



Złożenie pracy online:

**2012-07-10 23:53:41**

Kod pracy:

**7879**

Krzysztof Reczkowski  
(nr albumu: 17923\*INF/LIC)

Praca inżynierska

## **Aplikacja ASP MVC3 wspomagająca pracę nauczycieli**

## **ASP Application MVC3 supporting the work of teachers**

Wydział: Informatyki

Kierunek: Informatyka

Specjalność:

Promotor: dr Krzysztof Przybycień

## Spis treści

|  |           |
|--|-----------|
| <b>SPIS TREŚCI .....</b>   | <b>2</b>  |
| <b>ROZDZIAŁ 1. WPROWADZENIE.....</b>   | <b>4</b>  |
| 1.1. CEL I ZAKRES PRACY.....   | 4         |
| 1.2 ZAŁOŻENIA STAWIANE APLIKACJI.....  | 5         |
| 1.3 UKŁAD PRACY.....   | 5         |
| <b>ROZDZIAŁ 2. ROZWIĄZANIA DOSTĘPNE NA RYNKU.....</b>                                    | <b>6</b>  |
| <b>ROZDZIAŁ 3. ZAKRES WYKONANYCH PRAC WRAZ Z OPISEM WYKORZYSTANYCH TECHNOLOGII .....</b> | <b>8</b>  |
| 3.1. WYKORZYSTANE TECHNOLOGIE.....   | 8         |
| 3.2. ZAKRES FUNKCJONALNOŚCI.....   | 10        |
| 3.2.1. Dyrektor.....   | 10        |
| 3.2.1.1 Zarządzanie rodzicami.....   | 10        |
| 3.2.1.2 Zarządzanie rokiem szkolnym.....   | 11        |
| 3.2.1.3 Zarządzanie przedmiotami.....  | 12        |
| 3.2.1.4 Zarządzanie danymi szkoły.....   | 12        |
| 3.2.1.5 Zarządzanie nazwami semestrów.....   | 13        |
| 3.2.1.6 Zarządzanie nauczycielami.....   | 13        |
| 3.2.1.7 Zarządzanie klasami.....   | 13        |
| 3.2.1.8 Zarządzanie uczniami.....  | 14        |
| 3.2.1.9 Zarządzanie klasami w szkole.....  | 14        |
| 3.2.1.10 Zarządzanie semestrami w szkole.....  | 16        |
| 3.2.2. Wychowawca.....   | 16        |
| 3.2.2.1 Generowanie kartek dla rodziców.....   | 16        |
| 3.2.2.2 Generowanie średnich z przedmiotów.....  | 17        |
| 3.2.2.3 Generowanie ocen semestralnych.....  | 17        |
| 3.2.2.4 Generowanie listy uczniów z miejscem na podpis.....                              | 17        |
| 3.2.2.5 Generowanie Informacji dla rodziców.....   | 17        |
| 3.2.3. Nauczyciel.....   | 17        |
| 3.2.3.1 Oceny postępów w nauce.....  | 17        |
| 3.2.3.2 Sprawdziany.....   | 19        |
| 3.2.4. Rodzic - opiekun.....   | 19        |
| 3.2.5. Uczeń.....  | 20        |
| <b>ROZDZIAŁ 4. TECHNICZNE ASPEKTY IMPLEMENTACJI.....</b>                                 | <b>20</b> |
| 4.1. PARTIAL VIEW .....  | 20        |

|  |           |
|--|-----------|
| 4.2. WALIDACJA.....                        | 21        |
| 4.3. ZABEZPIECZENIA.....                   | 23        |
| 4.4. WYBRANE FRAGMENTY IMPLEMENTACJI.....  | 23        |
| 4.4.1 Definiowanie nowego nauczyciela..... | 23        |
| 4.4.2 Definiowanie nowego ucznia.....      | 25        |
| <b>ROZDZIAŁ 5. PODSUMOWANIE.....</b>       | <b>26</b> |
| 5.1 KIERUNKI DALSZEGO ROZWOJU.....         | 27        |
| <b>SPIS TABEL I RYSUNKÓW.....</b>          | <b>28</b> |
| <b>LITERATURA.....</b>                     | <b>28</b> |

## **Rozdział 1. Wprowadzenie.**

### **1.1. Cel i zakres pracy.**

Wychowawca klasy, czy nauczyciel, ciągle zmierza się wieloma czasochłonnymi czynnościami czy obliczeniami związanymi z prowadzeniem dokumentacji oraz przygotowaniem samych pomocy dydaktycznych, tj. kart pracy, testów, sprawdzianów czy kartkówek na zajęcia lekcyjne.

Jako osoba, u której w rodzinie jest wielu nauczycieli, m. in. żona, – obserwując, od kilku lat, na co dzień pracę nauczyciela, a przede wszystkim czynności wykonywane w domu, po zajęciach szkolnych – powziąłem sobie za cel, ułatwienie tego procesu. Powstał wówczas pomysł projektu "Nauczyciel - Help" mający na celu stworzenie skutecznych narzędzi wspomagających codzienną pracę nauczyciela a w szczególności wychowawcy klasowego.

Przedmiotem projektu mojej pracy inżynierskiej stało się zaprojektowanie i utworzenie systemu wspierającego pracę dyrektora, wychowawcy czy nauczyciela. W ramach projektu zaprojektowano i zaimplementowano aplikację webową. System został zaprogramowany w środowisku Microsoft Visual Studio 2010.

Podczas definiowania zakresu funkcjonalności aplikacji, wziąłem pod uwagę, głównie, czynności - związane z procesem nauczania - najczęściej wykonywane przez nauczycieli, które można w pewien sposób usprawnić lub wręcz całkowicie zautomatyzować.

Autoryzacja do systemu została podzielona na pięć głównych ról:

- Dyrektor – możliwość zarządzania, m. in. przydział czynności, tj. np.: danych przedmiotów nauczycielom uczącym w danym roku szkolnym i w danej klasie/oddziale, rozporządzanie pracą poszczególnych nauczycieli, (ma równocześnie uprawnienia nauczyciela i/lub wychowawcy);
- Wychowawca – możliwość generowania różnego typu niezbędnych dokumentów związanych z bieżącym prowadzeniem dokumentacji w procesie edukacji, (ma równocześnie uprawnienia nauczyciela);
- Nauczyciel – możliwość m.in.: wprowadzania ocen poszczególnym uczniom z nauczanych przez siebie przedmiotów;
- Rodzic – możliwość podglądu wprowadzonych ocen swojego dziecka;
- Uczeń – możliwość poglądu swoich osiągnięć.

