Politechnika Poznańska
Lokalizator użytkowników sieci WiFi

18 czerwca 2017

Maciej Michalak	121992	maciej.k.michalak@student.put.poznan.pl
Patryk Masiakowski	116285	patryk.masiakowski@student.put.poznan.pl
Jakub Kostrzewski	122039	jakub.k.kostrzewski@student.put.poznan.pl

Spis treści

Spis rysunków

1 Wstęp

1.1 Opis aplikacji

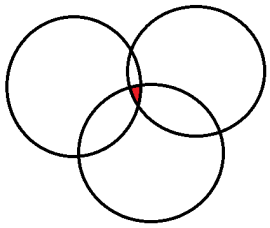
Celem projektu jest stworzenie aplikacji pozwalającej na określenie lokalizacji użytkownika sieci względem trzech urządzeń AP. System powinien przeskanować sieć w poszukiwaniu klientów podłączonych do punktów dostępowych. Użytkownik będzie mieć możliwość podejrzenia podstawowych informacji o każdym hoście w sieci, takicj jak nr. ip czy adres MAC. Do określenia względnej lokalizacji w sieci wykorzystana zostanie technika trilateracji. Aplikacja działa na zasadzie serwera interpretującego dane przesłane przez punkty dostępowe. Klienci powinni znaleść w swoim otoczeniu trzy punkty dostępowe, które mają wzlędem nich określoną siłę syganłu. Dane o tych sygnałach zostają przesyłane do serwera i interpretowane. W czasie rzeczywiśtym podawana jest lokalizacja danego klienta.

2 Działanie

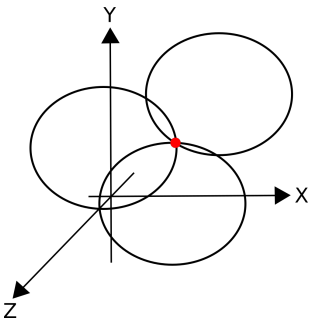
2.1 Trilateracja

Jest to metoda określania położenia obiektu w trójwymiarowej przestrzeni(w tym wypadku budynku). By metoda ta była skuteczna wymagana jest znajomość położenia trzech punktów dostępowych (AP).Znając odległości każdego punktu dostępowego od lokalizowanego urządzenia oraz współrzędne tych punktów można określić lokalizację urządzenia.

Każdą odległość AP od lokalizowanego urządzenia można przedstawić w przestrzeni jako sfere. Wyznaczenie współrzędnych klienta sprowadza się do znaleznienia miajsca przecięcia trzech sfer, każdej związanej z AP.



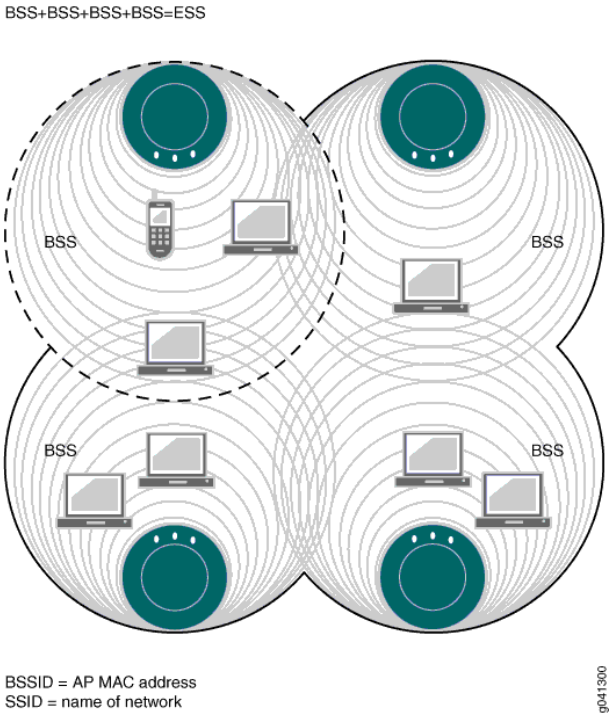
Rysunek 1: Sfery wyznaczone przez siłe sygnału AP



