

# POLITECHNIKA WROCŁAWSKA

## WYDZIAŁ ELEKTRONIKI

---

KIERUNEK: Elektronika i telekomunikacja (EIT)  
SPECJALNOŚĆ: Zastosowania inżynierii komputerowej  
w technice (EZI)

### PRACA DYPLOMOWA MAGISTERSKA

Projektowanie systemów zdalnego nauczania z  
wykorzystaniem usług sieciowych w środowisku  
MS .NET

Designing distance learning systems using Web  
Services in MS .NET environment

AUTOR:  
Michał Franc

PROWADZĄCY PRACĘ:  
dr inż. Robert Wójcik PWr, I-6

OCENA PRACY:



*Mamie i tacie :D*



# Spis treści

# Rozdział 1

## Wstęp

### 1.1 Cel pracy

Celem niniejszej pracy jest zaprojektowanie oraz zaimplementowanie systemu informatycznego wspomagającego proces zdalnego nauczania oparty o technologię firmy Microsoft oraz z wykorzystaniem mechanizmu usług sieciowych do komunikacji z bazą danych.

### 1.2 Zakres pracy

Analiza wymagań systemu zdalnego nauczania. Projekty systemu w oparciu o mechanizmy usług sieciowych oraz frameworka Asp.Net Mvc 3 oraz bazy danych MSSql. Testy wydajnościowe , funkcjonalne. Analiza proponowanego rozwiązania pod punktem czytelności oraz modyfikalności. Analiza wykorzystanych narzędzi oraz współczesnych technik programowania.

# Rozdział 2

## Kształcenie na odległość

### 2.1 Definicja kształcenia na odległość

Nie ma jednoznacznej definicji w pełni określającej charakter zdalnego nauczania. W literaturze można znaleźć wiele terminów odnoszących się do tego samego zwrotu.

*Terminy często używane w kontekście zdalnego nauczania to e-nauczanie, nauczanie internetowe, nauczanie rozproszone, nauczanie sieciowe, tele-nauczanie, wirtualne nauczanie, nauczanie wspomagane komputerowo, nauczanie webowe, oraz nauczanie na odległość. [?]*

W wielu opracowaniach naukowych termin ten sprowadza się po prostu do postaci **"kształcenie na odległość"**, w niniejszej pracy będę posługiwał się tym terminem.

Wszystkie zwroty łączy jedną wspólną cechą mianowicie zakładają one, że podstawą kształcenia na odległość jest niebezpośredni proces uczenia się. Tzn proces w którym osoba ucząca się nie przebywa w bezpośrednim kontakcie z nauczycielem. Interesant taki korzysta z dostępnych zasobów zdalnie. Najczęściej za pomocą komputera bądź telefonu komórkowego. Jest to nowoczesny sposób przekazywania wiedzy, który staje się bardzo popularny na świecie, większość instytucji edukacyjnych stara się wdrażać taki system.

Jak słusznie zauważa Alan Clarke *"E - learning ... Stanowi swoistą rewolucję, której skutki są często porównywalne do wpływu, który wcześniej na kształcenie wywarły wynalezienie druku i masowa produkcja książek ..."* [?]

Do kształcenia na odległość może zaliczyć zarówno proces nauczania jak i proces wspomagania nauczania. Do działań wspomagających możemy zaliczyć przede wszystkim. Sposób prezentacji materiałów oraz sposób dostarczania kolejnych materiałów bazując na wynikach uczącego się. Założenia te nie są częścią tej pracy dyplomowej. W niniejszej pracy skupimy się na stworzeniu platformy webowej posiadającej mechanizmy wspierające proces zdalnego nauczania zarówno od strony uczącego się oraz nauczyciela.

## 2.2 Korzyści płynące z używania systemów kształcenia na odległość

Proces kształcenia do 19 wieku przebiegał standardowo tzn. odbywał się w bezpośrednim kontakcie , najczęściej w specjalnie wydzielonej do tego sali, pomiędzy nauczycielem oraz uczniami. Rewolucja informatyczna oraz coraz większa znajomość technologii informatycznych wśród społeczeństwa spowodowały , że pod koniec 20 wieku doszło do ogromnego zainteresowania systemami kształcenia na odległość. Kształcenie na odległość posiada wiele cech pozytywnych w porównaniu do starego systemu i podejścia do nauczania

Alan Clark [?] wymienia wiele korzyści płynących z zastosowania e-learningu. Zwraca uwagę m.in na:

- dostępność materiałów. Przeważnie otrzymujemy całodobowy dostęp do interesujących nas materiałów.
- asynchroniczny charakter procesu kształcenia tzn. uczący sam decyduje kiedy chce korzystać z materiałów.
- swoboda wyboru miejsca , czasu oraz tempa uczenia się.
- ułatwiony proces dystrybucji materiałów oraz informacji
- ułatwienie komunikacji z osobami z całego świata

Józef Bednarek [?] zwraca również uwagę na zwiększoną efektywność procesu nauczania. Poprzez zwiększony przedział zrozumienia tematu oraz zaangażowania uczących się. Ważnym aspektem przemawiającym na korzyść kształcenia na odległość jest również oszczędność czasu poprzez ułatwiony dostęp do nauczyciela oraz materiału dydaktycznego.

## 2.3 Ograniczenia kształcenia na odległość

Oczywiście jak każde rozwiązanie kształcenie na odległość posiada plusy oraz minusy poza plusami. do najważniejszych problemów jakie należy wymienić możemy zaliczyć.

- konieczność przeszkolenia personelu dydaktycznego oraz uczniów.
- wdrożenie kosztownego systemu informatycznego obsługującego proces nauczania
- wymagana większe zaangażowanie i samodzielna inicjatywa od uczącego się.
- konieczność dopasowania obecnego materiału dydaktycznego do nowych standardów
- brak typowych dla bezpośredniego procesu nauczania mechanizmów grupy oraz integracji z grupą. Można temu zaradzić w pewnym stopniu poprzez odpowiednio stworzony system.
- anonimowość nauczyciela , który od wieków stał na straży zdobywania wiedzy pełniąc funkcję autorytetu.



## 2.4 Klasyfikacja systemów kształcenia na odległość

Zgodnie z A. Clarkiem [?] systemy E-learningowe możemy sklasyfikować w obrębie dwóch podstawowych rodzajów.

### 2.4.1 Systemy do zarządzania kursami(CMS)

RYSUNEK CMS-A

Aplikacje dostarczające przede wszystkim funkcji pozwalających zarządzać materiałami wspomagającymi proces nauczania , poprzez dostarczanie funkcji ułatwiających zarówno tworzenie nowych materiałów jak i udostępnianie materiałów. W systemach takich możemy wyszczególnić kilka podstawowych funkcji

- katalogowanie dostępnych materiałów dydaktycznych
- system administracji pozwalający zarządzać procesem udostępniania materiałów
- mechanizmy wspomagające proces tworzenia oraz publikowania nowych materiałów

Systemy takie zazwyczaj sprowadzają się do postaci katalogów online.

