Dokumentacja projektu ZPI

Dąbrowski Daniel
Gródek Damian
Jargut Piotr
Kocjan Katarzyna
Uryga Kamil

10 maja 2022

Streszczenie

Zarys dokumentacji projektowej stanowiący podstawę do realizacji projektu zespołowego.

Spis treści

Spis rysunków

1 Tytuł: System wspomagający szybkie parkowanie

2 Nazwa robocza: Parkingowy

3 Cel

Zoptymalizować ruch na parkingach i zniwelować długi czas szukania miejsca parkingowego, co pozwoli na zmniejszenie emisji spalin do atmosfery.

Przewidywany termin ukończenia projektu 10 Maja.

4 Zakres

4.1 Analiza wymagań

System:

- Określenie optymalnego rozmieszczenia emiterów beacon na podstawie planu parkingu;
- Instalacja brokera MQTT i serwera node red;
- Implementacja komunikacji pomiędzy czujnikami zajętości miejsca a brokerem MQTT;
- Stworzenie bazy danych MSSQL;
- Implementacja zapisu statusu czujników do bazy danych za pośrednictwem serwera node-red;
- Implementacja wyzwalacza funkcji MSSQL przeliczającej ilość wolnych miejsc w danych sektorach w przypadku zmiany statusu któregokolwiek z czujników;
- Implementacja odczytu ilości wolnych miejsc w danych sektorach po zmianie statusu któregokolwiek z czujników;
- Wysłanie informacji do monitorów znajdujących się na węzłach komunikacyjnych. Monitory wyświetlają ilość wolnych miejsc w danym sektorze;
- Implementacja komunikacji pomiędzy skanerami przy szlabanach a serwerem.

Użytkownik :

Aplikacja mobilna:

- System lokalizacji;
- System rejestracji;
- System logowania do aplikacji;
- Gotowy system.

System wspomagający parkowanie:

- System zarządzający wolnymi miejscami;
- System sygnałów świetlnych.

System lokalizacji pojazdu na parkingu:

- System nawigacji do samochodu;
- Zapamiętanie lokalizacji samochodu.

Administrator:

- Gotowe rozmieszczenie urządzeń/sprzętu;
- Gotowe rozmieszczenie czujników;
- Gotowe połączenia bezprzewodowe z urządzeniami i serwerem na celu serwisu i identyfikacji wymiany baterii bądź naprawy czujników.

4.2 Wymagania funkcjonalne i niefunkcjonalne

Wymagania funkcjonalne:

