

1 Teoria

1.1 Free-space path loss

Utrata w sile sygnału spowodowana przejściem fali elektromagnetycznej przez ośrodek (najczęściej powietrze). Wzór na obliczanie FSPL:

$$FSPL = P_{tx} + AG_{tx} + AG_{rx} - P_{rx} - FM - L \quad (1)$$

Gdzie symbole oznaczają:

- P_{tx} - siła trasmitera, wyrażona w dBm
- AG_{tx} - zysk energentyczny anteny trasmitera, wyrażony w dBi
- AG_{rx} - zysk energentyczny anteny odbiorcy, wyrażony w dBi
- P_{rx} - czułość odbiornika, wyrażona w dBm
- FM - margines zaniku sygnału (fade margin) - określa jak duży jest margines różnicy pomiędzy uzyskaną siłą sygnału, a czułością odbiornika
- L - straty wynikające np z oddziaływania innych trasmiterów, przeszkód itp.

Dodatkowo, FSPL można obliczyć, używając następujący wzór:

$$FSPL = 20\log_{10}\left(\frac{d}{d_0}\right) + 20\log_{10}(f) + K \quad (2)$$

Gdzie symbole oznaczają:

- d - dystans dzielący trasmitter od odbiorcy, wyrażony w metrach
- d_0 - dystans referencyjny - w tym wypadku 1 metr
- f - częstotliwość trasmitera - wyrażona w MHz
- K - stała, którą można określić wzorem:

$$K = 20\log_{10}\left(\frac{4\pi d_0}{C}\right) \quad (3)$$

gdzie d_0 to dystans referencyjny (taki sam jak we wzorze wyżej), a C to długość fali emitowanej przez trasmitter

Po przekształceniu wzoru, uzytkujemy:

$$d = 10^{\left(\frac{FSPL - K - 20\log_{10}(f)}{20}\right)} \quad (4)$$

A po połączeniu obu wzorów dostajemy:

$$d = 10^{\left(\frac{P_{tx} + AG_{tx} + AG_{rx} - P_{rx} - FM - L - K - 20\log_{10}(f)}{20}\right)} \quad (5)$$

Dodatkowo, margines zaniku można obliczyć ze wzoru:

$$FM = P_{rs} - P_{rx} \quad (6)$$

gdzie P_{rs} to siła odebranego sygnału.

Po podstawieniu powyższego wzoru do wzoru (5), uzyskujemy

$$d = 10^{\left(\frac{P_{tx} + AG_{tx} + AG_{rx} - P_{rs} - L - K - 20 \log_{10}(f)}{20}\right)} \quad (7)$$

1.2 Zysk energetyczny anteny

Zysk energetyczny anteny jest to stosunek mocy ateny wypromieniowanej w danym kierunku do mocy wypromieniowanej przez antenę wzorcową. Anteną wzorcową może być m.in. antena izotropowa, czyli antena bez fizycznych rozmiarów, która cały sygnał zasilany wysyła we wszystkich kierunkach. W takim wypadku, zysk energetyczny anteny wyrażany jest w dBi .

Na zysk energetyczny mają również wpływ kierunkowość oraz materiał, z którego wykonana jest antena.

1.3 Received signal strength indication

Received signal strength indication (skrótom RSSI) jest to miara określająca moc sygnału odbieranego. Przyjmuje ona wartości niedodatnie (gdzie 0 oznacza sygnał najsilniejszy). Jednostką, w jakiej określa się siłę sygnału jest dBm , która jest logarytmiczną jednostką miary mocy odniesiona do mocy $1mW$.

System Android pozwala na odczytanie siły odbieranego sygnału. Można do tego wykorzystać API *WifiManager* (w przypadku odczytu sygnału WiFi) oraz *BroadcastReceiver* (w przypadku odczytu sygnału Bluetooth).

1.4 Margines zaniku

2 Bluetooth

3 Eksperymenty

3.1 Wykorzystane urządzenia

1. Smartphone Sony Xperia Z1 Compact (D5503) - odbiornik
Dane techniczne:

- Częstotliwość - 2,4GHz
- Przyrost siły sygnału z anteny - 2dBi

2. Router TP-Link TD-W8970 - nadajnik
Dane techniczne:

- Częstotliwość - 2,4GHz

- Dwie zewnętrzne anteny kierunkowe
 - Przyrost siły sygnału z anteny - 4dBi
 - Siła transmitera - 16.5dBm
3. Router TP-Link TL-WA701ND - nadajnik
Dane techniczne:
- Częstotliwość - 2,4GHz
 - Jedna zewnętrzna antena kierunkowa
 - Przyrost siły sygnału z anteny - 2dBi
 - Siła transmitera - 15dBm
4. Smartphone Grand 2 (G7102) - nadajnik
Dane techniczne:
- Częstotliwość - 2,4GHz
 - Jedna antena wbudowana
 - Przyrost siły sygnału z anteny - 0dBi
 - Siła transmitera - 10dBm

3.2 Warunki

Wszystkie pomiary wykonywane były w pomieszczeniu zamknięty, bez przeszkód na drodze sygnału, dlatego jako margines zaniku sygnału została przyjęta wartość 22 dBm. Inne straty (np interferencja sygnałów z routerów) zostały pominięte i ich wykrycie jest jednym z celów eksperymentu.

3.3 Cele

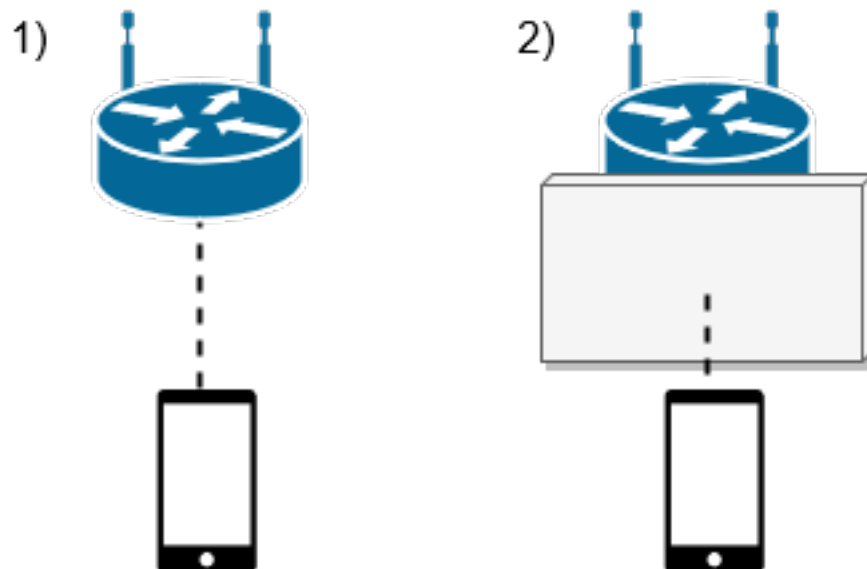
Celem eksperymentu jest ustalenie, jak zmierzona i obliczona, przy użyciu siły sygnałów, odległość między odbiornikiem i transponderami odnosi się do odległości rzeczywistej. Dodatkowo, będę się starał ustalić, jak duży wpływ na jakość sygnału mają przeszkody, kierunek, w jakim skierowane są względem siebie urządzenia oraz interferencja sygnałów.

3.4 Pomiar odległości

Eksperyment polegał na ustawieniu transpondera 1m od odbiornika na jednym poziomie, antenami do siebie. Następnie dodawana była przeszkoda (w tym wypadku książka) i pomiary zostały powtórzone. Eksperyment został wykonany dla wszystkich transponderów.

