

Dokumentacja do projektu końcowego z przedmiotu MAS

Autor: Dominik Suszek s23396

Wypożyczalnia pojazdów „Mobile4U”

Spis treści

1.	<i>Wymagania użytkownika</i>	3
2.	<i>Diagram przypadków użycia</i>	7
3.	<i>Diagram klas – analityczny</i>	8
4.	<i>Diagram klas – projektowy</i>	9
5.	<i>Scenariusz przypadku użycia</i>	10
6.	<i>Diagram aktywności dla przypadku użycia</i>	11
7.	<i>Diagram stanu dla klasy Pojazd</i>	12
8.	<i>Projekt GUI</i>	13
9.	<i>Omówienie decyzji projektowych i skutków analizy dynamicznej</i>	16

1. Wymagania użytkownika

1.1. Wymagania opisowe

- 1) W systemie są przechowywane dane osobowe różnych osób - pracowników i klientów. Klient nie może być jednocześnie pracownikiem. Jeśli któryś z pracowników chce korzystać z usług firmy, to musi założyć osobne konto jako klient (będzie wtedy posiadał dwa osobne konta). Każda z osób jest opisywana przez:
 - unikalny identyfikator,
 - imię,
 - nazwisko,
 - datę urodzenia.
- 2) Dodatkowo dla każdego klienta chcemy przechowywać informacje o dacie wydania ostatniego prawa jazdy (prawo jazdy wygasa po pewnym czasie, więc starsi kierowcy mogą mieć kilka praw jazdy w ciągu życia) oraz o przyszłej dacie wygaśnięcia tego prawa jazdy. Każdy kierowca musi być w stanie przedstawić prawo jazdy w trakcie podpisywania dokumentów przed rozpoczęciem wynajmu. Prawo jazdy musi być ważne przynajmniej do dnia zakończenia wynajmu włącznie. System nie musi obejmować automatycznej weryfikacji poprawności prawa jazdy. Odpowiedzialny za to jest pracownik wypożyczalni.
- 3) Klientowi może zostać przyznany status „Stałego Klienta”, który wiąże się z uzyskaniem dodatkowych zniżek. Taki status przyznaje pracownik firmy wedle własnego uznania (warunkiem jest minimum 20 wynajmów na koncie danego użytkownika).
- 4) Wśród pracowników wyróżniamy dwie główne grupy, tj. pracowników fizycznych oraz pracowników biurowych. Wśród pracowników fizycznych wyróżniamy mechaników. Wśród osób pracujących w biurze należy wyróżnić agentów sprzedaży oraz specjalistów od marketingu. Należy pamiętać o tym, że zatrudnieni są również pracownicy łączący cechy obu grup, tj. managerowie warsztatów samochodowych należących do firmy.
- 5) W przypadku pracowników fizycznych (obecnie są to jedynie mechanicy, ale może zmienić się to w przyszłości), system powinien umożliwiać przechowywanie informacji o ich zdolnościach manualnych (w których się specjalizują) oraz o dacie ważności ostatnio wykonanego badania BHP.
- 6) Dla pracowników biurowych należy przechowywać informacje o ich umiejętnościach najważniejszych z punktu widzenia wykonywanej pracy (zdobyte wykształcenie, uzyskane certyfikaty, znajomość języków obcych etc.) oraz o dacie ważności badania wzroku.
- 7) Dla specjalistów od marketingu należy przechowywać dwie dodatkowe informacje tekstowe:
 - opis portfolio,
 - opis poprzednich kampanii marketingowych.

8) System powinien obejmować możliwość rejestrowania sytuacji, w których pracownik zostaje zwolniony (lub zwalnia się na własną prośbę), a następnie ponownie zatrudniony. Pracownik nie może zostać zwolniony tego samego dnia, w którym go zatrudniono. Taką operację można wykonać dopiero następnego dnia. Każdy z pracowników za zgodą managera może dokonać również następujących zmian w zatrudnieniu:

- może zmienić oddział firmy, w którym pracuje,
- może złożyć wniosek o długą przerwę w pracy (trwającą więcej niż 30 dni),

W związku z tym, system musi przechowywać informacje o każdym z tych okresów zatrudnienia. Niezbędne dane to:

- unikalny identyfikator,
- data rozpoczęcia danego okresu zatrudnienia,
- data zakończenia danego okresu zatrudnienia,
- pensja.

9) Dla każdego z oddziałów firmy należy przechowywać następujące informacje:

- unikalny identyfikator,
- nazwę,
- adres.

10) Adres składa się z:

- unikalnego identyfikatora,
- nazwy ulicy,
- numeru ulicy,
- numeru lokalu,
- nazwy miasta,
- kodu pocztowego.

11) Dla każdego z regionów należy przechowywać następujące dane: unikalny identyfikator oraz nazwę.