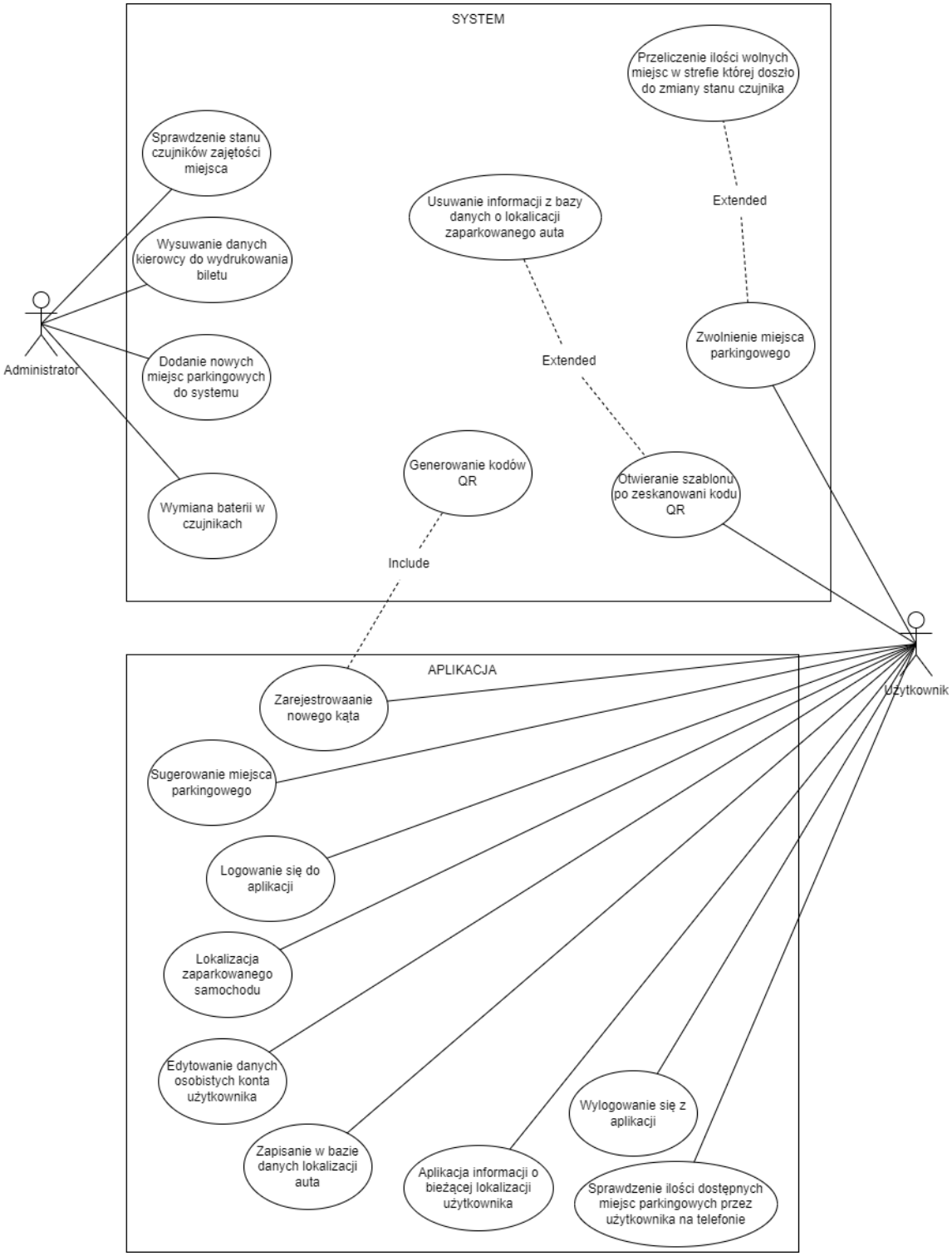
- Czujnik zajętości miejsca umieszczony w nawierzchni wysyła sygnał ultradźwiękowy, na podstawie czasu powrotu sygnału odbitego określa odległość do przeszkody. Odległość 0-40cm uznawana jest jako zajęte miejsce;
- Czujnik łączy się z brokerem MQTT wysyłając swoje id, status i stan naładowania baterii. Na serwerze node red Blok MQTT połączony z odpowiednim topic'iem w brokerze odbiera dane zajętości miejsca. Wysyła je do bloku funkcji, funkcja bazując na danych wejściowych wybiera odpowiednią procedurę modyfikującą status czujnika w bazie danych MSSQL. Funkcja wysyła procedurę do bloku MSSQL;
- Czujnik zajętości miejsca powinien wysyłać stan baterii do serwera MSSQL w przypadku kiedy poziom naładowania baterii spadnie poniżej 10%;
- Administrator za pomocą aplikacji webowej powinien mieć wgląd na status czujników zajętości i ich poziom naładowania baterii;
- Serwer MSSQL powinien przeliczać ilość wolnych miejsc w momencie odebrania informacji o zmianie statusu któregokolwiek z nich;
- Aplikacja mobilna powinna wyświetlać aktualna informacje o ilości wolnych miejsc, która jest na bieżąco aktualizowana z bazy danych MSSQL w momencie ponownego przeliczenia wolnych miejsc;
- Zmiana statusu zajętości któregokolwiek z miejsc spowoduje wywołanie funkcji obliczającej zajętość miejsc w danych sektorach bądź piętrach. Za pomocą protokołu MQTT wysłana zostanie informacja do sterownika WEMOS, który wyświetli ilość wolnych miejsc na tablicy znajdującej się w pobliżu węzłów komunikacyjnych;

- Aplikacja mobilna, za pomocą triangulacji, będzie wyznaczać aktualną pozycję pojazdu w przestrzeni dwuwymiarowej względem nadajników beacon;
- Aplikacja mobilna zapewni możliwość zapisania w bazie danych aktualnej lokalizacji samocho użytkownika poprzez wybranie odpowiedniej opcji, co umożliwi zlokalizowanie zaparkowanego samochodu;
- Aplikacja mobilna poprzez wybór opcji znalezienia samochodu, pobierze koordynaty zaparkowanego samochodu z bazy danych oraz wyznaczy aktualną pozycję użytkownika. Aplikacja wyświetla położenie użytkownika i jego samochodu;
- Aplikacja mobilna, na podstawie id użytkownika, wygeneruje unikalny kod QR, za pomocą którego rejestrowany będzie wjazd oraz wyjazd z parkingu;
- Skaner za pomocą transmisji szeregowej wyśle identyfikator użytkownika do bloku serial na serwerze node-red. Zostanie wywołana funkcja z procedurą SQL zmieniającą status użytkownika oraz zapisującą aktualną datę i godzinę wjazdu i wyjazdu;
- System umożliwi użytkownikom rejestrację oraz logowanie do aplikacji mobilnej.

Wymagania niefunkcjonalne:

- Aplikacja powinna być darmowa;
- Aplikacja powinna mieć intuicyjny i przejrzysty interfejs;
- Aplikacja powinna być kompatybilna z większością urządzeń aby mogło używać jej jak najwięcej użytkowników.

