

Informatyka, Wydział Elektroniki Politechniki Wrocławskiej

Edytor map 3D

Projekt realizowany na zajęcia 'inżynierii e-systemów w technologii JAVA'

***Autor:** Sebastian Wilgosz*

Prowadzący:
dr inż. Tomasz Walkowiak

3 czerwca 2015

Spis treści

1	Cel i zakres projektu	2
1.1	Cel projektu	2
1.2	Zakres projektu	2
2	Analiza i Specyfikacja	3
2.1	Opis słowny zadania	3
2.2	Specyfikacja wymagań funkcjonalnych	3
2.3	Specyfikacja wymagań niefunkcjonalnych	4
3	Instrukcja użytkownika	5
3.1	Wersja Produkcyjna	5
3.1.1	Logowanie i rejestracja	5
3.1.2	Kontrola dostępu	6
3.1.3	Wysyłanie emaili	7
3.1.4	Dodawanie mapy z pliku	7
3.2	Wersja lokalna	9
3.2.1	Wysyłanie emaili	9
4	Podsumowanie	10
4.1	Wykonane założenia	10
4.2	Niewykonane założenia	11

Rozdział 1

Cel i zakres projektu

1.1 Cel projektu

Celem projektu jest wykonanie aplikacji serwerowej w technologii Ruby on Rails, pozwalającej na zarządzanie mapami 3d budowanymi na podstawie plików CSV.

Sam sposób wdrożenia programu jak i jego zadania wraz z implementacją przebiegały wedle przedstawionego pomysłu, który został zaakceptowany przez prowadzącego kurs. Projekt miał na celu zapoznać studentów specjalizacji *Inżynieria Internetowa* z procesem budowania oprogramowania obecnie stosowanego w świecie aplikacji serwerowych. Projekt dodatkowo dzięki wybranemu tematowi pomógł zgłębić technologie strony klienckiej takie jak **JavaScript**.

1.2 Zakres projektu

Niniejszy projekt zawiera opis użytkowy i techniczny aplikacji oraz wykorzystanych narzędzi niezbędnych do stworzenia aplikacji w środowisku **Ruby on Rails**.

Proces tworzenia oprogramowania został podzielony na następujące etapy:

1. Analiza i Specyfikacja
2. Projektowanie
3. Implementacja
4. Testy
5. Ocena i Optymalizacja

Rozdział 2

Analiza i Specyfikacja

2.1 Opis słowny zadania

Celem pracy projektowej jest stworzenie aplikacji zdolnej do tworzenia i zarządzania mapami 3d terenu. Niniejsza aplikacja serwerowa ma za zadanie wspomagać tworzenia ciekawych wizualizacji terenu dla osób zajmujących się hobbystycznie jak i zawodowo kartografią. Docelowo tworzone przez użytkowników prace można będzie łączyć, zlecać wykonanie na nich danych obliczeń w celu wykorzystania ich w danym problemie. Nasz serwis z aplikacją nazwałem *"3dMap"*, ze względu na krótką i intuicyjną nazwę. Aplikację w przyszłości będzie można rozwinąć w celu szerszych zastosowań.

