

PROJEKT Z BAZ DANYCH 2

Bazodanowy system obsługi sprzętu komputerowego w firmie

AUTORZY:

Michał Wróbel

Indeks: 259132

E-mail:

259132@student.pwr.edu.pl

Kacper Tomkiewicz

Indeks: 262752

E-mail:

262752@student.pwr.edu.pl

PROWADZĄCY ZAJĘCIA:

Dr inż. Robert Wójcik, K30W04D0

OCENA PRACY:

Wrocław, 2023

Spis treści

Spis rysunków	4
Spis tabel	5
1 Wstęp	6
1.1 Cel projektu	6
1.2 Zakres projektu.....	6
2 Analiza wymagań	7
2.1 Opis działania i schemat logiczny systemu.....	7
2.2 Wymagania funkcjonalne	7
2.2.1 Diagramy przypadków użycia	8
2.2.2 Scenariusze przypadków użycia	8
2.3 Wymagania niefunkcjonalne	12
2.3.1 Wykorzystywane technologie i narzędzia	13
2.3.2 Wymagania dotyczące bezpieczeństwa systemu	13
2.4 Przyjęte założenia projektowe.....	13
3 Projekt systemu	14
3.1 Projekt bazy danych	14
3.1.1 Model fizyczny	14
3.1.2 Ograniczenia integralności danych	14
3.2 Projekt aplikacji użytkownika	16
3.2.1 Architektura aplikacji oraz metoda podłączenia do bazy danych.	16
3.2.2 Interfejs graficzny i struktura menu	16
4 Implementacja systemu	17
4.1 Realizacja bazy danych.....	17
4.1.1 Tworzenie tabel i definiowanie ograniczeń.....	17
4.1.2 Implementacja mechanizmów przetwarzania danych	18
4.1.3 Implementacja uprawnień i innych zabezpieczeń	18
4.2 Realizacja elementów aplikacji.....	18
4.2.1 Obsługa menu.....	18
4.2.2 Walidacja i filtracja.....	19
4.2.3 Implementacja interfejsu dostępu do bazy danych	20
4.2.4 Implementacja wybranych funkcjonalności systemu	20
4.2.5 Implementacja mechanizmów bezpieczeństwa.....	21
5 Testowanie systemu.....	22
5.1 Testowanie opracowanych funkcji systemu	22
5.1.1 Testowanie funkcji odpowiedzialnej za wyświetlenie listy użytkowników	22
5.1.2 Test funkcji odpowiedzialnej za przypisanie sprzętu do użytkownika	23

5.1.3	Test funkcji odpowiedzialnej za wyszukiwanie danych użytkownika	26
5.2	Testowanie mechanizmów bezpieczeństwa	28
5.3	Wnioski z testów	28
6	Podsumowanie	29

Spis rysunków

Rysunek 1 Schemat komunikacji, struktura systemu	6
Rysunek 2 Diagram przypadków użycia	8
Rysunek 3 Model fizyczny stworzonej bazy danych	14
Rysunek 4 Widok głównego Menu aplikacji po zalogowaniu jako administrator	16
Rysunek 5 Widok głównego Menu aplikacji po zalogowaniu jako użytkownik	17
Rysunek 6 Ekran logowania do aplikacji obsługującej bazę danych	17
Rysunek 7 Implementacja kodu służącego do stworzenia Tabeli przypisany_sprzet wraz z ograniczeniami	17
Rysunek 8 Implementacja kodu Triggera obsługujące próbę przypisania nieistniejącego sprzętu lub przypisania do nieistniejącego użytkownika	18
Rysunek 9 Implementacja kodu obsługującego przypisanie funkcji do wybranych przykładowych przycisków oraz widocznych w menu funkcji	18
Rysunek 10 Implementacja kodu odpowiedzialnego za działanie logowania do aplikacji	19
Rysunek 11 Implementacja kodu odpowiedzialnego za wytworzenie połączenia między bazą danych a aplikacją	20
Rysunek 12 Implementacja kodu obsługującego funkcję wyświetlania listy akcesoriów	20
Rysunek 13 Implementacja kodu obsługującego przypisanie sprzętu do użytkownika	21
Rysunek 14 Implementacja kodu odpowiedzialnego za sprawdzenie danych logowania	21
Rysunek 15 Lista użytkowników widoczna w aplikacji	22
Rysunek 16 Lista użytkowników widoczna w bazie danych	22
Rysunek 17 Widok statusu sprzętu w aplikacji przed wykonaniem operacji	23
Rysunek 18 Lista sprzętu w firmie widoczna w bazie danych wraz ze statusem przed wykonaniem operacji	23
Rysunek 19 Widok wprowadzanych danych (identyfikator_wewnetrzny_uzytkownika) do wykonania operacji	23
Rysunek 20 Widok wprowadzanych danych (identyfikator_wewnetrzny_sprzetu) do wykonania operacji	24
Rysunek 21 Widok statusu sprzętu w aplikacji po wykonaniu operacji	24
Rysunek 22 Lista sprzętu w firmie widoczna w bazie danych wraz ze statusem po wykonaniu operacji	24
Rysunek 23 Widok menu po wyszukaniu użytkownika, na którym wykonano operację przypisania sprzętu	25
Rysunek 24 Widok tabeli korelacja_uzytkownika_ze_sprzetem po wykonaniu operacji	25
Rysunek 25 Widok danych wprowadzanych do wykonania operacji (identyfikator_wewnetrzny_uzytkownika)	26
Rysunek 26 Widok menu aplikacji po wykonaniu operacji	26
Rysunek 27 Dane wprowadzane w celu wykonania operacji (identyfikator_wewnetrzny_uzytkownika)	27
Rysunek 28 Widok menu aplikacji po wykonaniu operacji	27
Rysunek 29 Komunikat widoczny po wprowadzeniu niewłaściwych danych logowania	28
Rysunek 30 Komunikat widoczny po wprowadzeniu właściwych danych logowania	28

Spis tabel

Tabela 1 Tabela przedstawiająca encje występujące w bazie danych wraz z opisem	14
Tabela 2 Tabela przedstawiająca atrybuty w enzi uzytkownicy wraz z opisem.....	15
Tabela 3 Tabela przedstawiająca atrybuty w enzi sprzet wraz z opisem.....	15
Tabela 4 Tabela przedstawiająca atrybuty w enzi akcesoria wraz z opisem	15
Tabela 5 Tabela przedstawiająca atrybuty w enzi historia_operacji wraz z opisem	15
Tabela 6 Tabela przedstawiająca atrybuty w enzi przypisane_akcesoria wraz z opisem	15
Tabela 7 Tabela przedstawiająca atrybuty w enzi przypisany_sprzet wraz z opisem	15
Tabela 8 Tabela przedstawiająca atrybuty w enzi korelacja_uzytkownika_ze_sprzetem wraz z opisem	16

1 Wstęp

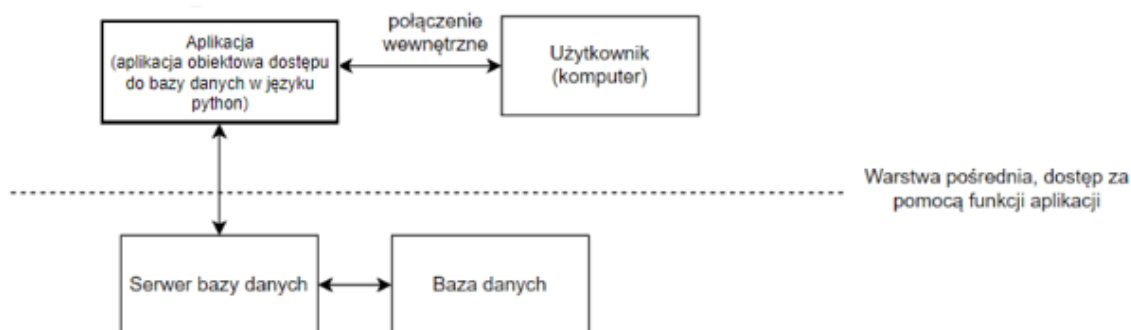
Zadanie projektowe polega na stworzeniu bazy danych wybranej przez autorów oraz zatwierdzonej przez prowadzącego, która miałaby realne oraz praktyczne zastosowanie w przedsiębiorstwie. W pracy należy przedstawić implementację systemu bazodanowego, umożliwiającego realizację funkcjonalności wynikających z określonego zastosowania (w tym przypadku jest to obsługa sprzętu komputerowego w firmie). Baza danych będzie obsługiwana za pośrednictwem bazy danych MySQL Workbench oraz systemu zarządzania np. Workbench uruchomionego lokalnie na komputerze. Interfejs zostanie zrealizowany w postaci aplikacji obiektowej w języku „Python” uruchamianej lokalnie na komputerze. Do specyfikacji funkcji systemu wykorzystany zostanie zunifikowany język modelowania UML. Każdy z przedstawionych etapów został uprzednio przedstawiony prowadzącemu i zaakceptowany po wprowadzeniu zadanych zmian.

1.1 Cel projektu

Temat: “Bazodanowy system obsługi sprzętu komputerowego w firmie”. Cel projektu: projekt oraz implementacja bazy danych oraz prostego interfejsu użytkownika przeznaczonych do obsługi sprzętu komputerowego w firmie.

1.2 Zakres projektu

Projekt ma umożliwiać zapisywanie danych o wypożyczonym/wydanym sprzęcie w przedsiębiorstwie średniej wielkości w oparciu o relacyjną bazę danych. Dostęp do lokalnej bazy danych będzie możliwy z poziomu aplikacji zainstalowanej na lokalnym komputerze. Do bazy danych będzie miał dostęp pracownik jako obserwator mogący filtrować bazę danych w celu znalezienia odpowiednich informacji oraz administrator jako edytujący mogący uzupełniać dane w aplikacji.



Rysunek 1 Schemat komunikacji, struktura systemu

2 Analiza wymagań

Wymagania zawarte w projekcie powinny uwzględniać możliwość przeglądania przez użytkownika (w tym przypadku pracownika firmy) bazy danych w zakresie dostępności sprzętu, jego ilości oraz jego parametrów, przypisanego do siebie sprzętu. Administrator powinien mieć umożliwione takie same funkcje jak pracownik dodatkowo rozszerzone o konkretne funkcjonalności takie jak np. edycja bazy danych. Administrator bazy danych powinien mieć możliwość przypisania danego sprzętu do konkretnego pracownika, usunięcie go, dodanie nowego sprzętu do bazy danych, jak i usunięcie nieużywanych już akcesoriów itd.

2.1 Opis działania i schemat logiczny systemu

System umożliwiać będzie zarządzanie sprzętem komputerowym w średnim przedsiębiorstwie w oparciu o relacyjną bazę danych (tabele opisujące dane o komputerach, np. adres mac, model, identyfikator, przypisany użytkownik itp.). Możliwy będzie dostęp do danych z poziomu aplikacji zainstalowanej na lokalnym komputerze (sama baza danych również będzie działała lokalnie), a także wykonywanie za pomocą tej aplikacji określonych operacji w zależności od rodzaju użytkownika (np. pracownik – wyszukiwanie użytkowników i sprzętów; administrator: zmiana danych istniejących użytkowników i sprzętów w systemie oraz dodawanie i usuwanie ich z bazy danych, przypisywanie użytkowników do sprzętu).

