Projekt zespołowy - StudNet

Patrycjusz Naklicki, Mateusz Nowak, Bartłomiej Pabisiak, Dawid Paleń, Damian Stachelczyk, gr. 322B

# Dokumentacja techniczna

StudNet – system komunikacji studenckiej

Spis treści

[1. Specyfikacja wymagań 4](#_Toc400017921)

[1.1. Definicja produktu 4](#_Toc400017922)

[1.2. Technologia 4](#_Toc400017923)

[1.3. Wymagania funkcjonalne 4](#_Toc400017924)

[2. Projekt techniczny 5](#_Toc400017925)

[2.1. Projekt interfejsu użytkownika 5](#_Toc400017926)

[Tabela 1. Główne okno aplikacji 6](#_Toc400017927)

[Rysunek 2. Kolejny widok 6](#_Toc400017928)

[Tabela 2. Pasek menu – menu „Plik” 6](#_Toc400017929)

[Rysunek 3. Pasek menu – menu „Import/eksport” 7](#_Toc400017930)

[Tabela 3. Pasek menu – menu „Import/eksport” 7](#_Toc400017931)

[Rysunek 6. Okno „Import” 8](#_Toc400017932)

[Tabela 6. Okno „Import” 8](#_Toc400017933)

[Rysunek 8. Okno "Szukaj wg albumu" 8](#_Toc400017934)

[Tabela 8. Okno "Szukaj wg albumu" 8](#_Toc400017935)

[Rysunek 10. Okno "O aplikacji...” 9](#_Toc400017936)

[Tabela 10. Okno "O aplikacji...” 9](#_Toc400017937)

[Rysunek 13. Komunikat o błędzie 9](#_Toc400017938)

[Tabela 13. Komunikat o błędzie 9](#_Toc400017939)

[2.2. Opisy funkcjonalności 10](#_Toc400017940)

[Funkcjonalność otwierania plików do edycji 10](#_Toc400017941)

[Funkcjonalność wyświetlania tagów 10](#_Toc400017942)

[Funkcjonalność edycji tagów załadowanych plików 10](#_Toc400017943)

[Funkcjonalność edycji okładki załadowanych plików 10](#_Toc400017944)

[Funkcjonalność importowania tagów z pliku tekstowego 11](#_Toc400017945)

[2.3 Diagramy UML 11](#_Toc400017946)

[2.3.1. Diagram przypadków użycia 11](#_Toc400017947)

[2.3.2. Diagramy aktywności 12](#_Toc400017948)

[2.3.4. Diagramy sekwencji 13](#_Toc400017949)

[2.3.5. Diagram obiektów 14](#_Toc400017950)

[2.3.6. Diagram klas 14](#_Toc400017951)

[3. Scenariusze testowe 15](#_Toc400017952)

[4. Plan testów jednostkowych 16](#_Toc400017953)

# 1. Specyfikacja wymagań

## 1.1. Definicja produktu

Aplikacja webowa StudNet służy do komunikacji studentów z tego samego roku akademickiego i wydziału. Zawierać ona będzie forum dla studentów z całego roku, a także fora dla podgrup, kalendarz z datami popraw i innymi ważnymi dla studentów terminami. Ma ona na celu ułatwienie starostom przekazywanie informacji studentom oraz ułatwienie studentom komunikowanie się i dzielenie się materiałami.

## 1.2. Technologia

Aplikacja została stworzona w technologii .NET w wersji 4.5.2. Do przechowywania danych wykorzystana została baza danych SQL umieszczona na platformie Microsoft Azure.

## 1.3. Wymagania funkcjonalne

Jako użytkownik możemy zarejestrować nowe konto.

Jako użytkownik możemy zalogować się do systemu.

Jako użytkownik możemy wylogować się z systemu.