## 1.2 Założenia stawiane aplikacji.

„Nauczyciel – Help” to program, przeznaczony do zarządzania wszelkimi danymi o uczniach szkół podstawowych, gimnazjalnych, licealnych i policealnych, zawodowych, technicznych oraz wszystkich innych gromadzących i zarządzających informacjami o uczniach. Program znajduje zastosowanie wszędzie tam, gdzie stosuje się system sprawdzania wiedzy i oceniania. „Nauczyciel - Help” jest serwisem skierowanym przede wszystkim dla nauczycieli. Umożliwia samodzielne zarządzanie kontem: ocenami ucznia, generowaniem testów, ale także pozwala dyrekcji szkoły na sprawne przydzielanie uprawnień dla nauczycieli, rodziców oraz uczniów. Rozbudowany system uprawnień i konfiguracji pozwala na obsługę nawet dużych placówek.

Dzięki aplikacji można, m.in.:

Gromadzić informacje:

- wprowadzać dane o uczniach (imię, nazwisko ucznia, imiona rodziców, telefon kontaktowy, adres kontaktowy);
- wprowadzać dane o przedmiotach (nazwa przedmiotu, nazwisko nauczyciela prowadzącego);
- przydzielać każdemu nauczycielowi przedmioty z puli już wprowadzonych do programu (późniejsze modyfikacje są możliwe);
- wprowadzać dla każdego ucznia oceny cząstkowe ze wszystkich przydzielonych mu przedmiotów;
- generować tzw. karteczki dla rodziców, z informacją o dotychczasowych osiągnięciach poszczególnych uczniów np. na spotkania wywiadowcze;
- prowadzić rozbudowane zbiory (archiwum) pomocy dydaktycznych, tj. np.: wzory przeprowadzonych testów, sprawdzianów, kartkówek w poszczególnych klasach, ze wszystkich przedmiotów;
- generować testy, sprawdziany, kartkówki.
  - Drukować:
- oceny cząstkowe ucznia dla rodziców na okresowe spotkania wywiadowcze.

## 1.3 Układ pracy.

Niniejsza praca została podzielona na cztery główne rozdziały:

- I. Wstęp: wprowadzenie do tematu pracy; w tej części zostały zawarte ogólne informacje na temat prezentowanej aplikacji.

- II. Część opisowa: dziedzina problemu; w następującej części zostały zawarte informacje na temat m.in. istniejących już na rynku podobnych aplikacji.
- III. Część praktyczna: ta część prezentuje użyte oprogramowanie do stworzenia aplikacji oraz opis biblioteki iText.
- IV. Techniczne aspekty implementacji wybranych części kodów źródłowych, wraz z opisem.
- V. Podsumowanie, przedstawienie wniosków.

## **Rozdział 2. Rozwiązania dostępne na rynku.**

Współcześnie nowoczesne technologie wkraczają do szkół. Dotychczasowe papierowe dzienniki powoli odchodzą w zapomnienie. Za przyzwoleniem Ministerstwa Edukacji Narodowej<sup>1</sup> (Rozporządzenie Ministra Edukacji Narodowej, mówiące m.in. o tym, że nauczyciele mogą stosować wyłącznie elektroniczne dzienniki, wcześniej szkoły mogły prowadzić dzienniki elektroniczne, ale wciąż wymagane było prowadzenie ich w wersji papierowej; nowelizacja ta, nie przesądza jednak o obowiązkowym ich prowadzeniu w szkołach) następuje powolny proces zastępowania ich dziennikami elektronicznymi.

Tego typu nowoczesne narzędzia elektroniczne przede wszystkim wspomagają pracę szkoły. Nauczyciele na bieżąco mogą uzupełniać dokumentację, wpisując m.in. oceny z poszczególnych przedmiotów, czy uwagi o zachowaniu ucznia.

Programy tego typu są dodatkową formą kontaktu między szkołą a rodzicami ucznia. Łatwe w obsłudze narzędzie pozwala bowiem na dostęp rodzica do informacji o postępach w nauce swojego dziecka, już nie przez osobistą wizytę w szkole lecz przez Internet. Dzięki bieżącym informacjom o ocenach swojego dziecka, rodzice mają możliwość stałej kontroli jego postępów w nauce. A dla nauczycieli/wychowawców jest to nieoceniona możliwość komunikacji – czasem tak utrudnionej - z rodzicami ucznia.

Z tego typu udogodnień korzystać mogą także uczniowie. O każdej porze, mają wgląd do informacji o sobie. Programy tego typu, umożliwiają bezpośredni dostęp do własnych ocen, uwag dotyczących zachowania i innych ważnych informacji.

Aplikacja jest dedykowana głównie dla szkół podstawowych, gimnazjów oraz szkół średnich. Obecnie na rynku pojawia się coraz więcej aplikacji których celem jest

---

<sup>1</sup> Rozporządzenie Ministra Edukacji Narodowej z 24 sierpnia 2010 r. zmieniające rozporządzenie w sprawie sposobu prowadzenia przez publiczne przedszkola, szkoły i placówki dokumentacji przebiegu nauczania, działalności wychowawczej i opiekuńczej oraz rodzajów tej dokumentacji (Dz. U. nr 116, poz. 977).

usprawnienie pracy nauczyciela. Wymienić tu można kilka najpopularniejszych ogólnodostępnych programów:

*Dziennik 3.09*: (<http://www.teacher.webd.pl/plan309.php>). Program w wersji 3.09 jest narzędziem wspomagającym planowanie i realizację zadań związanych z organizacją roku szkolnego. Kompleksowe podejście do organizacji roku szkolnego stanowi istotę i siłę programu i jest rozwiązaniem nowatorskim na rynku. Program typowo desktopowy; nauczyciele nie mają możliwości generowania dokumentów, np. testów. Ponadto, wychowawcy są zmuszeni do samodzielnego wprowadzania ocen z poszczególnych przedmiotów.