### 2.4.2 Systemy wspomagające proces nauczania (LMS)

RYSUNEK LMS

Systemy LMS są bardziej wyspecjalizowanymi systemami wspomagającymi proces nauczania. Poprzez dostarczanie niektórych funkcjonalności takich jak:

- moduł testów pozwalających zdalnie ewaluować umiejętności kursanta
- moduł ocen pozwalający dokumentować postęp oraz wyniki kursanta
- moduł wspomagający kolaborację oraz komunikację pomiędzy kursantami i nauczycielami
- moduł statystyk oraz raportów pozwalających ewaluować skuteczność procesu kształcenia
- moduł autonomicznego agenta podającego np sugestie kursantowi kierując oraz dostosowując jego tok nauczania

# Rozdział 3

## Wybrane technologie oraz techniki realizacji systemów webowych

### 3.1 Formaty danych JSON , XML

JSON (JavaScript Object Notation) jest formatem danych wykorzystywanym przy transferze danych po sieci. Jest on alternatywą do standardu XML. Jest to nowy format danych opisany w 2006 roku w publikacji RFC4627[?] . Jest to lekki format z bardzo prostą semantyką. Dane przesyłane w tym formacie składają się z pary ciągów znaków : nazwa , wartość.



Rys. Schemat formatu JSON

Semantyka formatu pozwala zdefiniować : [?]

- Obiekt
- Tablicę obiektów , wartości.
- Ciąg Znaków
- Liczba
- Wartość boolowska

#### Cechy formatu JSON

- ułatwione przetwarzanie formatu do obiektów języka javascript przy pomocy funkcji eval
- lekki prosty format
- szybki proces przetwarzania
- łatwy w modyfikacji

## Porównanie JSON , XML

W momencie wejścia na rynek usług sieciowych , były one głównie wykorzystywane w świecie biznesowym. Rynek Enterprise zaadoptował na początku format XML i protokół oparty na tym formacie SOAP. XML był idealnym formatem ponieważ był już dobrze znany i miał spore wsparcie narzędzi oraz języków.

Wzrost zainteresowania formatem JSON zaczął się w 2006 roku wraz z powstaniem pierwszego oficjalnego opisu oraz wraz z adopcją tego formatu przez takie firmy jak Google czy Yahoo.

Format XML charakteryzuje większym poziomem bezpieczeństwa dzięki bardziej stabilnej i mocno typowanej semantyce. W przypadku formatu JSON istnieje pewne ryzyko pozwalające wstrzyknąć niepożądany kod ponieważ JSON wykorzystuje funkcje eval.

W porównaniu do XML-a format JSON charakteryzuje się mniejszą wagą przesyłanego ładunku , jest lżejszy i czytelniejszy dla odbiorcy.



```

{
  "ID":1,
  "Role":"Author",
  "name":"Franc",
  "first-name":"Michal",
  "age":25,
  "hobbies":{
    "reading",
    "programming",
    {
      "sports":{
        "volley-ball",
        "football"
      }
    }
  },
  "address":{
    "City":"Wroclaw",
    "Street":"Mosiezna",
    "NR":"19/22"
  }
}

```

```

<?xml version="1.0" encoding="UTF-8" ?>
- <json>
  <ID>1</ID>
  <Role>Author</Role>
  <name>Franc</name>
  <first-name>Michal</first-name>
  <age>25</age>
  <hobbies>reading</hobbies>
  <hobbies>programming</hobbies>
- <hobbies>
  <sports>volley-ball</sports>
  <sports>football</sports>
</hobbies>
- <address>
  <City>Wroclaw</City>
  <Street>Mosiezna</Street>
  <NR>19/22</NR>
</address>
</json>

```

Rys. Porównanie standardów. Json , XML.

W pracy magisterskiej zastosowałem zarówno format JSON jak i XML. Dane nie wymagające dużego poziomu bezpieczeństwa przesyłane są w formacie JSON natomiast dane takie jak np. dane adresowe , logowania przesyłane są formatem XML.

## 3.2 Architektura REST , SOAP

### 3.3 Usługi sieciowe

Usługi sieciowe są technologią pozwalającą inicjować komunikację z pomiędzy dwoma urządzeniami podłączonymi do sieci komputerowej.

Rysunek usług sieciowych.

client na początku odkrywa jakie sa dostępne usługi następnie tworzy wiadomosc i otrzymuje odpowiedz z danymi. - podstawowa koncepcja web usług co gdzie jak - rysunek web usług i komunikacji

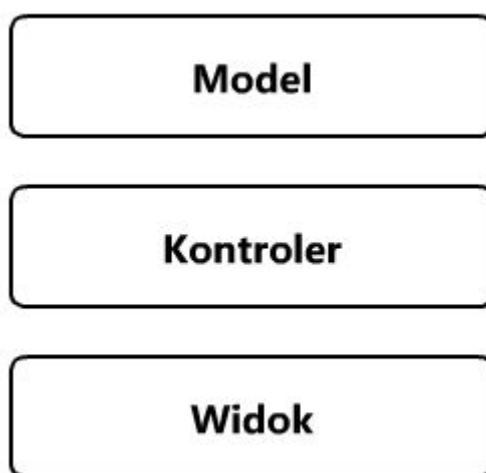
## 3.4 Framework Asp.Net Mvc 3

Asp.Net Mvc 3 jest platformą stworzoną przez firmę Microsoft służącą do tworzenia aplikacji webowych. Działa jako nakładka na platformę Asp.Net. Jest odpowiedzią na nowe trendy zdobywające coraz to większą popularność w środowisku programistów webowych. Cały Framework oparty jest na wzorcu projektowym MVC.

Wraz z ewolucją aplikacji webowych i ich poziomu skomplikowania pojawiały się nowe podejścia oraz sposoby wytwarzania aplikacji webowych pozwalające tworzyć aplikacje. Jednym z takich podejść jest wykorzystanie wzorca Model View Controller. Pierwszy opis wzorca można znaleźć w dokumencie z 1979[?].

Od 2004 roku wzorec ten zaczął zdobywać dużą popularność wraz z pojawieniem się nowych platform programistycznych : Ruby On Rails oraz Django.

Wprowadza on podział aplikacji na trzy oddzielne warstwy : Model , Widok , Kontroler.