- Router TP-Link TD-W8970

Rysunek 1: Szkic eksperymentu nr 1



Wersja bez przeszkody:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-41	1,16
2	-40	1,04
3	-37	0,73
4	-42	1,30
5	-37	0,73

Wersja z przeszkodą:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-38	0,82
2	-39	0,92
3	-42	1,30
4	-46	2,07
7	-43	1,46

- Router TP-Link TL-WA701ND

Wersja bez przeszkody:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-44	0,98
2	-44	0,98
3	-45	1,10
4	-47	1,38
5	-45	1,10

Wersja z przeszkodą:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-49	1,74
2	-47	1,38
3	-46	1,23
4	-47	1,38
5	-47	1,38

- Samsung Grand 2

Wersja bez przeszkody:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-51	1,38
2	-50	1,23
3	-49	1,10
4	-48	0,98
5	-53	1,74

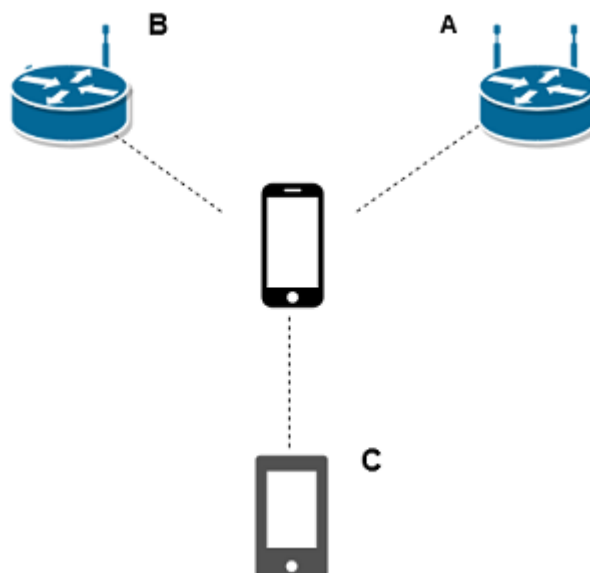
Wersja z przeszkodą:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-50	1,23
2	-54	1,95
3	-53	1,74
4	-55	2,19
5	-55	2,19

3.5 Pomiary zakłóceń

Eksperyment polegał na rozmieszczeniu trasmiterów na wierzchołkach trójkąta, w środku którego znajdował się odbiornik. Wszystkie urządzenia znajdowały się na tej samej wysokości. Mierzone były zmiany siły sygnału i obliczonej odległości w zależności od kąta położenia odbiornika w stosunku do trasmitera oraz ilości nakładających się na siebie sygnałów. Na początku, włączony był tylko trasmiter o indeksie A. Odbiornik znajdował się w stosunku do trasmitera

Rysunek 2: Model systemu do pomiaru zakłóceń



pod kątem około 50 stopni. Następnie włączony został transponder B. Na końcu do modelu został dodany transponder C.

Informacje o urządzeniach:

- Transponder A - TP-Link TD-W8970, współrzędne (1.80, 0)
- Transponder B - TP-Link TL-WA701ND, współrzędne (0, 0)
- Transponder C - Samsung Grand 2, współrzędne (1.07, 1.8)
- Odbiornik - współrzędne (1.2, 0.45)

Pomiar bez zakłóceń dla odległości 80cm przy kącie 50°:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-41	1,16
2	-43	1,46
3	-41	1,16
4	-42	1,30
5	-43	1,46

Pomiar z zakłóceniami z transmitera B dla odległości 80cm przy kącie 50°:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-44	1,64
2	-47	2,31
3	-45	1,84
4	-48	2,60
5	-47	2,31

Pomiar z zakłóceniami z obu transmiterów dla odległości 80cm przy kącie 50°:

Pomiar	Siła sygnału (w dBm)	Obliczona odległość (w metrach)
1	-48	2,60
2	-47	2,31
3	-44	1,64
4	-48	2,60
5	-46	2,06

3.6 Wyznaczanie lokalizacji użytkownika

Narazie mało do napisania. Z dwóch pomiarów dla modelu z góry, dostałem lokalizacje (-0,4; 1,2; -0,3) oraz (1,5; 1,67; -1,5).

4 Model wyznaczania lokalizacji

Stworzyłem model algorytmu w MatLabie. Nie wyobrażam sobie modelu w trzech wymiarach i z kolorem (według mnie wynikiem będzie prostopadłościan o granatowym dominującym kolorze ścian) , dlatego stworzyłem model 2D, który jest tak naprawdę przekrojem modelu 3D (płaszczyznę XY). Algorytm zmieniłem według zastrzeżeń Pana Doktora. Prostopadłościan, który zawiera w sobie "sfery" ruterów, dzielony jest na max 20 kawałków. Wyznacza się najlepszą pozycję, dla niej brane są sąsiadujące pozycje i uzyskany sześciąt dzieli się na 9 kawałków i ponownie wyznacza najlepszą pozycję. Obliczenia kończą się, jak spełnione jest założenie: $szerKawalka \leq okreslonaDokladnosc$.

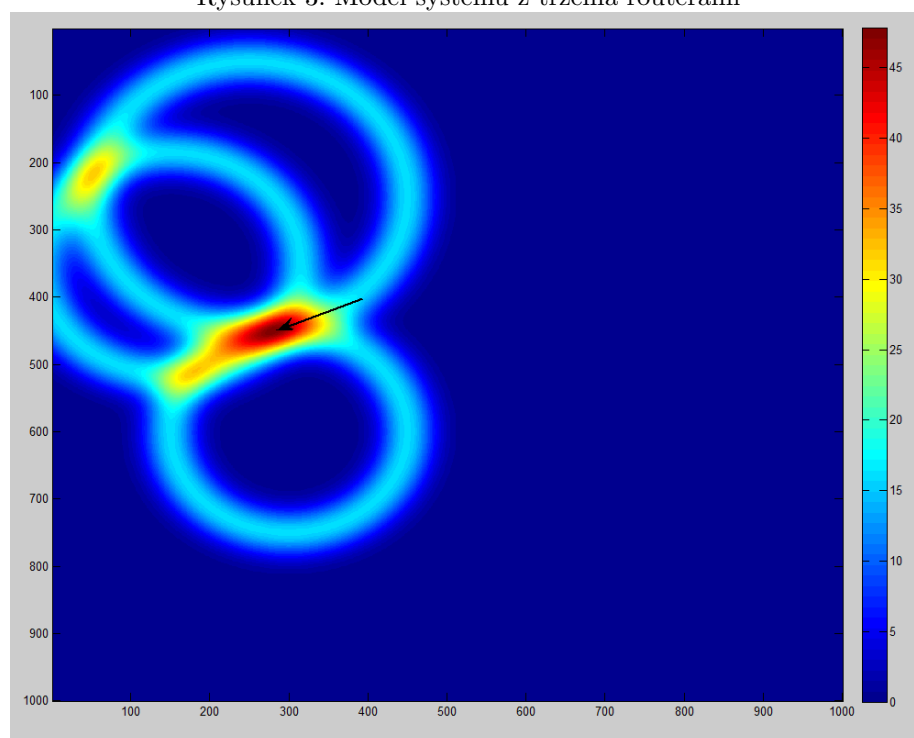
5 Implementacja

5.1 Serwer

Serwer jest aplikacją, która ma analizować zabrane dane, sterować urządzeniami zewnętrznymi oraz stanowić most pomiędzy klientem administracyjnym, a aplikacjami mobilnymi. Na zadania aplikacji serwerowej składają się:

- Zbieranie danych z aplikacji mobilnych oraz wyznaczanie lokalizacji użytkowników

Rysunek 3: Model systemu z trzema routerami



- Zezwalanie użytkownikowi administracyjnemu na konfigurację systemu poprzez żądań aplikacji klienckiej
- Sterowanie i zarządzanie urządzeniami zewnętrznymi analizując dane odebrane z aplikacji mobilnych
- Dostarczanie, w formie skumulowanej lub w czasie rzeczywistym, danych na temat położenia użytkowników do analizy i monitoringu

5.1.1 Lokalizacja użytkowników mobilnych

Pobranie i analiza danych Serwer pobiera dane od użytkowników w formie requestów HTTP. Każde żądanie wysłane do serwera musi posiadać adres MAC urządzenia Bluetooth wysyłającego oraz listę zarejestrowanych sygnałów. Każda pozycja na liście sygnałów musi posiadać nazwę urządzenia, którego sygnał został odebrany (w przypadku sygnału wysłanego przez router jest to SSID sieci, zaś w przypadku urządzeń Bluetooth jest to adres MAC), typ sygnału (WIFI albo Bluetooth) oraz zarejestrowaną siłę sygnału, określoną w dBm.

