Karol Pałka, Damian Zamojski, Krzysztof Zapędzki

Specyfikacja projektu zespołowego “Firma Transportowa”.  
  


**Spis treści**

[Wstęp 3](#__RefHeading___Toc19907_24875677)

[Podział pracy w zespole 3](#__RefHeading___Toc19909_24875677)

[Opis wymagań funkcjonalnych i niefunkcjonalnych 4](#__RefHeading___Toc19911_24875677)

[Wymagania funkcjonalne 5](#__RefHeading___Toc19913_24875677)

[Wymagania niefunkcjonalne 12](#__RefHeading___Toc19915_24875677)

[Diagramy przypadków użycia 14](#__RefHeading___Toc19917_24875677)

[Diagram klas 29](#__RefHeading___Toc19919_24875677)

[Diagramy aktywności 38](#__RefHeading___Toc19921_24875677)

[Diagramy stanów 43](#__RefHeading___Toc19923_24875677)

[Projekt bazy danych 45](#__RefHeading___Toc19925_24875677)

# Wstęp

Niniejsza specyfikacja ma za zadanie przedstawić projekt zespołowy “Firma transportowa”. Projekt został wykonany przez trzech studentów, którzy pracowali nad nim podczas asynchronicznych spotkań w sieci. Firma głównie będzie zajmowała się zarządzaniem personelem, flotą i biletami, na trasie Kraków- Katowice. Rozwój firmy i nowe możliwości rozwoju są jak najbardziej dopuszczalne i pożądane. Jesteśmy otwarci na różne modyfikacje w celu spersonalizowania możliwości osiąganych przez firmę.

# Podział pracy w zespole

Każda osoba otrzyma dokładny zakres zadań do wykonania. Konsultacje będą odbywały się poprzez chat online. Wszystkie części projektu zostaną odpowiednio przeanalizowane przez każdego członka grupy i odpowiednio wprowadzone do projektu.

*Przydział obowiązków:*

Szata graficzna, spis treści, wstęp- Karol Pałka,

Szczegółowy opis projektu- Krzysztof Zapędzki

Wymagania funkcjonalne i niefunkcjonalne- Krzysztof Zapędzki

Diagramy klas, stanów, przypadków użycia- Karol Pałka

Projekt bazy danych- Damian Zamojski,

Szczegółowe opisy pól i relacji w bazie danych- Damian Zamojski

# Opis wymagań funkcjonalnych i niefunkcjonalnych

Wymagania funkcjonalne zostały sporządzone na podstawie informacji i wymagań o projekcie z pliku PDF firma transportowa. Wyszczególniono funkcje które powinny być zawarte w aplikacji oraz opisano je a następnie wyszczególniono zgodnie z przykładem dane wejścia i wyjścia charakterystycznych dla funkcji danych oraz wymagania, techniczne i funkcyjne jakie powinna zawierać funkcja.

Wymagania niefunkcjonalne opisane zostały na podstawie opisu w formie cecha-miara w tabeli, gdzie wymienione zostały na podstawie przykładów logiki i dedukcji liczne wymagania niefunkcjonalne nie zbędne do funkcjonowania aplikacji, również tu znajdują się wszystkie funkcje dotyczące bezpieczeństwa, wymagań sprzętowych oraz do samej produkcji i jej kosztów

# Wymagania funkcjonalne

|  |  |
| --- | --- |
| Nazwa funkcji | Rejestracja użytkownika |
| Opis funkcji | Utworzenie indywidualnego konta użytkownika, gromadzenie punktów |
| Dane wejścia | Imię , nazwisko, hasło użytkownika, adres Email, data urodzenia, numer telefonu |
| Dane wyjścia | Unikalny numer klienta, login |
| wymagania | Połączenie bazy danych z Internetem, hasło o odpowiednich parametrach |

|  |  |
| --- | --- |
| Nazwa funkcji | Logowanie użytkownika |
| Opis funkcji | Zalogowanie się użytkownika do systemu w celu uzyskania pełnej obsługi kont |
| Dane wejścia | Hasło, login |
| Dane wyjścia | Dostęp do danych użytkownika, rezerwacji kursów |
| wymagania | Połączenie bazy danych z Internetem |

|  |  |
| --- | --- |
| Nazwa funkcji | Rezerwacja miejsca |
| Opis funkcji | Klient rezerwuje dany przejazd |
| Dane wejścia | ID użytkownika Godzina kursu, dzień kursu, trasa kursu, dostępne zniżki, punkty za kurs |
| Dane wyjścia | Zajęcie wybranych miejsc w wybranym przejeździe, przekierowanie do zewnętrznych serwisów transakcyjnych , dodanie klienta do listy pasażerów dostępnej dla kierowcy, wysłanie maila z biletem i potwierdzeniem rezerwacji link do cofnięcia transakcji  punkt startowy punkt docelowy |
| wymagania | Do rezerwacji może dojść najpóźniej 2 godziny przed wyjazdem pierwszego transportu w dni wybranego kursu, oraz maksymalnie tydzień naprzód  Zalogowanie się przez klienta |

|  |  |
| --- | --- |
| Nazwa funkcji | Wycofanie rezerwacji |
| Opis funkcji | Funkcja odpowiada za wyrejestrowanie klienta z danego kursu zwalniając mu miejsce jak i również cofa skutki punktowe i finansowe kursu |
| Dane wejścia | ID klienta, link z maila potwierdzającego rejestracje decyzja klienta o odwołaniu |
| Dane wyjścia | Wycofanie punktów za wycofany przejazd, wycofanie zapłaty w zewnętrznym serwisie transakcyjnym, dodanie wycofania do licznika wycofanych przez klienta transakcji |
| wymagania | Zalogowanie się przez klienta, kod z maila potwierdzającego zarezerwowanie miejsca |

|  |  |
| --- | --- |
| Nazwa funkcji | Przypomnienie o rezerwacji |
| Opis funkcji | Przypomina klientowi o zarezerwowanym kursie |
| Dane wejścia | Godzina kursu punkt startowy punkt docelowy |
| Dane wyjścia | Wiadomość automatyczna do klienta |
| wymagania | Automatyczna wiadomość jest wysyłana do klienta |

|  |  |
| --- | --- |
| Nazwa funkcji | Zmiana w kursie |
| Opis funkcji | Automatyczne przesłanie informacji o zmianie kursu na który zapisał się klient |
| Dane wejścia | Zmiana kursu, treść informacji dla klienta |
| Dane wyjścia | Wysłanie informacji o zmianie kursu dla klienta |
| wymagania | Kurs jeszcze do dotarł do punktu w którym dany użytkownik miał do niego dołączyć |

|  |  |
| --- | --- |
| Nazwa funkcji | Dodawanie punktów |
| Opis funkcji | Funkcja dodaje do konta klienta indywidualne punkty 1pkt. +1 km przejechanej trasy |
| Dane wejścia | Trasa kursu którą jedzie klient |
| Dane wyjścia | Przeliczenie km na punkty |
| wymagania | Zalogowanie się na konto użytkownika, wykupienie kursu, i nie wycofanie go |

|  |  |
| --- | --- |
| Nazwa funkcji | Blokada użytkownika |
| Opis funkcji | Funkcja blokuje użytkownika w konsekwencji wielu odwołanych rezerwacji |
| Dane wejścia | Ilość wycofanych rezerwacji użytkownika |
| Dane wyjścia | Blokada rezerwacji kursu na miesiąc dla użytkownika |
| wymagania | Ilość wycofanych rezerwacji użytkownika, jeżeli = 3 to blokada się uruchamia i zeruje liczbę odwołanych rezerwacji |

|  |  |
| --- | --- |
| Nazwa funkcji | Wydawanie punktów |
| Opis funkcji | Operacje wydawanie punktów użytkownika na wybrane promocje |
| Dane wejścia | Punkty użytkownika, wybrana promocja |
| Dane wyjścia | Odjęcie danej liczby punktów użytkownika, zmiana kwoty do opłacenia w zewnętrznym serwisie transakcyjnym |
| wymagania | Zalogowanie się na konto użytkownika,  posiadanie wymaganej liczby punktów  dostępność wybranej oferty promocyjnej |

Funkcje pracownicze

|  |  |
| --- | --- |
| Nazwa funkcji | Tworzenie konta pracowniczego |
| Opis funkcji | Tworzenie kont funkcyjnych dla wybranych pracowników |
| Dane wejścia | typ pracownika, dane osobowe, dane niezbędne pracodawcy, |
| Dane wyjścia | Id konta, dane do logowania |
| wymagania | Uprawnienia administratora/ właściciela firmy |

|  |  |
| --- | --- |
| Nazwa funkcji | Sprawdzanie grafiku pracownika |
| Opis funkcji | Sprawdzanie grafiku pracownika |
| Dane wejścia | Dane logowanie |
| Dane wyjścia | Wyświetlenie grafiku |
| wymagania | Zalogowanie się na konto pracownicze |

|  |  |
| --- | --- |
| Nazwa funkcji | Dane do grafiku |
| Opis funkcji | Pracownicy podają swoje dane do tworzenia przyszłych grafików (dostępny czas) |
| Dane wejścia | Kalendarz z wartościami dostępności i nie dostępności, dostępność i stan samochodów, miejsca zaparkowania dostępność pojazdów do poszczególnych tras |
| Dane wyjścia | Przekazanie danych do wybranego konta |
| wymagania | Zalogowanie na konto pracownicze |

|  |  |
| --- | --- |
| Nazwa funkcji | Sprawdzanie listy pasażerów |
| Opis funkcji | Lista pasażerów dla kierowcy gotowa do wydruku |
| Dane wejścia | Godzina kursu, klienci biorący udział w kursie ich dane osobowe, trasa kursu, godzina kursu data kursu |
| Dane wyjścia | List pasażerów gotowa do wydruku |
| wymagania | Zalogowanie na konto pracownika o charakterze kierowcy |

