|  |  |
| --- | --- |
| POLITECHNIKA WARSZAWSKA  Wydział Elektroniki i Technik Informacyjnych | Rok akademicki  2013/2014 |

**PRACA DYPLOMOWA MAGISTERSKA**

Adam Turski

**System śledzenia przesyłek w chmurze z użyciem technologii RFID**

Praca wykonana pod kierunkiem

dr inż. Piotra Witońskiego

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

ocena pracy

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

podpis Przewodniczącego Komisji

Warszawa 2014

# Życiorys

**Adam Turski**

Urodziłem się 24 marca 1986 roku w Warszawie. W latach 2001-2006 uczęszczałem do Technikum Mechatronicznego im. Stanisława Staszica w Siedlcach. Po ukończeniu technikum rozpocząłem studia 1-go stopnia w Instytucie Informatyki Akademii Podlaskiej w Siedlcach. W roku 2009 obroniłem swoją pracę licencjacką oraz rozpocząłem uzupełniające studia magisterskie na Wydziale Elektroniki i Technik Informacyjnych. Z początkiem roku 2013 zatrudniłem się w firmie Enigma SOI jako programista Javy. Zdobytą na studiach wiedzę wykorzystuje na co dzień w swojej pracy zawodowej.

Prywatnie interesuje się kinematografią.

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

podpis studenta

**EGZAMIN DYPLOMOWY**

Złożył egzamin dyplomowy w dniu ............................................................................... 200.....r.

z wynikiem ..................................................................................................................................

Ogólny wynik studiów .................................................................................................................

Dodatkowe wnioski i uwagi Komisji ............................................................................................

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

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

# Streszczenie

JSystem śledzenie przesyłek Enterprise Edition jest serwerową platformą do tworzenia aplikacji w języku Java opartych o wielowarstwową architekturę komponentową. Dzięki niej proces wytwarzania aplikacji webowych sprowadza się do wykorzystania gotowych, sprawdzonych komponentów. Nasze siły wówczas możemy skupić na realizacji założonego celu biznesowego. Takie podejście poprawia jakość oprogramowania oraz skraca czas jego realizacji. Otwarta architektura pozwala obniżyć całkowity koszt projektu jednocześnie ogranicza ryzyko uzależnienia od jednego dostawcy.

RFID jest technologią identyfikacji obiektów z wykorzystaniem fal radiowych. Dane identyfikacyjne są zapisane na etykiecie (transponder), która posiada wbudowaną antenę. Czytnik przy pomocy anteny nadajnika odczytuje dane z transpondera na odległość. Technika ta, w odróżnieniu od innych metod identyfikacyjnych np. kodów kreskowych, pozwala zautomatyzować proces identyfikacji oraz ograniczyć prawdopodobieństwo błędu ludzkiego

Celem mojej pracy magisterskiej było stworzenie skalowalnego systemu JEE z wykorzystaniem techniki RFID. Założenie to zrealizowałem tworząc system śledzenia przesyłek opatrzonych transponderami. System ten z powodzeniem może zostać wykorzystany w firmach z branż pocztowych oraz kurierskich.

**Abstract**

# Spis treści

[Życiorys 2](#_Toc372923070)

[Streszczenie 3](#_Toc372923071)

[Spis treści 5](#_Toc372923072)

[1 Wstęp 6](#_Toc372923073)

[Cel i zakres pracy 7](#_Toc372923074)

[2 Wprowadzenie do systemów JEE oraz technologii RFID 8](#_Toc372923075)

[2.1 8](#_Toc372923076)

[3 Opis systemu 9](#_Toc372923077)

[3.1 Wymagania funkcjonalne 9](#_Toc372923078)

[3.1.1 Wymagania funkcjonalne dla użytkownika 11](#_Toc372923079)

[3.1.2 Wymagania funkcjonalne serwisów 11](#_Toc372923080)

[4 Architektura 13](#_Toc372923081)

