



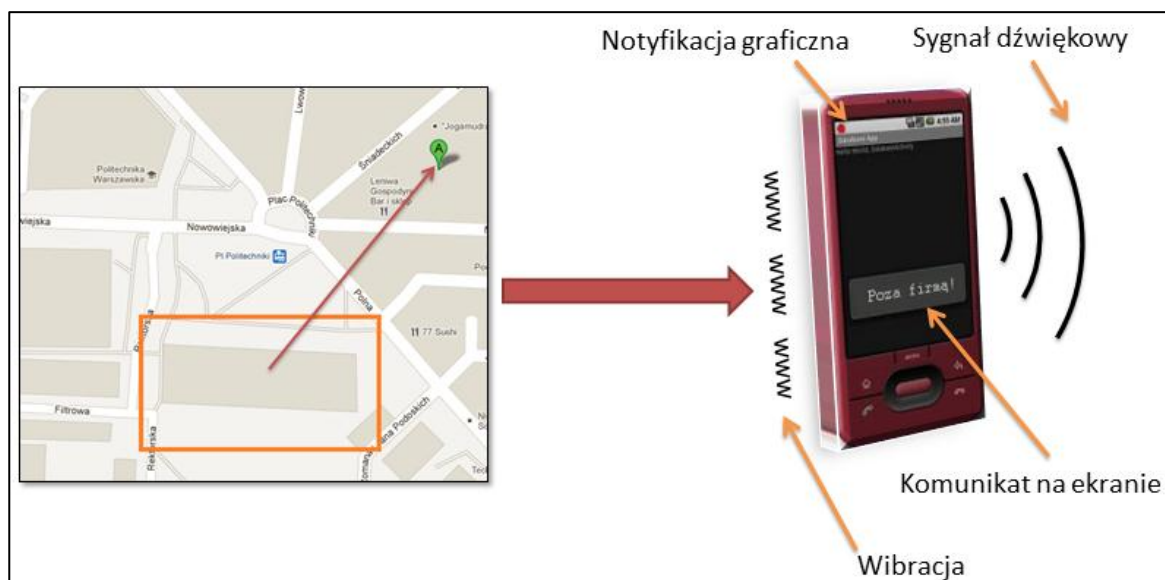
PIOTR JASTRZĘBSKI • PIOTR BARTOSZUK • PIOTR OKUŁA • WOJCIECH KACZOROWSKI

PROJEKT TIN 2011 • OPIEKUN: DR INŻ. PIOTR GAWKOWSKI

Opis funkcjonalności	3
Wymagania funkcjonalne	4
Wymagania niefunkcjonalne	4
Architektura	5
Protokoły i rozwiązywanie sytuacji krytycznych.....	6
Testowanie.....	7
Podział pracy.....	8

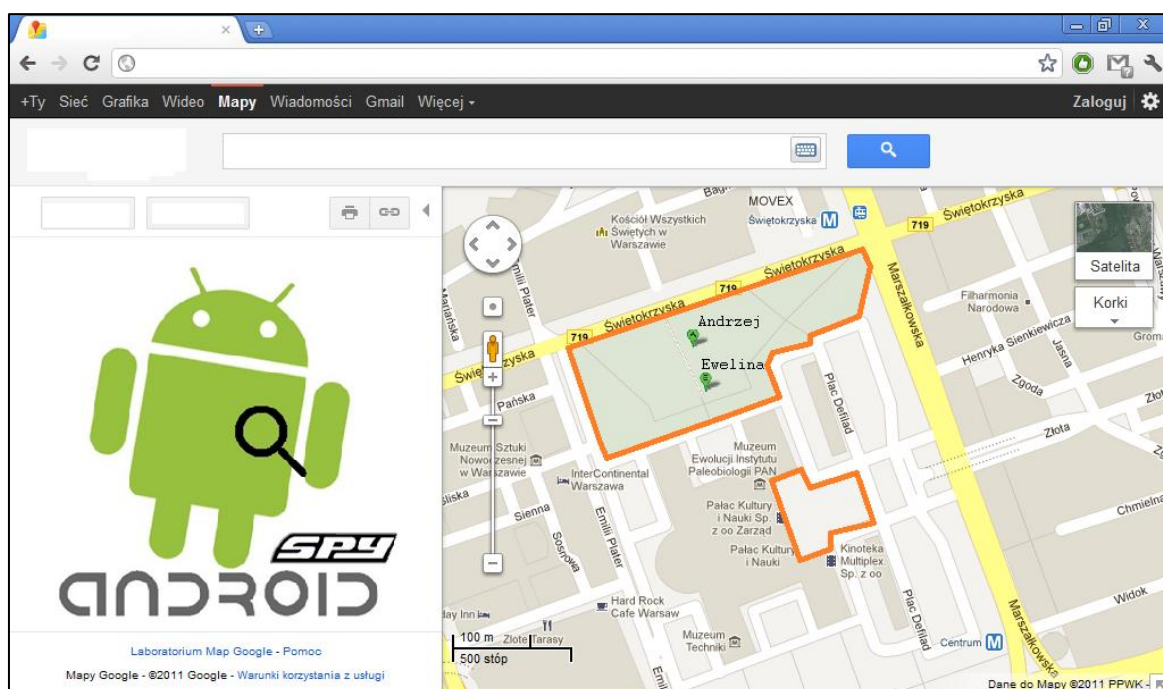
Dla końcowego użytkownika widoczne będą 2 z 3 przygotowanych modułów.