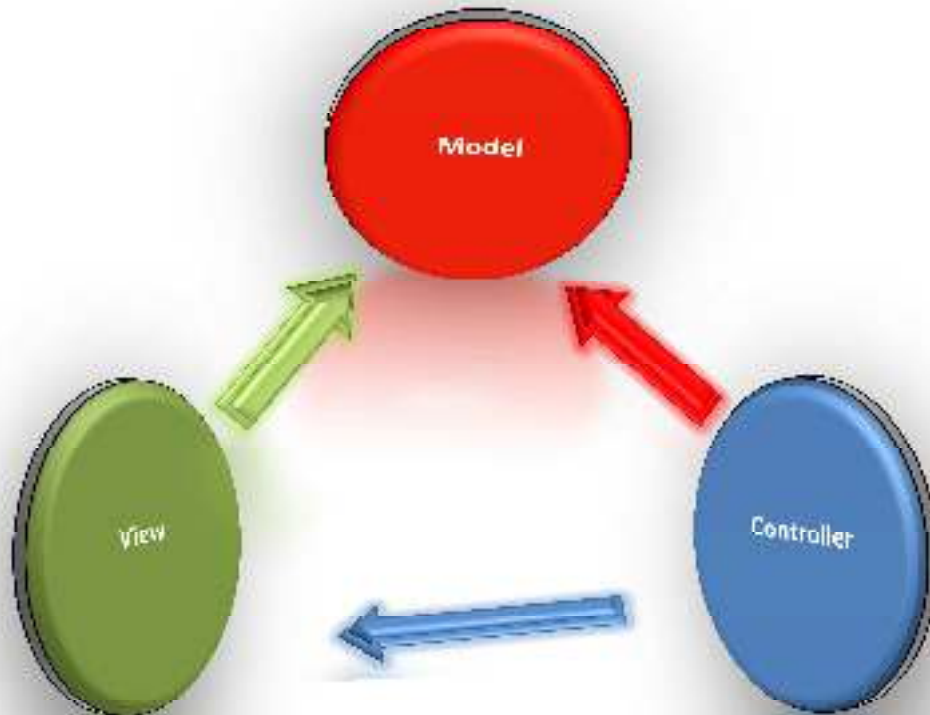
*Zestawienie Ocen*: (<http://edziennik.org/?edziennik=edziennik&dziennik=opis>). Aplikacja desktopowa oraz internetowa, (w zależności od wersji) - brak możliwości zarządzania dostępem nauczycieli przez dyrekcję, oraz generowania dokumentacji. Program umożliwia wprowadzanie informacji o uczniach, przedmiotach, przydzielania każdemu uczniowi przedmiotów, wprowadzanie ocen i ilości godzin nieobecnych, itp.. Program pozwala także na generowanie wydruków i zestawień statystycznych klasy, pomocnych przy np. wywiadówkach. Interfejs Zestawienia Ocen na pierwszy rzut oka może wydać się skomplikowany, ale jego obsługa nie sprawia większego kłopotu.

*Dziennik Nauczyciela 1.0.0 223pl*: [http://www.pcworld.pl/ftp/pc\\_14340/Dziennik.Nauczyciela.100223.pl.html](http://www.pcworld.pl/ftp/pc_14340/Dziennik.Nauczyciela.100223.pl.html). Program przeznaczony głównie do analizy ocen uczniów oraz tworzenia statystyk, tabel zbiorczych, wykresów.

Obecnie na rynku, istnieje bardzo dużo różnego typu oprogramowań dla szkół czy wspomagających pracę indywidualnych nauczycieli, jednakże żadna ze znanych mi aplikacji nie gromadzi, tych funkcjonalności, które ja zaproponowałem w aplikacji: „Nauczyciel – Help”. W większości przypadków, szkoły - jeśli już korzystają z tego typu pomocy, to są to - najczęściej aplikacje do generowania świadectw rocznych czy końcowych.

## Rozdział 3. Zakres funkcjonalności wraz z opisem wykorzystanych technologii.

### 3.1. Wykorzystane technologie.



Rysunek 1 Model-View-Controller

W pracy starałem się wykorzystać najnowsze znane mi technologie wytwarzania aplikacji biznesowych.

Do budowy architektury aplikacji wykorzystałem model Model-View-Controller(MVC), który dzieli aplikację na następujące trzy główne elementy: model, widok i kontroler, które są powiązane ze sobą w sposób przedstawiony na powyższym rysunku.

Środowisko programistyczne, które wybrałem do zrealizowania projektu, to Visual Studio 2010, w którym użyłem ASP.NET MVC 3 z silnikiem Razor. Jest to bardzo rozbudowane narzędzie dla programistów, umożliwia m.in.: testy jednostkowe, testy bazy, testy web, testy UI, testy obciążeniowe, – które są nieocenioną pomocą przy tworzeniu złożonych aplikacji biznesowych. Visual Studio daje możliwość zmapowania bazy danych na obiekty w .NET – co, w dużym stopniu, ułatwia zapytania do bazy danych, gdyż można użyć zapytań Linq, które w jednakowy sposób działają na wszystkich kolekcjach danych jak np. listach. Entity Framework oferuje różne możliwości dostępu do bazy danych; w projekcie skorzystałem z następującego rozwiązania: stworzyłem bazę danych w MS SQL Server 2008



i na jej podstawie za pomocą ADO .NET Entity Framework odwzorowałem dane na architekturę obiektową (ORM).

System zarządzania bazą danych, który wybrałem, to: MS SQL Server. Wyboru takiego dokonałem m. in. z uwagi na bezpieczeństwo, które gwarantuje. Najważniejsze cechy systemu to:

- różne tryby autoryzacji do bazy danych (Windows i SQL Server),
- szyfrowanie danych <sup>2</sup>,
- zastosowanie certyfikatów.

W projekcie generuję dokumenty .pdf za pomocą biblioteki iText <sup>3</sup>, która w łatwy i szybki sposób umożliwia utworzenie dokumentów na podstawie różnych formatów. Przykładowy kod do utworzenia dokumentu wraz z wymaganymi krokami w iText:

1. Ustawienie formatowania dla dodawanego tekstu:

```
var fontNaglowekPytanie =
```

```
    FontFactory.GetFont(FontFactory.TIMES, "CP1250", 11, Font.BOLD);
```

2. Utworzenie obiektu Document:

```
var document = new Document();
```

```
PdfWriter.GetInstance(document, new FileStream("plik", FileMode.Create, FileAccess.Write));
```

3. Utworzenie tabeli z 1 kolumną:

```
PdfPTable sprawdzian = new PdfPTable(1);
```

4. Utworzenie komórki oraz dodanie do niej tekstu i sformatowanie:

```
PdfPCell daneSzkołyNaglowek =
```

```
    new PdfPCell(new Phrase("Treść do wstawienia", fontNaglowekPytanie));
```

```
daneSzkołyNaglowek.HorizontalAlignment = Element.ALIGN_LEFT;
```

```
daneSzkołyNaglowek.Border = 0;
```

5. Dodanie komórki do tabeli:

```
sprawdzian.AddCell(daneSzkołyNaglowek);
```

6. Dodanie tabeli do dokumentu:

```
document.Add(sprawdzian);
```

7. Zamknięcie dokumentu:

```
document.Close();
```

---

<sup>2</sup> <http://technet.microsoft.com/pl-pl/library/mechanizm-szyfrowania-baz-danych-tde-w-sql-server-2008.aspx>

<sup>3</sup> <http://www.itextpdf.com/>

Jednym z atutów biblioteki iText, jest udostępnienie jej na licencji GPL. Dzięki zastosowaniu MVC, praca została podzielona na 3 oddzielne warstwy, dzięki czemu, budowa projektu mogła zostać usystematyzowana. Natomiast, naniesienie późniejszych zmian w wyglądzie aplikacji, nie niesie za sobą radykalnych zmian w logice aplikacji.