[4.1 Warstwa bazodanowa 13](#_Toc372923082)

[4.1.1 Projekt bazy danych 13](#_Toc372923083)

# Wstęp

W dzisiejszych czasach bardzo popularne jest powiedzenie "*Kto stoi w miejscu ten się cofa*". Chociaż prawda ta była, jest i będzie aktualna to jednak obecne czasy najlepiej to odzwierciedlają. Dużo firm traci swoją pozycję na rynku z powodu braku zmian w procesach firmy, począwszy od procesu technologicznego po wizerunek firmy. Techniki, które wczoraj były wystarczające, dziś wymagają zmiany.

Drugim popularnym powiedzeniem

Niniejsza praca magisterska opisuje architekturę oraz implementację systemu śledzenia przesyłek skalowalnego systemu zrealizowanego w standardzie JAVA EE[[1]](#footnote-1) z użyciem technologii RFID[[2]](#footnote-2). Aplikacje wykonane w standardzie JEE są opartego na wielowarstwowej architekturze komponentowej.//TODO(więcej o technologii JEE)

Technologia RFID, czyli identyfikacja za pomocą "tagów" //TODO(co to jest). Pozwala ona na zastąpienie innych typów identyfikacji np. kodów kreskowych, co przekłada się //TODO(na co).

## Cel i zakres pracy

Głównym celem niniejszej pracy było zaprojektowanie oraz implementacja skalowalnego systemu z wykorzystaniem technologii RFID w oparciu o technologię Java Enterprise Edition. Aby zrealizować to założenie został opracowany system wspomagający śledzenie przesyłek TRAK wykorzystujący transpondery jako znaczniki przesyłek.

W pierwszych rozdziałach zostaną omówione technologie JEE oraz RFID, a także proces śledzenia przesyłek. Kolejne przedstawiają architekturę oraz implementację systemu.

Rozdział 1. opisuje serwerową platformę programistyczną JEE jako standard tworzenia aplikacji opartych o wielowarstwową architekturę komponentową. Przedstawia technologię RFID w świetle obecnych zastosowań.

Rozdział 2. analizuje proces śledzenia przesyłek, a także dostępnych na rynku podobnych rozwiązań.

Rozdział 3. opisuje koncepcję oraz projekt platformy TRAK. Zostaną omówione wymagania stawiane systemowi, przedstawiony zostanie model konceptualny.

Rozdział 4. przedstawia narzędzia oraz biblioteki za pomocą których wynikowy system został zrealizowany.

Rozdział 5.

# Wprowadzenie do systemów JEE oraz technologii RFID

## 

# Projekt systemu

Systemem opisywanym w tej pracy magisterskiej będzie system śledzenia przesyłek TRAK. System ten pozwoli Aplikacja ta służy do rejestracji oraz śledzenia przesyłek.

TODO: Więcej co to bedzie robilo

System jest dedykowany dla firm z branż pocztowych oraz kurierskich. Pomysł na stworzenie takiego systemu przyszedł autorowi na myśl podczas pobytu w punkcie pocztowym i obserwacji nadawaniu paczek wykorzystując technologię kodów kreskowych. Technologia kodów kreskowych w dobie dzisiejszych możliwości ustępuje technologii RFID, jednocześnie wymaga większego zaangażowania zasobów ludzkich przy obsłudze takiego rodzaju identyfikacji.