12) Do jednego regionu może być przypisanych wiele oddziałów firmy, natomiast każdy z oddziałów może być przypisany tylko do jednego regionu. W przypadku likwidacji całego regionu należy usunąć dane dotyczące wszystkich oddziałów firmy, którego przynależały do regionu oraz powiązania między nimi. W takiej sytuacji należy również usunąć wszystkie powiązania pomiędzy likwidowanymi oddziałami firmy, a pojazdami, które do nich należały.

- 13) Dla każdego wynajmu system powinien przechowywać następujące dane:
- unikalny numer identyfikacyjny,
 - datę rozpoczęcia wynajmu,
 - datę zakończenia wynajmu,
 - stałą bazową opłatę za każdy przejechany kilometr, która obecnie wynosi 0.9.
- 14) Do każdego wynajmu jest przypisany jeden pojazd, a także jeden kierowca wynajmujący go.
- 15) Na zakończenie każdego wynajmu pracownik firmy będzie uzupełniał informację o przejechanym dystansie, który będzie podstawą do obliczenia całkowitego kosztu wynajmu.
- 16) Dla lepszej organizacji wynajmów, proces tworzenia nowego wynajmu powinien zaczynać się od wybrania regionu, następnie oddziału firmy, a dopiero w kolejnym kroku wybrania pojazdu (wśród dostępnych w danym oddziale).
- 17) System powinien umożliwiać klasyfikowanie wynajmu z uwzględnieniem dwóch kryteriów:
- czas trwania wynajmu (krótkoterminowy – poniżej 3 dni, oraz długoterminowy – 3 dni i więcej),
 - miejsce odbioru i zwrotu pojazdu (odebranie pojazdu z oddziału firmy, lub odebranie pojazdu z miejsca wybranego przez klienta – ta opcja wiąże się z dodatkową opłatą).
- 18) Wypożyczalnia udostępnia klientom różne pojazdy. W celu sprawnego rozróżniania i zarządzania flotą pojazdów, przechowujemy dla każdego z nich:
- unikalny identyfikator,
 - unikalny numer rejestracyjny,
 - nazwę marki,
 - nazwę modelu.
- 19) Wśród pojazdów możemy wyróżnić samochody osobowe i ciężarówki. Podział jest kompletny i rozłączny.
- 20) Każdy z pojazdów należy tylko do jednego z oddziałów firmy. Do oddziału firmy może być przypisanych wiele pojazdów.
- 21) Dodatkowo, każdy pojazd może być objęty jedną lub wieloma polisami ubezpieczeniowymi. Dana polisa ubezpieczeniowa może być przypisana tylko do jednego pojazdu.

22) Cechami charakterystycznymi samochodu osobowego są:

- pojemność silnika,
- rodzaj (zwykły samochód osobowy, samochód sportowy, samochód elektryczny, samochód premium),
- wysokość zawieszenia (dostępna jedynie dla samochodów sportowych),
- luksusowe elementy wystroju (dostępne jedynie dla samochodów sportowych),
- pojemność baterii (dostępna jedynie dla samochodów elektrycznych).

23) Cechą charakterystyczną ciężarówki jest ładowność (dopuszczalna masa całkowita – DMC).

24) Ponadto system powinien zawierać funkcjonalność, która automatycznie wyznacza ceny wynajmu za każdy przejechany kilometr, która różni się dla pojazdów w zależności od:

