

POLITECHNIKA WARSZAWSKA
WYDZIAŁ MATEMATYKI I NAUK INFORMACYJNYCH



APLIKACJE I USŁUGI GIS

Aplikacja lokalizująca telefon w Gmachu MiNI

Autorzy:

Paweł KUŹMICZ
Przemysław RZĄD
Mateusz PIELAT

27 maja 2015

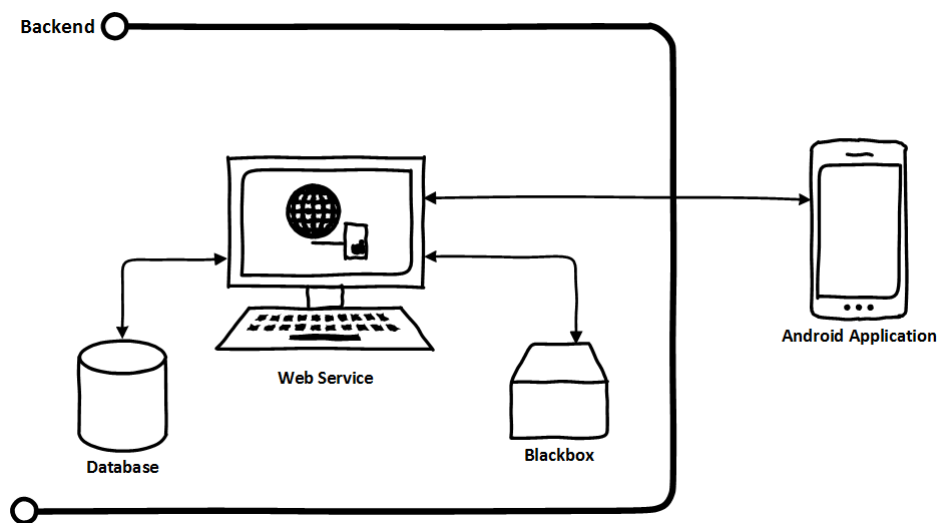
Informacje wstępne

Cel

Projekt ma na celu udostępnienie możliwości lokalizowania telefonu komórkowego w gmachu MINI. Całość składać się będzie z dwóch głównych części: klienta, uruchomionego na urządzeniu mobilnym oraz web serwisu udostępniającego dane o lokalizacji urządzenia.

Opis komponentów

Backend



Rysunek 1: Backend

Zadaniem backendu jest udostępnienie usług sieciowych dla aplikacji mobilnej. Centralnym elementem jest Web Service, który uruchomiony będzie na komputerze za pomocą IIS.Express. Web Service napisany jest w C#.

Do przewidywania lokalizacji używamy czarnej skrzynki dostarczonej przez inny zespół. Z powodu konieczności zainstalowanego R (oraz kilku sztuczek wymaganych aby działał RDotNet) zrezygnowaliśmy z umieszczenia backendu w chmurze na rzecz lokalnego komputera.

Baza danych służy do przechowywania lokalizacji (wraz z datą) powiązanych z użytkownikami. Dane te są potrzebne do prezentowania historii użytkownika.

