  |
| <b>Aktor</b>                | Lekarz/Rejestrator  |
| <b>Warunki początkowe</b>   | Lekarz/Rejestrator zalogował się do aplikacji co najmniej jeden raz.  |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Lekarz/Rejestrator korzysta z zakładki „Lista ostatnich logowań”, znajdującej się na stronie głównej.</li> <li>2. Wyświetlona zostaje lista ostatnich dziesięciu dat logowań w aplikacji.</li> <li>3. Lekarz/Rejestrator może wybrać opcję wyświetlenia ostatnich dwudziestu lub czterdziestu dat logowań w aplikacji.</li> </ol> |
| <b>Ścieżka alternatywna</b> |   |
| <b>Warunki końcowe</b>      | Wyświetlona została lista dat ostatnich logowań.  |

*Tab. 3. PU-3 – Wyświetlenie listy dat ostatnich logowań*

|                             |  |
|-----------------------------|--|
| <b>Kod przypadku użycia</b> | PU-4   |
| <b>Nazwa</b>                | Wyświetlenie listy lekarzy   |
| <b>Aktor</b>                | Lekarz/Rejestrator   |
| <b>Warunki początkowe</b>   | Lekarz/Rejestrator zalogował się do aplikacji  |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Lekarz/Rejestrator korzysta z zakładki „Lista lekarzy”, znajdującej się na stronie głównej.</li> <li>2. Wyświetlona zostaje lista lekarzy, domyślnie posortowana wg numeru ID Lekarza rosnąco.</li> <li>3. Lekarz/Rejestrator może wybrać opcję sortowania listy wg numeru ID Lekarza malejąco.</li> </ol> |
| <b>Ścieżka alternatywna</b> |  |
| <b>Warunki końcowe</b>      | Wyświetlona została lista lekarzy znajdujących się w bazie danych.   |

*Tab. 4. PU-4 – Wyświetlenie listy lekarzy*



|                             |  |
|-----------------------------|--|
| <b>Kod przypadku użycia</b> | PU-5   |
| <b>Nazwa</b>                | Wyświetlenie danych osobowych i kontaktowych użytkownika   |
| <b>Aktor</b>                | Lekarz/Rejestrator   |
| <b>Warunki początkowe</b>   | Lekarz/Rejestrator zalogował się do aplikacji.   |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Lekarz/Rejestrator korzysta z zakładki „Wyświetl i zmień swoje dane”, znajdującej się na stronie głównej.</li> <li>2. Wyświetlone zostają dane osobowe i kontaktowe Lekarza/Rejestratora, wraz z datą ostatniej zmiany danych oraz przyciskiem umożliwiającym rozpoczęcie edycji danych kontaktowych.</li> </ol> |
| <b>Ścieżka alternatywna</b> |  |
| <b>Warunki końcowe</b>      | Wyświetlone zostały dane osobowe i kontaktowe Lekarza/Rejestratora.  |

*Tab. 5. PU-5 – Wyświetlenie danych osobowych i kontaktowych użytkownika*

|                             |  |
|-----------------------------|--|
| <b>Kod przypadku użycia</b> | PU-6   |
| <b>Nazwa</b>                | Zmiana danych kontaktowych użytkownika   |
| <b>Aktor</b>                | Lekarz/Rejestrator   |
| <b>Warunki początkowe</b>   | <ul style="list-style-type: none"> <li>• Lekarz/Rejestrator zalogował się do aplikacji,</li> <li>• Lekarz/Rejestrator znajduje się na stronie wyświetlającej jego dane.</li> </ul>   |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Lekarz/Rejestrator klika przycisk „Chcę zmienić dane”, umożliwiający edycję danych kontaktowych.</li> <li>2. Po dokonaniu zmian, Lekarz/Rejestrator potwierdza je, klikając przycisk „Zmień dane”.</li> <li>3. Zmiany zostają zapisane w bazie danych, a na stronie głównej pojawia się odpowiedni komunikat.</li> </ol>   |
| <b>Ścieżka alternatywna</b> | <ol style="list-style-type: none"> <li>3a. W przypadku niewypełnienia wszystkich obowiązkowych pól podczas edycji danych, bądź gdy pole zostało zmienione w błędnym formacie, po kliknięciu przycisku „Zmień dane” pojawia się odpowiedni komunikat.</li> <li>3b. W przypadku niepowodzenia zmiany danych w bazie danych, na stronie z danymi Lekarza/Rejestratora pojawia się komunikat o błędzie.</li> </ol> |
| <b>Warunki końcowe</b>      | Dane kontaktowe Lekarza/Rejestratora zostały zmienione.  |

*Tab. 6. PU-6 – Zmiana danych kontaktowych użytkownika*

|                             |   |
|-----------------------------|---|
| <b>Kod przypadku użycia</b> | PU-7  |
| <b>Nazwa</b>                | Zmiana hasła użytkownika  |
| <b>Aktor</b>                | Lekarz/Rejestrator  |
| <b>Warunki początkowe</b>   | Lekarz/Rejestrator zalogował się do aplikacji.  |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Lekarz/Rejestrator korzysta z zakładki „Zmień hasło”, znajdującej się na stronie głównej.</li> <li>2. Lekarz/Rejestrator wpisuje swoje obecne hasło oraz dwukrotnie nowe hasło. Nowe hasło musi spełnić wymagania określone na stronie.</li> <li>3. Lekarz/Rejestrator potwierdza zmianę hasła, klikając przycisk „Zmień hasło”.</li> <li>4. Nowe hasło zostaje zapisane w bazie danych, a na stronie głównej pojawia się odpowiedni komunikat.</li> </ol>  |
| <b>Ścieżka alternatywna</b> | <ol style="list-style-type: none"> <li>1a. Po uzupełnieniu danych przez Lekarza/Rejestratora, zgodnie z PU-2, wyświetla się strona umożliwiająca zmianę hasła.</li> <li>3a. W przypadku niespełnienia wymagań przez nowe hasło, po kliknięciu przycisku „Zmień hasło” pojawia się odpowiedni komunikat.</li> <li>4a. W przypadku błędnego podania obecnego hasła lub gdy nowe hasło jest różne od hasła wpisanego w polu „Potwierdź nowe hasło”, na stronie do zmiany hasła pojawia się odpowiedni komunikat.</li> <li>4b. W przypadku niepowodzenia zmiany hasła użytkownika w bazie danych, na stronie do zmiany hasła wyświetla się odpowiedni komunikat.</li> </ol> |
| <b>Warunki końcowe</b>      | Hasło Lekarza/Rejestratora zostało zmienione.   |

*Tab. 7. PU-7 – Zmiana hasła użytkownika*

|                             |  |
|-----------------------------|--|
| <b>Kod przypadku użycia</b> | PU-8   |
| <b>Nazwa</b>                | Zarejestrowanie nowego pacjenta  |
| <b>Aktor</b>                | Rejestrator  |
| <b>Warunki początkowe</b>   | Rejestrator zalogował się do aplikacji.  |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Rejestrator korzysta z zakładki „Dodaj nowego pacjenta”, znajdującej się na stronie głównej.</li> <li>2. Rejestrator wypełnia formularz z danymi osobowymi i kontaktowymi nowego pacjenta.</li> <li>3. Rejestrator potwierdza chęć zarejestrowania nowego pacjenta, klikając przycisk „Dodaj pacjenta”.</li> <li>4. Nowy pacjent zostaje zapisany w bazie danych oraz pojawia się na liście zarejestrowanych pacjentów. Wyświetla się również odpowiedni komunikat.</li> </ol> |
| <b>Ścieżka alternatywna</b> | <ol style="list-style-type: none"> <li>4a. W przypadku niewypełnienia wszystkich obowiązkowych pól w formularzu, bądź gdy pole zostało uzupełnione w błędnym formacie, po kliknięciu przycisku „Dodaj pacjenta” pojawia się odpowiedni komunikat.</li> <li>4b. W przypadku niepowodzenia zapisania nowego pacjenta w bazie danych, na stronie przeznaczonej do dodania nowego pacjenta pojawia się odpowiedni komunikat.</li> </ol>  |
| <b>Warunki końcowe</b>      | Nowy pacjent został zarejestrowany w bazie danych.   |

*Tab. 8. PU-8 – Zarejestrowanie nowego pacjenta*

|                             |  |
|-----------------------------|--|
| <b>Kod przypadku użycia</b> | PU-9   |
| <b>Nazwa</b>                | Przeglądanie listy zarejestrowanych pacjentów  |
| <b>Aktor</b>                | Rejestrator  |
| <b>Warunki początkowe</b>   | <ul style="list-style-type: none"> <li>• Rejestrator zalogował się do aplikacji,</li> <li>• W bazie danych znajduje się co najmniej jeden zarejestrowany pacjent.</li> </ul>   |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Rejestrator korzysta z zakładki „Lista pacjentów”, znajdującej się na stronie głównej.</li> <li>2. Wyświetlona zostaje domyślnie lista wszystkich zarejestrowanych pacjentów.</li> </ol> |
| <b>Ścieżka alternatywna</b> |  |
| <b>Warunki końcowe</b>      | Lista zarejestrowanych pacjentów została wyświetlona.  |

*Tab. 9. PU-9 – Przeglądanie listy zarejestrowanych pacjentów*

|                             |  |
|-----------------------------|--|
| <b>Kod przypadku użycia</b> | PU-10  |
| <b>Nazwa</b>                | Wyszukiwanie pacjentów po numerze ID, PESELu lub nazwisku  |
| <b>Aktor</b>                | Rejestrator  |
| <b>Warunki początkowe</b>   | <ul style="list-style-type: none"> <li>• Rejestrator zalogował się do aplikacji,</li> <li>• Rejestrator znajduje się na stronie z listą zarejestrowanych pacjentów,</li> <li>• W bazie danych znajduje się co najmniej jeden zarejestrowany pacjent.</li> </ul>  |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Rejestrator korzysta z listy rozwijanej z dostępnymi opcjami wyszukiwania – ID, PESEL, nazwisko.</li> <li>2. Po wybraniu odpowiedniej opcji, Rejestrator wpisuje w polu obok wyszukiwaną wartość, zgodną z formatem wybranej opcji wyszukiwania i klika przycisk „Szukaj pacjenta”.</li> <li>3. Pacjent o podanym numerze ID, PESELu lub nazwisku wyświetla się na stronie.</li> </ol> |
| <b>Ścieżka alternatywna</b> | 3a. Jeżeli żaden z pacjentów nie zawiera w swoich danych wyszukiwanej wartości, na stronie pojawia się odpowiedni komunikat.   |
| <b>Warunki końcowe</b>      | Wyświetlony został pacjent z wyszukiwaną wartością.  |

*Tab. 10. PU-10 – Wyszukiwanie pacjentów na liście po numerze ID, PESELu lub nazwisku*

|                             |   |
|-----------------------------|---|
| <b>Kod przypadku użycia</b> | PU-11   |
| <b>Nazwa</b>                | Dodanie nowego pobytu   |
| <b>Aktor</b>                | Lekarz  |
| <b>Warunki początkowe</b>   | <ul style="list-style-type: none"> <li>• Lekarz zalogował się do aplikacji,</li> <li>• Pacjent, którego dotyczy pobyt, został wcześniej zarejestrowany.</li> </ul>  |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Lekarz korzysta z zakładki „Dodaj nowy pobyt”, znajdującej się na stronie głównej.</li> <li>2. Lekarz wpisuje ID pacjenta, którego dotyczy pobyt i klika przycisk „Znajdź pacjenta”.</li> <li>3. Na stronie pojawiają się podstawowe zgromadzone informacje o pacjencie o podanym numerze ID. Lekarz wybiera oddział szpitalny w którym przebywa pacjent, schorzenie pacjenta oraz może dodać początkowe informacje nt. pobytu w formie tekstu.</li> <li>4. Lekarz potwierdza wprowadzone informacje, klikając przycisk „Dodaj pobyt”.</li> <li>5. Na stronie głównej pojawia się komunikat o dodaniu nowego pobytu pacjenta, wraz z numerami ID pobytu i pacjenta. Pacjent pojawia się także na liście prowadzonych pacjentów dostępnej na stronie głównej.</li> </ol> |
| <b>Ścieżka alternatywna</b> | <ol style="list-style-type: none"> <li>5a. W przypadku nieokreślenia oddziału szpitalnego pacjenta i/lub jego schorzenia, na stronie przeznaczonej do dodania nowego pobytu wyświetla się odpowiedni komunikat.</li> <li>5b. W przypadku niepowodzenia zapisania nowego pobytu w bazie danych, na stronie przeznaczonej do dodania nowego pobytu wyświetla się odpowiedni komunikat.</li> </ol>   |
| <b>Warunki końcowe</b>      | Pobyt prowadzonego pacjenta został zapisany w bazie danych.   |

*Tab. 11. PU-11 – Dodanie nowego pobytu*

|                             |   |
|-----------------------------|---|
| <b>Kod przypadku użycia</b> | PU-12   |
| <b>Nazwa</b>                | Wyświetlenie informacji o pobycie pacjenta  |
| <b>Aktor</b>                | Lekarz  |
| <b>Warunki początkowe</b>   | <ul style="list-style-type: none"> <li>• Lekarz zalogował się do aplikacji,</li> <li>• Lekarz dodał wcześniej pobyt pacjenta.</li> </ul>  |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Lekarz korzysta z opcji „Wyświetl informacje o pobycie pacjenta”, znajdującej się na liście prowadzonych pacjentów na stronie głównej.</li> <li>2. Wyświetlone zostają informacje o pobycie pacjenta, w tym historia pobytu.</li> </ol> |
| <b>Ścieżka alternatywna</b> |   |
| <b>Warunki końcowe</b>      | Wyświetlone zostały informacje o pobycie pacjenta.  |

*Tab. 12. PU-12 – Wyświetlenie informacji o pobycie pacjenta*

|                             |   |
|-----------------------------|---|
| <b>Kod przypadku użycia</b> | PU-13   |
| <b>Nazwa</b>                | Aktualizacja informacji o pobycie pacjenta  |
| <b>Aktor</b>                | Lekarz  |
| <b>Warunki początkowe</b>   | <ul style="list-style-type: none"> <li>• Lekarz zalogował się do aplikacji,</li> <li>• Lekarz dodał wcześniej pobyt pacjenta,</li> <li>• Lekarz znajduje się na stronie z informacjami o pobycie.</li> </ul>  |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Lekarz klika przycisk „Aktualizuj informacje o pobycie”.</li> <li>2. Na stronie pojawia się formularz, w którym Lekarz może określić czy chce zmienić oddział szpitalny, w którym znajduje się prowadzony pacjent i/lub schorzenie pacjenta oraz w którym znajduje się pole tekstowe umożliwiające dodanie nowych informacji, uwag o pobycie w formie opisowej.</li> <li>3. Lekarz potwierdza wprowadzone zmiany i/lub dołączony opis, klikając przycisk „Aktualizuj pobyt”.</li> <li>4. Nowe informacje zostają dołączone do historii pobytu, wraz z odnotowaniem ewentualnych zmiany w oddziale szpitalnym prowadzonego pacjenta i/lub schorzenia pacjenta. Na stronie głównej wyświetla się odpowiedni komunikat.</li> </ol> |
| <b>Ścieżka alternatywna</b> | <ol style="list-style-type: none"> <li>4a. W przypadku gdy Lekarz zadeklarował chęć zmiany oddziału szpitalnego, w którym przebywa prowadzony pacjent i/lub schorzenia pacjenta, a nie dokonał wyboru z rozwijanej listy, na stronie pobytu pojawia się odpowiedni komunikat.</li> <li>5a. W przypadku niepowodzenia aktualizacji informacji o pobycie, na stronie pobytu pojawia się odpowiedni komunikat.</li> </ol>  |
| <b>Warunki końcowe</b>      | Informacje o pobycie pacjenta zostały zaktualizowane.   |

*Tab. 13. PU-13 – Aktualizacja informacji o pobycie pacjenta*

|                             |   |
|-----------------------------|---|
| <b>Kod przypadku użycia</b> | PU-14   |
| <b>Nazwa</b>                | Wypisanie pacjenta ze szpitala  |
| <b>Aktor</b>                | Lekarz  |
| <b>Warunki początkowe</b>   | <ul style="list-style-type: none"> <li>• Lekarz zalogował się do aplikacji,</li> <li>• Lekarz dodał wcześniej pobyt pacjenta,</li> <li>• Lekarz znajduje się na stronie z informacjami o pobycie.</li> </ul>  |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Lekarz korzysta z opcji „kliknij tutaj”, oznaczającej przejście do strony, na której potwierdza się wypis pacjenta ze szpitala.</li> <li>2. Wyświetlone zostają podstawowe dane pacjenta, wraz z polem tekstowym umożliwiającym podsumowanie pobytu pacjenta w formie opisu.</li> <li>3. Lekarz potwierdza chęć dokonania wypisu, klikając przycisk „Wypisz pacjenta”.</li> <li>4. Potwierdzenie wypisu pacjenta pojawia się w formie odpowiedniego komunikatu na stronie głównej.</li> </ol> |
| <b>Ścieżka alternatywna</b> | 4a. W przypadku niepowodzenia wypisu pacjenta, na stronie, na której potwierdza się wypis pacjenta, pojawia się odpowiedni komunikat.   |
| <b>Warunki końcowe</b>      | Pacjent został wypisany ze szpitala oraz wygenerowany został akt wypisu pacjenta.   |

