

System Zarządzania Pieczarkarnią

Dokumentacja projektu

Zespół projektowy

Miłosz Czaniecki 272872
Krzysztof Czerniachowicz 233182
Iwo Staykov 259259

Prowadzący

dr inż. Marcin Łopuszczyński

Przedmiot

Bazy Danych 2 - Projekt

22 listopada 2024

Spis treści

1	Etap 0 - Opis wycinka rzeczywistości	2
1.1	Przebieg rozwoju produktu	2
1.2	Uprawnienia pracowników zakładu	3
1.3	Modelowe scenariusze	3
2	Etap I - Modelowanie bazy danych	4
2.1	Wymagania funkcjonalne	4
2.2	Wymagania niefunkcjonalne	4
2.3	Diagram przypadków użycia	4
2.4	Przypadki użycia	5
2.5	Diagram ERD	10
3	Etap II - Implementacja bazy danych	11
3.1	Wymagania dostępu do bazy i jej zawartości	11
3.2	Utworzenie skryptu SQL	16
	3.2.1 Zdefiniowanie tabel na podstawie diagramu ERD	16
	3.2.2 Utworzenie i określenie wyzwalaczy i automatyzacji	16
3.3	Wdrożenie bazy danych	18

Spis rysunków

1	Diagram przypadków użycia	5
2	Diagram ERD	10
3	Zrzut ekranu z zaimplementowanej bazy danych	19

1 Etap 0 - Opis wycinka rzeczywistości

1.1 Przebieg rozwoju produktu

Głównym towarem produkowanym przez zakład są pieczarki oraz grzybnie od pieczarek, jednak dopuszczamy wprowadzanie nowych odmian jak boczniaki, grzywa lwa czy trufle.

W zależności od czasu zasiania życie grzyba dzieli się na wiele etapów o przybliżonych czasach trwania:

Etap 1.

Mikoryzacja - zarodniki zasiewane na pożywce ze słodu jęczmiennego i żelatyny agar na szalkach Petriego w sterylnych warunkach - 2 tygodnie

Etap 2.

Inokulacja - narzuta z białej pleśni pieczarki hodowana w wysterylizowanym ziarnie w szklanych naczyniach - 2 tygodnie

Etap 3.

Kolonizacja - inokulowane ziarna zasiewane na wysterylizowanym substracie ze słomy lub innej martwej materii organicznej - 2 tygodnie

Etap 4.

Owocowanie - grzyby wyrastają z zainfekowanej grzybni, wydają owoce do 4 tygodni, po czym wyjałowiona grzybnia oddawana jest do utylizacji lub sprzedawana lokalnym producentom żywności jako kompost.

Etap 5.

Sprzedaż - Owocniki trafiają do magazynu gdzie oczekują na sprzedaż.

Podczas kolejnych etapów życia grzyba upoważniony pracownik dokonuje transferu grzyba z jednego pomieszczenia do drugiego, gdzie numer pomieszczenia odpowiada etapowi życia grzyba. Krok jest wymagany ze względu na różne warunki wymagane na każdym etapie życia grzyba.

Gdy mija okres 6 tygodni, gotowa do owocowania grzybnia zostaje wystawiona w magazynie na sprzedaż przez klientów spoza organizacji LUB w okresie do 10 tygodnia zbierane są z grzybni owocniki i sprzedawane bezpośrednio w tym samym magazynie. Produkty są automatycznie wyceniane według przelicznika. Gdy produkty kupuje się hurtowo, ich ceny tanieją do 10%.

1.2 Uprawnienia pracowników zakładu

W firmie posiadamy 3 stopnie decyzyjności w hierarchi pracowników. Zaczynając od minimalnych do całkowitych uprawnień, są to:

I Stopień - Pracownicy produkcji

Pracownik produkcji może tworzyć nowe partie produktów, dba o prawidłowe warunki partii na danym etapie, przemieszcza je między pomieszczeniami, dokonuje oceny jakości, zajmuje się pracami związanymi z rozmnażaniem, utrzymywaniem i przenoszeniem grzybów.

II Stopień - Kierownicy zmiany

Kierownik zmiany posiada wszystkie uprawnienia Pracowników produkcji, a dodatkowo mogą usuwać partie, dokonywać wycen towaru z magazynu oraz nadzorować jakość i dokonywać zmian przebiegu całego procesu.

III Stopień - Menedżerzy

Menedżer może zarządzać pracownikami, ma prawo dodawać nowe gatunki grzyba w celu rozszerzenia oferty handlowej oraz decyduje kto jest kierownikiem.

1.3 Modelowe scenariusze

Przedstawiamy poniżej przykładowe sytuacje mogące wystąpić w trakcie funkcjonowania naszego zakładu.

1. Pracownik zasiewa nową partię grzybów. Oznacza ją odpowiednim identyfikatorem i zapisuje w dokumentacji.
2. Autoryzowany pracownik widzi, że dana partia musi być przeniesiona do kolejnego etapu. Po wykonaniu zadania uzupełnia się nowe informacje, w tym ocenia jakość danej partii.
3. Menedżer dodaje nowego pracownika i określa jego uprawnienia.
4. Firma postanowiła rozszerzyć ofertę o nowy gatunek grzybów. Menedżer uzupełnia wszystkie potrzebne informacje - takie jak wymagana temperatura, wilgotność lub czas rozwoju na danym etapie.
5. Klient kontaktuje się z odpowiedzialnym za sprzedaż pracownikiem. Dany pracownik widzi jakie partie są gotowe do sprzedaży oraz ich cenę.
6. Jeśli pracownik błędnie wprowadził dane o nowej partii, w celu usunięcia jej musi skontaktować się z pracownikiem o wyższych uprawnieniach kierowniczych.