### 3.2. Zakres funkcjonalności.

Aplikacja „Nauczyciel - Help” ma szereg różnych, użytecznych funkcjonalności, które moim zdaniem, są niezbędne do zapewnienia wygodnej pracy użytkownika.

#### 3.2.1. Dyrektor.

##### 3.2.1.1 Zarządzanie rodzicami.



| lp. | imie nazwisko      | adres                              | telefon     |
|-----|--------------------|------------------------------------|-------------|
| 1.  | Oliver Kłobukowski | ul. Towarowa 10 33 - 180 Lubaszowa | 788 478 545 |
| 2.  | Zbigniew Jackowski | ul. Dowolna 2                      | 111 222 111 |
| 3.  | Barbara Ciecjowska | ul. Partyzantow 4                  | 454 545 454 |
| 4.  | Zdzisław Rabak     | ul. Polna 5                        | 787 787 878 |
| 5.  | Martyna Dowlina    | ul. Towarowa 7                     | 787 456 789 |
| 6.  | Marcin Wróbel      | ul. Krynicka 24111                 | 788 786 290 |

Rysunek 2 Prezentacja listy rodziców.

Każda strona prezentująca dane jest podzielona na trzy podstawowe elementy:

1. Filtrowanie (jeżeli ilość danych tego wymaga).
2. Możliwość edycji (wybierając element z listy możemy go edytować).
3. Dodawanie nowego elementu (po prawej stronie na górze).

Dzięki temu, użytkownik, który zaznajomi się z jedną stroną prezentującą dane, nie będzie miał problemów z dalszą pracą.



Rysunek 3 Edycja danych rodzica.

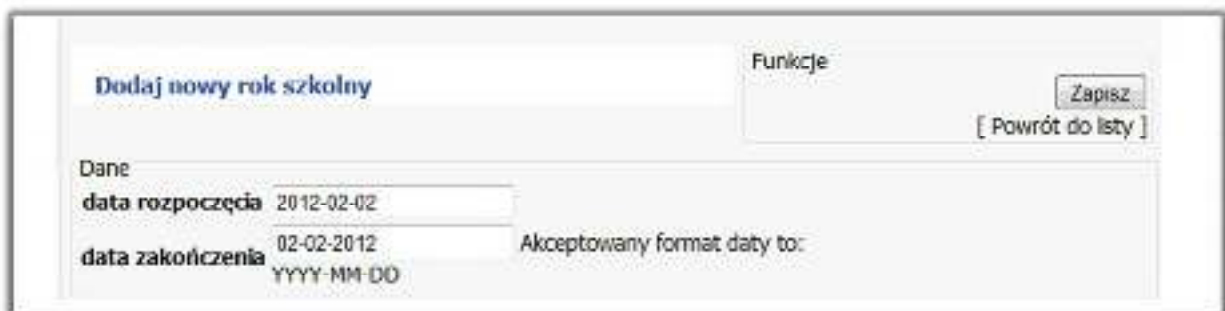
Każde strona umożliwiające edycje danych jest podzielona na dwa podstawowe elementy:

1. Funkcje z dwoma opcjami (zapisz oraz powrót do listy).
2. Panel umożliwiający edycję danych.

Formularz edycji danych jest w większości przypadków identyczny z formularzem tworzącym nowy element.

### 3.2.1.2 Zarządzanie rokiem szkolnym.

Wyświetlanie danych jest na identycznych zasadach, jak w przypadku zarządzania rodzicami.

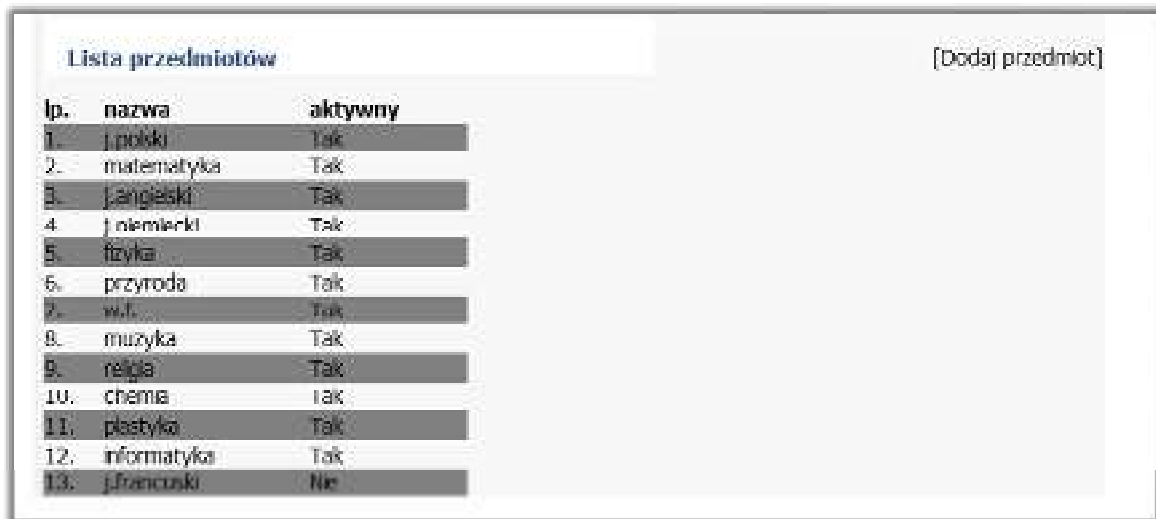


Rysunek 4 Walidacja wprowadzonego roku szkolnego.

Podczas wprowadzania danych w formularzach, które tego wymagały, została zastosowana walidacja danych. Gdy edytujemy dane wybranego roku szkolnego, mamy możliwość zdefiniowania nowego semestru (zostajemy przeniesieni do strony umożliwiającej zarządzanie semestrami).