4.3 Diagram przypadków użycia i diagram przepływu (opcjonalny)



Rysunek 1: Diagram przypadków użycia

4.4 Dobór technologii

- Emitery beacon pracujące na protokole BT low energy (Access pointy zintegrowane z emiterami Beacon);
- Ultradźwiękowe czujniki zajętości miejsca (Wi-Fi, MQTT);
- Wyświetlacze;
- Sterowniki do wyświetlaczy (Wemos D1);
- Windows;
- Broker MQTT (RabitMQTT, Mosquito);
- Android Studio(język JAVA);
- Node-RED;
- MSSQL;
- ASP NET 4.8 (C#);
- DevExpress 20.1.6.

5 Scenariusze

Scenariusz: Sugerowanie miejsca parkingowego
Nr: 01
Aktor: Użytkownik aplikacji mobilnej
Stan wejścia:
Warunek: Posiadanie aplikacji, znalezienie się przy wjeździe do parkingu.
Przebieg scenariusza:

1. Użytkownik rejestruje wjazd na parking skanując kod QR aplikacji przy szlabanie, informacja zawierająca ID użytkownika przekazywana jest portem szeregowym do serwera. W bazie danych zmieniany jest status użytkownika;
2. Użytkownik obserwuje monitory znajdujące się na węzłach komunikacyjnych informujące o ilości wolnych miejsc w najbliższym sektorze;
3. Użytkownik parkuje samochód na odpowiadającym mu miejscu. Czujnik zajętości rejestruje zajęcie miejsca parkingowego, wysyła informację o zmianie statusu miejsca na "zajęte" do serwera.

Wynik: Użytkownik bez problemu parkuje swój pojazd i zmieniony zostaje status użytkownika.

Scenariusz: Zwolnienie miejsca parkingowego
Nr: 02
Aktor: Użytkownik aplikacji mobilnej
Stan wejścia:
Warunek: Użytkownik ma już zaparkowane auto na parkingu.
Przebieg scenariusza:

1. Użytkownik opuszcza miejsce parkingowe;
2. Czujnik zajętości rejestruje opuszczenie miejsca parkingowego, wysyła informację o zmianie statusu miejsca na "wolne" do serwera;
3. Użytkownik rejestruje wyjazd z parkingu. Informacja zawierająca Id użytkownika jest przekazywana portem szeregowym do serwera. Wprowadzana jest zmiana statusu użytkownika w bazie danych oraz zerowana jest zapisana lokalizacja samochodu.

Wynik: Użytkownik opuszcza parking.

Scenariusz: Lokalizacja zaparkowanego samochodu
Nr: 03
Aktor: Użytkownik aplikacji mobilnej
Stan wejścia:
Warunek: Użytkownik ma już zaparkowane auto na parkingu oraz ma zainstalowaną aplikację i zapisał w aplikacji miejsce w którym zaparkował.
Przebieg scenariusza:

1. Użytkownik wraca np. z zakupów i zapomniał gdzie zaparkował;
2. Aplikacja pokazuje miejsce gdzie auto zostało zaparkowane i obecną lokalizację Użytkownika;
3. Użytkownik znajduje swój pojazd i wyjeżdza z parkingu.

Scenariusz alternatywny:

1. Użytkownik nie zapisał pkt parkowania przez co aplikacja nie pokazuje lokalizacji;
2. Użytkownik jest zmuszony szukać swojego samochodu bo nie zapisał lokalizacji gdzie zaparkował.

Wynik: Użytkownik znajduje swój pojazd i opuszcza parking.

Scenariusz: Rozładowanie telefonu
Nr: 04
Aktor: Użytkownik aplikacji mobilnej
Stan wejścia:
Warunek: Użytkownik ma już zaparkowane auto na parkingu.
Przebieg scenariusza:

1. Jeśli w czasie np. zakupów użytkownikowi rozładuje się telefon nie będzie on w stanie wyjechać;
2. Jeśli nie jest sam może on zalogowac sie z innego telefonu(osoby towarzyszącej);
3. Użytkownik loguje się z innego telefonu pobiera aplikacje;
4. Po wpisaniu danych do aplikacji loguje się i może wyjechać.

Scenariusz alternatywny 1:

1. Jeżeli osoba jest sama i jeśli posiada ładowarke samochodową ładuje telefon aby móc uruchomić aplikację;
2. Użytkownik wyjeżdza z parkingu.

Scenariusz alternatywny 2:

1. Użytkownik nie posiada ładowarki samochodowej;
2. Użytkownik idzie do budki ochroniarskiej i przekazuje swoje dane do identyfikacji;
3. Użytkownik dostaje bilecik dzięki któremu wyjedzie;
4. Użytkownik wyjeżdża z parkingu.

Wynik: Użytkownik ponownie uzyskuje dostęp do aplikacji / dostaje bilet z kodem QR i opuszcza parking.