Następnie, dla każdego elementu z listy dociągane są stałe dane zarejestrowane w systemie - lokalizacja oraz waga sygnału. Dane na temat routerów oraz stałych urządzeń Bluetooth (np. Beaconów) pobierane są bazy danych. Jeżeli jakiś sygnał Bluetooth nie widnieje w bazie danych, sprawdzane są ostatnie żądania od urządzeń mobilnych, dla których udało się określić lokalizację, a czas od ostatniej aktualizacji nie jest większy niż 4 sekundy. Jeżeli sygnał pochodzi od jednego z tych urządzeń, potrzebne informacje pobierane są z dynamicznie budowanej, lokalnej bazy wiedzy. Jeżeli sygnał nie figuruje ani w bazie danych, ani w bazie dynamicznej, zostaje uznany jako sygnał przypadkowy i odrzucony. Waga sygnału przyjmuje wartości w skali od 1 do 4. Domyślnie, sygnałowi pochodzącemu od routera WiFi nadawana jest waga 3, sygnałowi ze stałego urządzenia Bluetooth waga 2, zaś sygnały pochodzące od innych użytkowników mobilnych wagę 1. Wagę stałych urządzeń WiFi i Bluetooth można edytować przy użyciu panelu konfiguracyjnego w aplikacji klienckiej. W ostatnim kroku, dla każdego zarejestrowanego sygnału obliczana jest odległość urządzenia od użytkownika. Wykorzystywana do tego jest lokalizacja użytkownika, wzór na Free-space path loss oraz dane statyczne (jak siła anten, siła trasmitera itp.).

Algorytmiczne wyznaczenie lokalizacji Celem algorytmu jest wyznaczenie punktu, dla którego suma prawdopodobieństw wynikających z odległości użytkownika od urządzenia, jest największa.

Dane pobrane od użytkownika, uzupełnione o statyczne dane przechowywane w systemie, przekazane są do sekcji napisanej w języku F#. Na wstępie, do każdego zarejestrowanego algorytmu zostaje przypisana probabilistyczna Guassa, określająca prawdopodobieństwo znalezienia się użytkownika w danym punkcie w przestrzeni, gdzie stała μ przyjmuje wartość równą dystansowi obliczonemu na podstawie siły sygnału urządzenia. Dzięki takiemu podejściu, każdy sygnał można zwizualizować jako sferę, której powierzchnia zbliżona jest do chmury.

Największe zagęszczenie prawdopodobieństwa występuje dla średnicy równej odległości obliczonej z siły sygnału, a która rzadnie zbliżając się i oddalając od środka sfery.

Pierwszym krokiem algorytmu jest wyznaczenie prostopadłościanu, dla którego wykonywane będą obliczenia. Wielkość bryły dobrana jest tak, aby wewnątrz niej znalazły się wszystkie sfery sygnałów (przy uwzględnieniu zapasu równego 2σ). Następnie prostopadłościan oraz sfery są normalizowane w taki sposób, aby początek układu zaczął się w punkcie (0,0,0), zaś wszystkie wartości współrzędnych przyjmowały tylko wartości nieujemne. Celem takiej operacji jest uproszczenie algorytmu oraz wyzbycie się potrzeby skalowania iteratorów oraz odnośników do elementów w tablicach.

Kolejnym krokiem algorytmu jest podział prostopadłościanu na części, dla których liczona będzie suma prawdopodobieństw. Celem tego kroku jest podział pola działania na jednakowe sześciany w taki sposób, aby w żadnym wymiarze ilość sześciąt nie przekroczyła 100. Aby to uzyskać, najdłuższy bok prostopadłościanu zostaje podzielony na 100 części. Następnie boki w pozostałych 2 wymiarach zostają podzielone na sześciany o krawędziach równej długości. Dzięki takiemu zabiegowi dzieli się prostopadłościan na sześciany. Podział prostopadłościanu na sześciany pozwala na wyeliminowanie błędów obliczeniowych wynikających z nierealistycznego podziału pola oliczeń.

Kolejnym krokiem algorytmu jest wyliczenie sumy prawdopodobieństw ze wszystkich sfer sygnałów dla każdego sześcianu w prostopadłościanie. Każde prawdopodobieństwo, będące składową sumy, przemnażane jest przez wagę danego sygnału.

$$P(x, y, z) = \sum_{r=1}^{r=R} w_r * \frac{1}{\sigma\sqrt{2\pi}} e^{\left(\frac{-(d-\mu)^2}{2\sigma^2}\right)} \quad (8)$$

gdzie:

- x, y, z - współrzędne konkretnego sześcianu
- R - ilość routerów w modelu
- w_r - waga konkretnego routeru
- σ - odchylenie standardowe, w naszym modelu wynosi: długość najdłuższego boku prostopadłościanu podzielona przez 30
- μ - średnia, w naszym modelu jest to wartość równa, obliczonej na podstawie siły sygnału, odległości użytkownika od źródła sygnału
- d - odległość Euklidesowa pomiędzy sześcianem, a routerem

Następnie wybierany jest sześcian, dla którego suma prawdopodobieństwa jest najwyższa. Jeżeli długość boku sześcianu jest równa lub mniejsza od naszego przybliżenia, współrzędna sześcianu staje się lokalizacją naszego użytkownika. Jeżeli długość boku sześcianu jest większa od naszego przybliżenia, dla wybranego sześcianu dobierane są jego sześciany sąsiednie. Następnie wybrane 27

sześcianów staje się nowym modelem obliczeniowym. Każdy wymiar nowego pola dzielone jest na 9 równych części, dzięki czemu uzyskuje się 729 sześcianów. Algorytm zostaje powtórzony, aż lokalizacja nie zostanie określona z interesującym nas przybliżeniem. Sześcian o największej sumie prawdopodobieństw staje się lokalizacją użytkownika.

Ostatnim krokiem algorytmu jest przeliczenie obliczonej lokalizacji przy użyciu danych uzyskanych podczas normalizacji, aby obliczona lokalizacja odpowiadała lokalizacji dla danych wejściowych.

Zarządzanie lokalizacją użytkownika Po pozytywnym obliczeniu lokalizacji, zostaje ona zapisana w bazie danych, aby potem mogła być użyta do wyświetlenia skumulowanej mapy przepływu użytkowników albo do sterowania urządzeniami. Następnie, lokalizacja zostaje asynchronicznie wysłana do wszystkich klientów administracyjnych, którzy zarejestrowali swoją chęć pobierania danych w trybie real-time (w czasie rzeczywistym). Następnie, lokalizacja wraz z adresem MAC użytkownika zostaje przekazana do wątków urządzeń zewnętrznych, które wykorzystują te dane do podjęcia decyzji o wywołaniu przypisanego urządzeniowi eventu. W ostatnim kroku, lokalizacja zostaje dodana (lub podmieniona, jeżeli wpis o danym użytkowniku już istnieje) w bazie dynamicznej, aby ta informacja mogła posłużyć przy wyznaczaniu lokalizacji innych użytkowników.