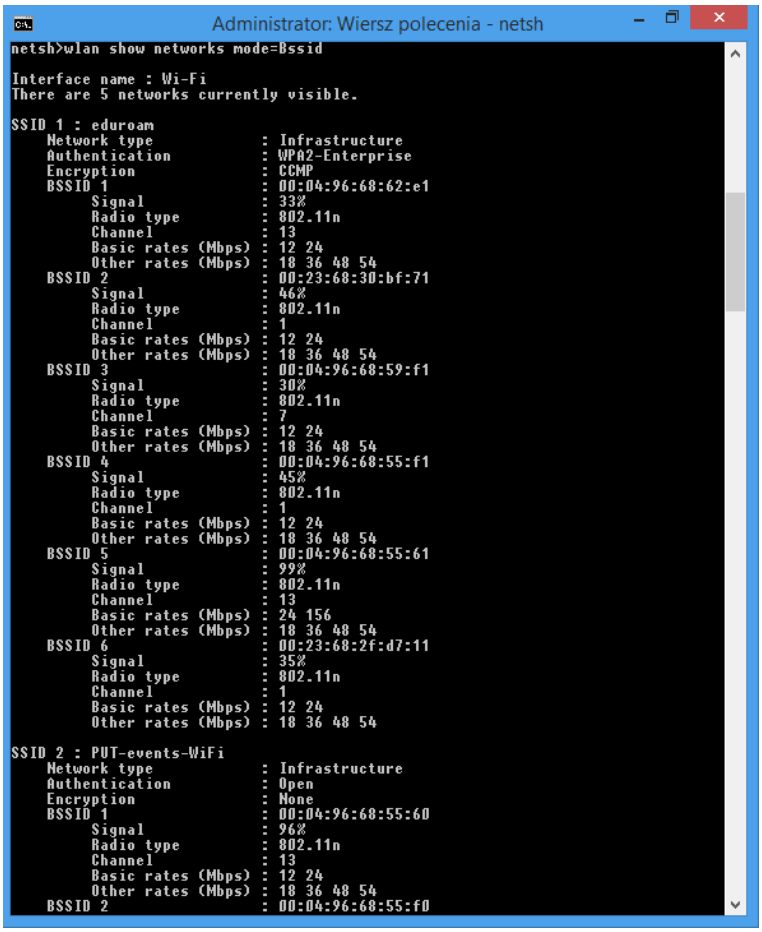
Rysunek 2: Siła sygnału w przestrzeni

2.2 SSID i BSSID

Początkowym etapem działania aplikacji będzie lokalizowanie pobliskich punktów dostępowych i ich identyfikacja na podstawie 48-bitowego numeru identyfikacyjnego BSSID czyli numeru MAC AP oraz identyfikatora SSID.



Rysunek 3: BSSID i ESSID



Rysunek 4: Listowanie dostępnych essid

2.3 RSSI

RSSI jest wskaźnikiem mocy sygnału nadawanego przez dany punkt dostępowy. Wykorzystując wartości tego wskaźnika, możliwe jest określenie odległości lokalizowanego urządzenia od punktu dostępowego.

2.4 Punkty dostępowe

Do przeprowadzenia testów aplikacji zostanie "zbudowana" prosta sieć utworzona na hotspotach z telefonów komórkowych. Do poprawnego działania będą potrzebne trzy lub więcej punkty dostępowe.

2.5 Beacony i ProbeRequest

Ramki Beacon służą punktom dostępowym do informowania potencjalnych klientów sieci o świadczeniu usługi połączenia bezprzewodowego. Zawierają informacje o adresie fizycznym AP.

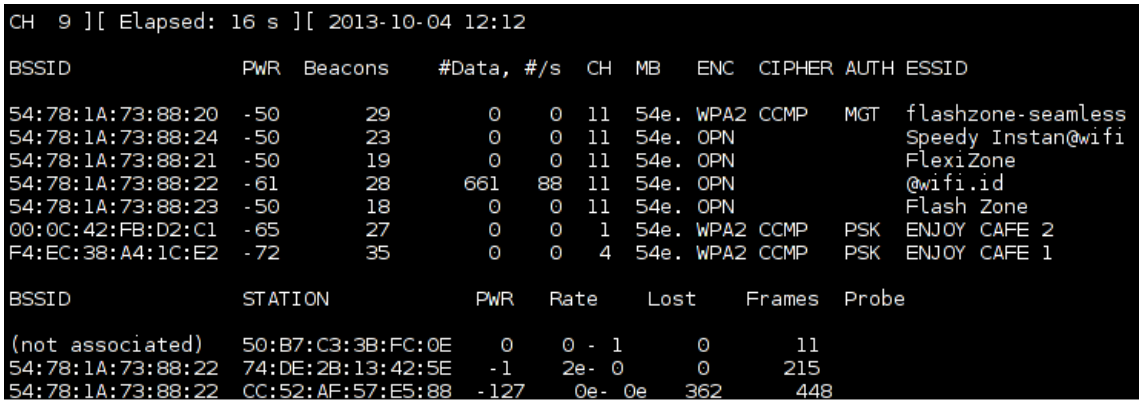
Probe Request to pakiety wysyłane przez urządzenia sieciowe(np. smartfony) w celu wykrycia dostępnych punktów dostępowych. AP po otrzymaniu takiego pakietu może odczytać jaka jest siła sygnału urządzenia klienta względem niego.

2.6 Aplikacja serwera

Serwer otrzymuje dane od klienta i na podstawie poniższego wzoru wyznacza odległości do trzech punktów dostępowych. Następnie rysuje sfery i podaje lokalizacje użytkownika (część wspólna sfer).

$$d = 10^{((TxPower - RSSI)/(10*n))}$$

d - odległość w metrach
TxPower - maksymalna siła sygnału AP
RSSI - sygnał AP z punktu widzenia klienta
n - stała propagacji(2 dla wolnych przestrzeni)



Rysunek 5: Przykładowe działanie airodump-ng

2.7 Harmonogram prac

HARMONOGRAM PRAC

MARZEC	-zaznajomienie się z tematyką pracy -zebranie potrzebnych materiałów -rozpoczęcie tworzenia dokumentacji -podział prac
KWIECIEŃ	-rozpoczęcie pracy nad aplikacją kliencką -rozpoczęcie pracy nad aplikacją serwera
MAJ	-wstępne testy aplikacji -kalibracja
CZERWIEC	-dodatkowe testy -poprawki

Rysunek 6: Harmonogram prac

2.8 Funkcje-klient

- wyszukiwanie dostępnych AP po BSSID,
- pobieranie danych o sygnale względem każdego BSSID,
- wysyłanie danych do serwera.