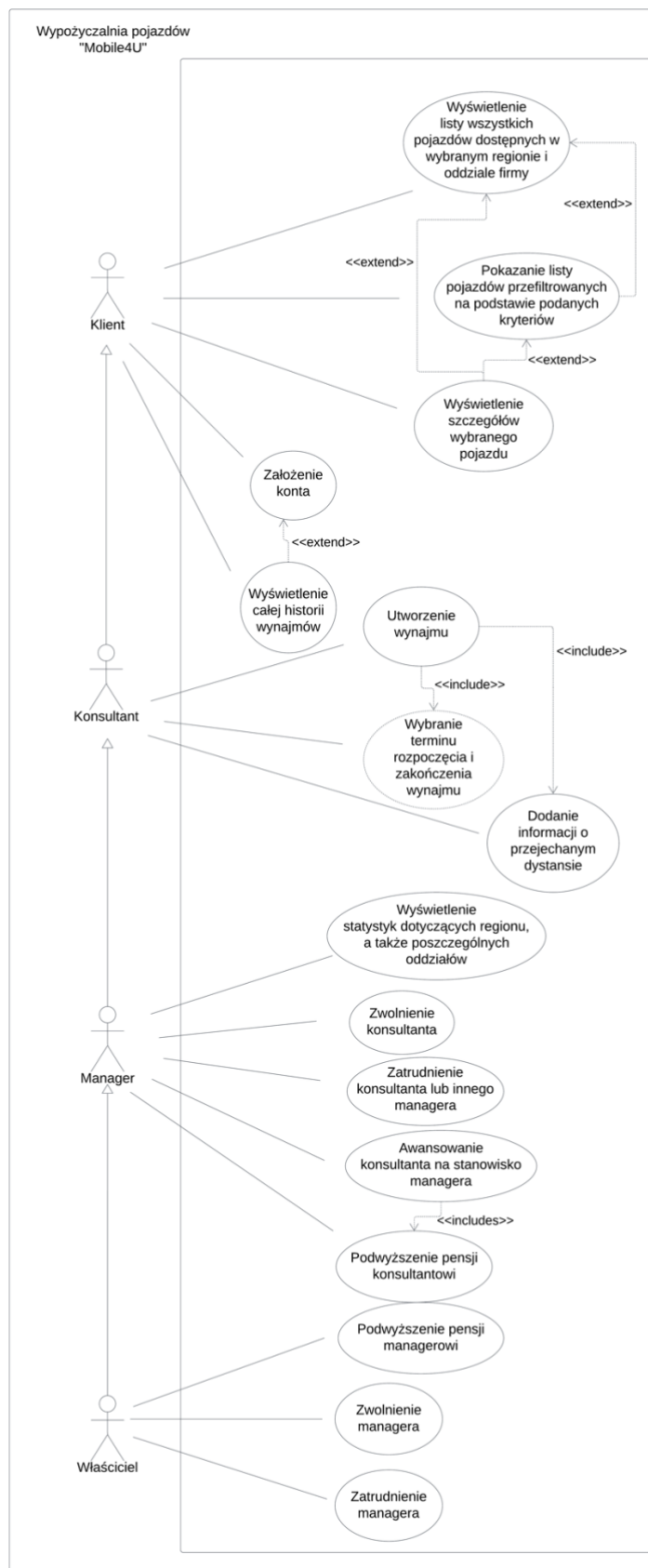
- pojemności silnika oraz rodzaju dla samochodu osobowego,
- ładowności dla ciężarówki.

25) System powinien przechowywać dane o każdej z napraw pojazdów:

- Unikalny identyfikator,
- Datę początkową,
- Datę końcową,
- Koszt,
- Opis dokonanej naprawy.

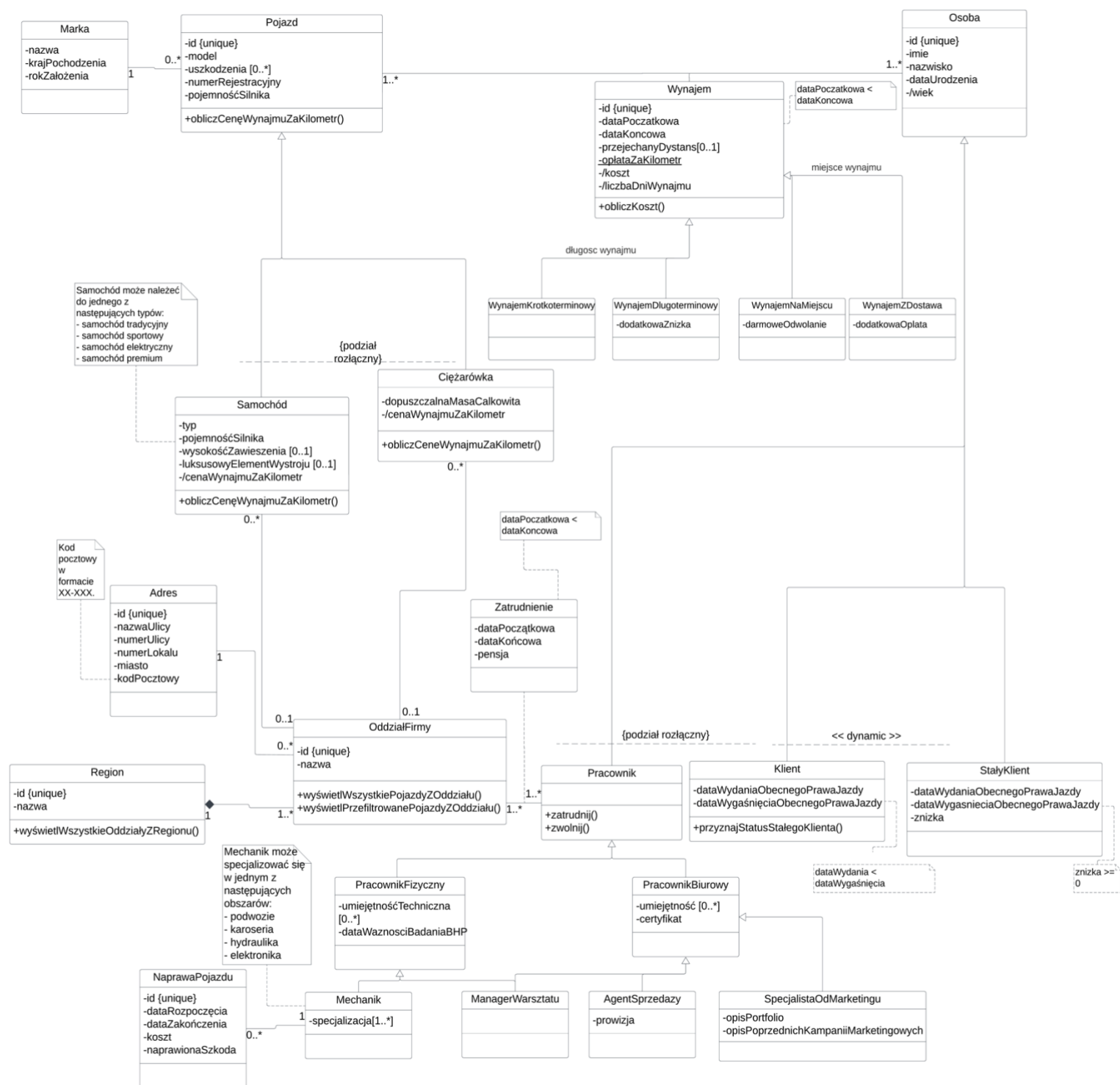
Każdy z pojazdów może być wielokrotnie naprawiany. Naprawa zawsze dotyczy tylko jednego pojazdu. Do każdej z napraw jest przypisany tylko jeden mechanik.

