

# Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: System do zarządzania mandatami parkingu miejskiego

Autorzy: **Valeriia Balanivska i Daria Rudowicz**

Grupa: I1-210A

Kierunek: informatyka

Rok akademicki: 2021

Poziom i semestr: I/4

Tryb studiów: niestacjonarne

<b>1</b>	<b>Spis treści</b>	
<b>2</b>	Odnośniki do innych źródeł	4
<b>3</b>	Słownik pojęć	5
<b>4</b>	Wprowadzenie	6
4.1	Cel dokumentacji	6
4.2	Przeznaczenie dokumentacji	6
4.3	Opis organizacji lub analiza rynku	6
4.4	Analiza SWOT organizacji	6
<b>5</b>	Specyfikacja wymagań	7
5.1	Charakterystyka ogólna	7
5.2	Wymagania funkcjonalne	7
5.3	Wymagania niefunkcjonalne	8
<b>6</b>	Zarządzanie projektem	9
6.1	Zasoby ludzkie	9
6.2	Harmonogram prac	9
6.3	Etapy/kamienie milowe projektu	9
<b>7</b>	Zarządzanie ryzykiem	10
7.1	Lista czynników ryzyka	10
7.2	Ocena ryzyka	10
7.3	Plan reakcji na ryzyko	10
<b>8</b>	Zarządzanie jakością	11
8.1	Scenariusze i przypadki testowe	11
<b>9</b>	Projekt techniczny	12
9.1	Opis architektury systemu	12
9.2	Technologie implementacji systemu	12
9.3	Diagramy UML	12
9.4	Charakterystyka zastosowanych wzorców projektowych	12
9.5	Projekt bazy danych	12
9.6	Projekt interfejsu użytkownika	12
9.7	Procedura wdrożenia	13
<b>10</b>	Dokumentacja dla użytkownika	14
<b>11</b>	Podsumowanie	15
11.1	Szczegółowe nakłady projektowe członków zespołu	15
<b>12</b>	Inne informacje	16

## **2 Odnośniki do innych źródeł**

- Zarządzania projektem – Jira
- Wersjonowanie kodu – Github
- System obsługi defektów – Jira

### 3 Słownik pojęć

Administrator - Osoba obsługująca **system**.

Kierowca pojazdu - Osoba będąca właścicielem pojazdu i korzystająca z usługi sieci **parkingów**.

Kontroler - Osoba zajmująca się wystawianiem **mandatów**.

Raport windykacyjny

Windykator - osoba, której zadaniem jest polubowne doprowadzenie do spłaty zadłużenia.

System - Przedmiot pracy, czyli projektowany system zarządzania parkowaniem miejskim.

## 4 Wprowadzenie

### 4.1 Cel dokumentacji

Opis projektu aplikacji (zakres działania i funkcje) na parkingu miejskim w Szczecinie, służącej do obsługi płatności za parking przez kierowców, usprawnienia procesów związanych z windykacją, inwentaryzacji dokonanych opłat za parkowanie oraz walki z korupcją wśród pracowników organizacji.

### 4.2 Przeznaczenie dokumentacji

Dokumentacja jest tworzona na cele przetargu na system do zarządzania mandatami na parkingu miejskim dla kierowców i pracowników Urzędu.

### 4.3 Opis organizacji lub analiza rynku

Organizacja, dla której realizowany będzie system to Urząd Miasta Szczecin. Organizacja ta zajmuje się zarządzaniem procesami miejskimi w mieście Szczecin. Jednym z tych procesów jest zarządzanie parkingami miejskimi. Dzięki wprowadzeniu systemu, zostanie zoptymalizowany proces naliczania kar za przekroczony czas parkowania, zostanie opanowane zjawisko korupcji oraz ułatwiony zostanie proces windykacyjny. Wszystko to sprawi, że zmniejszy się nakład pracy pracowników do obsługi mandatów.

### 4.4 Analiza SWOT organizacji

Tylko jeśli dla konkretnej organizacji

Wystarczy sama tabela 2x2 (silne-słabe-szanse-zagrożenia)

<ul style="list-style-type: none"><li>• duże zasoby finansowe</li><li>• brak możliwości bankructwa spółki</li></ul>	<ul style="list-style-type: none"><li>• zwiększanie się liczby samochodów, dzięki czemu rośnie się popyt na parkowanie miejskie</li></ul>
<ul style="list-style-type: none"><li>• brak umiejętności obsługi komputera przez pracowników organizacji</li><li>• przestarzałe komputery, uniemożliwiające instalację systemu</li><li>• brak odpowiedniej obsługi informatycznej</li></ul>	<ul style="list-style-type: none"><li>• powstanie prywatnych parkingowców oferujących lepsze warunki</li><li>• częste remonty infrastruktury drogowej powodujące zablokowanie części miejsc parkingowych</li><li>• ekologiczna polityka zachęcająca mieszkańców miasta do poruszania się samochodami</li></ul>

## **5 Specyfikacja wymagań**

### **5.1 Charakterystyka ogólna**

#### **5.1.1 Definicja produktu**

Opłaty za parkowanie i wystawione mandaty na parkingu miejskim w Szczecinie.

#### **5.1.2 Podstawowe założenia**

Wpłaty, inwentaryzacja dokonanych wpłat, przekazywanie danych do windykacji, zapobieganie korupcji.

#### **5.1.3 Cel biznesowy**

Wdrażanie systemu pozwoli skuteczne zarządzanie i korzystanie z systemu parkingu miejskiego.

Odciążenie pracowników w organizacji, długofalowo - oszczędność pieniędzy oraz papieru. Optymalizacja przepływu danych między windykacją, a pracownikami - skuteczne egzekwowanie długu od kierowców pojazdów. Dostosowanie się do polityki antykorupcyjnej. Prezentowanie danych w przejrzysty dla użytkownika sposób.

#### **5.1.4 Użytkownicy**

Kierowca, Administrator, Kontroler.

#### **5.1.5 Korzyści z systemu**

##### **Administrator**

**A-1.1.** Ułatwienie sprawnego zarządzania rozbudowaną siecią parkingów wszelakiego typu.

**A-1.2.** Oszczędność czasu spowodowana elektronicznym generowaniem raportu.

##### **Kontroler**

**K-2.1.** Oszczędność czasu spowodowana elektronicznym generowaniem mandatu bez pomyłek, w ujednoliconym standardzie co niesie za sobą szybsze obróbkę takich raportów przez dział windykacji.

**K-2.2.** Lista zapłaconych mandatów.

##### **Kierowcy**

**KP-3.1.** Płatności za parkowanie.

### **5.1.6 Ograniczenia projektowe i wdrożeniowe**

1. Finansowe - klient organizuje przetarg, w którym to kwota maksymalna została określona na 100.000,00 zł.
2. System powinien być zabezpieczony przed niepowołanym dostępem.
3. Techniczne - system powinien być powiązany z bazą danych, w której będzie gromadzona informacja dotycząca atrybutów opisujących parkingi, ewidencja pracowników, mandatów wystawianych klientom, listą opłaconych biletów. Nie ma możliwości ewidencji poszczególnych parkowań, nie ma obsługi sytuacji wyjątkowych (kradzieże pojazdów, mienia, dewastacje, remonty parkingów itp.) i pozbawiony komunikacji z urządzeniami peryferyjnymi, takimi jak np. parkometry.
4. Prawne - ustawa o ochronie danych osobowych, ustawa o prawach konsumenta, postępowanie mandatowe - Kodeks postępowania w sprawach o wykroczenia.
5. Technologie - język python
6. Protokoły komunikacyjne - HTTP, TCP.

## **5.2 Wymagania funkcjonalne**

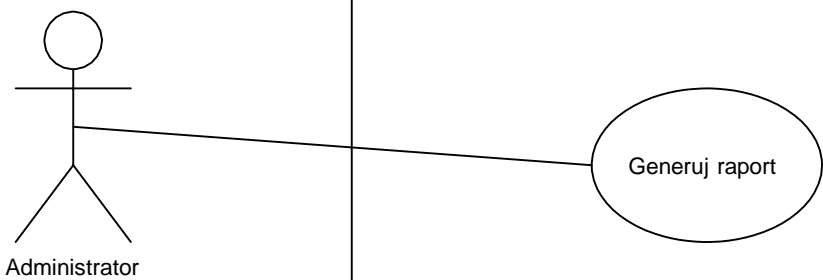
### **5.2.1 Lista wymagań**

1. Uruchomienie programu.
2. Edycja danych.
3. Płatności za parkowanie.
4. Wystawianie mandatów.
5. Lista mandatów.
6. Generowanie i wysyłanie raportów.

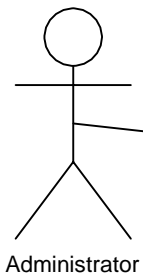
### **5.2.2 Diagramy przypadków użycia**



**Generowanie raportów**

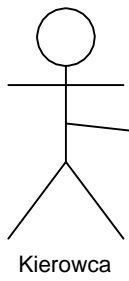


## Uruchomienie programu



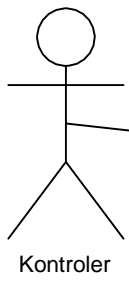
Zaloguj się

## Uruchomienie programu



Zaloguj się

## Uruchomienie programu

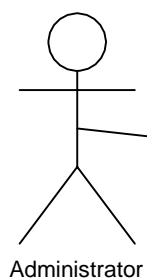


Zaloguj się

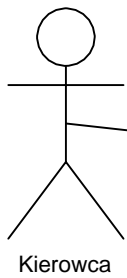
An oval containing the text "Zaloguj się", connected to the stick figure by a line.