5.1.2 Sterowanie i zarządzanie urządzeniami

Serwer, poza pobieraniem i analizą danych, zajmuje się również sterowaniem przydzielonymi mu urządzeniami. Sterowanie urządzeniami odbywa się na dwa sposoby:

- Przy użyciu eventów - urządzenia mogą mieć przypisaną klasę obsługującą zdarzenia specjalne. Mechanizm analizuje w czasie rzeczywistym ostatnie zarejestrowane lokalizacje użytkowników, a następnie na ich podstawie, oraz na podstawie wcześniej przypisanych sobie reguł decyduje, czy mają zostać podjęte jakieś działania. Jeżeli tak, komunikuje się z urządzeniem, aby ustawić mu wyznaczone parametry. Wywołana metoda zwraca zmienną typu Boolean, określającą, czy została podjęta decyzja o komunikacji z urządzeniem.
- Co określony interwał czasowy - mechanizm, który co jakiś określony czas oblicza parametry, jakie należy przekazać do urządzenia. Robi to na podstawie zarejestrowanych przez ten czas pozycji użytkowników. Do swoich obliczeń wykorzystuje wagi użytkowników, przypisane im w panelu administracyjnym. Mechanizm czasowy ma niższy priorytet niż mechanizm eventów, dlatego jeżeli klasa obsługująca zdarzenia przekazała urządzeniu parametry, system czasowy zostaje zawieszony aż do kolejnego przejścia pętli. Takie rozwiązanie zapobiega nadpisywaniu parametrów wysłanych do urządzenia, zanim wcześniejsze zostaną wykorzystane.

Informacje na temat urządzeń przechowywane są w bazie danych. Danymi, które są potrzebne do sterowania urządzeniem, niezależnie od jego typu są:

- Jego lokalizacja (określona przez 3 współrzędne)
- Ip urządzenia oraz port, na którym nasłuchuje
- Nazwę rozpoznawalną przez użytkownika (np żarówka na korytarzu)
- Sterownik określający sposób komunikacji, implementujący interfejs napisany do konkretnego typu urządzenia
- Moduł określający, czy dla danego urządzenia przypisane jest jakieś sterowanie eventami (np natychmiastowa zmiana siły oświetlenia, spowodowana zbliżeniem się określonego użytkownika)
- Flaga określająca, czy dane urządzenie jest aktywne

Dobór parametrów sterujących Dla każdego aktywnego urządzenia zapisanego w systemie uruchomiony jest na serwerze osobny wątek sterujący. Taki sposób pracy został przyjęty, aby sterowanie i obsługa eventów odbywała się płynnie i w równy sposób dla wszystkich urządzeń, a błąd czy problemy komunikacyjne jednego z urządzeń nie miały wpływu na inne. Do każdego wątku sterującego przypisany jest obiekt klasy zawierającej wszystkie potrzebne informacje oraz metody, aby kontrolować dany typ urządzenia (dla oświetlenia jest to klasa `LightDeviceControllingThread`).

Praca wątku polega na wywołaniu metody `StartControll()`, na której ciało składa się nieskończona pętla `while`. Niezależnie od typu urządzenia, którym zarządza wątek, każda klasa posiada obiekt kolejki, do której kontroler przyjmujący dane od urządzeń mobilnych wstawia obliczone lokalizacje (pod warunkiem, że dla urządzenia sterowanego przypisano klasę obsługującą zdarzenia).

Kolejnym krokiem, jaki wykonywany jest w metodzie, jest pobranie z kolejki wszystkich oczekujących lokalizacji, a następnie przekazanie ich do klasy sterującej eventami. W systemie zaimplementowana jest przykładowa klasa sterująca światłem - `ImportantUserFirstContr`. Wybiera z listy użytkowników, którzy są w bliskiej odległości od źródła światła, a następnie wyszukuje wśród nich użytkowników o najwyższej wadze. Jeżeli taki użytkownik zostanie znaleziony, system wysyła do urządzenia rozkaz ustawienia światła o największej mocy. Taki poziom zostanie utrzymany aż ostatni użytkownik uprzywilejowany nie oddali się od źródła światła. System poczeka wtedy dodatkowe 4 sekundy, a następnie wyśle do urządzenia informację o ustawieniu mocy światła na poprzednią wartość. System pozwala na rozszerzanie klas sterujących oraz tworzenie swoich i przypisywanie ich urządzeniom (o ile odpowiedni wpis zostanie dodany do bazy danych).

Jeżeli moduł zarządzania eventami zadecyduje o wysłaniu do urządzenia parametrów ustawiających, wykonuje się moduł zarządzania czasowego. Przed wykonaniem obliczeń, algorytm sprawdza, czy od ostatniej czasowej aktualizacji

minął odpowiedni okres czasu (domyślnie algorytm ma się wykonywać co godzinę). Data poprzedniej aktualizacji przechowywana jest w polu LastUpdate, które jest aktualizowane obecną datą za każdym razem, jak algorytm pozytywnie wyliczy parametry dla urządzenia. Jeżeli odpowieni czas minął, program pobiera z bazy wszystkie pozycje użytkowników, które zostały zarejestrowane w okresie od ostatniej aktualizacji. Następnie, algorytm wybiera z pobranej listy użytkowników, którzy zostali zarejestrowani w odległości nie większej niż, domyślnie, 5 metrów, a następnie sumuje ich wagi. Parametr, jaki ma zostać wysłany do urządzenia jest liczony na podstawie wzoru:

$$P_{ustalona} = (P_{max} - P_{min}) * \frac{N_{blisk}}{N_{og}} + P_{min} \quad (9)$$

gdzie zmienne w równaniu oznaczają:

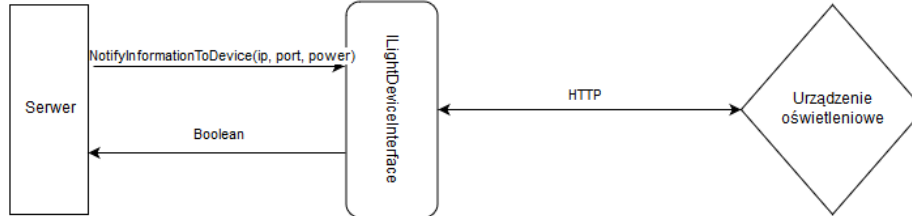
- $P_{ustalona}$ - moc światła jaka ma być ustawiona dla urządzenia
- P_{min} - minimalna moc światła ustawiona dla urządzenia
- P_{max} - maksymalna moc światła ustawiona dla urządzenia
- N_{blisk} - suma wag użytkowników, którzy zostali zarejestrowani wystarczająco blisko urządzenia
- N_{og} - suma wag wszystkich użytkowników w danym okresie czasowym

Wysyłany parametr jest ustawiany dodatkowo w zmiennej PreviousStaticPowerLevel, aby mógł być potem, w ramach potrzeby, wykorzystany przez moduł sterujący eventami. Na końcu, do zmiennej przechowywującej ostatnią aktualizację zostaje przypisana obecna data i godzina.

Jeżeli parametry urządzenia zostały zaktualizowane w obecnym przebiegu algorytmu, ponowne wykonywanie algorytmu rozpoczyna się od razu. Jeżeli nie, wątek zostaje uśpiony na 0,1 sekundy, aby nie zurzywać niepotrzebnie zasobów serwera.