2. Diagram przypadków użycia



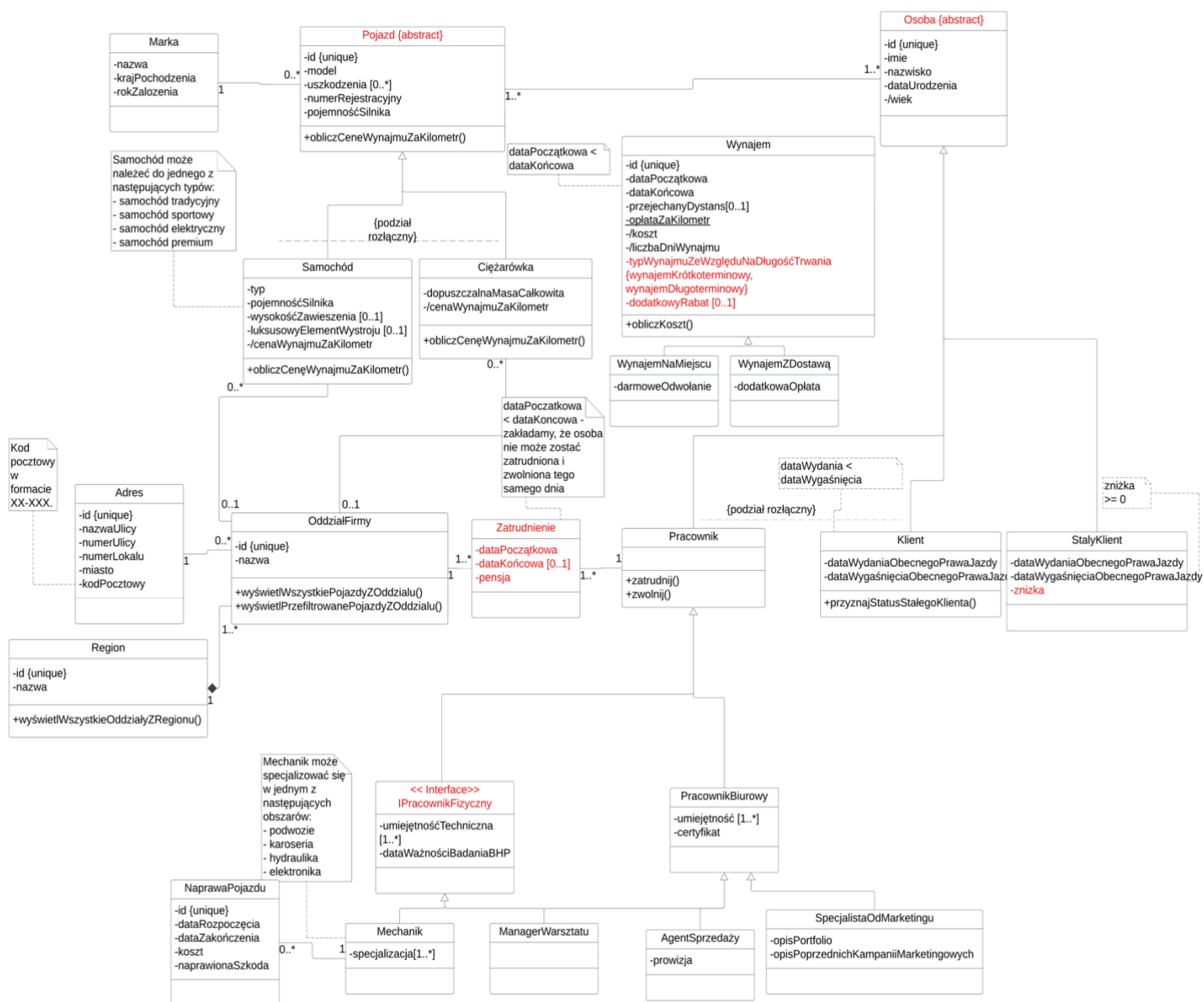
Rys. 1 Diagram przypadków użycia.