Krótki opis działania metod Web Service'u

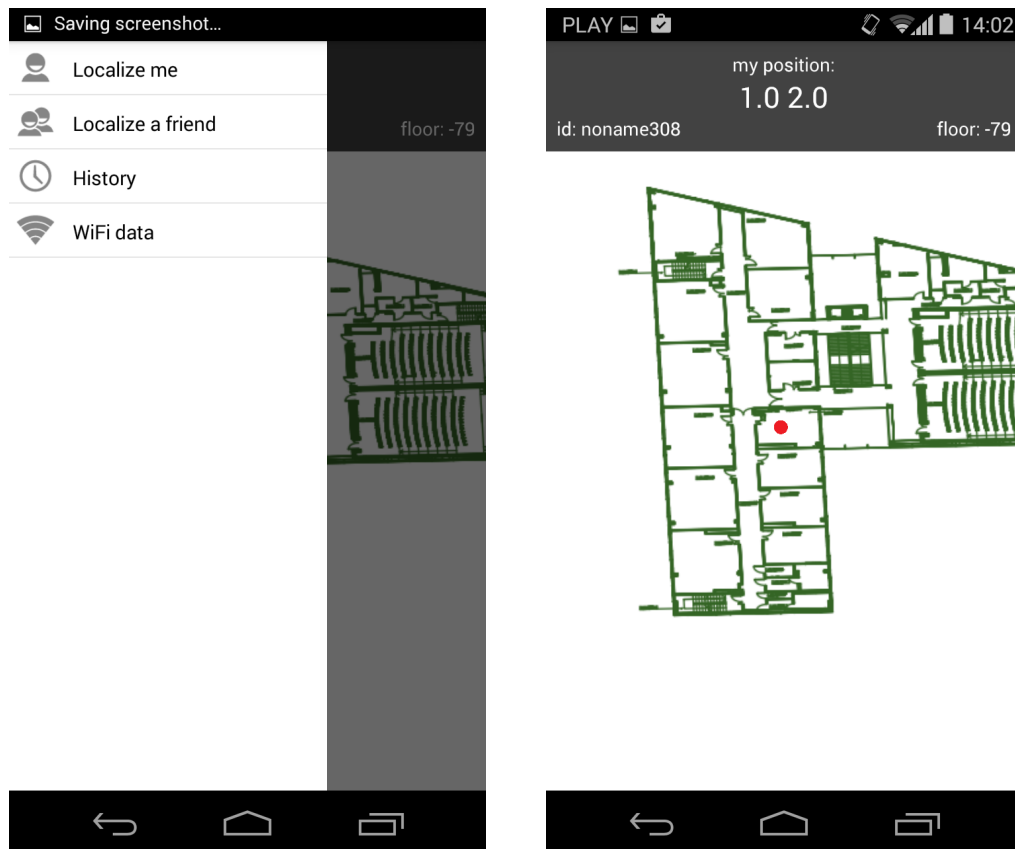
- `LocationDate` `getLocation(String id, WirelessSignal[] signals);`
 1. Utworzenie instancji czarnej skrzynki
 2. Przewidzenie lokalizacji na podstawie sygnałów
 3. Jeżeli w bazie danych nie istnieje użytkownik o takim id, stworzenie go
 4. Dodanie do historii lokalizacji użytkownika przewidzanej lokalizacji
 5. Zwrócenie lokalizacji (z aktualną datą)
- `String` `getLocationMap(Location location);`
 1. Pobranie obrazka z WMS (lokkom) odpowiedniego dla lokalizacji
 2. Dodanie znacznika wskazującego lokalizację
 3. Zwrócenie wyniku
- `LocationDate[]` `getHistory(String id);`
 1. Pobranie odpowiedniego użytkownika na podstawie id
 2. Jeżeli nie ma takiego, zwrócenie pustej tablicy
 3. Zwrócenie tablicy lokalizacji - najnowsza lokalizacja poprzedzona lokalizacjami znajdującymi się na tym samym piętrze
- `String` `getHistoryMap(String id);` (*może się zmienić*)
 1. Przygotowanie tablicy lokalizacji (jak wyżej)
 2. Pobranie obrazka z serwisu WMS
 3. Narysowanie znaczników na lokalizacjach
 4. (*być może*) Połączenie ich kreskami
 5. Zwrócenie wyniku

Aplikacja kliencka

Ekran startowy

- Start - uruchamia główną część aplikacji
- Settings - pozwala na zdefiniowanie adresu usługi GISowej

Ekran główny



Rysunek 2: Ekran główny aplikacji
(obecna szata graficzna jest w wersji mocno "technicznej" – do wersji finalnej ulegnie znaczącej zmianie)

Górna część ekranu głównego wskazuje aktualną pozycję lokalizowanego użytkownika poprzez wyświetlanie informacji o jego współrzędnych oraz piętrze na którym się znajduje.

Ekran główny posiada wysuwane z lewej strony menu z poniższymi (na chwilę obecną) funkcjonalnościami:

- Localize me - rozpoczyna pobieranie danych z sensora WiFi i uaktualnia bieżącą pozycję użytkownika wysyłając je do serwera. Opcja domyślna po wejściu w ekran główny.
- Localize a friend - pobiera i wyświetla dane o lokalizacji osoby wskazanej przez użytkownika.

- History - wyświetla listę ostatnich współrzędnych wraz z fragmentem mapy na której są one zaznaczone.
- WiFi data - wyświetla ostatnio zeskanowany przez sensor wektor sił sygnałów sieci WiFi.

Ważniejsze klasy klienta

- **HttpClient**
Klasa umożliwiająca wywoływanie metod usługi sieciowej poprzez zapytania HTTP. Z racji, że odpowiedzi serwera nie są natychmiastowe, wykorzystuje ona callbacki pozwalające na definiowanie handlerów w razie ewentualnego przyjścia danych. Ponadto, pozwala na zdefiniowanie timeoutu na nawiązanie połączenia i czytanie danych z bufora.
- **GisServiceClient**
Wrapper dla klasy *HttpClient* definiujący konkretne metody odpowiadające usługom serwera (*getLocation()*, *getLocationMap()*, etc.). Implementuje wzorzec Singleton, więc jest dostępny z każdego miejsca aplikacji.
- **WifiDataProvider**
Singleton skanujący BSSID i siły sygnałów access pointów. Umożliwia ustawienie listenera na kolejne porcje danych, dzięki czemu dane do serwera mogą być wysyłane tak szybko, jak zmienia się lokalizacja użytkownika. Umożliwia też sprawdzenie stanu i włączenie WiFi (jeżeli user o tym zapomniał).