## Edycja danych

Edytuj dane

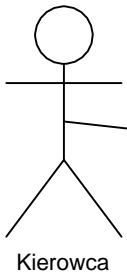


## Lista mandatów



Zobacz listę  
mandatów

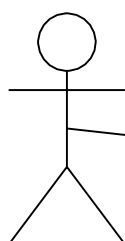
## Płatności za parkowanie



Kierowca

Zapłać za parkowanie

## Wystawianie mandatów



Kontroler

Wystaw mandat



### 5.2.3 Szczegółowy opis wymagań

**U.** Uruchomienie programu o przypadek Autoryzacja, brak możliwości uruchomienia programu przez osoby niepowołane. Trzy tryby uruchomienia – Administrator, Kontroler i Kierowca. Warunek początkowy: uruchomienie programu, powoduje to zaistnienie obiektu System. Częstotliwość – BW, istotność – BW.

1. Wyświetlone okno autoryzacji.
2. Wprowadzenie danych autoryzacyjnych.
3. 3.1. Autoryzacja poprawna - wyświetla się Główne okno programu.  
3.2. Autoryzacja niepoprawna - wyświetla się komunikat o błędzie autoryzacji.  
3.2.1. Przycisk OK - powrót do okna autoryzacji.  
3.2.2. Przycisk Anuluj - Zamknięcie programu.

Tryb Administratora (A)

**A-1.** Modyfikacja parkingu, Szukanie parkingu, Dodanie regionu. Warunek początkowy - przycisk "Szukaj" powoduje wyświetlenie okna dialogowego z opcjami wyszukiwania.

1. Jeśli użytkownik zrezygnuje z szukania, przycisk Anuluj spowoduje powrót do głównego okna programu.
2. W przeciwnym wypadku, po wypełnieniu okna dialogowego i wciśnięciu OK, aplikacja przejdzie w stan szukania. W zależności od rezultatu szukania, okno o braku znalezionych pozycji lub podświetlenie znalezionej.
3. Modyfikacja parkingu rozpoczyna się w momencie, w którym mając podświetlony dowolny parking, administrator wciśnie przycisk Edycja. Zostanie wtedy wyświetlone odpowiednie okno dialogowe, zawierające dane opisujące wybrany parking. Administrator może dokonać zmian w wyświetlonych danych, min. przypisać parking do innego regionu. Może też dodać nowy region przyciskiem 'Dodanie regionu', jeśli żaden z istniejących nie spełnia określonych warunków.
4. Kiedy użytkownik programu skończy dokonywać zmiany może zapisać je w bazie danych wciskając prz. 'OK', lub zrezygnować z wprowadzania zmian wciskając przycisk 'Anuluj'. Obie te sytuacje powodują powrót do okna głównego programu, jednakże w przypadku wciśnięcia prz. 'OK', nastąpi również aktualizacja bazy danych.

Częstotliwość – BW, istotność – BW.

**A-2.** Dodawanie i edycja danych o mandatach na podstawie kwitów z informacjami otrzymanymi od pracowników:

- imię i nazwisko wystawiającego kwit,
- data i godzina wystawienia,
- dane identyfikujące samochód: numer rejestracyjny, marka auta i typ auta (osobowy lub ciężarowy)
- parking na którym się pojazd znajdował (powiedzmy że są 4 parkingi).

Punktem wyjścia jest umieszczenie w kartotece wypisanych kwitów mandatowych przez pracownika. Następnie administrator dodaje każdy z tych kwitów do bazy. Jeśli użytkownik zgłosi chęć zapłacenia mandatu, administrator znajduje w bazie danych wprowadzony mandat, oraz edytuje go.

1. Dostarczenie wypełnionych kwitów administratorowi przez pracownika.
2. Wciśnięcie przez administratora przycisku 'Dodaj mandat'.
3. Otworzenie okna edycji mandatu przez system.
4. Wypełnienie okna przez administratora.
5. Zapisanie danych w bazie.
6. Uiszczenie opłaty przez klienta.
7. Znalezienie mandatu przez administratora, oraz jego modyfikację.

Częstotliwość - BW, istotność - BW.

**A-3. Generowanie raportów:**

- lista niezapłaconych mandatów (lista dłużników w celu rozpoczęcia windykacji),
- lista wystawionych mandatów przez poszczególnych pracowników (identyfikacja pracowników biorących łapówki),
- lista dokonanych wpłat dla określonego okresu (inventaryzacja).

Punkt wyjścia: umieszczenie w kartotece wypisanych kwitów mandatowych przez pracownika.

Częstotliwość - BW, istotność - BW.

Tryb Kontroler:

**K-1.** Wystawienia mandatu przyciskiem 'Dodaj kwit' oraz wydrukowaniu go w oknie edycji za pomocą sparowanego, przenośnego urządzenia do drukowania.

Częstotliwość - W, istotność - BW.

1. Wciśnięcie przez administratora przycisku 'Dodaj kwit'.
2. Otworzenie okna edycji mandatu przez system.
3. Wypełnienie okna przez kontrolera:
  - imię, nazwisko klienta,
  - data, godzina wystawienia mandatu.
4. Zapisanie danych w bazie przyciskiem 'Zapisz', co spowoduje przesłanie kwitu do administratora.
5. Druk.

Tryb Kierowca Pojazdu:

**KP-1.** Płatności za parkowanie. System powinien rozróżniać typy pojazdów korzystających z usług parkingu. Warunek początkowy - przycisk 'Szukaj' powoduje wyświetlenie okna dialogowego z opcjami wyszukiwania parkingu.

1. Jeśli użytkownik zrezygnuje z szukania, przycisk 'Anuluj' spowoduje powrót do głównego okna programu.
2. W przeciwnym wypadku, po wypełnieniu okna dialogowego i wciśnięciu OK, aplikacja przejdzie w stan szukania. W zależności od rezultatu szukania, okno o braku znalezionych pozycji lub podświetlenie znalezionej.
3. Wybór parkingu.
4. Wybór parametrów wpływających na cenę biletu:
  - typ pojazdu (samochód ciężarowy, samochód osobowy, pojazd jednośladowy)
  - czas trwania postoju (15 min, 30 min, 1h, 1.5h, oraz opcja bez limitu - klient ponosi opłatę po zakończeniu parkowania i odznaczeniu w aplikacji odpowiedniego wariantu).
5. Opłata z potwierdzeniem na e-mail.

Częstotliwość - BW, istotność - BW.

**KP-2.** Lista zapłaconych mandatów. Kierowca pojazdu ma możliwość sprawdzenia w aplikacji posiadanych mandatów.

Częstotliwość - W, istotność - W.

### **5.3 Wymagania niefunkcjonalne**

1. System musi umożliwiać integrację z bramką e-mail dostępną w infrastrukturze Klienta.
2. System musi zostać zaimplementowany w taki sposób aby wszystkie moduły funkcjonalne mogły działać niezależnie od siebie - awaria jednego modułu nie może blokować działania pozostałych.
3. Ochrona danych - uniemożliwienie dostępu nieuprawnionych użytkowników do poufnych danych innych użytkowników.
4. Zarządzanie współbieżnym dostępem - obsługa wielu użytkowników w taki sposób aby każdy z nich był niewidoczny dla innych użytkowników.
5. System musi informować Administratora o incydentach bezpieczeństwa za pomocą email.
6. Interfejs użytkownika musi być prezentowany w języku polskim oraz angielskim.
7. System musi spełniać wymagania rozporządzenia RODO.
8. System musi być skalowalny do min. 5000 Kierowców.
9. Instrukcja obsługi musi być dostarczona w formacie PDF w języku polskim.

## 6 Zarządzanie projektem

### 6.1 Zasoby ludzkie

Programiści python; Zarząd Dróg Miejskich w Szczecinie.

### 6.2 Harmonogram prac

- Analiza - rozpoznanie, wyjaśnianie, modelowanie, dokumentowanie rzeczywistości lub problemu będącego przedmiotem projektu. Logiczny model systemu, opisujący sposób realizacji przez system postawionych wymagań, lecz abstrahując od szczegółów implementacyjnych.
- Projektowanie. W odróżnieniu od fazy analizy, w trakcie projektowania dużą rolę odgrywa charakterystyka środowiska implementacji. Projektanci muszą więc posiadać dobrą znajomość języków, bibliotek i narzędzi stosowanych w trakcie implementacji. Projektanci muszą więc posiadać dobrą znajomość języków, bibliotek i narzędzi stosowanych w trakcie implementacji. Polega na dostosowaniu systemu do ograniczeń i możliwości środowiska implementacji.
- Implementacja - finalna faza tworzenia oprogramowania. Polega na samym programowaniu, czyli zamianie założeń wybranych w fazach analizy i - przede wszystkim - projektowania na faktyczny kod programu.