2 Etap I - Modelowanie bazy danych

2.1 Wymagania funkcjonalne

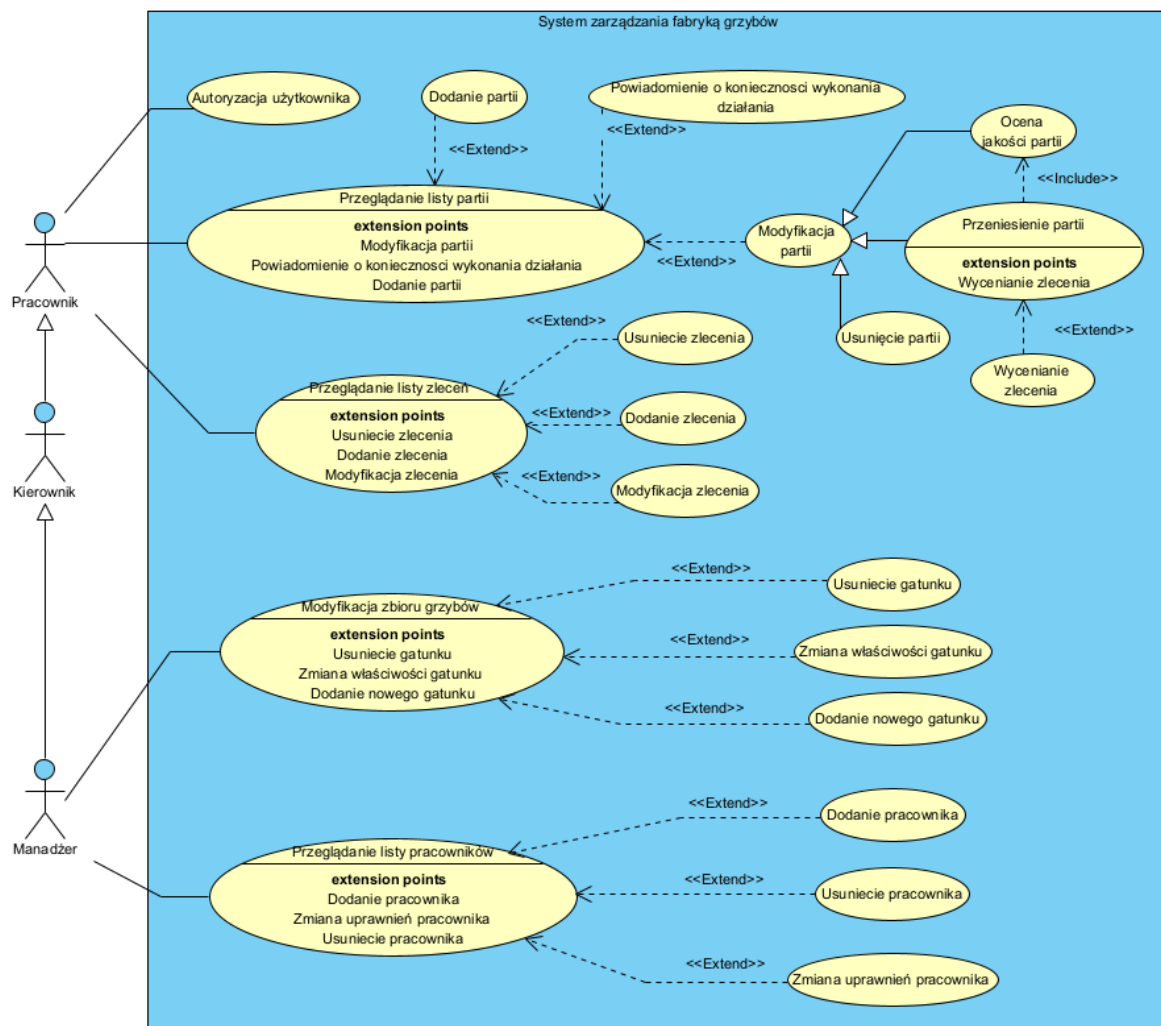
1. Można dodawać do systemu nowe gatunki hodowanych grzybów.
2. Możliwość dodawania nowych partii, które są obecnie zasiewane.
3. Każda partia opisana unikalnym numerem identyfikacyjnym, gatunkiem, datą zasiania i obecnym etapem rozwoju.
4. System powiadamia pracownika o konieczności przeniesienia danej partii do kolejnego pomieszczenia.
5. Dana partia przy zmianie pomieszczenia jest oceniana pod względem jakości.
6. Możliwość śledzenia, jaka partia jest na jakim etapie.
7. Produkty znajdujące się na końcowym etapie są wyceniane i przekazywane do sprzedaży.
8. Cena produktów obliczana jest automatycznie według ustalonego przelicznika.
9. Przy hurtowym zakupie naliczany jest ustalony rabat.
10. System pozwala na zarządzanie uprawnieniami pracowników oraz dodawanie i usuwanie ich z systemu.
11. Uprawnienia podzielone na 3 stopnie – pracownicy produkcji, kierownicy, menedżery.