Scenariusz: Wymiana baterii w czujnikach
Nr: 05
Aktor: Administrator
Stan wejścia:
Warunek: Rozładowanie baterii w czujnikach.
Przebieg scenariusza:

1. Czujniki na bieżąco wysyłają informacje o stanie baterii do serwera;
2. Administrator odczytuje w aplikacji web poziom naładowania baterii;
3. Administrator wymienia w wymaganych czujnikach baterie.

Wynik: Czujniki działają dalej sprawnie dzięki wymienionym bateriom.

Scenariusz: Dodanie nowych miejsc parkingowych do systemu
Nr: 06
Aktor: Administrator
Stan wejścia:
Warunek: Rozbudowa o nowe miejsca parkingowe.
Przebieg scenariusza:

1. Administrator konfiguruje nowe czujniki i odczytuje UID;
2. Administrator dodaje czujniki do do bazy danych, konfiguruje połączenie MQTT z brokerem, dodaje nowe bloki do node red'a i tworzy zapytania SQL;
3. Administrator zaleca programistom zaktualizowanie aplikacji o nowe miejsca parkingowe.

Wynik: Aplikacja została zaktualizowana, a baza danych zwiększona o nowe rekordy.

Scenariusz: Generowanie kodu QR dla użytkownika
Nr: 07
Aktor: Użytkownik aplikacji mobilnej
Stan wejścia:
Warunek: Tworzenie nowego konta przez użytkownika.
Przebieg scenariusza:

1. Użytkownik wpisuje swoje dane;
2. Aplikacja mobilna generuje kod QR na podstawie UUID użytkownika wygenerowanego w bazie danych podczas tworzenia konta.

Wynik: Aplikacja generuje kod QR.

Scenariusz: Logowanie się do aplikacji
Nr: 08
Aktor: Użytkownik aplikacji mobilnej / webowej
Stan wejścia:
Warunek: Użytkownik posiada już zarejestrowane konto.
Przebieg scenariusza:

1. Użytkownik wybiera opcję logowania;
2. Wpisuje dane logowania;
3. System logowania komunikuje się z bazą danych w celu weryfikacji danych;
4. System logowania otrzymuje potwierdzenie poprawności danych;
5. System logowania przenosi Użytkownika na główną stronę dla zalogowanych.

Scenariusz alternatywny:

1. Użytkownik podał błędne dane logowania / zostawił puste pole;
2. Aplikacja informuje o nieprawidłowych danych i prosi o nie jeszcze raz.

Wynik: Użytkownik loguje się do aplikacji mobilnej / webowej.

Scenariusz: Wylogowanie się z aplikacji
Nr: 09
Aktor: Użytkownik aplikacji mobilnej / webowej
Stan wejścia:
Warunek: Użytkownik posiada już zarejestrowane konto.
Przebieg scenariusza:

1. Użytkownik włącza aplikacje i przechodzi do ustawień gdzie wybiera opcje wylogowania;
2. Użytkownik po wylogowaniu może ponownie się zalogować.

Wynik: Użytkownik wylogował się z aplikacji mobilnej / webowej.

Scenariusz: Zarejestrowanie nowego konta
Nr: 10
Aktor: Użytkownik aplikacji mobilnej
Stan wejścia:
Warunek: Użytkownik ma zainstalowaną aplikacje.
Przebieg scenariusza:

1. Użytkownik wprowadza dane osobiste wymagane do utworzenia nowego konta (login, hasło, imię, nazwisko, adres, e-mail, nr. telefonu);
2. Wygenerowanie nowego kodu QR (scenariusz 7);
3. Aktualizacja bazy użytkowników.

Wynik: Użytkownik ma utworzone nowe konto do korzystania z aplikacji mobilnej.

Scenariusz: Edytowanie danych osobistych konta użytkownika
Nr: 11
Aktor: Użytkownik aplikacji mobilnej / webowej
Stan wejścia:
Warunek: Dane użytkownika nie są już aktualne.
Przebieg scenariusza:

1. Użytkownik wybiera opcję edytowania danych konta;
2. Aplikacja mobilna / webowa pobiera dane z bazy danych;
3. Użytkownik wpisuje aktualne dane osobiste;
4. Po zatwierdzeniu przez użytkownika poprawności danych baza danych zostanie zaktualizowana.

Scenariusz alternatywny:

1. Użytkownik podał błędne dane / zostawił puste pole;
2. Aplikacja informuje o nieprawidłowych danych i prosi o nie jeszcze raz.

Wynik: Użytkownik aktualizuje dane w aplikacji mobilnej / webowej.

Scenariusz: Otwieranie szlabanu po zeskanowaniu kodu QR
Nr: 12
Aktor: Użytkownik aplikacji mobilnej
Stan wejścia:
Warunek: Posiadanie aplikacji mobilnej.
Przebieg scenariusza:

1. Użytkownik podjeżdża pod szlaban otwiera aplikację mobilną;
2. Przybliża telefon do czujnika gdzie jest skanowany kod QR;
3. Po zeskanowaniu kodu szlaban zostaje podniesiony.