2.2 Specyfikacja wymagań funkcjonalnych

1. **Wysyłanie emaili kontaktowych** - użytkownicy powinni mieć możliwość wysyłania maili do administratorów w razie problemów.
2. **Prezentacja bazy punktów w formie mapy** - Jest to główna funkcja tej aplikacji, ma ona za zadanie z danego zbioru danych generować podgląd danego terenu w formie obrotowej mapki w przestrzeni trójwymiarowej.
3. **Oddzielne sesje dla każdego użytkownika** - Ponieważ aplikacja będzie zawierała w sobie proces tworzenia jakiegoś elementu terenu niezbędne będzie wyodrębnienie pojedynczych działań aplikacji w formie sesji użytkowników, którzy swoje gotowe mapy będą mogli dzięki powiązaniu z kontem przechowywać w zdalnej przestrzeni dyskowej
4. **Wczytywanie map z pliku** - Poza wczytywaniem map z bazy serwera możliwe będzie wczytywanie wcześniej zapisanych map z pliku. Funkcjonalność taka będzie przydatna gdy użytkownik po wykasowaniu mapy bądź usunięciu konta, chciałby odtworzyć swoje prace.
5. **Łączenie kilku prac w jedną** - Serwer będzie mógł według kilku definiowanych zasad łączyć kilka zasobów w jedną wspólną mapę, taka funkcjonalność będzie przydatna przy tworzeniu pracy opartej na wkładzie kilku użytkowników
6. **Generacja map** - Generacja map na podstawie już istniejącej z podanymi zasadami zmiany oraz tworzenie totalnie losowej mapy
7. **Zapisywanie prac na serwerze** - Każda stworzona mapa będzie dostępna niezależnie od miejsca i sprzętu użytkownika poprzez interfejs webowy dostępny w przeglądarce internetowej

8. Eksport gotowych prac do określonych formatówKażdą z gotowych prac będzie można wyeksportować do formatu możliwego do użytku w celu prezentacji/wizualizacji z założenia są to formaty ***.pdf *.jpeg**.

2.3 Specyfikacja wymagań niefunkcjonalnych

1. **Przejrzysty interfejs webowy** - Wymaganiem jest by aplikacja była nie przeładowana dodatkami i prosta w obsłudze dla osób nie posiadających zdolności programistycznych
2. **Szybkość i oszczędność łącza** - Aplikacja powinna większość pracy wykonywać bez potrzeby generowania zbędnego ruchu sieciowego co pomoże zaoszczędzić zasoby klienta
3. **Multiplatformowość** - Aplikacja powinna generować taki sam rezultat niezależnie od platformy sprzętowej klienta
4. **Baza danych MySQL** - Zasób danych powinien być przechowywany na łatwej w obsłudze i bezpłatnej dystrybucji bazy danych, takiej jak np. **MySQL**
5. **Dane o twórcach jak i projekcie** - Aplikacja będzie zawierała informacje o okolicznościach w jakich powstała i dla jakich celów
6. **Model MVC** - Aplikacja będzie opierać się na modelu MVC
7. **Dokumentacja** - Przebieg procesu powstawania jak i opis funkcjonalności i sposób wykorzystania poszczególnych funkcji będzie zawarty w dokumentacji projektu

Rozdział 3

Instrukcja użytkownika

3.1 Wersja Produkcyjna

Wersja produkcyjna aplikacji jest uruchomiona na serwerze heroku, pod adresem `http://terrain-editor.herokuapp.com`

Pierwsze uruchomienie wymaga uruchomienia serwera, który przechodzi w stan uśpienia po godzinie braku aktywności ze strony użytkowników w celu oszczędności zasobów. Z tego powodu po wpisaniu adresu w przeglądarce internetowej po raz pierwszy, należy poczekać nieco dłużej niż w przypadku kolejnych zapytań do serwera.

3.1.1 Logowanie i rejestracja

Aby mieć dostęp do wszystkich funkcji aplikacji, należy się zalogować, lub zarejestrować nowego użytkownika.

Ze względu na przeznaczenie projektu do celów edukacyjnych, postanowiłem nie walidować adresów email pod kątem prawdziwości, dlatego dopuszczalne do rejestracji są fałszywe adresy, pod warunkiem że zachowany zostanie poprawny format.