Komunikacja z urządzeniami Sposób komunikacji serwera z urządzeniem nie jest zapisany bezpośrednio w kodzie serwera, ale jest przechowywany w bazie danych i przypisywany indywidualnie do konkretnego urządzenia. Może zostać zmieniony przy użyciu panelu administracyjnego. Dla każdego typu urządzenia może być zdefiniowana pewna pula modułów komunikacji z urządzeniami. Każdy typ ma określony interfejs, po którym ma dziedziczyć klasa komunikująca się z tego typu urządzeniami. Wewnątrz modułu komunikacyjnego określony jest format wysyłanej wiadomości, kodowanie oraz kolejność parametrów. Ilość metod, które ma udostępniać klasa implementująca interfejs jest zależna od ilości parametrów, jakie mogą być ustawiane w danym typie urządzeń.

Rysunek 4: Model komunikacji z urządzeniami oświetleniowymi



W systemie został stworzony interfejs `ILightDeviceInterface`. Powinien być implementowany przez moduły do komunikacji z urządzeniami odpowiedzialnymi za oświetlenie. Z racji tego, iż światło posiada tylko jeden sterowalny parametr - moc światła, interfejs narzuca na klasie zaimplementowanej metody `NotifyInformationToDevice`, która jako parametry przyjmuje ip urządzenia, jego port oraz wartość mocy światła, jaka ma być wysłana do urządzenia. Zadaniem implementacji tej metody jest sformatowanie wiadomości, zawierającej ustawianą moc światła, według standardu urządzenia oraz wysłanie jej protokołem obsługiwanym przez urządzenie. Metoda ma zwrócić zmienną typu `Boolean`, informującą o tym, czy komunikacja i wysłanie wiadomości się powiodło. W ramach przykładu, stworzony został moduł do komunikacji z urządzeniami oświetleniowymi, `LoggerLightDeviceInterface`. Wysyła on protokołem HTTP na podany adres Ip wiadomość o formacie *"Należy ustawić wartość na: P_{swiatla} "*, gdzie P_{swiatla} to moc światła, którą chcemy ustawić.

5.1.3 Konfiguracja systemu

Aplikacja serwerowa pozwala na konfigurację elementów systemu bez potrzeby jego restartowania. Serwer wykorzystuje do tego kontroler `ConfController`, przyjmujący żądania HTTP. Podstawowymi elementami, które mogą być skonfigurowane w taki sposób są:

- Wielkość pola operacyjnego - wartość określająca, jak duży jest obszar, na którym zainstalowany jest system. Wewnątrz tego pola muszą znajdować się wszystkie obsługiwane routery i urządzenia Bluetooth, które nie zmieniają swojej pozycji (np. Becony). Jednostką, w jakiej zapisana jest wielkość mapy, są metry. Wartość ta wykorzystywana jest podczas wyświetlania lokalizacji użytkowników w aplikacji administracyjnej - współrzędna lokalizacji zostaje przeskalowana przy użyciu wielkości pola w taki sposób, aby realistycznie odzorowywała położenie na mapie wyświetlanej w aplikacji administracyjnej.
- Dane o routerach Wifi i nieporuszających się urządzeniach Bluetooth - administrator systemu ma możliwość zmiany kluczowych danych na temat urządzeń, na podstawie których lokalizowani są potem użytkownicy. Poza danymi technicznymi, dotyczącymi specyfikacji urządzenia (siła nadajnika, siła anten, straty wynikające z kształtu obudowy - np. antena wewnętrzna), zdefiniowane są również:

- typ sygnału wysyłanego przez urządzenie - Wifi albo Bluetooth
- nazwa, po której urządzenie jest rozpoznawalne przez aplikację mobilną - SSID w przypadku routera WiFi, Mac w przypadku urządzenia Bluetooth
- lokalizacja urządzenia - wymagane jest podanie lokalizacji w trzech wymiarach. Współrzędne urządzenia, jak wszystkie współrzędne w systemie, są podawane stosunku punktu określonego jako początek pola działania systemu.
- waga urządzenia, która uwzględniana jest podczas lokalizowania użytkownika. Wartość ta podczas dodawania nowego urządzenia, inicjalizowana jest wartościami domyślnymi, które potem można zmienić. Dopuszczalna wartość wagi jest liczbą całkowitą z zakresu $< 1 : 4 >$. Domyślne wartości to:
 - * dla urządzeń Bluetooth, dynamicznie zmieniających swoją pozycję (użytkownicy aplikacji mobilnej) - 1
 - * dla urządzeń Bluetooth, których lokalizacja jest określona - 2
 - * dla urządzeń Wifi - 3

Dane na temat routerów przechowywane są w bazie danych, skąd są pobierane podczas włączania się aplikacji serwerowej. Informacje te są również przechowywane lokalnie, w pamięci serwera. Ma to na celu zmniejszenie czasu wykonywania operacji, ponieważ pobieranie danych z bazy danych jest dużo wolniejsze niż korzystanie ze zmiennych zapisanych w pamięci. Dlatego, każdorazowa zmiana informacji o routerach powoduje zmianę danych przechowywanych lokalnie przez serwer. System zakłada, że dane o routerach dokonywane są tylko przy użyciu aplikacji serwerowej - zmiany dokonane na bazie przez zewnętrzne aplikacje (np GUI bazy danych) nie zostaną uwzględnione przez serwer aż do jego restartu.

- Dane na temat urządzeń sterowanych - podobnie jak w przypadku urządzeń do lokalizacji użytkowników, sterowane urządzenia mogą być dodawane, edytowane oraz usuwane przez administratorów systemu. Informacje różnią się w zależności od kategorii, do której należy dane urządzenie, jednak część z nich jest wspólna:
 - nazwa - nazwa urządzenia, która ma pozwolić na jego identyfikację przez użytkowników systemu
 - ip i port - podstawowe dane wykorzystywane do komunikacji z urządzeniem
 - lokalizacja - położenie urządzenia określone przez trzy współrzędne
 - moduł eventów - nazwa modułu, który ma obsługiwać wydarzenia związane z danym urządzeniem. Do urządzenia może nie być przypisany żaden moduł eventów
 - moduł komunikacyjny - nazwa modułu, który ma pozwolić na komunikację serwera z urządzeniem

- czy moduł jest aktywny - informacja ta jest wykorzystywana podczas sterowania urządzeniem

Informacje na temat sterowanych urządzeń, poza wpisami w bazie danych, przechowywane są również w pamięci serwera. W przeciwieństwie do routerów (których dane przechowywane są statycznie w klasie SystemDataKnowledge), dane na temat urządzeń znajdują się w przypisanych im obiektach sterujących. Jest to związane z tym, że wątki sterujące są jedynymi elementami aplikacji serwerowej, które często korzystają z tej informacji.

Z racji tego, iż z każdym sterowanym urządzeniem, zarządzanym przez system, związany jest obiekt sterujący, kontroler konfiguracyjny odpowiedzialny jest również za edycję wątków sterujących. W przypadku dodawania do systemu nowego sterowanego urządzenia, kontroler inicjalizuje nowy obiekt sterujący oraz startuje związany z nim wątek. W przypadku usuwania urządzenia, kontroler zatrzymuje wątek sterujący i usuwa jego obiekt z listy obiektów sterujących.

- Dane na temat użytkowników systemu - kontroler konfiguracyjny pozwala na edycję wagi użytkownika oraz jego nazwy. System rozpoznaje użytkowników po adresie MAC - zmiana nazwy użytkownika ma na celu jedynie pomóc osobie korzystającej z systemu łatwiejsze rozpoznawanie użytkowników. Waga używana jest przy sterowaniu urządzeniami - użytkownik o wadze 4 (przykładowo szef) jest ważniejszy z punktu widzenia modułu określający parametry urządzeń, niż osoba o wadze 1 (zwykły pracownik). Dodatkowo, waga może mieć wpływ na moduły obsługi zdarzeń, przypisany do urządzenia - światło urządzenia może być ustawiane wartością maksymalną w przypadku pojawienia się w okolicy użytkownika o wadze 4.