1. Aplikacja Android – program działający w tle w formie *Service* nie wyświetlający żadnych informacji do momentu opuszczenia przez użytkownika wyznaczonej strefy. Na podstawie przesyłanych komunikatów o współrzędnych położenia i otrzymanych od serwera ograniczeniach wynikających z zasięgu strefy aplikacja stwierdza czy użytkownik znajduje się w dozwolonym obszarze, czy też nie. Jeśli nie – telefon poinformuje go o tym wibracją, sygnałem dźwiękowym, monitem w formie *Android Toast* oraz notyfikacją graficzną na głównej belce systemu.



W momencie opuszczenia dozwolonej strefy, użytkownik ostrzegany jest o tym fakcie przez aparat telefoniczny. W podglądzie aplikacji zdalnej operator może śledzić jego położenie na mapie.

2. Aplikacja webowa – zapewnia możliwość przedstawienia graficznego na mapie pozycji użytkowników telefonów. Ułatwia wyznaczanie obszarów dozwolonych (tu – pomarańczowe wielokąty). Zapewnia możliwość wysłania komunikatów przywołania.



Przykładowy interfejs aplikacji webowej. Zapewnia możliwość wyznaczania obszarów i śledzenia użytkowników.

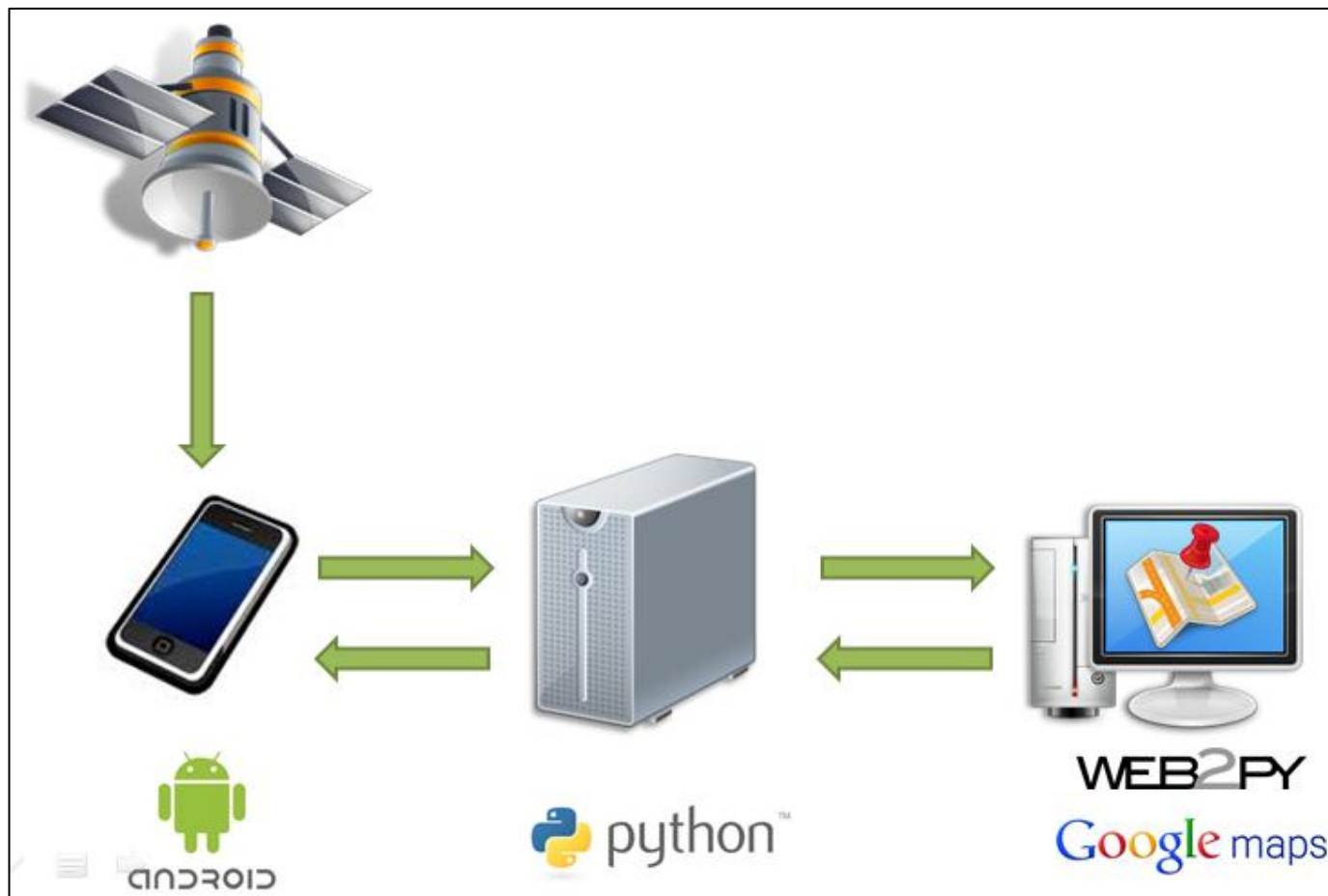
WYMAGANIA FUNKCJONALNE

ID	Funkcja	Opis	Priorytet
1	Funkcje dotyczące telefonu		
1.1	Zbieranie współrzędnych	Zbieranie koordynatów zarówno z GPSu, jak i ze wskazań widoczności nadajników sieci Wi-Fi w celu ustalenia maksymalnie dokładnej pozycji	Wysoki
1.2	Analiza położenia	W zależności od położenia zmiana częstotliwości wysyłania danych do serwera. Analizowanie czy telefon jest nadal w strefie dozwolonej	Wysoki
1.3	Informowanie o opuszczeniu strefy	W momencie opuszczenia strefy włączenie wszystkich mechanizmów informowania użytkownika o konieczności powrotu.	Średni
2	Funkcje dotyczące serwera		
2.1	Komunikacja	Nawiązania i obsługa komunikacji dwustronnej Serwer – Telefon oraz Serwer – Aplikacja webowa	Wysoki
3	Funkcje dotyczące aplikacji webowej		
3.1	Aktualizacja na mapie	Funkcja odwzorowująca pozycję śledzonych użytkowników na mapie	Wysoki