2.9 Funkcje-serwer

- wyświetlanie dostępnych klientów w sieci,
- pobieranie danych o sygnałach RSSI od klientów,
- obliczanie lokalizacji w przestrzeni na podstawie trilateracji.

3 Symulowanie działania aplikacji klienta

Z powodu problemów z odpowiednim odczytywaniem odległości punktów dostępowych, napisany został program symulujący wysyłanie "odpowiednich" dacnyh do serwera. Znaczy to w tym wypadku, że konkretna siła sygnału prezentuje zawsze tą samą odległości AP i wahania tych wartości nie są zbyt duże. Napisanie takiej aplikacji pozwoliło nam spreccyzować jaki efekt pracy chcemy uzyskać.

4 Napotkane problemy i potenjclane problemy przy dalszym rozwoju aplikacji

Mimo, że w teorii działanie aplikacji nie opiera się na żadnych skomplikowanych metodach czy obliczeniach, kilka rzeczy okazało się problematyczne. Projekty tego typu cieszą się raczaj wątpliwą popularnością i ciężko o udokumentowane przypadki naprawde dokładnego lokalizowania użytkownika sieci WiFi. Czymś co na pewno mogłoby uskutecznić działanie takiego systemu to możliwości finansowe. Pozwoliłoby to przygotować w pełni autorskie środowisko testowe.

Kwestiami problematycznymi w projekcie okazały sie:

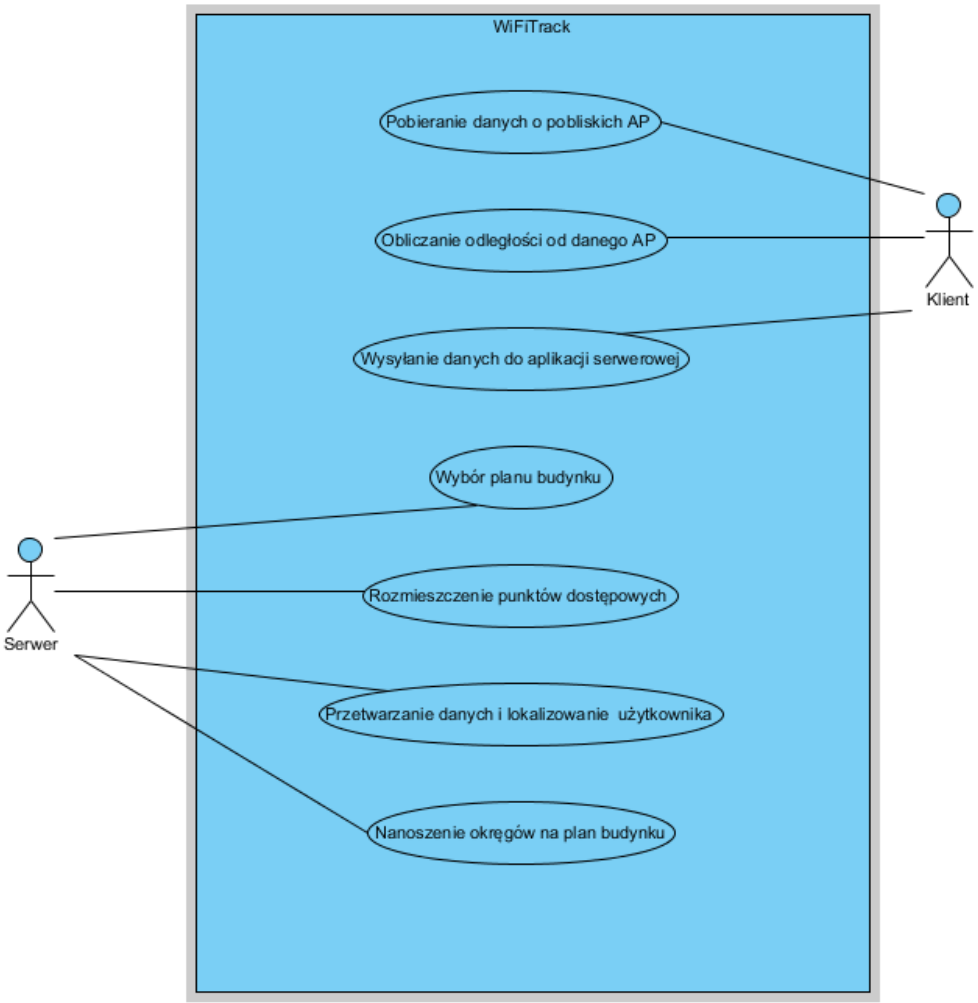
- **Sygnał RSSI nie jest dobrą miarą odległosci od AP** - siła sygnału punktu dostępowego może zależeć od wielu rzeczy. Mocy nadajnika, mocy odbirnika, poziomu naładowania baterii w takowych, modelu sprzętu, który służy w testowaniu. Podczas testów aplikacji okazało się, że w przypadku sprzętu róznych producentów, z różnymi kartami sieciowymi czy antenami, nie udało się uzyskać dokładnych wyników. Dwa urządzenia, znajdujące się w tej samej odległości od urządzenia z aplikacją kliencką, pokazywały na serwerze inne odległości. Znacząco inne.