Wynik: Użytkownik wjeżdża / wyjeżdża z parkingu.

Scenariusz: Przeliczenie ilości wolnych miejsc w strefie której doszło do zmiany stanu czujnika
Nr: 13
Aktor: Użytkownik aplikacji mobilnej
Stan wejścia:
Warunek: Użytkownik wjeżdża lub odjeżdża z miejsca parkingowego.
Przebieg scenariusza:

1. Czujnik wysyła do serwera status zajętości;
2. Serwer po otrzymaniu informacji inicjuje funkcję wysyłając zapytanie do serwera. Zapytanie mające przeliczyć ilość wolnych miejsc zgodnie z sektorem, w którym znajduje się czujnik;
3. Zapisanie ilości wolnych miejsc zgodnie z sektorem w którym doszło do zmiany;
4. Baz danych zwraca ilość wolnych miejsc w danym sektorze. Serwer wysyła informację o ilości wolnych miejsc do sterownika monitora znajdującego się w danym sektorze.

Wynik: Pomyślnie zostanie zaktualizowana baza zajętości miejsc oraz monitory wyświetlające ilość wolnych miejsc.

Scenariusz: Zapisanie w bazie danych lokalizacji auta (Triangulacja)
Nr: 14
Aktor: Użytkownik aplikacji mobilnej
Stan wejścia:
Warunek: Wybranie opcji zapisania lokalizacji pojazdu.
Przebieg scenariusza:

1. Aplikacja mobilna pobiera dane odległości z 3 najbliższych beaconów;
2. Przy użyciu triangulacji i pobranych danych zostaje wyliczona przybliżona obecna lokalizacja telefonu (pojazdu);
3. Aplikacja przypisuje do danych użytkownika lokalizacje telefonu.

Wynik: Zostaje zapisana lokalizacja telefonu (pojazdu) w bazie danych.

Scenariusz: Aktualizacja informacji o bieżącej lokalizacji użytkownika
Nr: 15
Aktor: Użytkownik aplikacji mobilnej
Stan wejścia:
Warunek: Uruchomienie aplikacji, znajdowanie się na obszarze parkingu.
Przebieg scenariusza:

1. Aplikacja mobilna pobiera dane odległości z 3 najbliższych beaconów;
2. Przy użyciu triangulacji i pobranych danych zostaje wyliczona obecna lokalizacja telefonu (użytkownika);
3. Aplikacja mobilna wyświetla na mapie obecną lokalizację.

Wynik: Wyświetlanie obecnej lokalizacji.

Scenariusz: Usuwanie informacji z bazy danych o lokalizacji zaparkowanego auta (po opuszczeniu parkingu)
Nr: 16
Aktor: Użytkownik aplikacji mobilnej
Stan wejścia:
Warunek: Zeskanowanie kodu QR na wyjeździe.
Przebieg scenariusza:

1. Użytkownik opuszcza parking;
2. Skaner wysyła identyfikator użytkownika do serwera;
3. Na podstawie identyfikatora użytkownika usuwane zostają tymczasowe dane z bazy danych i zmieniony zostaje status użytkownika.

Wynik: Z bazy danych usuwane zostają tymczasowe dane użytkownika (lokalizacja zaparkowanego pojazdu) i zmieniony zostaje status użytkownika.

Scenariusz: Wyszukanie danych kierowcy do wydrukowania biletu (kodu QR użytkownika)
Nr: 17
Aktor: Administrator
Stan wejścia:
Warunek: Użytkownik stracił dostęp do aplikacji mobilnej i nie może opuścić parkingu.
Przebieg scenariusza:

1. Administrator na podstawie danych do identyfikacji od użytkownika szuka jego rekordu w bazie danych;
2. Administrator drukuje bilet z unikalnym kodem QR użytkownika.

Wynik: Użytkownik otrzymuje bilet z kodem QR do opuszczenia parkingu.

Scenariusz: Sprawdzenie ilości dostępnych miejsc parkingowych przez użytkownika na telefonie
Nr: 18
Aktor: Użytkownik aplikacji mobilnej
Stan wejścia:
Warunek: Posiadanie Aplikacji .
Przebieg scenariusza:

1. Użytkownik otwiera aplikację w celu sprawdzenia ilości miejsc na parkingu na danym piętrze;
2. Aplikacja mobilna pobiera dane z bazy danych i wyświetla informacje o wolnych miejscach w strefach na danym piętrze.

Wynik: Użytkownik otrzymuje informacje o ilości wolnych miejsc na danym piętrze.