### 6.3 Etapy/kamienie milowe projektu

1. Ustalenie kontekstu projektu, wymagań użytkowników, wymagań organizacyjnych oraz inne ustalenia, np. dotyczące preferencji sprzętowych, preferencji w zakresie oprogramowania, ograniczeń finansowych, ograniczeń czasowych itd.
2. Opis implementacji systemu.
3. Program.

## **7 Zarządzanie ryzykiem**

### **7.1 Lista czynników ryzyka**

1. Naturalne - katastrofy.
2. Ludzkie - kradzież mienia, transportu, remont dróg, parkingów, awarie, konflikty socjalne, trudność w zarządzaniu pracownikami, dostępnością stron zaangażowanych, zła jakość prac podwykonawców.
3. Pochodzenia władzy administracyjnej lub sądowej (interwencja administracyjna, wprowadzenie nowych form modyfikujących specyfikację projektu).
4. Organizacyjnej, związanej z procesem decyzyjnym.

### **7.2 Ocena ryzyka**

1. Zagrożenie mało prawdopodobne. Niewielki wpływ.
2. Zagrożenie prawdopodobne i wpływ odczuwający.
3. Zagrożenie prawdopodobne i może mieć znaczny wpływ.
4. Zagrożenie prawdopodobne i może mieć odczuwający wpływ.

### **7.3 Plan reakcji na ryzyko**

1. Przepływ informacji.
2. Kierowanie projektem, przez dopasowanie go do zmieniających się warunków zewnętrznych i przygotowanie go na konfrontację ze zdarzeniami, które mogą zaistnieć.
3. Informowanie stron zaangażowanych w projekt o istniejących zagrożeniach dla projektu.
4. Lepsze poznanie i zrozumienie zagrożeń mogących się pojawić, a także przygotowaniu działań zapobiegawczych.
5. Opisanie i udokumentowanie zdarzeń.
6. Plan awaryjny.



## 8 Zarządzanie jakością

### 8.1 Scenariusze i przypadki testowe

- **A-2.** Dodawanie i edycja danych o mandatach na podstawie kwitów z informacjami otrzymanymi od pracowników - testujemy interfejs, działanie komunikacji między modułami.
- Testowanie integracyjne.
- Poprawnie wysłane prawidłowe kwity, poprawna edycja danych mandatów.
- Tester - Kontroler i Administrator.
- Testowanie ma być przeprowadzane po wdrożeniu całości programu.
- Stubs and Drivers.
- Przebieg działań.

lp.	Tester	System
1.	Kontroler wysyła wypełniony kwit	Wysłano i otrzymano kwit do administratora
2.	Administrator dodaje kwit do bazy	Dodaje możliwość edycji danych i ich zapisu.
3.	"Zapłacenie" mandatu. Znalezienie mandatu przez administratora, oraz jego modyfikacja.	Poprawne wyszukiwanie mandatu, edycja i zapis do bazy danych.

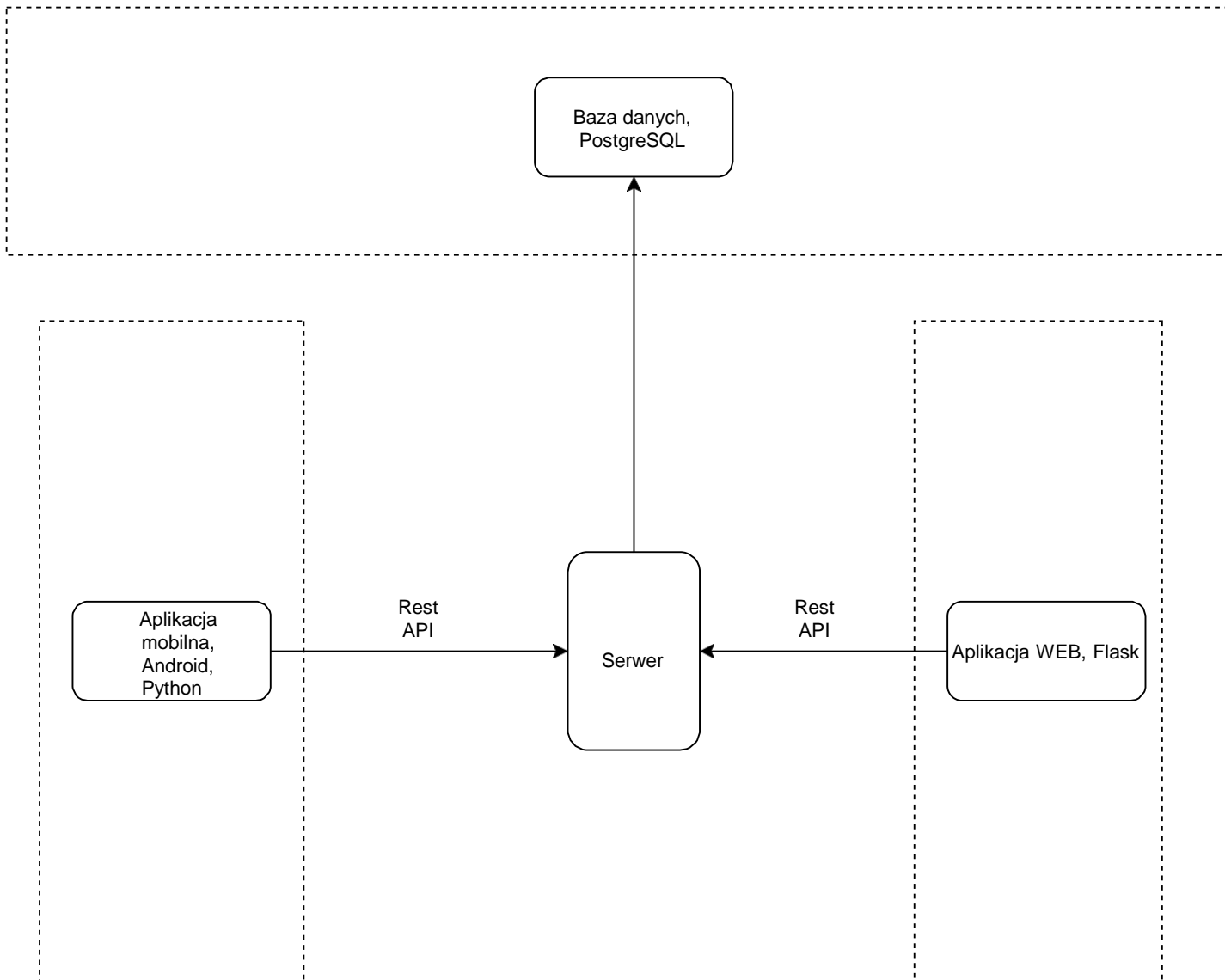
- Założenia, środowisko, warunki wstępne, dane wejściowe – przygotowanie przed uruchomieniem testów:
  1. Komponenty / moduły przeszły test jednostkowy.
  2. Wszystkie wady o wysokim priorytecie zostały rozwiązane i zamknięte.
  3. Wszystkie moduły są gotowe i można je zintegrować.
  4. Plan testów integracji, scenariusze testów i przypadki testowe są gotowe i zatwierdzone.
  5. Środowisko testowe jest gotowe do testów integracyjnych.
- Zestaw danych testowych – najlepiej w formie tabelarycznej – jakie konkretnie dane mają być użyte przez testera i zwrócone przez system w poszczególnych krokach przebiegu działań:
  - Imię i nazwisko wystawiającego kwit, data i godzina wystawienia, dane identyfikujące samochód: numer rejestracyjny, marka auta i typ auta (osobowy lub ciężarowy), parking na którym się pojazd znajdował (powiedzmy że są 4 parkingi)

## **9 Projekt techniczny**

### **9.1 *Opis architektury systemu***

Komponenty systemu:

