



AGH

**AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA
W KRAKOWIE**

**WYDZIAŁ INFORMATYKI, ELEKTRONIKI
I TELEINFORMATYKI**

KATEDRA INFORMATYKI

Dokumentacja Specyfikacyjna
Kryptonim: **OtoKonfa**

*Aplikacja umożliwiająca wyszukiwanie i porównywanie
ośrodków wynajmujących sale na spotkania o charakterze
konferencyjnym*

Autor:

S. Chmiela, M. Górka, M. Imiełowski, P. Stawicki, P. Wolak

Kierunek studiów:

Informatyka

Opiekun pracy:

mgr inż. Witold Rakoczy

Kraków, 2016

Spis treści

1. Stos technologiczny	6
1.1. Wybrana technologia	6
1.2. Zależności	6
2. Opis bazy danych	8
2.1. Diagram ERD	8
2.2. Krótki opis	8
3. Specyfikacja modułów	9
3.1. Użytkownicy	9
3.2. Deweloperzy	10
A. Diagramy ERD	13

1. Stos technologiczny

1.1. Wybrana technologia

Cały projekt został wykonany przy użyciu Ruby on Rails. Zdecydowaliśmy się na tę technologię, gdyż pozwala ona na szybki rozwój aplikacji, a co za tym idzie łatwe pisanie prototypów. Dodatkowo sama technologia oferuje dużą liczbę gemów (zależności Railsów), dzięki którym można znacznie uprościć pisanie kodu, gdyż znaczna część funkcjonalności jest już zaimplementowana w którymś z nich.

1.2. Zależności

W projekcie wykorzystujemy następujące zależności zewnętrzne:

1. **jQuery**

Mała i bogata w funkcje biblioteka JavaScript sprawiająca, że rzeczy takie jak np. obsługa zdarzeń, animacje czy Ajax stają się znacznie prostsze.

2. **Materialize**

Front-endowy framework znacznie upraszczający i przyspieszający pisanie widoków aplikacji webowych.

3. **Google Maps**

Skrypt umożliwiający wyświetlanie na stronie mapy (lokalizacji).

4. **Bootstrap Tags Input**

Dodatek do jQuery umożliwiający zarządzanie i wyświetlanie tagów w stylu Twittera.

5. **NoUISlider**

Skrypt napisany w języku JavaScript, umożliwiający wyświetlanie i obsługę zaawansowanych suwaków na stronie.

6. **Material Icons**

Czcionka od Google

7. **Typeahead**

Skrypt umożliwiający wyświetlanie podpowiedzi

8. Devise

Biblioteka znacznie znacznie ułatwiająca zaimplementowanie systemu użytkowników oraz ich autentykację

