**Zarządzanie projektem informatycznym**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kierunek studiów:** | *Informatyka* | **Rok studiów:** | *III* |
| **Numer grupy:** | *L5* | | |
| **Rok akademicki:** | *2014/2015* | **Semestr:** | *VI* |

**Sprawozdanie z wykonania projektu:**

**System zarządzania treścią dla szkół**

|  |  |  |
| --- | --- | --- |
| **Lp.** | **Nr indeksu** | **Imię i nazwisko** |
| 1. | 86316 | Radosław Szymański |
| 2. | 84135 | Damian Szymański |
| 6. | 84139 | Kamil Ślusarczyk |
| 4. | 84123 | Artur Stelmach |

|  |  |  |
| --- | --- | --- |
| **Termin zajęć:** | | **Prowadzący:** |
| dzień: | *Środa* | dr inż. Anna Zatwarnicka |
| godzina: | *12:50* |

## Główne założenia projektu

### Idea

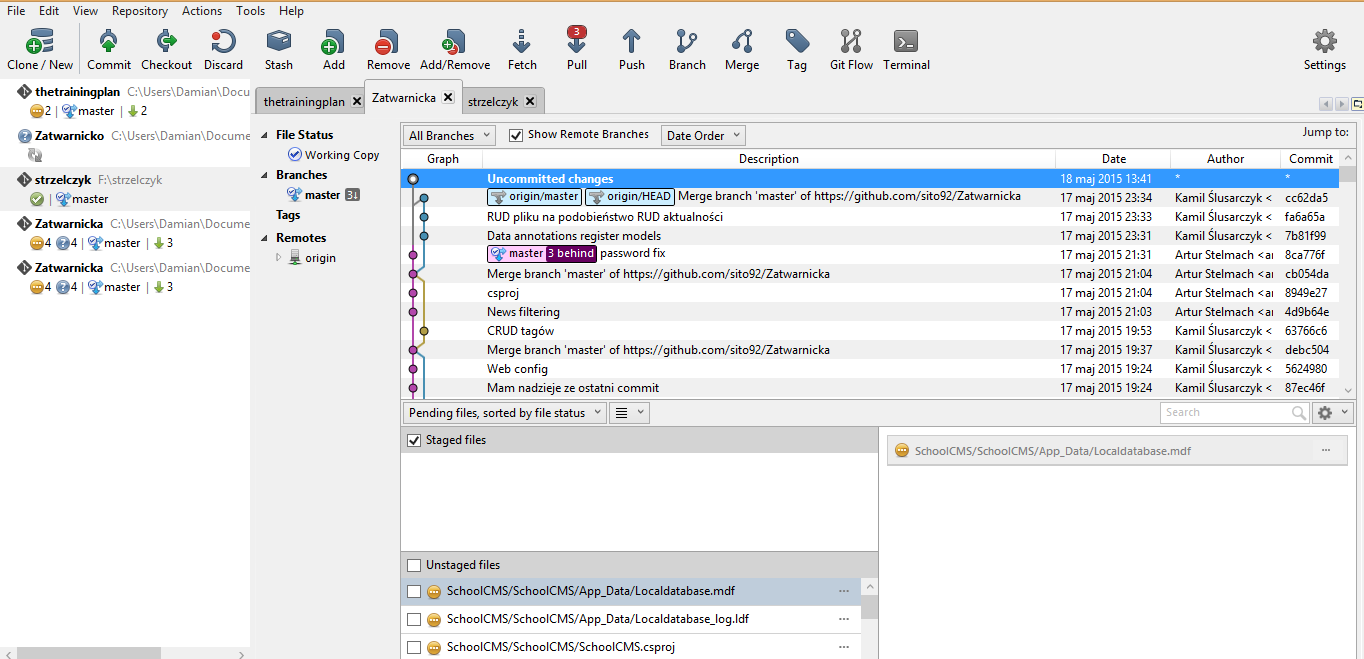
Nasza grupa postanowiła podjąć się wykonania systemu zarządzania treścią przeznaczonego dla szkolnych stron internetowych. Jednym z głównych argumentów, które przesądziły o tej decyzji było to, że niejednokrotnie, jak pokazuje rynek w ostatnich latach, potencjał komercyjny aplikacji zależy od prostoty systemu. Chcieliśmy więc wykonać system, który bez uszczuplania możliwych funkcjonalności będzie maksymalnie łatwy obsłudze, tak by grono potencjalnych odbiorców było jak największe.