2.2 Wymagania niefunkcjonalne

1. Należy stworzyć webową aplikację graficzną pozwalającą na łatwe i czytelne korzystanie z bazy danych. Preferowaną technologią jest Spring Boot dla backendu i Thymeleaf dla frontendu.
2. Baza danych ma być stworzona w systemie MySQL.
3. Dostęp do bazy danych i aplikacji powinien być odpowiednio zabezpieczony. Każdy użytkownik musi być autoryzowany.
4. Działania które mogą przeprowadzać użytkownicy są ograniczone w zależności od posiadanych uprawnień.

2.3 Diagram przypadków użycia

Na podstawie wymagań funkcjonalnych i niefunkcjonalnych, oraz na podstawie tekstowych przypadków użycia zaprojektowano diagram przypadków użycia, który przedstawiono na *Rysunku 1*.



Rysunek 1: Diagram przypadków użycia

2.4 Przypadki użycia

1. Dodanie nowego gatunku grzybów:

- **Aktor:** Kierownik.
- **Cel:** Dodanie nowego gatunku hodowanego grzyba do systemu.
- **Zdarzenie inicjujące:** Kierownik wybiera opcję dodania nowego gatunku.
- **Warunki wstępne:** Kierownik jest zalogowany i ma odpowiednie uprawnienia.
- **Warunki końcowe:** Nowy gatunek grzyba zostaje zapisany w systemie.
- **Scenariusz:**
 - (a) Kierownik wybiera opcję dodania nowego gatunku.

- (b) Aplikacja prosi o podanie nazwy nowego gatunku.
- (c) Kierownik wprowadza nazwę gatunku.
- (d) Aplikacja zapisuje nowy gatunek w bazie danych i potwierdza operację.

2. Dodanie partii:

- **Aktor:** Pracownik produkcji.
- **Cel:** Dodanie nowej partii grzybów, które są obecnie zasiewane.
- **Zdarzenie inicjujące:** Pracownik wybiera opcję dodania nowej partii.
- **Warunki wstępne:** Pracownik jest zalogowany i posiada odpowiednie uprawnienia.
- **Warunki końcowe:** Nowa partia zostaje zapisana w systemie.
- **Scenariusz:**
 - (a) Pracownik wybiera opcję dodania nowej partii.
 - (b) Aplikacja prosi o wprowadzenie danych: unikalny numer, gatunek, data zasiania, obecny etap rozwoju.
 - (c) Pracownik wprowadza dane.
 - (d) Aplikacja zapisuje partię w bazie danych i potwierdza operację.

3. Modyfikacja zlecenia

- **Aktor:** Menadżer.
- **Cel:** Modyfikacja zlecenia, które już widnieje w bazie
- **Zdarzenie inicjujące:** Nastąpiła zmiana wymagana przez klienta.
- **Warunki wstępne:** Menadżer jest zalogowany i posiada odpowiednie uprawnienia, a zlecenie jest już utworzone.
- **Warunki końcowe:** Menadżer wprowadza zmiany w zleceniu, które są widoczne zaraz po ich zatwierdzeniu. Pracownik od momentu zatwierdzenia zmiany widzi wprowadzone zmiany w swoim panelu.
- **Scenariusz:**
 - (a) Menadżer wyświetla listę zleceń.
 - (b) Menadżer wyszukuje zlecenie, które należy zmodyfikować i wprowadza zmiany, a następnie zatwierdza.
 - (c) Aplikacja uaktualnia zlecenia w bazie.

- (d) Pracownik będący na zmianie może zauważyć zmiany po wejściu w listę zleceń.

4. Powiadomienie o konieczności wykonania działania:

- **Aktor:** Aplikacja.
- **Cel:** Powiadomienie pracownika o konieczności przeniesienia partii do kolejnego pomieszczenia.
- **Zdarzenie inicjujące:** Partia osiąga etap wymagający przeniesienia.
- **Warunki wstępne:** Aplikacja ma dostęp do danych partii i informacji o etapach rozwoju.
- **Warunki końcowe:** Pracownik otrzymuje powiadomienie.
- **Scenariusz:**
 - (a) Aplikacja analizuje stan partii.
 - (b) Aplikacja identyfikuje partie, które osiągnęły etap wymagający przeniesienia.
 - (c) Aplikacja powiadamia pracownika o konieczności przeniesienia partii.

5. Ocena jakości partii:

- **Aktor:** Pracownik produkcji.
- **Cel:** Ocena jakości partii podczas przenoszenia do nowego pomieszczenia.
- **Zdarzenie inicjujące:** Pracownik wybiera opcję przeniesienia partii.
- **Warunki wstępne:** Partia jest gotowa do przeniesienia.
- **Warunki końcowe:** Aplikacja rejestruje ocenę jakości partii.
- **Scenariusz:**
 - (a) Pracownik wybiera opcję przeniesienia partii.
 - (b) Aplikacja prosi o ocenę jakości partii.
 - (c) Pracownik dokonuje oceny i zatwierdza.
 - (d) Aplikacja rejestruje ocenę jakości i aktualizuje dane partii.

6. Wycenianie zlecenia:

- **Aktor:** Aplikacja.
- **Cel:** Automatyczne obliczenie ceny produktów końcowych.