*Tab. 14. PU-14 – Aktualizacja informacji o pobycie pacjenta*



|                             |   |
|-----------------------------|---|
| <b>Kod przypadku użycia</b> | PU-15   |
| <b>Nazwa</b>                | Wyświetlenie aktu wypisu pacjenta   |
| <b>Aktor</b>                | Lekarz  |
| <b>Warunki początkowe</b>   | <ul style="list-style-type: none"> <li>• Lekarz zalogował się do aplikacji,</li> <li>• Lekarz wypisał co najmniej jednego pacjenta ze szpitala maksymalnie 48 godzin wcześniej.</li> </ul>  |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Lekarz korzysta z zakładki „Ostatnio zakończone pobyty”, znajdującej się na stronie głównej.</li> <li>2. Wyświetlone zostają, w formie listy, wszystkie pobyty pacjentów zakończone przez Lekarza w ostatnich 48 godzinach.</li> <li>3. Lekarz korzysta z opcji „Wyświetl akt wypisu pacjenta”, znajdującej się na ww. liście.</li> <li>4. Wyświetlony zostaje akt wypisu pacjenta w formie pliku .pdf, zawierający podstawowe dane pacjenta oraz informacje o jego zakończonym pobycie.</li> </ol> |
| <b>Ścieżka alternatywna</b> | 4a. W przypadku niepowodzenia wyświetlenia aktu wypisu pacjenta, na stronie z listą aktów wypisu pojawia się odpowiedni komunikat.  |
| <b>Warunki końcowe</b>      | Akt wypisu pacjenta został wyświetlony.   |

*Tab. 15. PU-15 – Wyświetlenie aktu wypisu pacjenta*

|                             |   |
|-----------------------------|---|
| <b>Kod przypadku użycia</b> | PU-16   |
| <b>Nazwa</b>                | Wyświetlenie danych osobowych i kontaktowych pacjenta   |
| <b>Aktor</b>                | Lekarz/Rejestrator  |
| <b>Warunki początkowe</b>   | <ul style="list-style-type: none"> <li>• Lekarz/Rejestrator zalogował się do aplikacji,</li> <li>• Lekarz musi być lekarzem prowadzącym pacjenta,</li> <li>• Rejestrator znajduje się na stronie z listą wszystkich zarejestrowanych pacjentów lub skorzystał z opcji wyszukania pacjenta.</li> </ul>   |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Lekarz/Rejestrator korzysta z opcji „Wyświetl i zmień dane pacjenta”: <ol style="list-style-type: none"> <li>a. W przypadku Lekarza opcja ta znajduje się na liście prowadzonych pacjentów dostępnej ze strony głównej,</li> <li>b. W przypadku Rejestratora opcja ta znajduje się na liście wszystkich zarejestrowanych pacjentów oraz na liście z wyszukanym(i) pacjentem(ami).</li> </ol> </li> <li>2. Wyświetlone zostają dane osobowe i kontaktowe pacjenta, wraz z przyciskiem umożliwiającym rozpoczęcie edycji danych kontaktowych oraz datą ostatniej zmiany danych pacjenta.</li> </ol> |
| <b>Ścieżka alternatywna</b> | 2a. Lekarz ma możliwość wyświetlenia historii medycznej pacjenta, który w przeszłości przebywał już w szpitalu.   |
| <b>Warunki końcowe</b>      | Wyświetlone zostały dane osobowe i kontaktowe pacjenta.   |

*Tab. 16. PU-16 – Wyświetlenie danych osobowych i kontaktowych pacjenta*

|                             |  |
|-----------------------------|--|
| <b>Kod przypadku użycia</b> | PU-17  |
| <b>Nazwa</b>                | Zmiana danych kontaktowych pacjenta  |
| <b>Aktor</b>                | Lekarz/Rejestrator   |
| <b>Warunki początkowe</b>   | <ul style="list-style-type: none"> <li>• Lekarz/Rejestrator zalogował się do aplikacji</li> <li>• Lekarz/Rejestrator skorzystał już z opcji „Wyświetl i zmień dane pacjenta”</li> </ul>  |
| <b>Ścieżka główna</b>       | <ol style="list-style-type: none"> <li>1. Lekarz/Rejestrator klika przycisk „Chcę zmienić dane pacjenta”, umożliwiający edycję danych kontaktowych.</li> <li>2. Po dokonaniu zmian, Lekarz/Rejestrator potwierdza je, klikając przycisk „Zmień dane pacjenta”.</li> <li>3. Zmiany zostają zapisane w bazie danych, a odpowiedni komunikat pojawia się: <ol style="list-style-type: none"> <li>a. na stronie głównej w przypadku Lekarza,</li> <li>b. na stronie z listą pacjentów w przypadku Rejestratora.</li> </ol> </li> </ol> |
| <b>Ścieżka alternatywna</b> | 3a. W przypadku niepowodzenia zmiany danych, na stronie z danymi pacjenta pojawia się odpowiedni komunikat.  |
| <b>Warunki końcowe</b>      | Dane kontaktowe pacjenta zostały zmienione.  |

*Tab. 17. PU-17 – Zmiana danych kontaktowych pacjenta*

## 4. Bezpieczeństwo serwera szpitalnego

W niniejszym rozdziale przedstawione zostaną różne zaimplementowane rozwiązania, wpływające na zwiększenie bezpieczeństwa pracy i bezpieczeństwa korzystania z najważniejszych komponentów tworzących oprogramowanie serwera – systemu operacyjnego Debian, serwera HTTP Apache i systemu zarządzania bazą danych MariaDB.

### 4.1 Bezpieczeństwo systemu operacyjnego Debian

Obsługę dostępu z zewnątrz do szpitalnego serwera zapewnia firewall, który jest bardzo ważną częścią systemu operacyjnego Debian. Bezpieczeństwo dostępu do informacji przechowywanych wewnątrz systemu zapewniane jest zaś przez rozsądne zarządzanie użytkownikami systemu oraz umiejętne (nie)przydzielanie tym użytkownikom uprawnień do określonych, ważnych plików.

#### 4.1.1 Użytkownicy systemu operacyjnego i rodzaje uprawnień do plików

Administrator (osoba zarządzająca serwerem szpitalnym) może pracować na systemie operacyjnym Debian jako użytkownik *user1* lub *root*. Użytkownik *user1* został utworzony przy instalacji systemu operacyjnego Debian, podczas gdy *root* jest domyślnie zaimplementowanym użytkownikiem systemów operacyjnych Linux takich jak Debian. Użytkownik *root* może wywołać każdą komendę w systemie i może przeglądać, edytować czy uruchomić każdy plik zapisany w systemie, bez względu na jego właściciela.

Ze względu na nieograniczone uprawnienia użytkownika *root*, powinien być on używany tylko w koniecznych do tego sytuacjach. Aby jednak możliwe było sprawne administrowanie systemem operacyjnym, użytkownik *user1* został dodany do grupy *sudo*. Przynależność do tej grupy użytkownika *user1* umożliwia mu wykonywanie komend z uprawnieniami użytkownika *root*. W tym celu, każdą taką komendę *user1* musi rozpocząć od wyrazu *sudo*. Po pierwszym użyciu *sudo* podczas każdej sesji terminala, użytkownik *user1* musi również potwierdzić swoją tożsamość przez wpisanie hasła.

W systemach opartych na jądrze Linux, w tym również w systemie Debian, występują trzy rodzaje uprawnień do pliku (bądź folderu) [23]:

- read (r) – umożliwia odczyt pliku
- write (w) – umożliwia zapis pliku
- execute (x) – umożliwia wykonanie pliku (otwarcie folderu)

Odmienne uprawnienia do pliku mogą zostać przydzielone właścicielowi pliku, grupie użytkowników oraz wszystkim pozostałym użytkownikom systemu.

```
-rw-r----- 1 root user1 13 12-07 16:24 testowy_plik.txt
```

*Rys. 9. Uprawnienia właściciela i grupy do pliku testowy\_plik.txt (źródło własne)*

Opierając się na przykładzie zawartym na rys. 9, użytkownik *root* jest właścicielem pliku. Właściciel może wyświetlić zawartość pliku (r) oraz zapisać wprowadzone zmiany (w). Grupa *user1*, którą tworzy użytkownik *user1* może tylko wyświetlić zawartość pliku (r), bez możliwości jej zmiany. Użytkownik *user1* może natomiast zmodyfikować i zapisać plik w momencie, gdy do jego otwarcia wykorzysta polecenie *sudo*. Wówczas *user1* otwiera plik *testowy\_plik.txt* z takimi samymi uprawnieniami jak *root*. Pozostali użytkownicy systemu nie posiadają żadnych uprawnień do tego pliku. Dzięki zdefiniowaniu właściciela pliku i grupy użytkowników, która posiada pierwszeństwo dostępu do pliku (folderu), możliwe jest ograniczenie ryzyka uzyskania przez nieuprawnione osoby dostępu do ważnych informacji.

#### 4.1.2 Firewall

Jak wspomniano na początku tego podrozdziału, za bezpieczeństwo dostępu z zewnątrz do serwera odpowiada firewall, będący częścią systemu operacyjnego Debian. Konfiguracja firewalla została zaprezentowana na rys. 10.

```
user1@debian:~$ sudo ufw status numbered
Status: active

      To      Action      From
      --      -
[ 1] 80      ALLOW IN    192.168.56.1
[ 2] 443     ALLOW IN    192.168.56.1
[ 3] 80      ALLOW IN    192.168.0.0/21
[ 4] 443     ALLOW IN    192.168.0.0/21
[ 5] 80      ALLOW IN    192.168.12.0/24
[ 6] 443     ALLOW IN    192.168.12.0/24

user1@debian:~$ █
```

Rys. 10. Konfiguracja serwerowego firewalla (źródło własne)

Pula adresów IP VLANu, do którego podłączone są komputery lekarzy to: 192.168.0.0/21

Pula adresów IP VLANu, do którego podłączone są komputery rejestratorów to:  
192.168.12.0/24

**Uwaga:**

*Adres IP 192.168.56.1 to adres komputera-hosta. Na potrzeby wykonania i testowania aplikacji webowej ustanowiona została lokalna sieć pomiędzy komputerem-hostem, a maszyną wirtualną, na której znajduje się serwer. Konfiguracja firewalla lub serwera HTTP Apache związana z adresem IP 192.168.56.1 nie jest przedmiotem rozważań w pracy dyplomowej.*

*Pule adresów IP VLANów wynikają z systemu sieciowego placówki szpitalnej, który opisany został w pracy dyplomowej „System zabezpieczeń teleinformatycznych dla placówki szpitalnej - sprzętowa struktura sieci” autorstwa Jakuba Słoty.*

Komputery o adresie IP należącym do jednego z wyżej wymienionych zakresów mogą komunikować się z serwerem przez porty 80 i 443. Są to porty serwera HTTP Apache. Port 80 odpowiada za komunikację z wykorzystaniem protokołu HTTP, a port 443 za komunikację z wykorzystaniem bezpieczniejszej wersji HTTP, czyli protokołu HTTPS. Otwarcie portów 80 i 443 dla tych komputerów oznacza, że lekarze i rejestratorzy mogą korzystać z aplikacji webowej znajdującej się na serwerze Apache.

## 4.2 Bezpieczeństwo serwera HTTP Apache

Obecny podrozdział porusza kwestie dotyczące zapewnienia bezpiecznej komunikacji z serwerem HTTP Apache, zablokowania dostępu do wybranych zasobów przechowywanych na serwerze Apache czy wykonywania kopii zapasowych określonych plików. Poruszona zostanie również kwestia logów Apache.

### 4.2.1 Protokół HTTPS

Niezbędnym w dzisiejszych czasach rozwiązaniem, wpływającym na zwiększenie bezpieczeństwa przesyłanych danych pomiędzy przeglądarką internetową, a serwerem HTTP (komunikacja klient-serwer) jest protokół HTTPS. Ustanowienie komunikacji klient-serwer za pośrednictwem protokołu HTTPS umożliwia szyfrowanie żądań HTTP i informacji w nich zawartych, tym samym zmniejszając ryzyko przechwycenia wrażliwych danych w przypadku nasłuchiwanie komunikacji przez niepożądaną osobę.

Aby serwer HTTP Apache mógł obsługiwać komunikację za pośrednictwem HTTPS, konieczne jest uzyskanie certyfikatu SSL dla serwera. Taki certyfikat można w prosty sposób wygenerować dzięki pakietowi *openssl*, który można pobrać na system operacyjny Debian. Wygenerowany w ten sposób certyfikat SSL jest certyfikatem „samodzielnie podpisanym”, a nie podpisanym przez zaufaną, uprawnioną instytucję, co może skutkować nieuznaniem takiego certyfikatu przez przeglądarkę internetową. Jednakże nie zmienia to faktu, że po załączeniu do odpowiedniego pliku konfiguracyjnego serwera Apache ścieżki dostępu do certyfikatu SSL (patrz: rys. 11), szyfrowanie komunikacji klient-serwer staje się możliwe.

Portem przeznaczonym do komunikacji za pośrednictwem protokołu HTTPS jest port 443. Podczas konfiguracji serwera Apache ustanowiono również przekierowywanie (*redirect*) żądań HTTP (port 80) na bezpieczny odpowiednik HTTPS (port 443, patrz: rys. 11), tym samym zapewniając komunikację klient-serwer z wykorzystaniem portu 443.

Proces generowania certyfikatu SSL oraz konfiguracji serwera Apache pod obsługę HTTPS oparty został na poradniku dostępnym na stronie [www.linuxhint.com](http://www.linuxhint.com) [24]. Podstawowe (ogólne) informacje o wygenerowanym certyfikacie SSL znajdują się na rys. 12.

```
GNU nano 7.2                                000-default.conf
<VirtualHost *:80>
    ServerName 192.168.56.101
    Redirect permanent / https://192.168.56.101/
</VirtualHost>
<VirtualHost *:443>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    ServerName 192.168.56.101

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn

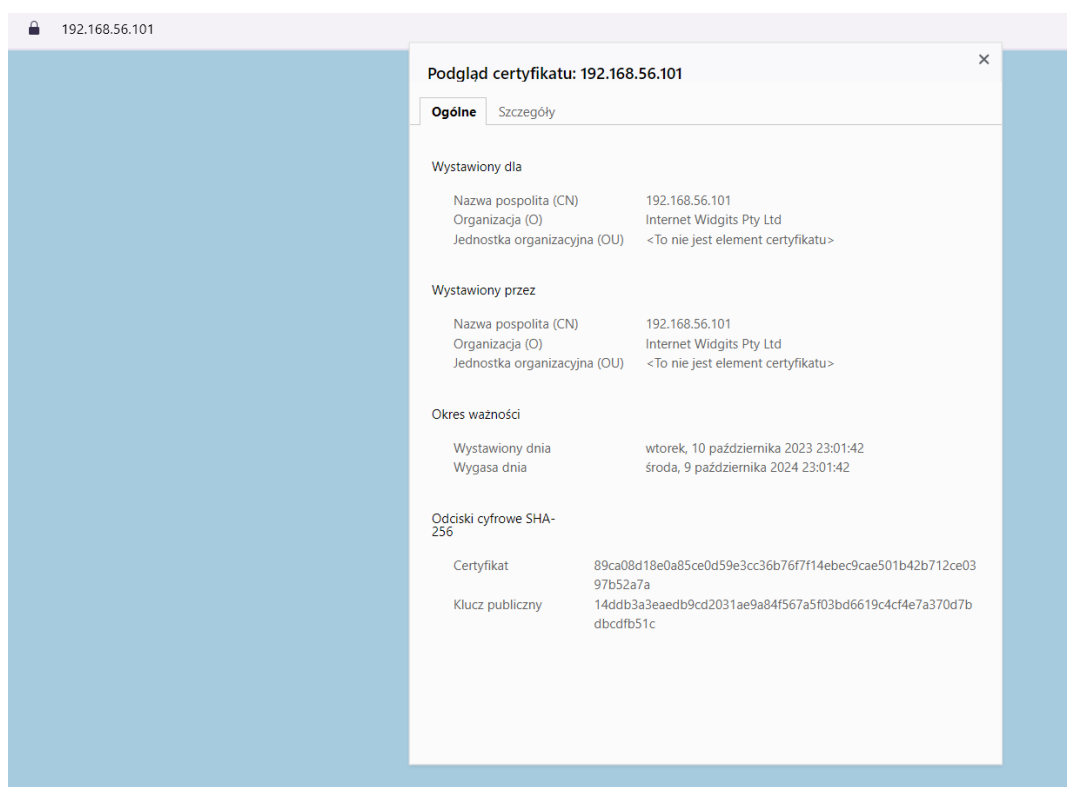
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    SSLEngine on
    SSLCertificateFile /etc/apache2/certificate/apache-certificate.crt
    SSLCertificateKeyFile /etc/apache2/certificate/apache.key
```

*Rys. 11. Konfiguracja serwera http Apache do obsługi protokołu HTTPS (źródło własne)*

**Uwaga:**

*Adres IP 192.168.56.101, widoczny na rys. 11 oraz rys. 12, to adres IP maszyny wirtualnej, na której znajduje się serwer. Adres IP 192.168.56.101 nie jest adresem IP serwera w systemie sieciowym placówki szpitalnej, który opisany został w pracy dyplomowej „System zabezpieczeń teleinformatycznych dla placówki szpitalnej - sprzętowa struktura sieci” autorstwa Jakuba Słoty.*