- baza danych PostgreSQL,
- serwer,
- aplikacja mobilna,
- aplikacja WEB



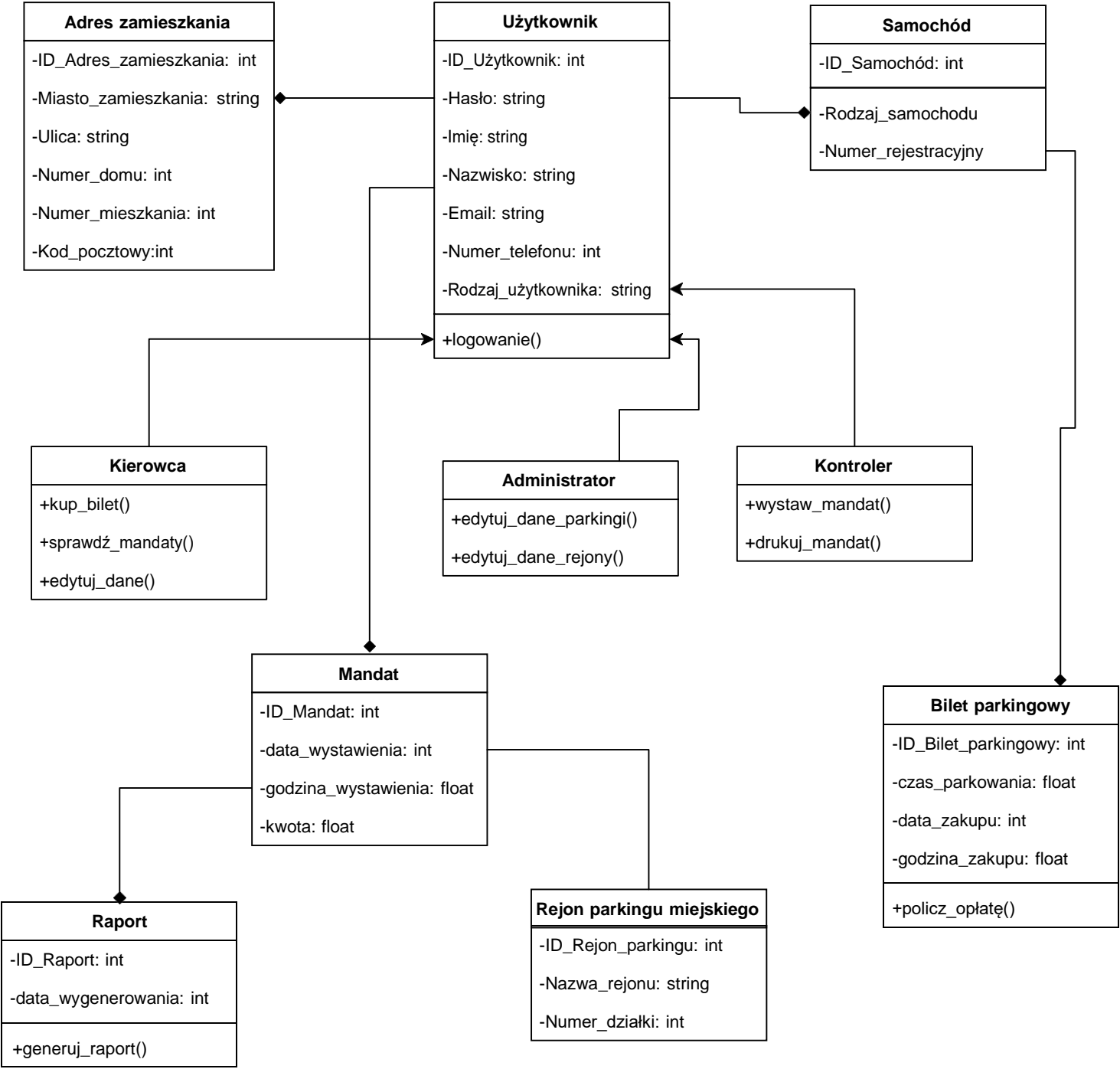
## 9.2 Technologie implementacji systemu

Technologia	Uzasadnienie
Baza danych PostgreSQL	Baza danych jest bezpłatna do wszystkich zastosowań komercyjnych, udostępniania wraz z pełnym kodem źródłowym, dzięki czemu istnieje możliwość dostosowywania bazy do indywidualnych potrzeb.
Flask framework	Dzięki zastosowaniu tego frameworka przy tworzeniu aplikacji WWW nie będzie konieczne wykorzystanie dodatkowego serwera WEB.
Rest API	Rest API jest uniwersalne, jego interfejs intuicyjny i wygodny. Duża liczba użytkowników sprawia, że ewentualne problemy będą szybko rozwiązane.
Python	
Android	Jest to najpopularniejszy system operacyjny na urządzenia mobilne, dzięki wykorzystaniu Androida system dotrze do jak największej liczby odbiorców.