|  |  |
| --- | --- |
| Nazwa funkcji | Obliczenie pasażerów na poszczególnych odcinkach trasy |
| Opis funkcji | Funkcja oblicza liczbę pasażerów na poszczególnych odcinkach trasy |
| Dane wejścia | Trasa, przystanki , liczba pasażerów wsiadająca na przystanku, liczba pasażerów wysiadających na przystanku |
| Dane wyjścia | Liczba pasażerów na danych odcinkach trasy |
| wymagania |  |

|  |  |
| --- | --- |
| Nazwa funkcji | Zapisywanie kosztów tankowania |
| Opis funkcji | Funkcja zlicza koszt zużycia paliwa na trasie dla danego kierowcy i pojazdu |
| Dane wejścia | Kierowca, pojazd, trasa, przebieg z dojazdami z trasy do parkingu bądź z objazdami |
| Dane wyjścia | Kilometry przejechane / litry zużyte w raportach dla różnych okresów czasu kierowców i pojazdów |
| wymagania | Podawanie przez kierowców rzeczywistych tras przejazdu oraz faktur za tankowanie |

|  |  |
| --- | --- |
| Nazwa funkcji | Rezerwacja miejsca w pojeździe dla klientów nie zarejestrowanych |
| Opis funkcji | Pracownicy sekretariatu mogą zajmować miejsca dla klientów niezarejestrowanych przez swoje konto |
| Dane wejścia | Dane pasażera , dane kursu |
| Dane wyjścia | Dopisanie pasażera do listy pasażerów, bilet za przejazd |
| wymagania | Żeby zająć miejsce dla klienta niezarejestrowanego muszą być wolne miejsca w autobusie zalogowanie na koncie pracowniczym |

|  |  |
| --- | --- |
| Nazwa funkcji | Zakładanie kont klientom niezarejestrowanym |
| Opis funkcji | Pracownik sekretariatu zakładają konta klientom nie zarejestrowanym |
| Dane wejścia | Dane potrzebne do rejestracji |
| Dane wyjścia | Dane rejestracji klienta, automatyczny e mai do klienta z aktywacją konta i ustaleniem hasła |
| wymagania | Dostęp do konta pracownika sekretariatu |

|  |  |
| --- | --- |
| Nazwa funkcji | Ustalanie grafiku kierowców |
| Opis funkcji | Pracownicy sekretariatu ustalają grafik kierowców |
| Dane wejścia | Dane podane do ustalenia grafiku przez kierowców, dane wprowadzone przez pracowników sekretariatu dostępność pojazdów |
| Dane wyjścia | Grafik dla kierowców, w postaci kalendarza |
| wymagania | Dostęp do konta pracownika sekretariatu, pokrycie się grafiku kierowców z dostępnymi samochodami |

Funkcje właściciela

|  |  |
| --- | --- |
| Nazwa funkcji | Ustalanie grafiku sekretariatu |
| Opis funkcji | właściciel ustalają grafik pracowników sekretariatu, poprawki wprowadzone przez właściciela, |
| Dane wejścia | Dane podane do ustalenia grafiku przez sekretariat, |
| Dane wyjścia | Grafik sekretariatu w formie kalendarza |
| wymagania | Zalogowanie na koncie właściciela |

|  |  |
| --- | --- |
| Nazwa funkcji | Ustalanie tras przystanków |
| Opis funkcji | Właściciel ustala trasy przystanków |
| Dane wejścia | Nazwa trasy, przystanki, odległość całej trasy odległość między przystankami |
| Dane wyjścia | Umieszczenie, bądź zmiana trasy w serwisie |
| wymagania | Zalogowanie na koncie właściciela |

|  |  |
| --- | --- |
| Nazwa funkcji | Ustalenie ceny przejazdów |
| Opis funkcji | Właściciel ustala ceny |
| Dane wejścia | Wprowadzony cennik za przejazdy |
| Dane wyjścia | Zmiany obliczania cen przekierowującego do zewnętrznych serwisów transakcyjnych |
| wymagania | Zalogowanie na koncie właściciela |

|  |  |
| --- | --- |
| Nazwa funkcji | Ustalenie punktów za nagrody |
| Opis funkcji | Ustalenie promocji dla użytkowników |
| Dane wejścia | Warunki promocji: liczba punktów potrzebą do skorzystania |
| Dane wyjścia | Wprowadzenie danych o promocji do serwisu, oraz możliwość ich wyboru przed dokonaniem opłaty za przejazd |
| wymagania | Zalogowanie na koncie właściciela |

Inne funkcje

|  |  |
| --- | --- |
| Nazwa funkcji | Zniżka studencka i dziecięca |
| Opis funkcji | Funkcja obniża ceny biletów dla wybranych pasażerów |
| Dane wejścia | Informacja czy bilet jest ulgowy przy zakupie biletu, liczba biletów ulgowych spośród wszystkich biletów |
| Dane wyjścia | Obniżenie ceny wybranych biletów o 30% |
| wymagania | Zalogowanie przez użytkownika biletów ulgowych nie może być więcej niż wszystkich kupionych biletów, bilet ulgowy i dziecięcy nie mogą się pokrywać |

|  |  |
| --- | --- |
| Nazwa funkcji | Zniżka dla dzieci do 5 lat |
| Opis funkcji | Zniżki dla dzieci poniżej 5 lat |
| Dane wejścia | Informacja czy bilet jest dla dziecka przy zakupie biletu, liczba biletów dziecięcych spośród wszystkich biletów |
| Dane wyjścia | 100% rabatu na dany bilet |
| wymagania | Zalogowany użytkownik, biletów dziecięcych należy kupić o co najmniej o jeden mniej niż łącznie wszystkich zakupionych biletów, bilet ulgowy i dziecięcy nie mogą się pokrywać |