- **Kalibracja** - aplikacja miała z założenia działać na zasadzie nakładania na plany budynków punktów dostepowych, rozmieszczania ich. Po rozmieszczeniu każdy z punktów, w czasie rzeczywistym, odbierał dane od aplikacji klienta i rysował okręgi zasięgów na planie. Należało w odpowiedni sposób skalibrować mapę z rzeczywistymi wymiarami np. pomieszczeń. Szukanie pewnego współczynnika skalującego metry na piksele okazało się dosyć problematyczne.

5 Diagramy aplikacji

5.1 Diagram przypadków użycia

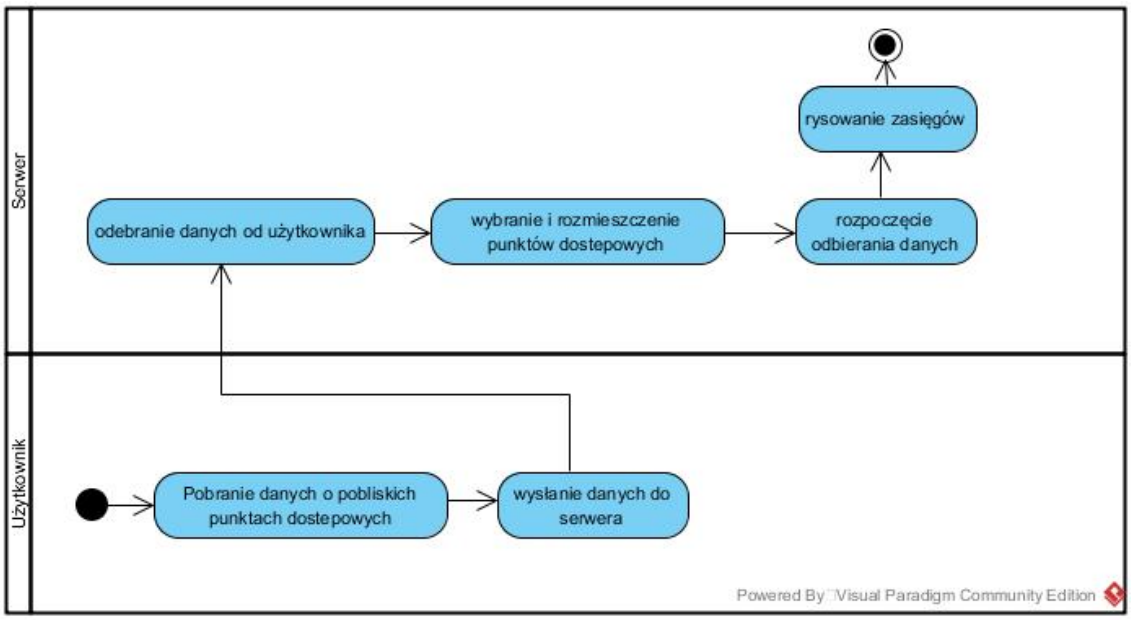
Poniżej znajduje się podstawowy diagram przypadków użycia aplikacji. Pokazuje wszystkie funkcjonalności które oferuję aplikacja kliencka oraz serwerowa, z podziałem na aktorów.



Rysunek 7: Diagram przypadkow użycia

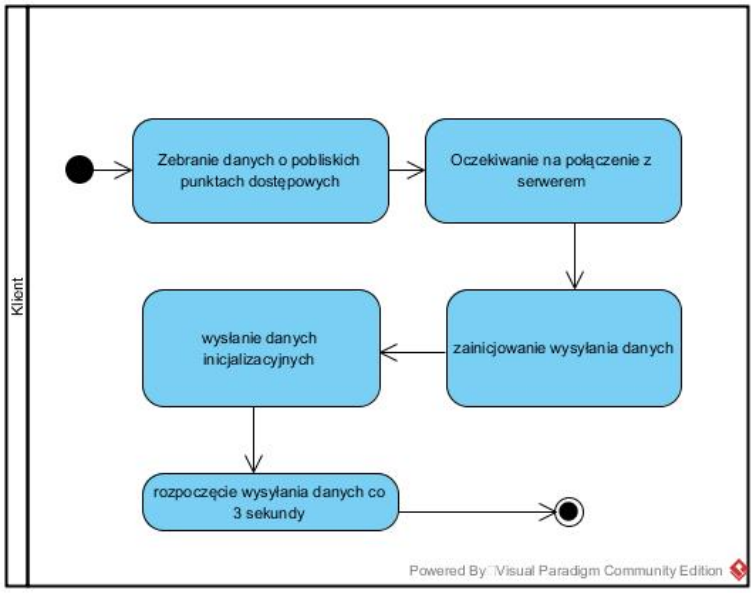
5.2 Diagramy aktywności

Diagram aktywności znajdujący się na rysunku poniżej przedstawia, jak przebiegają wszystkie procesy w naszym projekcie. Rozpoczynając od pobierania danych o punktach dostępowych z aplikacji klienckiej do rysowania okręgów odległości na ekranie aplikacji serwerowej.