## **9.3 *Diagramy UML***

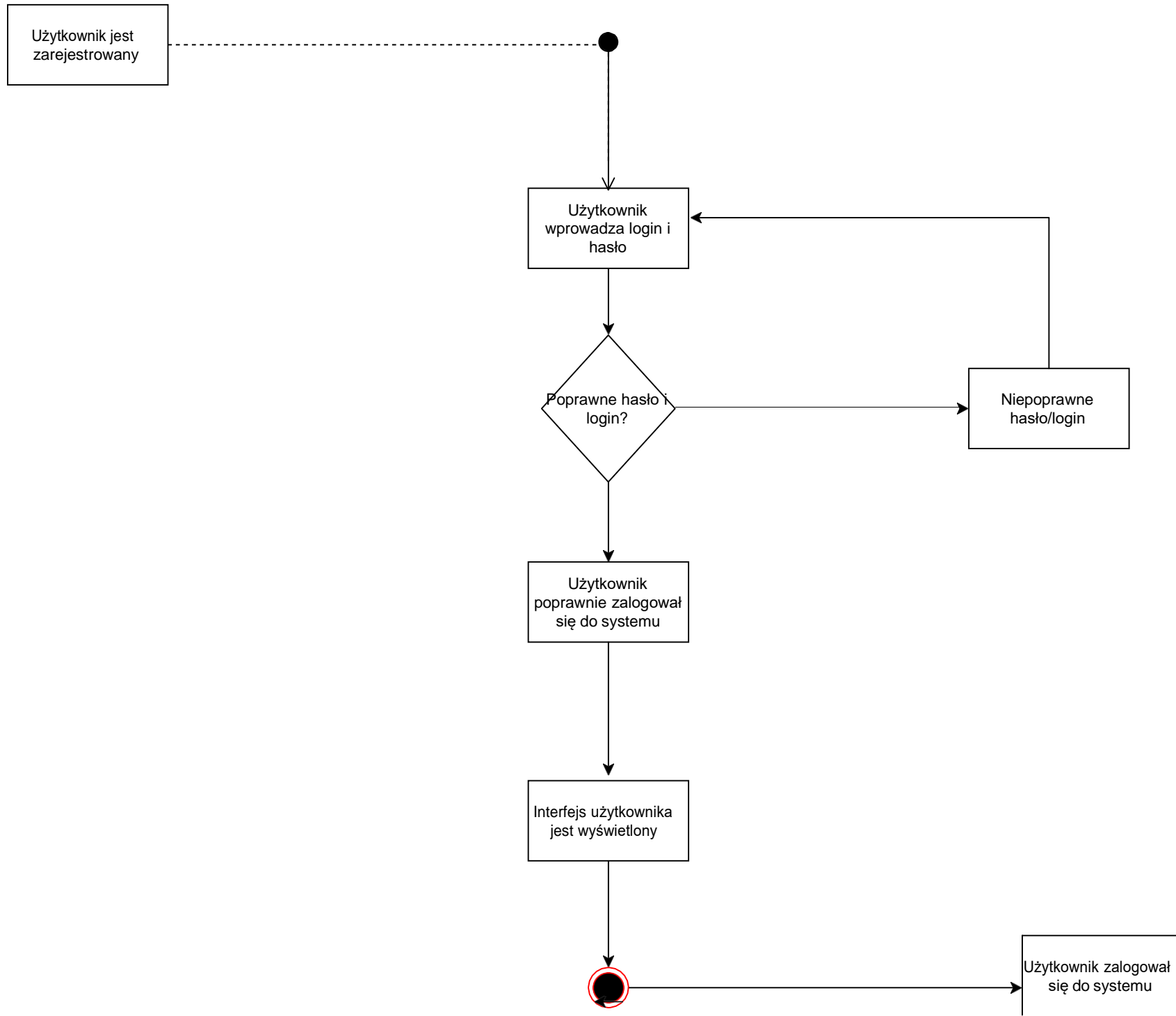
### **9.3.1 Diagram(-y) klas**

Diagram klas



### **9.3.2 Diagram(-y) czynności**

## Diagram czynności: Logowanie się do systemu przez użytkownika





### 9.3.3 Diagramy sekwencji

## Diagram sekwencji: Logowanie się do systemu

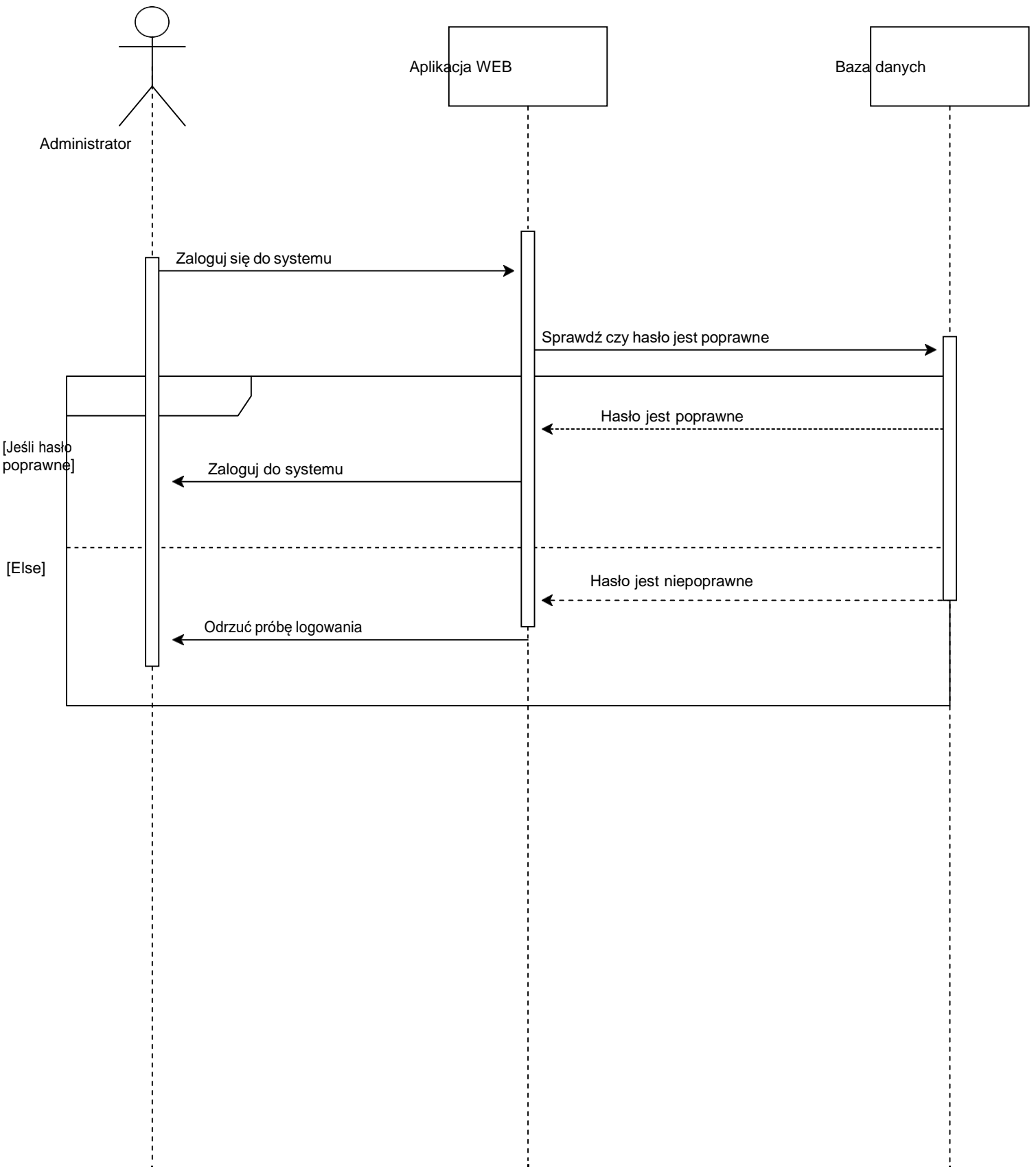
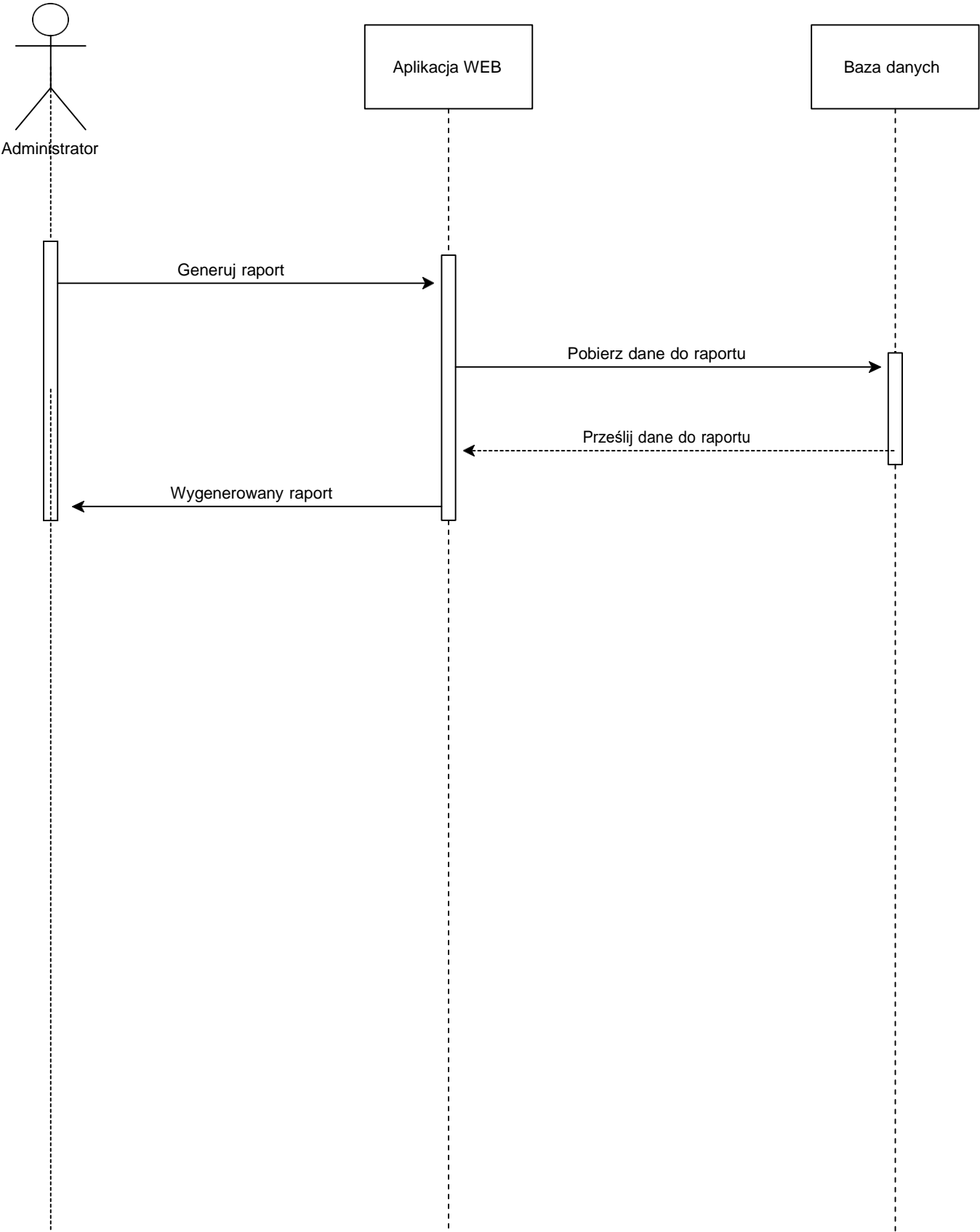
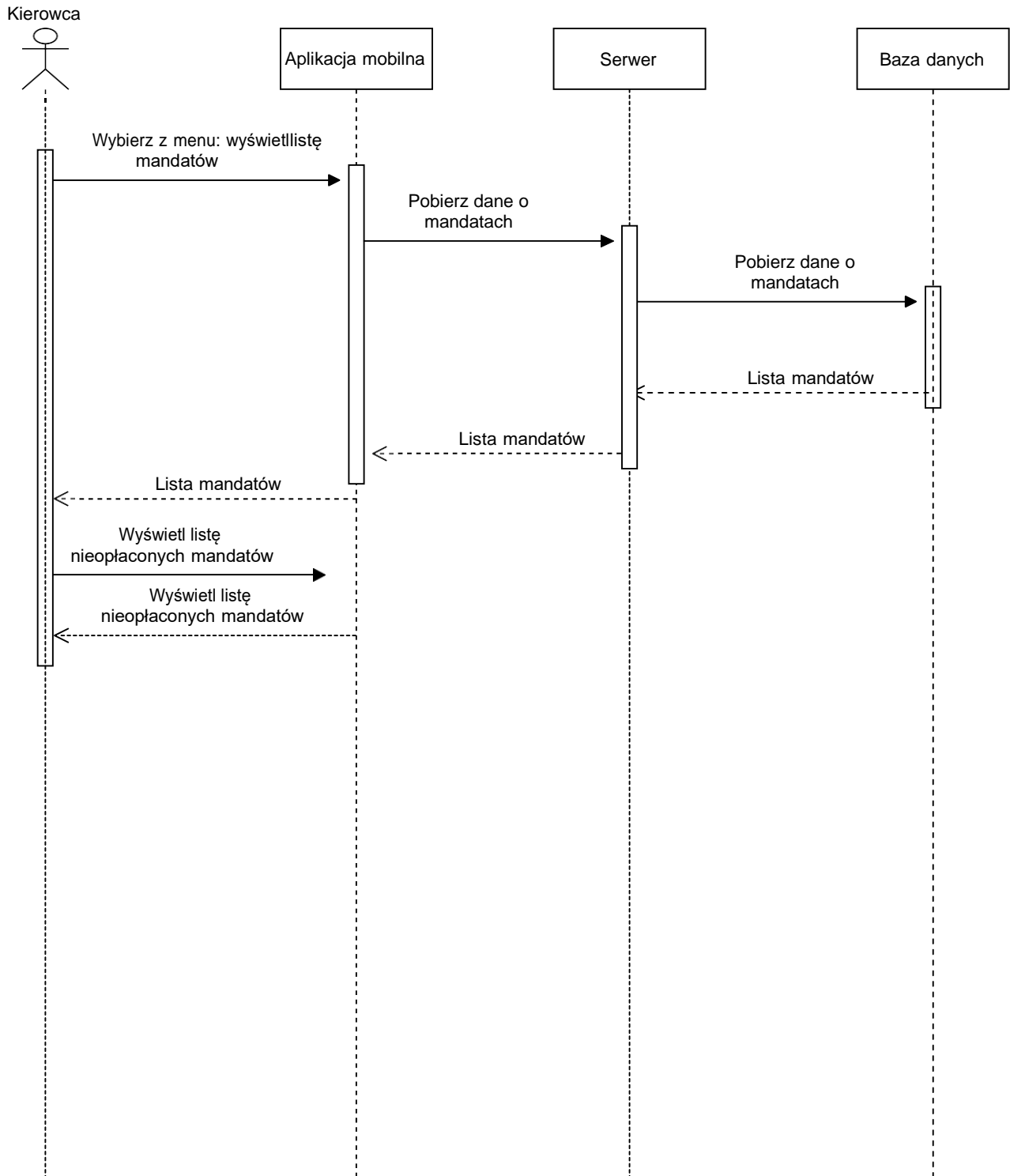