2.2 Wymagania funkcjonalne

Pracownik:

- Pracownik może przeglądać swoje dane w systemie
- Pracownik może prosić Administratora o edycję, dodanie lub usunięcie z bazy danych
- Pracownik może prosić o wygenerowanie nowego hasła do konta
- Pracownik może zalogować się na swoje konto pracownicze
- Pracownik może przeglądać sprzęt znajdujący się w bazie danych
- Pracownik może filtrować przeglądany sprzęt znajdujący się w bazie danych na podstawie wyboru konkretnych kategorii

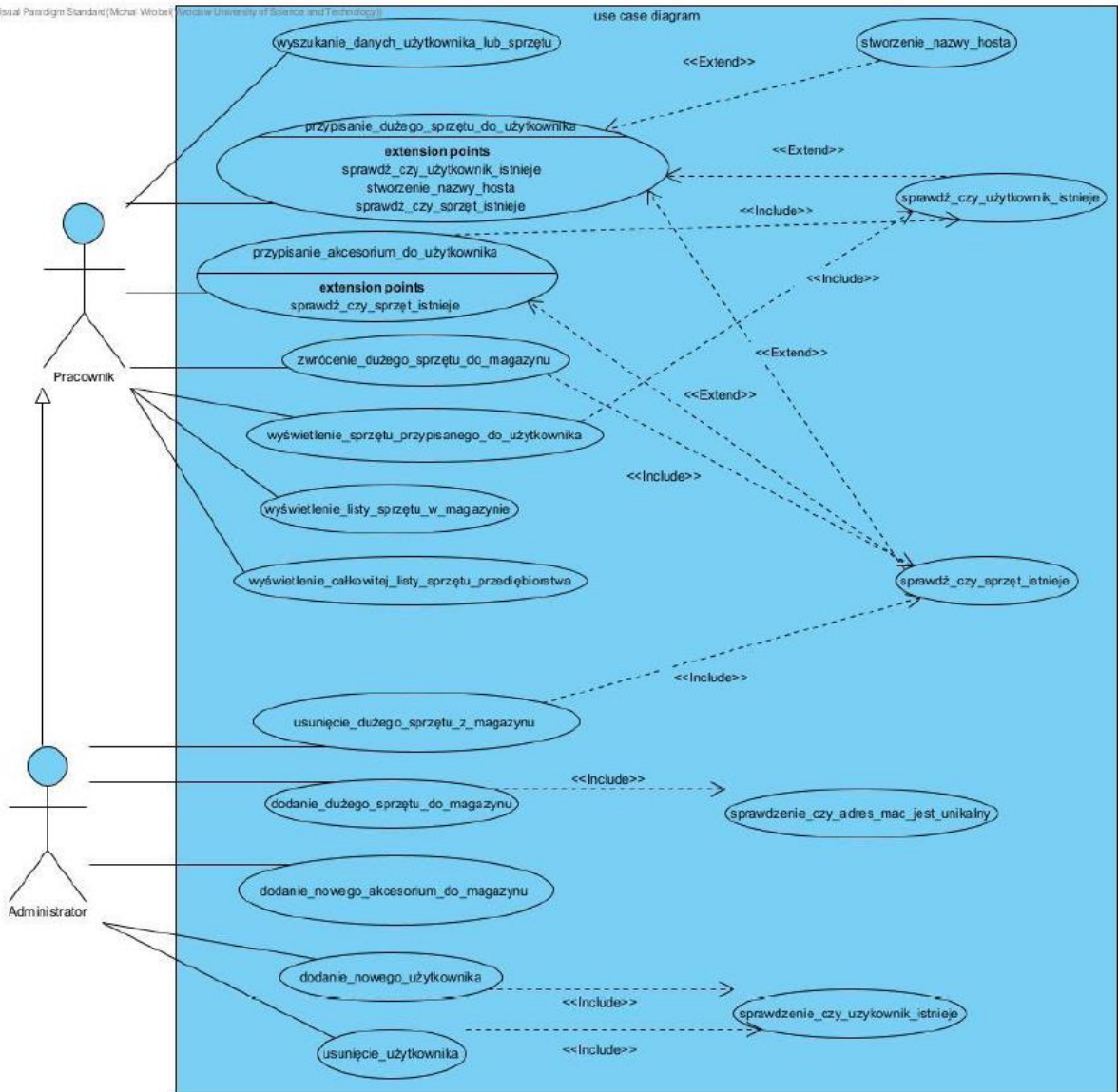
Administrator:

- Administrator może edytować dane Pracownika w bazie danych
- Administrator może usunąć/dodać sprzęt lub akcesoria w bazie danych
- Administrator może edytować dane o sprzęcie oraz akcesoriach w bazie danych
- Administrator może dodać lub usunąć Pracownika z bazy danych

System:

- System jest w stanie wygenerować identyfikator wewnętrzny Pracownika oraz sprzętu

Visual Paradigm Standard (Mohai Wobek) | Wobek University of Science and Technology



Rysunek 2 Diagram przypadków użycia

2.2.2 Scenariusze przypadków użycia

PU dodanie nowego_użytkownika

OPIS

CEL: dodanie nowego użytkownika, do którego można przypisać sprzęt w systemie
WS(warunki wstępne): możliwy do zainicjalizowania przez uruchomienie programu i
zalogowanie się w trybie pracownika

WK(warunki końcowe): Dodanie nowego użytkownika w systemie

PRZEBIEG:

1. Należy podać dane personalne Pracownika (imię, nazwisko oraz nr identyfikacyjny spełniający określone kryteria)
2. Należy wywołać PU sprawdzenie_czy_użytkownik_istnieje. Jeśli zwróci prawdę, należy zakończyć PU, w przeciwnym wypadku należy zapisać dane nowego użytkownika do tabeli.

PU wyszukanie_danych_uzytkownika_lub_sprzetu

OPIS

CEL: Wyszukanie interesujących danych użytkownika lub sprzętu w bazie danych

WS(warunki wstępne): Inicjalizacja przez Administratora lub Pracownika

WK(warunki końcowe): Wyświetlenie danych o szukanym użytkowniku lub sprzęcie

PRZEBIEG:

1. Należy podać dane, po których ma nastąpić filtrowanie użytkowników lub sprzętu.
2. Wyświetlenie krotek bazy danych z tabeli, które są zgodne z podanymi do wyszukania danymi.

PU przypisanie_duzego_sprzetu_do_uzytkownika

OPIS

CEL: przypisanie konkretnemu użytkownikowi dużego sprzętu, który został przez niego odebrany

WS(warunki wstępne): Inicjalizacja przez Administratora lub Pracownika

WK(warunki końcowe): przypisanie sprzętu o danym nr seryjnym do konkretnego użytkownika

PRZEBIEG:

1. Należy podać identyfikator_wewnetrzny_uzytkownika
2. Należy wywołać PU sprawdzenie_czy_uzytkownik_istnieje. Jeśli zwróci prawdę, należy przejść do pkt 3 . W przeciwnym wypadku należy przejść do pkt 6.
3. Należy wywołać PU stworzenie_nazwy_hosta.
4. Należy wywołać PU sprawdź_czy_sprzet_istnieje.
5. Należy stworzyć korelację między użytkownikiem a sprzętem oraz pominąć pkt 6
6. Należy wywołać PU dodanie_nowego_uzytkownika.

PU przypisanie_akcesorium_do_uzytkownika

OPIS

CEL: przypisanie konkretnemu użytkownikowi akcesorium, które zostało przez niego odebrane

WS(warunki wstępne): Inicjalizacja przez Administratora lub Pracownika

WK(warunki końcowe): przypisanie akcesorium do danego użytkownika

PRZEBIEG:

1. Należy podać identyfikator_wewnetrzny_uzytkownika
2. Należy wywołać PU sprawdzenie_czy_uzytkownik_istnieje. Jeśli zwróci prawdę, należy przejść do pkt 3 . W przeciwnym wypadku należy przejść do pkt 5.
3. Należy wywołać PU sprawdź_czy_sprzet_istnieje.
4. Należy stworzyć korelację między użytkownikiem a akcesorium oraz pominąć pkt 5
5. Należy wywołać PU dodanie_nowego_uzytkownika.

PU przypisanie_akcesorium_do_uzytkownika

OPIS

CEL: przypisanie konkretnemu użytkownikowi akcesorium, które zostało przez niego odebrane

WS(warunki wstępne): Inicjalizacja przez Administratora lub Pracownika lub Pracownika

WK(warunki końcowe): przypisanie akcesorium do danego użytkownika

PRZEBIEG:

1. Należy podać identyfikator_wewnetrzny_uzytkownika
2. Należy wywołać PU sprawdzenie_czy_uzytkownik_istnieje. Jeśli zwróci prawdę, należy przejść do pkt 3 . W przeciwnym wypadku należy przejść do pkt 5.
3. Należy wywołać PU sprawdź_czy_sprzet_istnieje.
4. Należy stworzyć korelację między użytkownikiem a akcesorium oraz pominąć pkt 5
5. Należy wywołać PU dodanie_nowego_uzytkownika.

PU zwrocenie_duzego_sprzetu_do_magazynu

OPIS

CEL: zwrócenie dużego sprzętu do magazynu i zmiana jego stanu w celu umożliwienia wypożyczenia go innemu użytkownikowi

WS(warunki wstępne): Inicjalizacja przez Administratora lub Pracownika

WK(warunki końcowe): zmiana stanu sprzętu na dostępny i możliwy do wypożyczenia przez innego użytkownika

PRZEBIEG:

1. Należy podać identyfikator_wewnętrzny_użytkownika
2. Należy wywołać PU sprawdź_czy_sprzet_istnieje.
3. Należy usunąć korelację między użytkownikiem a sprzętem
4. Należy zaktualizować cykl życia sprzętu
5. Należy zmienić status sprzętu na dostępny

PU wyświetlenie_sprzetu_przypisanego_do_użytkownika

OPIS

CEL: Wyświetlenie listy sprzętu przypisanego do danego użytkownika

WS(warunki wstępne): Inicjalizacja przez Administratora lub Pracownika

WK(warunki końcowe): wyświetlenie sprzętu przypisanego do wybranego użytkownika w postaci listy

PRZEBIEG:

1. Należy podać identyfikator_wewnętrzny_użytkownika
2. Należy wywołać PU sprawdzenie_czy_użytkownik_istnieje, jeśli zwróci prawdę, należy wyświetlić listę sprzętu przypisanego do użytkownika (listę korelacji użytkownika ze sprzętami)

PU wyświetlenie_listy_sprzetu_w_magazynie

OPIS

CEL: wyświetlenie sprzętu znajdującego się w magazynie

WS(warunki wstępne): Inicjalizacja przez Administratora lub Pracownika

WK(warunki końcowe): wyświetlenie listy sprzętu znajdującego się w magazynie

PRZEBIEG:

1. Wyświetlenie listy dostępnego sprzętu znajdującego się w magazynie

PU wyświetlenie_całkowitej_list_sprzetu_przedsiębiorstwa

OPIS

CEL: zwrócenie dużego sprzętu do magazynu i zmiana jego stanu w celu umożliwienia wypożyczenia go innemu użytkownikowi

WS(warunki wstępne): Inicjalizacja przez Administratora lub Pracownika

WK(warunki końcowe): zmiana stanu sprzętu na dostępny i możliwy do wypożyczenia przez innego użytkownika

PRZEBIEG:

1. Wyświetlenie listy sprzętu oraz akcesoriów dostępnych w przedsiębiorstwie

PU usunięcie_duzego_sprzetu_z_magazynu

OPIS

CEL: usunięcie danego sprzętu z bazy danych

WS(warunki wstępne): sprzęt istnieje w bazie danych

WK(warunki końcowe): usunięcie podanego sprzętu z listy bazy danych

PRZEBIEG:

1. Należy podać identyfikator_wewnętrzny_sprzetu
2. Należy wywołać PU sprawdzenie_czy_sprzet_istnieje, jeśli zwróci prawdę, należy usunąć sprzęt z bazy danych, w przeciwnym wypadku zakończyć PU.