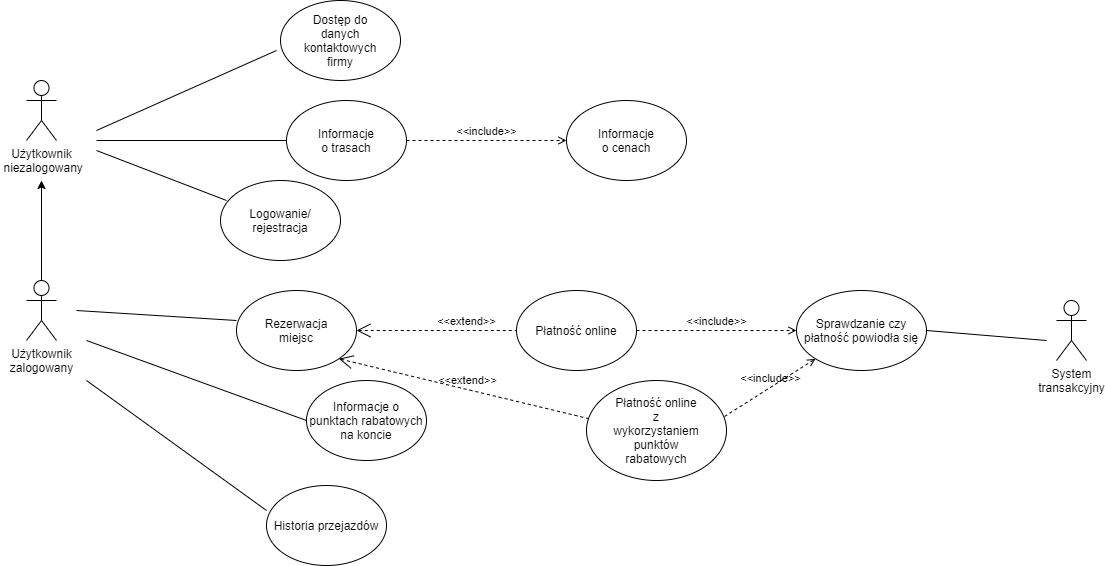
# Wymagania niefunkcjonalne

WYMAGANIA NIEFUNKCJONALNE firmy transportowej

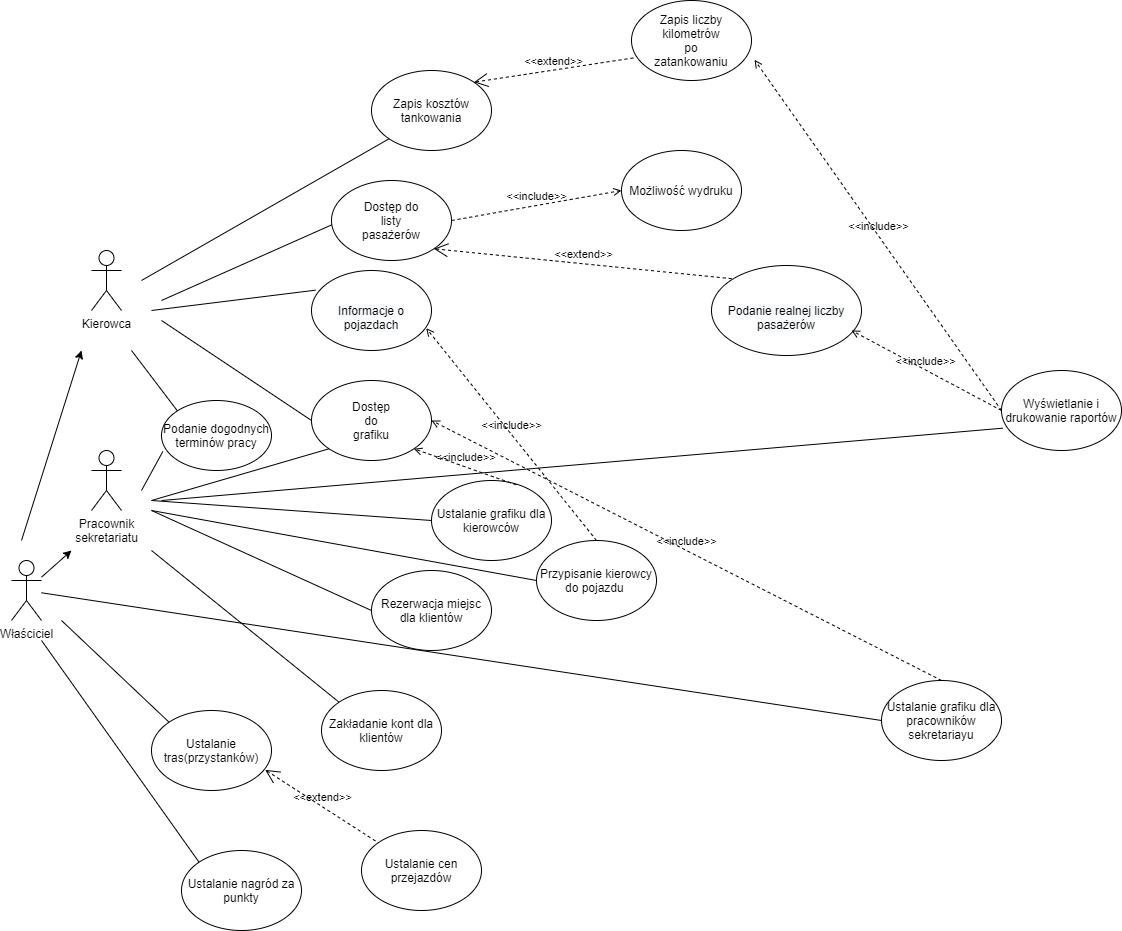
|  |  |
| --- | --- |
|  |  |
| Cecha | Miara |
| niezawodność | - prawdopodobieństwo błędnego wykonania podczas realizacji transakcji  -częstotliwość błędnych wykonań  - Średni czas między błędami wykonań  - minimalna prędkość przesyłu danych potrzebna do poprawnego wykonania |
| Przepustowość | - maksymalna ilość użytkowników i pracowników mogących jednocześnie dokonywać działań w aplikacji bez przeciążenia jej i serwerów  -maksymalna prędkość przesyłu danych odbierana i przetwarzana na bieżąco przez serwer |
| Dostępność | -System musi być dostępny 24h/d 7 dni w tygodniu  -maksymalny czas niedostępności to ?? |
| Rozmiar | -Wymagana pamięć ram  -wymagana pamięć |
| Prędkość | -minimalny i maksymalny czas ładowania formularzy w zależności od mocy łącza internetowego użytkownika  - liczba transakcji na sekundę |
| Uniwersalność | Działanie strony i aplikacji na różnych przeglądarkach systemach operacyjnych, i systemach mobilnych |
| przenośność | - procent kodu zależnego od platformy docelowej  -koszt przeniesienia na nową platformę |
| Dodatkowe usługi poza stworzeniem aplikacji | -Szkolenia  -wsparcie pogwarancyjne  -gwarancje  -koszta utrzymania aplikacji |
| Przywracalność systemu | -rodzaj i sposób tworzenia kopii zapasowych  -przywracanie danych w wypadku awarii bądź innego powodu ich utraty |
| grafika | - kolorystyka aplikacji  -ułożenie interfejsu  -układy formularzy  -responsywność interface-u  -placement loga firmy na stronie  -Rozmiar czcionki będzie dostosowywany do rozmiaru aplikacji na urządzeniu |
| Przejrzystość funkcjonowania | - wzory dokumentów i raportów generowanych przez aplikacje  -dane i wyliczenia umieszczane w raportach |
| Rzetelność aplikacji | -liczba błędów po wdrożeniu na liczbę godzi z niej korzystania i liczbę wykonanych transakcji |
| Bezpieczeństwo | - spełnienie określonych standardów bezpieczeństwa w związku z ochroną danych osobowych klientów danych firmy etc.  - identyfikacja użytkowników  - autentykacja użytkowników  - Wykrywanie ataków  - odporność na ataki  - Lokowanie ataków DOS(denial of service)  - uniemożliwianie zaprzeczeniu uczestnictwa w transakcji  - ochrona prywatności użytkowników  - sprawdzalność zabezpieczeń  -odporność podsystemu zabezpieczeń na zmiany dokonywane w innych częściach systemu  - bezpieczeństwo fizyczne realizowane na poziomie uprawnień dostępu do plików |
| Utrzymanie systemu | -koszt stworzenia aplikacji  - koszt utrzymania aplikacji  -koszt utrzymania bazy danych  -Koszt utrzymania serwera |
| Dostęp do informacji | -łatwość znalezienia informacji na stronie  - umieszczenie i możliwość edycji informacji na stronie  - łatwość dostępu do danych kontaktowych |
| Modułowość | -aplikacja powinna posiadać poszczególne moduły: nie zalogowany, użytkownicy, pracowniczy, administracyjny |
| Wsparcie | -czas usunięcia awarii systemu  -dedykowane adres e-mail do zgłaszania administratorom i twórcą programu błędów  -monitorowanie błędów aplikacji  -dostęp do zgłaszania błędów będą miał wszyscy zarejestrowaniu użytkownicy aplikacji |

# Diagramy przypadków użycia

**Diagram przypadków użycia numer 1(klienci):**



**Diagram przypadków użycia numer 2(pracownicy):**



Id: UC1

|  |  |
| --- | --- |
| Nazwa | Logowanie/ rejestracja |
| Aktorzy | Użytkownik niezalogowany |
| Opis | Użytkownik niezalogowany może zalogować lub zarejestrować się w systemie |
| Warunki wstępne | Do logowania należy przypomnieć sobie login i hasło. |
| Warunki końcowe | Użytkownik jest zalogowany lub system wysyła SMS do użytkownika i czeka na zakończenie procesu rejestracji. |
| Główny przepływ zdarzeń | 1. Użytkownik klika “Zaloguj się”.  2. Wpisuje login i hasło.  3. Jest zalogowany.  Lub  1. Użytkownik klika w przycisk „Rejestracja” znajdujący się na stronie firmy.  2.Wypełnia formularz z danymi  3. System wysyła SMS do użytkownika i czeka na potwierdzenie  4. Proces rejestracji ukończony |
| Alternatywny przepływ zdarzeń | 3. Wyświetla się okno z informacją o nieprawidłowym loginie lub haśle. Powrót do kroku 2.  Lub  3. Wyświetla się okno z informacją o nieprawidłowym wypełnieniu formularza, powrót do kroku 2. |

Id: UC2

|  |  |
| --- | --- |
| Nazwa | Dostęp do danych kontaktowych firmy |
| Aktorzy | Użytkownik niezalogowany/ zalogowany |
| Opis | Użytkownik ma możliwość sprawdzenia kanałów kontaktu z firmą. |
| Warunki wstępne | Brak |
| Warunki końcowe | Brak |
| Główny przepływ zdarzeń | 1. Kliknięcie przycisku „Kontakt” na dole strony internetowej firmy.  2. Informacje wyświetlone na ekranie. |
| Alternatywny przepływ zdarzeń | Brak |

Id: UC3

|  |  |
| --- | --- |
| Nazwa | Informacje o trasach |
| Aktorzy | Użytkownik niezalogowany/ zalogowany |
| Opis | Użytkownik ma możliwość uzyskania informacji o trasach. |
| Warunki wstępne | Brak. |
| Warunki końcowe | Brak |
| Główny przepływ zdarzeń | 1. Kliknięcie przycisku „Wyświetl trasy” na stronie internetowej firmy.  2. Wybór interesującej użytkownika trasy.  3. Wyświetlenie na ekranie szczegółów danej trasy. |
| Alternatywny przepływ zdarzeń | Brak |

Id: UC4

|  |  |
| --- | --- |
| Nazwa | Informacje o cenach |
| Aktorzy | Użytkownik niezalogowany/ zalogowany |
| Opis | Użytkownik ma możliwość uzyskania informacji o cenach. |
| Warunki wstępne | Uzyskanie informacji o trasie za pomocą UC3 |
| Warunki końcowe | Brak |
| Główny przepływ zdarzeń | 1. Kliknięcie przycisku „Wyświetl ceny na tej trasie”.  2. Ceny są wyświetlone na stronie. |
| Alternatywny przepływ zdarzeń | Brak |

Id: UC5

|  |  |
| --- | --- |
| Nazwa | Rezerwacja miejsc |
| Aktorzy | Użytkownik zalogowany |
| Opis | Możliwość zarezerwowania miejsca na danej trasie. |
| Warunki wstępne | Brak |
| Warunki końcowe | Potwierdzenie chęci rezerwacji kodem SMS |
| Główny przepływ zdarzeń | 1. Wybór trasy na stronie internetowej firmy.  2. Wybór daty odjazdu/ odjazdu i przyjazdu.  3. Wybór godziny kursu.  4. Zatwierdzenie rezerwacji przyciskiem potwierdź.  5. Wpisanie kodu SMS.  6. Rezerwacja jest potwierdzona |
| Alternatywny przepływ zdarzeń | 6. Kod SMS jest błędny, powrót do punktu 4. |

Id: UC6

|  |  |
| --- | --- |
| Nazwa | Płatność online |
| Aktorzy | Użytkownik zalogowany |
| Opis | Możliwość dokonania płatności za rezerwację. |
| Warunki wstępne | Dokonanie rezerwacji za pomocą UC5 |
| Warunki końcowe | Potwierdzenie płatności przez system transakcyjny |
| Główny przepływ zdarzeń | 1. Wybranie sposobu płatności.  2. Dokonanie przelewu.  3. Uzyskanie potwierdzenia przez system transakcyjny.  4. Otrzymanie biletu na telefon/ e-mail. |
| Alternatywny przepływ zdarzeń | 3. Brak potwierdzenia wpłaty przez system transakcyjny.  4. Powrót do przypadku UC5. |

Id: UC7

|  |  |
| --- | --- |
| Nazwa | Płatność online z wykorzystaniem punktów rabatowych |
| Aktorzy | Użytkownik zalogowany |
| Opis | Możliwość dokonania płatności za rezerwację z uwzględnieniem posiadanych punktów rabatowych. |
| Warunki wstępne | Dokonanie rezerwacji za pomocą UC5 |
| Warunki końcowe | Potwierdzenie płatności przez system transakcyjny |
| Główny przepływ zdarzeń | 1. Wybranie możliwej zniżki, uzyskanej dzięki posiadanym punktom rabatowym.  2. Wybranie sposobu płatności.  3. Dokonanie przelewu.  4. Uzyskanie potwierdzenia przez system transakcyjny.  5. Otrzymanie biletu na telefon/ e-mail. |
| Alternatywny przepływ zdarzeń | 3. Brak potwierdzenia wpłaty przez system transakcyjny.  4. Powrót do przypadku UC5. |