- **Zdarzenie inicjujące:** Zlecenie jest zrealizowane, ponieważ partia została przeniesiona do ostatniego etapu.
- **Warunki wstępne:** Aplikacja ma dostęp do danych dotyczących partii.
- **Warunki końcowe:** Cena produktów zostaje obliczona i zapisana w systemie.
- **Scenariusz:**
 - (a) Aplikacja identyfikuje zrealizowane zlecenia za pomocą przeniesień partii.
 - (b) Aplikacja pobiera dane dotyczące jakości i wielkości partii.
 - (c) Aplikacja oblicza cenę na podstawie odczytanej jakości, wielkości zlecenia i rodzaju grzyba.
 - (d) Aplikacja zapisuje cenę w systemie.

7. Usunięcie partii:

- **Aktor:** Kierownik.
- **Cel:** Usunięcie partii grzybów, która uległa zepsuciu lub została utworzona pomyłkowo.
- **Zdarzenie inicjujące:** Kierownik wybiera opcję usunięcia partii.
- **Warunki wstępne:** Kierownik jest zalogowany i ma odpowiednie uprawnienia.
- **Warunki końcowe:** Partia zostaje usunięta z systemu.
- **Scenariusz:**
 - (a) Kierownik wybiera opcję usunięcia partii.
 - (b) Aplikacja wyświetla listę partii.
 - (c) Kierownik wybiera partię do usunięcia.
 - (d) Aplikacja prosi o potwierdzenie usunięcia.
 - (e) Kierownik potwierdza.
 - (f) Aplikacja usuwa partię z bazy danych i potwierdza operację.

8. Modyfikacja partii:

- **Aktor:** Pracownik produkcji / Kierownik.
- **Cel:** Wprowadzenie modyfikacji do danych już istniejącej partii.
- **Zdarzenie inicjujące:** Pracownik wybiera opcję edycji partii.

- **Warunki wstępne:** Pracownik jest zalogowany i posiada odpowiednie uprawnienia.
- **Warunki końcowe:** Dane partii zostają zaktualizowane w systemie.
- **Scenariusz:**
 - (a) Pracownik wybiera opcję edycji partii.
 - (b) Aplikacja wyświetla listę partii.
 - (c) Pracownik wybiera partię do edycji.
 - (d) Aplikacja umożliwia edycję wybranych pól (np. gatunek, aktualny pokój, w którym znajduje się partia).
 - (e) Pracownik dokonuje zmian i zatwierdza.
 - (f) Aplikacja zapisuje zmiany w bazie danych i potwierdza operację.

9. Przeniesienie partii:

- **Aktor:** Pracownik produkcji.
- **Cel:** Przeniesienie partii do nowego pomieszczenia.
- **Zdarzenie inicjujące:** Pracownik wybiera opcję przeniesienia partii.
- **Warunki wstępne:** Partia znajduje się na etapie wymagającym zmiany pomieszczenia.
- **Warunki końcowe:** Partia zostaje przeniesiona do nowego pomieszczenia, a aplikacja aktualizuje jej lokalizację.
- **Scenariusz:**
 - (a) Pracownik wybiera opcję przeniesienia partii.
 - (b) Aplikacja wyświetla listę partii gotowych do przeniesienia.
 - (c) Pracownik wybiera partię do przeniesienia.
 - (d) Aplikacja rejestruje przeniesienie i aktualizuje etap oraz lokalizację partii.
 - (e) Aplikacja potwierdza przeniesienie partii.

10. Usunięcie gatunku:

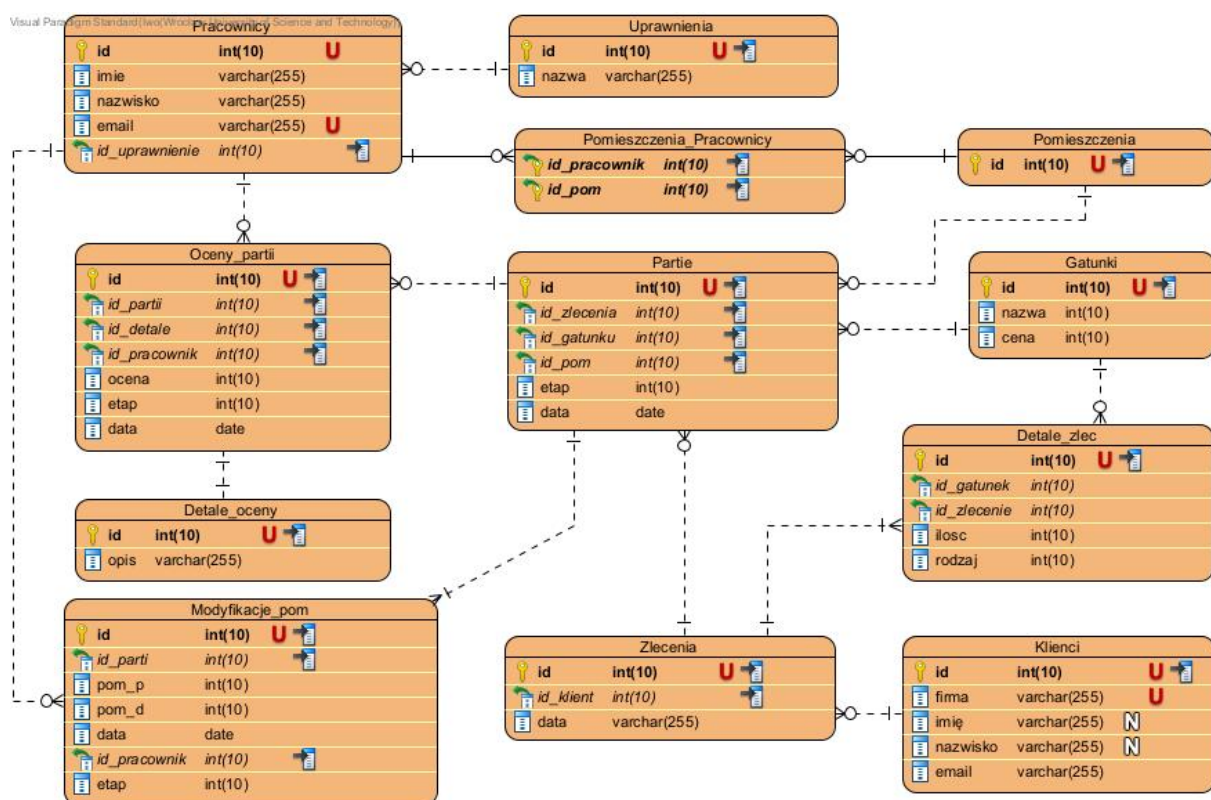
- **Aktor:** Menadżer.
- **Cel:** Usunięcie gatunku z bazy gatunków grzybów.
- **Zdarzenie inicjujące:** Menadżer podjął decyzję o zamknięciu produkcji da-