Diagram sekwencji: Generowanie raportu



## Diagram sekwencji: wyświetlanie listy mandatów



## Diagram sekwencji: Kupowanie biletu

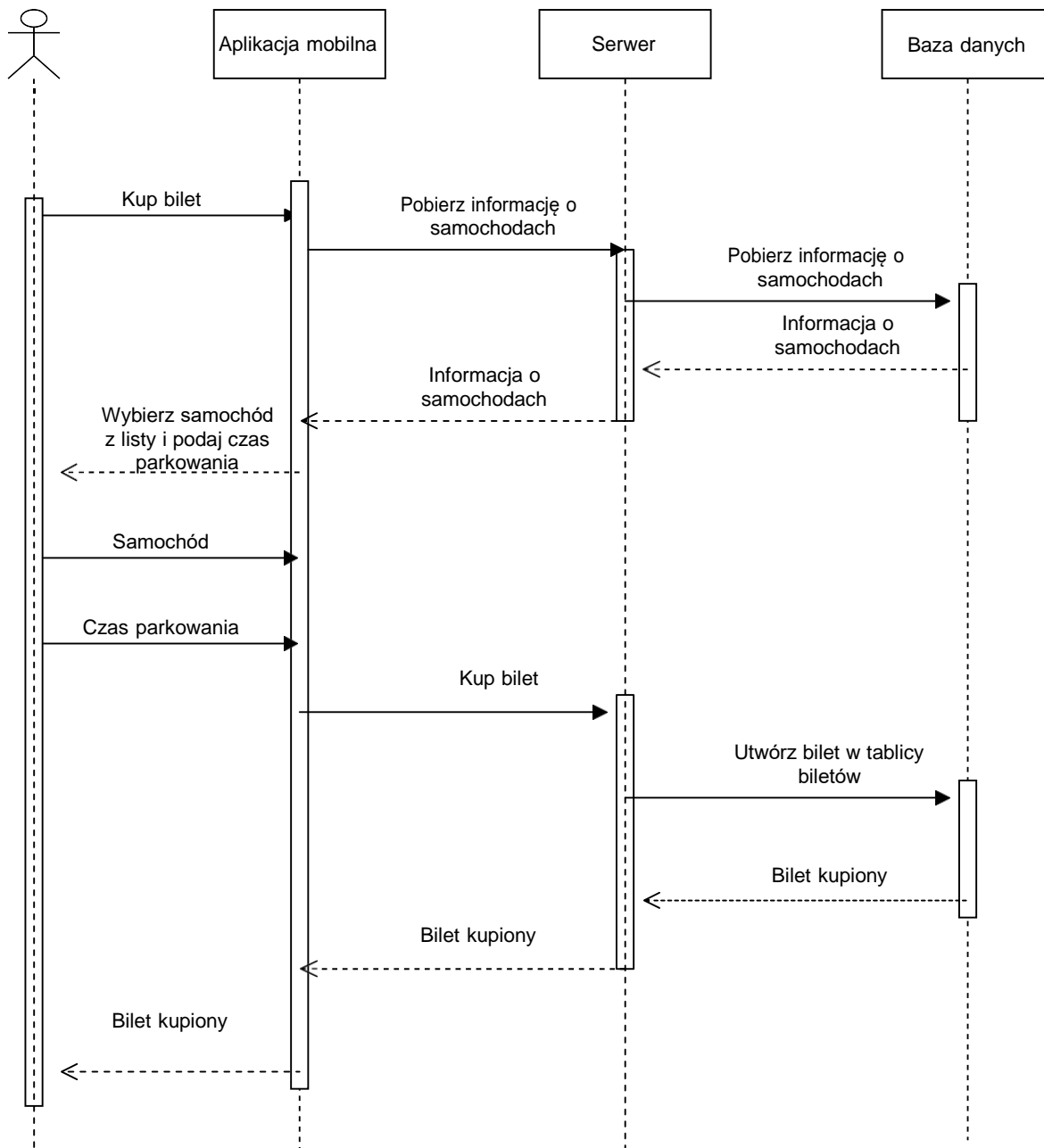
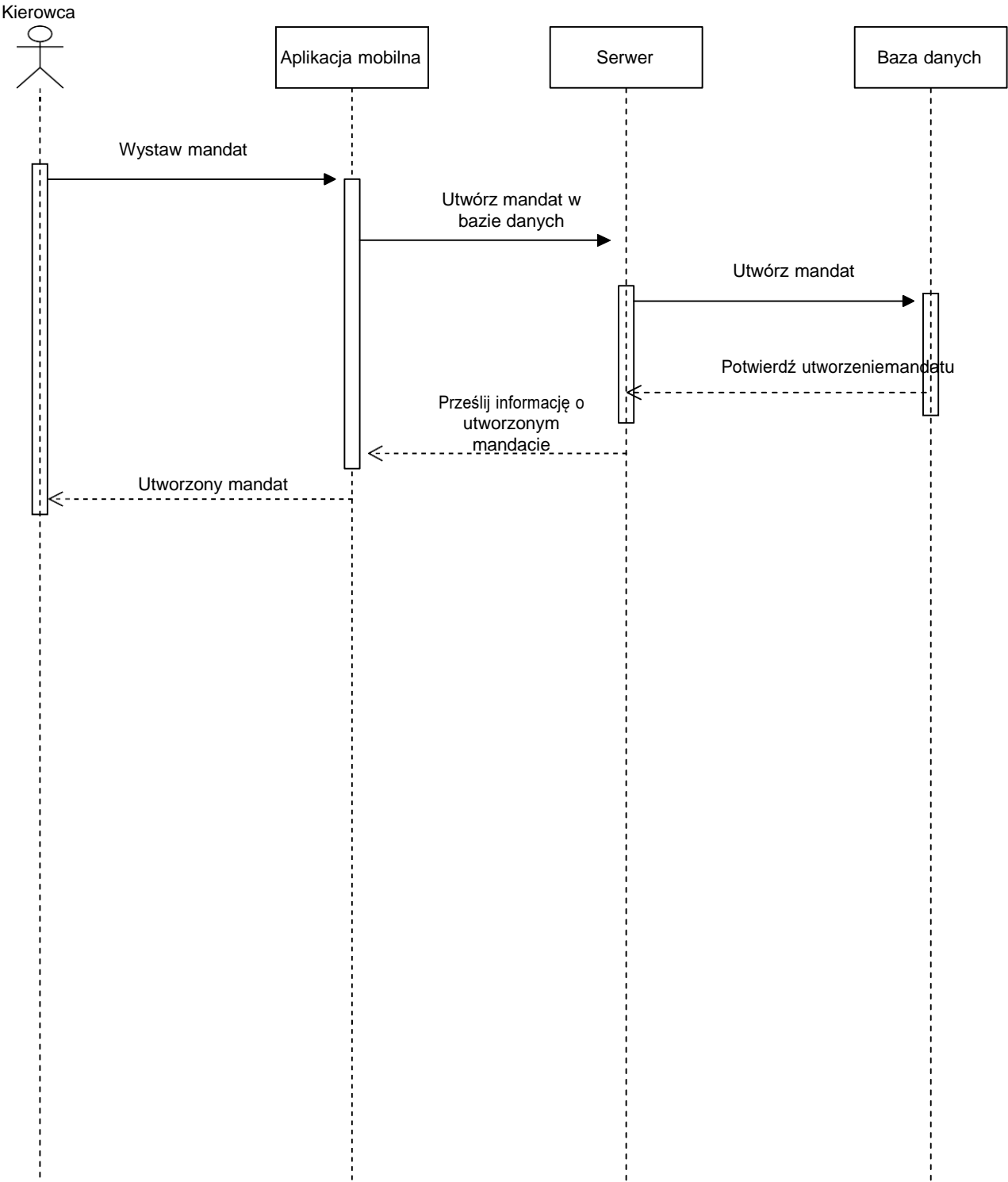
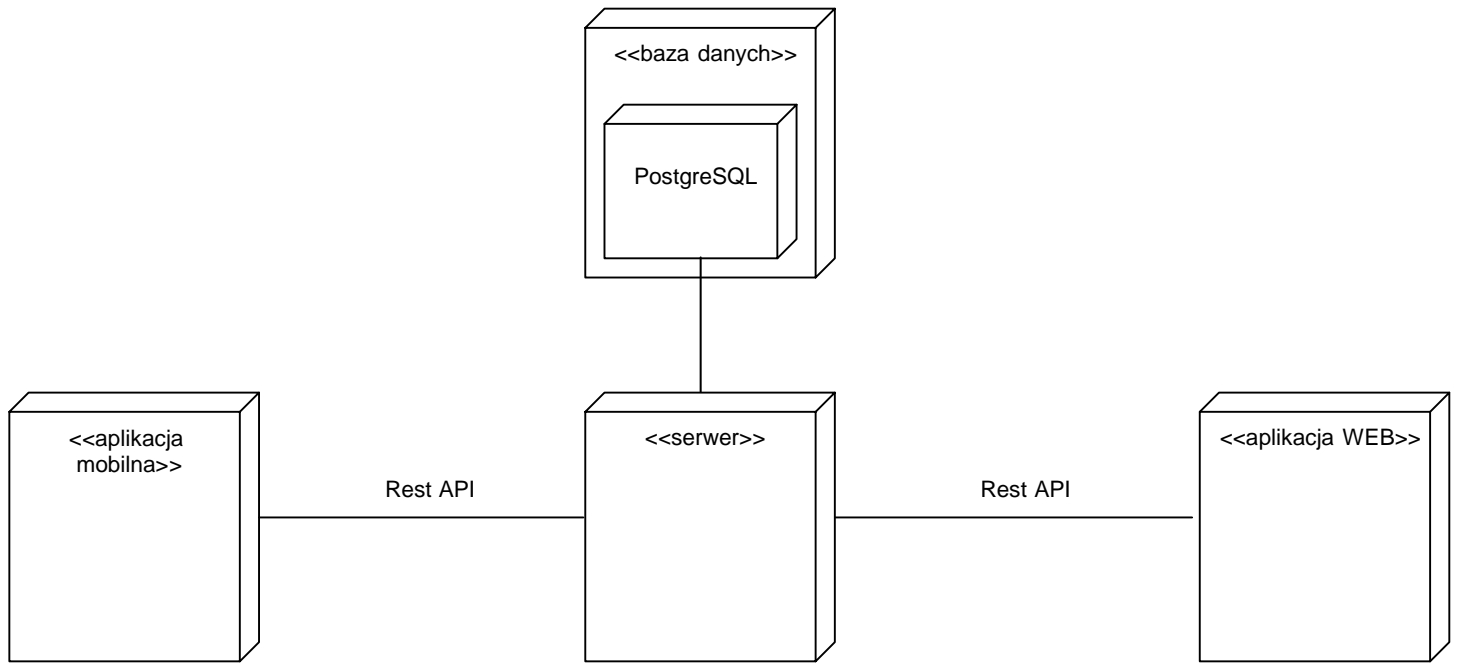