Id: UC8

|  |  |
| --- | --- |
| Nazwa | Sprawdzanie czy płatność powiodła się |
| Aktorzy | System transakcyjny |
| Opis | System sprawdza, czy płatność doszła do skutku. |
| Warunki wstępne | Dokonanie przelewu punkt 3 UC6 lub UC7 |
| Warunki końcowe | Brak |
| Główny przepływ zdarzeń | 1. System sprawdza, czy pieniądze zostały zaksięgowanie na koncie firmy.  2. Wysłanie potwierdzenia do użytkownika o poprawnym dokonaniu płatności. |
| Alternatywny przepływ zdarzeń | 2. Pieniędzy nie ma na koncie, brak potwierdzenia. |

Id: UC9

|  |  |
| --- | --- |
| Nazwa | Historia przejazdów |
| Aktorzy | Użytkownik zalogowany |
| Opis | Użytkownik może sprawdzić historię swoich zakupionych biletów. |
| Warunki wstępne | Brak |
| Warunki końcowe | Brak |
| Główny przepływ zdarzeń | 1. Kliknięcie w przycisk „Historia Twoich przejazdów” w prawym górnym rogu strony internetowej firmy.  2. Na stronie zostanie wyświetlona historia przejazdów. |
| Alternatywny przepływ zdarzeń | Brak |

Id: UC10

|  |  |
| --- | --- |
| Nazwa | Informacje o punktach rabatowych na koncie |
| Aktorzy | Użytkownik zalogowany |
| Opis | Użytkownik może uzyskać informację o liczbie punktów rabatowych na swoim koncie. |
| Warunki wstępne | Brak |
| Warunki końcowe | Brak |
| Główny przepływ zdarzeń | 1. Kliknięcie w przycisk „Punkty rabatowe” w lewym górnym rogu strony internetowej firmy.  2. Na ekranie zostaną wyświetlone informacje o liczbie punktów rabatowych na koncie użytkownika. |
| Alternatywny przepływ zdarzeń | Brak |

Id: UC11

|  |  |
| --- | --- |
| Nazwa | Ustalanie nagród za punkty |
| Aktorzy | Właściciel |
| Opis | Właściciel może zmieniać nagrody i ustalać nowe. |
| Warunki wstępne | Zalogowanie na konto właściciela. |
| Warunki końcowe | Zatwierdzenie hasłem właściciela. |
| Główny przepływ zdarzeń | 1. Właściciel musi być zalogowany na koncie admina.  2. Kliknięcie przycisku „Zmień/ dodaj nagrody”  3. Wpisanie/zmiana nazwy i ceny punktowej za nową/ zmienioną nagrodę.  4. Kliknięcie przycisku „Potwierdź”.  5. Wpisanie hasła admina.  6. Nagrody zostały zaktualizowane. |
| Alternatywny przepływ zdarzeń | 6. Hasło jest niepoprawne. Powrót do punktu 3. |

Id: UC12

|  |  |
| --- | --- |
| Nazwa | Ustalanie tras(przystanków) |
| Aktorzy | Właściciel |
| Opis | Właściciel może zmieniać trasy i ustalać nowe. |
| Warunki wstępne | Zalogowanie na konto właściciela. |
| Warunki końcowe | Zatwierdzenie hasłem właściciela. |
| Główny przepływ zdarzeń | 1. Właściciel musi być zalogowany na koncie admina.  2. Kliknięcie przycisku „Zmień/ dodaj trasy(przystanki)”  3. Wpisanie/zmiana trasy i przystanków.  4. Kliknięcie przycisku „Potwierdź”.  5. Wpisanie hasła admina.  6. Trasy zostały zaktualizowane. |
| Alternatywny przepływ zdarzeń | 6. Hasło jest niepoprawne. Powrót do punktu 3. |

Id: UC13

|  |  |
| --- | --- |
| Nazwa | Ustalanie cen przejazdów |
| Aktorzy | Właściciel |
| Opis | Zmiana cen przejazdów. |
| Warunki wstępne | Spełnienie pkt.2 UC12 |
| Warunki końcowe | Zatwierdzenie hasłem właściciela. |
| Główny przepływ zdarzeń | 1. Właściciel musi być zalogowany na koncie admina.  2. Kliknięcie przycisku „Zmień/ dodaj trasy(przystanki)”  3. Wybranie przycisku „Zmień cenę trasy” przy interesującej trasie.  4. Wpisanie nowej ceny.  5. Kliknięcie przycisku „Potwierdź”  6. Wpisanie hasła admina.  7. Ceny zostały zaktualizowane. |
| Alternatywny przepływ zdarzeń | 6. Hasło jest niepoprawne. Powrót do punktu 3. |

Id: UC14

|  |  |
| --- | --- |
| Nazwa | Zakładanie kont dla klientów |
| Aktorzy | Pracownik sekretariatu/ właściciel |
| Opis | Osoba może założyć konto na prośbę klienta, który nie chce/ nie potrafi założyć konta samemu. |
| Warunki wstępne | Brak |
| Warunki końcowe | Zatwierdzenie hasłem. |
| Główny przepływ zdarzeń | 1. Kliknięcie przycisku „Dodaj użytkownika”  2. Wpisanie danych podanych w papierowym formularzu przez użytkownika.  3. Zatwierdzenie przyciskiem „Potwierdź”.  4.Wpisanie hasła pracownika.  5. Użytkownik został dodany. |
| Alternatywny przepływ zdarzeń | 5. Hasło niepoprawne powrót do punktu 2. |

Id: UC15

|  |  |
| --- | --- |
| Nazwa | Ustalanie grafiku dla pracowników sekretariatu |
| Aktorzy | Właściciel |
| Opis | Właściciel może ustalić grafik dla pracowników sekretariatu, mając wgląd w obecny grafik. |
| Warunki wstępne | Zalogowanie na konto właściciela. |
| Warunki końcowe | Zatwierdzenie hasłem właściciela. |
| Główny przepływ zdarzeń | 1. Właściciel musi być zalogowany na koncie admina.  2. Kliknięcie przycisku „Zmień grafik dla pracowników sekretariatu”  3. Kliknięcie na zdjęcie konkretnego pracownika.  4. Ustalenie nowego grafiku.  5. Kliknięcie przycisku „Potwierdź”.  6. Wpisanie hasła admina.  7. Nowe grafiki zostały zaktualizowane. |
| Alternatywny przepływ zdarzeń | 7. Hasło jest niepoprawne. Powrót do punktu 4. |

Id: UC16

|  |  |
| --- | --- |
| Nazwa | Rezerwacja miejsc dla klientów |
| Aktorzy | Pracownik sekretariatu/ właściciel |
| Opis | Uprawniona osoba może zarezerwować miejsce dla klienta drogą telefoniczną. |
| Warunki wstępne | Klient prosi o rezerwację |
| Warunki końcowe | Zatwierdzenie hasłem |
| Główny przepływ zdarzeń | 1. Kliknięcie przycisku „Utwórz nową rezerwację”  2. Wpisanie danych klienta oraz trasy, daty i godziny przejazdu.  3. Kliknięcie przycisku „Potwierdź”.  4. Zatwierdzenie rezerwacji hasłem osoby uprawnionej.  5. Nowa rezerwacja została utworzona. |
| Alternatywny przepływ zdarzeń | 4. Brak miejsc, powrót do punktu 2. i wybranie innej godziny/daty/ trasy przejazdu  lub  5. Hasło jest niepoprawne. Powrót do punktu 2. |

Id: UC17

|  |  |
| --- | --- |
| Nazwa | Przypisanie kierowcy do pojazdu |
| Aktorzy | Pracownik sekretariatu/ właściciel |
| Opis | Przypisanie kierowcy do wolnego pojazdu na dany przejazd. |
| Warunki wstępne | Brak |
| Warunki końcowe | Zatwierdzenie hasłem. |
| Główny przepływ zdarzeń | 1. Kliknięcie przycisku „Przypisz kierowcę do pojazdu”  2. Wybranie kierowcy, a następnie wolnego pojazdu.  3. Kliknięcie przycisku „Potwierdź”  4. Zatwierdzenie przypisania hasłem osoby uprawnionej  5. Kierowca został przypisany |
| Alternatywny przepływ zdarzeń | 5. Hasło jest niepoprawne. Powrót do punktu 2. |

Id: UC18

|  |  |
| --- | --- |
| Nazwa | Ustalanie grafiku dla kierowców |
| Aktorzy | Pracownik sekretariatu/ właściciel |
| Opis | Możliwość zmiany, ustalenia grafików dla kierowców, mając wgląd w obecny grafik. |
| Warunki wstępne | Brak |
| Warunki końcowe | Zatwierdzenie hasłem. |
| Główny przepływ zdarzeń | 1. Kliknięcie przycisku „Zmień grafik dla kierowców”  2. Kliknięcie na zdjęcie konkretnego kierowcy.  3. Ustalenie nowego grafiku.  4. Kliknięcie przycisku „Potwierdź”.  5. Wpisanie hasła osoby uprawnionej.  6. Nowe grafiki zostały zaktualizowane. |
| Alternatywny przepływ zdarzeń | 6. Hasło jest niepoprawne. Powrót do punktu 2. |

Id: UC19

|  |  |
| --- | --- |
| Nazwa | Wyświetlanie i drukowanie raportów |
| Aktorzy | Pracownik sekretariatu/ właściciel |
| Opis | Osoba uprawniona ma możliwość wyświetlania i drukowania raportów, mając wgląd w zapisy kierowców o realnej liczbie pasażerów i liczbie przejechanych kilometrów po zatankowaniu. |
| Warunki wstępne | Brak |
| Warunki końcowe | Brak |
| Główny przepływ zdarzeń | 1. Kliknięcie przycisku „Wyświetl raporty”.  2. Wybranie interesującej kategorii raportów.  3. Kliknięcie w interesujący raport.  4. Raport został wyświetlony. aby go wydrukować należy wcisnąć ikonę drukarki w prawym górnym rogu ekranu. |
| Alternatywny przepływ zdarzeń | 1. Kliknięcie przycisku „Wyświetl raporty”.  2. Wybranie interesującej kategorii raportów.  3. Kliknięcie w interesujący raport.  4. Raport został wyświetlony.  5. Kliknięcie ikony drukarki w prawym górnym rogu ekranu.  6. Raport został wydrukowany. |