PU dodanie_dużego_sprzętu_do_magazynu

OPIS

CEL: dodanie dużego sprzętu do bazy danych

WS(warunki wstępne): Inicjalizacja przez Administratora

WK(warunki końcowe): dodanie nowego sprzętu do listy w bazie danych

PRZEBIEG:

1. Należy wywołać PU sprawdzenie_czy_sprzęt_istnieje, jeśli zwróci prawdę, należy zakończyć PU
2. Należy podać dane sprzętu (model sprzętu, nr seryjny, producenta, adresu mac ethernet, adresu mac wifi)
3. Należy wykonać PU sprawdzenie_czy_adres_mac_jest_unikalny
4. Należy dodać sprzęt o podanych parametrach do listy sprzętu w bazie danych

PU dodanie_nowego_akcesorium_do_magazynu

OPIS

CEL: dodanie akcesorium do bazy danych

WS(warunki wstępne): Inicjalizacja przez Administratora

WK(warunki końcowe): dodanie nowego akcesorium do listy w bazie danych

PRZEBIEG:

1. Należy podać dane sprzętu (model akcesorium, producenta, ilość)
2. Należy dodać sprzęt o podanych parametrach do listy sprzętu w bazie danych

PU usunięcie_użytkownika

OPIS

CEL: usunięcie użytkownika z bazy danych

WS(warunki wstępne): Inicjalizacja przez Administratora

WK(warunki końcowe): Usunięcie użytkownika z listy użytkowników bazy danych

PRZEBIEG:

1. Należy podać identyfikator_wewnętrzny_użytkownika)
2. Należy usunąć użytkownika z bazy danych

PU stworzenie_nazwy_hosta

OPIS

CEL: stworzenie nazwy hosta dla korelacji między użytkownikiem a sprzętem w celu identyfikacji

WS(warunki wstępne): Inicjalizacja przez PU przypisanie_dużego_sprzętu_do_użytkownika

WK(warunki końcowe): wygenerowanie nazwy hosta dla danej korelacji

PRZEBIEG:

1. Należy wywołać PU przypisanie_dużego_sprzętu_do_użytkownika
2. Nazwa zostaje losowo wygenerowana (?) dla danej korelacji

PU sprawdź_czy_użytkownik_istnieje

OPIS

CEL: sprawdzenie czy użytkownik znajduje się w bazie danych

WS(warunki wstępne): Inicjalizacja przez PU przypisanie_dużego_sprzętu_do_użytkownika

lub PU przypisanie_akcesorium_do_użytkownika lub PU

wyświetlenie_sprzętu_przypisanego_do_użytkownika

WK(warunki końcowe): zwrócenie odpowiedzi boolean (true/false) zależnie od wyniku

PRZEBIEG:

1. Wywołanie PU przypisanie_dużego_sprzętu_do_użytkownika lub PU przypisanie_akcesorium_do_użytkownika lub PU wyświetlenie_sprzętu_przypisanego_do_użytkownika
2. Przegląd identyfikatorów wewnętrznych użytkowników w celu znalezienia podanego przez użytkownika
3. Zwrócenie odpowiedzi typu boolean

PU sprawdź_czy_sprzęt_istnieje

OPIS

CEL: sprawdzenie czy sprzęt znajduje się w bazie danych

WS(warunki wstępne): Inicjalizacja przez PU przypisanie_dużego_sprzętu_do_użytkownika lub PU przypisanie_akcesorium_do_użytkownika lub PU

zwrócenie_dużego_sprzętu_do_magazynu lub PU usunięcie_dużego_sprzętu_z_magazynu

WK(warunki końcowe): zwrócenie odpowiedzi boolean (true/false) zależnie od wyniku

PRZEBIEG:

1. Wywołanie PU przypisanie_dużego_sprzętu_do_użytkownika lub PU przypisanie_akcesorium_do_użytkownika lub PU zwrócenie_dużego_sprzętu_do_magazynu lub PU usunięcie_dużego_sprzętu_z_magazynu
2. Przegląd identyfikatorów wewnętrznych dużego sprzętu w celu znalezienia podanego przez użytkownika
3. Zwrócenie odpowiedzi typu boolean

PU sprawdzenie_czy_adres_mac_jest_unikalny

OPIS

CEL: Sprawdzenie czy dany adres MAC już istnieje w bazie danych

WS(warunki wstępne): Inicjalizacja przez PU dodanie_dużego_sprzętu_do_magazynu

WK(warunki końcowe): zwrócenie odpowiedzi boolean (true/false) zależnie od wyniku

PRZEBIEG:

1. Wywołanie PU dodanie_dużego_sprzętu_do_magazynu
2. Przegląd adresów MAC w celu znalezieniu podanego przez użytkownika
3. Zwrócenie odpowiedzi typu boolean

2.3 Wymagania niefunkcjonalne

Pracownik:

- Pracownik nie może nadać sobie samemu numeru identyfikacyjnego
- Pracownik nie może edytować swoich danych w bazie
- Pracownik nie może dodawać oraz usuwać sprzętu
- Pracownik nie może edytować historii operacji

Administrator:

- Administrator nie może nadać dwóch takich samych identyfikatorów sprzętu oraz użytkowników

System:

- Dla każdego sprzętu, który posiada adres MAC, po jego przypisaniu tworzymy nazwę hosta
- Jedna jednostka sprzętu może być przypisana tylko do jednego użytkownika
- Nazwa hosta musi być unikalna
- Adres MAC sprzętu musi być unikalny
- Każdy użytkownik musi mieć indywidualny identyfikator
- Każdy duży sprzęt musi mieć indywidualny identyfikator
- Baza danych jest stworzona w języku SQL
- Baza danych jest stworzona w narzędziu MySQL Workbench
- Aplikacja jest napisana w języku Python
- Aplikacja jest napisana w narzędziu PyCharm
- Do modelowania bazy danych został użyty program MySQL Workbench

2.3.1 Wykorzystywane technologie i narzędzia

Baza danych będzie obsługiwana za pośrednictwem bazy danych zbudowanej w programie MySQL Workbench oraz systemu zarządzania np. MySQL Workbench uruchomionego lokalnie na komputerze. Interfejs został zrealizowany w postaci aplikacji obiektowej w języku Python uruchamianej lokalnie na komputerze napisanej w środowisku PyCharm. Do specyfikacji funkcji systemu wykorzystany został zunifikowany język modelowania UML w notacji Martina/IE (notacja kurzej stopki).

2.3.2 Wymagania dotyczące bezpieczeństwa systemu

System posiada zabezpieczenia w postaci logowania do aplikacji. Zależnie od rodzaju konta dostępne są inne funkcjonalności. Administrator posiada te same funkcjonalności co Pracownik oraz ma dodatkowo rozszerzone funkcjonalności o funkcje wymagane do obsługi i administracji bazy danych w celu utrzymania jej zaktualizowanej.

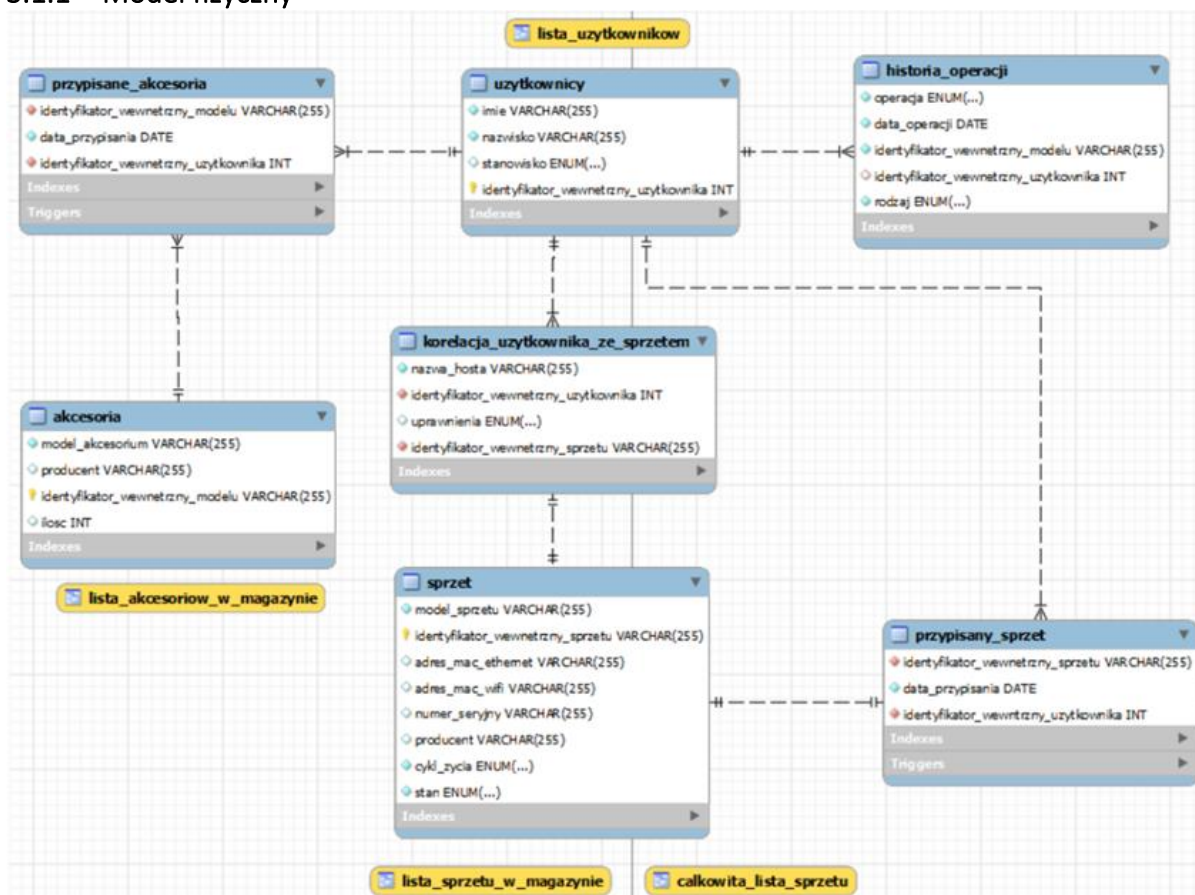
2.4 Przyjęte założenia projektowe

Projekt składa się z bazy danych napisanej w języku SQL połączonej z aplikacją napisaną w języku Python. Baza danych składa się z siedmiu tabel z czego każda z nich posiada przynajmniej trzy encje. Każda z encji posiada indywidualne ograniczenia w celu zachowania spójności danych w bazie. Każda z tabel jest połączona z przynajmniej jedną inną tabelą relacją umożliwiającą odpowiednie działanie projektu. Kolejnym istotnym elementem w projekcie jest aplikacja umożliwiająca logowanie do bazy danych jako Pracownik lub Administrator. Każda z ról posiada określone możliwości interakcji z bazą danych. Możliwości każdej z nich zostały opisane w podpunkcie 2.2.2 oraz ukazane na diagramie w podpunkcie 2.2.1.

3 Projekt systemu

3.1 Projekt bazy danych

3.1.1 Model fizyczny



Rysunek 3 Model fizyczny stworzonej bazy danych

3.1.2 Ograniczenia integralności danych

Tabela 1 Tabela przedstawiająca encje występujące w bazie danych wraz z opisem

Nazwa encji	Opis encji
uzytkownicy	Encja przechowująca dane identyfikujące użytkowników (Pracowników) oraz podstawowe dane na ich temat
sprzet	Encja przechowująca dane identyfikujące sprzęt znajdujący się w magazynie oraz podstawowe dane na jego temat
akcesoria	Encja przechowująca dane identyfikujące akcesoria znajdujące się w magazynie oraz podstawowe dane na ich temat
przypisany_sprzet	Encja przechowująca dane na temat sprzętu przypisanego do każdego użytkownika
przypisane_akcesoria	Encja przechowująca dane na temat akcesoriów przypisanych do każdego użytkownika
historia_operacji	Encja przechowująca dane na temat operacji przeprowadzanych przez użytkownika (historia wypożyczanych sprzętów/akcesoriów z magazynu)
korelacja_uzytkownika_ze_sprzetem	Encja przechowująca dane na temat sprzętu przypisanego do każdego użytkownika, uprawnień oraz ważnych informacji wymaganych do użytku sprzętu

Tabela 2 Tabela przedstawiająca atrybuty w encji uzytkownicy wraz z opisem

Atrybut	Opis atrybutu	Typ oraz ograniczenia atrybutu
imie	Atrybut będący ciągiem wynoszącym do 255 znaków przechowujących imię pracownika	VARCHAR, not null
nazwisko	Atrybut będący ciągiem wynoszącym do 255 znaków przechowujących imię pracownika	VARCHAR, not null
stanowisko	Atrybut będący ciągiem znaków zawierającym jedno z dostępnych stanowisk	ENUM, not null
identyfikador_wewnetrzny_uzytkownika	Klucz główny będący numerem ID pracownika zatrudnionego w firmie typu całkowitoliczbowego, generowany automatycznie za pomocą sekwencji, jest unikalny oraz nie może być pusty	Klucz główny, INTEGER, not null, unique

Tabela 3 Tabela przedstawiająca atrybuty w encji sprzet wraz z opisem

Atrybut	Opis atrybutu	Typ oraz ograniczenia atrybutu
model_sprzetu	Atrybut będący ciągiem wynoszącym do 255 znaków przechowujących nazwę modelu sprzętu	VARCHAR, not null
identyfikator_wewnetrzny_sprzetu	Klucz główny będący numerem ID sprzętu w firmie typu całkowitoliczbowego, generowany automatycznie za pomocą sekwencji, jest unikalny oraz nie może być pusty	Klucz główny, INTEGER, not null, unique
adres_mac_ethernet	Atrybut będący ciągiem wynoszącym do 255 znaków przechowujących adres mac ethernet przypisany do sprzętu	VARCHAR, not null
adres_mac_wifi	Atrybut będący ciągiem wynoszącym do 255 znaków przechowujących adres mac wifi przypisany do sprzętu	VARCHAR, not null
numer_seryjny	Atrybut będący ciągiem wynoszącym do 255 znaków przechowujących numer seryjny sprzętu	VARCHAR, not null
producent	Atrybut będący ciągiem wynoszącym do 255 znaków przechowujących nazwę producenta sprzętu	VARCHAR, not null
cykl_zycia	Atrybut będący ciągiem znaków zawierającym cykl życia sprzętu	ENUM, not null
stan	Atrybut będący ciągiem znaków zawierającym jeden z dostępnych możliwych stanów sprzętu	ENUM, not null