### Podział ról i organizacja pracy

Team managerem z uwagi na cechy charakteru został Artur Stelmach. Jego zastępcą został mianowany Kamil Ślusarczyk.

Przy podziale ról i zadań kierowaliśmy się jedynie równomiernym podziałem pracy. Powodem tego jest podobny profil zainteresowań każdego z nas, który jest skierowany na programowanie w środowisku .Net.

Dla łatwiejszego zarządzania projektem skorzystaliśmy z rozproszonego systemu kontroli wersji Git, a do jego obsługi aplikacji SourceTree. Oprócz tego poszczególne zadania („taski”) rozpisywaliśmy w internetowej aplikacji Producteev, która systematyzowała naszą prace. Wybór tych narzędzi był podyktowany wcześniejszymi doświadczeniami pracy w zespołowych projektach informatycznych. Stworzyliśmy także diagram Gantta, który z uwagi na dużą rozdzielczość **został załączony w załączniku nr 1 do sprawozdania**.



Rys.1. SourceTree

### Wybór narzędzi

Z uwagi na wcześniejsze doświadczenia i zainteresowania aplikację postanowiliśmy napisać w środowisku .Net. Wybraliśmy platformę aplikacyjną do budowy aplikacji internetowych opartych na wzorcu Model-View-Controller (MVC) ASP.NET MVC. Jest to środowisko i model, który umożliwia szybką pracę, jednocześnie nie zamykający możliwości programiście.

Użyliśmy również Enity Framework, który jest narzędziem typu ORM (Object Relational Mapping), pozwalającym odwzorować relacyjną bazę danych za pomocą architektury obiektowej. Skorzystaliśmy również z wzorca Dependency Injection poprzez jego implementację - Ninject.

# Wymagania funkcjonalne

Poniżej opisujemy wymagania funkcjonalne, których wykonanie założyliśmy.

1. Dodawanie wzorców stron do menu
2. Logowanie i wylogowywanie administratora i redaktorów.
3. Przypomnienie hasła dla administratora i redaktorów.
4. Tworzenie kont redaktorów przez administratorów.
5. CRUD aktualności dla administratora.
6. CRUD aktualności dla redaktorów.
7. Możliwość budowania menu z podstronami

Jest to funkcjonalność, która ma umożliwić użytkownikom systemu budowanie wielopoziomowych, zagnieżdżonych struktur tworzonej w naszym systemie CMS strony internetowej.

1. Możliwość budowania galerii
2. Dobieranie szablonów graficznych
3. Archiwum aktualności
4. Filtrowanie treści (tagi)
5. Możliwość dołączania załączników do podstron
6. E-dziennik
7. Edytor planu zajęć
8. Możliwość dodawania materiałów dydaktycznych dla konkretnych klas i zajęć

Funkcjonalność ta w praktyce oznacza możliwość załączania plików w różnych formatach(np. multimedialnych) w aktualnościach.

# 3. Wymagania niefunkcjonalne

Poniżej wymieniamy wymagania niefunkcjonalne:

* Możliwość wyświetlania strony na najpopularniejszych przeglądarkach: Chrome, Mozzila Firefox, Opera i Internet Explorer
* Zapewnienie ochrony kont użytkowników serwisu poprzez autentykację za pomocą hasła
* Umożliwienie personelowi nie technicznemu tworzenie własnych wersji serwisu
* Możliwość przechowywania wielu różnych plików: pdf, jpg ,itp.

# 4. Szacowanie kosztów

Szacowanie kosztów składa się z następujących punktów:

• koszt sprzętu będącego częścią tworzonego systemu

• koszt zakupu narzędzi

• koszt wyjazdów i szkoleń

• nakład pracy

Pierwsze 3 punkty nie dotyczą naszego projektu. Koszt sprzętu, dlatego iż zarówno programiści jak i klienci będą wykorzystywać wcześniej zakupiony sprzęt. Koszt zakupu narzędzi również nas nie dotyczy, ponieważ środowisko programistyczne i całe potrzebne oprogramowanie jest już w naszym posiadaniu. Jedynym kosztem wyjazdu jest koszt wyjazdu na szkolenie klienta, jednak jest to wartość pomijalna(czas które poświęcimy na to dodamy podczas szacowanie nakładu pracy).

Nakład pracy został przez nas oszacowany poprzez analogię. Zdecydowaliśmy się na tę metodę szacowaniu czasu pracy, ponieważ jest to kolejny projekt, który wykonujemy w tym samym składzie(lub prawie tym samym składzie). Szacowany czas wyniósł: 68 dni. Szczegółową rozpiskę poszczególnych zadań zamieszczamy w diagramie Gantt’a, który umieściliśmy w osobnym załączniku.