### 3.2.1.3 Zarządzanie przedmiotami

Wyświetlanie danych przebiega w sposób identyczny jak w przypadku zarządzania rodzicami.



| lp. | nazwa       | aktywny |
|-----|-------------|---------|
| 1.  | j.polski    | Tak     |
| 2.  | matematyka  | Tak     |
| 3.  | j.angielski | Tak     |
| 4.  | j.niemiecki | Tak     |
| 5.  | fizyka      | Tak     |
| 6.  | przyroda    | Tak     |
| 7.  | w.t.        | Tak     |
| 8.  | muzyka      | Tak     |
| 9.  | religia     | Tak     |
| 10. | chemia      | Tak     |
| 11. | plastyka    | Tak     |
| 12. | informatyka | Tak     |
| 13. | j.francuski | Nie     |

Rysunek 5 Lista przedmiotów w szkole.

Lista przedmiotów, która jest przedstawiona na powyższym rysunku, daje możliwość - podczas dodawania np. nowego nauczyciela – zdefiniowania od razu, jakie ma uprawnienia np. do nauczania danych przedmiotów.

### 3.2.1.4 Zarządzanie danymi szkoły.

Jest to panel odmienny od pozostałych – nie występuje w nim lista danych szkoły oraz możliwość dodania nowej szkoły, możemy tutaj jedynie edytować dane.



| Edycja danych szkoły |                    | Funkcje             |
|----------------------|--------------------|---------------------|
|                      |                    | Zapisz              |
|                      |                    | [ Powrót do listy ] |
| Dane szkoły          |                    |                     |
| Nazwa szkoły         | Podstawowa         |                     |
| Imię                 | Mikolaja Kopernika |                     |
| Ulica                | Tamowska           |                     |
| Nr                   | 1                  |                     |
| Kod pocztowy         | 33-113             |                     |
| Miejscowość          | Tuchów             |                     |

Rysunek 6 Edycja danych szkoły.

### 3.2.1.5 Zarządzanie nazwami semestrów.

Wyświetlanie danych jest na identycznych zasadach, jak w przypadku zarządzania rodzicami. Funkcjonalność ta, umożliwia zesłownikowanie nazw semestrów – które potem, wielokrotnie pojawiają się w listach rozwijalnych.

### 3.2.1.6 Zarządzanie nauczycielami.

Wyświetlanie danych jest na identycznych zasadach, jak w przypadku zarządzania rodzicami. Jest to panel jednak nieco odmienny od pozostałych, gdyż umożliwia dodanie nauczyciela do systemu oraz jednocześnie zdefiniowanie jego uprawnień nauczycielskich.

**Dodawanie nowego nauczyciela**

Funkcje

Zapisać

[ Powrót do listy ]

Dane nauczyciela

Imię

Nazwisko

Nr telefonu

Login:

Hasło

Powtórz Hasło

Aktywny  Aktywny

Przedmioty które może uczyć nauczyciel

- j.polski
- matematyka
- j.angielski
- j.niemiecki
- fizyka
- przyroda
- w.f.
- muzyka
- religia
- chemia
- plastyka
- informatyka
- j.francuski

Rysunek 7 Dodawanie nowego nauczyciela.

Tutaj także, dane o przedmiotach są zaciągane z zdefiniowanych przedmiotów w systemie.

### 3.2.1.7 Zarządzanie klasami.

Wyświetlanie danych jest na identycznych zasadach, jak w przypadku zarządzania rodzicami. Funkcjonalność ta, umożliwia zesłownikowanie nazw klas, które istnieją w szkole.

### **3.2.1.8 Zarządzanie uczniami.**

Wyświetlanie danych jest na identycznych zasadach, jak w przypadku zarządzania rodzicami. Proces dodawania nowego ucznia jest podzielony na dwa etapy:

Etap 1 - to wprowadzenie danych ucznia.

Etap 2 - to uzupełnienie danych o opiekunach danego ucznia.

Aplikacja, gdy znajdzie w systemie rodziców o podobnym nazwisku co wprowadzany do systemu uczeń, to wyświetli ich w liście sugerowanych rodziców. System potrafi także np. odciąć od nazwiska końcówkę –ski oraz –ska i wyświetlić jednocześnie obu rodziców. Jeżeli brak jest zdefiniowanego rodzica, w systemie mamy możliwość - za pomocą opcji: [dodaj rodzica] - dodać nowego; zostaniemy wówczas przeniesieni do formularza dodawania rodzica, gdzie pole: [nazwisko] jest automatycznie uzupełniane. Po zatwierdzeniu rodzica zostajemy przeniesieni do edycji danych ucznia.

### **3.2.1.9 Zarządzanie klasami w szkole.**

Wyświetlanie danych jest na identycznych zasadach, jak w przypadku zarządzania rodzicami. Jest to jedna z najbardziej rozbudowanych funkcjonalności w systemie. Dodawanie nowej klasy jest podzielone na trzy etapy:

Etap 1 - to wprowadzenie podstawowych danych klasy, czyli:

- a. wychowawcy (wybieramy z listy),
- b. numeru klasy (wybieramy z listy),
- c. semestru (wybieramy z listy),
- d. oddziału (wpisujemy ręcznie),
- e. rocznika (jest to rocznik, który obowiązuje w danej klasie).

Etap 2 - to wybranie uczniów należących do danej klasy z listy sugerowanych; po zatwierdzeniu wybranych uczniów zobaczymy ekran podobny do następującego:

**Dane klasy**

**Edycja danych klasy**

[Przypisanie przedmiotów do klasy] [Przypisanie przedmiotów do nauczycieli]

**Funkcje**  
[Zapisz]  
[Powrót do listy]

Dane klasy

**Wychowawca** Antoni Batowski

**Numer klasy** I

**Semestr** letni 2011 / 2012

**Oddział** B

**Rocznik** 1995

Lista dodanych uczniów

| lp | imię i nazwisko                                      |
|----|--|
| 1. | <input checked="" type="checkbox"/> Adam Romanowski  |
| 2. | <input checked="" type="checkbox"/> Adam Adamski     |
| 3. | <input checked="" type="checkbox"/> Piotr Rabak      |
| 4. | <input checked="" type="checkbox"/> Bartłomiej Rabak |

Lista sugerowanych uczniów

| lp | imię i nazwisko |
|----|-----------------|
|----|-----------------|

Rysunek 8 Zarządzanie uczniami w szkole.

