**Temat: Program obsługujący system informacyjny linii autobusowych**

**1 etap - Specyfikacja wymagań**

**Autorzy:**

# Wymagania funkcjonalne:

## Dla administratora:

1. **Logowanie:** Administrator ma możliwość zalogowania się do systemu za pomocą unikalnego identyfikatora (login) i hasła
2. **Zarządzanie rozkładami jazdy:** Administrator może aktualizować listę tras autobusowych, ich opisy, rozkłady jazdy i opłaty za bilety. Ma też możliwość planowania okresowych zmian w rozkładzie jazdy.
3. **Zarządzanie kontami:** Administrator ma prawo tworzyć, edytować i usuwać konta użytkowników aplikacji. Ma możliwość zarządzania uprawnieniami użytkowników i resetowania haseł.
4. **Backup i odzyskiwanie danych:** Administrator ma możliwość stworzenia backupu baz danych.
5. **Zarządzanie pojazdami:** Administrator ma możliwość dodawania nowych pojazdów, edytowania istniejących i usuwania tych, które nie są już w eksploatacji.
6. **Monitorowanie opóźnień i awarii:** Administrator może wprowadzać informacje o opóźnieniach i awariach, które są wyświetlane pasażerom w czasie rzeczywistym.

## Dla pracownika (kierowcy):

1. **Logowanie i rejestracja przejazdów:** Kierowcy mogą logować się do aplikacji, rejestrować swoje przejazdy i zmieniać status (rozpoczęty, zakończony).
2. **Monitorowanie opóźnień i awarii:** Kierowcy mogą zgłaszać opóźnienia i awarie w czasie rzeczywistym, co pozwoli na aktualizację informacji dla pasażerów.
3. **Powiadomienia o konieczności konserwacji:** System powinien informować kierowcę o konieczności konserwacji pojazdu, kiedy to będzie konieczne.

## Dla klienta:

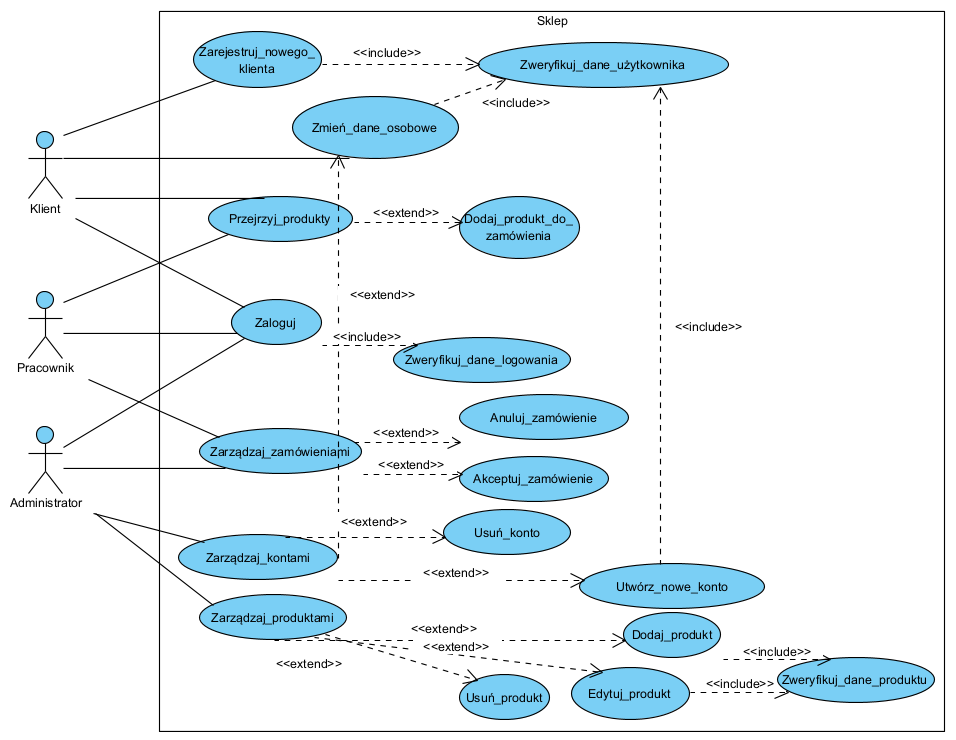
1. **Logowanie i rejestracja:** Pasażerowie mają możliwość tworzenia kont, logowania się i resetowania haseł.
2. **Przeglądanie tras i rozkładów jazdy:** Użytkownicy mogą przeglądać dostępne linie autobusowe, ich opisy i rozkłady jazdy.
3. **Zakup biletów:** Pasażerowie mają możliwość zakupu biletów online, wybierając rodzaj biletu i dokonując płatności.
4. **Śledzenie lokalizacji autobusu:** Możliwość sprawdzania aktualnej lokalizacji autobusów i przewidywanego czasu przyjazdu na danej trasie.

