|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metryka dokumentu | | | | | |
| Projket: | Praca inżynierska | | | Firma: | Politechnika Warszawska |
| Nazwa: | Projekt techniczny | | | | |
| Temat: | Projekt techniczny | | | | |
| Autor: | Łukasz Dragan, Marcin Fusiara | | | | |
| Plik: | doc/ProjektTechniczny.docx | | | | |
| Nr wersji: | 0.01 | Status: | roboczy | Data sporządzenia: | 2015-10-18 |
| Streszczenie: | Celem dokumentu jest przedstawienie projektu technicznego pracy inżynierskiej | | | | |
| Zatwierdził: |  | | | Data ostatniej modyfikacji: | 2015-10-18 |

|  |  |  |  |
| --- | --- | --- | --- |
| Historia zmian dokumentu | | | |
| Wersja | Data | Kto | Opis |
| 0.01 | 2015-10-18 | Łukasz Dragan | Stworzenie dokumentu; użyte technologie |
| 0.02 | 2015-11-08 | Łukasz Dragan | Drzewo podstron |

Spis treści

[1 Streszczenie 2](#_Toc434775332)

[2 Architektura systemu 2](#_Toc434775333)

[2.1 Warstwa dostępu do danych 3](#_Toc434775334)

[2.2 Warstwa logiki biznesowej 3](#_Toc434775335)

[2.3 Warstwa interfejsu użytkownika 3](#_Toc434775336)

[2.3.1 Część serwerowa 3](#_Toc434775337)

[2.3.2 Część kliencka 3](#_Toc434775338)

[3 Baza danych 3](#_Toc434775339)

[3.1 Schemat bazy danych 3](#_Toc434775340)

[4 Projekt klas 3](#_Toc434775341)

[5 Użytkownicy systemu 3](#_Toc434775342)

[6 Projekt stron systemu 4](#_Toc434775343)

[6.1 Drzewo stron 4](#_Toc434775344)

[7 Integracja systemu z forum 4](#_Toc434775345)

[8 Użyte technologie 4](#_Toc434775346)

# Streszczenie

Niniejszy dokument zawiera opis realizacji technicznej założeń zawartych w analizie biznesowej dla systemu będącego przedmiotem pracy inżynierskiej. Głównym celem ów systemu jest usprawnienie wymiany informacji pomiędzy nauczycielem akademickim a uczniami. W dokumencie zostały opisane m.in.: architektura systemu, schemat klas, baza danych, role użytkowników i wykorzystane technologie.

Jednym z głównych założeń, przy projektowaniu systemu jest łatwość obsługi i prostota implementacji, co wiąże się to ze spadkiem wydajności systemu. Jednak w założeniu system nie będzie też poddawany dużym obciążeniom, więc w związku z tym wydajność nie jest kwestią priorytetową.

# Architektura systemu

System opisywany w dokumencie realizuje architekturę warstwową. Składa się z trzech warstw:

1. Warstwa dostępu do danych
2. Warstwa logiki biznesowej
3. Warstwa interfejsu użytkownika
   1. Część serwerowa
   2. Część kliencka

Celem zastosowania takiej architektury jest uzyskanie łatwość utrzymania i modyfikacji, skalowalności i dostępności. Każda warstwa jest niezależna od reszty, więc zmiany dokonywane wewnątrz każdej z warstw nie wpływają na inne warstwy i system jako całość. Dzięki architekturze warstwowej ułatwiony jest dostęp do zawartości systemu, gdyż zmiany nie narażają reszty systemu na ewentualne nieświadome szkodliwe zmiany.

## Warstwa dostępu do danych

## Warstwa logiki biznesowej

## Warstwa interfejsu użytkownika

Warstwa ta jest zbudowana zgodnie ze wzorcem MVC: Model-View-Controller.

### Część serwerowa

Część tę stanowią kontrolery – klasy odpowiedzialne za obsługę zapytań http pochodzących z części klienckiej. Dzielą zawartość systemy na logicznie związane ze sobą segmenty.

Kontrolery:

1. Account
2. Page
3. File
4. …

### Część kliencka

Cześć kliencka składa się z widoków dostępnych z poziomu przeglądarki internetowej użytkownika końcowego.

# Baza danych

Struktura systemu opiera się o relacyjną bazę danych, której model zrealizowany został w postaci tabel zawierających przechowywane informacje.

Przy projektowaniu bazy priorytetem była prostota struktury.

Pliki dostępne z poziomu użytkowników systemu są przechowywane bezpośrednio w systemie katalogów serwera aplikacyjnego, ale ich metadane i ścieżki są znajdują się w bazie danych.

## Opis tabel

|  |  |  |
| --- | --- | --- |
| **Nazwa kolumny** | **Typ danych** | **Czy zezwala na null** |
| **Article** | | |
| Id | int | nie |
| Header | nvarchar(64) | nie |
| Content | text | nie |
| PublishDate | datetime | nie |
| Status | int | nie |
| Author\_Id | nvarchar(128) | nie |
| Subject\_Id | int | nie |
| Klucz główny: Id  Klucze obce: Author\_Id(Teachers), Subject\_Id(Subjects) | | |

|  |  |  |
| --- | --- | --- |
| **AspNetRoles** | | |
| Id | intnvarchar(128) | nie |
| Name | nvarchar(256) | nie |
| Klucz główny: Id | | |

|  |  |  |
| --- | --- | --- |
| **AspNetUserRoles** | | |
| UserId | nvarchar(128) | nie |
| RoleId | nvarchar(128) | nie |
| Klucz główny: UserId, RoleId  Klucze obce: UserId(AspNetUsers), RoleId(AspNetRoles) | | |

|  |  |  |
| --- | --- | --- |
| **AspNetUsers** | | |
| Id | nvarchar(128) | nie |
| Email | nvarchar(256) | tak |
| PasswordHash | nvarchar(MAX) | nie |
| SecurityStamp | nvarchar(MAX) | nie |
| PhoneNumber | nvarchar(MAX) | tak |
| UserName | nvarchar(256) | nie |
| Klucz główny: Id | | |

|  |  |  |
| --- | --- | --- |
| **Files** | | |
| Id | int | nie |
| FileName | nvarchar(256) | nie |
| FilePath | nvarchar(512) | nie |
| ContentType | nvarchar(128) | nie |
| UploadDate | datetime | nie |
| Subject\_Id | int | tak |
| User\_Id | nvarchar(128) | tak |
| Klucz główny: Id,  Klucze obce: SubjectId(Subjects), User\_Id(Users) | | |

|  |  |  |
| --- | --- | --- |
| **Files** | | |
| Id | int | nie |
| FileName | nvarchar(256) | nie |
| FilePath | nvarchar(512) | nie |
| ContentType | nvarchar(128) | nie |
| UploadDate | datetime | nie |
| Subject\_Id | int | tak |
| User\_Id | nvarchar(128) | tak |
| Klucz główny: Id,  Klucze obce: SubjectId(Subjects), User\_Id(Users) | | |

|  |  |  |
| --- | --- | --- |
| **Languages** | | |
| Id | int | nie |
| CountryCode | nvarchar(8) | nie |
| Name | mvarchar(64) | nie |
| Klucz główny: Id | | |

|  |  |  |
| --- | --- | --- |
| **Menus** | | |
| Id | int | nie |
| Language\_Id | int | nie |
| Klucz główny: Id,  Klucz obcy: Language\_Id(Languages) | | |

|  |  |  |
| --- | --- | --- |
| **Pages** | | |
| Id | int | nie |
| Title | nvarchar(64) | nie |
| UrlName | nvarchar(64) | nie |
| Content | text | nie |
| LangGroup | int | nie |
| CreationDate | datetime | nie |
| LastUpdateDate | datetime | nie |
| Language\_Id | int | nie |
| Parent\_Id | int | tak |
| Menu\_Id | int | tak |
| Klucz główny: Id,  Klucze obce: Language\_Id(Languages), Parent\_Id(Pages), MenuId(Menus) | | |

|  |  |  |
| --- | --- | --- |
| **Phrases** | | |
| GroupId | int | nie |
| Language\_Id | int | nie |
| Text | nvarchar(512) | nie |
| Klucz główny: Id, Language\_Id  Klucze obce: Language\_Id(Languages) | | |

|  |  |  |
| --- | --- | --- |
| **Semesters** | | |
| Id | int | nie |
| Description | nvarchar(256) | tak |
| Klucz główny: Id | | |

|  |  |  |
| --- | --- | --- |
| **SignUpRequests** | | |
| Id | int | nie |
| Status | int | nie |
| Student\_Id | nvarchar(128) | nie |
| Subject\_Id | int | nie |
| Klucz główny: Id  Klucze obce: Student\_Id(Students), Subject\_Id(Subjects) | | |