Rys. Koncepcja MVC

### Model

Reprezentuje warstwę bazy danych. Dostęp do modelu jest jedynie możliwy z poziomu kontrolera , który w wielu aplikacjach do komunikacji z modelem korzysta również z warstwy usług. Warstwa ta wystawia metody zdefiniowane w kontrakcie , z których korzysta kontroler.

### Widok

Reprezentuje warstwę dostępną dla użytkownika systemu. Buduje się na podstawie modelu przekazanego przez kontroler.

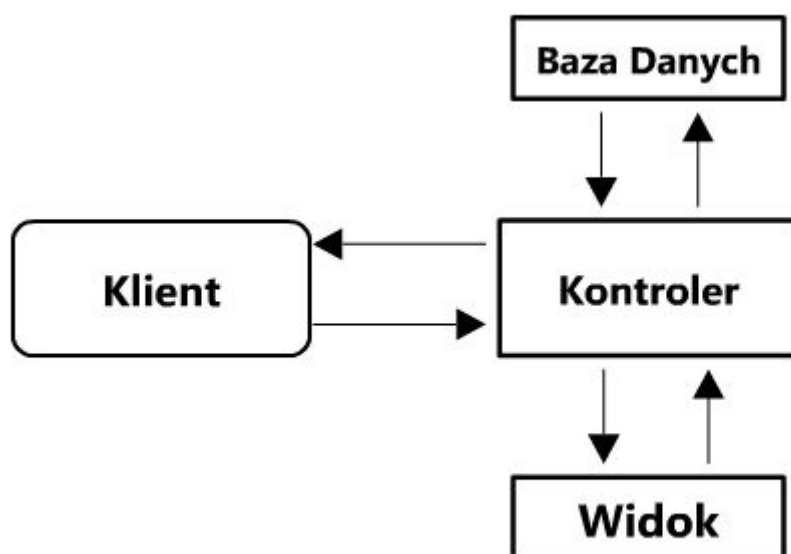
### Kontroler

Warstwa odpowiedzialna za sterowanie przepływem danych i przetwarzaniem tych danych by mogły być wyświetlone w warstwie widoku.

Zastosowanie wzorca MVC przy projektowaniu aplikacji webowej wymaga większego nakładu pracy w początkowej fazie projektu. Wymierne korzyści ze stosowania tego

wzorca zaczynają być odczuwane dopiero w późniejszych etapach życia projektu. Przede wszystkim zastosowane konwencje i jawna separacja odpowiedzialności na trzy warstwy pozwala oddzielić od siebie logikę biznesową dostępną z poziomu klienta od logiki obsługującej dostęp do bazy danych. Jest to bardzo ważne ponieważ zmiany zachodzące w warstwie modelu tzn bazy danych nie powinny powodować zmian w warstwie widoku. Dzięki takiemu rozdziałowi powstaje lepszy kod , łatwiejszy w rozbudowanie oraz utrzymaniu. Dodatkowo projekt jest bardziej czytelny. Programista wiedzący ze projekt został stworzony w oparciu o MVC automatycznie wie gdzie szukać poszczególnych implementacji systemu w celu przeprowadzenia modyfikacji.

### Opis komunikacji



Rys. Schemat komunikacji Mvc.

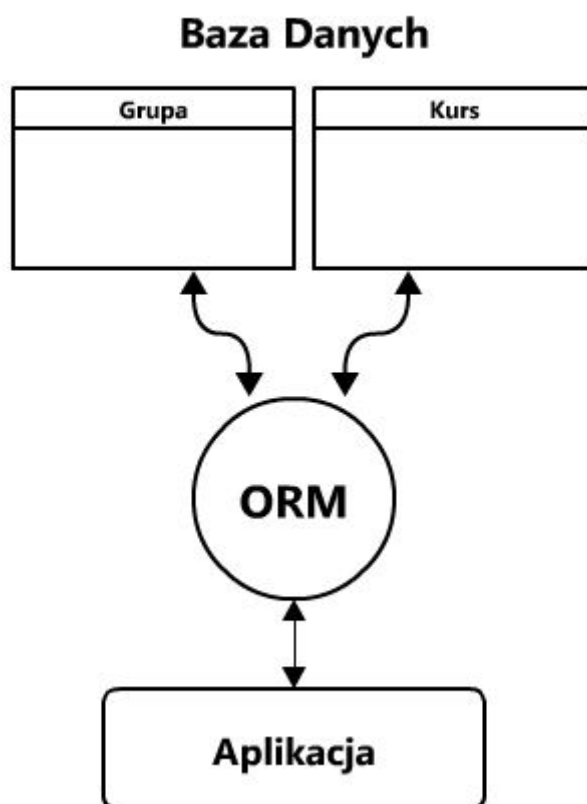
Klient realizuje zapytanie , które przechwytuje kontroler. w przypadku aplikacji webowej zapytanie będzie po prostu zwykłym odniesieniem się do określonego adresu url. Kontroler przejmuje żądanie. Jeżeli wygenerowanie odpowiedniego widoku nie wymaga pobrania danych z bazy danych. Kontroler pobiera dany widok i przekazuje go klientowi w swojej odpowiedzi na zadanie. Jeżeli widok wymaga pobrania danych z bazy danych , realizowane jest połączenie z modelem pobranie danych i wygenerowanie widoku z danymi i przekazanie go w wiadomości zwrotnej.

## 3.5 Mapowanie obiektowo relacyjne - NHibernate

Bazy danych są najważniejszą częścią systemu informatycznego. Stanowi większość implementacji aplikacji po stronie serwerowej. Przy tworzeniu systemów informatycznych logika dostępu do bazy danych pochłania bardzo dużo czasu. Dodatkowo jest podatna na błędy. Bezpośrednie tworzenie zapytań stało się zbyt kosztowne oraz trudne w utrzymaniu. Z rozwiązaniem takim wiąże się również inny problem mianowicie dochodzi do niekompatybilności zależności pomiędzy obiektami między systemami relacyjna bazodanowymi a systemami opartymi na dziedziczeniu i kompozycji klas znajdującymi się w języku programowania.

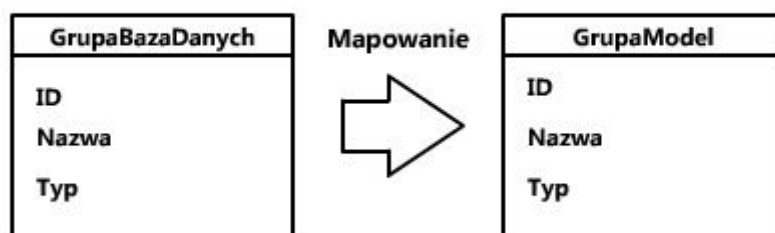
By ułatwić proces tworzenia kodu coraz więcej firm wykorzystuje specjalne biblioteki wspomagające proces tworzenia warstwy dostępu do danych. Nazywane one są Obiektowo relacyjnymi maopermai. Na rynku dostępnych jest wiele amperów najpopularniejszymi w środowisku jedna sa. Tworzony przez firmę Microsoft Entity Framework oraz NHibernate który jest implementacja frameworka Hibernate z technologii java na platformę .Net.