9. MySQL2

Biblioteka umożliwiająca użycie MySQL jako bazy danych Active Record

10. Turbolinks

Przyspiesza przechodzenie pomiędzy linkami w aplikacji

11. jBuilder

Zależność ułatwiająca budowanie JSON'ów w aplikacji

12. Sass-rails gem

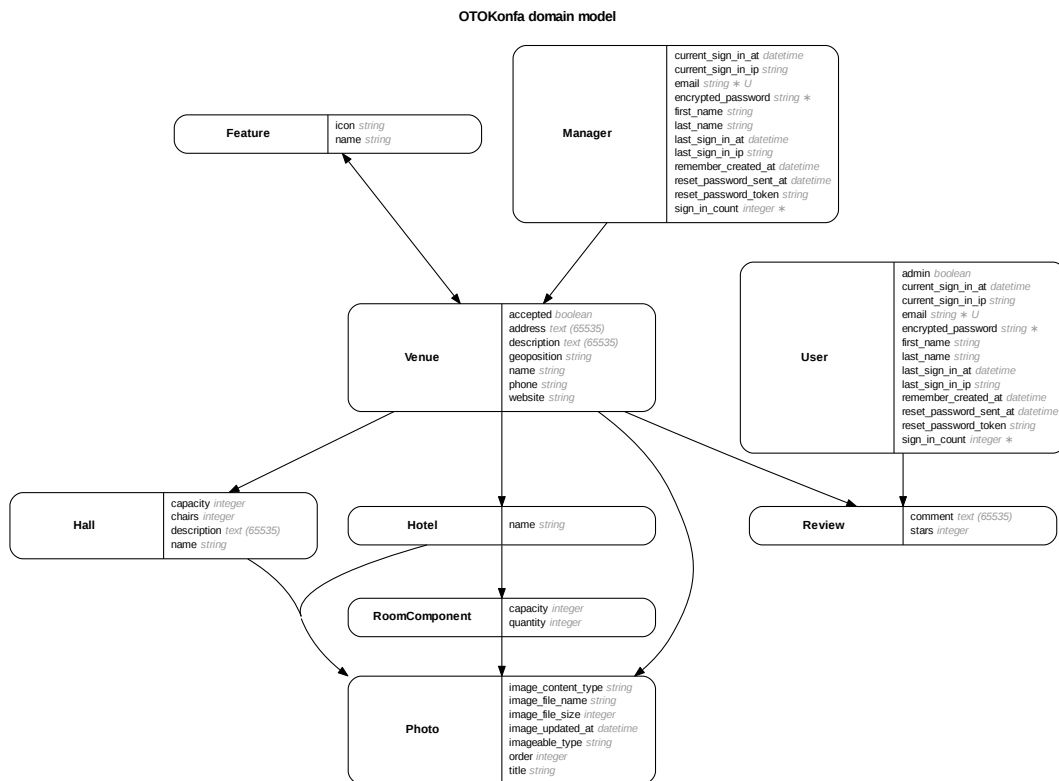
Do kompilacji Sass na CSS

13. Paperclip

Biblioteka do załączania zdjęć

2. Opis bazy danych

2.1. Diagram ERD



Rys. 2.1. Diagram ERD (zobacz pełny)

2.2. Krótki opis

Struktura bazy danych jest automatycznie generowana przez technologię w której wykonany został projekt (Ruby on Rails, a konkretnie jego komponent Active Record). Domyślnie projekt jest tworzony z użyciem adaptera do baz MySQL, jednak technologia oferuje łatwą możliwość podmiany adaptera na inny (PostgreSQL, SQLite i inne), bez konieczności obszernych zmian w kodzie.

3. Specyfikacja modułów

3.1. Użytkownicy

1. Moduł widoku głównego

Wyświetla główny widok aplikacji.

2. Moduł wyszukiwania

Pozwala na wyszukiwanie ośrodków za pomocą podanych kryteriów, takich jak: nazwa, lokalizacja, ocena, posiadane udogodnienia (np. siłownia, ksero). itp.

3. Moduł przeglądania

Umożliwia przeglądanie wyników (konkretnych ośrodków) oraz przejście do widoku szczegółowego ośrodka.

4. Moduł widoku szczegółowego

Wyświetla szczegółowe dane wybranego ośrodka, pozwala obejrzeć galerię, dodać i przeczytać oceny, skontaktować się z ośrodkiem za pomocą formularza kontaktowego.

5. Moduł edycji danych ośrodka

Umożliwia menadżerom na edycję danych o ośrodku: nazwy, opisu, danych kontaktowych, udogodnień.

6. Moduł formularza kontaktowego

Pozwala na wysłanie maila do ośrodka: należy podać swoje dane oraz wiadomość.

7. Moduł oceny

Umożliwia ocenienie ośrodka - wystarczy ocena, można dołączyć komentarz. Ocenianie jest możliwe tylko dla zalogowanych użytkowników.

8. Moduł rejestracji

Pozwala na rejestrację nowego użytkownika, wymagane dane to: imię, nazwisko, adres email oraz hasło.

9. Moduł rejestracji menadżera

Pozwala na rejestrację użytkownika jako menadżera. Wymagane dane są identyczne jak przy zwykłej rejestracji.

10. Moduł logowania

Umożliwia wcześniej zarejestrowanym użytkownikom zalogowanie się.

11. Moduł edycji danych użytkownika

Wyświetla formularz edycji danych zalogowanego użytkownika lub menadżera: imienia, nazwiska, adresu email oraz hasła.

3.2. Deweloperzy

1. Moduł widoku głównego

Wyświetla główny widok aplikacji.

Wejście: brak.

Wyjście: brak.

Pozwala na przejście do modułów: wyszukiwania (2), rejestracji (8), rejestracji menadżera (9), logowania (10), edycji danych użytkownika (11).

Przejście do modułu (11) możliwe tylko dla zalogowanych użytkowników,

Moduł potrzebuje dostępu do bazy danych.