## Dla wszystkich użytkowników:

1. **Weryfikacja danych logowania:** Weryfikacja czy konto istnieje w bazie danych dla podanych danych logowania.
2. **Weryfikacja wprowadzonych danych:** Weryfikacja wprowadzonych danych.

# Wymagania niefunkcjonalne:

1. **Autoryzacja operacji:** Każda operacja jest dostępna w zależności od typu konta użytkownika.
2. **Walidacja wprowadzonych danych:** Weryfikacja czy podane dane przez użytkownika są prawidłowe.
3. **Hashowanie haseł:** Zastosowanie funkcji haszujących do bezpiecznego przechowywania haseł w bazie danych.
4. **Jedno konto administratora:** Istnienie jednego wbudowanego konta administratora w bazie danych

Diagram przypadków użycia:

**Scenariusze przypadków użycia:**  
**Nazwa PU: Zarejestruj\_nowego\_klienta**

**Cel: Zarejestrowanie nowego klienta**

**Warunki początkowe: Uruchomienie programu aplikacji bazodanowej**

**Warunki końcowe: Wprowadzenie unikalnej nazwy użytkownika**

**Scenariusz:**

1. **Należy podać imię, nazwisko, adres, nazwę użytkownika i hasło.**
2. **Należy sprawdzić czy dany użytkownik już nie istnieje i czy hasło jest odpowiednio skonstruowane poprzez PU Zweryfikuj\_wprowadzone\_dane**
3. **Jeżeli wprowadzone dane nie są prawidłowe należy zakończyć PU, w przeciwnym wypadku należy zapisać dane w bazie danych.**

**Nazwa PU: Zweryfikuj\_dane\_użytkownika**

**Cel: Zweryfikowanie poprawności wprowadzonych danych**

**Warunki początkowe: Jest uruchamiany z następujących PU: Zarejestruj\_nowego\_klienta, Utwórz\_nowe\_konto, Zmień\_dane\_osobowe**

**Warunki końcowe: Zwraca wynik określający czy wprowadzone dane są prawidłowe**

**Scenariusz:**

1. **Porównuje nazwę użytkownika z nazwami użytkownika istniejącymi już w bazie danych.**
2. **W przypadku znalezienia użytkownika o podanej nazwie zwraca te nazwę użytkownika i kończy przeszukiwanie bazy danych.**
3. **W przypadku nieznalezienia podanej nazwy użytkownika zwraca wynik negatywny**

**Nazwa PU: Zmień\_dane\_osobowe**

**Cel: Zmiana danych osobowych klienta**

**Warunki początkowe: Uruchomienie programu aplikacji bazodanowej**

**Warunki końcowe: Wprowadzona nazwa użytkownika musi być unikalna**

**Scenariusz:**

1. **Należy podać dane, które chcemy zmienić**
2. **Jeżeli wśród zmienionych danych występuje nazwa użytkownika to uruchamia PU Zwerfikuj\_dane\_użytkownika.**
3. **Jeżeli wprowadzone dane nie są prawidłowe należy zakończyć PU, w przeciwnym wypadku należy zapisać dane w bazie danych.**

**Nazwa PU: Przejrzyj\_produkty**

**Cel: Przegląd dostępnych produktów**

**Warunki początkowe: Uruchomienie programu aplikacji bazodanowej**

**Warunki końcowe: Kryteria przeglądania muszą być poprawne**

**Scenariusz:**

1. **Użytkownik aplikacji podaje określone kryteria wedle których aplikacja filtruje produkty.**
2. **Zgodnie z określonymi kryteriami zwracana jest lista produktów.**