Dane o użytkownikach przechowywane są w bazie - nie ma potrzeby przechowywania lokalnej kopii.

Poza konfiguracją systemu, kontroler dostarcza aplikacji administracyjnej danych, potrzebnych do dokonywania zmian w systemie oraz poprawnego działania aplikacji klienckiej. Możliwe jest pobranie danych na temat:

- routerów
- typach sygnałów routerów
- sterowanych urządzeniach
- kategoriach sterowanych urządzeń
- dostępnych modułów sterujących eventami
- dostępnych modułów komunikacyjnych
- wielkości pola działania systemu

- użytkownikach

Dane, w zależności od rodzaju, pobierane są z bazy danych lub lokalnej pamięci serwera.

5.1.4 Zarządzanie pozostałymi funkcjami aplikacji administracyjnej

Do obsługi pozostałych, poza konfiguracją, żądań aplikacji klienckiej służy kontroler ClientController. Zawiera on metody pozwalające na:

- logowanie i wylogowywanie się użytkownika - użytkownika w momencie włączania aplikacji zmuszony jest do zalogowania się. W momencie logowania, na serwer zostaje wysłane żądanie zawierające jego login. Następnie wysłany login zostaje powiązany z adresem ip, z którego przyszło żądanie. Jest to robione w celu umożliwienia późniejszego wysyłania informacji do klienta bez potrzeby wysłania przez niego żądania - wykorzystywane w momencie, w którym użytkownik zadeklaruje chęć monitorowania użytkowników mobilnych.
- pobranie informacji o sterowanych urządzeniach zarejestrowanych w systemie. Ta metoda różni się od metody pobrania urządzeń z kontrolera konfiguracyjnego - lista urządzeń jest pobierana w celu wyświetlenia na mapie. Dodatkowo, pobierane są wszystkie przyrządy, bez względu na ich kategorię. Jedynymi zwracanymi informacjami są lokalizacja urządzenia (w celu wyświetlenia w odpowiednim miejscu) oraz jego kategoria (aby aplikacja kliencka mogła dobrać odpowiednią ikonkę).
- zgłoszenie żądania zapisania użytkownika do subskrypcji lokalizacji użytkowników - system zostaje poinformowany, że dany administrator chce monitorować wszystkich pojawiających się użytkowników aplikacji mobilnej. W momencie określenia lokalizacji użytkownika, do wszystkich użytkowników, na adres ip pobrany podczas logowania, zostaje wysłana notyfikacja zawierająca współrzędne użytkownika oraz jego adres MAC.
- pobranie lokalizacji użytkowników z danego okresu czasowego - administrator zgłasza chęć pobrania wszystkich lokalizacji, które zostały zarejestrowane w podanym przez niego okresie czasu. W odpowiedzi, do osoby żądającej, wysłana zostaje lista zawierająca współrzędne lokalizacji użytkowników oraz ich adresy MAC.

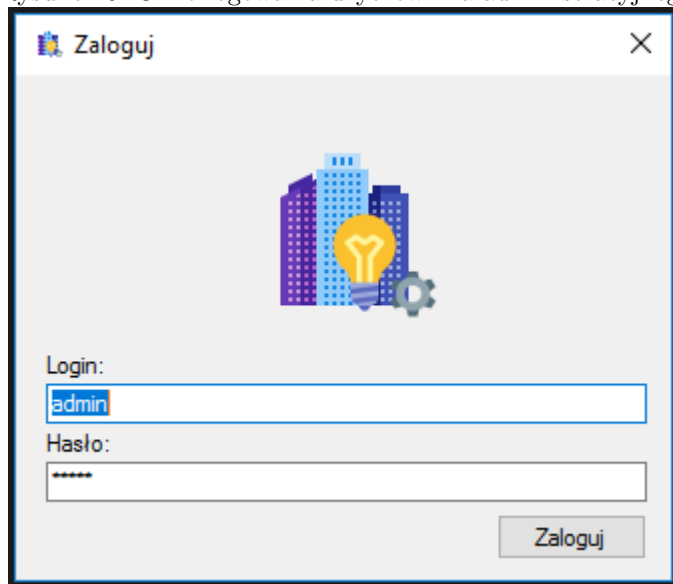
5.2 Aplikacja administracyjna

Głównym celem aplikacji klienckiej jest konfiguracja systemu i wyświetlanie oraz monitorowanie lokalizacji użytkowników. Dostęp do niej powinien być ograniczony, ze względów bezpieczeństwa, wyłącznie do uprzywilejowanych użytkowników.

5.2.1 Opis okien aplikacji

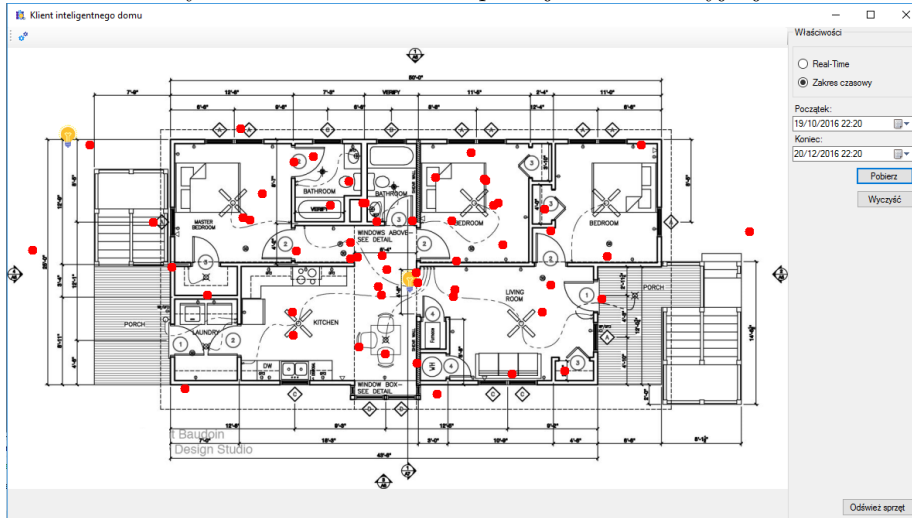
Okno logowania Pierwszym oknem, które pojawia się po włączeniu aplikacji, jest okno logowania. Osoba upoważniona wprowadza swój login oraz hasło. Naciśnięcie przycisku "Zaloguj" powoduje wysłanie zapytania do aplikacji serwerowej, która tworzy lokalny wpis o logowaniu się użytkownika oraz przechowuje adres IP, z którego przyszło żądanie. Zapisany adres wykorzystywany jest m.in. do wysyłania powiadomień o pojawieniu się lokalizacji mobilnej w sytuacji, gdy użytkownik zapisze się na subskrypcję notyfikacji.

Rysunek 5: Okno logowania użytkownika administracyjnego



Po pozytywnym przejściu procesu logowania, użytkownik zostaje przeniesiony do głównego okna aplikacji. Składa się ono z panelu z mapą, panelu wyboru trybu wyświetlania lokalizacji użytkowników oraz przycisku przejścia do konfiguracji.

Rysunek 6: Główne okno aplikacji administracyjnej



Główne okno W głównym panelu wyświetlana jest mapa systemu, do której ścieżka określona jest w pliku konfiguracyjnym "conf.xml". Jedynym wymaganiem co do mapy jest to, aby była ona plikiem graficznym. Mapa jest skalowalna - każda zmiana wielkości okna powoduje dopasowanie się obrazka do nowego panelu. To samo dotyczy wyświetlanych wewnątrz elementów. Na mapę nanoszone są urządzenia sterowane oraz lokalizacje użytkowników mobilnych. Urządzenia oznaczane są symbolem ich kategorii, zaś użytkownicy reprezentowani są przez czerwone kropki.