Rysunek 8: Diagram aktywności - rysowanie

Kolejny diagram aktywności przedstawia działanie aplikacji klienckiej znajdującej się na urządzeniu moblinym. Diagram ukazuje kroki, które wykonuje aplikacja w tle. Wszystkie procesy oprócz zainicjowania wysyłania danych są wykonywane bez ingerencji klienta.

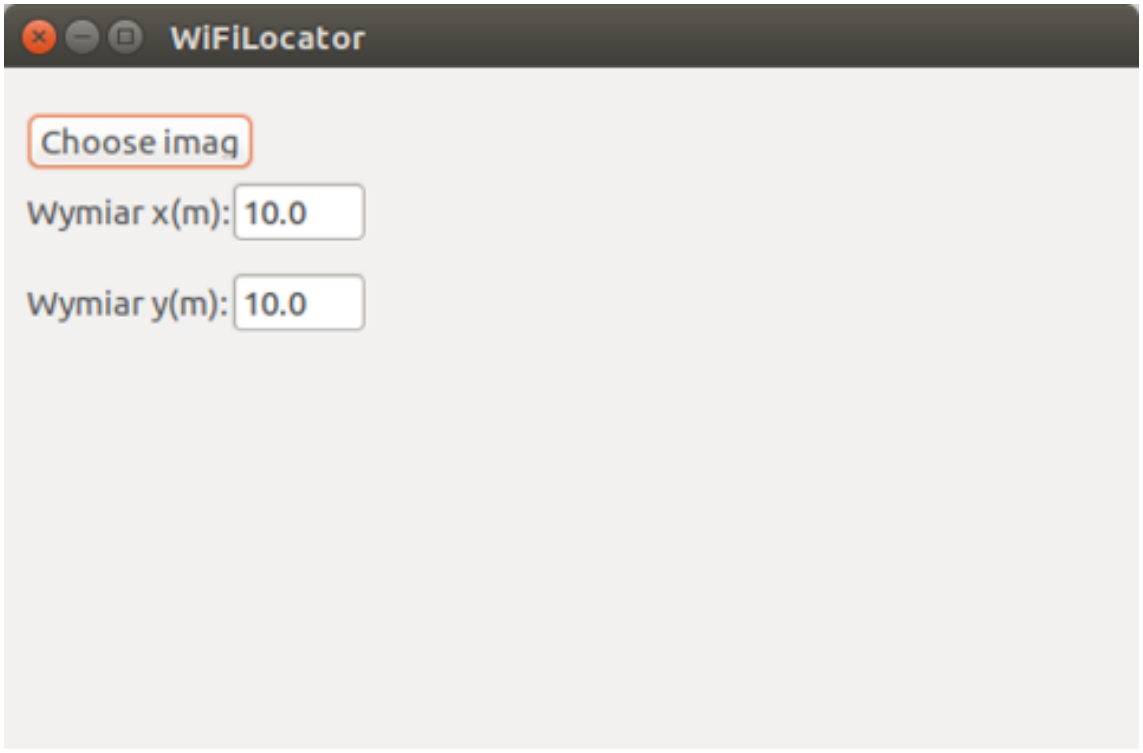


Rysunek 9: Działanie apliakcji klienta

6 Podstawowe mockupy aplikacji

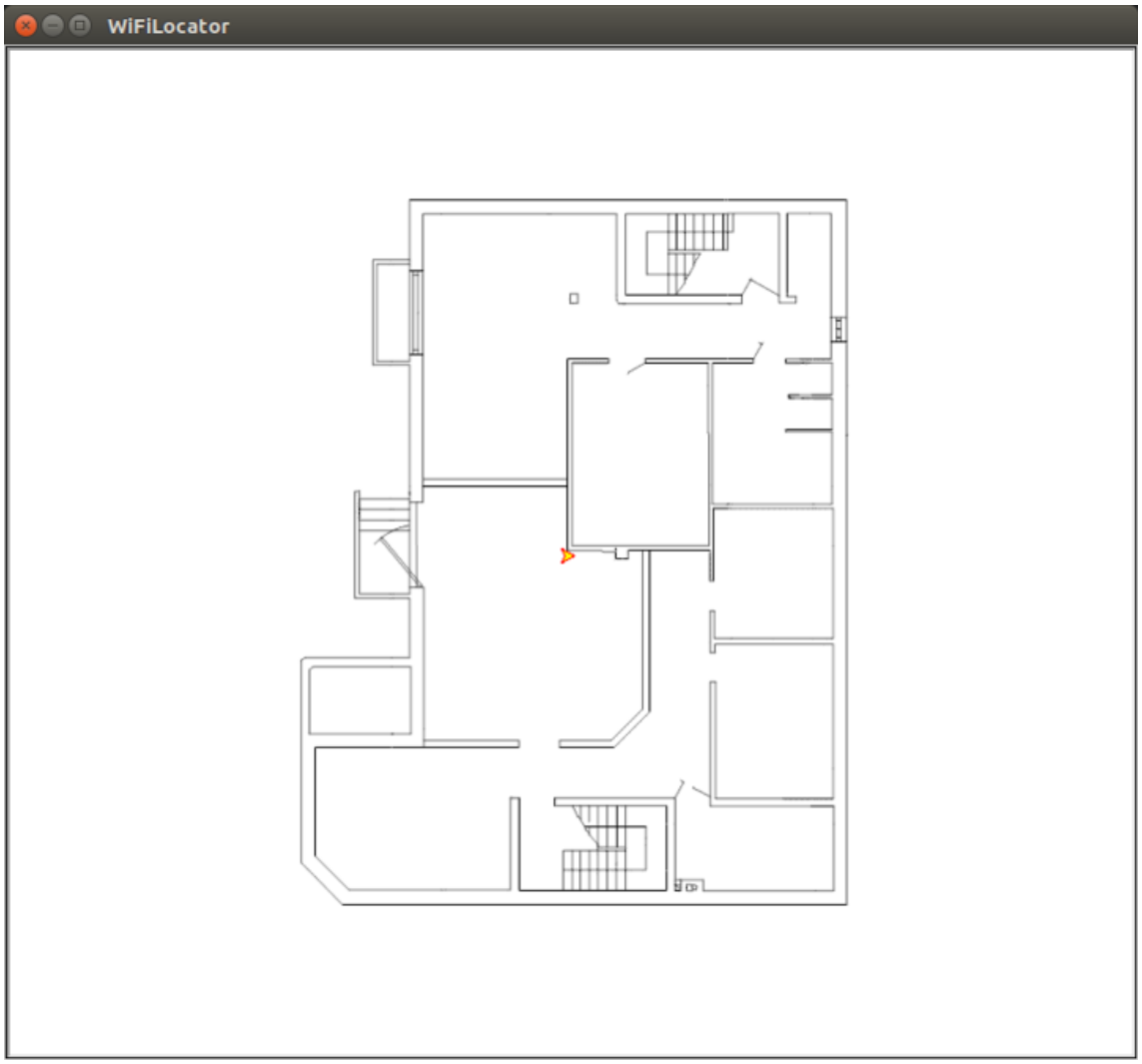
6.1 Aplikacja serwerowa

Widok aplikacji serwerowej zaraz po uruchomieniu programu. Mamy tutaj możliwość dostosowania