Id: UC20

|  |  |
| --- | --- |
| Nazwa | Podanie dogodnych terminów pracy |
| Aktorzy | Pracownik sekretariatu/ kierowca |
| Opis | Pracownicy mają możliwość podania własnych dogodnych terminów pracy, aby osoby nadrzędne mogły to uwzględnić przy tworzeniu nowego grafiku. |
| Warunki wstępne | Brak |
| Warunki końcowe | Brak |
| Główny przepływ zdarzeń | 1. Kliknięcie przycisku „Dodaj/ zmień dogodny termin pracy”  2. Wybranie terminu z kalendarza.  3. Zatwierdzenie terminu po określeniu, czy to ma być termin dostępny czy niedostępny. |
| Alternatywny przepływ zdarzeń | Brak |

Id: UC21

|  |  |
| --- | --- |
| Nazwa | Dostęp do grafiku |
| Aktorzy | Właściciel/ pracownik sekretariatu/ kierowca |
| Opis | Każdy pracownik ma możliwość sprawdzić grafik każdego pracownika, aby móc się dogadywać w sytuacjach kryzysowych. |
| Warunki wstępne | Brak |
| Warunki końcowe | Brak |
| Główny przepływ zdarzeń | 1. Kliknięcie przycisku „ Wyświetl grafik”  2. Wybranie przycisku „Wyświetl mój grafik”  3. Grafik osobisty został wyświetlony na ekranie. |
| Alternatywny przepływ zdarzeń | 1. Kliknięcie przycisku „ Wyświetl grafik”  2. Wybranie przycisku „Wyświetl grafik innego pracownika”  3. Kliknięcie zdjęcia wybranego pracownika.  4. Grafik innego pracownika został wyświetlony na ekranie. |

Id: UC22

|  |  |
| --- | --- |
| Nazwa | Informacje o pojazdach |
| Aktorzy | Kierowca/ Właściciel |
| Opis | Osoba uprawniona może zobaczyć statusy danych pojazdów oraz gdzie się znajdują |
| Warunki wstępne | Brak |
| Warunki końcowe | Brak |
| Główny przepływ zdarzeń | 1. Kliknięcie przycisku „Wyświetl informacje o pojazdach”  2. Na stronie wyświetlono wszystkie pojazdy(ich miniaturki, status, stan ) |
| Alternatywny przepływ zdarzeń | Brak |

Id: UC23

|  |  |
| --- | --- |
| Nazwa | Dostęp do listy pasażerów |
| Aktorzy | Kierowca/ właściciel |
| Opis | Kierowca posiada indywidualną listę osób, które będzie przewoził na danej trasie. |
| Warunki wstępne | Brak |
| Warunki końcowe | Brak |
| Główny przepływ zdarzeń | 1. Kliknięcie przycisku „Wyświetl listę pasażerów”  2. Kliknięcie: „Pokaż listę pasażerów na najbliższy przejazd”.  3. Lista pasażerów została wyświetlona na ekranie. |
| Alternatywny przepływ zdarzeń | Brak |

Id: UC24

|  |  |
| --- | --- |
| Nazwa | Zapis kosztów tankowania |
| Aktorzy | Kierowca/ właściciel |
| Opis | Kierowca może prowadzić dziennik kiedy tankował, ile litrów paliwa i za ile złotych. Pomaga to kontrolować koszty w firmie. |
| Warunki wstępne | Posiadanie wiedzy o ilości ostatnio zatankowanego paliwa, daty tankowania i ceny paliwa. |
| Warunki końcowe | Brak |
| Główny przepływ zdarzeń | 1. Kliknięcie przycisku „Zapisy kosztów tankowania”  2. Kliknięcie przycisku „Dodaj nowy zapis”  2. Wybranie daty tankowania, wpisanie ilości zatankowanego paliwa oraz ceny za litr paliwa na stacji.  3. Kliknięcie przycisku „Zatwierdź”.  4. Nowy zapis został dodany. |
| Alternatywny przepływ zdarzeń | Brak. |

Id: UC25

|  |  |
| --- | --- |
| Nazwa | Zapis liczby kilometrów po zatankowaniu |
| Aktorzy | Kierowca/ właściciel |
| Opis | Kierowca może dodać do ostatniego zapisu kosztów tankowania informację o ilości przejechanych kilometrów do następnego tankowania. |
| Warunki wstępne | Spisanie informacji o przejechanych kilometrach z licznika. |
| Warunki końcowe | Brak |
| Główny przepływ zdarzeń | 1.Kliknięcie przycisku „Zapisy kosztów tankowania” (Pkt. 1 UC24)  2. Kliknięcie przycisku „Dodaj inf. o ilości km. do ost. tankowania”  3. Należy wpisać liczbę kilometrów przejechaną od ostatniego tankowania.  4. Należy kliknąć „Zatwierdź”  5. Informacja została dodana |
| Alternatywny przepływ zdarzeń | Brak |

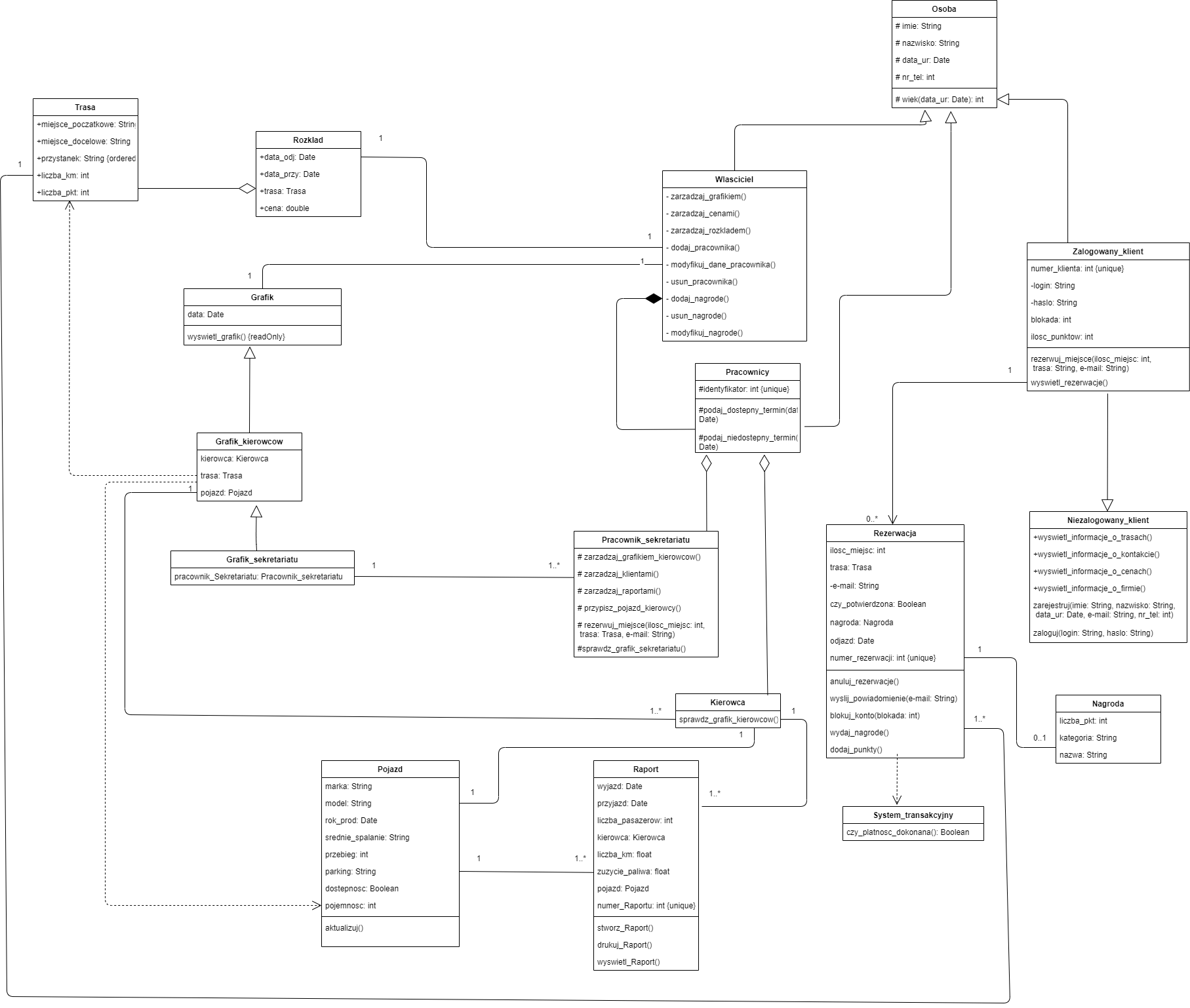
Id: UC26

|  |  |
| --- | --- |
| Nazwa | Możliwość wydruku |
| Aktorzy | Kierowca/ właściciel |
| Opis | Kierowca ma możliwość wydruku indywidualnej listy osób, które będzie przewoził na danej trasie. |
| Warunki wstępne | Wykonanie UC23 |
| Warunki końcowe | Posiadanie podpiętej drukarki |
| Główny przepływ zdarzeń | 1. Wykonanie UC23  2. Kliknięcie ikonki drukarki w prawym górnym rogu ekranu.  3. Wysłanie pliku do drukarki w celu wydruku.  4. Dokument został wydrukowany. |
| Alternatywny przepływ zdarzeń | 4. Nie udało się wydrukować pliku. |