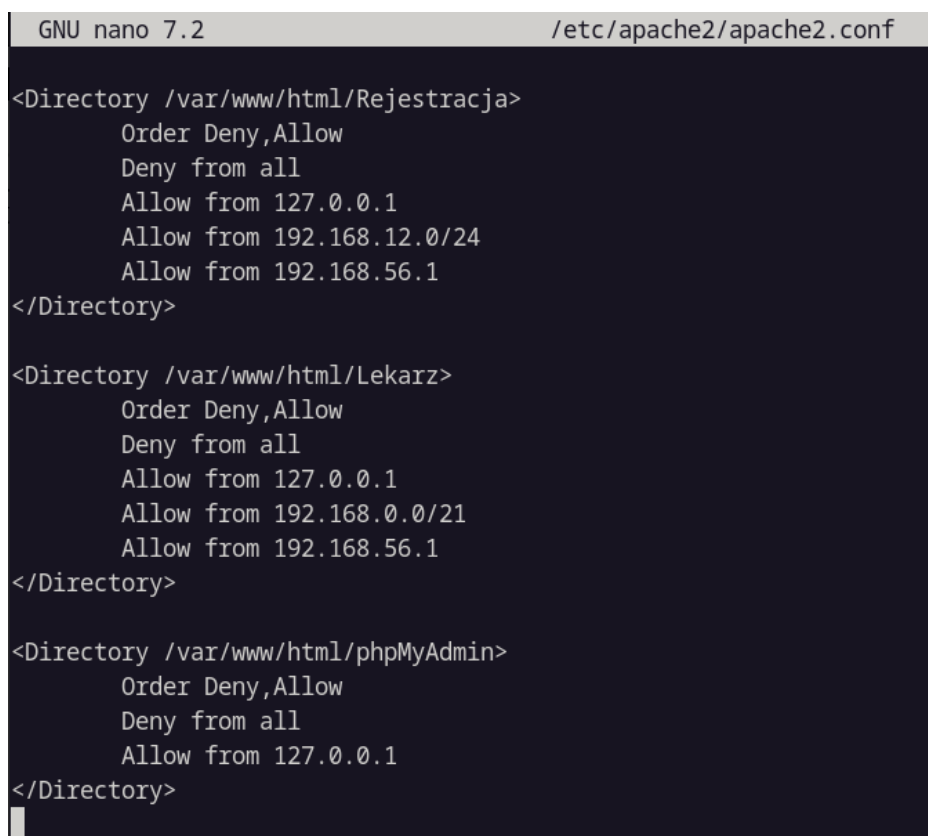
Rys. 12. Ogólne informacje o wygenerowanym certyfikacie SSL, dostępne z przeglądarki internetowej OperaGX (źródło własne)

#### 4.2.2 Blokada dostępu do folderów

Pliki aplikacji webowej są przechowywane w określonym w konfiguracji serwera Apache folderze, do którego zasobów (plików i folderów) ma dostęp każdy uprawniony użytkownik sieci (ang. *Document root*). Takim folderem domyślnie jest folder *html* o ścieżce dostępu */var/www/html*. W rozdziale 4.1.2 dotyczącym konfiguracji systemowego firewalla, określani zostali użytkownicy, a raczej zakresy adresów IP urządzeń (komputerów), które mogą komunikować się z serwerem HTTP Apache przez porty 80 i 443. Jako, że aplikacja webowa dzieli się w swojej strukturze na dwa foldery: *Lekarz* i *Rejestracja*, odpowiadające dwóm przewidzianym grupom użytkowników aplikacji, wprowadzone zostały blokady dostępu do tych folderów. Idea blokad jest prosta. Komputery z adresami IP wykorzystywanymi przez lekarzy nie mają dostępu do plików aplikacji przechowywanych w folderze *Rejestracja*. I odwrotnie, komputery z adresami IP wykorzystywanymi przez rejestratorów nie mają dostępu do plików aplikacji przechowywanych w folderze *Lekarz*. Celem blokad jest zmniejszenie ryzyka uzyskania dostępu do wrażliwych danych, np. danych medycznych pacjentów przez nieuprawnione

osoby. Takie ryzyko mogłoby powstać w sytuacji, gdyby lekarz zalogował się do aplikacji na komputerze przeznaczonym dla rejestratorów i zapomniał się następnie wylogować. Przez taką nieostrożność rejestrator mógłby przypadkowo uzyskać dostęp np. do historii pobytu któregoś z pacjentów. Blokadą dostępu objęty został również folder *phpMyAdmin*, który przechowuje pliki strony phpMyAdmin, umożliwiające zarządzanie bazą danych z poziomu przeglądarki internetowej. W tym przypadku, jedynie adres IP 127.0.0.1 (adres lokalny serwera) ma przyznany dostęp do folderu *phpMyAdmin*.

Ustawienie blokad do folderów odbywa się za pomocą zestawu reguł *order deny* (dostęp zabroniony), *allow* (dostęp przyznany), w głównym pliku konfiguracyjnym serwera HTTP Apache. W tym przypadku wystarczy określić adresy IP którym przyznaje się (*allow*) dostęp do danego folderu. Pozostałym, nieokreślonym adresom IP dostęp do folderu jest zabroniony (przez regułę *deny from all*). Stworzony zestaw reguł *order allow, deny* został przedstawiony na rys. 13.



```
GNU nano 7.2 /etc/apache2/apache2.conf

<Directory /var/www/html/Rejestracja>
    Order Deny,Allow
    Deny from all
    Allow from 127.0.0.1
    Allow from 192.168.12.0/24
    Allow from 192.168.56.1
</Directory>

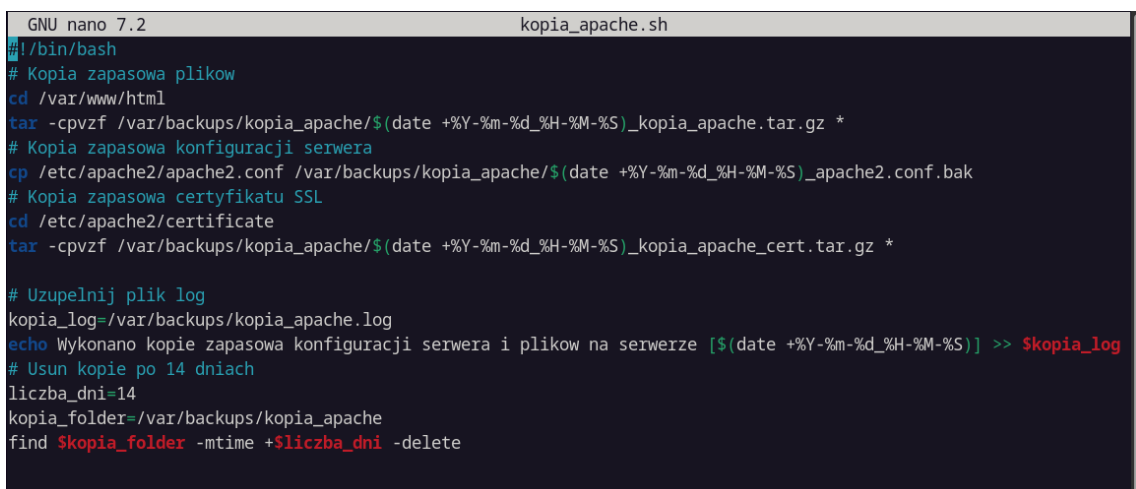
<Directory /var/www/html/Lekarz>
    Order Deny,Allow
    Deny from all
    Allow from 127.0.0.1
    Allow from 192.168.0.0/21
    Allow from 192.168.56.1
</Directory>

<Directory /var/www/html/phpMyAdmin>
    Order Deny,Allow
    Deny from all
    Allow from 127.0.0.1
</Directory>
```

Rys. 13. Reguły *order deny, allow* w głównym pliku konfiguracyjnym Apache (źródło własne)

#### 4.2.3 Automatyczna kopia zapasowa serwera oraz logi Apache

Dzięki programowi *crontab*, który umożliwia m.in. cykliczne uruchamianie plików z skryptem, codziennie o godzinie 23:00 tworzona jest w systemie Debian kopia zapasowa zawartości folderu *html* serwera Apache, głównego pliku konfiguracyjnego *apache2.conf* oraz folderu *certificate* zawierającego wygenerowany certyfikat SSL. Zapobiega to m.in. ryzyku utraty przechowywanej w folderze *html* aplikacji webowej. Kod skryptu *kopia\_apache.sh* znajduje się na rys. 14 i bazuje na skrypcie wykonującym kopię zapasową bazy danych „Szpital”, który przedstawiony zostanie w rozdziale 4.3.3. Skrypt po każdym wykonaniu kopii uzupełnia również plik *kopia\_apache.log* o odpowiedni wpis i usuwa starsze niż 14 dni kopie zapasowe.



```
GNU nano 7.2 kopia_apache.sh
#!/bin/bash
# Kopia zapasowa plikow
cd /var/www/html
tar -cpvzf /var/backups/kopia_apache/$(date +%Y-%m-%d_%H-%M-%S)_kopia_apache.tar.gz *
# Kopia zapasowa konfiguracji serwera
cp /etc/apache2/apache2.conf /var/backups/kopia_apache/$(date +%Y-%m-%d_%H-%M-%S)_apache2.conf.bak
# Kopia zapasowa certyfikatu SSL
cd /etc/apache2/certificate
tar -cpvzf /var/backups/kopia_apache/$(date +%Y-%m-%d_%H-%M-%S)_kopia_apache_cert.tar.gz *

# Uzupełnij plik log
kopia_log=/var/backups/kopia_apache.log
echo Wykonano kopie zapasowa konfiguracji serwera i plikow na serwerze [$(date +%Y-%m-%d_%H-%M-%S)] >> $kopia_log
# Usun kopie po 14 dniach
liczba_dni=14
kopia_folder=/var/backups/kopia_apache
find $kopia_folder -mtime +$liczba_dni -delete
```

Rys. 14. Kod pliku *kopia\_apache.sh* wykonującego kopie zapasowe plików związanych z serwerem Apache (źródło własne)

Na rys. 15 przedstawione zostały kopie zapasowe, wykonane o godzinie 23:00 w różnych dniach, potwierdzając tym samym poprawność działania skryptu.

```

root@debian:/var/backups/kopia_apache# ls -l
razem 78852
-rw-r----- 1 root root 13433887 11-24 23:00 2023-11-24_23-00-01_kopia_apache.tar.gz
-rw-r----- 1 root root      7351 11-24 23:00 2023-11-24_23-00-06_apache2.conf.bak
-rw-r----- 1 root root      9095 11-24 23:00 2023-11-24_23-00-06_kopia_apache_cert.tar.gz
-rw-r----- 1 root root 13433887 11-30 23:00 2023-11-30_23-00-01_kopia_apache.tar.gz
-rw-r----- 1 root root      7351 11-30 23:00 2023-11-30_23-00-05_apache2.conf.bak
-rw-r----- 1 root root      9095 11-30 23:00 2023-11-30_23-00-05_kopia_apache_cert.tar.gz
-rw-r----- 1 root root 13433887 12-03 23:00 2023-12-03_23-00-01_kopia_apache.tar.gz
-rw-r----- 1 root root      7351 12-03 23:00 2023-12-03_23-00-06_apache2.conf.bak
-rw-r----- 1 root root      9095 12-03 23:00 2023-12-03_23-00-06_kopia_apache_cert.tar.gz
-rw-r----- 1 root root 13435431 12-04 23:00 2023-12-04_23-00-01_kopia_apache.tar.gz
-rw-r----- 1 root root      7351 12-04 23:00 2023-12-04_23-00-05_apache2.conf.bak
-rw-r----- 1 root root      9095 12-04 23:00 2023-12-04_23-00-05_kopia_apache_cert.tar.gz
-rw-r----- 1 root root 13435431 12-06 23:00 2023-12-06_23-00-01_kopia_apache.tar.gz
-rw-r----- 1 root root      7351 12-06 23:00 2023-12-06_23-00-05_apache2.conf.bak
-rw-r----- 1 root root      9095 12-06 23:00 2023-12-06_23-00-05_kopia_apache_cert.tar.gz
-rw-r----- 1 root root 13435431 12-08 23:00 2023-12-08_23-00-01_kopia_apache.tar.gz
-rw-r----- 1 root root      7351 12-08 23:00 2023-12-08_23-00-05_apache2.conf.bak
-rw-r----- 1 root root      9100 12-08 23:00 2023-12-08_23-00-05_kopia_apache_cert.tar.gz
root@debian:/var/backups/kopia_apache#

```

*Rys. 15. Zawartość folderu w którym przechowywane są kopie zapasowe plików związanych z serwerem Apache (źródło własne)*

Logi serwera HTTP Apache zapisywane są domyślnie w plikach o ścieżce dostępu `/var/log/apache2`. Pliki te dzielą się na dwa typy: pliki `.log` z nazwą `error` oraz pliki `.log` z nazwą `access`. Pliki typu `error` przechowują logi z ostrzeżeniami i błędami, które wystąpiły w trakcie pracy serwera Apache. Pliki typu `access` zawierają natomiast logi z wszystkimi żądaniami ze strony przeglądarki internetowej. Oba typy plików `.log` pozwalają na analizę pracy serwera Apache oraz żądań przez niego obsługiwanych. Fragment przykładowego pliku `access.log` został pokazany na rys. 16.

```

192.168.56.1 - - [05/Dec/2023:00:00:22 +0100] "GET /Lekarz/lek-wyswietl-pobyt.php?id_pobytu=10047 HTTP/1.1" 200 3750 "https://192.168.56.1/01/Lekarz/lek-strona-glowna.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36 OPR/104.0.0.0"
192.168.56.1 - - [05/Dec/2023:00:00:24 +0100] "GET /Lekarz/lek-wyswietl-pobyt.php?id_pobytu=10048 HTTP/1.1" 302 370 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36 OPR/104.0.0.0"
192.168.56.1 - - [05/Dec/2023:00:00:24 +0100] "GET /Lekarz/lek-strona-glowna.php HTTP/1.1" 200 1225 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36 OPR/104.0.0.0"
192.168.56.1 - - [05/Dec/2023:00:00:47 +0100] "-" 408 319 "-" "-"
root@debian:/var/log/apache2#

```

*Rys. 16. Fragment jednego z plików `access.log` serwera HTTP Apache (źródło własne)*

### 4.3 Bezpieczeństwo systemu zarządzania bazą danych MariaDB

W tym podrozdziale skupiono się na przedstawieniu przyznanych uprawnień użytkownikom systemu MariaDB do operacji na danych przechowywanych w

poszczególnych tabelach i kolumnach bazy danych „Szpital”. Przedstawiono również formę zapisu większości danych przechowywanych w bazie. Podobnie jak w podrozdziale dotyczącym serwera HTTP Apache, ostatnia część podrozdziału przeznaczona jest wykonywaniu kopii zapasowej bazy danych „Szpital” oraz logom MariaDB.

#### 4.3.1 Uprawnienia przyznane użytkownikom systemu MariaDB

Zgodnie z rozdziałem 3.1.2, dostęp do bazy danych „Szpital” przyznany został pięciu utworzonym użytkownikom systemu MariaDB: *admin*, *lekarz*, *rejestracja*, *uwierzytelnienie* i *zmien\_haslo*. Użytkownik *admin*, posiadający wszystkie uprawnienia, wykorzystywany jest tylko podczas operacji na bazie podejmowanych przez Administratora z poziomu serwera. W celu komunikacji aplikacji z bazą danych wykorzystywani są więc pozostali użytkownicy.

Można stwierdzić, że użytkownicy MariaDB pełnią rolę *pośredników* w łączności aplikacji z bazą danych. Dzięki użytkownikom i ich uprawnieniom aplikacja m.in. uzyskuje dostęp (w zależności od treści zapytania SQL) do określonej tabeli w bazie danych i informacji w niej zawartych.

Użytkownicy *lekarz* i *rejestracja* są wykorzystywani najczęściej w komunikacji aplikacji z bazą danych. Łączenie się aplikacji z bazą danych przez użytkownika *lekarz* ma miejsce podczas korzystania z aplikacji przez lekarza, a łączenie się aplikacji z bazą danych przez użytkownika *rejestracja* ma miejsce podczas korzystania z aplikacji przez rejestratora. W związku z tym, użytkownikom *lekarz* i *rejestracja* przyznane zostały takie uprawnienia, które są adekwatne do działań, jakie może podjąć w aplikacji lekarz, a jakie może podjąć rejestrator.