naszej planszy(pomieszczenia), do powierzchni którą rzeczywiście zajmuję. Wymiary pomieszczenia podajemy w metrach. Po wpisaniu wymiarów wybieramy plik graficzny odzwierciedlający plany pomieszczenia lub budynku.



Rysunek 10: Wybór wymiarów

Widok aplikacji serwerowej w którym mamy możliwość ręcznego ustawienia położenia naszych AP za pomocą lewego przycisku myszy. Bardzo ważne jest żeby ustawić AP dokładnie, ponieważ od tego znacząco zależy dokładność naszej aplikacji.



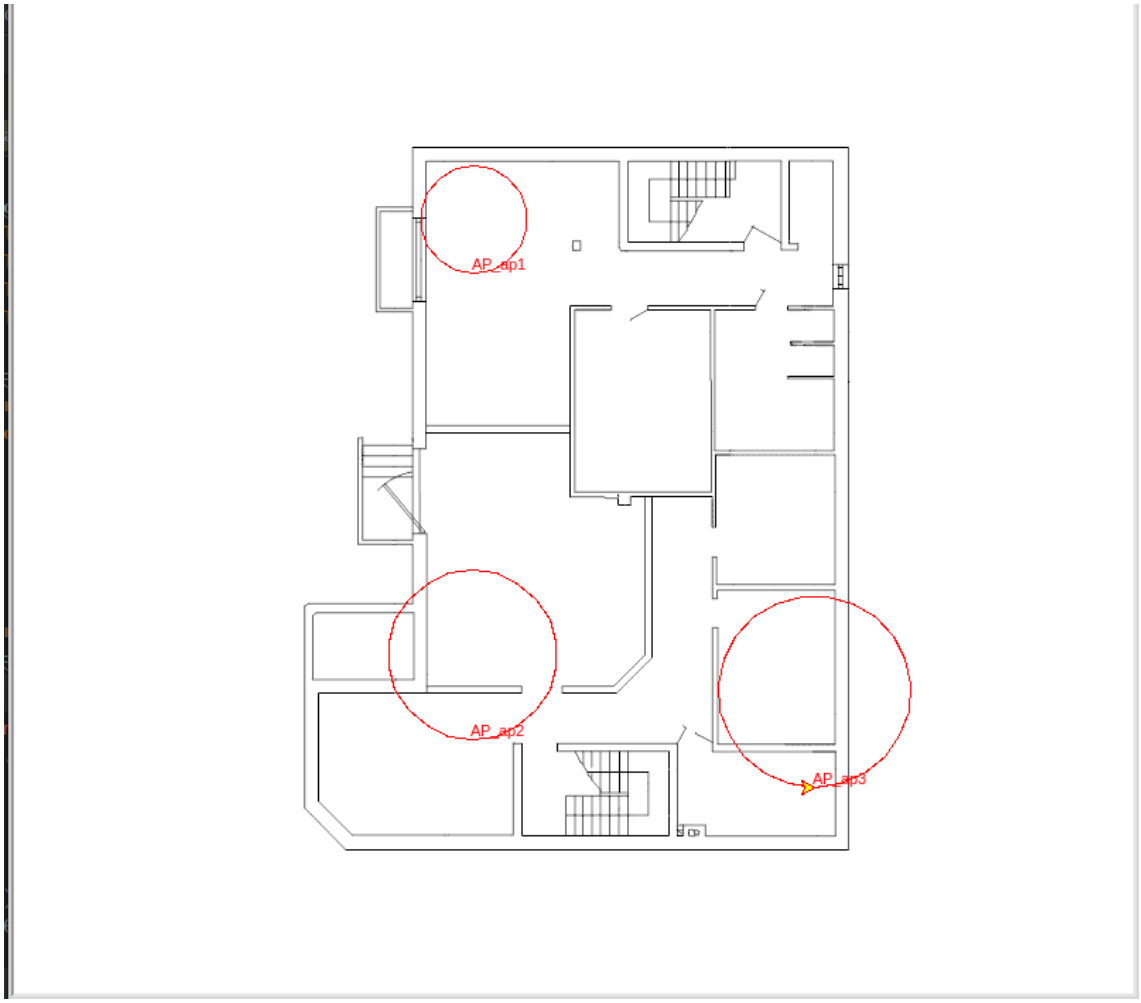
Rysunek 11: Aplikacja serwera po wybraniu pliku graficznego