**Nazwa PU: Dodaj\_produkt\_do\_zamówienia**

**Cel: Dodanie wybranego produktu do zamówienia**

**Warunki początkowe: Jest uruchamiany z PU Przejrzyj\_produkty**

**Warunki końcowe: Wybrany produkt musi być dostępny**

**Scenariusz:**

1. **Użytkownik wybiera produkt**
2. **Jeżeli produkt jest dostępny to dodaje go do zamówienia, w przeciwnym wypadku kończy PU**

**Nazwa PU: Zaloguj**

**Cel: Zalogowanie użytkownika**

**Warunki początkowe: Uruchomienie programu aplikacji bazodanowej**

**Warunki końcowe: Użytkownik musi podać prawidłowe dane logowania**

**Scenariusz:**

1. **Należy podać nazwę użytkownika oraz hasło**
2. **Uruchamiany jest PU Zweryfikuj\_dane\_logowania**
3. **Jeżeli podano prawidłowe dane logowania to użytkownik ma dostęp do aplikacji, w przeciwnym wypadku kończymy PU**

**Nazwa PU: Zweryfikuj\_dane\_logowania**

**Cel: Weryfikacja danych logowania użytkownika**

**Warunki początkowe: Jest uruchamiany z PU Zaloguj**

**Warunki końcowe: Zwraca wynik określający czy wprowadzone dane są prawidłowe**

**Scenariusz:**

1. **Porównuje podaną nazwę użytkownika z nazwami użytkowników istniejących już w bazie**
2. **W przypadku znalezienia podanej nazwy użytkownika następuje weryfikacja poprawności wprowadzonego hasła**
3. **Jeżeli zostały podane poprawne dane logowania zwracany jest wynik pozytywny, w innym wypadku kończymy PU**

**Nazwa PU: Zarządzaj\_zamówieniami**

**Cel: Zarządzanie zamówieniami**

**Warunki początkowe: Uruchomienie programu aplikacji bazodanowej**

**Warunki końcowe: Użytkownik musi być administratorem lub pracownikiem**

**Scenariusz:**

1. **Użytkownik wybiera zamówienie, które może zaakceptować korzystając z PU Akcpetuj\_zamówienie lub anulować korzystając z PU Anuluj\_zamówienie**
2. **W przypadku edycji zamówienia należy zapisać zmiany w bazie danych, w innym przypadku należy zakończyć PU**

**Nazwa PU: Anuluj\_zamówienie**

**Cel: Anulowanie zamówienia**

**Warunki początkowe: Jest uruchamiany z PU Zarządzaj\_zamówieniami**

**Warunki końcowe: Zamówienie musi znajdować się w bazie**

**Scenariusz:**

1. **W przypadku anulowania zamówienia jest ono usuwane z bazy danych**
2. **W innym wypadku należy zakończyć PU**

**Nazwa PU: Akceptuj\_zamówienie**

**Cel: Akceptacja zamówienia**

**Warunki początkowe: Jest uruchamiany z PU Zarządzaj\_zamówieniami**

**Warunki końcowe: Zamówienie musi istnieć**

**Scenariusz:**

1. **W przypadku akceptacji zamówienia jest ono dodawane do bazy danych**
2. **W innym wypadku należy zakończyć PU**

**Nazwa PU: Zarządzaj\_kontami**

**Cel: Zarządzanie kontami użytkowników**

**Warunki początkowe: Uruchomienie programu aplikacji bazodanowej**

**Warunki końcowe: Użytkownik musi być administratorem**

**Scenariusz:**

1. **Użytkownik wybiera konto i może edytować dane osobowe korzystając z PU Zmień\_dane\_osobowe lub usunać konto korzystając z PU Usuń\_konto**
2. **W przypadku, kiedy konto nie istnieje użytkownik może je utworzyć korzystając z PU Utwórz\_nowe\_konto**
3. **W wypadku edycji/utworzenia konta należy zapisać zmiany w bazie danych w innym przypadku należy zakończyć PU**