Dzięki zastosowaniu orma można wprowadzać bardzo szybko zmiany oraz w dużo łatwiejszy sposób wykonywać odpowiednie zapytania na bazie danych nie przejmując się tak naprawdę warstwa bazodanową. Dla programisty oporującego na tej warstwie cała komunikacja jest schowana pod interfejsami. Dzięki temu programista może skupić się na implementacji logiki oszczędzając czas na implementowaniu dostępu do bazy danych. Minusem takiego rozwiązania jest mniej wydajny proces pobierania danych. Problem ten można zniwelować poprzez odpowiednie sprofilowanie aplikacji i wyznaczanie najlepszych części systemu wymagających optymalizacji. W tym przypadku profilowane są zapytania SQL. Zapytania wymagające optymalizacji można zamienić na zwykłe zapytani sqlowe. Dzięki takiemu zabiegowi oszczędza się czas oraz fundusze przeznaczone na projekt.



Rysunek przedstawiający warstwę dostępu do danych  
realizowaną za pomocą mappera obiektowo relacyjnego

Proces mapowania sprowadza się do określenia mapowań poprzez wskazanie narzędziu , które pole z bazy danych ma być połączone z obiektem wykorzystywanym w aplikacji. w ten sposób tworzone są specjalne klasy pośredniczące w komunikacji pomiędzy bazą danych a aplikacją.



Rysunek przedstawiający proces mapowania.

W NHibernate można mapować za pomocą plików konfiguracyjnych xml. Jest to wygodne podejście jednakże podatne na błędy i nieczytelne. Dlatego często stosuje się rozwiązanie typu FluentNHibernate bibliotekę , która pozwala generować pliki XML na podstawie kodu napisanego w języku platformy .Net. Mapowanie takie staje się bardziej czytelne.

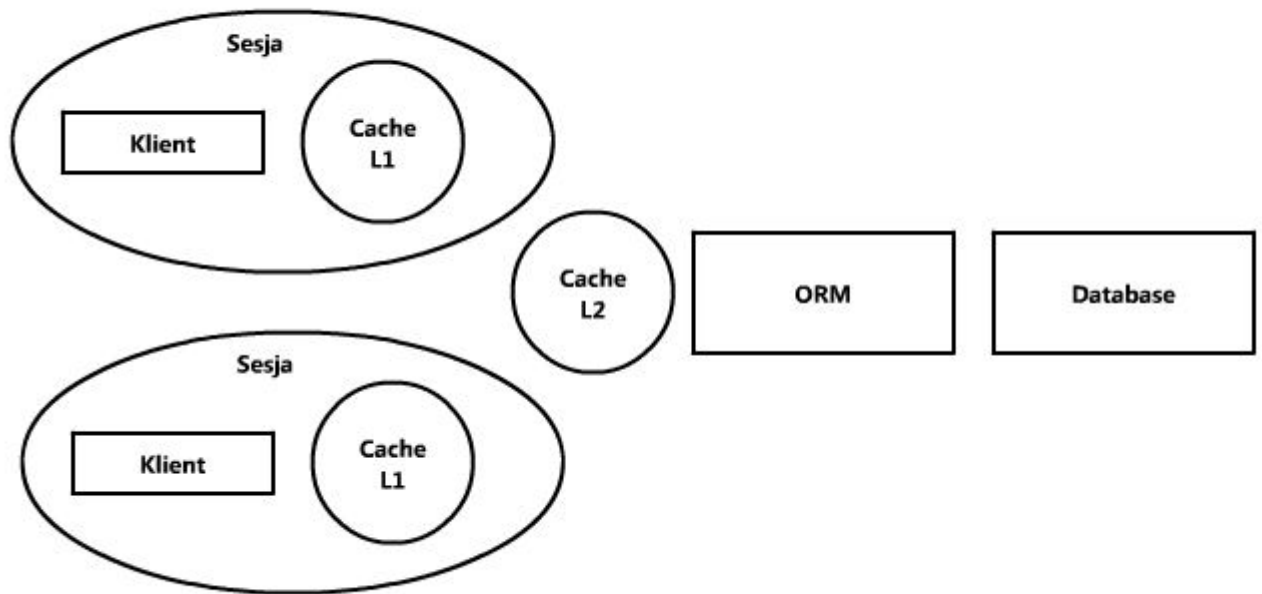
NHibernate dodatkowo wspiera mechanizm cachowania pierwszego oraz drugiego poziomu dzięki którym proces pobierania danych jest bardziej wydajny.

Cache pierwszego poziomu trzyma wartości z tabel i w momencie gdy nastąpi kolejne odwołanie do danych sprawdzany jest cache po nazwie tabeli oraz numerze ID. Jeżeli obiekt znajduje się na poziomie cache pierwszego poziomu jest pobierany z tego cache bez konieczności wykonywania zapytania do bazy danych.

Cache pierwszego poziomu dostępny jest w obrębie tylko jednej sesji. Tzn dla każdego klienta zaczynającego pracę z systemem tworzona jest sesja każdy klient posiada swój własny cache z którego korzysta. By współ dzielić cache pomiędzy różnymi klientami i ich sesjami stosuje się cache drugiego poziomu.

Odpowiednie skonfigurowanie obu sposobów cachowania pozwala zoptymalizować proces zapytań do bazy danych.





Rysunek przedstawiający poziomy cachowania

## 3.6 Testy Jednostkowe

Branża wytwarzania oprogramowania przechodzi przez okres ciągłych zmian szukając rozwiązań oraz metodyk najlepiej spełniających swoje zadanie. Metodyka Waterflass przejęta z innych dziedzin inżynierskich takich jak Automatyka Inżynieria budowała okazała się nie efektywna. Środowisko programistów zaczęło poszukiwać innych rozwiązań i w ten sposób powstała metodyka Extreme Programming. Która zakłada nie jeden okres planowania ale wiele okresów planowania następujących cyklicznie. By można było taką metodykę tworzyć oprogramowanie zaczęto używać wiele narzędzi wspomagających ten proces. Jednym z takich narzędzi stały się testy jednostkowe.