Najważniejszą różnicą między przyznanymi uprawnieniami użytkownikom *lekarz* (patrz: rys. 17) i *rejestracja* (patrz: rys. 18) jest brak możliwości wykonywania jakichkolwiek operacji na tabeli *pobyt* przez użytkownika *rejestracja*. W efekcie, rejestrator korzystający z aplikacji nie może uzyskać dostępu do danych medycznych związanych z pobytom pacjenta, np. rodzaju schorzenia pacjenta, czy opisu przebiegu pobytu. Użytkownik *rejestracja* nie może także wykonywać operacji na kolumnach *historia\_medyczna*, czy *id\_pobytu* znajdujących się w tabeli *pacjent*. Innym przykładem

różnic między uprawnieniami użytkowników *lekarz* i *rejestracja* jest brak możliwości wykonywania operacji INSERT na tabeli *pacjent* przez użytkownika *lekarz* – jedynym uprawnionym użytkownikiem aplikacji do rejestracji nowych pacjentów w bazie jest *rejestrator*.

```
GRANT USAGE ON *.* TO 'lekarz'@'localhost' IDENTIFIED BY PASSWORD '*857EF3AE6847A8DDF064DD2B5D18169B65A557AD'

GRANT SELECT ON `szpital`.`specjalnosc_lekarza` TO 'lekarz'@'localhost'

GRANT SELECT ON `szpital`.`udane_zalogowanie` TO 'lekarz'@'localhost'

GRANT SELECT ON `szpital`.`typ_uzytkownika` TO 'lekarz'@'localhost'

GRANT SELECT ON `szpital`.`schorzenie` TO 'lekarz'@'localhost'

GRANT SELECT ON `szpital`.`oddzial` TO 'lekarz'@'localhost'

GRANT SELECT, INSERT ('id_pacjenta', 'data_rozpoczecia_pobytu', 'id_schorzenia', 'id_oddzialu', 'historia_pobytu', 'id_pobytu', 'id_lekarza'), UPDATE ('podsumowanie_pobytu', 'czy_wygenerowano_akt_wypisu', 'id_schorzenia', 'data_ostatniej_zmiany_danych_pobytu', 'data_zakonczenia_pobytu', 'id_oddzialu', 'historia_pobytu') ON `szpital`.`pobyt` TO 'lekarz'@'localhost'

GRANT SELECT ('id_typu_uzytkownika', 'czy_uzupelniono_dane', 'id_uzytkownika'), UPDATE ('czy_uzupelniono_dane') ON `szpital`.`uzytkownik` TO 'lekarz'@'localhost'

GRANT SELECT, UPDATE ('nr_domu', 'data_rozpoczecia_ostatniego_pobytu', 'nr_mieszkania', 'data_zakonczenia_ostatniego_pobytu', 'id_pobytu', 'historia_pacjenta', 'miasto', 'email', 'kod_pocztowy', 'data_ostatniej_zmiany_danych', 'telefon', 'id_lekarza', 'wojewodztwo', 'ulica') ON `szpital`.`pacjent` TO 'lekarz'@'localhost'

GRANT SELECT, INSERT, UPDATE ('nr_domu', 'email', 'telefon', 'kod_pocztowy', 'wojewodztwo', 'data_ostatniej_zmiany_danych', 'miasto', 'nr_mieszkania', 'ulica') ON `szpital`.`lekarz` TO 'lekarz'@'localhost'
```

Rys. 17. Uprawnienia przyznane użytkownikowi MariaDB *lekarz* (źródło własne)

```
GRANT USAGE ON *.* TO 'rejestracja'@'localhost' IDENTIFIED BY PASSWORD '*57AFC39EEE41A561788D3B04098D6719724C668'

GRANT SELECT ('id_lekarza', 'email', 'telefon', 'imie', 'specjalnosc', 'nazwisko') ON `szpital`.`lekarz` TO 'rejestracja'@'localhost'

GRANT SELECT ON `szpital`.`oddzial` TO 'rejestracja'@'localhost'

GRANT SELECT ('id_typu_uzytkownika', 'czy_uzupelniono_dane', 'id_uzytkownika'), UPDATE ('czy_uzupelniono_dane') ON `szpital`.`uzytkownik` TO 'rejestracja'@'localhost'

GRANT SELECT ('id_pacjenta', 'data_dodania_do_bazy', 'nr_mieszkania', 'data_rozpoczecia_ostatniego_pobytu', 'miejsce_urodzenia', 'data_ostatniej_zmiany_danych', 'nazwisko', 'email', 'imie', 'kod_pocztowy', 'telefon', 'id_lekarza', 'wojewodztwo', 'obywatelstwo', 'pesel', 'miasto', 'nr_domu', 'plec', 'data_urodzenia', 'ulica'), INSERT ('data_dodania_do_bazy', 'nr_mieszkania', 'miejsce_urodzenia', 'nazwisko', 'email', 'imie', 'kod_pocztowy', 'telefon', 'wojewodztwo', 'obywatelstwo', 'pesel', 'miasto', 'nr_domu', 'plec', 'data_urodzenia', 'ulica'), UPDATE ('nr_mieszkania', 'data_ostatniej_zmiany_danych', 'email', 'kod_pocztowy', 'telefon', 'wojewodztwo', 'miasto', 'nr_domu', 'ulica') ON `szpital`.`pacjent` TO 'rejestracja'@'localhost'

GRANT SELECT ON `szpital`.`udane_zalogowanie` TO 'rejestracja'@'localhost'

GRANT SELECT ON `szpital`.`specjalnosc_lekarza` TO 'rejestracja'@'localhost'

GRANT SELECT ON `szpital`.`typ_uzytkownika` TO 'rejestracja'@'localhost'

GRANT SELECT, INSERT, UPDATE ('nr_domu', 'email', 'telefon', 'kod_pocztowy', 'wojewodztwo', 'data_ostatniej_zmiany_danych', 'miasto', 'nr_mieszkania', 'ulica') ON `szpital`.`rejestrator` TO 'rejestracja'@'localhost'
```

Rys. 18. Uprawnienia przyznane użytkownikowi MariaDB *rejestracja* (źródło własne)

Do komunikacji aplikacji z bazą danych w ściśle określonych okolicznościach wykorzystywani są również użytkownicy *uwierzytelnienie* i *zmien\_haslo*. Pierwszy z

wymienionych wykorzystywany jest do połączenia z bazą danych w momencie próby uwierzytelnienia logującego się lekarza lub rejestratora. Drugi natomiast wykorzystywany jest do połączenia z bazą danych w momencie próby zmiany hasła użytkownika aplikacji. Uprawnienia obu użytkowników (patrz: rys. 19 i 20) są ograniczone do wymaganego minimum i są ściśle związane z operacjami, które muszą wykonać.

```
MariaDB [szpital]> show grants for uwierzytelnienie@localhost;
+-----+
| Grants for uwierzytelnienie@localhost |
+-----+
| GRANT USAGE ON *.* TO `uwierzytelnienie`@`localhost` IDENTIFIED BY PASSWORD '*93F17CC691EE1A843D6C9E512ECCDA8A9C999D29' |
| GRANT SELECT ON `szpital`.`uzytkownik` TO `uwierzytelnienie`@`localhost` |
| GRANT INSERT ON `szpital`.`udane_zalogowanie` TO `uwierzytelnienie`@`localhost` |
+-----+
3 rows in set (0,000 sec)
```

*Rys. 19. Uprawnienia przyznane użytkownikowi MariaDB uwierzytelnienie (źródło własne)*

```
MariaDB [szpital]> show grants for zmien_haslo@localhost;
+-----+
| Grants for zmien_haslo@localhost |
+-----+
| GRANT USAGE ON *.* TO `zmien_haslo`@`localhost` IDENTIFIED BY PASSWORD '*E9360E56822DCB6598A859F858019FDC93FCE693' |
| GRANT SELECT ('haslo', 'czy_uzupelniono_dane', 'id_uzytkownika'), UPDATE ('haslo') ON `szpital`.`uzytkownik` TO `zmien_haslo`@`localhost` |
| GRANT SELECT ('id_rejestratora'), UPDATE ('data_ostatniej_zmiany_hasla') ON `szpital`.`rejestrator` TO `zmien_haslo`@`localhost` |
| GRANT SELECT ('id_lekarza'), UPDATE ('data_ostatniej_zmiany_hasla') ON `szpital`.`lekarz` TO `zmien_haslo`@`localhost` |
+-----+
4 rows in set (0,000 sec)
```

*Rys. 20. Uprawnienia przyznane użytkownikowi MariaDB zmien\_haslo (źródło własne)*

#### 4.3.2 Zaszyfrowane dane w tabelach

Większość danych przechowywanych w bazie danych „Szpital” jest zapisana w formie zaszyfrowanej. Dotyczy to danych osobowych, kontaktowych lekarzy, rejestratorów i pacjentów oraz, co najważniejsze, danych medycznych pacjentów. Kolumny, które przechowują dane w formie zaszyfrowanej odznaczają się jednolitym typem: *varchar(128)* lub *longtext* w przypadku kolumn, które mogą przechowywać znaczne ciągi znaków, jak np. kolumna *historia\_pacjenta* w tabeli *pacjent*, przechowującą historię zakończonych pobytów pacjenta w szpitalu.

Wszystkie kolumny o typie *varchar(128)* lub *longtext* znajdują się na diagramie zawartym w rys. 6. Na poniższym rysunku (rys. 21) przedstawiono przykładowe wyświetlenie zaszyfrowanych danych (w tym przypadku numerów PESEL pacjentów).



```

MariaDB [szpital]> select pesel from pacjent;
+-----+
| pesel |
+-----+
| SzhvSTdvTjZEY3M4VlPvWXhrd1NyUT090jr5C7g40BKathbZcWF7cKjn |
| a3NtL0IzZkx0MWtqNWRwWdk3SXVUzz090jowAv4u4M/CQ15YRc8EAFKA |
| OWNDNHB3NzgwVXVsTERmT1RWZxp1dz090jqHBVPWpH8nQ0CNFLgYqmZy |
| UjRyK3dsR2JwQ2RpK0xESFNKMThaUT090jraxqAwPBLZSRtX+ueUHS3w |
| QmlrTk00WmNHTExYT1AxY3BLUlp3dz090jrwRBCo8hzK2K+KvqVobYZM |
| cCtVOEN1NGdSdUd4MDE2Sk10WGJldz090jqnVwDnY5H6PUBxwHACwUd6 |
| Wk5DZGxVUmXSU2tkSnpJdHVkMDFtUT090jpQkvGIsvDWfZS1IDD96V1I |
+-----+
7 rows in set (0,001 sec)

```

Rys. 21. Numery PESEL pacjentów w zaszyfrowanej formie (źródło własne)

Sposób w jaki dane zostają zaszyfrowane przed zapisaniem ich w bazie danych przedstawiony zostanie w rozdziale 5.4.

#### 4.3.3 Automatyczna kopia zapasowa bazy danych „Szpital” oraz logi MariaDB

Wykonywanie kopii zapasowych bazy danych gwarantuje możliwość odzyskania wszystkich informacji w niej zgromadzonych. Kopia zapasowa bazy może być szczególnie przydatna w przypadku wystąpienia problemów z pracą systemu MariaDB lub gdy konieczne będzie przywrócenie bazy do stanu sprzed np. błędnej modyfikacji wielu danych.

Kopia zapasowa całej bazy danych „Szpital” wykonywana jest automatycznie o godzinie 23:00 każdego dnia. Wykonuje ją skrypt, którego kod (patrz: rys. 22) uruchamiany jest codziennie o podanej wcześniej godzinie dzięki programowi *crontab*. Skrypt powstał na bazie rozwiązania dostępnego na stronie [www.spacerex.co](http://www.spacerex.co) [25]. Na potrzeby skryptu, dostęp do bazy danych „Szpital” przyznany został użytkownikowi MariaDB *kopia\_szpital*, którego uprawnienia pozwalają na pobieranie (operacja SELECT) wszystkich tabel z bazy. Po każdym powodzeniu wykonania kopii, do stworzonego pliku typu .log dodawany jest odpowiedni wpis. Skrypt zapobiega również tworzeniu nadmiaru przechowywanych kopii, usuwając te, od których powstania minęło ponad 14 dni. Na rys. 23 przedstawione zostały wykonane o godzinie 23:00 kopie zapasowe bazy danych, potwierdzając tym samym poprawność działania skryptu.



```

GNU nano 7.2                                kopia_szpital.sh
#!/bin/bash
# Sciezka do folderu z kopiami zapasowymi bazy
kopia_folder=/var/backups/kopia_szpital
# Sciezka do pliku log
kopia_log=/var/backups/kopia_szpital.log
# Uzytkownik bazy
uzytkownik=kopia_szpital
# Haslo uzytkownika bazy
haslo=l3nZ58gZgpT5yJR
# Liczba dni przechowywania kopii
liczba_dni=14
# Tworzenie kopii
kopia_plik=$kopia_folder/kopia_szpital_$(date +%Y-%m-%d_%H-%M-%S).sql
echo Poczatek wykonywania kopii zapasowej bazy szpital [$(date +%Y-%m-%d_%H-%M-%S)] >> $kopia_log

/usr/bin/mysqldump -u$uzytkownik -p$haslo --databases szpital >> $kopia_plik
if [ $? == 0 ]; then
    echo 'Kopia zapasowa bazy szpital wykonana pomyslnie' >> $kopia_log
else
    echo [error] mysqldump return non-zero code $? >> $kopia_log
    exit
fi

# Usuwanie kopii po 14 dniach
find $kopia_folder -mtime +$liczba_dni -delete

[ Wczytano 25 linii ]

```

Rys. 22. Kod pliku `kopia_szpital.sh` wykonującego kopię zapasową bazy danych „Szpital” (źródło własne)

```

root@debian:/var/backups/kopia_szpital# ls -l
razem 1484
-rw-r----- 1 root root 197378 11-24 23:00 kopia_szpital_2023-11-24_23-00-01.sql
-rw-r----- 1 root root 258038 11-30 23:00 kopia_szpital_2023-11-30_23-00-01.sql
-rw-r----- 1 root root 258524 12-03 23:00 kopia_szpital_2023-12-03_23-00-01.sql
-rw-r----- 1 root root 263955 12-04 23:00 kopia_szpital_2023-12-04_23-00-01.sql
-rw-r----- 1 root root 264864 12-06 23:00 kopia_szpital_2023-12-06_23-00-01.sql
-rw-r----- 1 root root 264569 12-08 23:00 kopia_szpital_2023-12-08_23-00-01.sql
root@debian:/var/backups/kopia_szpital#

```

Rys. 23. Zawartość folderu w którym przechowywane są kopie zapasowe bazy danych „Szpital” (źródło własne)

Logi MariaDB zapisywane są w plikach o ścieżce dostępu `/var/log/mysql`. Szczególnie istotnym plikiem jest `error.log`, który przechowuje logi związane z błędami, które wystąpiły podczas pracy systemu MariaDB. Jedną z kategorii logów w tym pliku są odnotowane odmowy dostępu do bazy danych lub próby nieudanych logowań. Takie logi (zawarte na rys. 24) pozwalają na zauważenie podejrzanych prób uzyskania dostępu do systemu MariaDB i baz danych w niej zawartych.

```
root@debian:/var/log/mysql# mysql -u admin -p
Enter password:
ERROR 1045 (28000): Access denied for user 'admin'@'localhost' (using password: YES)
root@debian:/var/log/mysql# mysql -u admin -p
Enter password:
ERROR 1045 (28000): Access denied for user 'admin'@'localhost' (using password: YES)
root@debian:/var/log/mysql# tail -n 2 error.log
2023-12-07 14:26:52 32 [Warning] Access denied for user 'admin'@'localhost' (using password: YES)
2023-12-07 14:27:04 33 [Warning] Access denied for user 'admin'@'localhost' (using password: YES)
```

*Rys. 24. Dwukrotna nieudana próba zalogowania użytkownika admin do systemu MariaDB oraz odnotowanie tych prób w pliku error.log (źródło własne)*

## 5. Implementacja aplikacji webowej oraz testy funkcjonalne aplikacji

W niniejszym rozdziale przedstawione zostaną języki programowania za pomocą których napisano aplikację webową. Następnie przybliżone zostaną wybrane mechanizmy działania aplikacji: łączenie z bazą danych, ochrona przed atakami typu SQL Injection oraz proces szyfrowania i deszyfrowania danych. Ostatnią część rozdziału stanowią testy funkcjonalne aplikacji webowej.

### 5.1 Wykorzystane technologie programowania

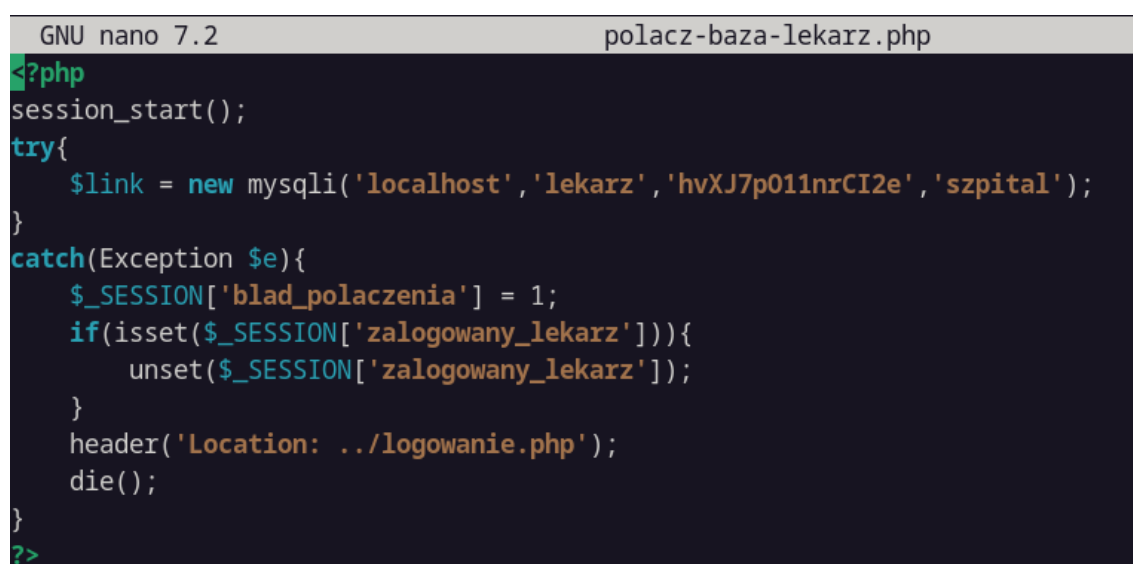
Do utworzenia back-endu aplikacji webowej użyto języków programowania HTML oraz PHP. Język HTML posłużył do utworzenia struktury (szkieletu) stron aplikacji, które są widoczne dla użytkownika aplikacji. Za pomocą tagów języka HTML na stronach umieszczono nagłówki, akapity z tekstem, tabele, listy czy odsyłacze. Dzięki tagom `<form></form>` umieszczono na stronach formularze, które z wykorzystaniem zmiennych globalnych POST języka PHP, pozwalają na przetwarzanie danych wprowadzonych przez użytkownika w aplikacji. Inną wykorzystywaną w aplikacji zmienną globalną języka PHP jest zmienna SESSION. Dzięki tej zmiennej możliwa jest weryfikacja, czy użytkownik jest zalogowany w aplikacji. Zmienne SESSION pozwalają również na wyświetlanie komunikatów na stronie, np. po udanej zmianie hasła przez użytkownika. Łączenie z bazą danych, obsługa zapytań w języku SQL, szyfrowanie i deszyfrowanie danych czy przekierowywanie użytkownika na określoną stronę – wszystko to gwarantują różne klasy, metody i funkcje języka PHP.