nego gatunku.

- **Warunki wstępne:** Menadżer jest zalogowany.
- **Warunki końcowe:** Gatunek grzyba zostaje usunięty z bazy danych.
- **Scenariusz:**
 - (a) Menadżer przechodzi do widoku modyfikacji zbioru grzybów.
 - (b) Aplikacja wyświetla listę gatunków.
 - (c) Menadżer wybiera gatunek do usunięcia.
 - (d) Aplikacja sprawdza, czy aktualnie gatunek nie jest w trakcie produkcji.
 - (e) Jeśli żadna partia grzybów nie jest wskazanym gatunkiem, aplikacja usuwa gatunek z bazy danych.

2.5 Diagram ERD

Diagram ERD (Entity Relationship Diagram) przedstawia relację pomiędzy różnymi tabelami w bazie danych poprzez łączące je klucze - wybrane atrybuty jednoznacznie określające każdy rekord w tabeli.



Rysunek 2: Diagram ERD

Na *Rysunku 2* przedstawiono wszystkie 12 encji oraz ich wzajemne relacje w bazie danych

pieczarkarni:

1. Gatunek

Tabela posiada wszystkie gatunki grzybów z oferty handlowej.

2. Zlecenie

Pojedyncze zlecenie może posiadać wiele partii. Ze względu na możliwość sprzedaży grzybów lub grzybni występuje rozróżnienie tego czym jest partia. Partia grzyba to umownie 10kg, a partia grzybni to 20kg, ponieważ z 2kg grzybni wyrasta 1kg grzyba. Jedno zlecenie może posiadać partie o różnych gatunkach.

3. Partia

Zawiera informacje o gatunku i zleceniu do którego należy (*ID_zlecenia*). Etap życia musi być zgodny z numerem pokoju, a etap życia zależy od daty zasadzenia (każdy etap co 2 tygodnie).

4. Pokój

Dany pokój może posiadać wiele zleceń oraz ma przypisanych pracowników.

5. Uprawnienia

Rola oznacza kim w hierarchii jest pracownik (pracownik fizyczny, kierownik albo menadżer) oraz dodatkowo jego uprawnienia do modyfikacji partii lub zarządzania pracownikami.

6. Pracownicy

Oprócz danych identyfikacyjnych (imię, nazwisko, *ID_pracownika*) ma klucz obcy Rola z tabeli Uprawnienia. Pracownik ma przypisany pokój, w którym pracuje.

Widoczny na diagramie *Pokój_pracownicy* jest encją stworzoną automatycznie przez *Visual Paradigm*, ponieważ pojedynczy pracownik może być przypisany do wielu pokoi oraz wielu zleceń naraz.

3 Etap II - Implementacja bazy danych

3.1 Wymagania dostępu do bazy i jej zawartości

W ramach realizacji wymagań dostępu do bazy założono, że w trakcie implementacji aplikacji zostanie utworzona rola w sekcji uprawnień o nazwie *Administrator systemu*, który domyślnie ma całkowity dostęp do bazy oraz jej zawartości. Roli *Administratora systemu* nie uwzględniamy w poniższej liście wymagań dostępu, ponieważ posiada wszystkie możliwe uprawnienia do manipulacji bazą danych.