3. Diagram klas – analityczny



Rys. 2 Analityczny diagram klas.

4. Diagram klas – projektowy



Rys. 3 Projektowy diagram klas.

Po uwzględnieniu części systemu, których nie da się bezpośrednio zaimplementować w języku Java, a także informacji uzyskanych podczas przeprowadzania analizy dynamicznej zdecydowano się na wprowadzenie zmian zaznaczonych na czerwono na diagramie.

5. Scenariusz przypadku użycia

Przypadek użycia „Utworzenie nowego wynajmu”

Przebieg: wybranie parametrów dotyczących poszukiwanego pojazdu.

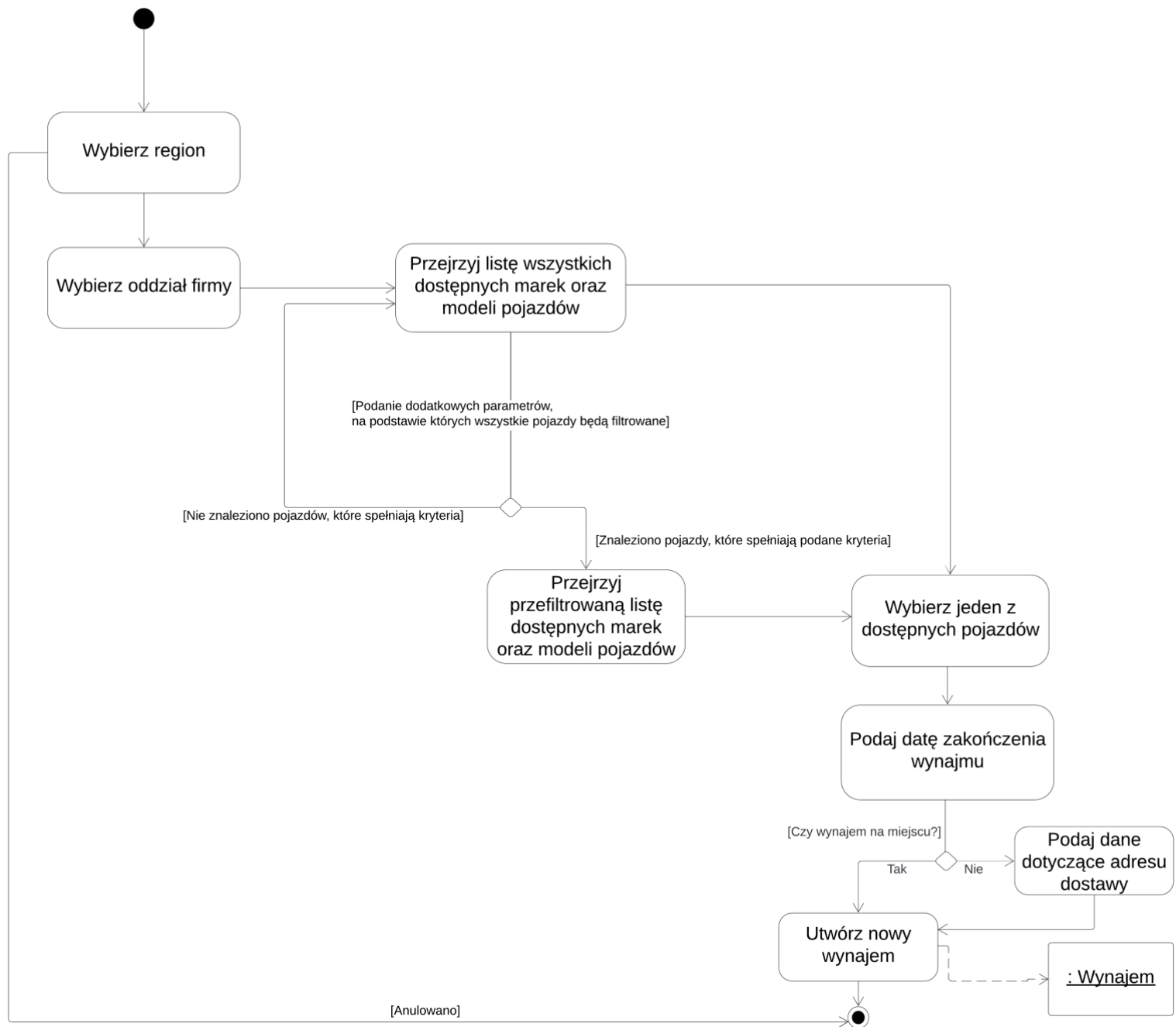
Zakończenie: wyświetlenie potwierdzenia wynajmu.