Scenariusz: Sprawdzenie stanu czujników zajętości miejsca
Nr: 19
Aktor: Administrator
Stan wejścia:
Warunek: Zepsuty czujnik zajętości miejsca.
Przebieg scenariusza:

1. Czujniki wysyłają regularnie informacje o poprawnym działaniu;
2. Administrator sprawdza na aplikacji webowej stan działania czujników;
3. Administrator wymienia czujniki które przestały wysyłać informacje o poprawnym działaniu;
4. Administrator koryguje dane czujnika w bazie danych zgodnie z schematem.

Wynik: Wszystkie czujniki działają poprawnie.

6 Estymacja czasowa

Lp.	Task Name	Duration	Resource Names
1	Zapoznanie się z wymaganiami projektu	3 hrs	Damian;Daniel;Kamil ;Piotr;Kasia
2	Dobór odpowiednich technologii	20 hrs	Daniel;Piotr;Kasia
3	Stworzenie planu parkingu naniesienie oznaczeń miejsc, położenie lamp, ekranów oraz beaconów	24 hrs	Kasia
4	Tworzenie bazy MSSQ oraz wypełnienie jej danymi	8 hrs	Daniel
5	Implementacja komunikacji pomiędzy czujnikiem zajętości miejsca a brokerem MQTT	5 hrs	Piotr
6	Implementacja komunikacji brokerMQTT-NodeRed-serwer MSSQL	2 hrs	Piotr
7	Stworzenie symulacji przepływu danych w systemie przy pomocy serwera NodeRed	16 hrs	Piotr
8	Aplikacja WEB	3 hrs	Damian
9	Rest API	24 hrs	Damian
10	Szkielet aplikacji mobilnej	5 hrs	Kamil
11	Algorytm triangulacji	3 hrs	Kamil
12	Generowanie QR code na podstawie id użytkownika	2 hrs	Kamil
13	Interaktywny plan parkingu w aplikacji	3 hrs	Kamil
14	Skanowanie sygnału beacon, obliczanie odległości	40 hrs	Kamil ;Piotr
15	Połączenie z API aplikacji mobilnej	18 hrs	Kamil; Damian
16	Wyświetlanie informacji na temat zajętości miejsca w aplikacji	2 hrs	Daniel
17	System rejestracji I logowania do aplikacji mobilnej	6 hrs	Kamil
18	Integracja pracy	2 hrs	Damian;Daniel;Kamil ;Kasia;Piotr
19	Testy	5 hrs	Damian;Daniel;Kamil ;Kasia;Piotr
20	Koniec projektu	4 hrs	Damian;Daniel;Kamil ;Kasia;Piotr

Rysunek 2: Estymacja czasowa

7 Implementacja

Implementacja symulacji pracy czujników zajętości miejsca przy pomocy Node -Red i Brokera MQTT. Symulacja pokazuje przepływ informacji oraz zapis do bazy danych na podstawie danych otrzymanych z czujnika. Zasymulowany został również przepływ informacji pomiędzy serwerem mssql a kontrolerami oświetlenia sygnalizacyjnego oraz ekranami wyświetlającymi ilość wolnych miejsc.

Aplikacja Web napisana przy użyciu ASP.NET z biblioteką DevExpress, wykorzystywana do wyświetlania listy miejsc parkingowych, wstawiania nowych, oraz poziomu naładowania czujników zajętości miejsca. Strona pozwala również na sprawdzenie historii zajmowania miejsc parkingowych oraz sprawdzić listę beaconów i wprowadzić nowe beacony do bazy.

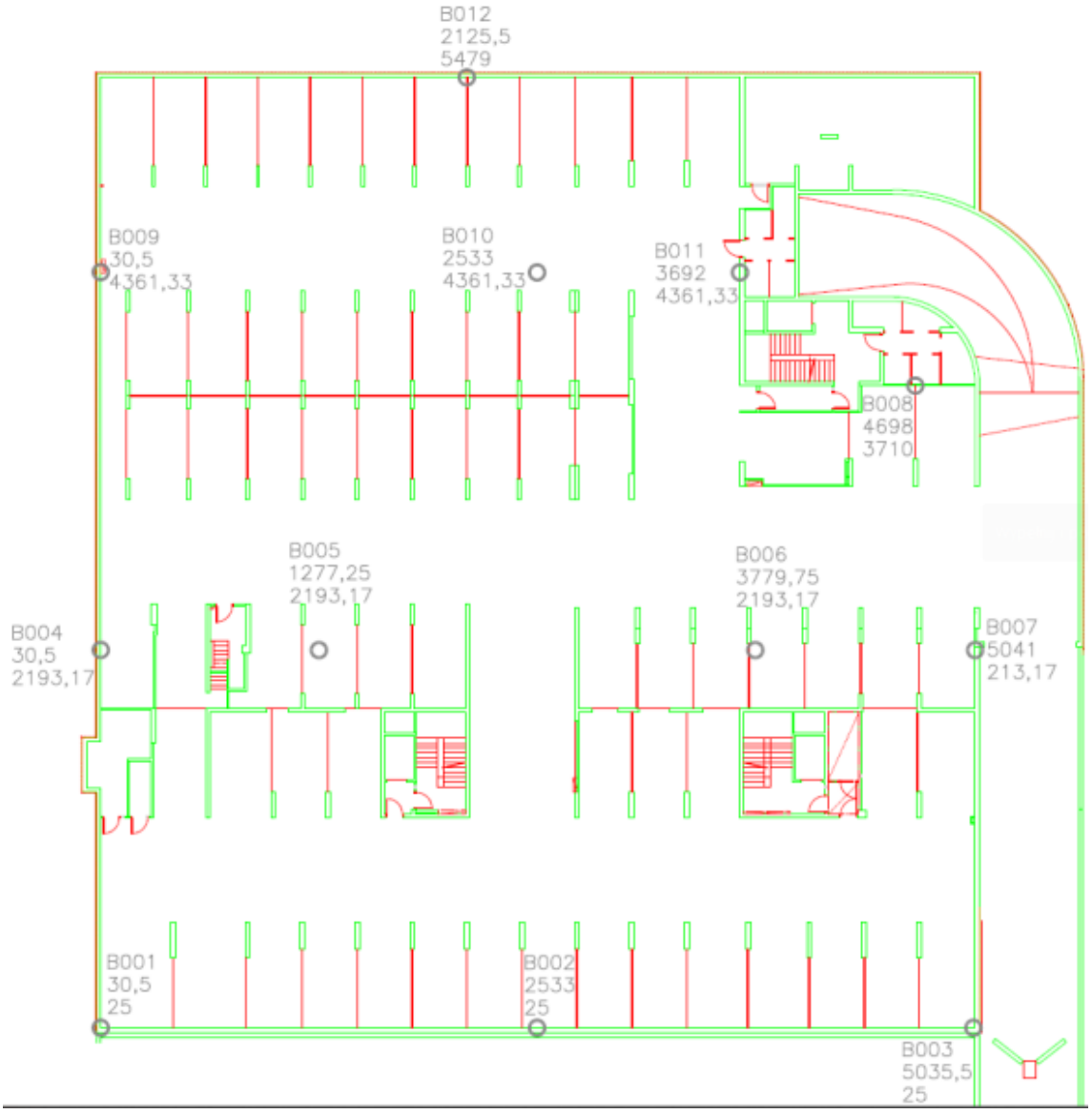
Rest API - Wykonany w Node.JS, pozwala na komunikację aplikacji mobilnej z serwerem SQL. Za pomocą tokenów JSON Web Token jest wykonywana autoryzacja użytkownika, po 5 minutach token jest unieważniany.

Aplikacja Mobilna - Łączy się z serwerem Rest API w Node.JS. Pozwala na zarejestrowanie użytkownika, zalogowanie się, pobranie ile jest wolnych miejsc na parkingu. Wykorzystywana jest do zarejestrowania kiedy użytkownik wjechał na parking, na zapisanie gdzie zaparkowany jest samochód i zapisanie kiedy użytkownik wyjechał z parkingu.