Etap 3 - to przypisanie nauczycieli do przedmiotów:

### Lista nauczycieli prowadzących w klasie

Funkcje  
  
[ Powrót do listy ]

Dane klasy  
Klasa: I B  
Wychowawca Krzysztof Reczkowski

Nauczyciele - uprawnienia

**j.polski**

Krzysztof Reczkowski  
 Marzena Reczkowska  
 Mieczysław Faber  
 Marcin Twolek

**matematyka**

Oleksy Adamiak  
 Aleksandra Kowalska  
 Antoni Batowski

**j.angielski**

**j.niemiecki**

Krzysztof Reczkowski  
 Aleksandra Kowalska  
 Antoni Batowski

**fizyka**

**przyroda**

Mieczysław Faber

**w.f.**

Krzysztof Reczkowski

Rysunek 9 Przypisanie nauczyciela do danego przedmiotu.

### 3.2.1.10 Zarządzanie semestrami w szkole.

Wyświetlanie danych jest na identycznych zasadach jak w przypadku zarządzania rodzicami. Tutaj definiujemy aktualny obowiązujący semestr w szkole.

### 3.2.2. Wychowawca.

#### 3.2.2.1 Generowanie kartek dla rodziców .

Po wybraniu opcji: kartki dla rodziców z menu, tworzony jest dokument w formacie .pdf.



### **3.2.2.2 Generowanie średnich z przedmiotów.**

Po wybraniu opcji: średnie z przedmiotów z menu, tworzony jest dokument w formacie .pdf.

### **3.2.2.3 Generowanie ocen semestralnych.**

Po wybraniu opcji: oceny semestralne z menu, tworzony jest dokument w formacie .pdf.

### **3.2.2.4 Generowanie listy uczniów z miejscem na podpis.**

Po wybraniu opcji: Lista uczniów + podpis z menu, tworzony jest dokument w formacie .pdf.

### **3.2.2.5 Generowanie Informacji dla rodziców.**

Po wybraniu opcji: Informacja dla rodziców z menu, tworzony jest dokument w formacie .pdf.

## **3.2.3. Nauczyciel.**

### **3.2.3.1 Oceny postępów w nauce.**

Funkcjonalność ta, umożliwia dodawanie nowych ocen oraz edycję już wprowadzonych, rozpoznawalny format ocen to cyfra, która może zostać poprzedzona znakiem +, -, =.

**OCENY POSTĘPÓW W NAUCE**

**Zajęcia edukacyjne (przedmiot) j.polski**

**Imię i nazwisko nauczyciela Krzysztof Reczkowski**

Lista ocen

sprawdzian z 2012-06-02  
aktywność

| Nr  | Imię i nazwisko   |   |  |
|-----|-------------------|---|--|
| 1.  | Adam Romanowski   | S |  |
| 2.  | Adam Adamski      |   |  |
| 3.  | Piotr Rabak       |   |  |
| 4.  | Bartłomiej Rabak  |   |  |
| 5.  | Bartek Stachon    |   |  |
| 6.  | Tomek Zdzisław    |   |  |
| 7.  | Olaf Dobry        |   |  |
| 8.  | Tomek Tworek      |   |  |
| 9.  | Zdzisław Dobry    |   |  |
| 10. | Zdzisław Wróbel   |   |  |
| 11. | Damian Kębukowski | 3 |  |

Rysunek 10 Oceny postępów w nauce.

Aby dodać nową ocenę, wybieramy z menu po lewej stronie [nowa ocena], uzupełniamy wymagane pola i wybieramy: [zapisz].

**OCENY POSTĘPÓW W NAUCE**

**Zajęcia edukacyjne (przedmiot) j.polski**  
**Imię i nazwisko nauczyciela Krzysztof Reczkowski**

Opis dla oceny (za m):

Wystaw oceny Zapisz

| Lp  | Imię i nazwisko      | Ocena                |
|-----|----------------------|----------------------|
| 1.  | Adam Romanowski      | <input type="text"/> |
| 2.  | Adam Adamski         | <input type="text"/> |
| 3.  | Piotr Rabak          | <input type="text"/> |
| 4.  | Bartłomiej Rabak     | <input type="text"/> |
| 5.  | Bartek Stachoń       | <input type="text"/> |
| 6.  | Tomek Zdzisław       | <input type="text"/> |
| 7.  | Olaf Dobry           | <input type="text"/> |
| 8.  | Tomek Tworek         | <input type="text"/> |
| 9.  | Zdzisław Dobry       | <input type="text"/> |
| 10. | Zdzisław Wróbel      | <input type="text"/> |
| 11. | Damian Kłębukowski 3 | <input type="text"/> |

Rysunek 11 Dodawanie nowej oceny.

W celu edycji ocen należy nacisnąć na [opis oceny], wówczas zostanie wyświetlony ekran identyczny, jak przedstawiony na rys. 10.

### 3.2.3.2 Sprawdziany.

Każdy nauczyciel prowadzący dany przedmiot może zdefiniować sprawdzian, który jest widoczny dla wszystkich nauczycieli prowadzących identyczny przedmiot w danej klasie, dzięki temu, w szkole zostanie stworzona baza sprawdzianów dla uczniów, która w dużym stopniu usprawni i ułatwi pracę.

**Sprawdziany z przedmiotu: fizyka dla klasy: I**

| Dostępne sprawdziany |                           |
|----------------------|---------------------------|
| lp.                  | nazwa działu              |
| 1.                   | Podstawowe wzory i zasady |
| 2.                   | Wzory                     |

| lista sprawdzianów                   |  |
|--------------------------------------|--|
| Podstawowe prawa ( Zdzisław Maniak ) |  |
| pytania otwarte ( Zdzisław Maniak )  |  |
| Podstawowe wzory ( Zdzisław Maniak ) |  |
| Wzory ( Zdzisław Maniak )            |  |

Rysunek 12 Archiwum. Lista sprawdzianów.

### 3.2.4. Rodzic - opiekun.

Ma możliwość sprawdzenia ocen swojego podopiecznego.

### 3.2.5. Uczeń.

Ma możliwość sprawdzenia swoich ocen.