3.2	Wyznaczenie obszaru	Wyznaczenie wielokąta stanowiącego obszar, w którym będzie mógł przebywać użytkownik bez wywołania alarmu opuszczenia strefy	Wysoki
3.3	Przywołanie użytkownika	Wprowadzenie i wysłanie do zadanego użytkownika komunikatu	Niski
3.4	Synchronizacja danych	Przesłanie danych i informacji wprowadzonych po stronie aplikacji webowej w stronę serwera pośredniczącego	Wysoki

WYMAGANIA NIEFUNKCJONALNE

- Racjonalny czas odpowiedzi aplikacji (zarówno po stronie telefonu jak i aplikacji webowej), umożliwiającą pracę interaktywną.
- Przez zastosowanie wielowątkowego serwera możliwość komunikacji z większą ilością użytkowników i równoległe przetwarzanie otrzymywanych informacji.
- Możliwość niezawodnego gromadzenia danych niezależnie lub zależnie w mniejszym stopniu od widoczności satelitów na niebie (pozycjonowanie na podstawie analizy sygnału sieci WLAN)
- Graficzny interfejs użytkownika musi podnieść walory użytkowe programu i ułatwić wdrożenie. Prezentacja danych i codzienna obsługa musi być intuicyjna i prosta, a z każdą opcją powiązana powinna być czytelna ikona lub prosty, jednoznaczny opis.
- Niskie wymagania co do pamięci RAM – minimalny interfejs typu *Android Service*
- Zmienne co do czasu interwały przesyłania informacji w sieci do serwera, uzależnione od przebywania w środku lub na zewnątrz strefy dozwolonej
- Ze względu na koszty, zredukowana do minimum ilość informacji przesyłanych przez sieć

Projekt Android Spy Application jest złożony z kilku elementów: aplikacji klienckiej uruchamianej na telefonie z systemem Android oraz części serwerowej odpowiedzialnej za przetrzymywanie i przesyłanie informacji między modułami. Nieodłączną częścią systemu jest także aplikacja webowa pozwalająca za pośrednictwem serwera utrzymywać dwustronną komunikację z telefonem użytkownika.



Schemat połączeń i kierunku wymiany informacji pomiędzy modułami.

Wymagania co do platformy:

- web2py w wersji 1.66.2 lub wyższej
- Python w wersji 2.7.2 lub wyższej
- System operacyjny Androd w wersji 1.5 lub wyższej

Protokoły komunikacyjne:

1. System GPS → Smartfon

- Protokół jednostronnej komunikacji GPS, wymaga widoczności minimum 4 satelitów
- W przypadku braku widoczności satelitów lub wyłączonej funkcji GPS, system Android potrafi ustalić z mniejszą dokładnością pozycję na podstawie widoczności access-pointów sieci Wi-Fi

2. Smartfon ↔ Serwer

- Komunikacja HTTP – informacje przesyłane zgodne z typem MIME: plain/text lub multipart jeśli wyniknie konieczność wprowadzenia przesyłu obiektu wieloczęściowego
- W celu utrzymania wysokiej kultury protokołu komunikacyjnego mechanizm obsługi wyjątków, w przypadku utraty połączenia, odebrania złych danych itp., zapewni wyświetlenie informacji o błędzie w przyjaznej użytkownikowi formie (Android toast)

3. Serwer ↔ Aplikacja webowa

- Komunikacja HTTP – ze względu na fakt, że aplikacja webowa jest tak naprawdę zwykłą stroną internetową, w przypadku problemów system standardowych komunikatów o błędach zwróci odpowiedni kod (404, 502) lub zapewniony zostanie własny mechanizm informowania o błędach.
- Jeśli problem wyniknie w części serwerowej aplikacji webowej, wszystkie komunikaty o błędach otrzymują zgodnie ze specyfikacją web2py odpowiedni ticket błędu, co znacznie upraszcza rozwiązywanie problemów, a jednocześnie nie prowadzi do przechowywania, przetrzymywania informacji niespójnych.

Charakter aplikacji wymaga testowanie jej w warunkach zbliżonych do naturalnych tj. w miejscu gdzie istnieje widoczność dostatecznej ilości satelitów i gdzie będzie zapewnione połączenie z serwerem. Dla łatwiejszego testowania, zanim zaczniemy etap końcowy na rzeczywistych urządzeniach (większa niepewność pomiarowa, wpływ innych czynników zewnętrznych), wykorzystamy możliwości wirtualnej maszyny Android. Dostarczony przez firmę Google Android Development Kit posiada mechanizm maszyn wirtualnych dla każdej wersji systemu Android. Taka symulacja zapewnia dostęp do wszystkich ustawień, charakterystyki pracy i właściwości systemu Android. Istnieje także, poprzez wtyczkę do programu Eclipse, możliwość dowolnego symulowania położenia poprzez podanie odpowiednich współrzędnych geograficznych (długości i szerokości).

Do testowania aplikacji wykorzystamy zestaw złożony z komputera, telefonu z systemem Android oraz routera Wi-Fi (komunikacja zgodnie z założeniami aplikacji może odbywać się przy wykorzystaniu sieci telefonicznej, ale sieć WLAN znacznie obniży koszt testów). Aktualnie mamy możliwość sprawdzenia aplikacji pod kątem zgodności z różnymi telefonami i wersjami systemu z jednym z następujących urządzeń:

- Samsung Galaxy S Plus (Android 2.3.3)
- HTC Wildfire (Android 2.2.1)
- HTC Desire (Android 2.2.1).

W celu przetestowania równoległej obsługi wielu użytkowników końcowych przez wielowątkowy serwer, przygotowany zostanie prosty symulator wielowątkowy w Javie (ze względu na duże podobieństwo użytych mechanizmów w odpowiadającej wersji Javy pod Androidem), symulujący dużą liczbę użytkowników, który będzie odpytywał serwer.

Podział pracy ze względu na modułowy charakter projektu jest łatwy do ustalenia. Jedynie część dotycząca aplikacji na system Android, ze względu na dużą złożoność i konieczny nakład pracy została podzielona na 2 osoby. Sam przebieg pracy nad aplikacją smartfonową, jak i związany z nią zakres odpowiedzialności będzie modyfikowany na bieżąco w toku prac.

Piotr Jastrzębski:

- Dokumentacja wstępna i końcowa
- Moduł aplikacji smartfonowej: Android – obustronna komunikacja telefon – serwer, wysyłanie współrzędnych i odbieranie zakresów obszarów dozwolonych, interfejs graficzny, część odpowiedzialna za zbieranie danych z czujników i odpowiednie ich przetwarzanie oraz analizowanie.

Piotr Bartoszek:

- Moduł serwerowy: Python – odpowiednie obsłużenie informacji przychodzących z telefonu oraz od strony aplikacji web2py, aktualizacje stanu przechowywanych danych, komunikacja zwrotna (alert, nowy nieprzekraczalny obszar pobytu, wiadomość).

Piotr Okuła:

- Moduł aplikacji webowej: web2py + API Google Maps – przedstawienie danych o użytkowniku w sposób najbardziej czytelny i jednoznaczny za pomocą API Google Maps. Dodanie możliwości zdalnego ustalania obszaru pobytu użytkownika telefonu i wymuszenie przywołania.

Wojciech Kaczorowski:

- Moduł aplikacji smartfonowej: Android – jak w punkcie pierwszym.