Lokalizacje urządzeń pobierane są podczas załadowania aplikacji (jednak po poprawnym przejściu procesu logowania) oraz automatycznie wyświetlane. Z racji tego, iż zmiany w konfiguracji urządzeń przez innych użytkowników administracyjnych nie odświeżają ikon na mapie, w prawym dolnym rogu aplikacji znajduje się przycisk, którego naciśnięcie spowoduje ponowne pobranie z serwera lokalizacji sprzętów sterowanych oraz przerysowanie symboli na mapie.

Lokalizacje użytkowników mobilnych mogą być wyświetlane w dwóch trybach:

- w czasie rzeczywistym - aplikacja administracyjna zgłasza na serwerze chęć zapisania się na subskrypcję lokalizacji. Dzięki temu, za każdym razem, jak tylko serwer ustali lokalizację użytkownika, zostaną o tym powiadomione wszystkie zalogowane aplikacje administracyjne. Punkty oznaczające użytkownika wyświetlane są na okres 4 sekund.
- w formie skumulowanej - administrator wysyła na serwer zakres czasowy, dla którego chce wyświetlić pozycje użytkowników. Aplikacja serwerowa pobiera lokalizacje z bazy, a następnie wysyła je spowrotem do klienta, na którym zostają wyświetlone. Punkty oznaczające użytkownika wyświetlane są bez określonego okresu ważności.

Przejście pomiędzy trybami powoduje usunięcie zachowanych w pamięci lokalizacji.

Punkty reprezentujące użytkowników oraz ikony sprzętów sterowanych są skalowane, aby realistycznie odwzorowywać swoją rzeczywistą pozycję. Wykorzystuje się do tego podane podczas konfiguracji wielkości rzeczywistego obiektu.

Jeżeli stosunek wysokości mapy do jej szerokości jest większy niż stosunek wysokości panelu wyświetlającego mapę do jego szerokości, pozycję ikony w panelu oblicza się na podstawie wzoru:

$$\begin{cases} x = \frac{rzecz_lok_urz_x}{rzecz_szer_mapy} * szer_pan_mapy \\ y = \frac{rzecz_lok_urz_y}{rzecz_wys_mapy} * wys_przel_mapy + wys_marg \end{cases} \quad (10)$$

gdzie:

- x i y oznaczają współrzędne w panelu mapy
- $rzecz_lok_urze_x$ i $rzecz_lok_urze_y$ to fizyczne współrzędne użytkownika mobilnego lub urządzenia sterowanego
- $rzecz_szer_mapy$ i $rzecz_wys_mapy$ to rzeczywiste wymiary prostokąta, wewnątrz którego znajduje się system
- $szer_pan_mapy$ jest to szerokość panelu wyświetlającego mapę (w tym wypadku szerokość mapy i szerokość panelu jest taka sama)
- wys_przel_mapy to wysokość mapy. W tym wypadku wysokość mapy nie jest równa wysokości panelu - aby zachować realistyczne proporcje, powyżej i poniżej obrazku mapy dodany jest margines. Wysokość mapy obliczana jest ze wzoru:

$$wys_przel_mapy = \frac{szer_pan_mapy * wys_obraz_mapy}{szer_obraz_mapy} \quad (11)$$

w którym wys_obraz_mapy i $szer_obraz_mapy$ to wymiary obrazka mapy, określone w pikselach.

- wys_marg to wysokość marginesu, jaki musi zostać dodany, aby zachować proporcję mapy. Obliczony jest ze wzoru:

$$wys_marg = \frac{wys_pan_mapy - wys_przel_mapy}{2} \quad (12)$$

W przypadku, jeżeli stosunek wysokości mapy do jej szerokości jest niższy niż stosunek wysokości panelu wyświetlającego mapę do jego szerokości, obliczenia wykonuje się analogicznie, ale z tą różnicą, że w tym wypadku to wysokość mapy będzie równa wysokości panelu i dlatego to faktyczna szerokość mapy w oknie oraz szerokość bocznych marginesów muszą zostać wyliczone.

Okno konfiguracji Wciśnięcie przycisku "Konfiguracja" (reprezentowane w oknie jako ikoną dwóch trybów), otwiera osobne okno, pozwalające na konfigurację kluczowych elementów systemu.

Pierwszy panel w oknie konfiguracji pozwala na edycję wielkości fizycznej systemu oraz danych na temat routerów (umożliwia ich również dodawanie oraz usuwanie). W dolnej części okna wyświetlone są wszystkie zarejestrowane routery. Wciśnięcie wybranego z nich powoduje przejście do trybu edycji - pola edycji routerów zostaną wypełnione danymi wybranego urządzenia.

Rysunek 7: Panel edycji danych routerów

The screenshot shows a window titled 'Konfiguracja' with three tabs: 'Lokalizacja', 'Zarządzanie sprzętem', and 'Użytkownicy'. The 'Router' tab is selected, displaying a form for editing router data. The form includes fields for SSID (TP-LINK_ACP), Sila transmitera (16), Zysk anteny (2), Waga (3), and Rodzaj (WIFI). There are also location fields (X: 80, Y: 20, Z: 0). Below the form is a table of registered routers.

	Id	SSID	TrasmitterPower	AntennaGain	LocationX	LocationY	LocationZ	RouterCategory	Weight
▶	2	TP-LINK_ACP	16	2	80	20	0	1	3
	3	AndroidAP	10	2	50	80	0	1	3
	1	TP-LINK_7C...	16,5	5	10	50	0	1	4

Poza polami wymaganymi do lokalizacji użytkownika, którymi są:

- lokalizacja routera - współrzędne X, Y, Z oraz SSID
- zysk anteny
- siła transmitera

aplikacja pozwala dodatkowo na zadeklarowanie typu sygnału wysyłanego przez urządzenie. Ma to wpływ na sposób obliczania odległości do użytkowników, ale także na wartość początkową wagi przydzielonej temu transmitterowi. Wagę można zmienić ręcznie, ale trzeba pamiętać, że musi się znaleźć w zakresie [1:4]. Drugi panel pozwala na definiowanie i edycję urządzeń sterowanych.

Rysunek 8: Panel edycji urządzeń sterowanych

Kategoria	Id	Nazwa
Light	1	Tester
Heater	2	Nowe

Nazwa	Tester
Ip	192.168.1.105
Port	52011
Minimalne oświetlenie (%)	20
Maksymalne oświetlenie (%)	100
X	10
Y	20
Z	0
Moduł komunikacyjny:	Logger
Obsługa zdarzeń:	ImportantUserFirstContr
<input checked="" type="checkbox"/> Czy aktywny	

Wszystkie urządzenia zarejestrowane w systemie podzielone są na kategorie. Dostępne kategorie przechowywane są w bazie danych i stamtąd są pobierane, kiedy aplikacja klientowa wyśle żądanie na serwer. Wybranie kategorii z lewego panelu powoduje pobranie z serwera wszystkich urządzeń, które należą do wybranej kategorii. Do każdej kategorii przydzielona jest inna akcja w kontrolerze ConfController na serwerze - przykładowo, dla urządzeń oświetleniowych, jest to akcja GetLightDevices.