W wersji pokazowej zarejestrowany został użytkownik: **email:** test.user@example.com **hasło:** password



Rysunek 3.1: Strona główna aplikacji

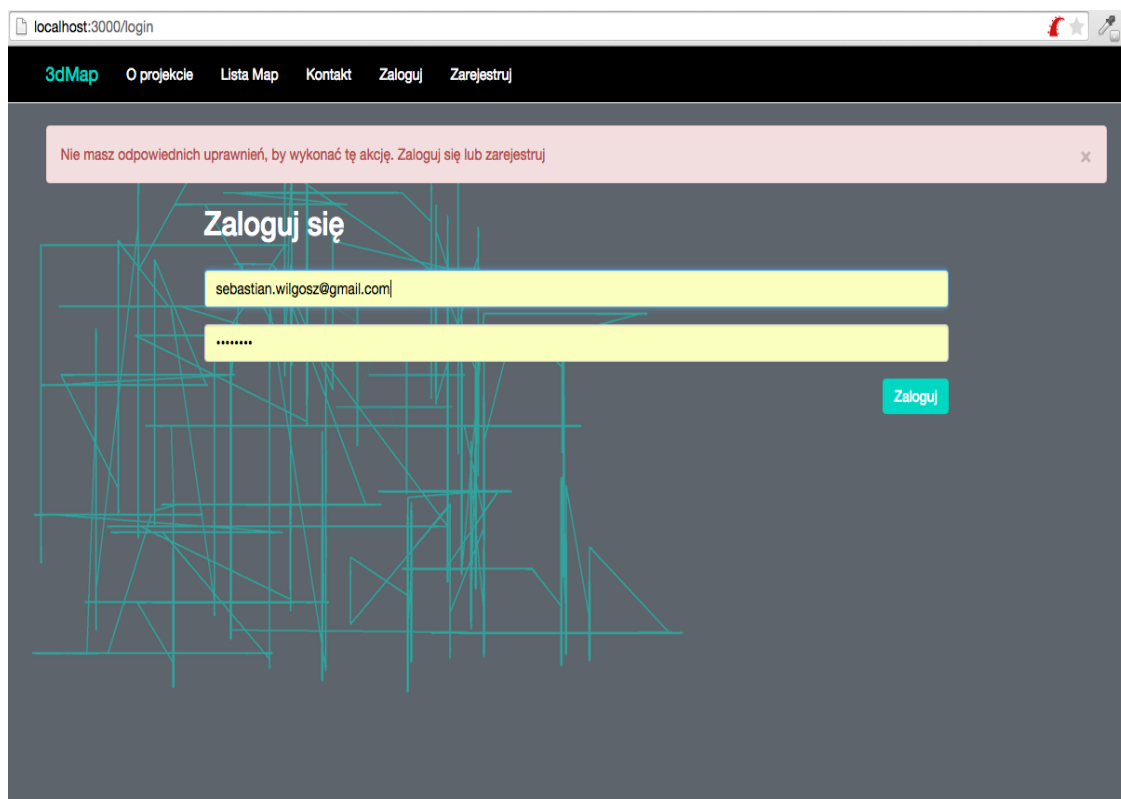
3.1.2 Kontrola dostępu

Wykorzystanie sesji logowania pozwoliło na ograniczenie praw dla użytkowników niezalogowanych, jak i dla zalogowanych.

- Użytkownicy niezalogowani mogą:
 - Przeglądać mapy
 - Logować się, i rejestrować
 - Wysyłać emaile kontaktowe
- Użytkownicy zalogowani mogą:
 - Wykonywać wszystkie akcje niezalogowanych użytkowników z wyjątkiem rejestracji i logowania.
 - Dodawać mapy
 - Edytować swoje mapy

W przypadku próby wykonania akcji zabronionej, użytkownik niezalogowany jest przekierowywany do ekranu logowania, zaś zalogowany użytkownik do strony głównej.

Na górze ekranu pojawi się wiadomość o braku uprawnień do wykonania danej akcji.