Id: UC27

|  |  |
| --- | --- |
| Nazwa | Podanie realnej liczby pasażerów |
| Aktorzy | Kierowca/ Właściciel |
| Opis | Kierowca ma możliwość podania liczby pasażerów, która realnie odbyła kurs na danej trasie. Są to tylko dane statystyczne. Dostęp do tych danych mają inni pracownicy, którzy mogą wyświetlać raporty. |
| Warunki wstępne | Wykonanie UC23 |
| Warunki końcowe | Brak |
| Główny przepływ zdarzeń | 1. Wykonanie UC23  2. Kliknięcie przycisku „Dodaj liczbę pasażerów, która odbyła kurs na tej trasie”.  3. Wpisanie liczby pasażerów, którzy realnie byli obecni na trasie.  4. Kliknięcie przycisku „Zatwierdź”.  5. Liczba pasażerów została dodana do statystyk. |
| Alternatywny przepływ zdarzeń | Brak |

# Diagram klas



|  |  |
| --- | --- |
| **Nazwa klasy** | **Pracownik\_sekretariatu** |
| **Atrybuty** | Brak |
| **Nazwa metody** | **zarzadzaj\_grafikiem\_kierowcow()** - pracownik sekretariatu ma możliwość edytowania grafiku kierowców.  **zarzadzaj\_klientami()** - pracownik ma możliwość tworzenia i usuwania kont użytkowników.  **zarzadzaj\_raportami()** - pracownik ma możliwość sprawdzania raportów stworzonych przez kierowców i przetwarzania danych.  **przypisz\_pojazd\_kierowcy()** - sekretariat może przypisać dany pojazd dla konkretnego kierowcy na odpowiedni kurs.  **rezerwuj\_miejsce(ilosc\_miejsc: int, trasa: Trasa, e-mail: String)** – sekretariat może zarezerwować przejazd dla osoby, która poda potrzebne dane(ilość miejsc, trasę i e-mail). Na e-mail przyjdzie numer biletu, który należy koniecznie okazać kierowcy przy wsiadaniu do pojazdu.  **sprawdz\_grafik\_sekretariatu()** - pracownik ma możliwość przeglądania swojego grafiku. |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Pracownicy** |
| **Atrybuty** | **identyfikator: int {unique}** – Unikatowy numer określający danego pracownika |
| **Nazwa metody** | **podaj\_dostepny\_termin(data: Date)** - pracownik może zaproponować termin, kiedy mógłby pracować.  **podaj\_niedostepny\_termin(data: Date)** - pracownik może podać termin, kiedy nie będzie miał możliwości przyjścia do pracy. |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Wlasciciel** |
| **Atrybuty** | brak |
| **Nazwa metody** | **zarzadzaj\_grafikiem()** - właściciel może zarządzać czasem pracy wszystkich pracowników firmy.  **zarzadzaj\_cenami()** - właściciel może zmienić ceny przejazdów.  **zarzadzaj\_rozkladem()** - właściciel może zmienić rozkład zawierający dane trasy i przystanki.  **dodaj\_pracownika()** - właściciel może dodać nowego pracownika, który musi posiadać konto klienta w systemie.  **modyfikuj\_dane\_pracownika()** – właściciel może modyfikować dane zatrudnionego pracownika.  **usun\_pracownika()** - właściciel może zwolnić pracownika z firmy.  **dodaj\_nagrode()** - właściciel może dodać nową nagrodę do systemu.  **usun\_nagrode()** - właściciel może usunąć nagrodę z systemu.  **modyfikuj\_nagrode()** - właściciel może zmodyfikować warunki obecnej nagrody. |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Osoba** |
| **Atrybuty** | **imie: String** – określa imię danej osoby.  **nazwisko: String** – określa nazwisko danej osoby.  **data\_ur: Date** – określa datę urodzin danej osoby.  **nr\_tel: int** – określa numer telefonu danej osoby. |
| **Nazwa metody** | **wiek(data\_ur: Date): int** – funkcja oblicza wiek danej osoby na podstawie daty urodzin. |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Kierowca** |
| **Atrybuty** | Brak |
| **Nazwa metody** | **sprawdz\_grafik\_kierowcow() -** kierowca może sprawdzić grafik wszystkich kierowców z przypisanymi pojazdami i trasami. |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Rezerwacja** |
| **Atrybuty** | **ilosc\_miejsc: int** - ilość miejsc przypisanych do rezerwacji.  **trasa\_godz: Rozklad** - trasa oraz godzina, na którą została zrobiona rezerwacja.  **e-mail: String** – adres e-mail przypisany do rezerwacji.  **czy\_potwierdzona: Boolean** – sprawdzenie czy rezerwacja została potwierdzona przez system transakcyjny.  **nagroda: Nagroda** – opcjonalna nagroda wybrana za punkty lojalnościowe przy tworzeniu rezerwacji.  **odjazd: Date** – data oraz godzina planowego odjazdu autobusu, na którą dokonano rezerwacji.  **numer\_rezerwacji: int {unique} –** unikatowy numer, który jednoznacznie identyfikuje rezerwację i umożliwia przejazd. |
| **Nazwa metody** | **anuluj\_rezerwacje()** - możliwość anulowania rezerwacji.  **wyslij\_powiadomienie(e-mail: String)** - powiadomienie wysyła się automatycznie, aby przypomnieć o podróży osobie, która zarezerwowała przejazd.  **blokuj\_konto(blokada: int)** - blokada nakłada się automatycznie na miesiąc, po trzech nieopłaconych rezerwacjach.  **wydaj\_nagrode()** – wybrana nagroda zostaje wysłana na adres klienta, jeśli nagroda zostanie  **dodaj\_punkty()** – jeśli transakcja przebiegnie pomyślnie, punkty za rezerwację zostają przypisane do konta klienta. |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Zalogowany\_klient** |
| **Atrybuty** | **numer\_klienta: int {unique}** - każdy zalogowany klient posiada swój unikalny identyfikator, który zabezpiecza przed pomyłką z innym klientem.  **login: String** - klient posiada swój własny login, który jest potrzebny do zalogowania się na konto.  **haslo: String** - klient posiada swoje własne konto, które jest potrzebne do zalogowania się na konto.  **blokada: int** - jest to licznik niezapłaconych rezerwacji w ciągu ostatniego miesiąca.  **ilosc\_punktow: int** - liczba punktów lojalnościowych na koncie klienta. |
| **Nazwa metody** | **rezerwuj\_miejsce(ilosc\_miejsc: int,**  **trasa: String, e-mail: String)** - zalogowany klient może zarezerwować przejazd dla siebie lub również dla osób towarzyszących.  **wyswietl\_rezerwacje()** - wyświetla historię rezerwacji dokonanych przez użytkownika. |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Niezalogowany\_klient** |
| **Atrybuty** | brak |
| **Nazwa metody** | **wyswietl\_informacje\_o\_trasach()** - każda osoba może wyświetlić informacje o obecnych trasach, na których odbywają się przejazdy.  **wyswietl\_informacje\_o\_kontakcie()** - każda osoba może wyświetlić informacje o możliwościach kontaktu z firmą.  **wyswietl\_informacje\_o\_cenach()** - każda osoba może wyświetlić informacje o cenach biletów obowiązujących na danych trasach.  **zarejestruj(imie: String, nazwisko: String, data\_ur: Date, e-mail: String, nr\_tel: int)** – niezalogowany klient może założyć konto. W calu założenia konta należy podać informacje dotyczące imienia, nazwiska, daty urodzin, e-maila oraz numeru telefonu użytkownika.  **zaloguj(login: String, haslo: String)** - zalogowanie polega na podanie loginu oraz hasła. Po zalogowaniu klient ma dostęp do narzędzi rezerwacji. |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Nagroda** |
| **Atrybuty** | **liczba\_pkt: int** - koszt nagrody wyrażony w liczbie punktów lojalnościowych.  **kategoria: String** - określa, jakiej kategorii dotyczy nagroda.  **nazwa: String** - określa nazwę nagrody. |
| **Nazwa metody** | Brak |

|  |  |
| --- | --- |
| **Nazwa klasy** | **System\_transakcyjny** |
| **Atrybuty** | brak |
| **Nazwa metody** | **czy\_platnosc\_dokonana(): Boolean** - system sprawdza, czy pieniądze zostały pomyślnie przelane na konto firmy. |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Raport** |
| **Atrybuty** | **wyjazd: Date** - rzeczywista data i godzina odjazdu autobusu.  **przyjazd: Date** - rzeczywista data i godzina przyjazdu autobusu.  **liczba\_pasazerow: int** - liczba pasażerów odbywających podróż w danym kursie.  **kierowca: Kierowca** - dane kierowcy wykonującego kurs.  **liczba\_km: float** – liczba kilometrów pokonanych przez autobus po przebyciu trasy.  **zuzycie\_paliwa: float** - ilość paliwa, które zostało zużyte na wykonanie kursu.  **pojazd: Pojazd** - autobus, który obsługiwał dany kurs.  **numer\_Raportu: int {unique}** - unikalny numer, określający raport. |
| **Nazwa metody** | **stworz\_Raport()** - umożliwia uprawnionej osobie stworzyć raport.  **drukuj\_Raport()** - umożliwia wydrukowanie raportu.  **wyswietl\_Raport()** - metoda umożliwia wyświetlenie raportów oraz zawartych w nich danych. |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Pojazd** |
| **Atrybuty** | **marka: String** – marka pojazdu.  **model: String** – model pojazdu.  **rok\_prod: Date** - rok produkcji pojazdu.  **srednie\_spalanie: String** – średnie spalanie pojazdu.  **przebieg: int** – aktualny przebieg pojazdu.  **parking: String** - nazwa parkingu, na którym aktualnie znajduje się pojazd.  **dostepnosc: Boolean** – określa czy pojazd aktualnie stoi na parkingu, czy jest w trasie.  **pojemnosc: int** - określa liczbę pasażerów, którą maksymalnie dopuszczalnie może przewozić pojazd. |
| **Nazwa metody** | **aktualizuj() -** funkcja, która sprawdza położenie pojazdu i aktualizuje dane dotyczące pojazdu(parking, dostępność). |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Grafik** |
| **Atrybuty** | **data: Date –** data i czas określające godziny pracy dla pracowników. |
| **Nazwa metody** | **wyswietl\_grafik() {readOnly}** - metoda pozwalająca na wyświetlenie obecnego grafiku pracowników. Grafik można tylko wyświetlić. Prawo do edycji całego grafiku ma tylko właściciel. |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Grafik\_kierowcow** |
| **Atrybuty** | **kierowca: Kierowca** - kierowca przypisany do konkretnej daty, trasy i pojazdu w grafiku.  **trasa: Trasa** - trasa przypisana do konkretnego kierowcy, daty, pojazdu.  **Pojazd: Pojazd** - określa pojazd, w którym dany kierowca przejedzie określoną trasę według grafiku. |
| **Nazwa metody** | Brak |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Grafik\_sekretariatu** |
| **Atrybuty** | **pracownik\_Sekretariatu: Pracownik\_sekretariatu** - pracownik przypisany do określonego terminu pracy. |
| **Nazwa metody** | Brak |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Trasa** |
| **Atrybuty** | **miejsce\_poczatkowe: String** - nazwa miejsca, z którego odjeżdża autobus.  **miejsce\_docelowe: String** - nazwa miejsca, docelowego danej trasy.  **przystanek: String {ordered}** - przystanki na danej trasie, odpowiednio uporządkowane.  **liczba\_km: int** - odległość (w kilometrach) od miejsca początkowego do miejsca docelowego.  **liczba\_pkt: int** - liczba punktów, która zostanie przypisana do konta po udanej rezerwacji. |
| **Nazwa metody** | Brak |