# 2. Projekt techniczny

## 2.1. Projekt interfejsu użytkownika

### Rysunek 1. Strona główna aplikacji

### Tabela 1. Strona główna aplikacji

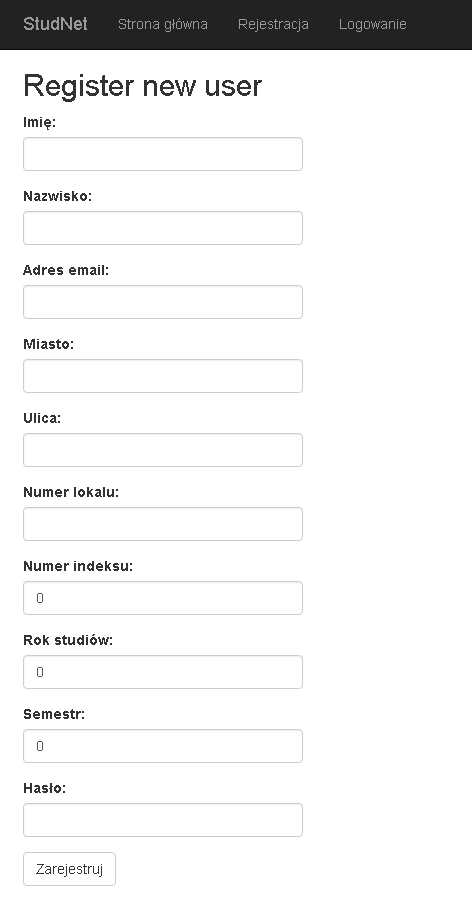
|  |  |  |
| --- | --- | --- |
| ID | Nazwa | Opis |
| 1 | Link  "StudNet" | Przenosi użytkownika na stronę główną. |
| 2 | Link "Strona Główna" | Przenosi użytkownika na stronę główną. |
| 3 | Link "Rejestracja" | Przenosi użytkownika na stronę rejestracji nowego konta. |
| 4 | Link "Logowanie" | Przenosi użytkownika na stronę logowania. |

### Rysunek 2. Strona główna aplikacji dla zalogowanego użytkownika

### Tabela 2. Strona główna aplikacji dla zalogowanego użytkownika

|  |  |  |
| --- | --- | --- |
| ID | Nazwa | Opis |
| 5 | Przycisk  "Wyloguj" | Wylogowuje użytkownika z systemu. |

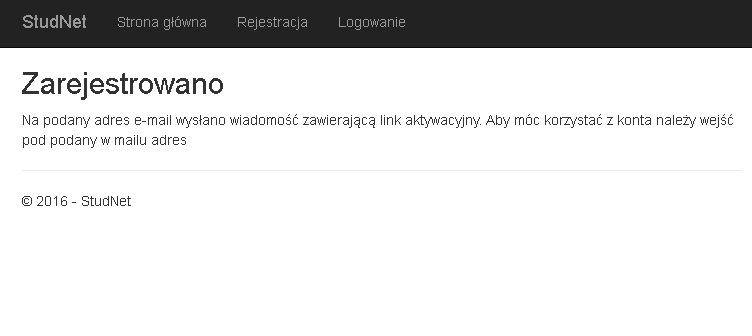
### Rysunek 3. Strona rejestracji nowego konta



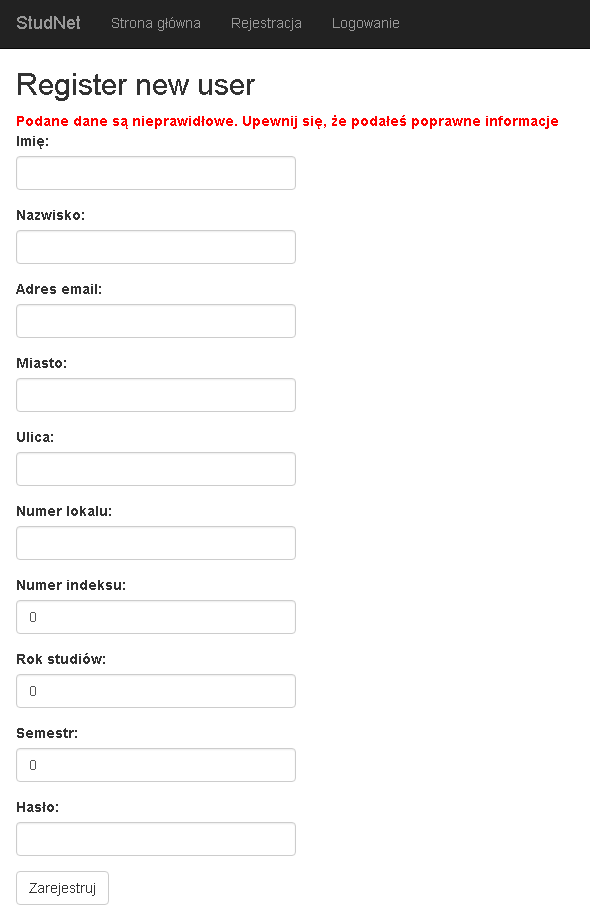
### Tabela 3. Strona rejestracji nowego konta.