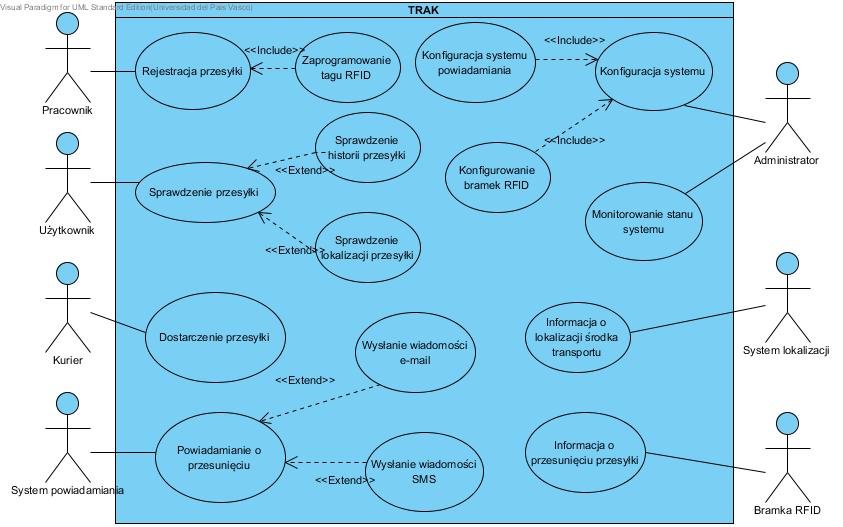
Aby lepiej zrozumieć

TODO: WIecej dla kogo to ejst

## Wymagania funkcjonalne

Analiza wymagań funkcjonalnych pozwala na zidentyfikowanie i opisanie oczekiwanego zachowania systemu. W myśl jednej z definicji, wymagania funkcjonalne to stwierdzenie, jakie usługi ma oferować system, jak ma reagować na określone dane wejściowe oraz jak ma się zachowywać w określonych sytuacjach.

Zebrane wymagania funkcjonalne zostaną przedstawione poniżej na diagramie przypadków użycia (Diagram przypadków użycia).



Rysunek Diagram przypadków użycia

Najważniejsze przypadki użycia systemu przedstawione zostały poniżej w formie formularzy wymagań funkcjonalnych.

Tabela Opis wymagań funkcji Rejestruj przesyłkę

|  |  |
| --- | --- |
| **Nazwa funkcji** | **Rejestruj przesyłkę** |
| Opis | Funkcja pozwalająca na rejestrację przesyłki w systemie |
| Dane wejściowe | Numer identyfikacyjny przesyłki |
| Źródło danych wejściowych | Pracownik poczty |
| Wynik | Rejestracja przesyłki w systemie |
| Warunek wstępny | Przesyłka została dostarczona do punktu pocztowego w celu nadania |
| Warunek końcowy | Przesyłka została otagowana znacznikiem RFID |
| Powód | Śledzenie przesyłki |

Tabela Opis wymagań funkcji Zmień status przesyłki

|  |  |
| --- | --- |
| **Nazwa funkcji** | **Zmień status przesyłki** |
| Opis | Funkcja pozwalająca na zmianę statusu przesyłki |
| Dane wejściowe | Numer identyfikacyjny przesyłki, nowy status |
| Źródło danych wejściowych | System TRAK, pracownik poczty, |
| Wynik | Zaktualizowanie statusu przesyłki w systemie |
| Warunek wstępny | Wprowadzony został numer identyfikacyjny przesyłki oraz nowy status |
| Warunek końcowy | Przesyłka jest aktywna |
| Powód | Zmiana statusu przesyłki |

Tabela Opis wymagań funkcji Sprawdź status przesyłki

|  |  |
| --- | --- |
| **Nazwa funkcji** | **Sprawdź status przesyłki** |
| Opis | Funkcja pozwalająca na sprawdzenie statusu przesyłki |
| Dane wejściowe | Numer identyfikacyjny przesyłki |
| Źródło danych wejściowych | Nadawca przesyłki |
| Wynik | Informacja o statusie przesyłki |
| Warunek wstępny | Wprowadzony został numer identyfikacyjny przesyłki |
| Warunek końcowy | Wprowadzono dane weryfikacyjne |
| Powód | Sprawdzenie statusu przesyłki |

### Wymagania funkcjonalne dla użytkownika

Wymagania funkcjonalne dla użytkownika to:

* Sprawdzenie statusu przesyłki

### Wymagania funkcjonalne serwisów

Wymagania funkcjonalne dla serwisów to:

* zmiana statusu przesyłki

# Architektura

## Architektura systemu

System śledzenie przesyłek TRAK został zaprojektowany jako SaaS dla chmury obliczeniowej Google Cloud w oparciu o rozwiązanie PaaS jakim jest Google App Engine.