Część front-endowa aplikacji została natomiast zaprogramowana przy użyciu CSS oraz języka programowania JavaScript, w tym jednej z jej bibliotek – jQuery. CSS umożliwia dodawanie tzw. stylów do tagów HTML. Style pozwalają na zmianę wyglądu strony, w tym zmianę koloru tła strony lub jej części, zmianę czcionki tekstu, jej koloru i rozmiaru oraz zmianę wyglądu tabel, list, czy przycisków. Język JavaScript i biblioteka jQuery odpowiada zaś za część interaktywną strony. Dzięki użyciu tych technologii użytkownik aplikacji może np. na bieżąco sprawdzać, czy wpisane przez niego nowe hasło spełnia określone na stronie wymagania.

Aplikacja webowa jako całość została napisana w programie Visual Studio Code na systemie operacyjnym Windows. Finalnie aplikacja została przeniesiona na serwer.

## 5.2 Łączenie aplikacji z bazą danych

W celu ustanowienia połączenia pomiędzy aplikacją i bazą danych, na początku kodu strony aplikacji dołączony jest jeden z czterech plików o nazwie *polacz-baza-<nazwa użytkownika MariaDB>.php*. Przykładowo w plikach aplikacji znajdujących się w folderze Lekarz, dołączony jest plik o nazwie *polacz-baza-lekarz.php*. Zawartość każdego z tych plików (patrz: rys. 25) jest bardzo podobna i różni się przede wszystkim wpisaną nazwą użytkownika MariaDB oraz hasłem.



```
GNU nano 7.2                                polacz-baza-lekarz.php
<?php
session_start();
try{
    $link = new mysqli('localhost','lekarz','hvXJ7p011nrCI2e','szpital');
}
catch(Exception $e){
    $_SESSION['blad_polaczenia'] = 1;
    if(isset($_SESSION['zalogowany_lekarz'])){
        unset($_SESSION['zalogowany_lekarz']);
    }
    header('Location: ../logowanie.php');
    die();
}
?>
```

Rys. 25. Kod pliku *polacz-baza-lekarz.php* (źródło własne)

Za ustanowienie połączenia odpowiada nowy obiekt `$link` klasy `mysqli`. Wykonywanie w dalszej części kodu aplikacji zapytań do bazy nie jest możliwe bez wywołania obiektu `$link`, który reprezentuje połączenie z bazą. Pliki *polacz-baza* „wylogowywują” i przekierowują użytkownika aplikacji na stronę logowania w przypadku niepowodzenia ustanowienia połączenia z bazą danych.

Jako że pliki *polacz-baza* przechowują w formie jawnej hasła użytkowników systemu MariaDB, konieczny jest zapis tych plików w miejscu, które jest nieosiągalne dla użytkowników sieci szpitalnej. Takim miejscem jest folder *polacz-baza*, znajdujący

się poziom wyżej w ścieżce dostępu niż folder *html*, czyli wspomniany już wcześniej *Document root* serwera Apache. Dostęp do folderu *polacz-baza* i jego zawartości możliwy jest jedynie z poziomu serwera dla użytkowników mogących korzystać z uprawnień *roota* oraz dla domyślnego użytkownika serwera Apache *www-data* (patrz: rys. 26).

```
root@debian:/var/www/polacz-baza# cd /var/www/polacz-baza
root@debian:/var/www/polacz-baza# ls -l
razem 16
-rw-r----- 1 root www-data 329 12-04 19:18 polacz-baza-lekarz.php
-rw-r----- 1 root www-data 351 12-04 22:39 polacz-baza-rejestracja.php
-rw-r----- 1 root www-data 459 12-04 19:19 polacz-baza-uwierzytelnienie.php
-rw-r----- 1 root www-data 452 12-04 19:19 polacz-baza-zmien-haslo.php
root@debian:/var/www/polacz-baza#
```

Rys. 26. Uprawnienia do plików w folderze *polacz-baza* (źródło własne)

### 5.3 Ochrona przed atakami SQL Injection

SQL Injection [26] jest popularną wśród cyberprzestępców metodą ataku na bazę danych. Atak ten opiera się na manipulacji przez atakującego zapytań SQL, wysyłanych przez aplikację do bazy danych. Atakujący wykorzystuje podatności w kodzie aplikacji webowej, które umożliwiają mu „wstrzyknięcie” do zapytań SQL - najczęściej przez formularze aplikacji webowej - swoich własnych, przygotowanych zapytań SQL. Wykonanie tak zmodyfikowanych zapytań może sprawić, że atakujący uzyska dostęp np. do wszystkich loginów i haseł przechowywanych w bazie lub innych, cennych informacji.

Aby zminimalizować ryzyko nieautoryzowanego dostępu do bazy danych przez atakującego, w kodzie napisanej aplikacji webowej wykorzystano tzw. *prepared statements*, czyli przygotowane instrukcje, wykorzystujące metody klasy *mysqli* języka PHP. Przykład *prepared statement* został zaprezentowany na rys. 27.

```
$zapytanie_pacjent = $link->prepare("SELECT * FROM pacjent WHERE id_pacjenta=? AND id_lekarza=?");
$zapytanie_pacjent->bind_param('ii', $_GET['id_pacjenta'], $_SESSION['zalogowany_id']);
$zapytanie_pacjent->execute();
```

Rys. 27. Przykładowy *prepared statement* w kodzie aplikacji (źródło własne)

W pierwszej linijce powyższego *prepared statement* podany jest szkielet zapytania SQL, który zostaje przesłany do bazy danych przez wywołanie obiektu `mysqli $link`. Charakterystyczną cechą *prepared statement* jest wykorzystanie w szkielecie zapytania SQL znaku(ów) pytajnika. W następnej linijce w metodzie klasy `mysqli bind_param()` deklarowane zostają dwie zmienne (deklaruje się tyle zmiennych, ile pytajników) wraz z ich typem. Metoda `bind_param()` zastępuje pytajniki w przesłanym do bazy zapytaniu SQL na wartości zadeklarowanych zmiennych. W przypadku tego *prepared statement*, obie zadeklarowane zmienne przechowują wartości liczbowe, dlatego we wnętrzu funkcji znajdują się dwie litery „i”, od zmiennej typu *integer*. Zmienne typu *string* symbolizuje litera „s”. W ostatniej linijce *prepared statement* metoda `execute()` wykonuje zapytanie SQL, uzupełnione wcześniej o wartości zadeklarowanych zmiennych.

*Prepared statements* chronią zapisane w kodzie aplikacji zapytania SQL przed ich niepożądaną zmianą, minimalizując ryzyko ataku typu SQL Injection na bazę danych.

## 5.4 Szyfrowanie i deszyfrowanie danych

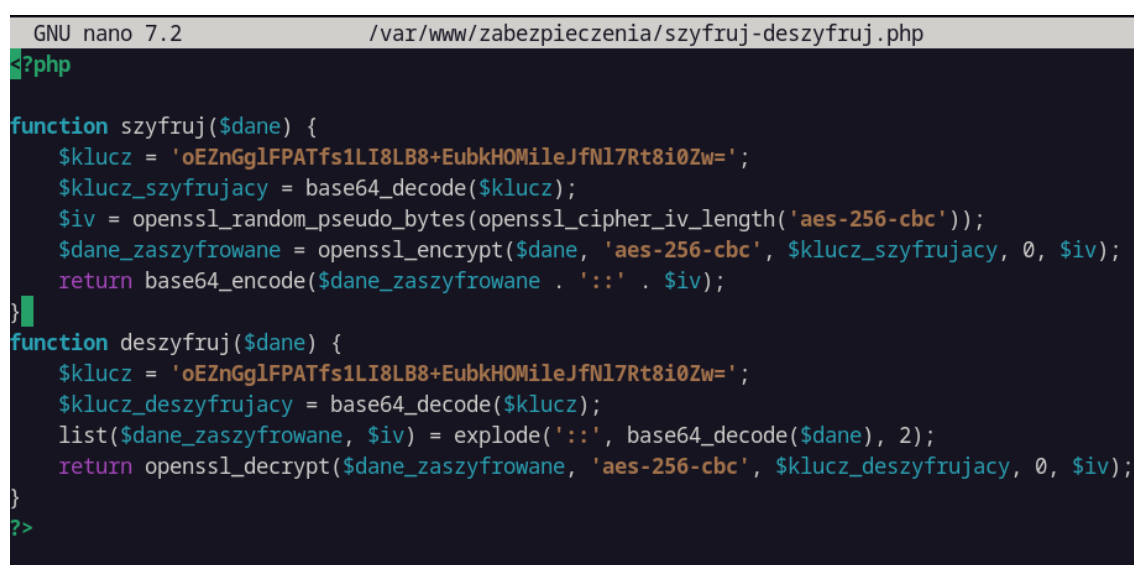
Szyfrowanie i deszyfrowanie danych w aplikacji odbywa się za pomocą dwóch funkcji `szyfruj()` i `deszyfruj()`. Funkcje powstały na bazie rozwiązania dostępnego na stronie <https://www.meridianoutpost.com/> [27].

W funkcji `szyfruj()` dane są szyfrowane z wykorzystaniem algorytmu symetrycznego AES. Szyfrowanie symetryczne oznacza, że do szyfrowania i deszyfrowania danych wykorzystywany jest ten sam klucz. Dodatkowo przy szyfrowaniu danej wewnątrz funkcji `openssl_encrypt()` wykorzystywany jest losowy wektor inicjalizujący, który powoduje, że dane mające tę samą wartość, po zaszyfrowaniu mogą przyjmować odmienne ciągi znaków. Funkcja `szyfruj()` zwraca zaszyfrowaną informację wraz z jej wektorem inicjalizującym w postaci zakodowanej, dzięki użyciu funkcji `base64_encode()`. Tak przygotowana informacja następnie zostaje zapisana w bazie danych.

W funkcji `deszyfruj()` dane są deszyfrowane z wykorzystaniem tego samego algorytmu AES. Najpierw następuje dekodowanie (funkcja `base64_decode()`)

zaszyfrowanej informacji pobranej z bazy danych oraz podział na samą zaszyfrowaną informację oraz jej wektor inicjalizujący. Ostatecznie funkcja zwraca daną w formie jawnej za pomocą funkcji `openssl_decrypt()`.

Funkcje `szyfruj()` i `deszyfruj()` znajdują się w pliku `szyfruj_deszyfruj.php`, którego zawartość została pokazana na rys. 28. Podobnie jak wcześniej wspomniane pliki *polacz-baza*, plik ten jest przechowywany w bezpiecznym, niedostępnym z sieci miejscu w systemie operacyjnym Debian. Dostęp do tego pliku został przyznany wyłącznie użytkownikom mogącym korzystać z uprawnień *roota* oraz dla domyślnego użytkownika serwera Apache *www-data*.



```
GNU nano 7.2 /var/www/zabezpieczenia/szyfruj-deszyfruj.php
?php

function szyfruj($dane) {
    $klucz = 'oEZnGglFPATfs1LI8LB8+EubkHOMileJfNl7Rt8i0Zw=';
    $klucz_szyfrujacy = base64_decode($klucz);
    $iv = openssl_random_pseudo_bytes(openssl_cipher_iv_length('aes-256-cbc'));
    $dane_zaszyfrowane = openssl_encrypt($dane, 'aes-256-cbc', $klucz_szyfrujacy, 0, $iv);
    return base64_encode($dane_zaszyfrowane . '::' . $iv);
}

function deszyfruj($dane) {
    $klucz = 'oEZnGglFPATfs1LI8LB8+EubkHOMileJfNl7Rt8i0Zw=';
    $klucz_deszyfrujacy = base64_decode($klucz);
    list($dane_zaszyfrowane, $iv) = explode(':', base64_decode($dane), 2);
    return openssl_decrypt($dane_zaszyfrowane, 'aes-256-cbc', $klucz_deszyfrujacy, 0, $iv);
}
?>
```

Rys. 28. Kod pliku `szyfruj-deszyfruj.php` (źródło własne)

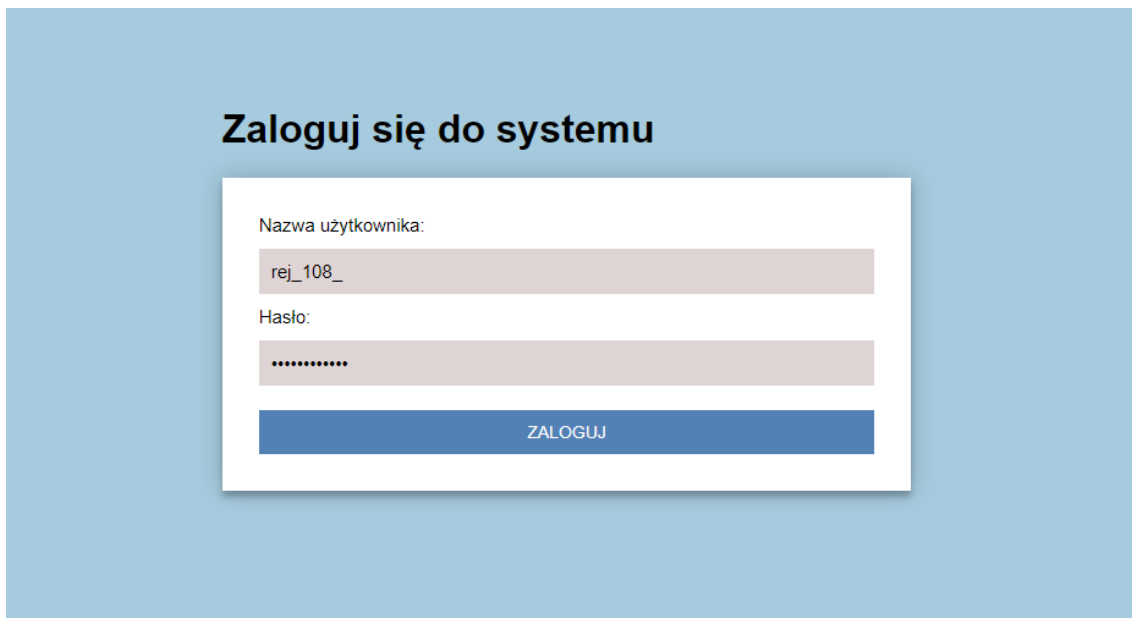
## 5.5 Testy funkcjonalne aplikacji webowej

Testy funkcjonalne aplikacji webowej skupiają się na zaprezentowaniu interfejsu graficznego aplikacji i, przede wszystkim, na przedstawieniu różnych funkcjonalności, które oferuje napisana aplikacja webowa, a które zostały sprecyzowane w wymaganiach funkcjonalnych w rozdziale 3.2. Celem uporządkowania treści obecnego podrozdziału, testowane funkcjonalności aplikacji odwołują się do zdefiniowanych wcześniej przypadków użycia, które zostały opisane w rozdziale 3.3. Każdy rysunek (zrzut ekranu) jest poprzedzony krótkim opisem działania użytkownika, którego efekt został zaprezentowany właśnie na rysunku. Zrzuty ekranu aplikacji związane z przypadkiem

użycia, w którym zdefiniowany został tylko jeden aktor zamieszczono w oddzielnych podrozdziałach.

### **PU-1 – Logowanie się do aplikacji**

Do aplikacji loguje się użytkownik (w tym przypadku rejestrator) – rys. 29.



The image shows a login form titled "Zaloguj się do systemu" (Log in to the system) centered on a light blue background. The form itself is a white rectangle with a subtle drop shadow. It contains two input fields: the first is labeled "Nazwa użytkownika:" (Username:) and contains the text "rej\_108\_"; the second is labeled "Hasło:" (Password:) and contains a series of dots. Below these fields is a blue button with the white text "ZALOGUJ" (Log in).

*Rys. 29. PU-1 – Formularz logowania do aplikacji (źródło własne)*

### **PU-2 – Uzupełnienie danych osobowych i kontaktowych po pierwszym zalogowaniu w aplikacji przez użytkownika.**

Formularz uzupełnił wcześniej zalogowany użytkownik – rys. 30.



Panel rejestracja

Wyloguj się

Na początku uzupełnij swoje dane, które zostaną zapisane w systemie.  
Po prawidłowym uzupełnieniu danych, ta strona po zalogowaniu **nie będzie się już wyświetlać**.  
Swoje dane będziesz mógł zaktualizować w każdym momencie, po wybraniu odpowiedniej opcji w panelu Rejestracji.

Dane osobowe i zawodowe

Pola z gwiazdką są obowiązkowe.

Imię\*

Aneta

Nazwisko\*

Jaskółka

Płeć\*

Kobieta

PESEL\*

78021255486

Data urodzenia\*

12 . 02 . 1978

Miejsce urodzenia\*

Gdańsk

Dane kontaktowe

Pola z gwiazdką są obowiązkowe.

Ulica\*

Akacjowa

Nr domu\*

12

Nr mieszkania (jeżeli dotyczy)

Miasto\*

Kraków

Województwo\*

Małopolskie

Kod pocztowy\*

31-466

Adres e-mail\*

aneta\_jaskolka@gmail.com

Telefon komórkowy\*

521 894 282

UZUPEŁNIJ DANE

Rys. 30. PU-2 – Formularz uzupełnienia danych (źródło własne)

Po zapisaniu danych w bazie, użytkownik ma możliwość zmiany hasła przydzielonego przez Administratora lub przejścia do strony głównej – rys. 31. Zrzut ekranu strony przeznaczonej do zmiany hasła zostanie zaprezentowany w dalszej części rozdziału.

Panel rejestracja

Wyloguj się

**Dane zostały prawidłowo uzupełnione.**

Teraz możesz zdecydować czy chcesz zmienić swoje hasło dostępu do systemu, przyznane przez Administratora.  
**Pamiętaj, że hasło powinno być silne, zgodnie z poniższymi zaleceniami.**  
Jeżeli nie chcesz zmieniać hasła przyznanego przez Administratora, kliknij w link [Przejdź do strony głównej](#)

[Przejdź do strony głównej](#)

Rys. 31. PU-2 – Potwierdzenie uzupełnienia danych podanych wcześniej przez użytkownika po pierwszym logowaniu do aplikacji (źródło własne)

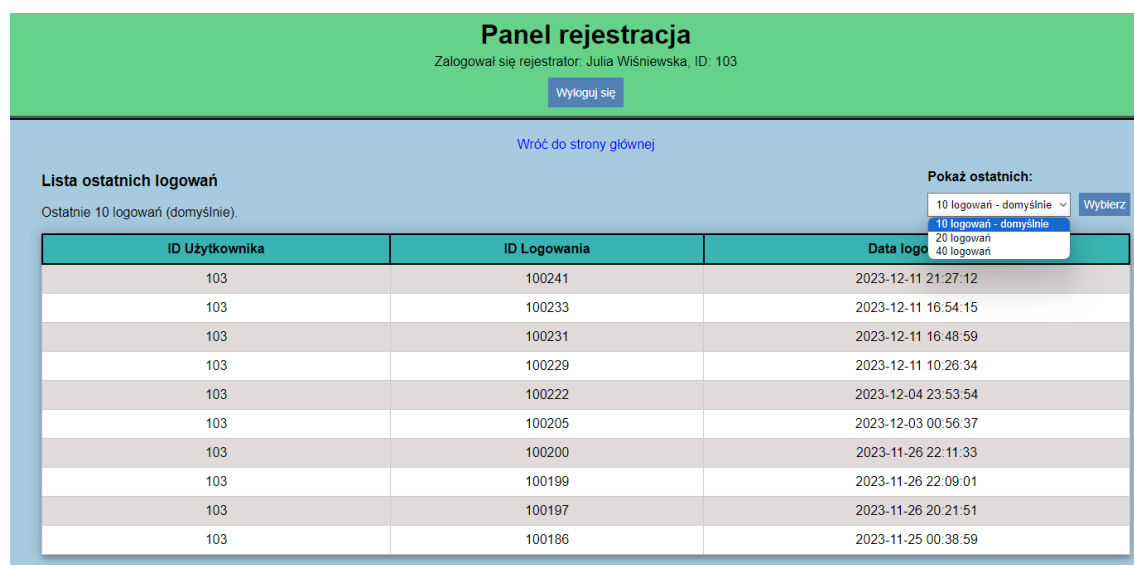
Użytkownik przeszedł do strony głównej – rys. 32.



Rys. 32. PU-2 – Widok strony głównej rejestratora (źródło własne)

### PU-3 – Wyświetlenie listy dat ostatnich logowań

Użytkownik wybrał zakładkę „Wyświetl daty ostatnich logowań” ze strony głównej. Użytkownik ma możliwość wyświetlenia dat ostatnich 20 lub 40 logowań – rys. 33.



Rys. 33. PU-3 – Widok strony z listą ostatnich dat logowań użytkownika (źródło własne)

### PU-4 – Wyświetlenie listy lekarzy

Użytkownik wybrał zakładkę „Lista lekarzy” ze strony głównej. Użytkownik ma możliwość sortowania listy po nr ID lekarza rosnąco lub malejąco – rys. 34.

## Panel rejestracja

Zalogował się rejestrator: Julia Wiśniewska, ID: 103

Wyloguj się

Wróć do strony głównej

Lista lekarzy

Lista posortowana wg ID Lekarza rosnąco (domyślnie).

Sortuj listę:

Sortuj po ID (rosnąco) - domyślnie

Sortuj

| ID Lekarza | Imię     | Nazwisko     | Specjalność        | Telefon     | E-mail                                  |
|------------|----------|--------------|--------------------|-------------|---|
| 101        | Rafał    | Kowalczyk    | Onkolog            | 523 456 898 | kowalczyk_rafal_lekarz@gmail.com        |
| 104        | Anna     | Nowacka      | Choroby wewnętrzne | 541 490 333 | anna.nowacka@gmail.com                  |
| 105        | Jan      | Rożek        | Chirurg            | 345 746 231 | rozek_jan@poczta.wp.pl                  |
| 109        | Danuta   | Kaczmarczyk  | Pediatra           | 501 492 281 | danutakaczmarczyk@gmail.com             |
| 110        | Michał   | Wawrzykowski | Anestezjolog       | 632 379 192 | anestezjologia.wawrzykowski@outlook.com |
| 111        | Krystyna | Mirowska     | Choroby zakaźne    | 451 238 928 | krystyna_mirowska@op.pl                 |
| 112        | Tadeusz  | Rajkowski    | Ginekolog          | 602 382 459 | rajkowski.ginekologia@gmail.com         |

Rys. 34. PU-4 – Widok strony z listą lekarzy (źródło własne)

## PU-5 – Wyświetlenie danych osobowych i kontaktowych użytkownika

Użytkownik wybrał zakładkę „Wyświetl i zmień swoje dane” ze strony głównej – rys. 35.

## Panel rejestracja

Zalogował się rejestrator: Julia Wiśniewska, ID: 103

Wyloguj się

Wróć do strony głównej

Ostatnia zmiana danych: 2023-12-11 21:51:25

Jeżeli chcesz zmienić dane kontaktowe (adres zamieszkania, telefon, email), kliknij przycisk [Chcę zmienić dane](#).  
Wszystkie pola przeznaczone do edycji muszą być wypełnione. Możesz wycofać się z operacji zmiany danych, klikając przycisk [Anuluj](#).  
W przypadku chęci zmiany danych osobowych, skontaktuj się z Administratorem.

Chcę zmienić dane

**Dane osobowe**

| Imię  | Nazwisko   | PESEL       | Data urodzenia | Miejsce urodzenia |
|-------|------------|-------------|----------------|-------------------|
| Julia | Wiśniewska | 97062379548 | 1997-06-23     | Wieliczka         |

**Dane kontaktowe**

| Ulica        | Nr domu            | Nr mieszkania |
|--------------|--------------------|---------------|
| Główna       | 17                 | 1             |
| Kod pocztowy | Miasto             | Województwo   |
| 32-020       | Wieliczka          | Małopolskie   |
| Telefon      | E-mail             |               |
| 290 209 382  | j.wisniewska@op.pl |               |

Rys. 35. PU-5 – Widok strony z danymi osobowymi i kontaktowymi użytkownika (źródło własne)

## PU-6 – Zmiana danych kontaktowych użytkownika

Użytkownik kliknął przycisk „Chcę zmienić dane” na stronie z swoimi danymi i zmienił swój numer telefonu – rys. 36.

## Panel rejestracja

Zalogował się rejestrator: Julia Wiśniewska, ID: 103

Wyloguj się

[Wróć do strony głównej](#)

Ostatnia zmiana danych: 2023-12-11 21:51:25

Jeżeli chcesz zmienić dane kontaktowe (adres zamieszkania, telefon, email), kliknij przycisk *Chcę zmienić dane*.  
Wszystkie pola przeznaczone do edycji muszą być wypełnione. Możesz wycofać się z operacji zmiany danych, klikając przycisk *Anuluj*.  
W przypadku chęci zmiany danych osobowych, skontaktuj się z Administratorem.

Anuluj

### Dane osobowe

| Imię  | Nazwisko   | PESEL       | Data urodzenia | Miejsce urodzenia |
|-------|------------|-------------|----------------|-------------------|
| Julia | Wiśniewska | 97062379548 | 1997-06-23     | Wieliczka         |

### Dane kontaktowe

| Ulica        | Nr domu            | Nr mieszkania |
|--------------|--------------------|---------------|
| Główna       | 17                 | 1             |
| Kod pocztowy | Miasto             | Województwo   |
| 32-020       | Wieliczka          | Małopolskie   |
| Telefon      | E-mail             |               |
| 639 282 181  | l_wisniewska@op.pl |               |

ZMIEN DANE

Rys. 36. PU-6 – Widok zmiany danych użytkownika (źródło własne)

Na stronie głównej pojawia się komunikat o zmianie danych użytkownika – rys. 37.

## Panel rejestracja

Zalogował się rejestrator: Julia Wiśniewska, ID: 103

Wyloguj się

Twoje dane zostały pomyślnie zaktualizowane.

|                                 |
|---------------------------------|
| Dodaj nowego pacjenta           |
| Lista pacjentów                 |
| Lista lekarzy                   |
| Wyświetl i zmień swoje dane     |
| Wyświetl daty ostatnich logowań |
| Zmień hasło                     |

Rys. 37. PU-6 – Widok strony głównej rejestratora po pomyślnej aktualizacji (zmianie) danych użytkownika (źródło własne)

Nowy numer telefonu wyświetla się w danych użytkownika – rys. 38.

Ostatnia zmiana danych: 2023-12-11 21:56:48

Jeżeli chcesz zmienić dane kontaktowe (adres zamieszkania, telefon, email), kliknij przycisk [Chcę zmienić dane](#).  
 Wszystkie pola przeznaczone do edycji muszą być wypełnione. Możesz wycofać się z operacji zmiany danych, klikając przycisk [Anuluj](#).  
 W przypadku chęci zmiany danych osobowych, skontaktuj się z Administratorem.

[Chcę zmienić dane](#)

**Dane osobowe**

| Imię  | Nazwisko   | PESEL       | Data urodzenia | Miejsce urodzenia |
|-------|------------|-------------|----------------|-------------------|
| Julia | Wiśniewska | 97062379548 | 1997-06-23     | Wieliczka         |

**Dane kontaktowe**

|              |                    |               |
|--------------|--------------------|---------------|
| Ulica        | Nr domu            | Nr mieszkania |
| Główna       | 17                 | 1             |
| Kod pocztowy | Miasto             | Województwo   |
| 32-020       | Wieliczka          | Małopolskie   |
| Telefon      | E-mail             |               |
| 639 282 181  | j_wisniewska@op.pl |               |

Rys. 38. PU-6 – Widok danych użytkownika po dokonanej zmianie (źródło własne)

## PU-7 – Zmiana hasła użytkownika

Użytkownik wybrał zakładkę „Zmień hasło” ze strony głównej – rys. 39.

**Panel rejestracja**

Zalogował się rejestrator: Julia Wiśniewska, ID: 103

[Wyloguj się](#)

[Wróć do strony głównej](#)

Ostatnia zmiana hasła: 2023-12-11 10:34:33

Jeżeli chcesz zmienić swoje hasło, skorzystaj z formularza obok.

**Hasło powinno:**

- ☒ Zawierać co najmniej osiem znaków i maksymalnie dwadzieścia cztery znaki
- ☒ Zawierać co najmniej jedną wielką literę
- ☒ Zawierać co najmniej jedną cyfrę
- ☒ Zawierać co najmniej jeden znak specjalny (!, ?, &, \*)

Wprowadź obecne hasło:

Wprowadź nowe hasło:

Powtórz nowe hasło:

[ZMIEN HASŁO](#)

Rys. 39. PU-7 – Widok strony umożliwiającej zmianę hasła użytkownika (źródło własne)

Użytkownik wpisał nowe hasło spełniające wymagania – rys. 40.

*Rys. 40. PU-7 – Widok strony umożliwiającej zmianę hasła użytkownika po wprowadzeniu odpowiedniego nowego hasła (źródło własne)*

Na stronie głównej pojawia się komunikat o zmianie hasła użytkownika – rys. 41.

*Rys. 41. PU-7 – Widok strony głównej rejestratora po pomyślnej aktualizacji (zmianie) hasła użytkownika (źródło własne)*

#### 5.5.1 Testy funkcjonalne aplikacji z perspektywy rejestratora

### PU-8 – Rejestracja nowego pacjenta

Rejestrator wybrał zakładkę „Dodaj nowego pacjenta” ze strony głównej i wypełnił formularz – rys. 42.

Wypełnij formularz o dane osobowe i kontaktowe nowego pacjenta.  
Dane kontaktowe będą mogły być zmienione później w zakładce Wyświetl i zmień dane pacjenta na liście pacjentów, dostępnej z strony głównej.

| Dane osobowe pacjenta  | Dane kontaktowe pacjenta  |
|--|---|
| <p>Pola z gwiazdką są obowiązkowe.</p> <p>Imię*</p> <input type="text" value="Jadwiga"/>                               | <p>Pola z gwiazdką są obowiązkowe.</p> <p>Ulica*</p> <input type="text" value="Centralna"/> |
| <p>Nazwisko*</p> <input type="text" value="Królewicz"/>  | <p>Nr domu*</p> <input type="text" value="11"/>   |
| <p>Płeć*</p> <input type="text" value="Kobieta"/>  | <p>Nr mieszkania (jeżeli dotyczy)</p> <input type="text" value="2"/>                        |
| <p><input type="checkbox"/> Inne obywatelstwo niż polskie</p> <p>Obywatelstwo</p> <input type="text" value="polskie"/> | <p>Miasto*</p> <input type="text" value="Kraków"/>  |
| <p>PESEL*</p> <input type="text" value="59110926922"/>   | <p>Województwo*</p> <input type="text" value="Małopolskie"/>                                |
| <p>Data urodzenia*</p> <input type="text" value="09.11.1959"/>   | <p>Kod pocztowy*</p> <input type="text" value="310579"/>                                    |
| <p>Miejsce urodzenia*</p> <input type="text" value="Myślenice"/>   | <p>Adres e-mail</p> <input type="text" value="przykladowy@email.com"/>                      |
|  | <p>Telefon komórkowy</p> <input type="text" value="123 456 789"/>                           |

**DODAJ PACJENTA**

Rys. 42. PU-8 – Formularz rejestracji nowego pacjenta (źródło własne)

Rejestrator wpisał kod pocztowy w złym formacie – wyskoczył komunikat o wymaganym formacie danych – rys. 43.

Kod pocztowy\*

Adres e-mail

Podaj wartość w wymaganym formacie.

Rys. 43. PU-8 – Komunikat o wymaganym formacie danych (źródło własne)

Po podaniu kodu pocztowego w poprawnym formacie pacjent został dodany do bazy, co potwierdza komunikat – rys. 44.

**Panel rejestracja**  
 Zalogował się rejestrator: Aneta Jaskółka, ID: 108  
[Wyloguj się](#)

[Wróć do strony głównej](#)

Nowy pacjent został dodany pomyślnie. ID Pacjenta: 1016

**Szukaj pacjenta:**


[Szukaj pacjenta](#)

**Lista zarejestrowanych pacjentów**

| ID Pacjenta | Imię i nazwisko   | PESEL       | Data dodania do systemu | Czy pacjent znajduje się aktualnie w szpitalu? | Działania   |
|-------------|-------------------|-------------|-------------------------|--|---|
| 1016        | Jadwiga Królewicz | 59110926922 | 2023-12-11 22:58:56     | NIE  | <a href="#" style="color: #4169E1; text-decoration: none;">Wyświetl i zmień dane pacjenta</a> |

*Rys. 44. PU-8 – Widok strony z listą zarejestrowanych pacjentów po dodaniu pacjenta na tę listę (źródło własne)*

## PU-9 – Przeglądanie listy pacjentów

Rejestrator wybrał zakładkę „Lista pacjentów” ze strony głównej – rys. 45.

**Panel rejestracja**  
 Zalogował się rejestrator: Aneta Jaskółka, ID: 108  
[Wyloguj się](#)

[Wróć do strony głównej](#)

**Szukaj pacjenta:**


[Szukaj pacjenta](#)

**Lista zarejestrowanych pacjentów**

| ID Pacjenta | Imię i nazwisko    | PESEL       | Data dodania do systemu | Czy pacjent znajduje się aktualnie w szpitalu?                       | Działania   |
|-------------|--------------------|-------------|-------------------------|--|---|
| 1016        | Jadwiga Królewicz  | 59110926922 | 2023-12-11 22:58:56     | NIE  | <a href="#" style="color: #4169E1; text-decoration: none;">Wyświetl i zmień dane pacjenta</a> |
| 1015        | Anna Nowak         | 00211054944 | 2023-12-11 10:38:39     | TAK<br>Pobyt od: 2023-12-11 22:42:30<br>ID Lekarza prowadzącego: 101 | <a href="#" style="color: #4169E1; text-decoration: none;">Wyświetl i zmień dane pacjenta</a> |
| 1010        | Marcin Tomaszewski | 91041925494 | 2023-11-26 20:25:32     | TAK<br>Pobyt od: 2023-12-11 22:45:43<br>ID Lekarza prowadzącego: 105 | <a href="#" style="color: #4169E1; text-decoration: none;">Wyświetl i zmień dane pacjenta</a> |
| 1008        | Kazimierz Twaróg   | 59072221578 | 2023-11-20 22:08:16     | NIE  | <a href="#" style="color: #4169E1; text-decoration: none;">Wyświetl i zmień dane pacjenta</a> |
| 1005        | Alina Kadłubek     | 78080937279 | 2023-11-08 14:35:43     | NIE  | <a href="#" style="color: #4169E1; text-decoration: none;">Wyświetl i zmień dane pacjenta</a> |
| 1004        | Halina Nowak       | 78080937278 | 2023-11-08 12:53:27     | TAK<br>Pobyt od: 2023-12-11 22:44:42<br>ID Lekarza prowadzącego: 104 | <a href="#" style="color: #4169E1; text-decoration: none;">Wyświetl i zmień dane pacjenta</a> |
| 1003        | Krzysztof Rosolek  | 84051552877 | 2023-11-03 17:16:42     | NIE  | <a href="#" style="color: #4169E1; text-decoration: none;">Wyświetl i zmień dane pacjenta</a> |
| 1002        | Jean Pierre        | Nie podano  | 2023-10-27 19:28:33     | NIE  | <a href="#" style="color: #4169E1; text-decoration: none;">Wyświetl i zmień dane pacjenta</a> |
| 1001        | Teresa Rosolek     | 88051991362 | 2023-10-27 17:29:19     | NIE  | <a href="#" style="color: #4169E1; text-decoration: none;">Wyświetl i zmień dane pacjenta</a> |

*Rys. 45. PU-9 – Widok strony z listą zarejestrowanych pacjentów (źródło własne)*

## PU-10 – Wyszukiwanie pacjentów po numerze ID, PESELu lub nazwisku

Rejestrator wybrał opcję z listy rozwijanej – wyszukaj pacjenta po ID - i wpisał obok wartość „1020” – rys. 46.