Ten widok aplikacji przedstawia przykładowe przesłane informacje z aplikacji klienckiej do aplikacji serwerowej. Pierwsze przesłanie danych skutkuje możliwością rozstawienia punktów dostepowych na wybranej przez nas wcześniej mapie. Kluczową kwestią w tym kroku jest dokładne rozstawienie punktów na planach budynku.

```
['34', '59', '66']
[1360.0, 2360.0, 2640.0]
[(-98, -1282.0), (-87, -2426.0), (184, -2856.0)]
[(-98, -1122.0), (-87, -2466.0), (184, -2696.0)]
['34', '57', '68']
[1360.0, 2280.0, 2720.0]
[(-98, -1122.0), (-87, -2466.0), (184, -2696.0)]
[(-98, -1122.0), (-87, -2386.0), (184, -2776.0)]
['38', '55', '68']
```

Rysunek 12: Przykładowe dane przesłane do aplikacji serwerowej

Widok aplikacji poniżej przedstawia w pełni działającą aplikację serwerową. W tym miejscu następuje przesyłanie odległości klienta od zaznaczonych punktów dostepowych, które są odświeżane w odstępie czasowym 3 sekundy.



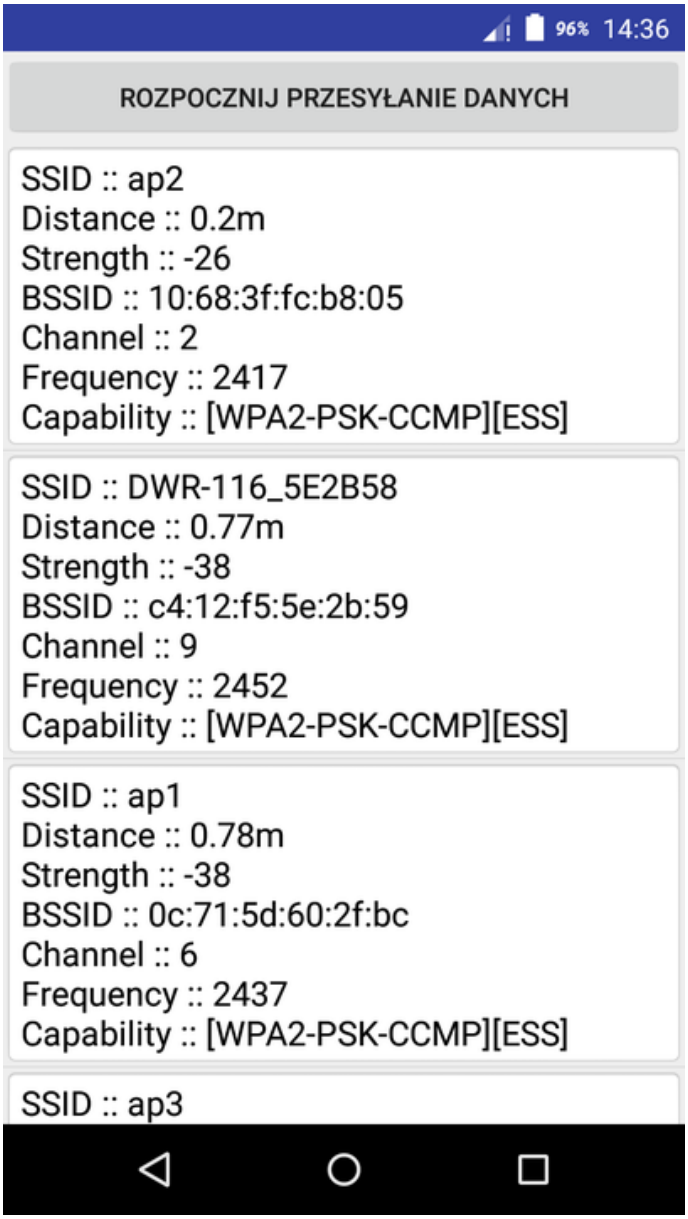
Rysunek 13: Widok w pełni działającej aplikacji serwerowej

6.2 Aplikacja kliencka

Widok okna aplikacji klienta pokazujący się gdy serwer nie jest aktualnie włączony. Aplikacja przejdzie do nowego okna widoku oraz rozpocznie przesyłanie danych zaraz po włączeniu serwera.