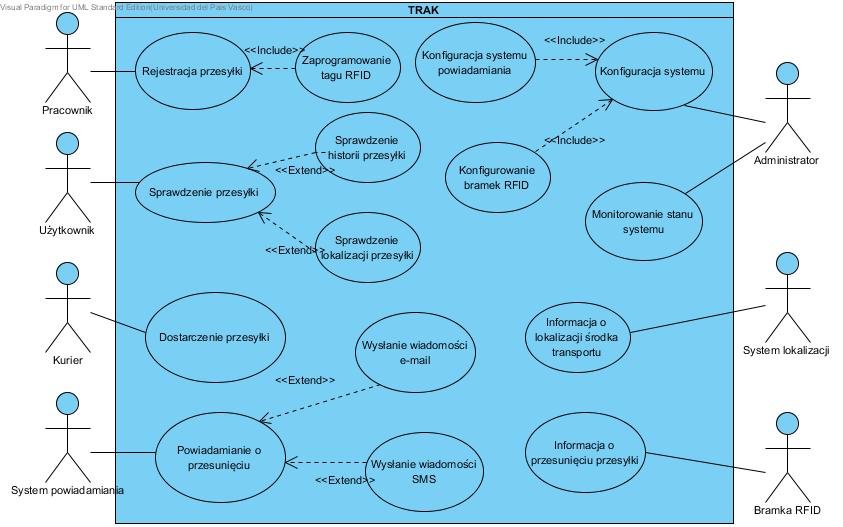
Poniższy rysunek przedstawia ogólną koncepcję systemu.



Rysunek Diagram koncepcyjny

System będzie umiejscowiony w chmurze obliczeniowej Google Cloud, która jest dostępna przez Internet. Dla użytkowników zostaną udostępnione aplikacje webowe tj. moduł administratora, moduł pracownika oraz moduł klienta. Bramki RFID oraz środki transportu będą komunikować się z systemem poprzez usługi sieciowe (webservice). Dla kurierów zostanie udostępniona osobna aplikacja na system Android, która podobnie jak bramki RFID będzie komunikować się z aplikacją poprzez usługi sieciowe.

//TODO: Coś więcej o architekturze w ramach chmury



## Warstwa bazodanowa

Warstwę bazodanową będzie pełniła usługa CloudSQL. Dzięki niej dostajemy w pełni zabezpieczoną, replikowalną w wielu geograficznych lokalizacjach, wysokiej dostępności bazę danych. CloudSQL wykonuje za nas backup bazy danych dzięki czemu możemy ją odtworzyć w razie potrzeby. Niepodważalną zaletą tej usługi jest możliwość zarządzania kosztami, tzn. w przypadku gdy nasza instancja nie jest obciążona możemy zmniejszyć parametry bazy danych i płacić mniej, a gdy będzie taka potrzeba zwiększyć je. Dodatkowo możemy regulować sposób płatności, do wyboru są płatność za faktyczne użycie bazy danych (operacje I/O) lub opłata miesięczna.