Tabela 4 Tabela przedstawiająca atrybuty w encji akcesoria wraz z opisem

Atrybut	Opis atrybutu	Typ oraz ograniczenia atrybutu
model_akcesorium	Atrybut będący ciągiem wynoszącym do 255 znaków przechowujących nazwę modelu sprzętu	VARCHAR, not null
producent	Atrybut będący ciągiem wynoszącym do 255 znaków przechowujących nazwę producenta sprzętu	VARCHAR, not null
ilosc	Atrybut będący typem całkowitoliczbowym odpowiadającym ilości sprzętu danego rodzaju w firmie	INTEGER
identyfikator_wewnetrzny_modelu	Klucz główny będący numerem ID akcesorium w firmie typu całkowitoliczbowego, generowany automatycznie za pomocą sekwencji, jest unikalny oraz nie może być pusty	Klucz główny, INTEGER, not null, unique

Tabela 5 Tabela przedstawiająca atrybuty w encji historia_operacji wraz z opisem

Atrybut	Opis atrybutu	Typ oraz ograniczenia atrybutu
operacja	Atrybut będący ciągiem wynoszącym do 255 znaków przechowujących nazwę operacji	ENUM, not null
data_operacji	Atrybut przechowujący datę operacji jakiej dokonano na sprzęcie	DATE, not null
identyfikator_wewnetrzny_modelu	Atrybut przechowujący ID akcesorium w firmie typu całkowitoliczbowego, generowany automatycznie za pomocą sekwencji, jest unikalny oraz nie może być pusty	INTEGER, not null
identyfikator_wewnetrzny_uzytkownika	Atrybut przechowujący ID pracownika zatrudnionego w firmie typu całkowitoliczbowego, generowany automatycznie za pomocą sekwencji, jest unikalny oraz nie może być pusty	INTEGER, not null, unique
rodzaj	Atrybut przechowujący informację na temat rodzaju wykonanej operacji	ENUM, not null

Tabela 6 Tabela przedstawiająca atrybuty w encji przypisane_akcesoria wraz z opisem

Atrybut	Opis atrybutu	Typ oraz ograniczenia atrybutu
identyfikator_wewnetrzny_modelu	Klucz obcy będący numerem ID akcesorium w firmie typu całkowitoliczbowego, generowany automatycznie za pomocą sekwencji, jest unikalny oraz nie może być pusty	Klucz obcy, INTEGER, not null, unique
data_przypisania	Atrybut przechowujący datę przypisania akcesorium do użytkownika	VARCHAR, not null
identyfikator_wewnetrzny_uzytkownika	Atrybut przechowujący ID pracownika zatrudnionego w firmie typu całkowitoliczbowego, generowany automatycznie za pomocą sekwencji, jest unikalny oraz nie może być pusty	Klucz obcy, INTEGER, not null, unique

Tabela 7 Tabela przedstawiająca atrybuty w encji przypisany_sprzet wraz z opisem

Atrybut	Opis atrybutu	Typ oraz ograniczenia atrybutu
identyfikator_wewnetrzny_sprzetu	Klucz obcy będący numerem ID sprzętu w firmie typu całkowitoliczbowego, generowany automatycznie za pomocą sekwencji, jest unikalny oraz nie może być pusty	Klucz obcy, INTEGER, not null, unique
data_przypisania	Atrybut przechowujący datę przypisania akcesorium do użytkownika	VARCHAR, not null
identyfikator_wewnetrzny_uzytkownika	Atrybut przechowujący ID pracownika zatrudnionego w firmie typu całkowitoliczbowego, generowany automatycznie za pomocą sekwencji, jest unikalny oraz nie może być pusty	Klucz obcy, INTEGER, not null, unique

Tabela 8 Tabela przedstawiająca atrybuty w encji korelacja_uzytkownika_ze_sprzetem wraz z opisem

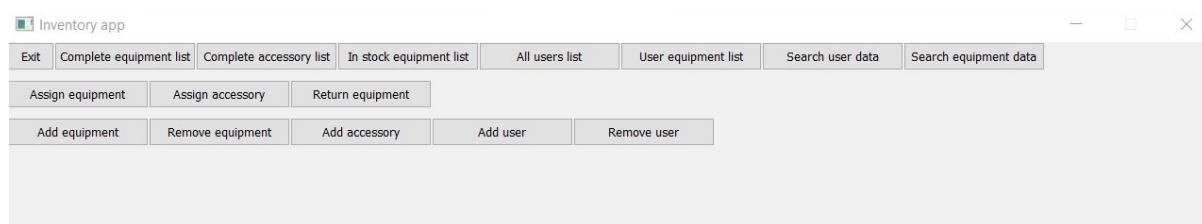
Atrybut	Opis atrybutu	Typ oraz ograniczenia atrybutu
identyfikator_wewnetrzny_sprzetu	Klucz obcy będący numerem ID sprzętu w firmie typu całkowitoliczbowego, generowany automatycznie za pomocą sekwencji, jest unikalny oraz nie może być pusty	Klucz obcy, INTEGER, not null, unique
nazwa_hosta	Atrybut przechowujący nazwę hosta przypisaną do danej korelacji	VARCHAR, not null
identyfikator_wewnetrzny_uzytkownika	Atrybut przechowujący ID pracownika zatrudnionego w firmie typu całkowitoliczbowego, generowany automatycznie za pomocą sekwencji, jest unikalny oraz nie może być pusty	Klucz obcy, INTEGER, not null, unique
uprawnienia	Atrybut przechowujący informację na temat rodzaju uprawnień przypisanych użytkownikowi	ENUM, not null

3.2 Projekt aplikacji użytkownika

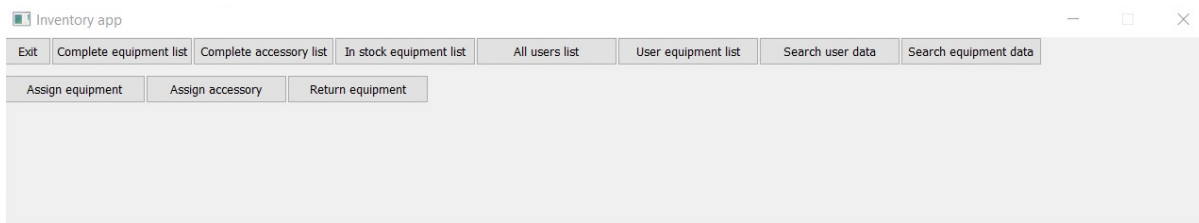
3.2.1 Architektura aplikacji oraz metoda podłączenia do bazy danych.

Baza danych została napisana w aplikacji MySQL oraz funkcjonuje w formie hosta lokalnego. Do jej stworzenia użyto aplikacja MySQL Workbench. Interfejs wykonany został w IDE PyCharm przy użyciu języka Python. Połączenie ze sobą bazy danych oraz aplikacji następuje przy pomocy modułu MySQL Conector (w projekcie jest on używany przez klasę **my_database**). W celu uzyskania dostępu do bazy poprzez aplikację należy wywołać funkcję **connect to database**, która jest wywoływana automatycznie przy włączeniu aplikacji oraz podaniu odpowiednich danych logowania (w tym przypadku jako administrator lub pracownik). Jeśli dane zostaną podane niepoprawne, pojawi się komunikat informujący o błędzie. Odłączenie od bazy danych następuje za pomocą wywołania funkcji **disconnect** oraz dzieje się to automatycznie przy wyjściu z aplikacji. Wszystkie operacje w menu napisanego przez nas programu wywoływane są za pomocą funkcji **perform command**, która wykonuje konkretną, przypisaną do przycisku komendę w bazie danych w języku SQL. **Perform command** następuje po wybraniu odpowiedniego przycisku w menu oraz wprowadzeniu wymaganych danych. W wyniku wykonania komendy następuje ewentualne zwrócenie do aplikacji tych samych danych, które zwraca baza. W wykonanym projekcie występują dwa rodzaje okien informacyjnych: **error box** oraz **info box**, które pojawiają się np. po niepoprawnym wprowadzeniu danych logowania.

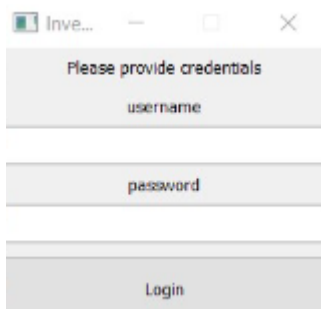
3.2.2 Interfejs graficzny i struktura menu



Rysunek 4 Widok głównego Menu aplikacji po zalogowaniu jako administrator



Rysunek 5 Widok głównego Menu aplikacji po zalogowaniu jako użytkownik



Rysunek 6 Ekran logowania do aplikacji obsługującej bazę danych

4 Implementacja systemu

4.1 Realizacja bazy danych

4.1.1 Tworzenie tabel i definiowanie ograniczeń

```

1 CREATE TABLE `przypisany_sprzet` (
2   `identyfikator_wewnetrzny_sprzetu` varchar(255) NOT NULL,
3   `data_przypisania` date NOT NULL,
4   `identyfikator_wewnetrzny_uzytkownika` int NOT NULL,
5   UNIQUE KEY `identyfikator_wewnetrzny_sprzetu_UNIQUE` (`identyfikator_wewnetrzny_sprzetu`),
6   KEY `identyfikator_wewnetrzny_uzytkownika_idx` (`identyfikator_wewnetrzny_uzytkownika`),
7   CONSTRAINT `przypisany_identyfikator_wewnetrzny_sprzetu` FOREIGN KEY (`identyfikator_wewnetrzny_sprzetu`) REFERENCES `sprzet`
   (`identyfikator_wewnetrzny_sprzetu`),
8   CONSTRAINT `przypisany_identyfikator_wewnetrzny_uzytkownika` FOREIGN KEY (`identyfikator_wewnetrzny_uzytkownika`) REFERENCES `uzytkownicy`
   (`identyfikator_wewnetrzny_uzytkownika`)
9 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;

```

Rysunek 7 Implementacja kodu służącego do stworzenia Tabeli przypisany_sprzet wraz z ograniczeniami

Każda z tabel została stworzona w analogiczny sposób z podmianą konkretnych nazw kolumn oraz kryteriów, które muszą spełnić wprowadzane dane.

4.1.2 Implementacja mechanizmów przetwarzania danych

```
1 CREATE DEFINER='root'@'localhost' TRIGGER `przypisany_sprzet_BEFORE_INSERT` BEFORE INSERT ON `przypisany_sprzet` FOR EACH ROW BEGIN
2   IF (SELECT COUNT(sprzet.identyfikator_wewnetrzny_sprzetu) FROM sprzet WHERE sprzet.identyfikator_wewnetrzny_sprzetu =
new.identyfikator_wewnetrzny_sprzetu) != 1 THEN
3     SIGNAL SQLSTATE '45000'
4     SET MESSAGE_TEXT = 'identyfikator_wewnetrzny_sprzetu does not exist';
5   END IF;
6
7   IF (SELECT COUNT(uzytkownicy.identyfikator_wewnetrzny_uzytkownika) FROM uzytkownicy WHERE
uzytkownicy.identyfikator_wewnetrzny_uzytkownika = new.identyfikator_wewnetrzny_uzytkownika) != 1 THEN
8     SIGNAL SQLSTATE '45000'
9     SET MESSAGE_TEXT = 'identyfikator_wewnetrzny_uzytkownika does not exist';
10  END IF;
11 END
```

Rysunek 8 Implementacja kodu Triggera obsługujące próbę przypisania nieistniejącego sprzętu lub przypisania do nieistniejącego użytkownika

Każdy z triggerów został stworzony analogicznie z odpowiednim dostosowaniem do pełnionych przez niego funkcji.