## Rozdział 4. Techniczne aspekty implementacji

### 4.1. Partial View

W panelu dyrektora kilkakrotnie używałem funkcjonalności, takiej jak: [zapisz] oraz [powrót do listy], aby nie powtarzać wielokrotnie tego samego kodu oraz zachować identyczny wygląd na każdej stronie, utworzyłem plik o nazwie „Views/Shared/\_DodawanieFunkcjePartial.cshtml”, którego zadaniem było wyświetlanie ww. opcji na wybranych stronach za pomocą funkcji: @Html.Partial("\_LogOnPartial"). Zawartość pliku podana poniżej:

```
<div id="content-Zatwierdzenie">
  <fieldset>
    <legend id="legend">Funkcje</legend>
    <div>
      <input type="submit" value="Zapisz" />
    </div>
    <div id="menudisplay">
      @Html.ActionLink("[ Powrót do listy ]", "Index")
    </div>
  </fieldset>
</div>
```

Dodatkowo, wielokrotnie na stronach używałem filtrowania po nazwiskach, więc w identyczny sposób jak poprzednio, utworzyłem plik o nazwie: „Views/Shared/\_FiltrowaniePartial.cshtml”:

```
@model Szkola_1_0.Models.Helpers.FiltrowanieHelpers
@using (Html.BeginForm())
{
  @Html.ValidationSummary(true)
  <div id="content-Dane">
    <fieldset>
      <legend id="legend">Filtrowanie</legend>
```

```
<p>
```

```
Nazwisko: @Html.EditorFor(model => model.NazwiskoFiltr)
```

```
<input type="submit" value="Filtruj" />
```

```
</p>
```

```
</fieldset>
```

```
</div>
```

```
}
```

W celu poprawnego działania tej funkcjonalności, każda klasa, która miała obsługiwać filtrowanie, musiała dziedziczyć po klasie `FiltrowanieHelpers`. Definicja, klasy `FiltrowanieHelpers`:

```
public class FiltrowanieHelpers
{
    public string NazwiskoFiltr { get; set; }
}
```

Przykładowa klasa wykorzystująca filtrowanie:

```
public class FiltrowanieRocziceHelpers : FiltrowanieHelpers
{
    public List<Rodzice> IQueryRodzice { get; set; }
}
```

#### 4.2. Walidacja.

Każda aplikacja wymaga - w pewnym miejscu - zastanowienia się nad sprawdzeniem poprawności danych wprowadzonych przez użytkownika. W przestrzeni nazw: `System.ComponentModel.DataAnnotations` znajduje się klasa `ValidationAttribute`, po której, jeżeli wykonamy dziedziczenie, umożliwi nam stworzenie własnego mechanizmu sprawdzania poprawności danych. Przykładowa walidacja daty użyta w aplikacji:

```
public class WalidacjaDatyAttribute : ValidationAttribute
{
    protected override ValidationResult IsValid
        (object value, ValidationContext validationContext)
    {
        string sErrorMessage = "Akceptowany format daty to: YYYY-MM-DD";
    }
}
```

```
String data = (String)value;
int year=0;
int month=0;
int day=0;
try
{
    year = int.Parse(data.Substring(0,4));
    month = int.Parse(data.Substring(5,2));
    day = int.Parse(data.Substring(8,2));
    if (data[4] == '-' && data[7] == '-')
    {
        return ValidationResult.Success;
    }
    else
    {
        sErrorMessage = "Akceptowany format daty to: YYYY-MM-DD";
        return new ValidationResult(sErrorMessage);
    }
}
catch
{
    sErrorMessage = "Akceptowany format daty to: YYYY-MM-DD";
    return new ValidationResult(sErrorMessage);
}
}
```

Przykład jej wykorzystania w klasie: RokSzkolnyHelpers:

```
public class RokSzkolnyHelpers
{
    public int idRokSzkolny { get; set; }
    [Required]
    [WalidacjaDaty]
    [Display(Name = "data rozpoczęcia")]
}
```

```
public String DataRozpoczecia { get; set; }  
[Required]  
[WalidacjaDaty]  
[Display(Name = "data zakończenia")]  
public String DataZakonczenia { get; set; }  
}
```

W podobny sposób utworzyłem sprawdzanie poprawności wpisania roku; nazwa pliku, w którym jest utworzona klasa sprawdzająca poprawności daty; *Models/ValidatorMethod/WalidacjaRocznikaAttribute.cs*.

### 4.3. Zabezpieczenia

Dane wprowadzane przez użytkownika takie jak: imiona, nazwiska, adresy, numery telefonów są szyfrowane prostym algorytmem XOR, w celu poprawy bezpieczeństwa (żeby administrator bazy danych nie mógł w prosty sposób sprawdzić danych użytkowników). Zastosowanie algorytmu symetrycznego ułatwiło mi w dużym stopniu filtrowanie danych po nazwiskach.

### 4.4. Wybrane fragmenty implementacji

#### 4.4.1 Definiowanie nowego nauczyciela.

W celu, dodawania nowego nauczyciela oraz utworzenia łatwego sposobu definiowania przedmiotów których może uczyć, - dla użytkownika systemu - utworzyłem następującą klasę:

```
public class DaneNauczyciela  
{  
    [HiddenInput(DisplayValue = false)]  
    public int IdNauczyciela { get; set; }  
    [StringLength(50, ErrorMessage = "Maksymalnie 50 znaków")]  
    [Required(AllowEmptyStrings = true, ErrorMessage = "Uzupełnij imię.")]  
    [Display(Name = "Imię")]  
    public string Imie { get; set; }  
    [StringLength(50, ErrorMessage = "Maksymalnie 50 znaków")]  
    [Required(AllowEmptyStrings = true, ErrorMessage = "Uzupełnij nazwisko.")]  
    [Display(Name = "Nazwisko")]
```

```
public string Nazwisko { get; set; }  
[StringLength(20, ErrorMessage = "Maksymalnie 20 znaków")]  
[Required(AllowEmptyStrings = true, ErrorMessage = "Uzupełnij nr telefonu lub wpisz  
brak")]  
[Display(Name = "Nr telefonu")]  
public string Telefon { get; set; }  
[StringLength(20, ErrorMessage = "Maksymalnie 20 znaków")]  
[Required(AllowEmptyStrings = true, ErrorMessage = "Uzupełnij login")]  
[Display(Name = "Login")]  
public string Login { get; set; }  
[Display(Name = "Aktywny")]  
public bool Aktywny { get; set; }  
[Required]  
[ValidatePasswordLength(ErrorMessage = "Hasło musi być dłuższe niż 6 znaków")]  
[DataType(DataType.Password)]  
[Display(Name = "Hasło")]  
public string Haslo { get; set; }  
[DataType(DataType.Password)]  
[Display(Name = "Powtórz Hasło")]  
[Compare("Haslo", ErrorMessage = "Hasła niepoprawne")]  
public string ConfirmHaslo { get; set; }  
public List<UprawnieniaNauczycieliDodawanie> ListaUprawnien { get; set; }  
}
```