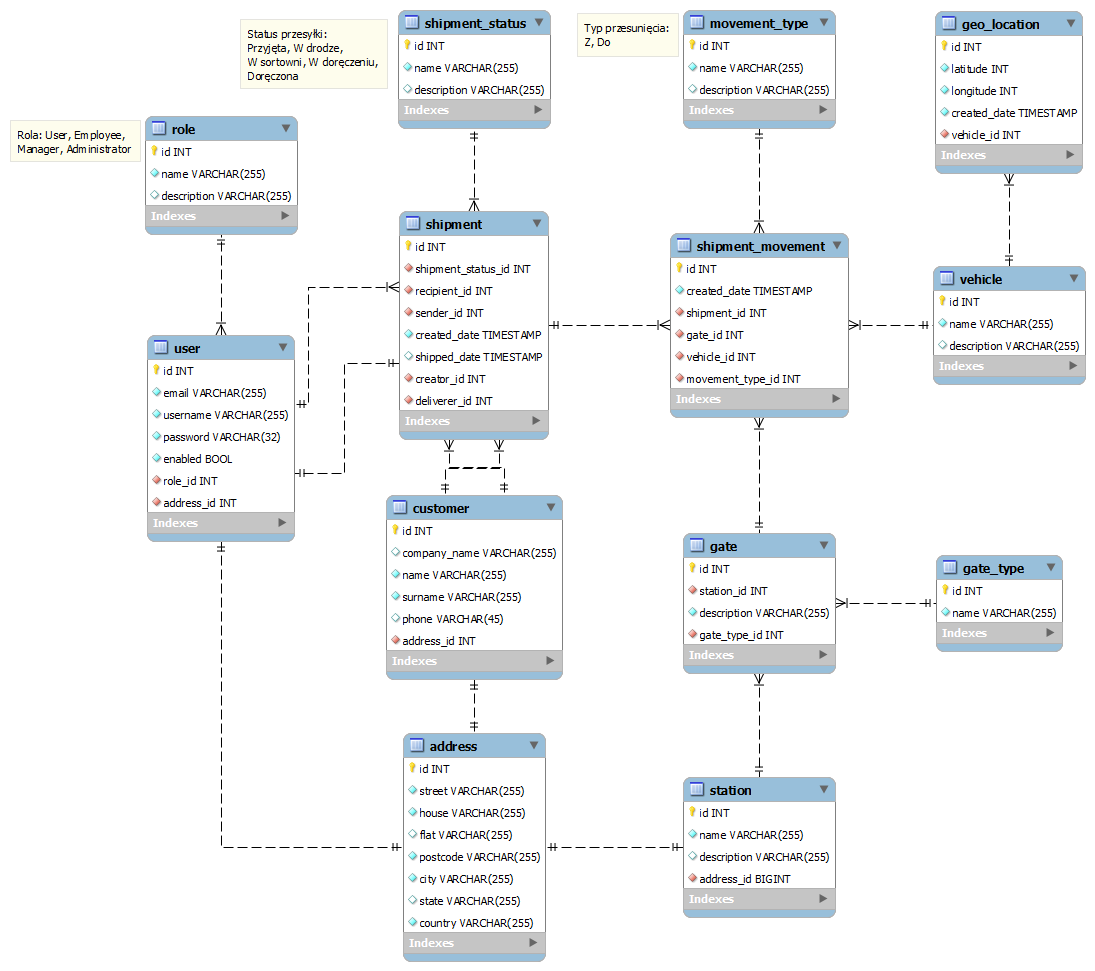
Usługa CloudSQL jest to silnik bazy danych MySQL, który jest umiejscowiony w chmurze obliczeniowej. Posiada wszystkie możliwości i funkcjonalności silnika MySQL z kilkoma dodatkowymi oraz kilkoma niewspieranymi właściwościami. Instancje bazy danych mogą mieć maksymalnie do 16GB RAM oraz do 500GB danych co znacznie przekracza potrzeby systemu TRAK.

### Projekt bazy danych

Baza danych została zaprojektowana z myślą przechowywania danych potrzebnych do śledzenia przesyłek. Z punktu widzenia śledzenia przesyłek nie ma potrzeby odzwierciedlania całej struktury jak również procesów biznesowych firmy, dlatego cały model jest nieskomplikowany i przejrzysty. Model bazodanowy przedstawia niezbędne struktury, które pozwolą na dostarczenie informacji o podstawowych danych, historii oraz lokalizacji przesyłki. Dodatkowo aby użytkownicy mogli korzystać z systemu została stworzona struktura odzwierciedlająca użytkowników systemu.