|  |  |
| --- | --- |
| **Nazwa klasy** | **Rozklad** |
| **Atrybuty** | **data\_odj: Date** – data i godzina odjazdu autobusu z miejsca początkowego  **data\_przy: Date** – data i godzina przyjazdu autobusu do miejsca docelowego..  **trasa: Trasa** – trasa, na której będzie odbywał się przejazd.  **cena: Cena** – ceny biletów na danej trasie. |
| **Nazwa metody** | Brak |

# Diagramy aktywności

Diagram aktywności z rejestracją konta nowego użytkownika

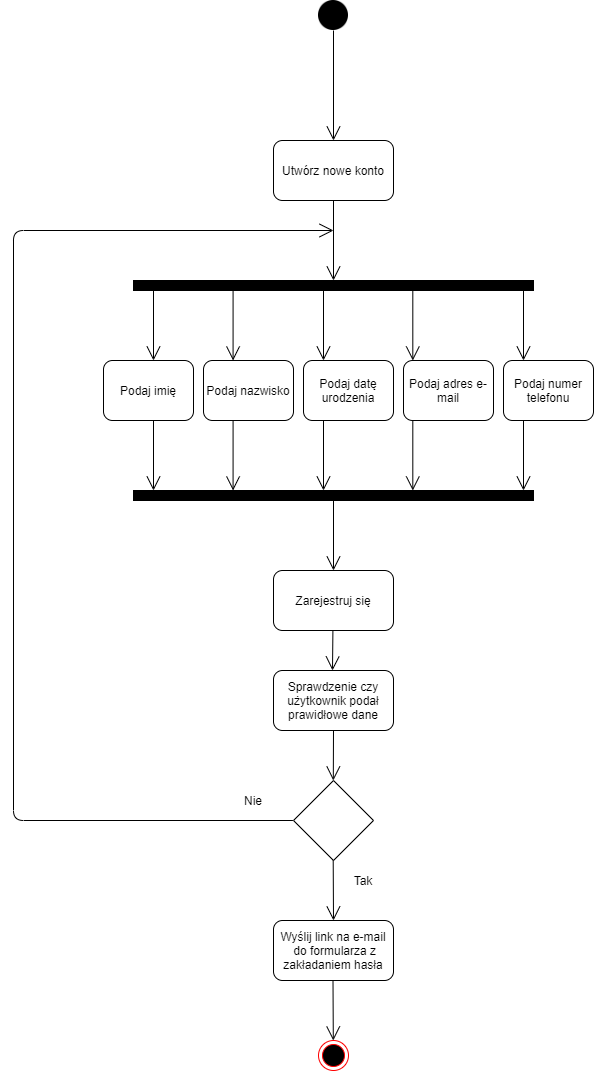


Diagram aktywności z rezerwacją miejsca przez zalogowanego klienta

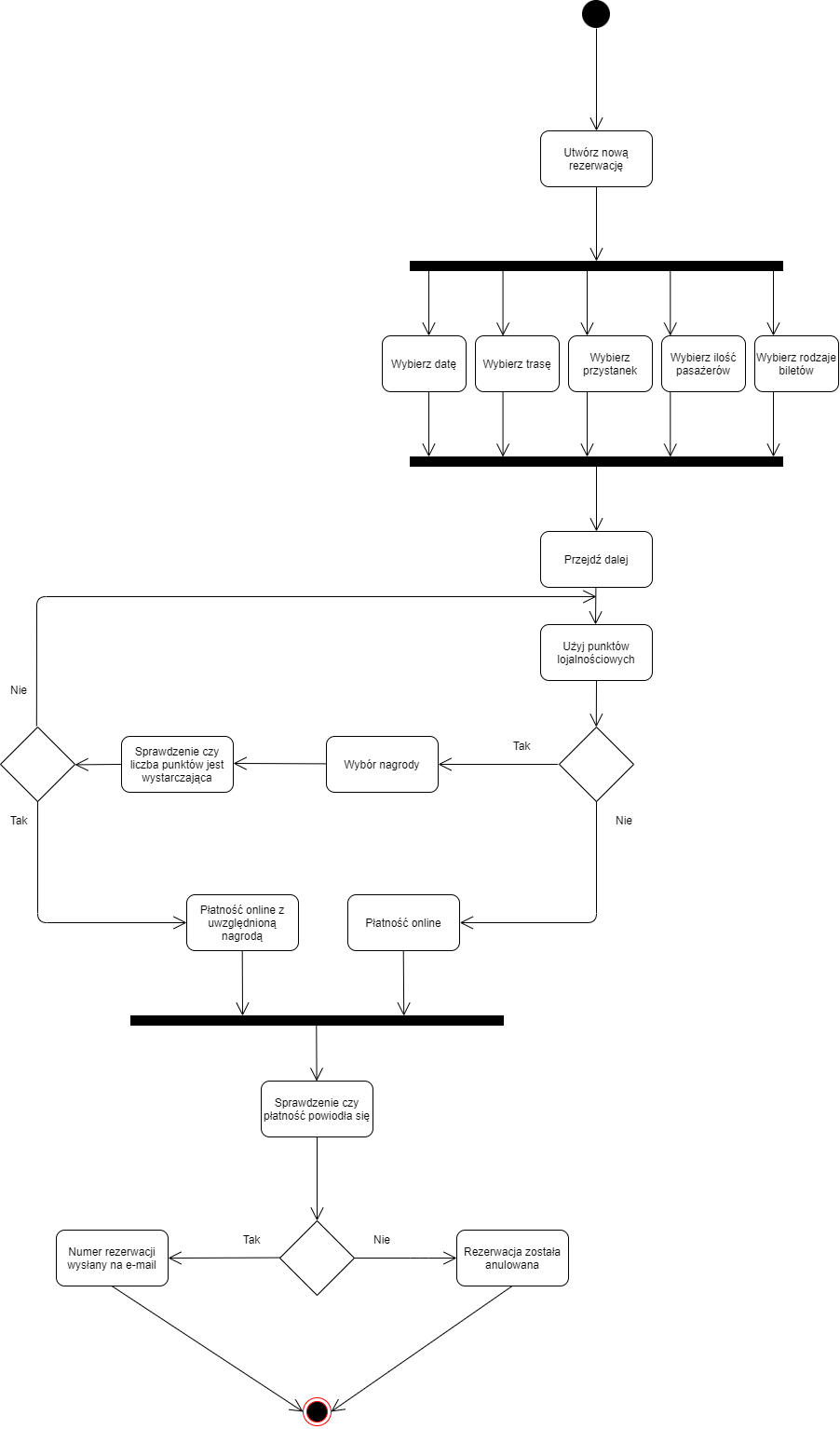


Diagram aktywności z tworzeniem raportu przez kierowcę

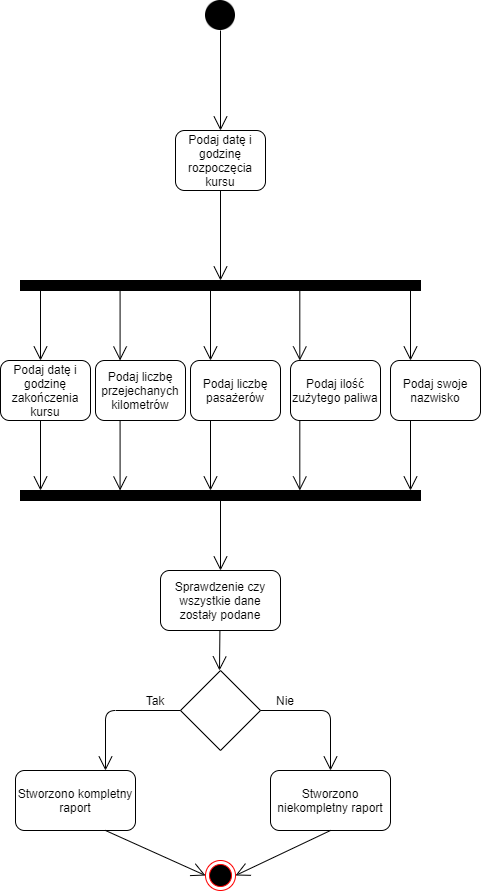
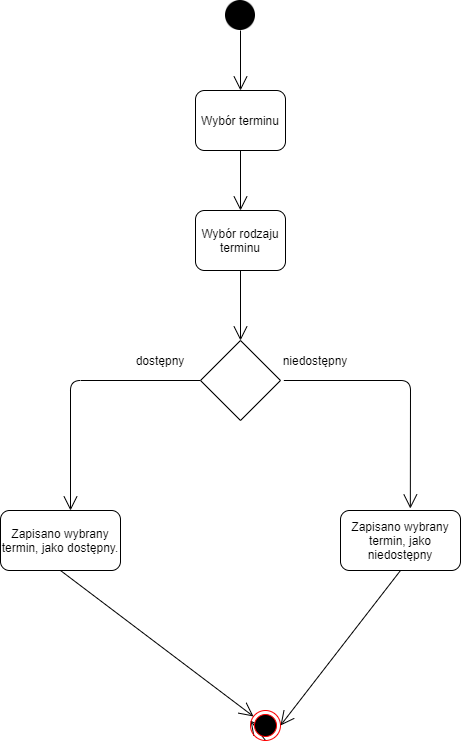