Testowanie od początku było dziedziną którą była przeprowadzana przez dział testów. Jednakże środowisko programistyczne odkryło że można przeprowadzać testy w kodzie podczas tworzenia oprogramowania tym samym polepszając jakość tworzonego kodu. Testy jednostkowe są formą kodu który testuje inny kod. Jest to kod napisany w ten sposób że wykorzystuje stworzony kod i testuje pewne założenia.



Procedura przeprowadzania testu jednostkowego

Pozwalają ograniczyć ilość błędów. Powodują jednak że marnujemy trochę czasu a odpowiednio zdefiniowanie testów ale tak naprawdę programista i tak spędziłby podobną ilość czasu przeklikując np. aplikację.

```
[Test]
public void Can_get_all_courses_signatures()
{
    #region Arrange
    #endregion

    #region Act

    var courseSignatures = new CourseService().GetAllSignatures();

    #endregion

    #region Assert
    Assert.That(courseSignatures, Is.Not.Null);
    Assert.That(courseSignatures.Count, Is.EqualTo(3));
    Assert.That(courseSignatures.First(), Is.InstanceOf(typeof(CourseSignatureDto)));
    #endregion
}
```

### Przykładowy test jednostkowy

Pisanie testów jednostkowych nie dość że daje nam możliwość przetestowania najszej aplikacji ale również wymusza się trzymanie lepszych wytycznych i wzorców projektowych ponieważ nie wszystkie dane da się łatwo zrobić.

#### 3.6.1 Mockowanie oraz Stubowanie obiektów

Aplikacje biznesowe mają często bardzo skomplikowaną logikę wykorzystującą wiele modułów oraz zależności. Piszac testy jednostkowe powinno testować się tylko pojedyncze funkcjonalności. W momencie gdy np. nasza aplikacja do przeprowadzenia pewnej czynności wykorzystuje zewnętrzne zasoby np. dane z bazy danych w momencie gdy przeprowadzamy taki test musimy wykorzystać bazy danych. W tym momencie nie testujemy tylko jednej funkcjonalności ale również same połączenie bądź poprawną konfigurację połączenia.

By móc testować aplikację wraz z takimi zależnościami musimy bądź przeprowadzać testy operując na prawdziwej bazie danych bądź emulować bazy danych. Oba te podejścia wiążą się z pewnymi ograniczeniami i narzucają wiele czynności.

Innym podejściem zgoła innym jest tworzenie specjalnych klas udających połączenie z bazą danych. Klasy takie nazywamy Stubami bądź mockami w zależności od tego jak są konstruowane.

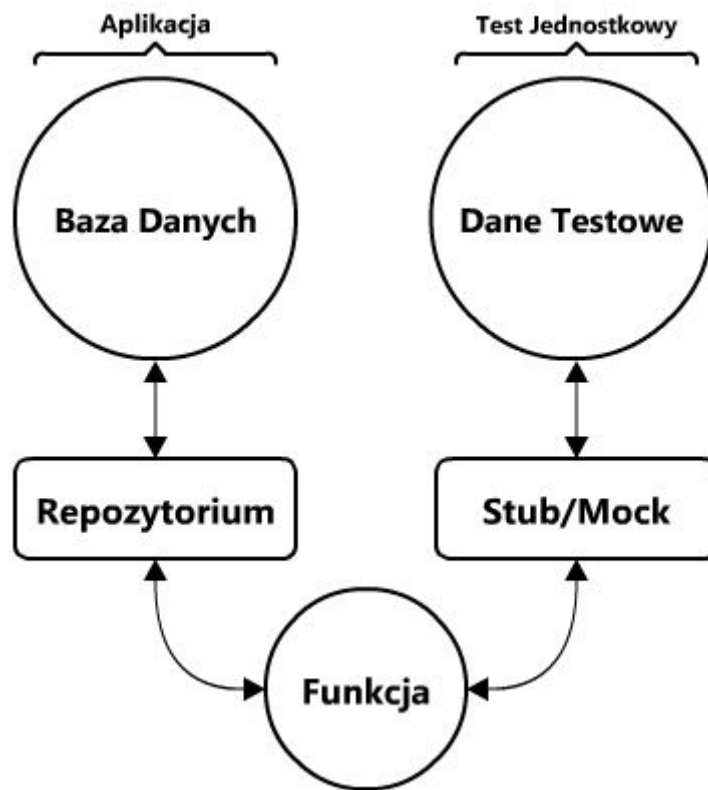
#### Stub

Stub jest specjalnym obiektem imitującym klasę. W ten sposób że posiada te same funkcje oraz testowe dane jak testowana klasa. W przypadku naszej komunikacji do bazy danych. Imitujący obiekt posiadałby w miejscu funkcji pobierającej dane funkcję zwracającą stałą kolekcję obiektów.

#### Mock

Działa podobnie jak Stub. Różnica jednak polega na tym, że obiekty typu mock działają na prawdziwej klasie podczepiając się tylko pod wybrane funkcjonalności. Tzn. w naszym przypadku pobierania danych z bazy danych. Będziemy wykorzystywać naszą normalną klasę podniejając tylko jej funkcję dostępu do bazy danych.

## Zastosowanie



Zastosowanie

# Rozdział 4

## Projekt systemu zdalnego nauczania

### 4.1 Wymagania Projektowe

#### 4.1.1 Wymagania funkcyjne

Podstawową funkcją systemu jest możliwość zarządzania kursami zdalnego nauczania wraz z możliwością tworzenia materiałów , testowania kursantów oraz wystawiania o cen.

#### **Tworzenie użytkowników oraz przydzielanie uprawnień**

System musi posiadać mechanizm pozwalający zarządzać użytkownikami systemu wraz z możliwością przydzielania im odpowiednich poziomów uprawnień.

#### **Zarządzanie kursami**

Każdy Autor oraz Administrator może stworzyć jeden lub więcej kursów , którymi zarządza. Kurs opisany jest parametrami

- Nazwa
- Krótki opis
- Logo
- Rodzaj kursu

Każdy kurs posiada swoją odrębną listę materiałów nauczania , testów , mechanizm wymiany krótkich wiadomości (Shoutbox) oraz pojedynczą grupę użytkowników przynależących do danego kursu. Autor ma możliwość edycji tylko i wyłącznie kursów , którymi zarządza. Administrator może modyfikować każdy kurs znajdujący się w systemie.

#### **Zarządzanie materiałami nauczania**

Materiały nauczania są integralną częścią kursu. Każdy kurs może posiadać więcej niż jeden lub więcej materiałów nauczania. W skład materiału nauczania wchodzi parametry opisowe pozwalające sklasyfikować obiekt.

- Poziom materiału Początkujący , Średnio-Zaawansowany , Zaawansowany
- Opis
- Logo

## **Materiały nauczania - proces nauczania**

Materiał nauczania prócz parametrów opisowych podzielony jest na trzy obszary.

### **Obszar Informacyjny**

Zawiera segmenty :

- Informacyjny - parametry kursu
- Opisowy - opisuje kontekst kursu
- Celów - opisuje cele jakie osiągnie kursant biorący udział w kursie

### **Obszar nauczania**

W skład obszaru nauczania wchodzi sekcje zawierające materiały i wiedzę z , której korzysta kursant. Materiał nauczania może posiadać jeden lub więcej segmentów. Każdy segment opisany jest tytułem oraz polem zawartości , w którym umieszczamy wiedzę oraz materiały , które chcemy przekazać w procesie nauczania.

### **Obszar podsumowania**

Zawiera segmenty

- Podsumowania - szybkie podsumowanie zdobytej wiedzy i najważniejszych rzeczy , które zapamiętują kursant.
- Więcej informacji - posiada zbiór dodatkowych materiałów linków do nich i pozwala nakierować kursanta jeżeli ten chce poszerzyć wiedzę wykraczającą poza zakres materiału nauczania.
- Testów - posiada test stworzony na potrzeby materiału nauczania pozwalający sprawdzić wiedzę kursanta

Mechanizm nauczania stworzony jest w sposób liniowy. Kursant po kolei odkrywa kolejne segmenty oraz sekcje wchodzące w skład materiału nauczania. Na koniec kursant może wykonać notatki z przeprowadzonego procesu nauczania oraz sprawdzić swoją wiedzę wykonując test.

### **Zarządzanie testami**

Aplikacja pozwala tworzyć testy będące istotnym elementem procesu nauczania. Testy mogą być powiązane z kursem lub materiałem nauczania. Kursant posiada możliwość wyświetlenia wszystkich testów wchodzących w skład kursu bądź materiału nauczania. Możliwość tworzenia oraz edycji testów posiada Administrator oraz Autor kursu posiadający odpowiednie uprawnienia.

### **Mechanizm rozwiązywania oraz sprawdzani testów**

Kursant po skończeniu procesu nauczania może przystąpić do rozwiązywania testów. Test składa się z segmentu pytania oraz odpowiedzi. Po rozwiązaniu testu wyświetlana jest ocena oraz wiadomość informująca o zaliczeniu bądź nie testu. Następnie ocena zapisywana jest do dzienniczka ucznia

### **Zarządzanie ocenami**

Mazdy kursant może przeglądać listę swoich ocen otrzymanych po rozwiązaniu testu. Kursant ma możliwość także automatycznego wyliczenia średniej ocen z danego kursu.

### **Filtrowanie kursów ora z materiałów**

Kursant ma możliwość filtrowania kursów na podstawie typu kursu. Może również wyświetlić listę kursów do , których jest już zapisany. Kursant ma możliwość filtrowania materiałów nauczania na podstawie poziomu trudności może również sortować materiały na podstawie nazwy materiału.

## **4.1.2 Wymagania niefunkcjonalne**

### **Łatwość modyfkacji platformy**

Platforma stworzona zostanie zgodnie z nowoczesnymi trendami oraz zasadami dobrego projektowania aplikacji tak by w przyszłości proces modyfikacji i utrzymania kodu był jak najtańszy.

### **Bezpieczeństwo aplikacji**

Dostęp do poszczególnych funkcjonalności systemu jest ograniczony w obrębie trzech ról przydzielanych do poszczególnych użytkowników platformy. Wyróżniamy trzy role:

#### **Administrator**

Administrator reprezentuje użytkownika odpowiedzialnego za zarządzanie całym systemem zdalnego nauczania.

#### **Autor**

Autor jest użytkownikiem posiadającym możliwość tworzenia nowych , edytowania kursów , materiałów nauczania oraz testów.

#### **Kursant**

Kursant jest podstawowym użytkownikiem posiadającym możliwość przeglądania zasobów aplikacji i interakcji z systemem bez możliwości wpływania na zawartość systemu. Kursanci posiadają możliwość umieszczania krótkich wiadomości tekstowych w module ShoutBox.

### **Bezpieczeństwo transmisji danych**

Platforma komunikuje się z warstwą bazodanową za pomocą usług sieciowych. Transmisja danych przebiegała będzie na dwóch poziomach zabezpieczeń. Pierwszy poziom bez szyfrowania i bez zabezpieczeń będzie służył do przesyłania danych zawierających jedynie dane na temat kursów , testów oraz materiałów nauczania. Dane przesyłane na tym poziomie będą w przesyłane w formacie JSON by zmniejszyć ilość przesyłanych danych oraz ułatwić tworzenie aplikacji po stronie klienta. Dane poufne takie jak dane logowania , hasła będą przesyłane bardziej bezpiecznym sposobem transmisji. Komunikacja będzie szyfrowana oraz będzie implementowała mechanizm tokenów pozwalający zweryfikować czy nadawca bądź odbiorca są poprawni i nie doszło do nieuprawnionego przekierowania

zapytania bądź odpowiedzi. Poufne dane będą przesyłane w formacie XML z zastosowaniem rozwiązań protokołu SOAP zdefiniowanych w dokumencie WS-Security.

### **Komunikacja za pomocą usług sieciowych**

Dostęp do bazy danych wystawiony będzie w za warstwą usług sieciowych.

### **Mechanizm dynamicznego odnajdywania serwera bazodanowego**

By zwiększyć niezawodność systemu stworzony zostanie system , który w momencie problemów z komunikacją z podstawowym serwerem bazy danych przełączy system na zapasową bazę danych.

### **Tworzenie kopii zapasowych bazy danych**

Każdego dnia o określonej porze będzie tworzona kopia zapasowa bazy danych. Jednorazowo trzymanych będzie siedem kopii.

### **Automatyczny mechanizm replikacji danych**

Ponieważ rozwiązanie zakłada możliwość przekierowania na innych serwer usług posiadający kopię zapasową bazy danych. Musi zostać zapewniony mechanizm replikacji uaktualniający bazę danych na zapasowym serwerze. Replikacja taka będzie wykonywana raz dziennie.

### **Dostępność aplikacji**

Aplikacja będzie dostępna z poziomu przeglądarki internetowej. Będzie dostosowana funkcjonalnie do przeglądarek :

- Chrome
- Firefox 4,5
- Internet Explorer 9
- Opera

Zostanie zagwarantowany dostęp do funkcjonalności z poziomu wymienionych wyżej przeglądarek. Nie zostanie zagwarantowany idealnie taki sam sposób prezentacji aplikacji.