|  |  |  |
| --- | --- | --- |
| ID | Nazwa | Opis |
| 6 | Pole "Imię" | Pole do wpisania imienia użytkownika. |
| 7 | Pole "Nazwisko" | Pole do wpisania nazwiska użytkownika. |
| 8 | Pole "Adres email" | Pole do wpisania adresu email użytkownika. |
| 9 | Pole "Miasto" | Pole do wpisania miasta użytkownika. |
| 10 | Pole "Ulica" | Pole do wpisania adresu użytkownika. |
| 11 | Pole "Numer lokalu" | Pole do wpisania numeru lokalu użytkownika. |
| 12 | Pole "Numer indeksu" | Pole do wpisania numeru indeksu użytkownika. |
| 13 | Pole "Rok studiów" | Pole do wpisania roku studiów użytkownika. |
| 14 | Pole "Semestr" | Pole do wpisania semestru użytkownika. |
| 15 | Pole "Hasło" | Pole do wpisania hasła użytkownika. |
| 16 | Przycisk "Zarejestruj" | Powoduje zarejestrowanie nowego konta użytkownika. |

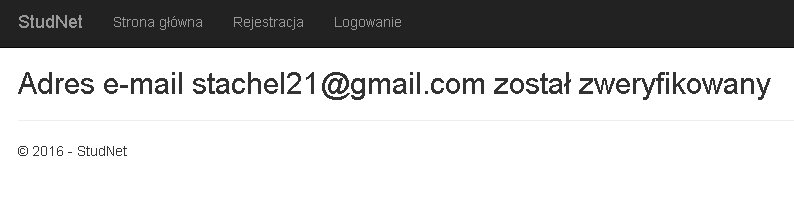
### Rysunek 4. Strona rejestracji nowego konta - poprawna rejestracja



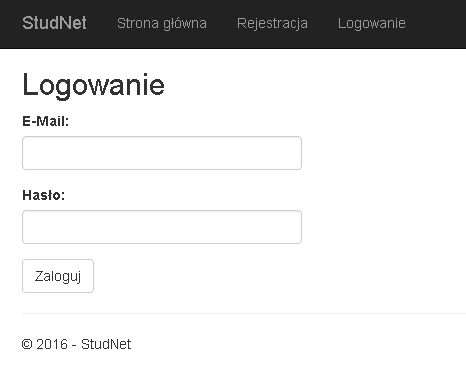
### Rysunek 5. Strona rejestracji nowego konta - niepoprawna rejestracja