Średnio każdego dnia będziemy pracować 1 godzinę nad projektem. Mnożąc to przez ilość osób w projekcie i ilość dni, otrzymujemy 272 roboczogodziny.

# 5. Diagram baz danych

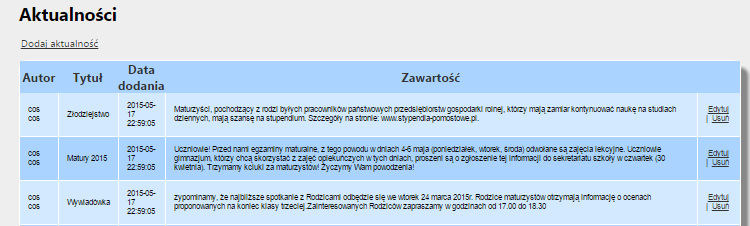
Zanim mogliśmy rozpocząć pracę stricte implementacyjną stworzyliśmy diagram relacyjnej baz danych. Skorzystaliśmy tutaj z program MS Office Visio. Z uwagi na dużą rozdzielczość **został on załączony w załączniku nr 2 do sprawozdania**.

# 6. Zaimplementowane funkcjonalności

Omawianie wszystkich funkcjonalności systemu zaczniemy od panelu administratora.

### 6.1. Aktualności

Administrator ma możliwość dodawania, edycji i usuwania dowolnych aktualności.



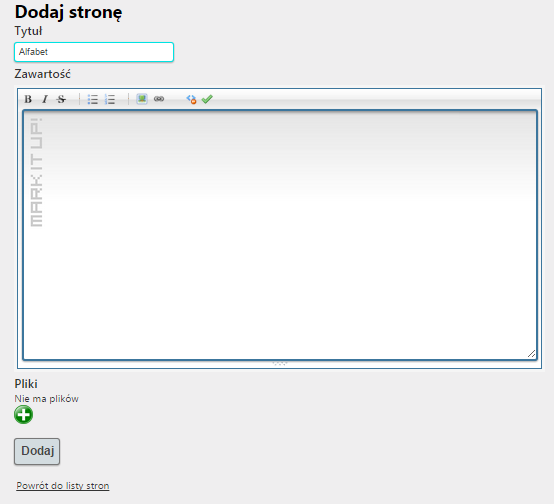
Rys.1. Zarządzanie aktualnościami



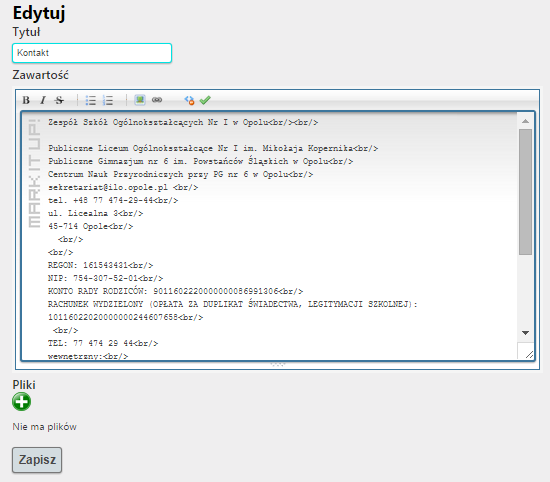
Rys.2. Dodawanie aktualności

### 6.2. Strony

Administrator może dodawać, edytować i usuwać strony. Strony można podpinać pod wybrane menu.



Rys.6. Dodawanie stron



Rys.4. Edytowanie stron

### 6.4. Zarządzanie redaktorami

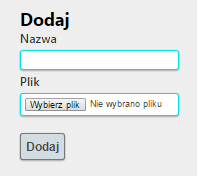
Admin ma możliwość tworzenia, edycji i usuwania redaktorów(funkcjonalności z punktu widzenia redaktora omówimy w dalszej części)



Rys.6. Dodawanie redaktorów

### 6.6. Zarządzanie plikami

W systemie istnieje możliwość wgrywania plików na serwer. Pliki można później wykorzystywać w różnych funkcjonalnościach (np. podczas tworzenia aktualności). Takie rozwiązanie jest zaimplementowane we wszystkich wiodących Systemach Zarządzania Treścią.