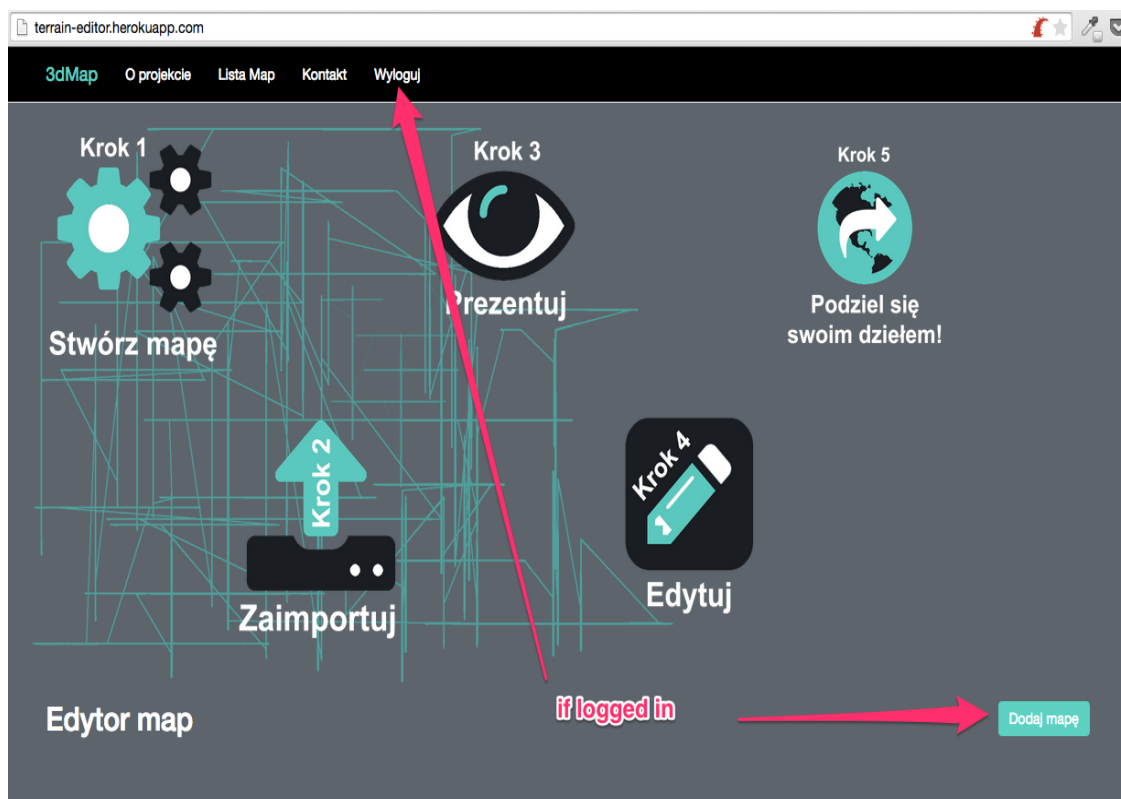
Rysunek 3.2: Próba wykonania akcji zabronionej

3.1.3 Wysyłanie emaili

Aplikacja umożliwia wysłanie wiadomości kontaktowej do administratorów systemu. Z powodu udostępnienia aplikacji online, postanowiłem wyłączyć funkcję wysyłania maili na rzeczywistą skrzynkę odbiorczą. Zamiast tego zintegrowałem projekt z aplikacją *Mailtrap*, która przechwytuje wszystkie emaile wysyłane z konta pocztowego. W celu rekonfiguracji konta, zobacz sekcję: 3.2.1.

3.1.4 Dodawanie mapy z pliku

Dodawanie nowych map jest dostępne wyłącznie dla zalogowanych użytkowników. Jeżeli chcesz dodać mapę, będąc zalogowanym kliknij w przycisk na stronie głównej, liście map albo stronie pojedynczej mapy.



Rysunek 3.3: Dodawanie mapy, krok 1

Po kliknięciu w przycisk pojawi się formularz dodawania nowej mapy. Schemat należy wczytać z odpowiednio sformatowanego pliku tekstowego.