### Rysunek 6. Strona potwierdzania adresu email



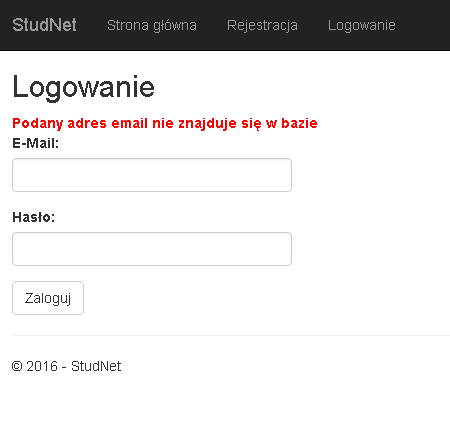
### Rysunek 7. Strona logowania



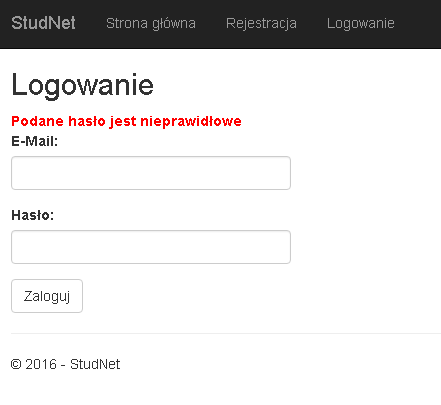
### Tabela 4. Strona logowania

|  |  |  |
| --- | --- | --- |
| ID | Nazwa | Opis |
| 17 | Pole "E-Mail" | Pole do wpisania adresu email użytkownika. |
| 18 | Pole "Hasło" | Pole do wpisania hasła użytkownika. |
| 19 | Przycisk "Zaloguj" | Zalogowuje użytkownika do systemu. |

### Rysunek 8. Strona logowania – niepoprawny adres email



### Rysunek 9. Strona logowania – niepoprawne hasło



## 2.2. Opisy funkcjonalności

### Funkcjonalność rejestracji nowego konta

#### Scenariusz główny

1. Użytkownik klika link "Rejestracja" (Rys. 1).
2. Użytkownik prawidłowo wypełnia pola "Imię", "Nazwisko", "Adres email", "Miasto", "Ulica", "Numer lokalu", "Numer indeksu", "Rok studiów", "Semestr", "Hasło" (Rys. 3).
3. Użytkownik klika przycisk "Zarejestruj" (Rys. 3).
4. Użytkownikowi wyświetla się komunikat o poprawnej rejestracji (Rys. 4).
5. Użytkownik potwierdza adres email poprzez kliknięcie w link aktywacyjny.
6. Użytkownikowi wyświetla się komunikat o poprawnym potwierdzeniu adresu email (Rys. 6).

#### Scenariusz alternatywny – użytkownik podaje niepoprawne dane

2a. Użytkownik niepoprawnie wypełnia lub pozostawia puste jedno lub więcej z pól "Imię", "Nazwisko", "Adres email", "Miasto", "Ulica", "Numer lokalu", "Numer indeksu", "Rok studiów", "Semestr", "Hasło" (Rys. 1).  
4a. Użytkownikowi wyświetla się komunikat o niepoprawnej rejestracji (Rys. 5).

### Funkcjonalność logowania

#### Scenariusz główny

1. Użytkownik klika link "Logowanie" (Rys. 1).
2. Użytkownik prawidłowo wypełnia pola "E-Mail" oraz "Hasło" (Rys. 7).
3. Użytkownik klika przycisk "Zaloguj" (Rys. 7).
4. Użytkownik zostaje zalogowany i przekierowany na stronę główną (Rys. 2).

#### Scenariusz alternatywny – użytkownik podaje niepoprawny adres email

2a. Użytkownik niepoprawnie wypełnia lub pozostawia puste pole "E-Mail" (Rys. 7).

4a. Użytkownikowi wyświetla się komunikat o niepoprawnym adresie email (Rys. 8).

#### Scenariusz alternatywny – użytkownik podaje niepoprawne hasło

2b. Użytkownik niepoprawnie wypełnia lub pozostawia puste pole "Hasło" (Rys. 7).

4b. Użytkownikowi wyświetla się komunikat o niepoprawnym haśle (Rys. 9).

### Funkcjonalność wylogowania

#### Scenariusz główny

1. Użytkownik klika przycisk "Wyloguj" (Rys. 2).
2. Użytkownik zostaje wylogowany i przekierowany na stronę główną (Rys. 1).

## 2.3 Diagramy UML

### 2.3.1. Diagram przypadków użycia

### Diagramy aktywności

### 2.3.4. Diagramy sekwencji

### 2.3.5. Diagram obiektów

### 2.3.6. Diagram klas

# Scenariusze testowe

# 4. Plan testów jednostkowych

Testy wykonywane przy użyciu „Microsoft Unit Testing Framework for C++”