4.1.3 Implementacja uprawnień i innych zabezpieczeń

4.2 Realizacja elementów aplikacji

4.2.1 Obsługa menu

```
1 import re
2 from PyQt5.QtCore import Qt
3 from PyQt5.QtWidgets import QPushButton, QLabel, QTableWidgetItem, QDialog, QLineEdit
4 from PyQt5.QtWidgets import *
5 from datetime import date
6
7 from my_database import my_database
8
9
10 class user_interface(QMainWindow):
11     def __init__(self):
12         super().__init__()
13
14         self.database = my_database()
15         self.connection = None
16
17         self.setWindowTitle("Inventory app")
18
19         # Obiekty okna logowania
20         self.label_login = QLabel("Please provide credentials", self)
21         self.label_user_name = QLabel("username", self)
22         self.label_password = QLabel("password", self)
23         self.text_box_user = QLineEdit(self)
24         self.text_box_password = QLineEdit(self)
25         self.text_box_password.setEchoMode(QLineEdit.Password)
26         self.button_login = QPushButton("Login", self)
27         self.hide_login_menu()
```

Rysunek 9 Implementacja kodu obsługującego przypisanie funkcji do wybranych przykładowych przycisków oraz widocznych w menu funkcji

```

96     def hide_login_menu(self):
97         self.label_login.hide()
98         self.label_user_name.hide()
99         self.label_password.hide()
100        self.text_box_user.hide()
101        self.text_box_password.hide()
102        self.button_login.hide()
103
104    def create_connection(self):
105        user_name = self.text_box_user.displayText()
106        self.text_box_password.setEchoMode(QLineEdit.Normal)
107        password = self.text_box_password.displayText()
108
109        self.connection = self.database.connect_to_database(user_name, password)
110
111        try:
112            if self.connection.is_connected():
113                self.hide_login_menu()
114                self.display_main_menu()
115        except:
116            return

```

Rysunek 10 Implementacja kodu odpowiedzialnego za działanie logowania do aplikacji

Do implementacji interfejsu został użyty moduł PyQt5. Przy jego użyciu zdefiniowane zostały pola tekstowe (takie jak np. pole tekstowe służące do wprowadzenia danych do logowania) przyciski widoczne w menu aplikacji w pkt. 3.2.2 oraz tabele wyświetlające wyniki operacji wykonywanych na bazie danych. Każdy z obiektów interfejsu wyświetla się oraz zostaje ukryty poprzez wywołanie odpowiednich funkcji.

4.2.2 Walidacja i filtracja

Napisana przez nas aplikacja weryfikuje poprawność wprowadzanych danych tylko na poziomie bazy danych za pomocą triggerów, takich jak przykładowy opisany w pkt. 4.1.2. Naszym celem było pozostawienie jak największej ilości możliwości modyfikacji wprowadzanych danych zależnie od wymagań firmy używającej naszej aplikacji.

4.2.3 Implementacja interfejsu dostępu do bazy danych

```
6 class my_database:
7
8     def connect_to_database(self, user_name, password):
9         connection = []
10        try:
11            connection = mysql.connector.connect(host='localhost', database='projektodb', user=user_name, password=password)
12
13            if connection.is_connected():
14                cursor = connection.cursor()
15                cursor.execute("select database();")
16                record = cursor.fetchone()
17                record_message = str(record)
18                self.display_info_box("You're connected to database: " + record_message)
19
20        except Error as e:
21            error_message = str(e)
22            self.display_error_box("Error occurred while connecting to the database: " + error_message)
23
24        return connection
25
26    def disconnect_from_database(self, connection):
27        if connection.is_connected():
28            connection.cursor().close()
29            connection.close()
30            self.display_info_box("MySQL connection was closed")
31        else:
32            self.display_info_box("MySQL connection was not closed, because it was not created.")
33
34    def perform_command(self, connection, command):
35        try:
36            if connection.is_connected():
37                cursor = connection.cursor()
38                cursor.execute(command)
39                result = cursor.fetchall()
40                return result
41
42        except Error as e:
43            error_message = str(e)
44            self.display_error_box("Database returned an error while performing the action: " + error_message)
```

Rysunek 11 Implementacja kodu odpowiedzialnego za wytworzenie połączenia między bazą danych a aplikacją

Opis Rysunku 11 został zamieszczony w podpunkcie 3.2.1.

4.2.4 Implementacja wybranych funkcjonalności systemu

```
278 def display_complete_accessory_list(self):
279     self.table_result.show()
280     self.hide_search_user_tables()
281     result = self.database.perform_command(self.connection, "select * from lista_akcesoriow_w_magazynie")
282     if not result:
283         self.table_result.clear()
284         self.table_result.setRowCount(0)
285         self.table_result.setColumnCount(0)
286     else:
287         self.table_result.setColumnCount(4)
288         self.table_result.setHorizontalHeaderLabels(
289             ["model_akcesorium", "producent", "identyfikator_wewnetrzny_modelu", "ilosc"])
290     row_count = 0
291     for model_akcesorium, producent, identyfikator_wewnetrzny_modelu, ilosc in result:
292         self.table_result.setRowCount(row_count + 1)
293         self.table_result.setItem(row_count, 0, QTableWidgetItem(model_akcesorium))
294         self.table_result.setItem(row_count, 1, QTableWidgetItem(producent))
295         self.table_result.setItem(row_count, 2, QTableWidgetItem(identyfikator_wewnetrzny_modelu))
296         self.table_result.setItem(row_count, 3, QTableWidgetItem(str(ilosc)))
297         row_count = row_count + 1
```

Rysunek 12 Implementacja kodu obsługującego funkcję wyświetlania listy akcesoriów

```

536     def assign_equipment(self):
537         response_input_dialog_user_id = QInputDialog.getText(self, 'Input Dialog', 'Enter user id:')
538         if response_input_dialog_user_id[1]:
539             response_input_dialog_equipment_id = QInputDialog.getText(self, 'Input Dialog', 'Enter equipment id:')
540             if response_input_dialog_equipment_id[1]:
541                 self.database.perform_command(self.connection, "insert into przypisany_sprzet (identyfikator_wewnetrzny_sprzetu,
542                 self.connection.commit()

```

Rysunek 13 Implementacja kodu obsługującego przypisanie sprzętu do użytkownika

Funkcja widoczna na Rysunku 12 wywoływana jest przez przycisk interfejsu oraz służy do wyświetlenia w tabeli pełnej listy sprzętu. W pierwszej kolejności jest wywoływana odpowiednia komenda w bazie danych w języku SQL. W przypadku, gdy komenda nie zwróci żadnej wartości, tabela jest czyszczona, aby pozostała pusta, w przeciwnym wypadku zawartość tabeli jest wyświetlana przy pomocy pętli for.

Funkcja widoczna na Rysunku 13 wywoływana jest przez przycisk interfejsu oraz służy do przypisania sprzętu do konkretnego pracownika. W pierwszej kolejności wyświetla się okno dialogowe, które wymaga podania **ID użytkownika**. Następnie pojawia się kolejne okno dialogowe wymagające wprowadzenia **ID sprzętu**. Po kliknięciu przycisku OK zostaje wykonana operacja przypisania, w wypadku gdy zostanie naciśnięty przycisk **Cancel**, operacja nie zostanie wykonana. W przypadku podania nieprawidłowych danych, aplikacja wyświetli puste tabele

4.2.5 Implementacja mechanizmów bezpieczeństwa

```

104     def create_connection(self):
105         user_name = self.text_box_user.displayText()
106         self.text_box_password.setEchoMode(QLineEdit.Normal)
107         password = self.text_box_password.displayText()
108
109         self.connection = self.database.connect_to_database(user_name, password)
110
111         try:
112             if self.connection.is_connected():
113                 self.hide_login_menu()
114                 self.display_main_menu()
115         except:
116             return

```

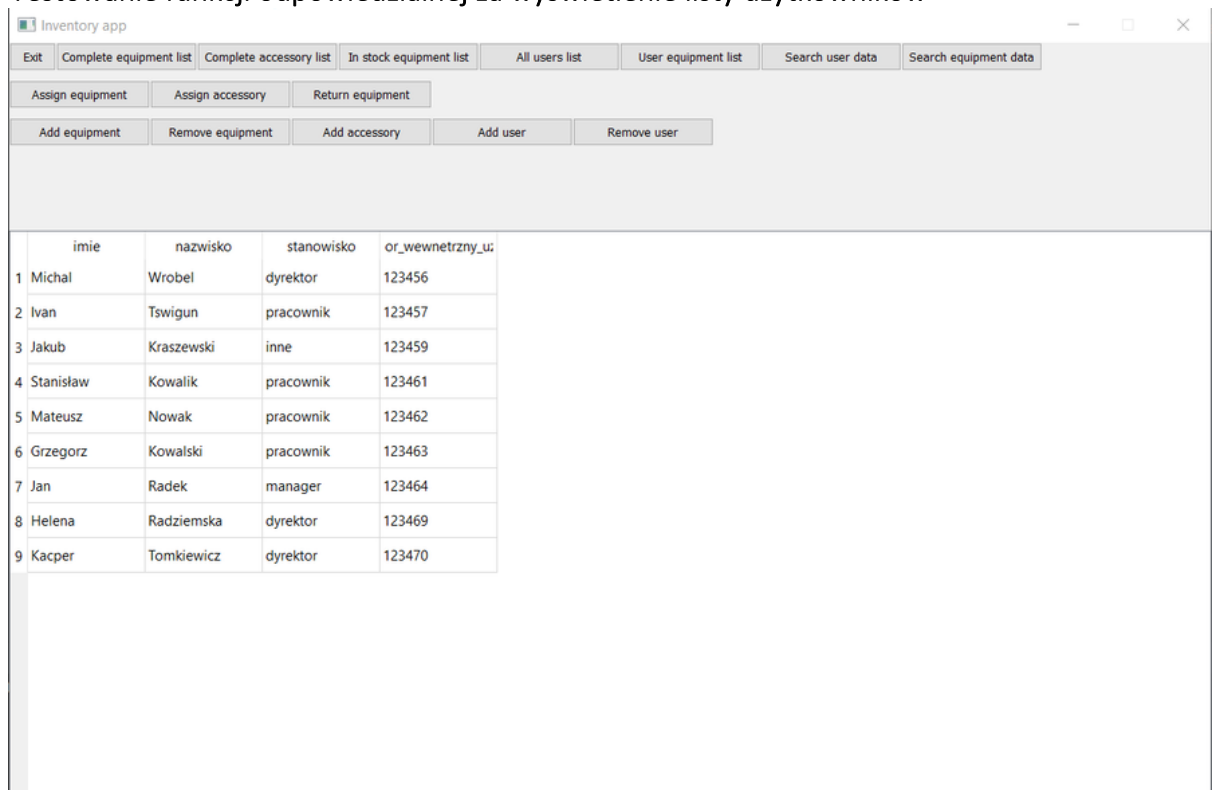
Rysunek 14 Implementacja kodu odpowiedzialnego za sprawdzenie danych logowania

Funkcja **create_connection** pobiera login oraz hasło z pól tekstowych oraz następnie przekazuje je do funkcji **connect_to_database** klasy **my_database**. Jeżeli wprowadzone dane są zgodne z przechowywanymi w kodzie aplikacji, pole logowania zniknie wprowadzając użytkownika w odpowiedni widok, zależnie od uprawnień konta. Każdy z widoków został przedstawiony w pkt. 3.2.2.

5 Testowanie systemu

5.1 Testowanie opracowanych funkcji systemu

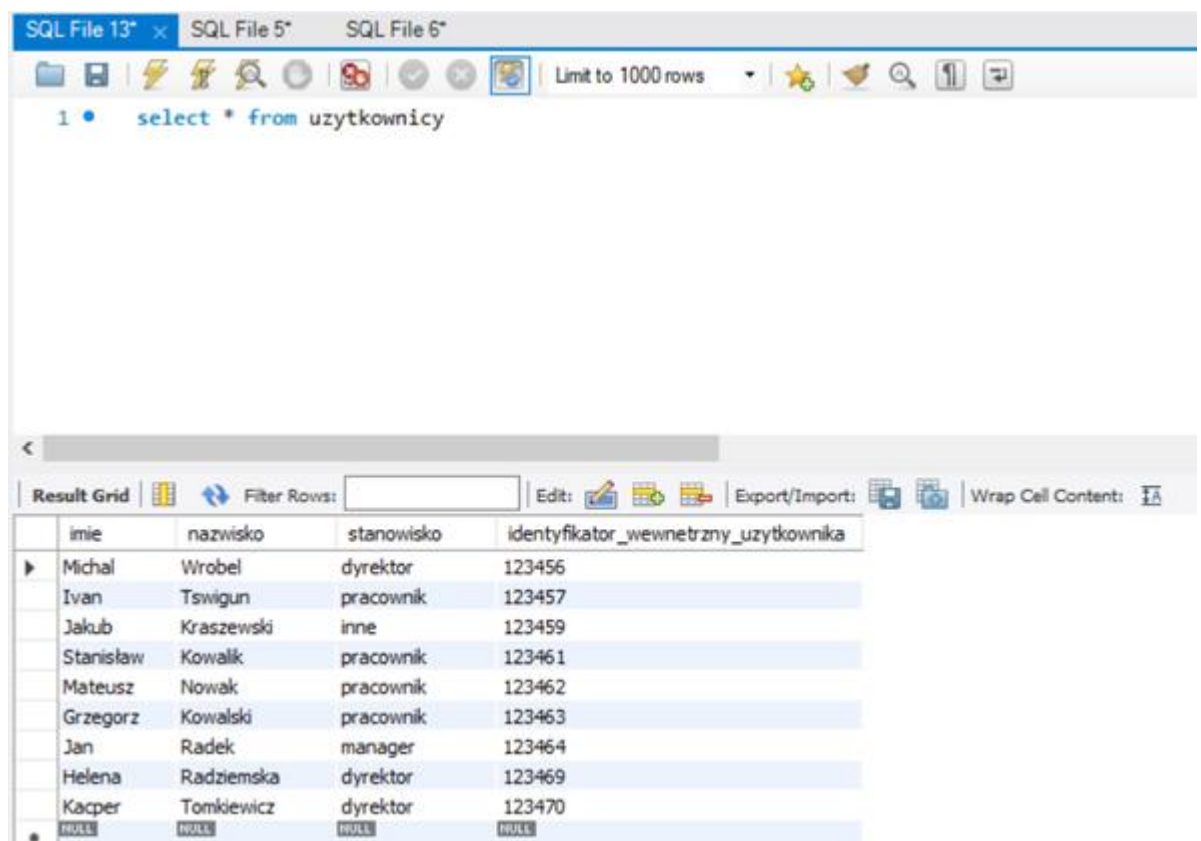
5.1.1 Testowanie funkcji odpowiedzialnej za wyświetlenie listy użytkowników



The screenshot shows a web application titled 'Inventory app'. It has a top navigation bar with buttons: 'Exit', 'Complete equipment list', 'Complete accessory list', 'In stock equipment list', 'All users list', 'User equipment list', 'Search user data', and 'Search equipment data'. Below this is a secondary bar with 'Assign equipment', 'Assign accessory', and 'Return equipment'. A third bar contains 'Add equipment', 'Remove equipment', 'Add accessory', 'Add user', and 'Remove user'. The main content area displays a table with 9 rows of user data.

	imie	nazwisko	stanowisko	or_wewnetrny_u;
1	Michał	Wrobel	dyrektor	123456
2	Ivan	Tswigun	pracownik	123457
3	Jakub	Kraszewski	inne	123459
4	Stanisław	Kowalik	pracownik	123461
5	Mateusz	Nowak	pracownik	123462
6	Grzegorz	Kowalski	pracownik	123463
7	Jan	Radek	manager	123464
8	Helena	Radziemska	dyrektor	123469
9	Kacper	Tomkiewicz	dyrektor	123470

Rysunek 15 Lista użytkowników widoczna w aplikacji



The screenshot shows a SQL client interface with three tabs: 'SQL File 13*', 'SQL File 5*', and 'SQL File 6*'. The active tab shows a query: `1 • select * from uzytkownicy`. Below the query editor is a 'Result Grid' showing the same 9 rows of user data as in the application screenshot. The columns are: 'imie', 'nazwisko', 'stanowisko', and 'identyfikator_wewnetrny_uzytkownika'.

	imie	nazwisko	stanowisko	identyfikator_wewnetrny_uzytkownika
▶	Michał	Wrobel	dyrektor	123456
	Ivan	Tswigun	pracownik	123457
	Jakub	Kraszewski	inne	123459
	Stanisław	Kowalik	pracownik	123461
	Mateusz	Nowak	pracownik	123462
	Grzegorz	Kowalski	pracownik	123463
	Jan	Radek	manager	123464
	Helena	Radziemska	dyrektor	123469
	Kacper	Tomkiewicz	dyrektor	123470
•	NULL	NULL	NULL	NULL

Rysunek 16 Lista użytkowników widoczna w bazie danych

Funkcja jest aktywowana za pomocą przycisku **All users list**. Po kliknięciu zostaje wywołana odpowiednia komenda w języku SQL w bazie danych, co powoduje wyświetlenie tabeli zawierającej dane użytkowników takie jak widoczne na Rysunku 15. Następnie dane są przekazywane do aplikacji, która wyświetla taki sam widok jak baza danych. Rysunek 16 przedstawia dane wyświetlane przez bazę danych po wprowadzeniu komendy odpowiedzialnej za wyświetlenie użytkowników w celu porównania jej zawartości z widoczną w aplikacji na Rysunku 15.

5.1.2 Test funkcji odpowiedzialnej za przypisanie sprzętu do użytkownika

2	Laptop	7800002	00:1B:45:11:3A:B7	10:1B:44:12:3A:B7	D232DQWD3DD	Asus	nowy	w_magazynie
---	--------	---------	-------------------	-------------------	-------------	------	------	-------------

Rysunek 17 Widok statusu sprzętu w aplikacji przed wykonaniem operacji

model_sprzetu	identyfikator_wewnetrzny_sprzetu	adres_mac_ethernet	adres_mac_wifi	numer_seryjny	producent
Laptop	7800001	00:1B:44:11:3A:D7	08:1B:44:12:3A:B7	QWE31324DE32	Asus
Laptop	7800002	00:1B:45:11:3A:B7	10:1B:44:12:3A:B7	D232DQWD3DD	Asus
Laptop	7800003	00:1B:44:11:2A:B7	30:1B:44:12:3A:B7	23D23D423DQD	Asus
Laptop	7800004	00:3B:44:11:3A:B7	40:1B:44:12:3A:B7	FD43FWERF355	Asus
Laptop	7800005	00:1B:48:11:3A:B7	05:1B:44:12:3A:B7	34F43FDSF5F5	Asus
Laptop	7800006	00:1B:44:11:3A:B7	06:1B:44:12:3A:B7	G54G6567H7556	Asus
Laptop	7800008	00:1B:34:11:3A:B7	22:1B:44:12:3A:B7	BRT4B46654654	Asus
Laptop	7800009	00:1B:44:11:DA:B7	11:1B:44:12:3A:B7	QWE313245E32	Asus
Laptop	7800010	10:1B:44:11:3A:B7	13:1B:44:12:3A:B7	QWE61324DE32	Asus

Rysunek 18 Lista sprzętu w firmie widoczna w bazie danych wraz ze statusem przed wykonaniem operacji

model_sprzetu	ator_wewnetrzny	dres_mac_ethernet	adres_mac_wifi	numer_seryjny	producent	cykl_zycia	stan
1 Laptop	7800001	00:1B:44:11:3A:D7	08:1B:44:12:3A:B7	QWE31324DE32	Asus	uzywany	przypisany
2 Laptop	7800002	00:1B:45:11:3A:B7	10:1B:44:12:3A:B7	D232DQWD3DD	Asus	nowy	w_magazynie
3 Laptop	7800003	00:1B:44:11:2A:B7	30:1B:44:12:3A:B7			uzywany	przypisany
4 Laptop	7800004	00:3B:44:11:3A:B7	40:1B:44:12:3A:B7			uzywany	przypisany
5 Laptop	7800005	00:1B:48:11:3A:B7	05:1B:44:12:3A:B7			nowy	w_magazynie
6 Laptop	7800006	00:1B:44:11:3A:B7	06:1B:44:12:3A:B7			uzywany	w_magazynie
7 Laptop	7800008	00:1B:34:11:3A:B7	22:1B:44:12:3A:B7	BRT4B46654654	Asus	nowy	w_magazynie
8 Laptop	7800009	00:1B:44:11:DA:B7	11:1B:44:12:3A:B7	QWE313245E32	Asus	nowy	w_magazynie
9 Laptop	7800010	10:1B:44:11:3A:B7	13:1B:44:12:3A:B7	QWE61324DE32	Asus	nowy	w_magazynie
10 Laptop	7800011	02:1B:44:11:3A:B7	14:1B:44:12:3A:B7	QWE31334DE32	Asus	nowy	w_magazynie
11 Macbook	7800013	02:01:44:11:3A:B7	02:02:44:11:3A:B7	2D3D232DAWFF	Apple	uzywany	w_magazynie
12 Macbook	7800014	02:03:44:11:3A:B7	02:04:44:11:3A:B7	2DQWDWDSFGD	Apple	uzywany	przypisany
13 Macbook	7800015	02:05:44:11:3A:B7	02:06:44:11:3A:B7	453454FRFRF44	Apple	nowy	w_magazynie
14 Macbook	7800016	02:07:44:11:3A:B7	02:08:44:11:3A:B7	SDF3F4344FFFF	Apple	uzywany	w_magazynie
15 Macbook	7800017	02:09:44:11:3A:B7	02:10:44:11:3A:B7	3FRWFR34RF4F	Apple	nowy	w_magazynie

Rysunek 19 Widok wprowadzanych danych (identyfikator_wewnetrzny_uzytkownika) do wykonania operacji

Inventory app

Exit Complete equipment list Complete accessory list In stock equipment list All users list User equipment list Search user data Search equipment data

Assign equipment Assign accessory Return equipment

Add equipment Remove equipment Add accessory Add user Remove user

	model_sprzetu	ator_wewnetrzn	dres_mac_etherne	adres_mac_wifi	numer_seryjny	producent	cykl_zycia	stan
1	Laptop	7800001	00:1B:44:11:3A:D7	08:1B:44:12:3A:B7	QWE31324DE32	Asus	uzywany	przypisany
2	Laptop	7800002	00:1B:45:11:3A:B7	10:1B:44:12:3A:B7	D232DQWD3DD	Asus	nowy	w_magazynie
3	Laptop	7800003	00:1B:44:11:2A:B7	30:1B:44:12:3A:B7			uzywany	przypisany
4	Laptop	7800004	00:3B:44:11:3A:B7	40:1B:44:12:3A:B7			uzywany	przypisany
5	Laptop	7800005	00:1B:48:11:3A:B7	05:1B:44:12:3A:B7			nowy	w_magazynie
6	Laptop	7800006	00:1B:44:11:3A:B7	06:1B:44:12:3A:B7			uzywany	w_magazynie
7	Laptop	7800008	00:1B:34:11:3A:B7	22:1B:44:12:3A:B7	BRT4B46654654	Asus	nowy	w_magazynie
8	Laptop	7800009	00:1B:44:11:DA:B7	11:1B:44:12:3A:B7	QWE313245E32	Asus	nowy	w_magazynie
9	Laptop	7800010	10:1B:44:11:3A:B7	13:1B:44:12:3A:B7	QWE61324DE32	Asus	nowy	w_magazynie
10	Laptop	7800011	02:1B:44:11:3A:B7	14:1B:44:12:3A:B7	QWE31334DE32	Asus	nowy	w_magazynie
11	Macbook	7800013	02:01:44:11:3A:B7	02:02:44:11:3A:B7	2D3D232DAWFF	Apple	uzywany	w_magazynie
12	Macbook	7800014	02:03:44:11:3A:B7	02:04:44:11:3A:B7	2DQWDWDSFGD	Apple	uzywany	przypisany
13	Macbook	7800015	02:05:44:11:3A:B7	02:06:44:11:3A:B7	453454FRFRF44	Apple	nowy	w_magazynie
14	Macbook	7800016	02:07:44:11:3A:B7	02:08:44:11:3A:B7	SDF3F4344FFFF	Apple	uzywany	w_magazynie
15	Macbook	7800017	02:09:44:11:3A:B7	02:10:44:11:3A:B7	3FRWFR34RF4F	Apple	nowy	w_magazynie

Rysunek 20 Widok wprowadzanych danych (identyfikator_wewnetrzny_sprzetu) do wykonania operacji