Szukaj pacjenta: Wyszukaj pacjenta po ID (domyślnie) Wpisz ID Pacjenta Szukaj pacjenta

Wyszukaj pacjenta po ID (domyślnie)  
Wyszukaj pacjenta po PESEL  
Wyszukaj pacjenta po nazwisku

Brak pacjenta w systemie z nr ID: 1020

Wróć do całej listy

Rys. 46. PU-10 – Widok strony z listą zarejestrowanych pacjentów po wyszukaniu pacjenta o nr ID 1020 (źródło własne)

Rejestrator wybrał opcję z listy rozwijanej – wyszukaj pacjenta po PESEL - i wpisał obok wartość „00211054944” – rys. 47.

Szukaj pacjenta: Wyszukaj pacjenta po ID (domyślnie) Wpisz ID Pacjenta Szukaj pacjenta

Lista zarejestrowanych pacjentów

| ID Pacjenta | Imię i nazwisko | PESEL       | Data dodania do systemu | Czy pacjent znajduje się aktualnie w szpitalu?                       | Działania                      |
|-------------|-----------------|-------------|-------------------------|--|--------------------------------|
| 1015        | Anna Nowak      | 00211054944 | 2023-12-11 10:38:39     | TAK<br>Pobyt od: 2023-12-11 22:42:30<br>ID Lekarza prowadzącego: 101 | Wyświetl i zmień dane pacjenta |

Wróć do całej listy

Rys. 47. PU-10 – Widok strony z listą zarejestrowanych pacjentów po wyszukaniu pacjenta o nr PESEL „00211054944” (źródło własne)

Rejestrator wybrał opcję z listy rozwijanej – wyszukaj pacjenta po nazwisku - i wpisał obok wartość „Rosolek” – rys. 48.

Szukaj pacjenta: Wyszukaj pacjenta po ID (domyślnie) Wpisz ID Pacjenta Szukaj pacjenta

Lista zarejestrowanych pacjentów

| ID Pacjenta | Imię i nazwisko   | PESEL       | Data dodania do systemu | Czy pacjent znajduje się aktualnie w szpitalu? | Działania                      |
|-------------|-------------------|-------------|-------------------------|--|--------------------------------|
| 1001        | Teresa Rosolek    | 88051991362 | 2023-10-27 17:29:19     | NIE  | Wyświetl i zmień dane pacjenta |
| 1003        | Krzysztof Rosolek | 84051552877 | 2023-11-03 17:16:42     | NIE  | Wyświetl i zmień dane pacjenta |

Wróć do całej listy

Rys. 48. PU-10 – Widok strony z listą zarejestrowanych pacjentów po wyszukaniu pacjenta o nazwisku „Rosolek” (źródło własne)

## 5.5.2 Testy funkcjonalne aplikacji z perspektywy lekarza

### PU-11 – Dodanie nowego pobytu

Lekarz wybrał zakładkę „Dodaj nowy pobyt” ze strony głównej – rys. 49.



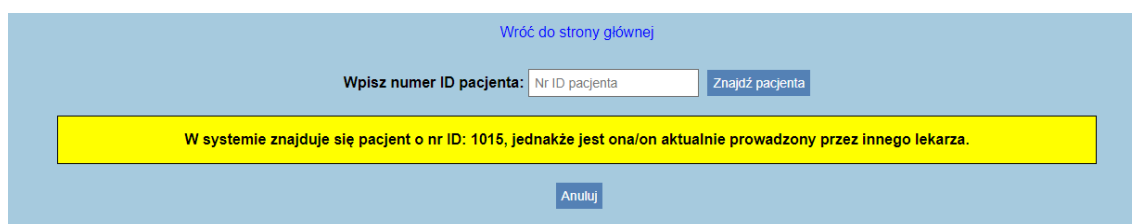
Rys. 49. PU-11 – Widok strony głównej lekarza (źródło własne)

Aby dodać pobyt pacjenta, Lekarz wpisał nr ID pacjenta. Pacjent o nr ID 1017 nie znajduje się w bazie danych (nie został zarejestrowany przez rejestratora), co potwierdza komunikat – rys. 50.



Rys. 50. PU-11 – Widok strony przeznaczonej do dodania nowego pobytu po wpisaniu nr ID pacjenta, który nie znajduje się w bazie danych (źródło własne)

Lekarz wpisał nr ID pacjenta, który znajduje się w bazie danych, ale jego pobyt jest już prowadzony przez innego lekarza, co potwierdza komunikat – rys. 51.



Rys. 51. PU-11 – Widok strony przeznaczonej do dodania nowego pobytu po wpisaniu nr ID pacjenta, który jest już prowadzony przez innego lekarza (źródło własne)

Lekarz wpisał nr ID pacjenta, który znajduje się w bazie danych i który nie jest prowadzony przez innego lekarza. Lekarz wypełnił formularz – rys. 52.

[Anuluj dodawanie](#)

**Dane pacjenta o nr ID: 1016**

Imię: **Jadwiga**  
Nazwisko: **Królewicz**  
PESEL: **59110926922**  
Data urodzenia: **1959-11-09**  
Historia pacjenta: **Brak w systemie**

**Dołącz dane związane z pobytem**

ID Pacjenta:

1016

Oddział:

Oddział Chorób Wewnętrznych i Hematologii

Rodzaj schorzenia:

ICD-I - Choroby układu krążenia

Początkowe uwagi:

Pacjentka ma stwierdzone nadciśnienie tętnicze i przyjmuje leki.

[DODAJ POBYT](#)

Rys. 52. – PU-11 Formularz dodawania pobytu pacjenta (źródło własne)

Pobyt pacjenta został dodany, co potwierdza komunikat – rys. 53

**Panel lekarz**  
Zalogował się lekarz: Anna Nowacka, ID: 104  
[Wyloguj się](#)

[Dodane pobyty](#)   [Lista lekarzy](#)   [Wyświetl i zmień swoje dane](#)   [Wyświetl daty ostatnich logowań](#)   [Zmień hasło](#)

**Nowy pobyt pacjenta został dodany pomyślnie. ID Pacjenta: 1016, ID Pobytu: 10004**

**Lista Twoich pacjentów**

| ID Pacjenta | Imię i nazwisko   | PESEL       | Data rozpoczęcia pobytu | ID Pobytu | Działania  |
|-------------|-------------------|-------------|-------------------------|-----------|--|
| 1004        | Halina Nowak      | 78080937278 | 2023-12-11 22:44:42     | 10002     | <a href="#">Wyświetl informacje o pobycie pacjenta</a><br><a href="#">Wyświetl i zmień dane pacjenta</a> |
| 1016        | Jadwiga Królewicz | 59110926922 | 2023-12-13 15:13:57     | 10004     | <a href="#">Wyświetl informacje o pobycie pacjenta</a><br><a href="#">Wyświetl i zmień dane pacjenta</a> |

Rys. 53. PU-11 – Widok strony głównej lekarza po dodaniu pobytu pacjenta (źródło własne)

## PU-12 – Wyświetlenie informacji o pobycie pacjenta

Lekarz wybrał opcję działania „Wyświetl informacje o pobycie pacjenta”, znajdującą się na liście prowadzonych pacjentów – rys. 54.

The screenshot shows the 'Panel lekarz' (Doctor Panel) interface. At the top, it says 'Zalogował się lekarz: Anna Nowacka, ID: 104'. Below this is a 'Wyloguj się' button. A link 'Wróć do strony głównej' is also present. A message states: 'Pobyt nr ID: 10004. Ostatnia zmiana danych pobytu: Brak aktualizacji informacji związanych z pobytem'. Below this, instructions are given: 'Jeżeli chcesz zaktualizować informacje związane z pobytem Pacjenta, kliknij przycisk Aktualizuj informacje o pobycie. Możesz anulować dodawanie nowych informacji, klikając przycisk Anuluj. Jeżeli chcesz wypisać pacjenta kliknij tutaj'. The main section is titled 'Dane związane z pobytem' and contains two tables. The first table has columns: ID Pobytu, ID Pacjenta, Imię, and Nazwisko. The second table has columns: Data rozp. pobytu, ID Lekarza, ID Oddziału, and ID Schorzenia. To the right, there is a box titled 'Historia pobytu nr ID: 10004' containing details about the start of the stay, patient and doctor IDs, department, and ICD code, followed by a note about the patient's condition and a 'KONIEC WPISU' (End of entry) message. At the bottom, there is an 'Aktualizuj informacje o pobycie' button.

| ID Pobytu | ID Pacjenta | Imię    | Nazwisko  |
|-----------|-------------|---------|-----------|
| 10004     | 1016        | Jadwiga | Królewicz |

| Data rozp. pobytu   | ID Lekarza | ID Oddziału | ID Schorzenia |
|---------------------|------------|-------------|---------------|
| 2023-12-13 15:13:57 | 104        | O-CHW       | ICD-I         |

Rys. 54. PU-12 – Widok strony z informacjami o pobycie pacjenta (źródło własne)

## PU-13 – Aktualizacja informacji o pobycie pacjenta

Lekarz kliknął przycisk „Aktualizuj informacje o pobycie” na stronie z informacjami o pobycie pacjenta. Lekarz wypełnił formularz – rys. 55.

The screenshot shows the 'Aktualizuj informacje związane z pobytem pacjenta' (Update patient stay information) form. At the top right is an 'Anuluj' button. The form has two sections. The first section is titled 'Aktualizuj informacje związane z pobytem pacjenta' and contains two checkboxes: 'Chcę zmienić oddział pacjenta' (checked) and 'Chcę zmienić schorzenie pacjenta' (unchecked). Below the first checkbox is a dropdown menu for 'Oddział' with 'Oddział Kardiologiczny' selected. Below the second checkbox is a dropdown menu for 'Rodzaj schorzenia' with 'Wybierz schorzenie' selected. The second section is titled 'Dodaj nowe informacje, uwagi:' and contains a text area with the following text: 'U pacjentki zdiagnozowano problemy z sercem, związane z nadciśnieniem tętniczym. Konieczna zmiana oddziału na Kardiologię.' At the bottom of the form is a large blue button labeled 'AKTUALIZUJ POBYT'.

Rys. 55. PU-13 – Formularz aktualizacji informacji o pobycie pacjenta (źródło własne)

Historia pobytu na stronie z informacjami o pobycie pacjenta zostaje zaktualizowana – rys. 56

Pobyt nr ID: 10004. Ostatnia zmiana danych pobytu: 2023-12-13 15:58:39

Jezeli chcesz zaktualizować informacje związane z pobytem Pacjenta, kliknij przycisk **Aktualizuj informacje o pobycie**.  
 Możesz anulować dodawanie nowych informacji, klikając przycisk **Anuluj**.  
 Jezeli chcesz wypisać pacjenta kliknij **tutaj**.

**Dane związane z pobytem**

| ID Pobytu | ID Pacjenta | Imię    | Nazwisko  |
|-----------|-------------|---------|-----------|
| 10004     | 1016        | Jadwiga | Królewicz |

| Data rozp. pobytu   | ID Lekarza | ID Oddziału | ID Schorzenia |
|---------------------|------------|-------------|---------------|
| 2023-12-13 15:13:57 | 104        | O-KAR       | ICD-I         |

[Aktualizuj informacje o pobycie](#)

**Historia pobytu nr ID: 10004**

----- POCZĄTEK POBYTU NR: 10004, DATA ROZPOCZĘCIA: 2023-12-13 15:13:57 -----  
 Nr ID Pacjenta: 1016, Nr ID Lekarza: 104, Oddział: O-KAR, ICD-10: ICD-I  
 Początkowe uwagi: Pacjentka ma stwierdzone nadciśnienie tętnicze i przyjmuje leki.  
 ----- KONIEC WPISU -----

----- AKTUALIZACJA POBYTU NR: 10004, DATA AKTUALIZACJI: 2023-12-13 15:58:39 -----  
 Nr ID Pacjenta: 1016, Nr ID Lekarza: 104, Oddział: O-KAR, ICD-10: ICD-I  
 Nowe informacje: U pacjentki zdiagnozowano problemy z sercem, związane z nadciśnieniem tętniczym. Konieczna zmiana oddziału na Kardiologię.  
 Brak zmiany schorzenia.  
 Zmieniono oddział: O-KAR  
 ----- KONIEC WPISU -----

Rys. 56. PU-13 – Widok strony z informacjami o pobycie pacjenta po dokonanej aktualizacji (źródło własne)

## PU-14 – Wypisanie pacjenta ze szpitala

Lekarz wybrał opcję „kliknij tutaj” ze strony z informacjami o pobycie pacjenta, oznaczającą przejście do strony, na której potwierdza się wypis pacjenta ze szpitala. Lekarz wypełnia pole „Podsumowanie pobytu” – rys. 57.

**Panel lekarz**  
 Zalogował się lekarz: Anna Nowacka, ID: 104  
[Wyloguj się](#)

[Wróć do karty pobytu](#)

Pobyt nr ID: 10004. Ostatnia zmiana danych pobytu: 2023-12-13 15:58:39

Jezeli chcesz wypisać pacjenta o poniższych danych, kliknij przycisk **Wypisz pacjenta**. **Operacja wypisu pacjenta jest ostateczna.**  
 Akt wypisu pacjenta z podsumowaniem pobytu, który możesz dodać do aktu w polu poniżej, będzie dostępny przez 48 godzin od wypisu pacjenta.  
**Historia obecnego pobytu, po wypisaniu pacjenta, zostanie zapisana w historii pacjenta związanej z kontem pacjenta w systemie.**  
 Możesz wrócić do karty pobytu, klikając odnośnik [Wróć do karty pobytu](#).

**Czy chcesz wypisać pacjenta o poniższych danych?**

ID Pacjenta: 1016  
 ID Pobytu: 10004  
 Data rozpoczęcia pobytu: 2023-12-13 15:13:57  
 Imię: **Jadwiga**  
 Nazwisko: **Królewicz**  
 PESEL: 69110926922

Podsumowanie pobytu:

Pacjentka powinna oszczędzać się przez najbliższe 14 dni, zostać w domu i przyjmować leki.  
 W przypadku pogorszenia stanu zdrowia, pacjentka powinna jak najszybciej wrócić do szpitala.

**WYPISZ PACJENTA**

Rys. 57. PU-14 – Widok strony na której lekarz potwierdza wypis pacjenta ze szpitala (źródło własne)

Pacjent został wypisany, co potwierdza komunikat – rys. 58.

Udało się poprawnie wypisać pacjenta o nr ID: 1016

Lista Twoich pacjentów

| ID Pacjenta | Imię i nazwisko | PESEL       | Data rozpoczęcia pobytu | ID Pobytu | Działania  |
|-------------|-----------------|-------------|-------------------------|-----------|--|
| 1004        | Halina Nowak    | 78080937278 | 2023-12-11 22:44:42     | 10002     | <a href="#">Wyświetl informacje o pobycie pacjenta</a><br><a href="#">Wyświetl i zmień dane pacjenta</a> |

Rys. 58. PU-14 – Widok strony głównej lekarza po wypisaniu pacjenta ze szpitala  
(źródło własne)

### PU-15 – Wyświetlenie aktu wypisu pacjenta

Lekarz wybrał zakładkę „Ostatnio zakończone pobyty” ze strony głównej – rys. 59.

Panel lekarz

Zalogował się lekarz: Anna Nowacka, ID: 104

Wyloguj się



[Wróć do strony głównej](#)

Lista zakończonych pobytów w ostatnich 48 godzinach

| ID Pobytu | ID Pacjenta | Data rozpoczęcia pobytu | Data zakończenia pobytu | Działania                                    |
|-----------|-------------|-------------------------|-------------------------|--|
| 10004     | 1016        | 2023-12-13 15:13:57     | 2023-12-13 16:18:33     | <a href="#">Wyświetl akt wypisu pacjenta</a> |

Rys. 59. PU-15 – Widok strony z listą zakończonych pobytów w ostatnich 48 godzinach  
(źródło własne)

Lekarz wybrał opcję działania „Wyświetl akt wypisu pacjenta”, znajdującą się na liście zakończonych pobytów – rys. 60.

1 / 1 | - 80% + |  

## Akt wypisu pacjenta

### Dane osobowe pacjenta

Nr ID Pacjenta: 1016

Imię: Jadwiga

Nazwisko: Królewicz

PESEL: 59110926922

Data urodzenia: 1959-11-09

### Dane lekarza prowadzącego

Imię: Anna

Nazwisko: Nowacka

Specjalność: Choroby wewnętrzne

### Pobyt nr ID: 10004

Data rozpoczęcia pobytu: 2023-12-13 15:13:57

Oddział: Oddział Kardiologiczny

Schorzenie (wg ICD-10):

Choroby układu krążenia

Data zakończenia pobytu: 2023-12-13 16:18:33

Podsumowanie pobytu:

Pacjentka powinna oszczędzać się przez najbliższe 14 dni, zostać w domu i przyjmować leki.

W przypadku pogorszenia stanu zdrowia, pacjentka powinna jak najszybciej wrócić do szpitala.