Aby przejść do widoku wyświetlania i edycji danych urządzenia, należy wybrać interesującą nas pozycję ze środkowego panelu. Powoduje to pojawienie się w prawym panelu formularza z danymi danego urządzenia. Formularze są tworzone dynamicznie, a rodzaj i ilość ich pól zależne są od kategorii urządzenia. Dla urządzenia z kategorii "Światło", dostępne są następujące pola:

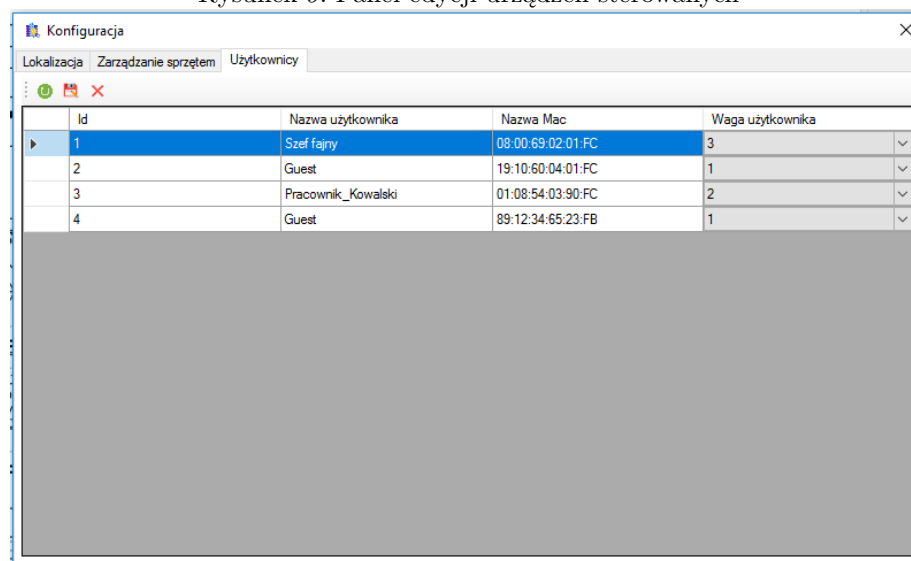
- lokalizacji - współrzędne XYZ urządzenia. pozwalające umiejscowić je w rzeczywistym środowisku
- adres ip i port - pozwalają na komunikację z urządzeniem
- nazwa - pozwala użytkownikom odróżniać urządzenia między sobą
- minimalne i maksymalne oświetlenie - zakres, określony w procentach, z którego może korzystać system podczas sterowania urządzeniem. W przypadku sterowania w interwałach czasowych, minimalne oświetlenie przydzielane jest w sytuacji, gdy, w podanym okresie czasu, obok urządzenia nie przeszedł żaden użytkownik. Analogicznie, maksymalne oświetlenie przydzielane jest, gdy wszystkie obliczone lokalizacje użytkowników znajdowały się w niedalekiej odległości od sterowanego urządzenia

- moduł komunikacyjny - określa sterownik, sposób komunikacji serwera z urządzeniem. Moduły komunikacyjne definiowane są w bazie danych, a związane z nimi klasy obsługujące muszą zostać dodane do kodu serwera
- obsługa zdarzeń - nazwa modułu, który na bieżąco steruje urządzeniem, w zależności od lokalizacji użytkowników zlokalizowanych w bliskiej odległości. Z modułem związana jest klasa, która obsługuje przypisany do urządzenia moduł

Panel udostępnia przyciski, które pozwalają na zapis wprowadzonych zmian, jak również dodanie lub usunięcie urządzenia.

Ostatni panel to panel użytkowników. Są to osoby, które korzystały z aplikacji mobilnej, a ich pozycja została chociaż raz obliczona.

Rysunek 9: Panel edycji urządzeń sterowanych



	Id	Nazwa użytkownika	Nazwa Mac	Waga użytkownika
▶	1	Szef fajny	08:00:69:02:01:FC	3
	2	Guest	19:10:60:04:01:FC	1
	3	Pracownik_Kowalski	01:08:54:03:90:FC	2
	4	Guest	89:12:34:65:23:FB	1

Okno składa się z tabeli, w której znajdują się zapisani użytkownicy oraz przycisków:

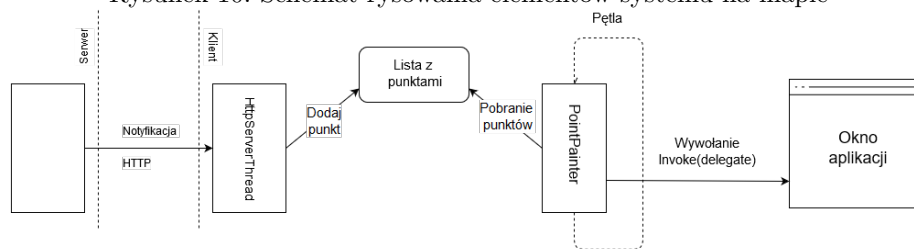
- odświeżania danych
- zapisania wszystkich użytkowników
- usunięcia wybranego w tabeli użytkownika

Widok pozwala na podglądnięcie użytkowników oraz na zmianę ich nazwy oraz wagi. Nazwa pozwala na różnicowanie użytkowników w sposób łatwiejszy niż po adresie MAC. Waga odgrywa kluczową rolę podczas obliczania parametrów do sterowania urządzeniami - czym wyższa waga użytkownika, tym większe ma on znaczenie dla urządzenia, w którego pobliżu znajduje się obliczona lokalizacja.

Dodatkowo, pojawienie się użytkownika o określonej wadze może mieć wpływ na zajęcie, zdefiniowanego dla urządzenia, zdarzenia. Wagi przyjmują wartość od 0 (użytkownik o najniższym priorytecie, nie brany pod uwagę podczas określania parametrów) do 4 (użytkownik o najwyższym priorytecie).

Wyświetlanie lokalizacji użytkowników W trybie wyświetlania lokalizacji w czasie rzeczywistym, aplikacja musi wyświetlać uzyskane dane bez odświeżania panelu czy ingerencji użytkownika. W związku z tym potrzebne było wprowadzenie mechanizmu opartego na osobnym wątku. Klasą zarządzającą rysowanie punktów jest `PointPainter`. W wątku wykonuje się metoda tej klasy, `DrawPoint()`, zawierająca nieskończoną pętlę, usypianą na 0,5 sekundy przy każdym przejściu. Wątek ma dostęp do listy punktów, które powinien wyświetlić. Lista uzupełniania jest przez obiekt klasy `HttpServerThread`, na podstawie notyfikacji przychodzących z serwera. Każdy wpis na liście ma informacje o lokalizacji (współrzędne XY) oraz dacie ważności danego punktu. Punkty stworzone na podstawie notyfikacji z serwera mają okres ważności równy 4 sec, za to punkty wyświetlane w sposób skumulowany mają datę ważności ustawioną na wartość null (co oznacza, że są wyświetlane dopóki nie zostanie wywołana komenda wyczyszczenia mapy).

Rysunek 10: Schemat rysowania elementów systemu na mapie



Metoda wyświetlająca punkty nie rysuje ich bezpośrednio na wyświetlonej mapie, ponieważ odświeżanie wymagałoby ponownego przerysowania nie tylko punktów, ale i całej mapy. Mogłoby to powodować opóźnienia w płynności wyświetlania, a wiązałoby się z tym nieprzyjemne dla oka "mruganie" mapy. Dlatego, punkty nanoszone są na przeźroczysty obrazek, który później wyświetlany jest na warstwie wyświetlającej punkty. Zanim jednak na obrazku zostaną naniesione lokalizacje użytkowników, klasa `PointPainter` zaznacza na przeźroczystej warstwie urządzenia sterowane, rysując odpowiednią ikonę kategorii danego sprzętu. Dodatkowo, ponieważ metoda rysująca punkty działa na innym wątku niż panel zawierający mapę, wyświetlenie punktów nie może być wywołane bezpośrednio z poziomu klasy `PointPainter`. Z tego powodu, funkcja rysująca obraz z punktami i ikonami urządzeń na mapie zostaje opakowana w delegatę (typ danych przechowujący funkcję), a następnie referencja do niej zostaje wstawiona jako parametr funkcji `Control.Invoke(delegate)`, dzięki czemu panel sam wywoła przekazaną mu jako parametr funkcję podczas kolejnego wykonywania się jego

watku.