1. Pracownicy (Tabela: Pracownicy)

- **Użytkownik:** Menedżer
 - **Wstawianie:** Tak (dodawanie nowych pracowników).
 - **Modyfikacja:** Tak (aktualizacja danych osobowych, takich jak e-mail).
 - **Usuwanie:** Tak (usuwanie pracowników, którzy przestali pracować).
 - **Wyszukiwanie:** Tak (przeglądanie listy pracowników i ich uprawnień).
- **Użytkownik:** Kierownik, Pracownik produkcji
 - **Wstawianie:** Nie
 - **Modyfikacja:** Nie
 - **Usuwanie:** Nie
 - **Wyszukiwanie:** Tak (przeglądanie listy pracowników w celu poprawnej komunikacji w zakładzie)

2. Uprawnienia (Tabela: Uprawnienia)

- **Użytkownik:** Menedżer
 - **Wstawianie:** Tak (dodawanie nowych uprawnień).
 - **Modyfikacja:** Tak (zmiana nazw uprawnień).
 - **Usuwanie:** Tak (usuwanie nieużywanych uprawnień).
 - **Wyszukiwanie:** Tak (przeglądanie wszystkich uprawnień przypisanych pracownikom).
- **Użytkownik:** Kierownik, Pracownik produkcji
 - **Wstawianie:** Nie
 - **Modyfikacja:** Nie
 - **Usuwanie:** Nie
 - **Wyszukiwanie:** Nie

3. Pomieszczenia (Tabela: Pomieszczenia)

- **Użytkownik:** Menedżer, Kierownik
 - **Wstawianie:** Tak (dodawanie nowych pomieszczeń).

- **Modyfikacja:** Tak (zmiana danych dotyczących pomieszczeń).
- **Usuwanie:** Tak (usuwanie nieużywanych pomieszczeń).
- **Wyszukiwanie:** Tak (przeglądanie wszystkich pomieszczeń).
- **Użytkownik:** Pracownik produkcji
 - **Wstawianie:** Nie
 - **Modyfikacja:** Tak (zmiana danych dotyczących pomieszczeń).
 - **Usuwanie:** Nie
 - **Wyszukiwanie:** Tak (przeglądanie wszystkich pomieszczeń).

4. Partie (Tabela: Partie)

- **Użytkownik:** Menedżer, Kierownik
 - **Wstawianie:** Tak (rejestrowanie nowych partii).
 - **Modyfikacja:** Tak (aktualizacja danych o partiach, takich jak etap produkcji czy data).
 - **Usuwanie:** Tak
 - **Wyszukiwanie:** Tak (przeglądanie danych partii według zlecenia i etapu produkcji).
- **Użytkownik:** Pracownik produkcji
 - **Wstawianie:** Tak (rejestrowanie nowych partii).
 - **Modyfikacja:** Tak (aktualizacja danych o partiach, takich jak etap produkcji czy data).
 - **Usuwanie:** Nie (usuwanie partii wymaga uprawnień kierownika).
 - **Wyszukiwanie:** Tak (przeglądanie danych partii według zlecenia i etapu produkcji).

5. Oceny partii (Tabela: Oceny_partii)

- **Użytkownik:** Kierownik
 - **Wstawianie:** Tak (dodawanie ocen dla partii).
 - **Modyfikacja:** Tak (korygowanie ocen, np. w przypadku błędu).
 - **Usuwanie:** Tak (usuwanie pomyłek).

- **Wyszukiwanie:** Tak (przeglądanie ocen dla poszczególnych partii).
- **Użytkownik:** Kierownik, Pracownik produkcji
 - **Wstawianie:** Tak (dodawanie ocen dla partii).
 - **Modyfikacja:** Tak (korygowanie ocen, np. w przypadku błędu).
 - **Usuwanie:** Nie (zachowanie pełnej historii ocen).
 - **Wyszukiwanie:** Tak (przeglądanie ocen dla poszczególnych partii).

6. Gatunki (Tabela: Gatunki)

- **Użytkownik:** Menedżer, Kierownik, Pracownik produkcji
 - **Wstawianie:** Tak (dodawanie nowych gatunków produktów).
 - **Modyfikacja:** Tak (zmiana nazwy lub ceny gatunku).
 - **Usuwanie:** Nie (ze względu na spójność danych).
 - **Wyszukiwanie:** Tak (przeglądanie dostępnych gatunków).

7. Zlecenia (Tabela: Zlecenia)

- **Użytkownik:** Menedżer, Kierownik
 - **Wstawianie:** Tak (dodawanie nowych zleceń klientów).
 - **Modyfikacja:** Tak (aktualizacja szczegółów zlecenia).
 - **Usuwanie:** Nie (ze względu na historię zleceń).
 - **Wyszukiwanie:** Tak (przeglądanie zleceń według klientów i dat).
- **Użytkownik:** Pracownik produkcji
 - **Wstawianie:** Nie (obsługą zleceń zajmuje się kierownik)
 - **Modyfikacja:** Nie (obsługą zleceń zajmuje się kierownik)
 - **Usuwanie:** Nie (ze względu na historię zleceń).
 - **Wyszukiwanie:** Tak (przeglądanie zleceń według klientów i dat).

8. Klienci (Tabela: Klienci)

- **Użytkownik:** Menedżer, Kierownik
 - **Wstawianie:** Tak (dodawanie nowych klientów).

- **Modyfikacja:** Tak (aktualizacja danych klientów, np. adres e-mail).
- **Usuwanie:** Nie (utrzymanie historii klientów).
- **Wyszukiwanie:** Tak (wyszukiwanie klientów według nazwiska lub firmy).

- **Użytkownik:** Pracownik

Pracownik nie uczestniczy w relacjach z klientem.

- **Wstawianie:** Nie.
- **Modyfikacja:** Nie.
- **Usuwanie:** Nie.
- **Wyszukiwanie:** Nie.