Plik powinien mieć następujący format:

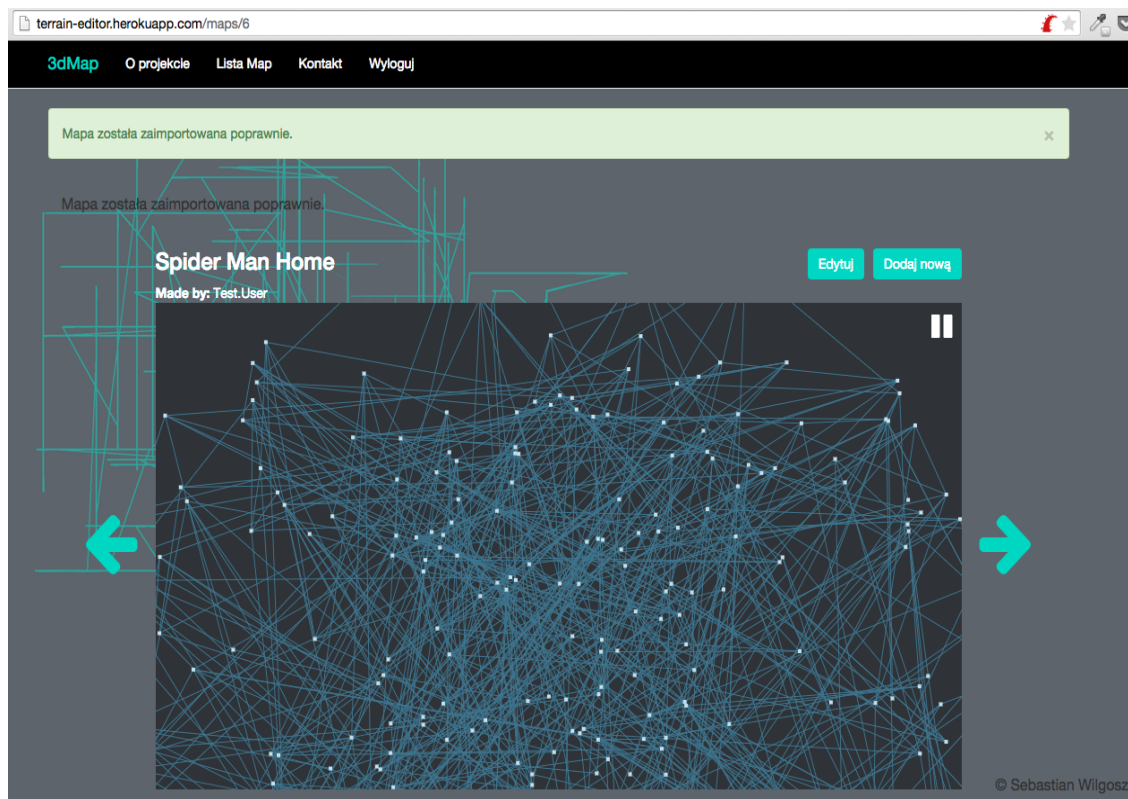
Source X	Source Y	Source Z	Target X	Target Y	Target Z
----------	----------	----------	----------	----------	----------

Tablica 3.1: Format pliku z danymi mapy do importu.

Lub powinny być plikiem tekstowym, w którym każda linia odpowiada kolejnemu wierszowi, zaś komórki są oddzielone średnikiem (;).

W celach prezentacyjnych do repozytorium zostały dołączone przykładowe pliki z danymi do zainportowania w katalogu `app/assets/csv/`.

Jeżeli import mapy przebiegł poprawnie, użytkownik zostanie przekierowany do widoku właśnie dodanej mapy.



Rysunek 3.4: Widok mapy

Ten ekran wyświetla mapę jako obracającą się wizualizację 3D. Użytkownik może zatrzymać i wznowić animację, klikając w ikonę w prawym górnym rogu animacji.

Ponadto użytkownik może w prosty sposób przeglądać inne mapy zapisane na serwerze, klikając w strzałki po prawej i lewej stronie.