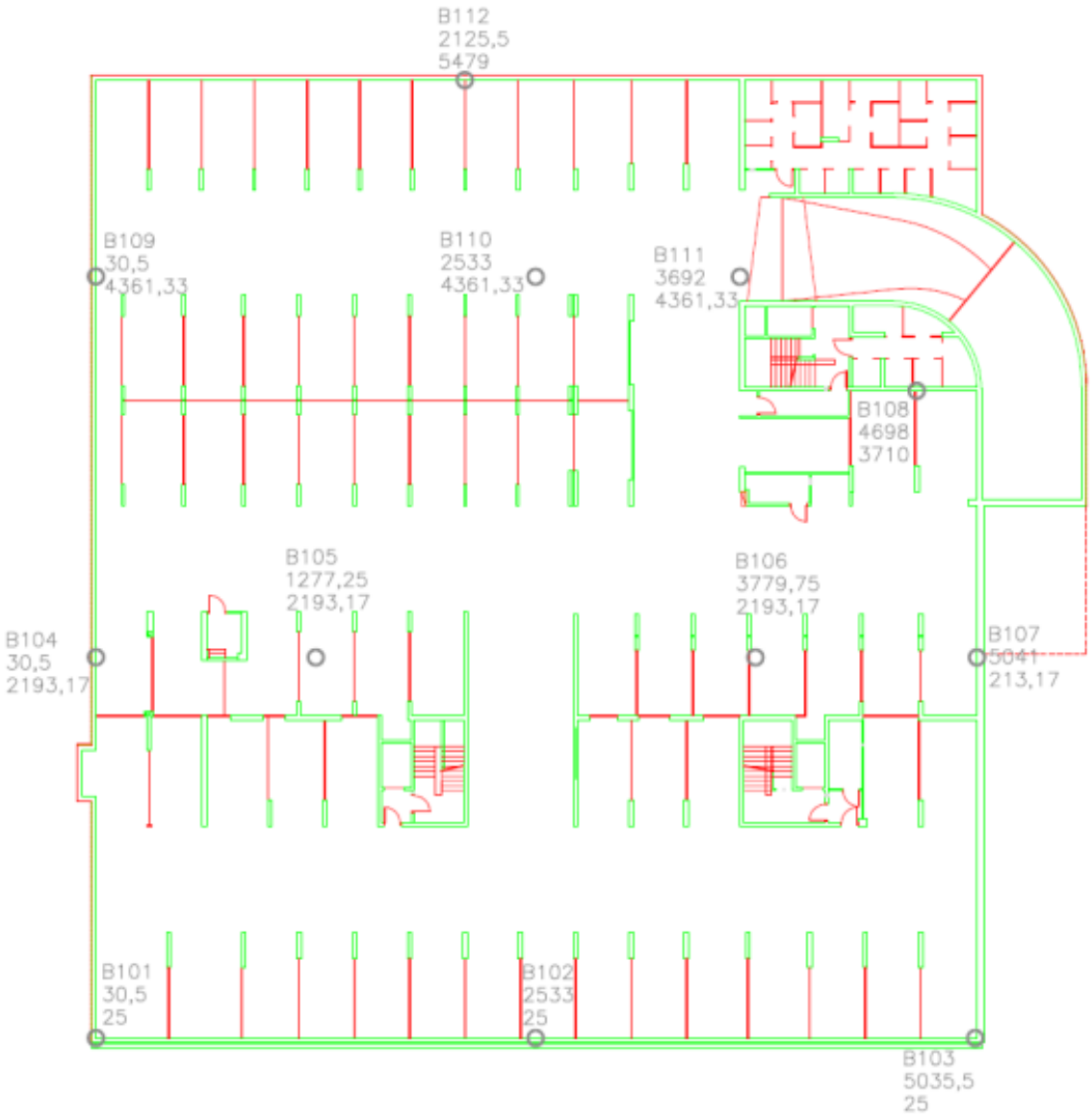
Aplikacja łączy się z beaconami w celu ustalenia lokacji użytkownika i samochodu.

Projekt dwóch pięter parkingu z miejscami parkingowymi.

Beacony naniesione na plan parkingu.

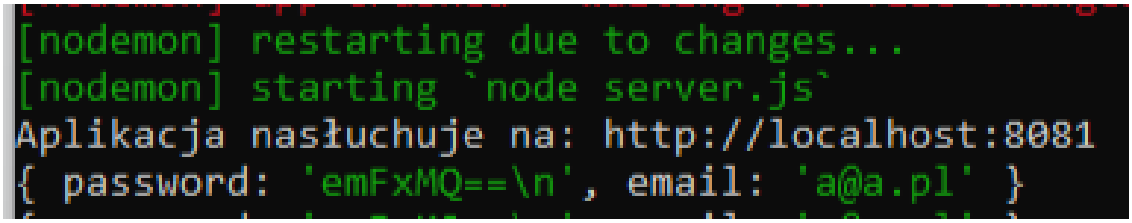


Rysunek 3: Beacony piętro 0



Rysunek 4: Beacony piętro -1

8 Testy i ich wyniki



Rysunek 5: Odbiór danych logowania w body

Rysunek 6: Postman - Odbiór tokena przy logowaniu

9 Podsumowanie i bilans

Podsumowanie:

Założenia projektu zawarte w wymaganiach funkcjonalnych zostały wypełnione min.

- Komunikacja między czujnikami zajętości miejsca a serwerem Ms Sql;
- Aplikacja webowa,system logowania, system zarządzania;
- Aplikacja mobilna, rejestracja, logowanie, wyświetlanie ilości wolnych miejsc, generowanie QR kodu na podstawie ID użytkownika, zapis bieżącej lokalizacji użytkownika, system nawigacji wew. budynkowej na podstawie sygnałów beacon;
- System sygnalizacji na parkingu.

Bilans:

Początkowe założenia projektu zawierały system nawigowania do najbliższego wolnego miejsca parkingowego. Po dokładnej analizie tej funkcjonalności doszliśmy do wniosku że rezygnujemy z tego systemu ze względu na czynnik ludzki (nie zmusimy człowieka do zaparkowania w wyznaczonym miejscu jeśli wcześniej będą wolne miejsca).

Po przeanalizowaniu potrzeb użytkowników w późniejszej fazie projektu, zdecydowaliśmy się na wprowadzenie systemu zapamiętywania lokalizacji zaparkowanego samochodu oraz późniejszej nawigacji do pojazdu.

Pozostałe założenia projektu zostały wykonane w całości.