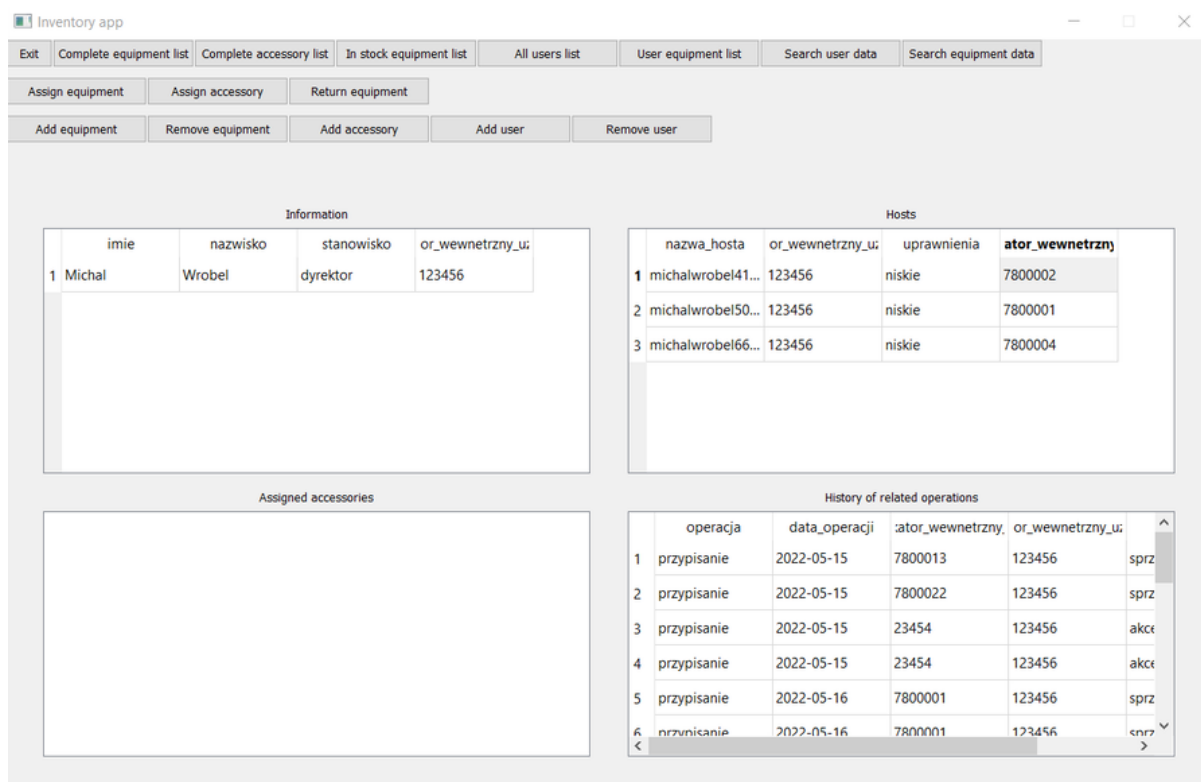
2	Laptop	7800002	00:1B:45:11:3A:B7	10:1B:44:12:3A:B7	D232DQWD3DD	Asus	uzywany	przypisany
---	--------	---------	-------------------	-------------------	-------------	------	---------	------------

Rysunek 21 Widok statusu sprzętu w aplikacji po wykonaniu operacji

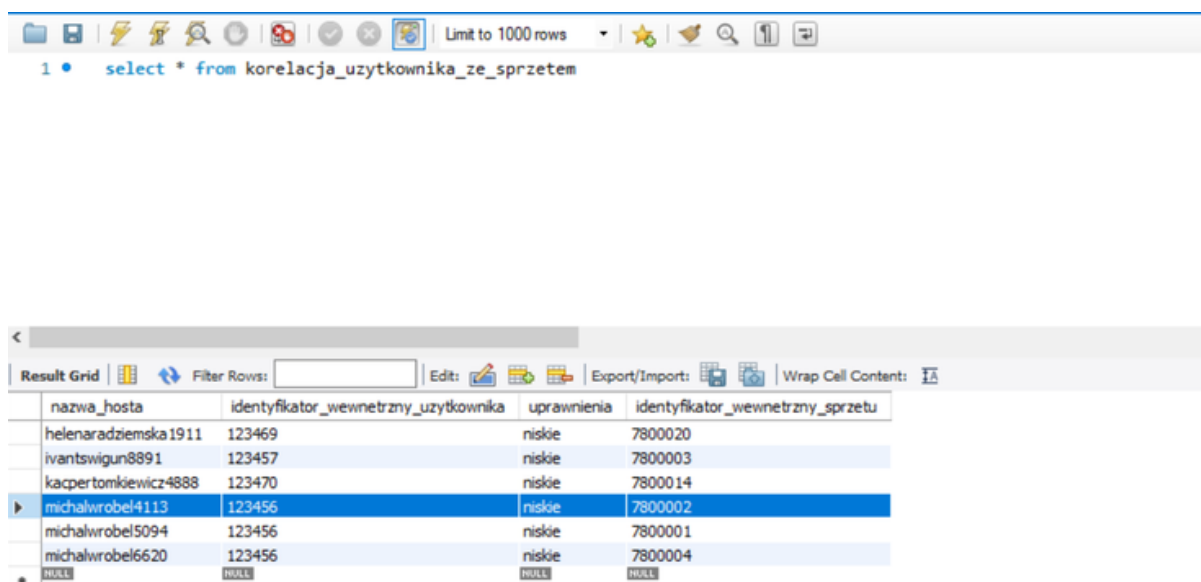
1 • select * from sprzet

	identyfikator_wewnetrzny_sprzetu	adres_mac_ethernet	adres_mac_wifi	numer_seryjny	producent	cykl_zycia	stan
7800001		00:1B:44:11:3A:D7	08:1B:44:12:3A:B7	QWE31324DE32	Asus	uzywany	przypisany
7800002		00:1B:45:11:3A:B7	10:1B:44:12:3A:B7	D232DQWD3DD	Asus	uzywany	przypisany
7800003		00:1B:44:11:2A:B7	30:1B:44:12:3A:B7	23D23D423DQD	Asus	uzywany	przypisany
7800004		00:3B:44:11:3A:B7	40:1B:44:12:3A:B7	FD43FWERF355	Asus	uzywany	przypisany
7800005		00:1B:48:11:3A:B7	05:1B:44:12:3A:B7	344F43FDSF5F5	Asus	nowy	w_magazynie
7800006		00:1B:44:11:3A:B7	06:1B:44:12:3A:B7	G54G6567H7556	Asus	uzywany	w_magazynie
7800008		00:1B:34:11:3A:B7	22:1B:44:12:3A:B7	BRT4B46654654	Asus	nowy	w_magazynie
7800009		00:1B:44:11:DA:B7	11:1B:44:12:3A:B7	QWE313245E32	Asus	nowy	w_magazynie
7800010		10:1B:44:11:3A:B7	13:1B:44:12:3A:B7	QWE61324DE32	Asus	nowy	w_magazynie

Rysunek 22 Lista sprzętu w firmie widoczna w bazie danych wraz ze statusem po wykonaniu operacji



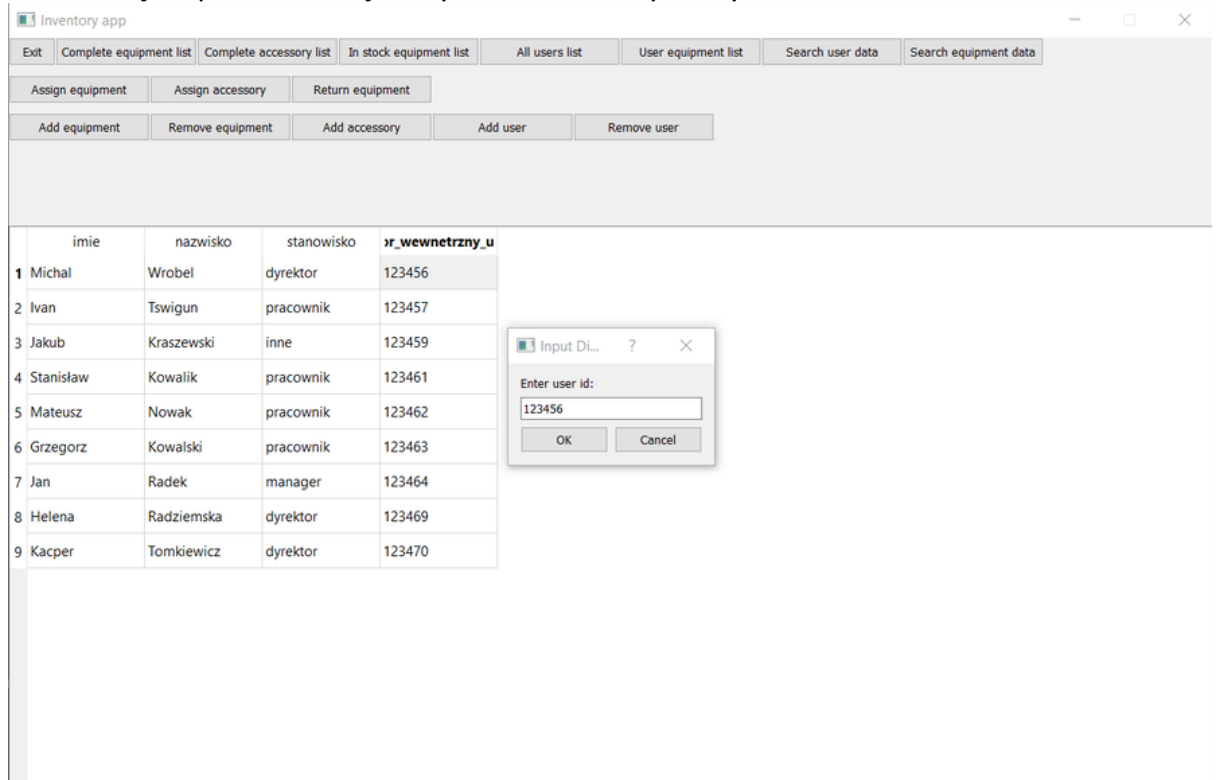
Rysunek 23 Widok menu po wyszukaniu użytkownika, na którym wykonano operację przypisania sprzętu



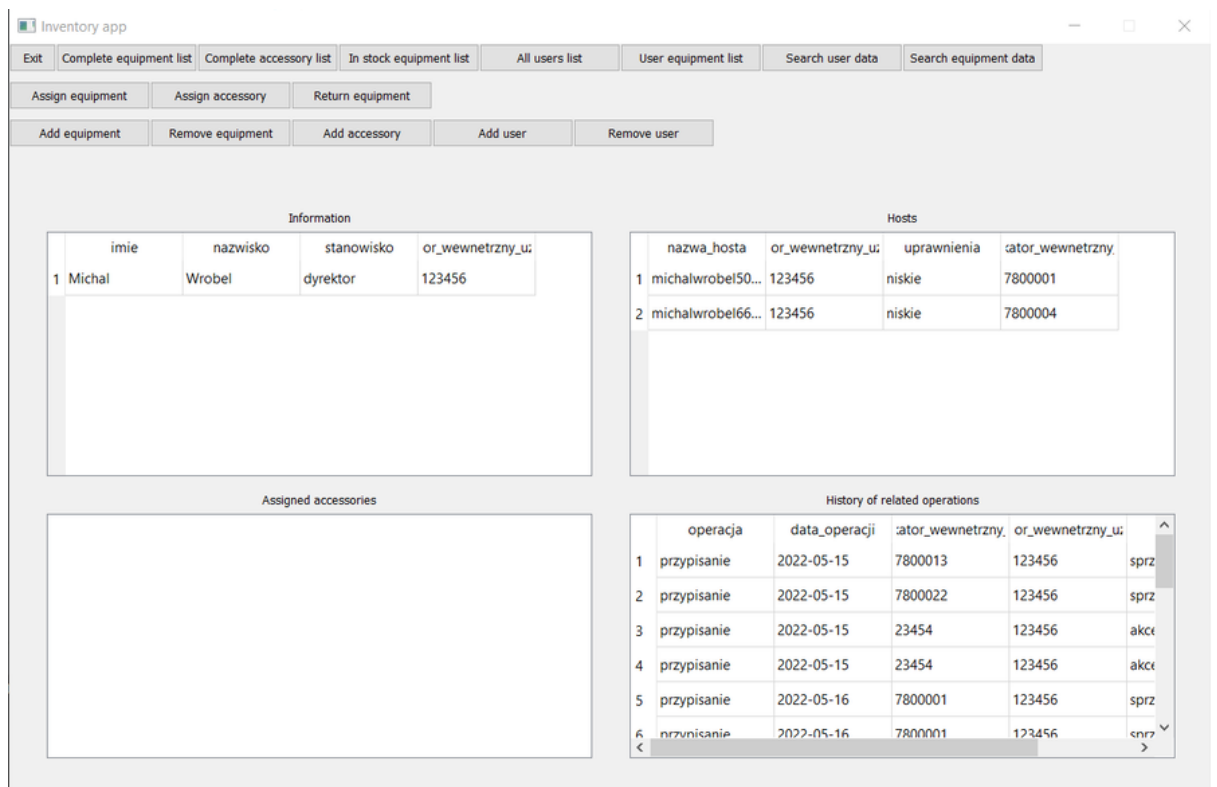
Rysunek 24 Widok tabeli korelacja_uzytkownika_ze_sprzetem po wykonaniu operacji

Funkcja została sprawdzona pod względem przypisania sprzętu do wybranego użytkownika. W tym przypadku została przetestowana na użytkowniku Michał Wróbel o identyfikatorze wewnętrznym 123456 oraz sprzęcie (laptopie) o identyfikatorze wewnętrznym 7800002. Rysunki 17 i 18 przedstawiają tabelę danych na temat sprzętu widoczną w aplikacji oraz bazie danych przed wykonaniem operacji na użytkowniku i sprzęcie (status sprzętu: w magazynie). Rysunek 19 oraz 20 ukazują wprowadzanie danych wymaganych po naciśnięciu przycisku **Assign equipment**. Po wprowadzeniu wymaganych informacji status sprzętu automatycznie się zmienia (status: przypisany) oraz powstaje jego korelacja z wybranym użytkownikiem, tak jak na Rysunku 21, 22 oraz 24. Dodatkowo po wyświetleniu danych o użytkowniku można zobaczyć, że również tam sprzęt jest do niego przypisany (Rysunek 23).