### **Obciążalność**

Aplikacja będzie w stanie obsłużyć jednorazowo 100 żądań i będzie gwarantować czas odpowiedzi w granicach maksymalnie 1 sekundy



## 4.2 Diagramy przepływu pracy

## 4.3 Diagramy przypadków użycia

## 4.4 Projekt Bazy Danych

### 4.4.1 Model Konceptualny

#### **Kurs**

Zawiera dane dotyczące kursu : Id , Typ Kursu , Nazwę , Logo , Data Utworzenia , Opis , Krótki Opis , Pole Wiadomości

#### **TypKursu**

Encja słownik zawierająca , Identyfikator typu , klucz do kursu oraz nazwę typu.

#### **UkonczonyTest**

Encja reprezentująca ukończony test. Zawiera otrzymaną ocenę oraz datę ukończenia testu.

#### **Test**

Reprezentuje Test wypełniany przez kursanta należący do materiału nauczania.

#### **TestType**

Słownik opisujący typ Testu

#### **PytanieTestowe**

Reprezentuje pytanie należące do zbioru pytań wchodzących w skład testu. Zawiera odpowiedzi , tytuł oraz treść pytania.

#### **PytanieTestoweOdpowiedz**

Reprezentuje odpowiedzi będące częścią pytania. Zawiera treść odpowiedzi oraz wskaźnik czy jest to poprawna odpowiedź.

#### **ShoutBox**

Encja reprezentująca moduł krótkich wiadomości tekstowych zawiera listę wiadomości.

#### **ShoutBoxWiadomosc**

Reprezentuje wiadomość wyświetlaną w obrębie ShoutBoxa. Zawiera treść , login użytkownika oraz datę przesłania.

### Dziennik

Encja reprezentująca dziennik ocen. Każdy kursant posiada pojedynczy obiekt dziennika dla każdego kursu do którego dołączył.

### DziennikOcena

Encja ocena wchodząca w skład dziennika. Reprezentuje pojedynczą ocenę otrzymywaną po skończeniu testu.

### MaterialNauczania

Opisuje pojedynczy materiał nauczania będący podstawowym środkiem przekazu informacji dla kursanta.

### Section

Sekcja jest integralną częścią materiału nauczania. Zawiera informacje dydaktyczne. Materiał nauczania może posiadać wiele sekcji.

### Profil

Reprezentuje użytkownika systemu.

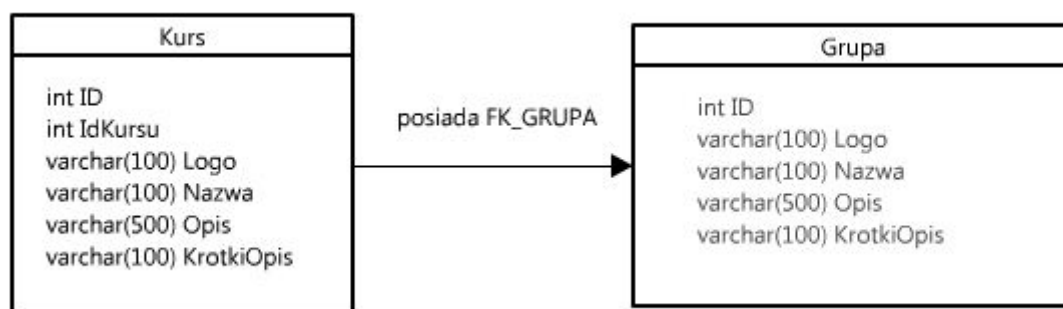
### GrupaUzytkownikow

Encja opisuująca grupę użytkowników. Każdy kurs posiada pojedynczą grupę użytkowników.