2. Moduł wyszukiwania

Umożliwia użytkownikom wyszukiwanie ośrodka.

Wejście: dane wprowadzone przez użytkownika (nazwa, lokalizacja, atuty).

Wyjście: lista ośrodków spełniających kryteria użytkownika; przekazuje dane do modułu przeglądania (3).

Moduł potrzebuje dostępu do bazy danych.

3. Moduł przeglądania

Umożliwia przeglądanie wyników z modułu wyszukiwania. Pozwala na przejście do modułu widoku szczegółowego (34).

Wejście: lista ośrodków z modułu wyszukiwania (2).

Wyjście: wybrany ośrodek; przekazywany do modułu widoku szczegółowego (4).

Moduł potrzebuje dostępu do bazy danych.

4. Moduł widoku szczegółowego

Wyświetla szczegółowe dane wybranego ośrodka. Pozwala na przejście do modułów: edycji danych ośrodka (5), formularza kontaktowego (6), oceny (7).

Wejście: wybrany ośrodek z modułu przeglądania (3).

Wyjście: dane ośrodka do modułów (5), (6), (7).

Moduł potrzebuje dostępu do bazy danych.

5. Moduł edycji danych ośrodka

Umożliwia menadżerom na edycję danych ośrodka.

Wejście: dane ośrodka z modułu widoku szczegółowego (4).

Wyjście: zmienione dane do bazy danych.

Po zakończeniu edycji powrót do modułu widoku szczegółowego (4).

Moduł potrzebuje dostępu do bazy danych.

6. Moduł formularza kontaktowego

Pozwala na wysłanie maila do ośrodka.

Wejście: dane ośrodka z modułu widoku szczegółowego (4).

Wyjście: wiadomość wysłana na adres email ośrodka z wejścia.

Mail nie powinien zostać wysłany, jeżeli nie zostanie podane któreś z następujących: email, nr telefonu, wiadomość.

Po wysłaniu maila powrót do modułu widoku szczegółowego (4).

Moduł potrzebuje dostępu do bazy danych.

7. Moduł oceny

Umożliwia zalogowanym użytkownikom ocenienie ośrodka.

Wejście: dane ośrodka z modułu widoku szczegółowego (4).

Wyjście: dodane ocena do bazy danych.

Ocena nie powinna zostać dodana jeśli nie zostanie podana ocena. Komentarz jest nie-obowiązkowy.

Po dodaniu oceny powrót do modułu widoku szczegółowego (4).

Moduł potrzebuje dostępu do bazy danych.

8. Moduł rejestracji

Pozwala na rejestrację nowego użytkownika.

Wejście: użytkownik podaje dane wymagane do rejestracji: imię, nazwisko. adres email, hasło, powtórzone hasło.

Wyjście: dodanie nowego użytkownika do bazy danych.

Użytkownik nie powinien zostać dodany jeśli brakuje wymaganych danych.

Po zakończeniu powrót do poprzedniego modułu.

Moduł potrzebuje dostępu do bazy danych.

9. Moduł rejestracji menadżera

Pozwala na rejestrację użytkownika jako menadżera.

Wejście: użytkownik podaje dane wymagane do rejestracji: imię, nazwisko. adres email, hasło, powtórzone hasło.

Wyjście: dodanie nowego menadżera do bazy danych.

Menadżer nie powinien zostać dodany jeśli brakuje wymaganych danych.

Po zakończeniu powrót do poprzedniego modułu.

Moduł potrzebuje dostępu do bazy danych.

10. **Moduł logowania**

Umożliwia wcześniej zarejestrowanym użytkownikom zalogowanie się.

Wejście: użytkownik podaje dane do weryfikacji: email, hasło.

Wyjście: brak, użytkownik zostaje zalogowany.

Po zakończeniu powrót do poprzedniego modułu.

Moduł potrzebuje dostępu do bazy danych.

11. **Moduł edycji danych użytkownika**

Pozwala na edycję danych zalogowanego użytkownika/menadżera.

Wejście: użytkownik podaje dane, które chce zmienić (jedna lub więcej): imię, nazwisko, adres email, hasło.

Po zakończeniu powrót do poprzedniego modułu.

Moduł potrzebuje dostępu do bazy danych.

A. Diagramy ERD

[Powrót do diagramów komunikacji w tekście...](#)

OTOKonfa domain model