|  |  |  |
| --- | --- | --- |
| **Subjects** | | |
| Id | int | nie |
| Name | nvarchar(64) | nie |
| Schedule\_Id | int | tak |
| Semester\_Id | int | nie |
| Syllabus\_Id | int | tak |
| Klucz główny: Id  Klucze obce: Schedule \_Id(Articles), Subject\_Id(Subjects), Syllabus\_Id(Syllabus) | | |

|  |  |  |
| --- | --- | --- |
| **SubjectsStudents** | | |
| Subject\_Id | int | nie |
| Student\_Id | nvarchar(128) | nie |
| Klucz główny: Subject\_Id, Student\_Id  Klucze obce: Subject \_Id(Subjects), Student \_Id(Students) | | |

|  |  |  |
| --- | --- | --- |
| **SubjectsTeachers** | | |
| Subject\_Id | int | nie |
| Teacher\_Id | nvarchar(128) | nie |
| Klucz główny: Subject\_Id, Teacher\_Id  Klucze obce: Subject \_Id(Subjects), Teacher\_Id(Teacher) | | |

W bazie odzwierciedlona jest hierarchia dziedziczenia z modelu obiektowego w języku C# dotycząca relacji User-Teacher i User-Student.

Potem diagram relacji

Baza danych stworzona została na systemie Microsoft SQL Server 2012. Jest to relacyjna baza danych w trzeciej postaci normalnej.

Zastosowane zostało podejście „Code First”, tzn. schemat bazy danych został wygenerowany na podstawie klas języka C# opatrzonych atrybutami.

Zapytania do bazy realizowane są poprzez zastosowanie mapera obiektowo-relacyjnego EntityFramework. Użycie mapera pozwala na znaczne uproszczeniem implementacji dostępu do danych zawartych w bazie, lecz wiąże się ze spadkiem wydajności. Jednak

## Schemat bazy danych

# Projekt klas

# Użytkownicy systemu

W systemie zostały wyróżnione 4 grupy użytkowników:

1. Gość
2. Uczeń
3. Nauczyciel
4. Administrator
   1. Super Administrator

Grupom tym odpowiadają określone uprawnienia odzwierciedlone w projekcie bazy danych i klas systemu.

# Projekt stron systemu

## Drzewo stron

Poniższy diagram przedstawia drzewo podstron serwisu wraz z nazwą użytkownika posiadającego uprawnienia o tej strony.

Rysunek 1 Drzewo podstron

screeny widoków

# Integracja systemu z forum

# Użyte technologie

* C# 5 - obiektowy język programowania rozwijany przez firmę Microsoft. Jego użycie w projekcie jest silnie związane z użyciem Visual Studio oraz ASP.NET.
* Visual Studio 2015 – zintegrowane środowisko programistyczne firmy Microsoft służące m.in. do tworzenia aplikacji z technologii ASP.NET
* ASP.NET MVC 5 – platforma aplikacyjna służąca do budowy aplikacji internetowych przy użyciu wzorca architektonicznego Model-View-Controller oparta na technologii ASP.NET. Użyta z powodu dobrej znajomości przez autorów niniejszej pracy.
* MS SQL Server 2012 – system zarządzania bazą danych, rozwijany przez firmę Microsoft. Użyty z powodu wysokiej kompatybilności z EntityFramework, komfortowego środowiska i dobrej znajomości przez autorów pracy.
* HTML5, CSS3, JavaScript – powszechnie obowiązujące języki wykorzystywane przy tworzeniu aplikacji internetowych po stronie klienta.
* Tiny MCE 4.2.7 – edytor języka HTML typu WYSIWYG napisany w języku JavaScript. Umożliwia edycję i podgląd dokumentu napisanego w języku HTML bezpośrednio w przeglądarce.
* jQuery – najpopularniejsza biblioteka JavaScript ułatwiająca nadanie dynamiki stronom Web.
* Bootstrap – biblioteka CSS/JavaScript ułatwiająca rozmieszczenie i projektowanie wyglądu elementów na stronie internetowej.
* Autofac 3.4.0 – biblioteka wspierająca wykorzystanie wzorca projektowego Inversion of Control. Użyta z powodu dobrej dokumentacji i wielu przykładów użycia dostępnych w Internecie.
* EntityFramework 6.1.3 - object-relational mapper, umożliwiający korzystanie z bazy danych przy użyciu klas napisanych w języku C#, obecnych w warstwie domeny projektu.
* Moq – najpopularniejszy framework dla technologii .Net wspierający mockowanie. Wykorzystywany przy testach jednostkowych.
* NUnit 2.6 – wykonany w języku C# framework dla technologii .Net usprawniający testy jednostkowe. Bazuje na JUnit napisanym w języku Java.
* MVC Forum