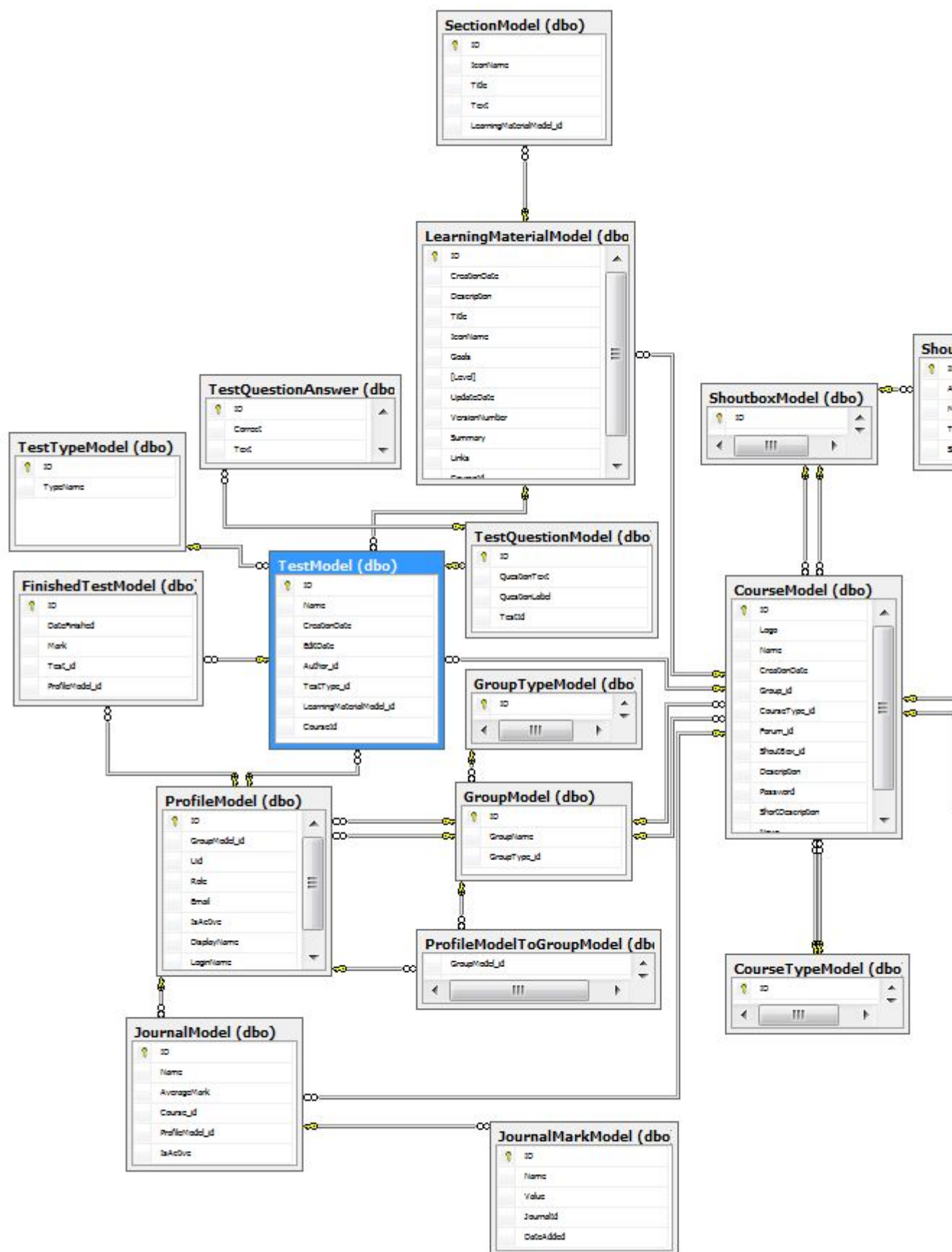
## 4.4.2 Opis Relacji



## 4.4.3 Model fizyczny bazy danych



## 4.4.4 Diagram



## 4.5 Architektura Systemu

4.5.1 Diagram klas

4.5.2 Diagram sekwencji

4.5.3 Schemat Przepływu Danych

4.5.4 Komunikacja z usługami sieciowymi

4.5.5 Mechanizmy zabezpieczeń



## Rozdział 5

# Implementacja systemu zdalnego nauczania

### 5.1 Realizacja Bazy Danych

### 5.2 Wykorzystane narzędzia

#### 5.2.1 Mechanizm logowania zdarzeń - NLog

#### 5.2.2 Testy Jednostkowe - NUnit

#### 5.2.3 Dependancy Injection - NInject

#### 5.2.4 Mockowanie zależności - RhinoMocks

#### 5.2.5 Mapowanie obiektowo relacyjne - NHibernate

#### 5.2.6 Tworzenie mapowań - FluentNHibernate

### 5.3 Realizacja Aplikacji

#### 5.3.1 Realizacja mechanizmów przetwarzania danych

#### 5.3.2 Realizacja protokołu komunikacji

#### 5.3.3 Realizacja mechanizmów zabezpieczeń

#### 5.3.4 Realizacja zewnętrznego API - przykładowe zastosowanie

#### 5.3.5 Realizacja mechanizmu dynamicznej zmiany serwera

### 5.4 Interfejs użytkownika - Moduły

#### 5.4.1 Testy

#### 5.4.2 Kursy

#### 5.4.3 Dziennik ocen

#### 5.4.4 Listy

# Rozdział 6

## Testowanie i ocena efektywności

### 6.1 Wybrane Testy Funkcjonalne

### 6.2 Testy Mechanizmów Zabezpieczeń

### 6.3 Testy Zabezpieczeń

### 6.4 Testy wydajności mechanizmów przetwarzania danych

#### 6.4.1 Analiza zapytań generowanych przez NHibernate za pomocą NHProfilea

#### 6.4.2 Przykładowa optymalizacja zapytań

#### 6.4.3 Testy obciążeniowe

### 6.5 Testy wydajności mechanizmów komunikacji sieciowej

#### 6.5.1 Porównanie formatu Json , Xml

#### 6.5.2 Badanie czasu odpowiedzi usług sieciowych

#### 6.5.3 Testy konfiguracji rozłożenia usług sieciowych

### 6.6 Wnioski z testów i badań

# Rozdział 7

## Podsumowanie



# Bibliografia

- [1] ADL. Scorm documentation. <http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/2004n/Documentation.aspx>.
- [2] A. Al-Ajlan, H. Zedan. E-learning (moodle) based on service oriented architecture. <http://www.cse.dmu.ac.uk/STRL/research/publications/pubs/2007/2007-23.pdf>.
- [3] A. Alkou, S. A.El-Seoud. Web services based authentication system for e-learning. *International Journal of Computing and Information Sciences*, 5(2):74–78, 2007.
- [4] T. Anderson, F. Elloumi. *Theory and Practice of Online Learning*. Athabasca University, 2004.
- [5] J. Bednarek, E. Lubina. *Kształcenie na odległość podstawy dydaktyki*. PWN, 2008.
- [6] E. Bertino, L. D. Martino, F. Paci, A. C. Squicciarini. *Security for Web Services and Service-Oriented Architectures*. Springer, 2010.
- [7] S. Carliner, P. Shank. *The E-Learning HandBook Past Promises , Present Challenges*. Pfeiffer, 2008.
- [8] E. Cerami. *Web Services Essentials - Distributed Applications with XML-RPC, SOAP, UDDI , WSDL*. O'Reilly, 2002.
- [9] A. Clarke. *E-learning : nauka na odległość*. WKŁ, 2007.
- [10] T. Earl. *Soa Design Patterns*. Prentice Hall, 2009.
- [11] B. Evjen, S. Hanselman, D. Rader. *Asp.Net 3.5 z wykorzystaniem Csharp. Zaawansowane Programowanie*. Helion, 2010.
- [12] M. Fowler. *Architektura systemów zarządzania przedsiębiorstwem : wzorce projektowe*. Helion, 2005.
- [13] M. Fowler. *UML w kropelce*. LTP, 2005.
- [14] M. Fowler, K. Beck. *Refaktoryzacja : ulepszanie struktury istniejącego kodu*. WNT, 2006.
- [15] Z. Fryźlewicz, A. Salamon. *Podstawy architektury i technologii usług XML sieci Web*. PWN, 2008.
- [16] E. Gamma, R. Helm, R. Johnson, J. M. Vissides. *Wzorce Projektowe : elementy oprogramowania wielokrotnego użytku*. WNT, 2005.

- [17] A. Grewal, S. Rai, R. Phillips, C. C. Fung. The e-learning lifecycle and its services: The web services. *Proceedings of the Second International Conference on eLearning for Knowledge-Based Society*, 2005.
- [18] J. Górski. *Inżynieria Oprogramowania w projekcie informatycznym*. MIKOM, 2000.
- [19] M. A. Jab, H. K. Al-Omari. e-learning management system using service oriented architecture. *Journal of Computer Science*, 6(3):285–295, 2010.
- [20] C. McMurtry, M. Mercuri, N. Watling, M. Winkler. *Windows Communication Foundation Unleashed*. Sams, 2007.
- [21] X. Qi, A. Jooloor. Web service architecture for e-learning. *Systemics, Cybernetics and Informatics*, 3(5), 2006.
- [22] S. Resnick, R. Crane, C. Bowen. *Essential Windows Communication Foundation with .Net 3.5*. Addison Wesley, 2008.
- [23] M. Rosen, B. Lublinsky, K. T. Smith, M. J. Baler. *Service-Oriented Architecture and Design Strategies*. Wiley, 2008.
- [24] S. Schwinge. *Intelligent Information Technologies and Applications*, rozdział 1. Oakland University, 2008.
- [25] S. Weerawarana, F. Curbera, F. L. T. Storey, D. F. Ferguson. *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging*. Prentice Hall, 2005.