**Nazwa PU: Usuń\_konto**

**Cel: Usunięcie konta użytkownika**

**Warunki początkowe: Jest uruchamiany z PU Zarządzaj\_kontami**

**Warunki końcowe: Wybrane konto musi znajdować się w bazie**

**Scenariusz:**

1. **W przypadku usunięcia konta jest ono usuwane z bazy danych**
2. **W przeciwnym wypadku należy zakończyć PU**

**Nazwa PU: Utwórz\_nowe\_konto**

**Cel: Utworzenia nowego konta użytkownika**

**Warunki początkowe: Jest uruchamiany z PU Zarządzaj\_kontami**

**Warunki końcowe: Wprowadzone dane do konta muszą być unikalne**

**Scenariusz:**

1. **Należy zweryfikować czy podane dane do konta nie znajdują się już w bazie danych za pomocą PU Zweryfikuj\_dane\_użytkownika**
2. **Jeżeli wprowadzone dane są unikalne należy zapisać nowe konto w bazie danych, w przeciwnym wypadku należy zakończyć PU**

**Nazwa PU: Zarządzaj\_produktami**

**Cel: Zarządzanie produktami**

**Warunki początkowe: Uruchomienie programu aplikacji bazodanowej**

**Warunki końcowe: Użytkownik musi być administratorem**

**Scenariusz:**

1. **Użytkownik wybiera produkt, który może edytować korzystając z PU Edytuj\_produkt lub usunąć korzystając z PU Usuń\_produkt**
2. **Jeżeli produkt nie istnieje użytkownik może go utworzyć korzystająć z PU Dodaj\_produkt**
3. **W przypadku edycji/utworzenia produktu należy zapisać zmiany w bazie danych, w przeciwnym wypadku należy zakończyć PU**

**Nazwa PU: Dodaj\_produkt**

**Cel: Dodawanie produktu**

**Warunki początkowe: Jest uruchamiany z PU Zarządzaj produktami**

**Warunki końcowe: Wprowadzone dane produktu muszą być unikalne**

**Scenariusz:**

1. **Należy zweryfikować czy wprowadzone dane produktu są unikalne za pomocą PU Zweryfikuj\_dane\_produktu**
2. **Jeżeli wprowadzone dane produktu są unikalne i poprawne następuje dodanie produktu do bazy danych**
3. **W innym przypadku należy zakończyć PU**

**Nazwa PU: Zweryfikuj\_dane\_produktu**

**Cel: Weryfikacja poprawności danych produktu**

**Warunki początkowe: Jest uruchamiany z PU Dodaj\_produkt lub PU Edytuj\_produkt**

**Warunki końcowe: Zwraca wynik określający czy wprowadzone dane są poprawne lub nie**

**Scenariusz:**

1. **Porównuje dane produktu z danymi produktów znajdujących się już w bazie danych**
2. **W przypadku znalezienia produktu o podanych danych zwracany jest wynik pozytywny i następuje koniec przeszukiwania bazy danych**
3. **W innym przypadku zwracany jest wynik negatywny**

**Nazwa PU: Usuń\_produkt**

**Cel: Usunięcie produktu**

**Warunki początkowe: Jest uruchamiany z PU Zarządzaj\_produktami**

**Warunki końcowe: Wybrany produkt musi znajdować się w bazie**

**Scenariusz:**

1. **W przypadku usunięcia produkt jest usuwany z bazy danych**
2. **W innym wypadku należy zakończyć PU**

**Nazwa PU: Edytuj\_produkt**

**Cel: Edycja produktu**

**Warunki początkowe: Jest uruchamiany z PU Zarządzaj\_produktami**

**Warunki końcowe: Wprowadzone dane produktu muszą być unikalne**

**Scenariusz:**

1. **Należy zweryfikować czy wprowadzone dane produktu nie znajdują się już w bazie danych za pomocą PU Zweryfikuj\_dane\_produktu**
2. **Jeżeli dane produktu nie pokrywają się z danymi z bazy danych należy zaktualizować produkt w bazie danych, w innym wypadku należy zakończyć PU**