Diagram sekwencji: wystawienie mandatu



#### **9.3.4 Inne diagramy**

# Diagram wdrożenia





## Diagram maszyny stanowej: wystawienie mandatu

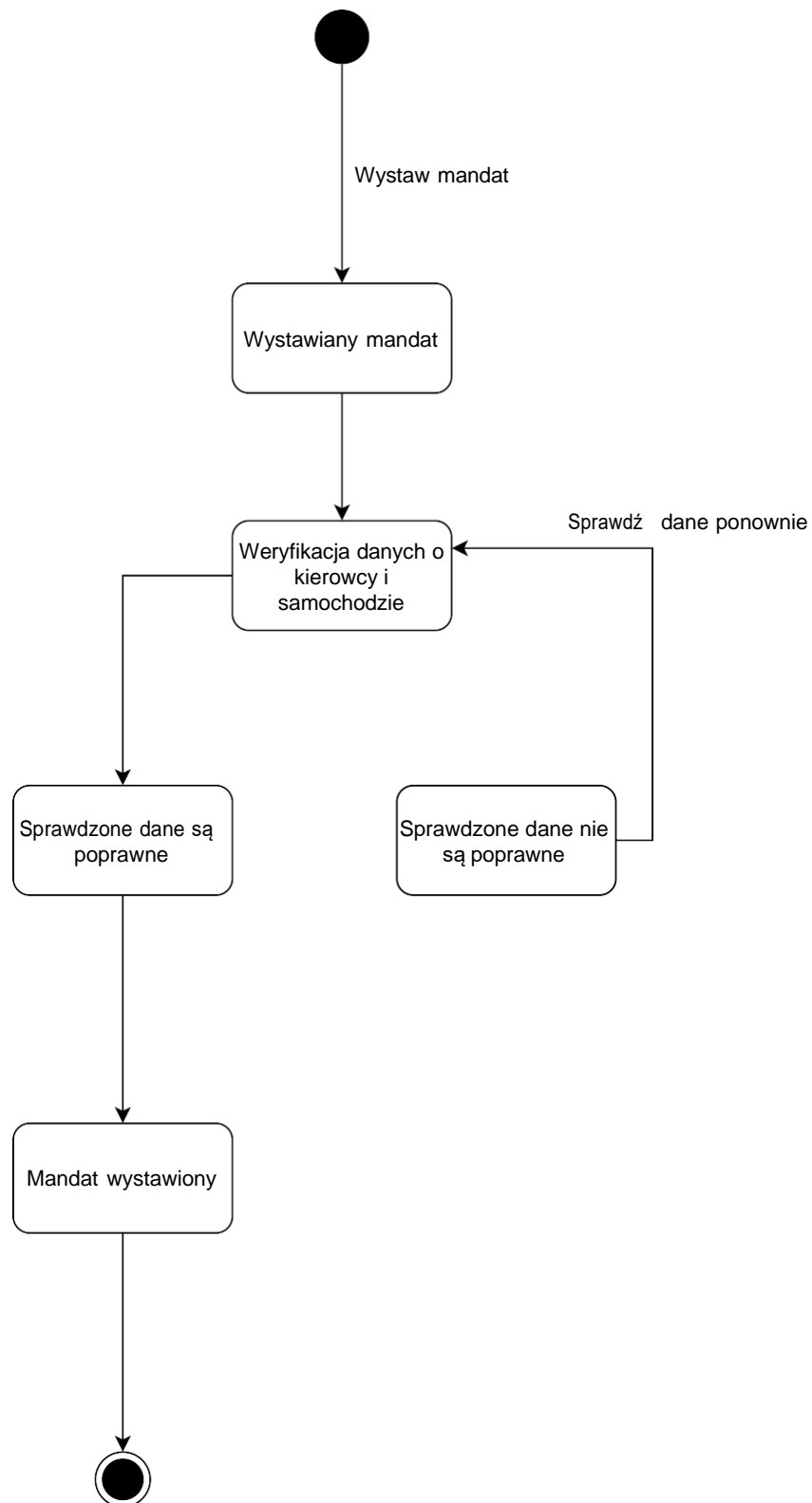
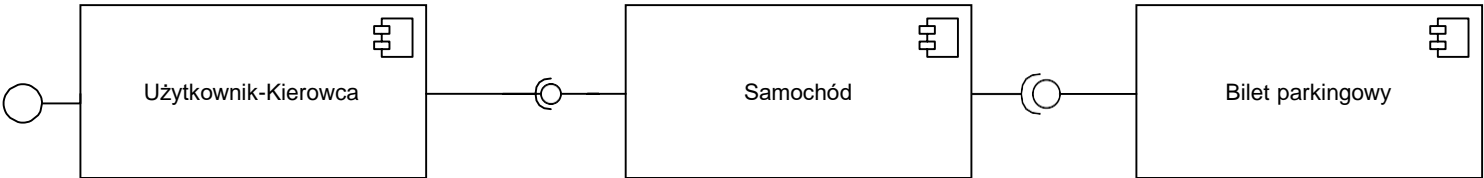