Tutaj, we właściwości o nazwie: ListaUprawnien, przechowuje listę przedmiotów wraz z opcją czy dany nauczyciel może jej uczyć. Definicja klasy: UprawnieniaNauczycieliDodawanie, poniżej:

```
public class UprawnieniaNauczycieliDodawanie  
{  
    [HiddenInput(DisplayValue = false)]  
    public int idPrzedmiotu { get; set; }  
    public string Nazwa { get; set; }  
    public bool czyMaUprawnienia { get; set; }  
}
```



#### 4.4.2 Definiowanie nowego ucznia.

Opcja: dodawanie uczniów - w podobny sposób postępowalem w przypadku dodawania nowego ucznia, gdzie miałem dwie dodatkowe listy, które przechowywały aktualnie dodanych opiekunów, oraz listę sugerowanych opiekunów tworzonych na podstawie podobieństwa nazwisk; poniżej definicja obu klas:

```
public class DodawanieUczniow {
    public int IdUcznia { get; set; }
    [StringLength(50, ErrorMessage = "Maksymalnie 50 znaków")]
    [Required(AllowEmptyStrings = true, ErrorMessage = "Uzupełnij imię.")]
    [Display(Name = "Imię")]
    public string Imie { get; set; }
    [StringLength(50, ErrorMessage = "Maksymalnie 50 znaków")]
    [Required(AllowEmptyStrings = true, ErrorMessage = "Uzupełnij nazwisko.")]
    [Display(Name = "Nazwisko")]
    public string Nazwisko { get; set; }
    [StringLength(100, ErrorMessage = "Maksymalnie 100 znaków")]
    [Required(AllowEmptyStrings = true, ErrorMessage = "Uzupełnij adres.")]
    [Display(Name = "Adres")]
    public string AdresZamieszkania { get; set; }
    [StringLength(20, ErrorMessage = "Maksymalnie 20 znaków")]
    [Required(AllowEmptyStrings = true, ErrorMessage = "Uzupełnij nr telefonu lub wpisz
brak")]
    [Display(Name = "Nr telefonu")]
    public string NrTelefonu { get; set; }
    [StringLength(20, ErrorMessage = "Maksymalnie 20 znaków")]
    [Required(AllowEmptyStrings = true, ErrorMessage = "Uzupełnij login")]
    [Display(Name = "Login")]
    public string Login { get; set; }
    [Required]
    [WalidacjaDaty]
    [Display(Name = "data urodzenia")]
    public String DataUrodzenia { get; set; }
    [Display(Name = "Aktywny")]
```

```
public bool Aktywny { get; set; }  
[Required]  
[ValidatePasswordLength(ErrorMessage = "Hasło musi być dłuższe niż 6 znaków")]  
[DataType(DataType.Password)]  
[Display(Name = "Hasło")]  
public string Haslo { get; set; }  
[DataType(DataType.Password)]  
[Display(Name = "Powtórz Hasło")]  
[Compare("Haslo", ErrorMessage = "Hasła niepoprawne")]  
public string ConfirmHaslo { get; set; }  
public List<RodziceDodawanieBool> ListaRodzicowUcznia { get; set; }  
public List<RodziceDodawanieBool> ListaSugerowanychRodzicowUcznia { get; set; }  
}  
public class RodziceDodawanieBool : Rodzice  
{  
    public bool czyDodac { get; set; }  
}
```

## Rozdział 5. Podsumowanie.

W ramach pracy inżynierskiej została zaprojektowana, zbudowana i przetestowana aplikacja webowa.

Prezentowana praca to opis systemu, który służy do wsparcia pracy nauczyciela. W powyższej dokumentacji starałem się ukazać zasadę działania oraz wygląd aplikacji tego rodzaju. Natomiast, poprzez przytoczenie fragmentów kodu przedstawiłem, w jaki sposób programować tego typu aplikacje webowe.

## **5.1 Kierunki dalszego rozwoju.**

Obecnie, na rynku informatycznym coraz większą popularnością cieszą się urządzenia mobilne – moim zdaniem, kierunek dalszego rozwoju, to właśnie przygotowanie oprogramowania działającego na urządzeniach mobilnych. W pierwszej kolejności, należałoby utworzyć odpowiednio rozbudowany WCF Service, – który można w łatwy sposób zbudować w Visual Studio 2010, a następnie np. przy pomocy wzorca projektowego MVVM, zbudować aplikację która mogła by w łatwy sposób być przeniesiona na różne platformy tj. np. na: Windows 8, Windows Phone 7, Android, IOS.

## **Spis tabel i rysunków.**

|  |           |
|--|-----------|
| <b>RYSUNEK 1 MODEL-VIEW-CONTROLLER.....</b>                        | <b>8</b>  |
| <b>RYSUNEK 2 2 PREZENTACJA LISTY RODZICÓW .....</b>                | <b>10</b> |
| <b>RYSUNEK 3 EDYCJA DANYCH RODZICA .....</b>                       | <b>11</b> |
| <b>RYSUNEK 4 WALIDACJA WPROWADZONEGO ROKU SZKOLNEGO .....</b>      | <b>11</b> |
| <b>RYSUNEK 5 LISTA PRZEDMIOTÓW W SZKOLE.....</b>                   | <b>12</b> |
| <b>RYSUNEK 6 EDYCJA DANYCH SZKOŁY.....</b>                         | <b>12</b> |
| <b>RYSUNEK 7 DODAWANIE NOWEGO NAUCZYCIELA.....</b>                 | <b>13</b> |
| <b>RYSUNEK 8 ZARZĄDZANIE UCZNIAMI W SZKOLE .....</b>               | <b>15</b> |
| <b>RYSUNEK 9 PRZYPISANIE NAUCZYCIELA DO DANEGO PRZEDMIOTU.....</b> | <b>16</b> |
| <b>RYSUNEK 10 OCENY POSTĘPÓW W NAUCE.....</b>                      | <b>18</b> |
| <b>RYSUNEK 11 DODAWANIE NOWEJ OCENY .....</b>                      | <b>19</b> |
| <b>RYSUNEK 12 ARCHIWUM. LISTA SPRAWDZIANÓW.....</b>                | <b>19</b> |

## **Literatura.**

- ❖ Gajda W., HTML, XHTML, iCSS, wyd. Helion, Gliwice 2007.
- ❖ Sharp J., Microsoft Visual C# 2010, wyd. APN Promise, Warszawa 2010.
- ❖ Matulewski J., Borycki P., Krause G., Visual Studio 2010 dla programistów C#, wyd. Helion, Gliwice 2011.
- ❖ Stefańczuk A., Sekrety języka C#. Tworzenie aplikacji w Visual Studio .Net 2005, Gliwice 2005.