3.2 Wersja lokalna

ubuntu/reverse

3.2.1 Wysyłanie emaili

Rozdział 4

Podsumowanie

Projekt był początkowo z przeznaczony dla dwóch osób. Niestety, dwa tygodnie przed oddaniem drugiego członka zespołu zrezygnował z kontynuowania nauki, nie dołączając żadnego wkładu ze swojej strony.

Z tego powodu nie udało mi się wykonać wszystkich funkcji, jakie zostały założone w specyfikacji początkowej.

4.1 Wykonane założenia

1. **Wysyłanie emaili kontaktowych** - użytkownicy powinni mieć możliwość wysyłania maili do administratorów w razie problemów.
2. **Prezentacja bazy punktów w formie mapy** - Jest to główna funkcja tej aplikacji, ma ona za zadanie z zadanego zbioru danych generować podgląd danego terenu w formie obrotowej mapki w przestrzeni trójwymiarowej.
3. **Oddzielne sesje dla każdego użytkownika** - Ponieważ aplikacja będzie zawierała w sobie proces tworzenia jakiegoś elementu terenu niezbędne będzie wyodrębnienie pojedynczych działań aplikacji w formie sesji użytkowników, którzy swoje gotowe mapy będą mogli dzięki powiązaniu z kontem przechowywać w zdalnej przestrzeni dyskowej
4. **Wczytywanie map z pliku** - Poza wczytywaniem map z bazy serwera możliwe będzie wczytywanie wcześniej zapisanych map z pliku. Funkcjonalność taka będzie przydatna gdy użytkownik po wykasowaniu mapy bądź usunięciu konta, chciałby odtworzyć swoje prace.
5. **Generacja map** - Generacja map na podstawie już istniejącej z podanymi zasadami zmiany oraz tworzenie totalnie losowej mapy
6. **Zapisywanie prac na serwerze** - Każda stworzona mapa będzie dostępna niezależnie od miejsca i sprzętu użytkownika poprzez interfejs webowy dostępny w przeglądarce internetowej
7. **Przejrzysty interfejs webowy** - Wymaganiem jest by aplikacja była nie przeładowana dodatkami i prosta w obsłudze dla osób nie posiadających zdolności programistycznych
8. **Szybkość i oszczędność łącza** - Aplikacja powinna większość pracy wykonywać bez potrzeby generowania zbędnego ruchu sieciowego co pomoże zaoszczędzić zasoby klienta
9. **Multiplatformowość** - Aplikacja powinna generować taki sam rezultat niezależnie od platformy sprzętowej klienta

10. **Baza danych MySQL** - Zasób danych powinien być przechowywany na łatwej w obsłudze i bezpłatnej dystrybucji bazy danych, takiej jak np. **MySQL**
11. **Dane o twórcach jak i projekcie** - Aplikacja będzie zawierała informacje o okolicznościach w jakich powstała i dla jakich celów
12. **Model MVC** - Aplikacja opiera się na modelu MVC
13. **Dokumentacja** - Przebieg procesu powstawania jak i opis funkcjonalności i sposób wykorzystania poszczególnych funkcji będzie zawarty w dokumentacji projektu

4.2 Niewykonane założenia

1. Eksport gotowych prac do określonych formatów Każdą z gotowych prac będzie można wyeksportować do formatu możliwego do użytku w celu prezentacji/wizualizacji z założenia są to formaty ***.pdf *.jpeg**.
2. **Łączenie kilku prac w jedną** - Serwer będzie mógł według kilku definiowanych zasad łączyć kilka zasobów w jedną wspólną mapę, taka funkcjonalność będzie przydatna przy tworzeniu pracy opartej na wkładzie kilku użytkowników

Bibliografia

- [1] <http://blog.wilgosz.pl/pl/posts/ruby-on-rails/ruby-on-rails-instalacja-ubuntu-12-04>, Instalacja Ruby On Rails na Ubuntu, Sebastian Wilgosz 2013