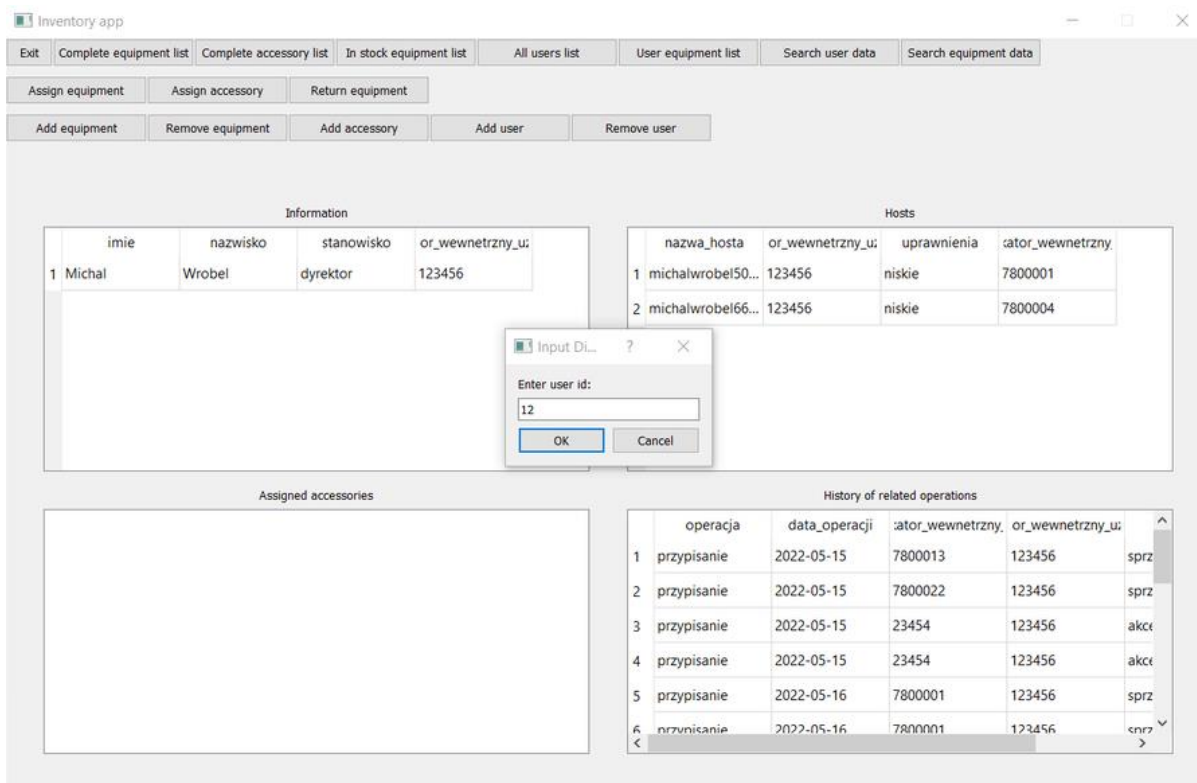
5.1.3 Test funkcji odpowiedzialnej za wyszukiwanie danych użytkownika



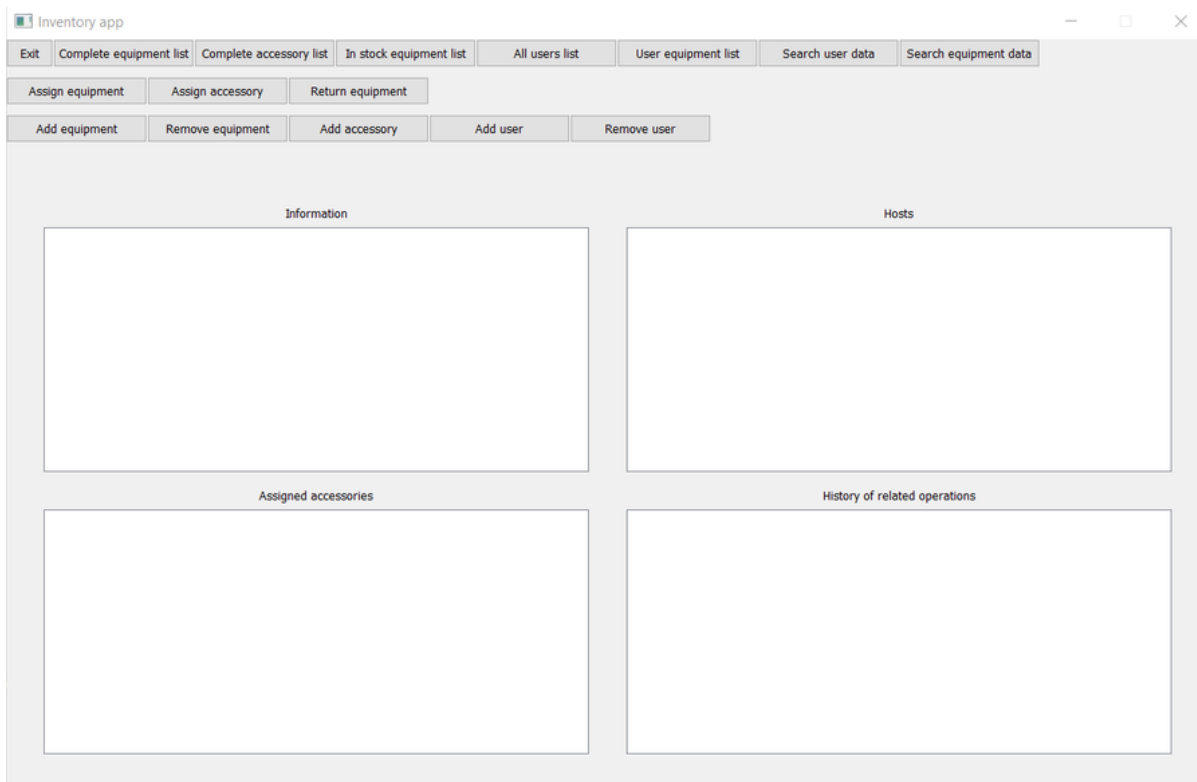
Rysunek 25 Widok danych wprowadzanych do wykonania operacji (identyfikator_wewnetrzny_użytkownika)



Rysunek 26 Widok menu aplikacji po wykonaniu operacji



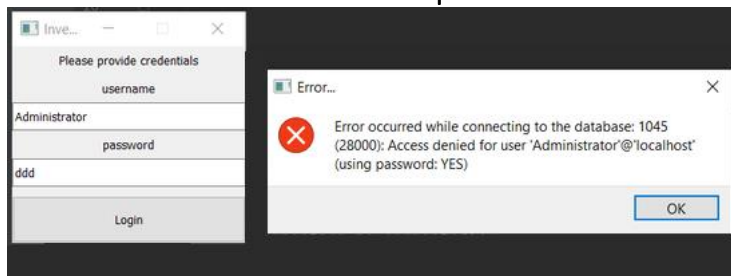
Rysunek 27 Dane wprowadzane w celu wykonania operacji (identyfikator_wewnetrzny_uzytkownika)



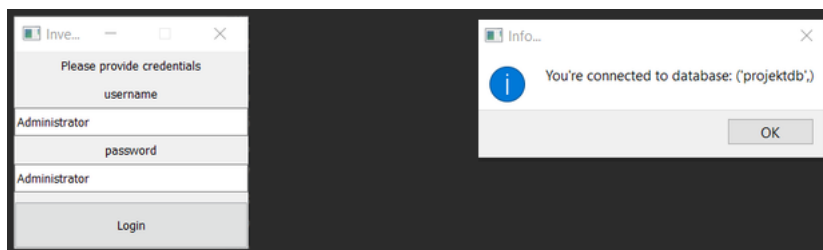
Rysunek 28 Widok menu aplikacji po wykonaniu operacji

Na Rysunku 25 został przedstawiony widok menu po kliknięciu przycisku **Search user** data służącego do wyszukiwania danych o konkretnym pracowniku. W celu wykonania operacji należy podać ID użytkownika, czyli identyfikator_wewnetrzny_uzytkownika. Po jego wprowadzeniu aplikacja pokazuje nam podstawowe dane o użytkowniku, sprzęt oraz akcesoria jakie są do niego przypisane oraz historię operacji. W przypadku podania odpowiednich danych można zaobserwować widok taki jak na Rysunku 26. Sytuacja, w której podany zostaje nieistniejący identyfikator jest przedstawiona na Rysunkach 27 oraz 28.

5.2 Testowanie mechanizmów bezpieczeństwa



Rysunek 29 Komunikat widoczny po wprowadzeniu niewłaściwych danych logowania



Rysunek 30 Komunikat widoczny po wprowadzeniu właściwych danych logowania

Na Rysunku 29 oraz 30 zostały przedstawione dwa scenariusze logowania. Jeden z nich to logowanie niewłaściwe. Po wprowadzeniu danych, które nie znajdują się w bazie aplikacji zostaje wyświetlony komunikat błędu (Rysunek 29), w przeciwnym wypadku użytkownik otrzymuje informację o nawiązaniu połączenia z bazą danych (Rysunek 30).

5.3 Wnioski z testów

Testy ukazały poprawne działanie wybranych funkcji. Sprawdzeniu podlegały tylko niektóre z nich, ponieważ każda kolejna działa analogicznie oraz wyniki są pokazywane w ten sam sposób. Aplikacja jest w pełni sprawna, jednak początkowo wymagała poprawek w celu zmiany widoku użytkownika, który miał w swoim menu widoczne te same przyciski jak administrator, jednak brakiem możliwości wykorzystania ich (po użyciu którejś z funkcji dla niego niedostępnych pojawiało się okno z błędem). Wyniki wprowadzonej modyfikacji można zauważyć w pkt. 3.2.2, gdzie ostatni rząd przycisków nie jest widoczny. Testy widoczne na Rysunkach zostały przeprowadzone z poziomu administratora, co pozwalało na pełny dostęp do każdej z operacji w celu testowania w pełnym zakresie oraz ułatwienia sprawdzenia ich poprawności. Warto wspomnieć, że takim samym testom podlegały operacje z poziomu zalogowania użytkownika i miały identyczne skutki.

6 Podsumowanie

Celem projektu było stworzenie systemu bazodanowego pozwalającego na efektywne zarządzanie sprzętem w firmie. W ramach projektu została zbudowana baza danych umożliwiająca przechowywanie informacji na temat wypożyczonego sprzętu oraz akcesoriów, pracowników firmy i ich korelacji ze urządzeniami. System został zintegrowany z napisaną przez nas aplikacją, co pozwalało na automatyczne przesyłanie oraz aktualizowanie danych. Całość naszego projektu działała lokalnie, co oznacza, że zarówno aplikacja jak i baza danych musi być zainstalowana na komputerze w celu poprawnego działania. Wszystkie dane są zabezpieczone przed nieuprawnionym dostępem oraz istnieje system hierarchiczny, który pozwala na używanie konkretnych operacji jedynie przez wyznaczone do tego jednostki. Podczas realizacji projektu udało nam się zdobyć wiedzę na temat odpowiedniego tworzenia baz danych, aplikacji służącej jako menu w celu obsługi baz danych w prosty sposób oraz połączeniu ze sobą tych dwóch komponentów w celu odpowiedniego działania i współpracy. Ważnym elementem zadania było przeprowadzenie testów, które potwierdziły ostatecznie poprawność naszego programu. Testy wykonywane były manualnie przez co trwały one stosunkowo długo, jednak były niezbędnym elementem w procesie twórczym.

Literatura

- [1] <https://pynative.com/python-mysql-database-connection/>
- [2] <https://pypi.org/project/PyQt5/>
- [3] https://www.w3schools.com/sql/sql_create_db.asp
- [4] <https://www.inmotionhosting.com/support/website/create-diagram-in-mysql-workbench/>
- [5] <https://www.lucidchart.com/pages/ER-diagram-symbols-and-meaning>
- [6] <https://www.python.org/>