Diagram komponentów: kupno biletu parkingowego

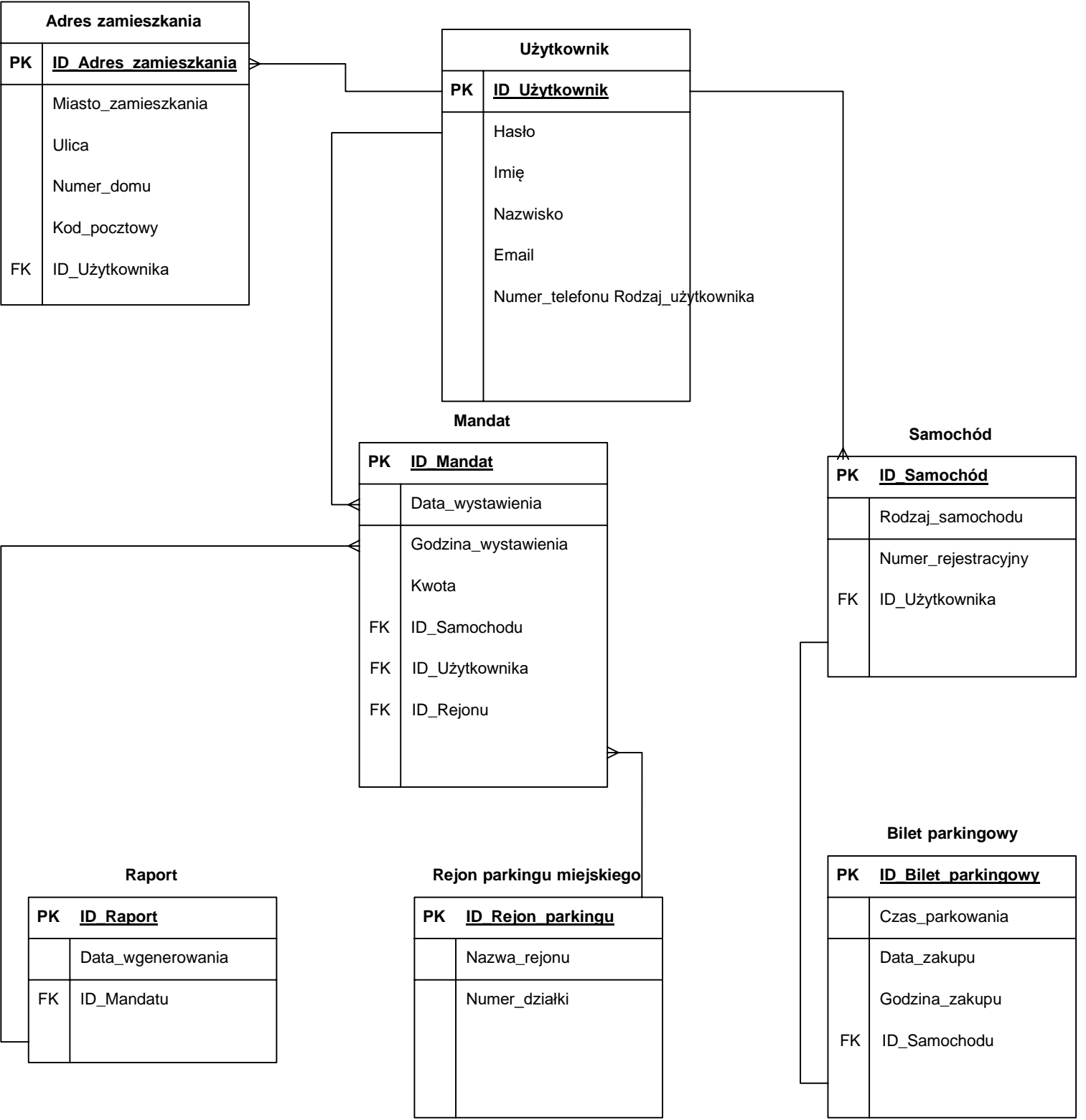


#### **9.4 Charakterystyka zastosowanych wzorców projektowych**

NIE WYKORZYSTANO WZORCÓW PROJEKTOWYCH

## **9.5 *Projekt bazy danych***

### **9.5.1 Schemat**



### 9.5.2 Projekty szczegółowe tabel

Użytkownik	
Pole	Opis
ID_Kierowca	Klucz główny - generowany automatycznie
Hasło	Hasło nie zaszyfrowane
Imię	Imię
Nazwisko	Nazwisko
Email	Adres e-mail
Numer_telefonu	Numer telefonu
Rodzaj_użytkownika	Rodzaj użytkownika: administrator, kierowca, kontroler

Adres zamieszkania	
Pole	Opis
ID_Adres_zamieszkania	Klucz główny - generowany automatycznie
Miasto_zamieszkania	Miasto zamieszkania
Ulica	Ulica
Numer_domu	Numer domu
Kod_pocztowy	Kod pocztowy, np. 71000
ID_Użytkownika	Klucz obcy

Rejon parkingu miejskiego	
Pole	Opis
ID_Rejon_parkingu	Klucz główny - generowany automatycznie
Nazwa_rejonu	Nazwa inwentaryzacyjna
Numer_działki	Numer działki z księgi wieczystej

Raport	
Pole	Opis
ID_Raport	Klucz główny - generowany automatycznie
Data_wygenerowania	Data wygenerowania raportu
ID_Mandatu	Klucz obcy

Bilet parkingowy	
Pole	Opis
ID_Bilet_parkingowy	Klucz główny - generowany automatycznie
Czas_parkowania	Czas parkowania podany
Data_zakupu	Data zakupu biletu
Godzina_zakupu	Godzina zakupu biletu
ID_samochodu	Klucz obcy

Mandat	
Pole	Opis
ID_Mandat	Klucz główny - generowany automatycznie
Data_wystawienia	Data wystawienia mandatu
Godzina_wystawienia	Godzina wystawienia mandatu
Kwota	Kwota mandatu
ID_Samochodu	Klucz obcy
ID_Użytkownika	Klucz obcy
ID_Rejonu	Klucz obcy

Samochód	
Pole	Opis
ID_Bilet_parkingowy	Klucz główny - generowany automatycznie
Rodzaj_samochodu	Rodzaj samochodu, np. ciężarowy, jednośladowy, osobowy
Numer_Rejestracyjny	Numer rejestracyjny, np. ZS1234
ID_Użytkownika	Klucz obcy

## 9.6 Projekt interfejsu użytkownika

Projekt dla aplikacji kierowcy.

### 9.6.1 Lista głównych elementów interfejsu

- ekran logowania
- ekran niepoprawnego logowania
- ekran główny: przyciski kup bilet, zobacz listę mandatów, edytuj dane, wyloguj się
- ekran listy mandatów: przycisk pokaż nieopłacone mandaty, lista mandatów, przycisk powrót do menu
- ekran kupna biletu: przycisk wybierz rodzaj auta, wpisz czas parkowania, dokonaj zakupu, powrót do menu



- ekran edycji danych: przycisk edytuj dane adresowe, edytuj dane o samochodzie, powrót do menu
- ekran wylogowania: przycisk zaloguj się ponownie
- kupno biletu
- wyświetlanie listy mandatów
- edycja danych

### **9.6.2 Przejścia między głównymi elementami**

- przejście z ekranu logowania do ekranu niepoprawnego logowania
- przejście z ekranu logowania do ekranu głównego
- przejście z ekranu głównego do ekranu kupna biletu
- przejście z ekranu głównego do ekranu edycji danych
- przejście z ekranu głównego do ekranu listy mandatów
- przejście z ekranu głównego do ekranu wylogowania

### **9.6.3 Projekty szczegółowe poszczególnych elementów**

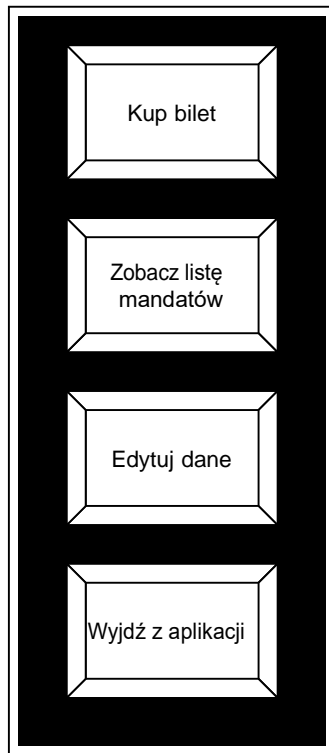
## K01 Ekran logowania



## K02 Ekran niepoprawnego logowania



## K03 Ekran główny



## K04 Ekran listy mandatów

Pokaż  
niezapłacone

Powrót do menu

## K05 Ekran kupna biletu

Kup bilet

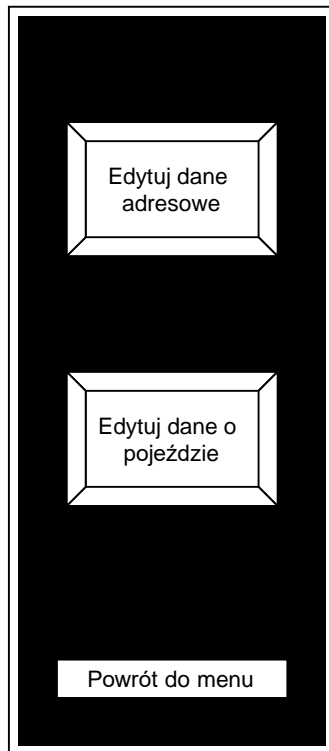
Wybierz rodzaj  
pojazdu

Wpisz czas  
parkowania

Dokonaj zakupu

Powrót do menu

## K06 Ekran edycji danych



The image shows a vertical rectangular screen with a black background. It contains three white rectangular buttons with black text, arranged vertically. The top button is labeled 'Edytuj dane adresowe', the middle button is labeled 'Edytuj dane o pojeździe', and the bottom button is labeled 'Powrót do menu'.

Edytuj dane  
adresowe

Edytuj dane o  
pojeździe

Powrót do menu



## K07 Ekran wylogowania



### **9.7 Procedura wdrożenia**

1. Przygotowanie dokumentacji powykonawczej
2. Przygotowanie i konfigurowanie infrastruktury technicznej
3. Zainstalowanie i skonfigurowanie systemu informatycznego do testów
4. Testowanie systemu
5. Zainstalowanie i skonfigurowanie systemu informatycznego do eksploatacji
6. Szkolenie użytkowników
7. Uruchomienie produkcyjne systemu - rozpoczęcie pracy przez użytkowników
8. Dokumentacja powdrożeniowej

## 10 Podsumowanie

### *10.1 Szczegółowe nakłady projektowe członków zespołu*

	Valeriia Balanivska	Daria Rudowicz
Analiza	4 dni, 57,2%	3 dni, 42,8%
Projektowanie	6,4 dni, 42,8%	8,6 dni, 57,2%
Implementacja	5 dni, 50%	5 dni, 50%