Wykorzystane dane: region, oddział firmy, marka pojazdu, model pojazdu, czy wynajem ma być na miejscu (jeśli nie, to również adres dostawy pojazdu), data zakończenia wynajmu. Opcjonalnie wybrane parametry (rodzaj pojazdu (samochód osobowy/ciężarówka), wielkość silnika).

1. System odpytuje o dane dotyczące lokalizacji: region oraz oddział firmy.
2. Użytkownik wprowadza dane.
3. System odpytuje o markę oraz model pojazdu, który interesuje użytkownika.
4. Użytkownik wprowadza dane.
5. System wyświetla pojazdy, które są obecnie dostępne i spełniają wymogi.
6. Użytkownik przegląda zwrócone pojazdy i opcjonalnie wybiera opcję filtrowania na podstawie dodatkowych parametrów.
7. (Opcjonalnie) System odpytuje o dodatkowe parametry (rodzaj pojazdu (samochód osobowy/ciężarówka), wielkość silnika).
8. (Opcjonalnie) Użytkownik wprowadza dane.
9. (Opcjonalnie) System zwraca pojazdy, które spełniają dodatkowe wymagania.
10. Użytkownik wybiera interesujący go pojazd.
11. System odpytuje o datę zakończenia wynajmu oraz to, czy wynajem ma być na miejscu, czy z dostawą pojazdu pod wybrany adres.
12. Użytkownik wprowadza dane.
13. (Opcjonalnie) Jeśli użytkownik wybrał opcję wynajmu z dostawą pojazdu pod wskazany adres, system odpytuje o dane dotyczące adresu.
14. (Opcjonalnie) Użytkownik wprowadza dane.
15. System wyświetla potwierdzenie wynajmu i wiadomość podsumowującą.
16. Nowy wynajem zostaje utworzony i zarejestrowany w systemie.

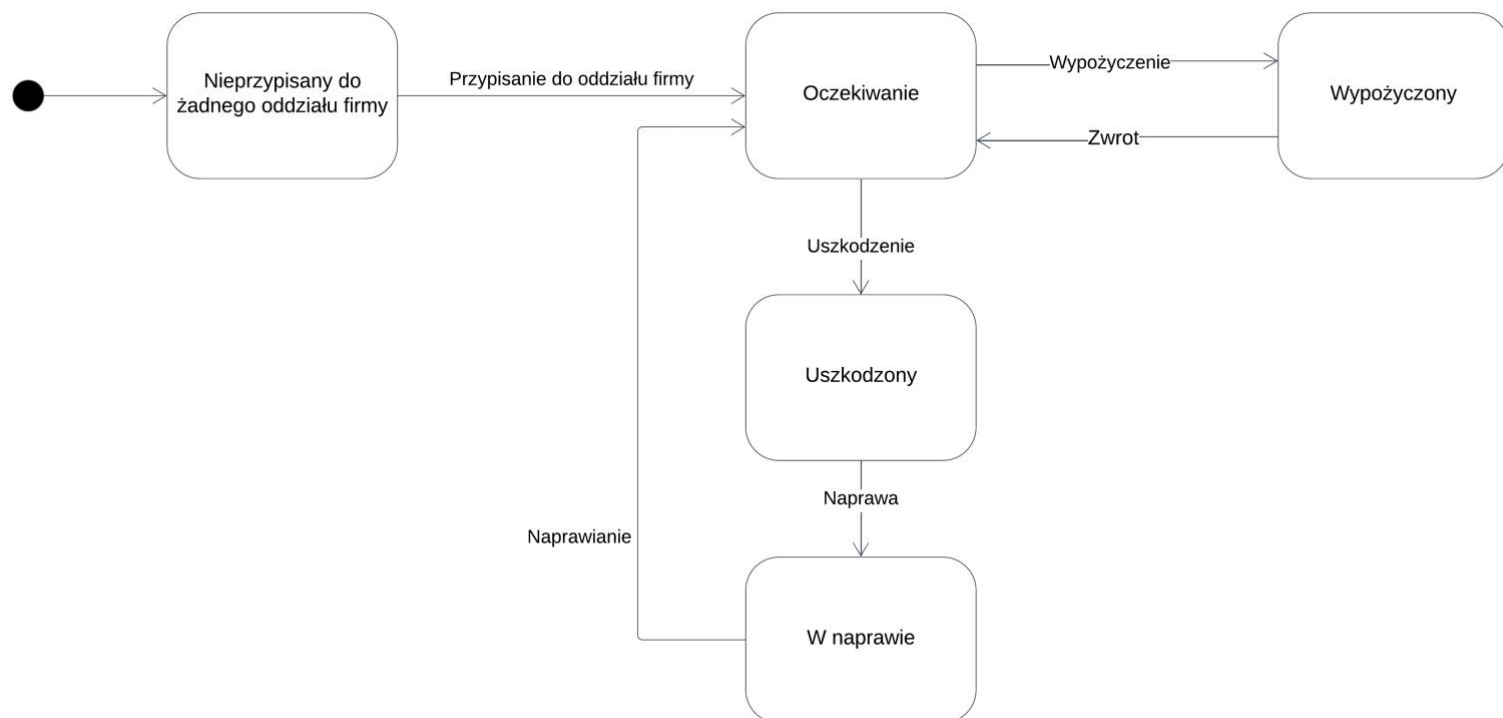
6. Diagram aktywności dla przypadku użycia