Diagram aktywności z wyborem dogodnego terminu pracy przez pracownika



# Diagramy stanów

Diagram stanu dla pojazdu

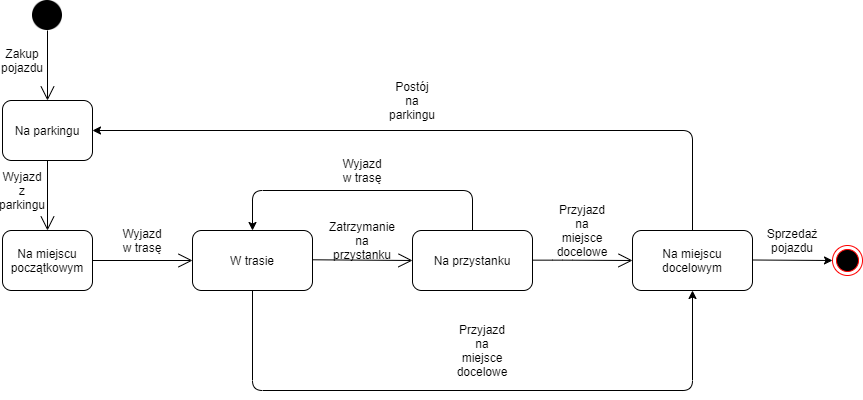


Diagram stanu dla raportu

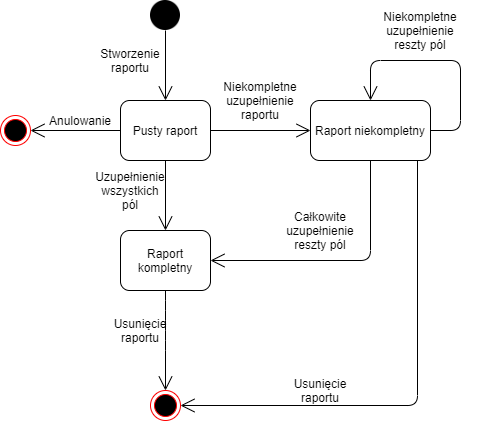


Diagram stanu dla rezerwacji

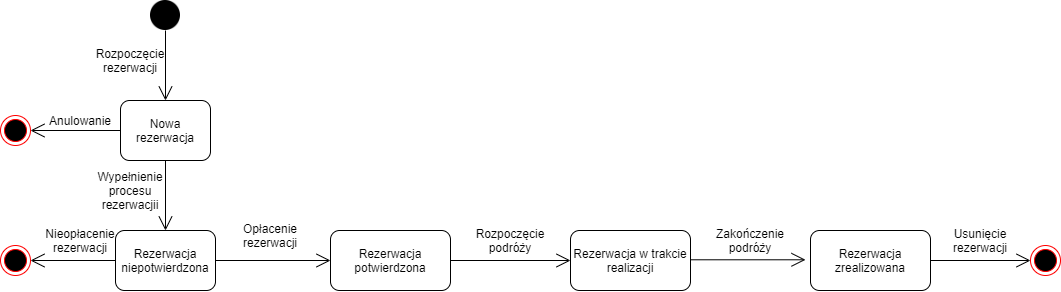
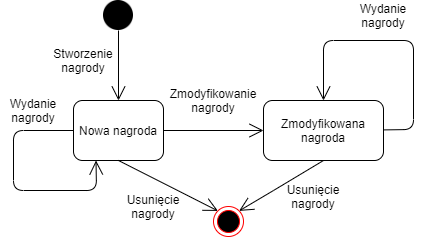
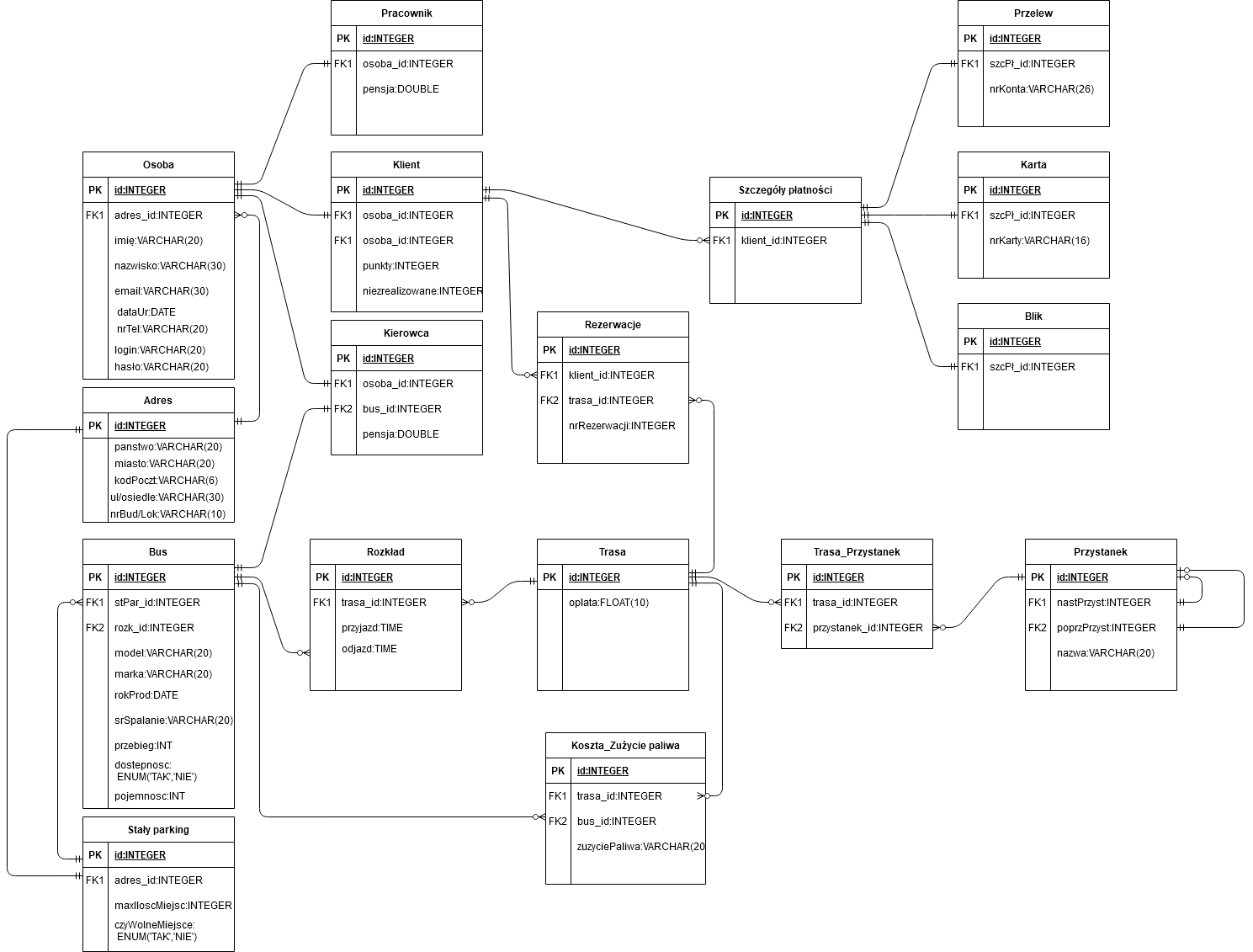


Diagram stanu dla nagrody  


# Projekt bazy danych



Podstawową encją jest “Osoba”, z którą encje “klient”, “pracownik”,”kierowca” powiązane są przez klucz obcy, reprezentując w ten sposób różne rodzaje kont. Zabieg ten jest utrwaleniem mechanizmu dziedziczenia w bazie danych, gdyż w warstwie biznesowej “osoba” jest klasą abstrakcyjną po której dziedziczą pozostałe trzy. “Klient” ma atrybut “punkty” utrwalający liczbę punktów zdobytą przez klienta, oraz “niezrealizowane” oznaczający liczbę niezrealizowanych rezerwacji, który po naliczeniu trzech oraz po okresie blokady, resetuje się. “Klient” może posiadać wiele obiektów typu “Szczegóły płatności”, każdy powiązany z jednym z kilku, które reprezentują różne strategie rozliczeń, co pozwala na dodawanie w przyszłości kolejnych opcji zapłaty. “Klient”, tak samo jak “trasa”, może posiadać wiele “rezerwacji”. Do encji “trasa” oraz “przystanek” może odwoływać się wiele encji “trasa\_przystanek”. “Przystanek” posiada co najmniej jeden z dwóch związków unarnych z innym wierszem tej samej tabelki, tj nastepny przystanek, oraz poprzedni przystanek. “Przystanek” posiada też nazwę. “Rozkład”, posiadający atrybuty “przyjazd” oraz “odjazd” może odwoływać się do “trasy”. “Bus” może odwoływać się do wielu “rozkładów”, oraz posiada wiele atrybutów charakteryzujących danego busa, np “mode”, “srspalanie”, “pojemnosc” itd, oraz może być dostępny, lub nie. Każdy “bus” posiada miejsce na stałym parkingu. “Staly parking” odwołuje się do specjalnie wydzielonej w tym celu encji “adres”, do której mogą odwoływać się też wszystkie inne encje potrzebujące utrwalić informacje o lokalizacji, np “osoba”. “Koszta\_Zużycie paliwa” odwołuje się “bus” oraz “trasa” i utrwala, obliczone przez aplikację, średnie zużycie paliwa na kilometr dla wybranego pojazdu na wybranej trasie oraz potencjalne wszelkie inne atrybuty związane z opłacalnością. “Pracownik” oraz “kierowca” mają pole “pensja”. “Kierowca” jest powiązany z “busem”.