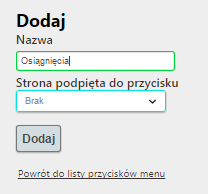
Rys.6. Dodawanie plików



Rys.7. Lista plików

### 6.6. Tworzenie menu

Jest to jedna z głównych funkcjonalności systemu. Pozwala ona w łatwy sposób budować rozgałęziona strukturę strony. Procedura tworzenia menu została pokazana na poniższych rysunkach, zawartych w punkcie 2.6.



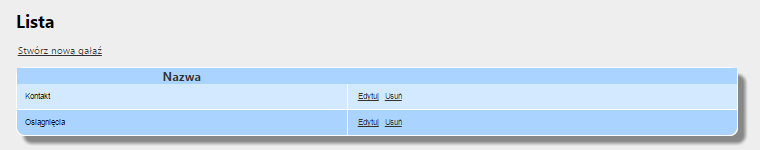
Rys.8. Dodawanie menu

Nowo utworzone menu pojawia się w prawym górnym rogu.



Rys.9. Nowo utworzone menu

Następnie możemy przejść do edycji utworzonego menu poprzez listę menu:



Rys.10. Lista menu

Następnie możemy dodać przycisk menu:



Rys.11. Dodanie przycisku menu

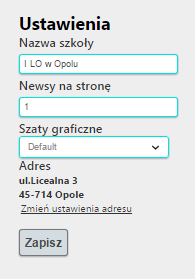
Stworzyliśmy rozgałęzioną strukturę. W prawym górnym menu mamy rozwijalne menu, które możemy dowolnie edytować, budując wielopoziomowe zagnieżdżenia. Dzięki temu nawet osoba bez informatycznego doświadczenia może stworzyć stronę, która w pełni wyczerpie jego oczekiwania.



Rys.12. Utworzenie rozgałęzionej struktury

### 6.7. Ustawienia

Tutaj możemy zmienić najbardziej podstawowe informacje o szkole, szatę graficzną oraz ilość wiadomości, pojawiających się na stronę.



Rys.16. Ustawienia

***Teraz omówimy funkcjonalności systemu z punktu widzenia redaktora.***

### 6.8. Aktualności



Rys.14. Dodawanie aktualności

Każdy redaktor ma możliwość usuwania i edycji **jedynie swoich aktualności.**



Rys.16. Lista aktualności

### 6.9. Pliki

Każdy redaktor ma możliwość dodawania, edycji i usuwania swoich plików na serwerze.



Rys.16. Dodawanie plików

Dodatkowymi funkcjonalnościami miało być wyszukiwanie po tagach i pobieranie plików z aktualności. Te funkcjonalności zostały zaprezentowane w punktach 6.10 i 6.11.

### 6.10. Tagi-wyszukiwanie aktualności

System posiada możliwość otagowywania aktualności. Jest to wykorzystane przy wyszukiwaniu aktualności.



Rys.17. Dodawanie tagów



Rys.18. Wyszukiwanie aktualności



Rys.19. Znalezione aktualności

### 6.11. Pobieranie plików z newsów

Dodaliśmy również możliwość pobierania plików z aktualności:



Rys.20. Pobieranie załączonych plików

### 6.12.Walidacja danych

W systemie zadbaliśmy o walidację danych. Każde błędne wypełnienie pól kończy się komunikatem ze strony systemu. Poniżej zaimplementowana walidacja na przykładzie procesu dodawania redaktora do systemu.



Rys.21. Walidacja danych

# 7. Testowanie i poprawki systemu

W trakcie procesu rozwoju aplikacji istotną fazą jest faza testów. W przypadku naszego programu testowanie było podzielone na dwie fazy – zewnętrzne oraz wewnętrzne testy.

Wewnętrzne testy były przeprowadzane metodą testów użytkownika – podczas rozwoju aplikacji programiści na bieżąco sprawdzali jakość i bezawaryjność swoich rozwiązań wcielając się chwilowo w potencjalnego użytkownika.

Zewnętrzne testy zostały przeprowadzone przez osoby z zewnątrz projekty, które starały się wyłapać jak najwięcej niedociągnięć i błędów m.in. za pomocą :

- SQL Injection,

- testowania walidacji wprowadzanych danych przez aplikacje,

- testowania stopnia realizacji wymagań funkcjonalnych przez autorów,

- testowania procesów autoryzacji w aplikacji.

Poniżej zamieszczono tabelę, która przedstawia błędy znalezione podczas fazy rozwoju oraz testów.

|  |  |  |  |
| --- | --- | --- | --- |
| błąd | zewnętrzny/wewnętrzny | Rozwiązany(tak/nie) | komentarz |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |