POLITECHNIKA WARSZAWSKA

Wydział Elektroniki i Technik Informacyjnych

Instytut Telekomunikacji

**PRACA DYPLOMOWA**

**INŻYNIERSKA**

**Sebastian Łuczak, Maciej Nowak**

**TYTUŁ**

Kierujący pracą:

mgr inż. Marcin Golański

..........................................

ocena pracy

..........................................

data i podpis Przewodniczącego

Komisji Egzaminacyjnej

Warszawa, wrzesień 2011

## SPIS TREŚCI :

1. Wstęp
2. Cel i zakres pracy
3. wstęp teoretyczny
   1. opis technologii
      1. android
      2. nexus
      3. oracle
      4. nfc
      5. J2EE
      6. Spring
      7. Hibernate
      8. Jersey
      9. Restful application – json
   2. przyczyny doboru ?
4. koncepcja pracy
5. opis systemu
   1. scenariusz użycia
   2. opis telefonu
      1. aplikacja rejestrująca
      2. aplikacja administratora
   3. opis Web-service – usługa internetowa
      1. 4 pod serwisy
   4. opis bazy danych
   5. opis panelu administracyjnego
6. korzyści płynące z rozwiązania
7. podsumowanie
8. wnioski
9. bibliografia

## Cel i zakres pracy

Celem naszej pracy inżynierskiej jest zrealizowanie kompletnego i nowatorskiego systemu kontroli czasu pracy z przeznaczeniem dla pracowników mobilnych. Innowacyjne wykorzystanie wchodzącej na rynek technologii NFC (Near Field Communication) , telefonii komórkowej oraz usług internetowych czyni nasze rozwiązanie unikalnym na rynku.

Dzięki infrastrukturze telefonii 3G, oraz złożonemu systemowi monitorowania uzyskujemy pełną i pewną kontrolę nad czasem poświęconym przez pracownika na realizację powierzonych mu zadań.

Kompletna realizacja zakłada dostarczenie zestawu aplikacji współpracujących ze sobą, na który składają się: baza danych, aplikacja serwerowa będąca usługą sieciową, internetowy portal informacyjny o przeznaczeniu administracyjnym oraz dwie łączące się z nimi aplikacje mobilne (kliencka i administracyjna) na telefon komórkowy z systemem Android. Każdy z tych elementów zostanie przedstawiony oddzielnie w dalszej części pracy.

Technologie użyte do wykonania pracy są powszechnie wykorzystywane do wytwarzania oprogramowania biznesowego przeznaczonego dla dużej ilości jednoczesnych użytkowników i zgodne z powszechnymi trendami realizacji usług w oparciu o sieć Internet.

Prezentowana realizacja ma być w założeniu alternatywą dla istniejących już systemów opartych o architektury stacjonarne, zrealizowane przy pomocy stacjonarnych czytników RFiD i kart zbliżeniowych, a także rozwiązaniem problemu kontroli czasu pracy osób których profil zatrudnienia zakłada stałe przemieszczanie się po kraju w celu serwisowania urządzeń bezpośrednio u klienta.

Dotychczasowe systemy ze względu na złożoną, specyficzną i kosztowną infrastrukturę nie odpowiadały na potrzeby dużej grupy pracodawców, którzy współpracę z kontrahentami opierają na umowach okresowych (serwis urządzeń, wsparcie techniczne IT itp.), a wynagrodzenie pracowników uzależniają od czasu poświęconego na wykonanie zlecenia. W takiej sytuacji potrzebna jest duża łatwość modyfikacji rozmieszczenia punktów kontrolnych, oraz ich bezproblemowa instalacja u kontrahentów, nie wymagająca żadnych narzędzi i ingerencji w otoczenie, a także zapewnienie transparentności i łatwości użytkowania dla pracowników.

Zarówno koncepcja jak i projekt aplikacji został wymyślony i zrealizowany wspólnie. Ze względu na modułowość zaprojektowanego przez nas rozwiązania, dokonaliśmy podziału pracy nad implementacją systemu na dwie części – mobilną i stacjonarną, leżącą po stronie serwera.

Aplikacje przeznaczone na telefon komórkowy stworzył Sebastian Łuczak, z kolei aplikacje usługi internetowej oraz portalu administracyjnego Maciej Nowak.

W dalszej części pracy przedstawione zostaną nasze rozwiązania na temat scenariuszu użycia kompletnego systemu, przyczyny doboru poszczególnych technologii oraz powody zastosowania konkretnych technik implementacji.

## Wstęp teoretyczny

## Motywacja wyboru rozwiązań – telefon

Jednym z kluczowych czynników doboru technologii w której zrealizowana miała być ta część pracy, była popularność danego rozwiązania na rynku konsumenckim tak, aby uzyskany system miał również zastosowanie pozanaukowe. Po zapoznaniu się z obecnym podziałem rynku (PRZYPIS DO TABELKI Z JAKIMIŚ ANALIZAMI) okazało się, że większość obecnie sprzedawanych urządzeń typu smartphone, wyposażona jest w system Android. Dodatkowo, po obserwacji dostępności aparatów ze wsparciem NFC, okazało się, że również w tym wypadku system ten wiedzie prym pod względem jakości dostarczonego API i uniwersalności (DOWODY), a także liczby zapowiedzianych modeli telefonów.

W efekcie tej analizy, zdecydowaliśmy się na pozyskanie jedynego wówczas dostępnego telefonu tego typu, a mianowicie Google Samsung Nexus S z systemem operacyjnym Android Gingerbread 2.3.

Decyzja ta powodowana była również uniwersalnością i łatwością programowania na tę platformę, dzięki czemu można było skupić się na samym rozwiązaniu, a nie walce z narzędziami.

W pierwszej fazie projektu, dodatkowym problemem był fakt iż system Android Gingerbread w wersji 2.3 nie dostarczał interfejsu programistycznego pozwalającego na pełną obsługę NFC. Sami twórcy przyznali się do faktu iż funkcje NFC w tej wersji były sztucznie ograniczone: niemożliwe między innymi było zapisywanie informacji na Tag, obsługiwana była mniejsza liczba dostępnych technologii, nie istniała możliwość obsłużenia zbliżonego Tagu przez aplikację obecnie uruchomioną (system zawsze pytał się która aplikacja ma być wywołana dla danego Tagu).(DOWÓD)

Początkowy projekt aplikacji zakładał występowanie tych ograniczeń, szybko jednak okazało się, że w aktualizacji 2.3.3 (DATA) wspomniane funkcje zostały dostarczone, a dostępne wcześniej API, zmodyfikowane i w wielu przypadkach zauważalnie poprawione (DOWÓD).

Ponieważ kolejnym założeniem które przyświecało projektowi była prostota i szybkość działania, postanowiliśmy jednak nie zmieniać założeń w sposób drastyczny, gdyż nie było ku temu bardzo wyraźnych powodów.

Uproszczony opis zadań powierzonych aplikacji na telefon zawiera w sobie:

1. automatyczny odczyt zbliżonego Tagu w sytuacji w której aplikacja jest włączona
2. zgłoszenie się do odczytu Tagu zgodnego z aplikacją z poziomu menu systemu operacyjnego
3. uwierzytelnienie użytkownika i automatyzacja logowania z zachowaniem oczekiwanej kontroli tożsamości
4. Komunikacja z użytkownikiem w oparciu o graficzny interfejs
5. Duża szybkość reakcji na działania użytkownika
6. Komunikacja z serwerem w celu obsługi odczytanego Tagu
7. Obsługa zdarzeń które miały miejsce przy braku dostępu do sieci Internet

W celu maksymalnego, możliwego odciążenia aplikacji na telefon, zgodnie z ideologią pisania aplikacji na powyższy system (DOWÓD), cała logika systemu zrealizowana jest w aplikacji serwerowej, telefon z kolei formułuje zapytania i przetwarza odpowiedzi. Komunikacja w architekturze REST (Representational State Transfer – literatura) opiera się o zapytania w formie JSON (JavaScript Object Notation). Bezpośrednio oznacza to, że aplikacja zgłasza do serwera zapytanie w momencie gotowości do przejścia w inny stan działania (np. przedstawienie obecnego stanu zgłoszenia). Forma tego zapytania jest ściśle znana i wysyłana w sposób jawny w formacie JSON. Zaletą tego formatu jest ogólnodostępność i niezależność od języka programowania, idące jednak w parze z kompletnym wsparciem.

Przebieg wywołania aplikacji.

Weryfikacja autentyczności użytkownika, z powodu upatrywanych zagrożeń w postaci ludzkiej pomysłowości, Nie mogła przebiegać z użyciem samej nazwy użytkownika i hasła. Z tego powodu do grupy danych którymi musi się uwierzytelnić aplikacja kliencka, doszły numer IMEI oraz IMSI (rozwinięcie skrótów), dzięki czemu upewniamy się, że użytkownik o danym identyfikatorze zgłasza swoją aktywność przy pomocy swojego telefonu (numer IMEI) oraz swojej karty SIM (numer IMSI), co eliminuje zagrożenia płynące z wymiany telefonami i danymi autoryzacyjnymi pomiędzy użytkownikami systemu. Dodatkowo dane przesyłane do serwera są poddane działaniom kryptograficznym w postaci szyfrowania symetrycznego AES 256bit (hmmm???), dzięki czemu użytkownik nie może z poziomu innego terminala *udawać*, że jest użytkownikiem systemu.

W chwili obecnej, z przyczyn finansowych, nie zapisujemy nic na Tag w postaci „tylko do odczytu” i opieramy działanie systemu o unikatowe identyfikatory tagów, które powiązane są w bazie z daną lokalizacją.

Ponieważ jedną z mocnych stron prezentowanego systemu jest jego cena wdrożeniowa i eksploatacyjna, zdecydowaliśmy się na wykorzystanie tanich i szybkich tagów Mifare Ultralight typu NFC Type 2 w formie giętkiej nalepki wprasowanej w papier.

Znaczniki te cechują się pojemnością (ILE) bitów, co pozwala na zapis (Tego i owego), a także dostarczają możliwość jednokrotnego zablokowania zapisu na znaczniku. Wadą tego rozwiązania jest jednak niemożność zmiany kodów dostępu do odczytu, przez co wszystko co jest zapisane na znaczniku, może być odczytane przez aplikację zgodną z NFC. Z tego powodu, dane te zapisywane są w sposób zaszyfrowany, a jest identyfikator lokalizacji Tagu.

W naszym rozwiązaniu dokonaliśmy odwrócenia klasycznego scenariusza w którym czytnik RFiD ulokowany jest na stałe w ścianie lub bramce, a użytkownik posiada kartę RFiD. Sytuacja ta ma ekonomiczne uzasadnienie w sytuacji w której mówimy o stałej lokalizacji pracowników i ich dużej liczbie. Problemy pojawiają się jednak gdy lokalizacji jest wiele (tyle ilu klientów) i mogą się zmieniać, pracownicy stale poruszają się pomiędzy nimi, a pracodawcy zależy na informacjach na temat jakości obsługi klienta i sumienności swoich pracowników. W takiej sytuacji drogi czytnik musiałby zostać umieszczony w dziesiątkach jak nie setkach punktów, a całość musiałaby być spięta w dodatkową sieć zarządzającą dostępem.

W sytuacji w której czytnikiem jest telefon, a zatem urządzenie w które z powodów cywilizacyjnych każdy pracownik tego typu jest wyposażony, nie ma problemu związanego z dodatkowym kosztem (gdyż zawiera się w cenie urządzenia domyślnie używanego przez pracownika), a niezwykle (CENA!) tanie Tagi RFiD zgodne ze standardem NFC rozmieszczane zostają na każdym z serwisowanych urządzeń, powodujemy zmianę rozkładu kosztów i zwiększenie szczegółowości informacji (gdyż wiemy dokładnie nad którym zleceniem pracuje w danym momencie dany pracownik).

Dane transmitowane są w czasie rzeczywistym, dzięki czemu koordynator zna obecny stan zajętości zasobów ludzkich, a także jest w stanie ustalić z większą dokładnością średni czas realizacji danej usługi serwisowej.

Oczywiście zauważalną wadą zaproponowanego rozwiązania jest wrażliwość na dostępność do usług transmisji danych, niestety jednak wprowadzenie opcji pracy w trybie offline mogłoby spowodować ogromną wrażliwość na przekłamanie danych docierających do bazy, a raczej celową modyfikację daty systemowej przez użytkowników. Co prawda baza przechowuje dla każdego wpisu zarówno znacznik czasowy terminala użytkownika jak i serwera obsługującego zdarzenie, jednak zakładając działanie w trybie offline i kolejkowanie wysyłania danych do serwera, użytkownik mógłby celowo emulować przebywanie poza zasięgiem sieci tak, aby zamaskować opóźnione rozpoczęcie lub zakończenie realizacji zlecenia.

Podczas testów wydajnościowych stwierdzono, ze aplikacja nie ma praktycznie żadnego wpływu na zużycie baterii, tak samo jak stale włączony moduł NFC (zużycie poniżej 0.1% baterii).

Aplikacja administracyjna używana jest do rejestracji lokalizacji przez administratora systemu. Za jej pomocą wiąże on dany Tag z klientem i konkretnym urządzeniem.