9. Detale zleceń (Tabela: Detale_zlec)

- **Użytkownik:** Menedżer, Kierownik

- **Wstawianie:** Tak (dodawanie szczegółów do zleceń).
- **Modyfikacja:** Tak (zmiana szczegółów dotyczących ilości czy rodzaju).
- **Usuwanie:** Nie (utrzymanie historii szczegółów zleceń).
- **Wyszukiwanie:** Tak (przeglądanie szczegółów zleceń).

- **Użytkownik:** Pracownik

Utrzymanie i zarządzanie zleceniami są obowiązkami kierownika.

- **Wstawianie:** Nie.
- **Modyfikacja:** Nie.
- **Usuwanie:** Nie.
- **Wyszukiwanie:** Nie.

10. Modyfikacje pomieszczeń (Tabela: Modyfikacje_pom)

- **Użytkownik:** Menedżer, Kierownik

- **Wstawianie:** Tak (rejestracja modyfikacji pomieszczeń).
- **Modyfikacja:** Nie (zachowanie spójnej historii).
- **Usuwanie:** Nie (zachowanie pełnych logów modyfikacji).
- **Wyszukiwanie:** Tak (przeglądanie historii modyfikacji pomieszczeń).

- **Użytkownik:** Pracownik produkcji

Zarządzanie pomieszczeniami nie jest obowiązkiem pracownika produkcji.

- **Wstawianie:** Nie.
- **Modyfikacja:** Nie.
- **Usuwanie:** Nie.
- **Wyszukiwanie:** Nie.

3.2 Utworzenie skryptu SQL

3.2.1 Zdefiniowanie tabel na podstawie diagramu ERD

Dla każdej z encji przydzielono klucz główny oraz klucze obce obrazujące relację danej encji z pozostałymi tabelami. Określono typ danych dla wszystkich kolumn zgodnie z diagramem ERD.

```
1 CREATE TABLE Partie (  
2 id int(10) NOT NULL AUTO_INCREMENT,  
3 id_zlecenia int(10) NOT NULL,  
4 id_gatunku int(10) NOT NULL,  
5 id_pom int(10) NOT NULL,  
6 etap int(10) NOT NULL,  
7 data date NOT NULL,  
8 PRIMARY KEY (id),  
9 UNIQUE INDEX (id),  
10 INDEX (id_zlecenia),  
11 INDEX (id_gatunku),  
12 INDEX (id_pom));  
13  
14 CREATE TABLE Modyfikacje_pom (  
15 id int(10) NOT NULL AUTO_INCREMENT,  
16 id_parti int(10) NOT NULL,  
17 pom_p int(10) NOT NULL,  
18 pom_d int(10) NOT NULL,  
19 data date NOT NULL,  
20 id_pracownik int(10) NOT NULL,  
21 etap int(10) NOT NULL,  
22 PRIMARY KEY (id),  
23 UNIQUE INDEX (id),  
24 INDEX (id_parti),  
25 INDEX (id_pracownik));
```

Listing 1: Tworzenie tabel w MySQL

3.2.2 Utworzenie i określenie wyzwalaczy i automatyzacji

Poniżej przedstawiono automatyzacje oraz wyzwalacze, które usprawniają oraz zabezpieczają korzystanie z bazy danych w zależności od uprawnień użytkowników:

- **Restrict Permissions Modifications:**

Pierwsze trzy wyzwalacze:

`restrict_permissions_modifications,`

`restrict_permissions_modifications_update,`

oraz `restrict_permissions_modifications_delete,`

mają na celu zapewnienie, że tylko użytkownicy posiadający rolę `menedzer` mogą wprowadzać zmiany (wstawianie, aktualizowanie, usuwanie) w tabeli `Uprawnienia`. W przypadku naruszenia tego ograniczenia, wyzwalacze generują komunikat o błędzie, uniemożliwiając dalsze operacje.

- **Logowanie Modyfikacji:**

Wyzwalacz `log_room_modifications` rejestruje modyfikacje w tabeli `Pomieszczenia`. Po każdej aktualizacji danych w tej tabeli, wyzwalacz wstawia odpowiedni wpis do tabeli `Modyfikacje_pom`, zapisując informacje o zmianach, takie jak stary i nowy identyfikator, data oraz etap.

- **Walidacja Dat Partii:**

Dwa wyzwalacze `validate_batch_dates` oraz `validate_batch_dates_update` sprawdzają, czy data ważności wstawianych lub aktualizowanych rekordów w tabeli `Partie` nie jest wcześniejsza niż data produkcji. W przypadku naruszenia tej zasady generowany jest komunikat o błędzie.

- **Blokowanie Usuwania Ocen:**

Wyzwalacz `prevent_ratings_deletion` zapobiega usuwaniu ocen z tabeli `Oceny_partii`. Próba usunięcia rekordu powoduje wygenerowanie komunikatu o błędzie.

- **Walidacja Ceny Gatunków:**

Dwa wyzwalacze

`validate_species_price` oraz

`validate_species_price_update`

walidują ceny gatunków wstawianych lub aktualizowanych w tabeli `Gatunki`. Cena nie może być ujemna, a w przypadku naruszenia tej zasady generowany jest komunikat o błędzie.

- **Zapobieganie Usuwaniu Klientów:**

Wyzwalacz `prevent_client_deletion` uniemożliwia usuwanie klientów z tabeli `Klienci`. Usunięcie klienta powoduje wygenerowanie komunikatu o błędzie, ponieważ historia klienta musi zostać zachowana.

- **Walidacja Ilości w Zleceniach:**

Dwa wyzwalacze

`validate_order_details_quantity` oraz `validate_order_details_quantity_update`

sprawdzają, czy ilość wstawianą lub aktualizowaną w tabeli `Detale_zlec` jest większa niż zero. Jeśli wartość ta jest niepoprawna, wyzwalacz generuje komunikat o błędzie.

- **Zakaz Zmiany Logów Modyfikacji:**

Wyzwalacze

`restrict_modification_changes` oraz

`restrict_modification_changes_delete`

zapobiegają modyfikacji i usuwaniu logów modyfikacji pomieszczeń w tabeli `Modyfikacje_pom`. Próbuując zmienić lub usunąć rekord, użytkownik otrzymuje komunikat o błędzie.

Przykładowa implementacja automatyzacji w MySQL

```
1 CREATE TRIGGER prevent_client_deletion
2 BEFORE DELETE ON Klienci
3 FOR EACH ROW
4 BEGIN
5     SIGNAL SQLSTATE '45000'
6     SET MESSAGE_TEXT = 'Nie można usunąć klienta. Historia musi
7     zostać zachowana.';
8 END;
```

Listing 2: Trigger w MySQL

3.3 Wdrożenie bazy danych

Przygotowany skrypt SQL uruchomiono w systemie MySQL w kontenerze Dockera w środowisku IntelliJ IDEA. W celu przeprowadzenia testu działania bazy, wprowadzono przykładowe dane dla wszystkich encji.

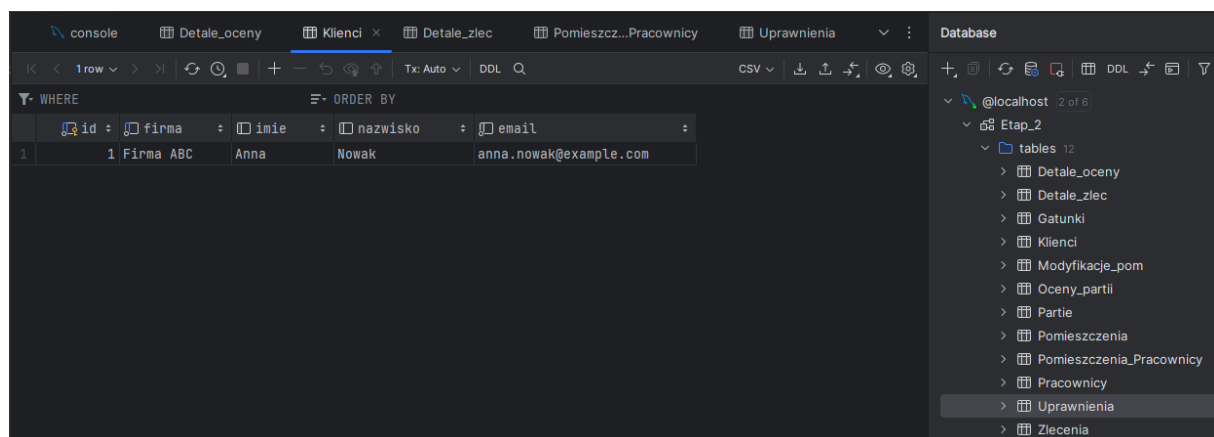
```
1 INSERT INTO Uprawnienia (id, nazwa) VALUES (1, 'Administrator');
2
3 INSERT INTO Pracownicy (id, imie, nazwisko, email, id_uprawnienie)
4     VALUES (1, 'Jan', 'Kowalski', 'jan.kowalski@example.com', 1);
5
6 INSERT INTO Pomieszczenia (id) VALUES (1);
7
8 INSERT INTO Pomieszczenia_Pracownicy (id_pracownik, id_pom) VALUES
9     (1, 1);
10
11 INSERT INTO Detale_oceny (id, opis) VALUES (1, 'Dobra jakość');
```

```

11 INSERT INTO Partie (id, id_zleczenia, id_gatunku, id_pom, etap, data
    ) VALUES (1, 1, 1, 1, 1, '2023-11-21');
12
13 INSERT INTO Gatunki (id, nazwa, cena) VALUES (1, 'Gatunek A', 100);
14
15 INSERT INTO Detale_zlec (id, id_gatunek, id_zleczenie, ilosc, rodzaj
    ) VALUES (1, 1, 1, 100, 1);
16
17 INSERT INTO Zlecenia (id, id_klient, data) VALUES (1, 1, '
    2023-11-21');
18
19 INSERT INTO Klienci (id, firma, imie, nazwisko, email) VALUES (1, '
    Firma ABC', 'Anna', 'Nowak', 'anna.nowak@example.com');
20
21 INSERT INTO Oceny_partii (id, id_partii, id_detale, id_pracownik,
    ocena, etap, data) VALUES (1, 1, 1, 1, 5, 1, '2023-11-21');
22
23 INSERT INTO Modyfikacje_pom (id, id_parti, pom_p, pom_d, data,
    id_pracownik, etap) VALUES (1, 1, 1, 2, '2023-11-21', 1, 1);

```

Listing 3: Przykładowe wartości w bazie danych



Rysunek 3: Zrzut ekranu z zaimplementowanej bazy danych