Baza danych systemu TRAK została wykonana tak aby jak najprościej umożliwić pracę jako dodatkowy moduł śledzenia przesyłek typu SaaS. Prostota ta polega na braku danych szczegółowych obiektów np. wagi przesyłki, ponieważ z punktu widzenia śledzenia przesyłek nie są one potrzebne, a na pewno są przechowywane w obecnej bazie firmy.

Poniższy diagram związków encji przedstawia pełny model bazodanowy.



Rysunek Diagram związków encji

Opis tabel:

* Tabela **role**: Przechowuje role zdefiniowane w systemie tj. USER, EMPLOYEE, MANAGER, ADMINISTRATOR
* Tabela **user**: Przechowuje informacje o użytkownikach tj. email, login, hasło, status konta, rola, dane adresowe. Pozostaje w relacji n:1 z tabelą role (użytkownik posiada jedną rolę) oraz 1:1 z tabelą address (użytkownik posiada jeden adres)
* Tabela **address**: Przechowuje dane adresowe
* Tabela **shipment\_status**: Przechowuje statusy jakie może posiadać przesyłka tj. Przyjęta, W drodze, W sortowni, W doręczeniu, Doręczona
* Tabela **customer**: Przechowuje informacje o nadawcach i odbiorcach przesyłek tj. nazwa firmy, imię, nazwisko, telefon, dane adresowe. Pozostaje w relacji 1:1 z tabelą address (dostawca/odbiorca posiada jeden adres)
* Tabela **shipment**: Przechowuje podstawowe dane o przesyłce tj. nadawca, odbiorca, status, data utworzenia, data dostarczenia, osoba doręczająca, osoba tworząca. Pozostaje w relacji n:1 z tabelą customer (przesyłka posiada jednego nadawcę i jednego odbiorcę) oraz w relacji n:1 z tabelą user (przesyłka posiada jedną osobę doręczającą oraz jedną osobę która utworzyła przesyłkę)
* Tabela **station**: Przechowuje dane o stacjach spedycyjnych firmy tj. nazwa, opis, dane adresowe. Pozostaje w relacji 1:1 z tabelą address (stacja spedycyjna może mieć jeden adres)
* Tabela **gate\_type**: Przechowuje typy bramek RFID tj. wejściowa, wyjściowa, sorter
* Tabela **gate**: Przechowuje dane o bramce RFID tj. stacja, opis, typ bramki. Pozostaje w relacji n:1 z tabelą station (bramka należy do jednej stacji spedycyjnej) oraz n:1 z tabelą gate\_type (bramka posiada jeden typ)
* Tabela **movement\_type**: Przechowuje typ przesunięcia przesyłki tj. Z lub Do
* Tabela **vehicle**: Przechowuje dane o środku transportu jakim będzie przewożona przesyłka tj. nazwa, opis.
* Tabela **shipment\_movement**: Przechowuje dane o przesunięciu przesyłki tj. data przesunięcia, przesyłka, bramka, środek transportu, typ przesunięcia. Pozostaje w relacji n:1 z tabelą shipment (przesunięcie dotyczy jednej przesyłki), n:1 z tabelą movement\_type (przesunięcie posiada jeden typ przesunięcia), n:1 z tabelą gate (przesunięcie zostało zgłoszone przez jedną bramkę), n:1 z tabelą vehicle (przesunięcie z/do jednego środka transportu).
* Tabela **geo\_location**: Przechowuje dane geolokalizacyjne środka transportu tj. szerokość geograficzna, wysokość geograficzna, data utworzenia, środek transportu. Pozostaje w relacji n:1 z tabelą vehicle (dane geolokalizacjyjne dotyczą jednego środka transportu)

# Implementacja

1. Java EE - Java Enterprise Edition [↑](#footnote-ref-1)
2. RFID - Radio Frequency ID [↑](#footnote-ref-2)