

Projekt programistyczny

Podstawy internetu rzeczy

Prowadzący laboratorium: dr inż. Krzysztof Chudzik

Kacper Gaudyn, nr indeksu 266873

Kuba Krapiec, nr indeksu 266503

Michał Pesta, nr indeksu 266899

Michał Trojanowski, nr indeksu 266864

Data ukończenia pracy: ...

Spis treści

1	Wymagania projektowe	2
1.1	Podstawowe wymagania funkcjonalne	2
1.1.1	Kierowca	2
1.1.2	Pracownik	2
1.1.3	Administrator	2
1.2	Podstawowe wymagania niefunkcjonalne	2
2	Opis architektury systemu	3
2.1	Elementy architektury z opisem	3
2.2	Graficzna reprezentacja architektury	3
2.3	Baza danych	4
2.3.1	Schemat bazy danych	4
2.3.2	Scenariusze i ich wpływ na dane	4

1 Wymagania projektowe

1.1 Podstawowe wymagania funkcjonalne

1.1.1 Kierowca

- Jako kierowca chcę zmieniać przystanki za pomocą terminalu w trakcie wykonywania kursu
- Jako kierowca chcę wybierać trasę kursu w terminalu przy rozpoczynaniu jazdy
- Jako kierowca chcę za pomocą terminalu rozpoczynać jazdę i ją kończyć na dowolnym przystanku
- Jako kierowca chcę, aby po dotarciu na ostatni przystanek jazda kończyła się automatycznie

1.1.2 Pracownik

- Jako pracownik chcę wsiadać na przystanku i płacić za przejazd za pomocą swojej karty pracowniczej
- Jako pracownik chcę, aby koszt przejazdu był zależny od przejechanych przystanków i był pobierany dopiero po zakończeniu przejazdu (przyłożeniu karty pracowniczej drugi raz)

1.1.3 Administrator

- Jako administrator chcę zarządzać¹ wszystkimi kursami
- Jako administrator chcę zarządzać wszystkimi przystankami
- Jako administrator chcę zarządzać wszystkimi pracownikami, w szczególności ilością pieniędzy na ich koncie

1.2 Podstawowe wymagania нефunkcjonalne

- System powinien obsługiwać więcej niż jednego pracownika naraz
- System powinien obsługiwać wiele pojazdów (w tym terminalów) naraz
- Operacje wykonywane na terminalu (płatność przez pracownika, zmiana trasy i przystanków przez kierowcę) powinny być jak najszybciej przekazywane do bazy danych

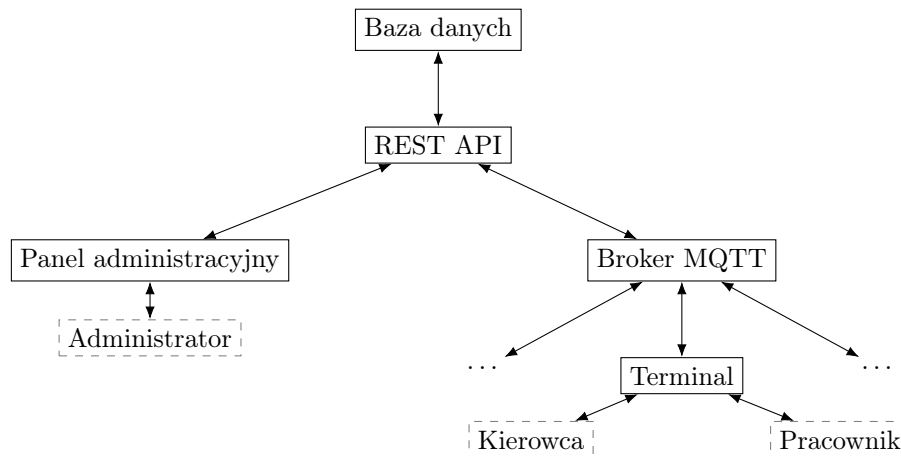
¹Poprzez zarządzanie rozumiemy operacje: dodawania, odczytu, aktualizowania i usuwania

2 Opis architektury systemu

2.1 Elementy architektury z opisem

- Panel administracyjny - służy administratorom do zarządzania danymi
- Baza danych - zawiera wszystkie dane
- REST API - zarządza danymi bazy danych i udostępnia odpowiednie operacje innym podmiotom
- Broker MQTT - komunikuje się z terminalami w pojazdach i przekazuje informacje o przejechanych przystankach i płatnościach
- Terminal - znajduje się w każdym pojeździe, umożliwia kierowcy ustalanie trasy i zmianę przystanków, oraz umożliwia pracownikom płacenie za swoje przejazdy
- Użytkownicy
 - Kierowca - kieruje pojazdem i na terminalu może zmieniać trasy, przystanki
 - Pracownik - może wsiadać do pojazdów i płacić w terminalu za przejazd kartą
 - Administrator - zarządza danymi w systemie

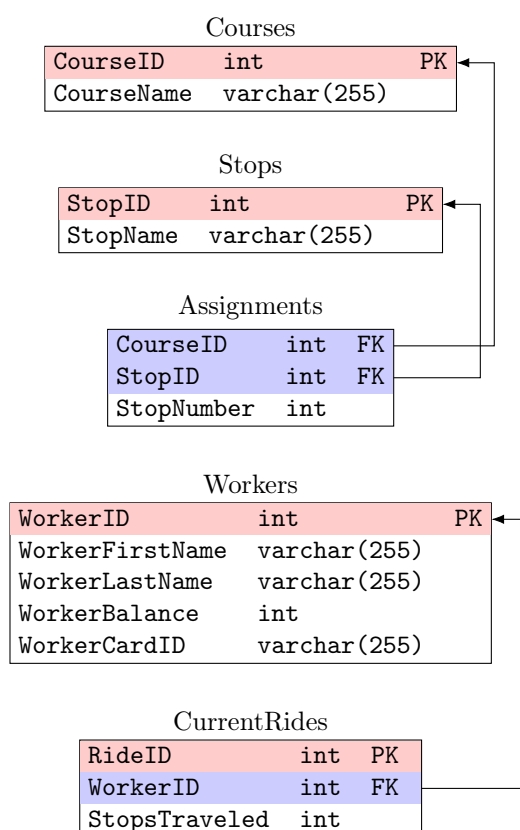
2.2 Graficzna reprezentacja architektury



Rysunek 1: Diagram elementów architektury z kierunkami przekazywania danych

2.3 Baza danych

2.3.1 Schemat bazy danych



Rysunek 2: Tabele bazy danych z zaznaczonymi relacjami między nimi

2.3.2 Scenariusze i ich wpływ na dane

1. Pracownik wsiada na przystanku, przykłada kartę i po przejechaniu 3 przystanków przykłada ją znowu aby zapłacić i wysiada.
Zakładamy, że każdy przystanek kosztuje 10, czyli pracownik zapłaci 30.

Workers				
WorkerID	WorkerFirstName	WorkerLastName	WorkerBalance	WorkerCardID
1	Jan	Kowalski	100	ABC

Rysunek 3: Dane w bazie przed wykonaniem scenariusza

Workers				
WorkerID	WorkerFirstName	WorkerLastName	WorkerBalance	WorkerCardID
1	Jan	Kowalski	70	ABC

CurrentRides		
RideID	WorkerID	StopsTraveled
⋮		
5	1	3
⋮		

Rysunek 4: Dane w bazie po wykonaniu scenariusza

W tabeli **Workers** zmieniła się kolumna **WorkerBalance**, a w tabeli **CurrentRides** został dodany nowy rekord.