Przypadek użycia „Utworzenie wynajmu”



Rys. 4 Diagram aktywności dla przypadku użycia „Utworzenie wynajmu”

7. Diagram stanu dla klasy Pojazd



Rys. 5 Diagram stanu dla klasy Pojazd

8. Projekt GUI

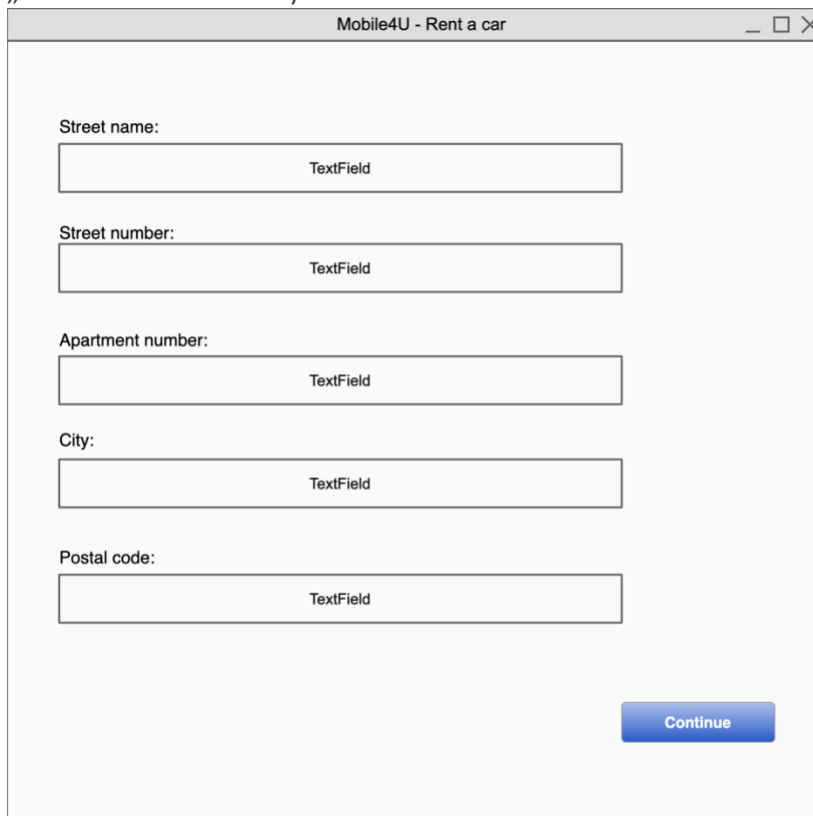
Wybranie regionu oraz oddziału firmy:

The screenshot shows a window titled "Mobile4U - Rent a car". Inside, there are two selection steps. The first step, "Select region:", is followed by a "ComboBox" button. The second step, "Select company branch:", is followed by a large "ListBox" area. At the bottom right, there is a blue "Continue" button.

Wybranie marki, modelu, daty zakończenia wynajmu (i opcjonalnie innych parametrów):

The screenshot shows the same window titled "Mobile4U - Rent a car", but at a later stage. It now includes "Select brand:" and "Select model:" labels, each followed by a "ComboBox" button. A green "Search" button is positioned to the right of the "Select model:" ComboBox. Below these is the label "All available vehicles:" followed by a large "ListBox" area. At the bottom, there are two blue buttons: "Display vehicle details" and "Add filters". Below "Display vehicle details" is the label "Select end date:" followed by a "DatePicker" button. To the right of the "Add filters" button is a checkbox labeled "Rental door to door". A blue "Continue" button is located at the bottom right.