Rys. 60. PU-15 – Widok aktu wypisu pacjenta w formie pliku .pdf (źródło własne)

## PU-16 – Wyświetlenie danych osobowych i kontaktowych pacjenta

Lekarz wybrał opcję działania „Wyświetl i zmień dane pacjenta”, znajdującą się na liście prowadzonych pacjentów – rys. 61

### Panel lekarz

Zalogował się lekarz: Anna Nowacka, ID: 104

[Wyloguj się](#)

[Wróć do strony głównej](#)

Dane pacjenta nr ID: 1004. Ostatnia zmiana danych: 2023-12-13 16:51:03

Jeżeli chcesz zmienić dane kontaktowe pacjenta (adres zamieszkania, email, telefon), kliknij przycisk [Chcę zmienić dane pacjenta](#).  
Możesz wycofać się z operacji zmiany danych, klikając przycisk [Anuluj](#).  
Pola z gwiazdką oznaczają pola konieczne do wypełnienia. Pola do wypełnienia bez gwiazdki - jeżeli nie znasz ich wartości - pozostaw puste.

[Chcę zmienić dane pacjenta](#)

#### Dane osobowe

| ID Pacjenta | ID Lekarza prowadzącego | ID obecnego pobytu | Data dodania do systemu | Historia medyczna pacjenta |
|-------------|-------------------------|--------------------|-------------------------|----------------------------|
| 1004        | 104                     | 10002              | 2023-11-08 12:53:27     | Pierwszy pobyt             |

| Imię   | Nazwisko | PESEL       | Obywatelstwo | Data urodzenia | Miejsce urodzenia |
|--------|----------|-------------|--------------|----------------|-------------------|
| Halina | Nowak    | 78080937278 | polskie      | 1978-08-09     | Rzeszów           |

#### Dane kontaktowe

| Ulica*        | Nr domu* | Nr mieszkania |
|---------------|----------|---------------|
| Łipowa        | 14       |               |
| Kod pocztowy* | Miasto*  | Województwo*  |
| 36-704        | Kraśków  | Małopolskie   |

| Telefon    | E-mail     |
|------------|------------|
| Nie podano | Nie podano |

Rys. 61. PU-16 – Widok danych pacjenta (źródło własne)

W przypadku pacjenta, który w przeszłości przebywał już w szpitalu, lekarz ma możliwość wyświetlenia na stronie z danymi pacjenta jego historii medycznej, obejmującej historie poszczególnych, zakończonych pobyków pacjenta – rys. 62.

**Historia medyczna pacjenta nr ID: 1016**

[Zamknij widok](#)

**Historia pobytu nr ID: 10004**

----- POCZĄTEK POBYTU NR: 10004, DATA ROZPOCZĘCIA: 2023-12-13 15:13:57 -----

Nr ID Pacjenta: 1016, Nr ID Lekarza: 104, Oddział: O-CHW, ICD-10: ICD-I

Początkowe uwagi: Pacjentka ma stwierdzone nadciśnienie tętnicze i przyjmuje leki.

----- KONIEC WPISU -----

----- AKTUALIZACJA POBYTU NR: 10004, DATA AKTUALIZACJI: 2023-12-13 15:58:39 -----

Nr ID Pacjenta: 1016, Nr ID Lekarza: 104, Oddział: O-KAR, ICD-10: ICD-I

Nowe informacje: U pacjentki zdiagnozowano problemy z sercem, związane z nadciśnieniem tętniczym. Konieczna zmiana oddziału na Kardiologię.

Brak zmiany schorzenia.

Zmieniono oddział: O-KAR

----- KONIEC WPISU -----

----- ZAKOŃCZENIE POBYTU NR: 10004, DATA ZAKOŃCZENIA: 2023-12-13 16:18:33 -----

Nr ID Pacjenta: 1016, Nr ID Lekarza: 104

Podsumowanie pobytu: Pacjentka powinna oszczędzać się przez najbliższe 14 dni, zostać w domu i przyjmować leki.

W przypadku pogorszenia stanu zdrowia, pacjentka powinna jak najszybciej wrócić do szpitala.

----- KONIEC WPISU -----

Rys. 62. PU-16 – Widok historii medycznej pacjenta (źródło własne)

**Uwaga:**

*Dane pacjenta (ale bez jego historii medycznej) może wyświetlić również rejestrator.*

**PU-17 – Zmiana danych kontaktowych pacjenta**

Lekarz kliknął przycisk „Chcę zmienić dane pacjenta” na stronie z danymi pacjenta i uzupełnił pole „E-mail” – rys. 63.



Dane pacjenta nr ID: 1004. Ostatnia zmiana danych: 2023-12-13 17:03:56

Jeżeli chcesz zmienić dane kontaktowe pacjenta (adres zamieszkania, email, telefon), kliknij przycisk *Chcę zmienić dane pacjenta*.  
 Możesz wycofać się z operacji zmiany danych, klikając przycisk *Anuluj*.  
 Pola z gwiazdką oznaczają pola konieczne do wypełnienia. Pola do wypełnienia bez gwiazdki - jeżeli nie znasz ich wartości - pozostaw puste.

[Anuluj](#)

**Dane osobowe**

| ID Pacjenta | ID Lekarza prowadzącego | ID obecnego pobytu | Data dodania do systemu | Historia medyczna pacjenta |
|-------------|-------------------------|--------------------|-------------------------|----------------------------|
| 1004        | 104                     | 10002              | 2023-11-08 12:53:27     | Pierwszy pobyt             |

| Imię   | Nazwisko | PESEL       | Obywatelstwo | Data urodzenia | Miejsce urodzenia |
|--------|----------|-------------|--------------|----------------|-------------------|
| Halina | Nowak    | 78080937278 | polskie      | 1978-08-09     | Rzeszów           |

**Dane kontaktowe**

| Ulica* | Nr domu* | Nr mieszkania |
|--------|----------|---------------|
| Lipowa | 14       |               |

| Kod pocztowy* | Miasto* | Województwo* |
|---------------|---------|--------------|
| 30-704        | Kraków  | Małopolskie  |

| Telefon | E-mail                   |
|---------|--------------------------|
|         | halinanowak78@interia.pl |

[ZMIEN DANE PACJENTA](#)

Rys. 63. PU-17 – Widok zmiany danych pacjenta (źródło własne)

Na stronie głównej pojawia się komunikat o zmianie danych pacjenta – rys. 64.

Dane pacjenta o nr ID: 1004 zostały zmienione pomyślnie.

**Lista Twoich pacjentów**

| ID Pacjenta | Imię i nazwisko   | PESEL       | Data rozpoczęcia pobytu | ID Pobytu | Działania  |
|-------------|-------------------|-------------|-------------------------|-----------|--|
| 1004        | Halina Nowak      | 78080937278 | 2023-12-11 22:44:42     | 10002     | <a href="#">Wyświetl informacje o pobycie pacjenta</a><br><a href="#">Wyświetl i zmień dane pacjenta</a> |
| 1016        | Jadwiga Królewicz | 59110926922 | 2023-12-13 16:53:19     | 10005     | <a href="#">Wyświetl informacje o pobycie pacjenta</a><br><a href="#">Wyświetl i zmień dane pacjenta</a> |

Rys. 64. PU-17 – Widok strony głównej lekarza po zmianie danych pacjenta (źródło własne)

**Uwaga:**

*Dane kontaktowe pacjenta może zmienić również rejestrator.*

## 6. Podsumowanie

Celem pracy dyplomowej było utworzenie bezpiecznego środowiska serwera szpitalnego oraz napisanie aplikacji webowej, która umożliwi pracę na danych zgromadzonych w serwerze szpitalnym.

Cel pracy nie zostałby osiągnięty bez wyboru i zaimplementowania odpowiedniego zestawu oprogramowania tworzącego środowisko serwera szpitalnego oraz napisania aplikacji webowej przeznaczonej dla lekarzy i rejestratorów pracujących w placówce szpitalnej. Wybrany zestaw oprogramowania LAMP pozwolił na bezproblemowe skonfigurowanie na potrzeby pracy dyplomowej systemu operacyjnego Debian, serwera HTTP Apache oraz systemu zarządzania bazą danych MariaDB. Konfiguracja poszczególnych komponentów oprogramowania serwera pozwoliła m.in. na zablokowanie, za pomocą systemowego firewalla, dostępu do serwera szpitalnego nieuprawnionym urządzeniom, ustanowienie bezpiecznej komunikacji klient-serwer za pomocą protokołu HTTPS oraz zaimplementowanie projektu bazy danych „Szpital”, wraz z przechowywaniem danych w bazie w zaszyfrowanej formie. Bezpieczeństwo pracy serwera zwiększa także codzienne, automatyczne wykonywanie kopii zapasowych bazy danych „Szpital” oraz plików przechowywanych przez serwer HTTP Apache. Zestaw oprogramowania LAMP okazał się więc bardzo dobrym rozwiązaniem do utworzenia bezpiecznego środowiska serwera szpitalnego.

Napisana aplikacja webowa spełnia wszystkie postawione w pracy wymagania funkcjonalne i нефункционалне, umożliwiając lekarzom i rejestratorom bezpieczną i sprawną pracę z danymi osobowymi i medycznymi pacjentów. Aplikacja łączy się z bazą danych „Szpital”, deszyfruje dane zgromadzone w bazie i szyfruje je przed zapisaniem w bazie. *Prepared statements*, zawarte w kodzie aplikacji, zabezpieczają bazę danych i informacje w niej zawarte przed nieautoryzowanym dostępem w wyniku ataku typu SQL Injection. Przypadki użycia aplikacji zostały przetestowane i działają poprawnie, zgodnie z zadeklarowanymi ścieżkami głównymi i alternatywnymi.

Aplikacja webowa może zostać w przyszłości rozwinięta o kolejne funkcjonalności. Bezpieczeństwo logowania do aplikacji może zostać zwiększone przez zaimplementowanie dwustopniowej autentyfikacji, składającej się np. z hasła oraz kodu

przychodzącego na numer telefonu logującego się użytkownika. Aplikacja mogłaby także umożliwiać w przyszłości wyświetlanie i zapisywanie w bazie danych obrazów, np. zdjęć rentgenowskich. Ciekawą, nową funkcjonalnością aplikacji byłaby również możliwość wysyłania wiadomości do innych użytkowników aplikacji. Należałoby przy tym rozważyć, czy taka funkcjonalność miałaby przypominać komunikację za pośrednictwem wiadomości e-mail, czy też konwersacje „dynamiczną”, jak w popularnych komunikatorach internetowych. Wymienione wyżej propozycje nowych funkcjonalności aplikacji webowej udowadniają, że posiada ona spory potencjał do dalszego rozwoju.

## Bibliografia

- [1] Najwyższa Izba Kontroli, „RODO w szpitalu” 14 Listopad 2019. [Online]. Dostępny w internecie: <https://www.nik.gov.pl/aktualnosci/rodo-w-szpitalu.html>. [Data uzyskania dostępu: 2 Styczeń 2024].
- [2] S. Simic, „What is LAMP Stack?” 6 Styczeń 2022. [Online]. Dostępny w internecie: <https://phoenixnap.com/kb/what-is-a-lamp-stack>. [Data uzyskania dostępu: 2 Grudzień 2023].
- [3] M. Bosko, „MEAN Vs. LAMP: Which is Better” 16 Marzec 2021. [Online]. Dostępny w internecie: <https://phoenixnap.com/kb/mean-vs-lamp>. [Data uzyskania dostępu: 2 Grudzień 2023].
- [4] Amazon Web Services, „What’s the Difference Between MariaDB and MySQL?” [Online]. Dostępny w internecie: <https://aws.amazon.com/compare/the-difference-between-mariadb-vs-mysql/>. [Data uzyskania dostępu: 2 Grudzień 2023].
- [5] Wikipedia - wolna encyklopedia, „Linux” [Online]. Dostępny w internecie: <https://pl.wikipedia.org/wiki/Linux>. [Data uzyskania dostępu: 2 Grudzień 2023].
- [6] OSnews - serwis poświęcony nowym technologiom, „Londyńska giełda od dziś oficjalnie na Linuksie” [Online]. Dostępny w internecie: <http://osnews.pl/londyńska-giełda-od-dzis-oficjalnie-na-linuksie/>. [Data uzyskania dostępu: 3 Grudzień 2023].
- [7] Przystajnik - blog o Linuksie, „RTG, DICOM i Linux z pomocą” 13 Kwiecień 2016. [Online]. Dostępny w internecie: <https://404.g-net.pl/2016/04/fotografia-uzyteczna-rtg-dicom/>. [Data uzyskania dostępu: 3 Grudzień 2023].

- [8] OSnews - serwis poświęcony nowym technologiom, „Amerykańska marynarka wojenna stawia na Linuksa” [Online]. Dostępny w internecie: <http://osnews.pl/amerykanska-marynarka-wojenna-stawia-na-linuksa/>. [Data uzyskania dostępu: 3 Grudzień 2023].
- [9] Wikipedia - wolna encyklopedia w j. ang., „Debian” [Online]. Dostępny w internecie: <https://en.wikipedia.org/wiki/Debian>. [Data uzyskania dostępu: 3 Grudzień 2023].
- [10] Oficjalna strona systemu Debian, „Our Philosophy: Why we do it and how we do it” [Online]. Dostępny w internecie: <https://www.debian.org/intro/philosophy>. [Data uzyskania dostępu: 3 Grudzień 2023].
- [11] Oficjalna strona systemu Debian, „Reasons to use Debian” [Online]. Dostępny w internecie: [https://www.debian.org/intro/why\\_debian.en.html](https://www.debian.org/intro/why_debian.en.html). [Data uzyskania dostępu: 3 Grudzień 2023].
- [12] Oficjalna strona systemu Debian, „Security Information” [Online]. Dostępny w internecie: <https://www.debian.org/security/index.en.html>. [Data uzyskania dostępu: 3 Grudzień 2023].
- [13] Oficjalna strona serwera Apache, [Online]. Dostępny w internecie: <https://httpd.apache.org>. [Data uzyskania dostępu: 3 Grudzień 2023].
- [14] R. Góra, „Serwer Apache – co to jest?” [Online]. Dostępny w internecie: <https://www.lh.pl/pomoc/apache-co-to-jest/>. [Data uzyskania dostępu: 3 Grudzień 2023].
- [15] Blog netcraft, „May 2020 Web Server Survey” 26 Maj 2020. [Online]. Dostępny w internecie: <https://www.netcraft.com/blog/may-2020-web-server-survey/>. [Data uzyskania dostępu: 3 Grudzień 2023].

- [16] Wikipedia - wolna encyklopedia, „HTTPS” [Online]. Dostępny w internecie: <https://pl.wikipedia.org/wiki/HTTPS>. [Data uzyskania dostępu: 3 Grudzień 2023].
- [17] K. Kroc, O. Kizun i M. Skublewska-Paszkowska, „Analiza porównawcza wydajności relacyjnych baz danych MySQL, PostgreSQL, MariaDB oraz H2” *Journal of Computer Sciences Institute*, 2020.
- [18] Z. Rudnicki, „Relacyjne systemy baz danych” [Online]. Dostępny w internecie: [http://www.kkiem.agh.edu.pl/sites/all/old\\_kkiem/dydakt/Wyklady/8\\_Bazy\\_2017.pdf](http://www.kkiem.agh.edu.pl/sites/all/old_kkiem/dydakt/Wyklady/8_Bazy_2017.pdf). [Data uzyskania dostępu: 3 Grudzień 2023].
- [19] Wikipedia - wolna encyklopedia w j. ang., „MariaDB” [Online]. Dostępny w internecie: <https://en.wikipedia.org/wiki/MariaDB>. [Data uzyskania dostępu: 3 Grudzień 2023].
- [20] T. Kozon, „MariaDB – system zarządzania relacyjną bazą danych” 13 Styczeń 2023. [Online]. Dostępny w internecie: <https://boringowl.io/blog/mariadb>. [Data uzyskania dostępu: 3 Grudzień 2023].
- [21] Oficjalna strona MariaDB, „Database Account Two-Factor Authentication (2FA)” [Online]. Dostępny w internecie: <https://mariadb.com/docs/skysql-previous-release/service-management/options/database-account-2fa/>. [Data uzyskania dostępu: 3 Grudzień 2023].
- [22] w3 Techs - Web Technology Surveys, „Usage statistics of PHP for websites” [Online]. Dostępny w internecie: <https://w3techs.com/technologies/details/pl-php>. [Data uzyskania dostępu: 3 Grudzień 2023].
- [23] D. Stlemach, „Uprawnienia do plików i folderów” [Online]. Dostępny w internecie: <https://pasja-informatyki.pl/sieci-komputerowe/ubuntu-server-uprawnienia-plikow-folderow/>. [Data uzyskania dostępu: 7 Grudzień 2023].

- [24] J. Otieno, „How Do I Enable HTTPS On Apache Web Server” [Online]. Dostępny w internecie: <https://linuxhint.com/enable-https-apache-web-server/>. [Data uzyskania dostępu: 8 Grudzień 2023].
- [25] Spacerex, „How to Automatically Backup a MySQL or MariaDB Server with MySQLDump” 11 Kwiecień 2023. [Online]. Dostępny w internecie: <https://www.spacerex.co/how-to-automatically-backup-a-mysql-or-mariadb-server-with-mysqldump/>. [Data uzyskania dostępu: 7 Grudzień 2023].
- [26] J. Różańska, „Co to jest SQL Injection?” 11 Czerwiec 2023. [Online]. Dostępny w internecie: <https://nordvpn.com/pl/blog/sql-injection-co-to/>. [Data uzyskania dostępu: 10 Grudzień 2023].
- [27] Meridian Outpost - Digital Business & IT Support, „How to Encrypt and Decrypt Data in PHP” [Online]. Dostępny w internecie: <https://www.meridianoutpost.com/resources/articles/programming/PHP-how-to-encrypt-decrypt-data.php>. [Data uzyskania dostępu: 11 Grudzień 2023].