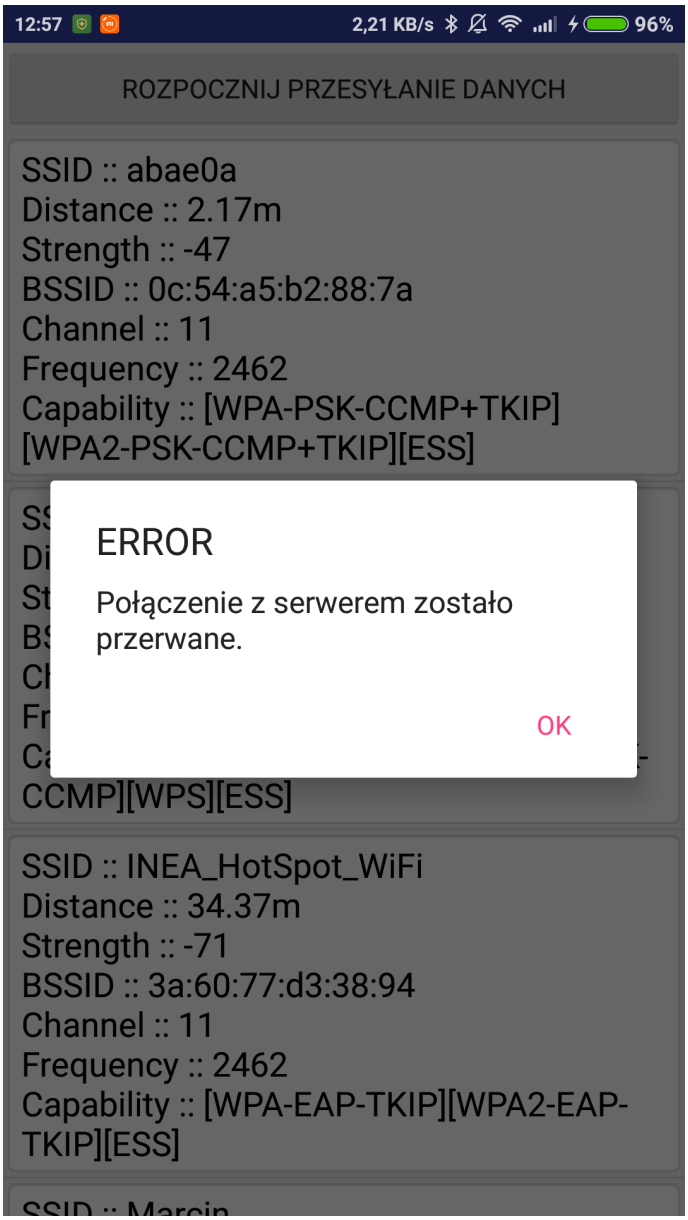
Rysunek 14: Aplikacja czekająca na odpowiedź serwera

Główny widok aplikacji klienta przedstawia listę AP z nazwami, dystansem, siłą signału, kanałem, częstotliwością oraz zabezpieczeniami. Po kliknięciu przycisku rozpoczęcia przesyłania danych wysyłane są wiadomości do serwera w odstępie czasowym 3 sekundy.



Rysunek 15: Punkty dostępowe w apliakcji klienckiej

Widok okna aplikacji klienta podczas obsługi błędu nagłej utraty połączenia z serwerem.



Rysunek 16: Okno błędu rozłączenia z serwerem

7 Możliwości rozwoju aplikacji

- **Podsumowanie odległości w czasie** - w przyszłości aplikacja mogłaby wyliczać odległość przebytą przez klienta na podstawie danych przesyłanych na serwer, w ten sposób moglibyśmy wyliczyć sume długości jaką klient przebył w trakcie miesiąca bądź roku.
- **Rysowanie tras** - do aplikacji klienckiej mógłby zostać dodany moduł który wyświetlałby narysowane przez serwer trasy po których się poruszaliśmy w trakcie naszego użytkowania aplikacji.
- **Obliczanie prędkości** - w aplikacji klienckiej w przyszłości moglibyśmy zaimplementować widok, w którym zostały by wyświetlane średnie prędkości w danych godzinach. Stwarzałoby to możliwość analizowania w których okresach w ciągu dnia jesteśmy najbardziej aktywni.
- **Widok aplikacji** - na potrzeby projektu skupiliśmy się nad funkcjonalnością naszego projektu. W przyszłości moglibyśmy się skupić nad bardziej przejrzystym oraz milszym dla oka interfejsem aplikacji klienckiej oraz serwerowej.