Wybór miejsca odbioru pojazdu (opcjonalnie – jedynie w przypadku, gdy zaznaczono opcję „Rental door to door”):



Mobile4U - Rent a car

Street name:

TextField

Street number:

TextField

Apartment number:

TextField

City:


TextField

Postal code:

TextField

Continue

Podsumowanie – wynajem na miejscu (z oddziału firmy):



Mobile4U - Rent a car

Region:

TextField; TdO

Company branch:

TextField; TdO

Vehicle:

TextField; TdO

End date:

TextField; TdO

Save

Podsumowanie – wynajem z dostawą pojazdu:

Mobile4U - Rent a car

Region:

TextField; TdO

Company branch:

TextField; TdO

Vehicle:

TextField; TdO

End date:

TextField; TdO

Address:

TextField; TdO

Save

9. Omówienie decyzji projektowych i skutków analizy dynamicznej

9.1. Ekstensje klas oraz trwałość ekstensji

Ekstensje klas są zaimplementowane przy użyciu klasy ObjectPlus, po której dziedziczy każda klasa bazowa (Vehicle, Person, etc.) Trwałość zapisu danych będzie zapewniona przez serializację obiektów i zapisywanie danych do pliku „rentalData.dat”.

Klasa ObjectPlus zawiera prywatny atrybut klasowy – mapę, w której zapisywane są klucze (nazwy klas poszczególnych ekstensji), oraz wartości będące kolekcjami z referencjami do obiektów danej klasy.

Zaimplementowano metody służące do dodawania, usuwania elementów z ekstensji, a także do zwracania wszystkich obiektów będących w ekstensji danej klasy.

9.2. Klasy abstrakcyjne

Zdecydowano, że klasy Pojazd oraz Osoba nie powinny mieć bezpośrednich wystąpień. Służą w programie do tworzenia hierarchii dziedziczenia i porządkowania.

9.3. Asocjacja z atrybutem

Asocjację z atrybutem dodano pomiędzy klasami OddziałFirmy oraz Pracownik. Wprowadzono klasę pośredniczącą, którą nazwano Zatrudnienie. Klasa Zatrudnienie będzie zawierała trzy pola: dataPoczątkowa, dataKońcowa oraz pensja. Pola dataPoczątkowa oraz pensja będą znane od razu, natomiast atrybut dataKońcowa będzie opcjonalny (uzupełniany dopiero w momencie zwolnienia pracownika, nie w momencie utworzenia obiektu klasy Zatrudnienie).

9.4. Dziedziczenie dynamiczne

Dziedziczenie dynamiczne pomiędzy klasami Klient oraz StałyKlient zaimplementowano za pomocą „sprytnego” kopiowania obiektów. W tym przypadku w trakcie przechodzenia od obiektu klasy Klient do obiektu klasy StałyKlient nie trzeba dodatkowo usuwać wybranych atrybutów/asocjacji, ponieważ osoba, której zostanie przypisany status stałego klienta ma nadal te same dane i powinna posiadać te same powiązania (wygenerowane w przeszłości) z obiektami innych klas (np. wypożyczenia).

9.5. Kompozycja

W celu zaimplementowania kompozycji pomiędzy klasami `CompanyBranch` oraz `Region` wykorzystano podejście związane z modyfikacją rozwiązania z natywnymi referencjami. Konstruktor klasy `CompanyBranch` jest prywatny, dzięki czemu zablokowane zostało samodzielne tworzenie części. Obiekt klasy będącej częścią (`CompanyBranch`) można połączyć z obiektem będącym całością (`Region`) poprzez wywołanie metody tworzącej części w oparciu o przekazaną całość – `CompanyBranch.createCompanyBranch()`.

To rozwiązanie cechuje się tym, że:

- Część (obiekt klasy `CompanyBranch`) zawsze musi należeć do jakiejś „całości” (obektu klasy `Region`).
- Przy usuwaniu obiektu będącego „całością” zostają usunięte wszystkie połączenia z częściami.
- Jeśli „część” (`CompanyBranch`) ma dalsze powiązania z obiektami pozostałych klas, również zostaną usunięte.
- Samodzielne tworzenie „części” (`CompanyBranch`) jest zablokowane.
- „Części” (`CompanyBranch`) nie mogą być współdzielone przez wiele „całości” (`Region`).

9.6. Dziedziczenie wieloaspektowe

Dziedziczenie wieloaspektowe zaimplementowano w przypadku klasy `Wynajem` oraz podklas `WynajemKrotkoterminowy`, `WynajemDlugoterminowy` (podział wynajmów ze względu na długość trwania), oraz `WynajemNaMiejscu`, `WynajemZDostawa` (podział wynajmów ze względu na miejsce odbioru i zwrotu pojazdu).

Jeden aspekt dziedziczenia (`WynajemNaMiejscu`, `WynajemZDostawa`) zaimplementowano za pomocą wbudowanych (prostych) mechanizmów dziedziczenia. Innymi słowy, klasy `WynajemNaMiejscu`, `WynajemZDostawa` dziedziczą po klasie `Wynajem`.

Drugi aspekt dziedziczenia (długość trwania wynajmu) zaimplementowano za pomocą grupowania w nadklasie. Dla rozróżnienia wynajmów dodano dodatkowe pole typu boolean, które służy jako flaga (dyskryminator).