

UNIwersytet Gdański  
Wydział Matematyki, Fizyki i Informatyki

Marcin Dawidowski  
nr albumu: 231010

## Let's Bid It – portal aukcyjny

Praca licencjacka na kierunku:

INFORMATYKA

Promotor:

dr Włodzimierz Bzyl

Gdańsk 2017



## Streszczenie

W pracy przedstawiono rozwojową wersję aplikacji webowej „Let's Bid It” do tworzenia i publikowania aukcji oraz ogłoszeń.

W aplikacji zaimplementowano ich kategoryzację, wyszukiwanie, a także tworzenie kategorii w hierarchii.

Zaprojektowano widok strony głównej, który wyświetla losowe aukcje znajdujące się w portalu oraz panele administracyjny i użytkownika, które są widoczne po zalogowaniu się na konkretny typ konta. Każda z aukcji wyświetlana jest w kolejności rosnącej według czasu zakończenia.

Do tworzenia aplikacji wykorzystano Ruby on Rails, Twitter Bootstrap, Plataformatec Devise, reCaptcha, CKEditor, jQuery Turbolinks, Chartkick, ActsAsTree.

Projekt wdrożony został na platformie heroku . com i jest dostępny pod adresem:  
<https://ror-aukcja.herokuapp.com/>.

## Słowa kluczowe

ruby on rails, gem, heroku, auction, online shopping, bid

# Spis treści

Wprowadzenie . . . . .	6
1. Let's Bid It w praktyce . . . . .	8
1.1. Porównanie z dostępnymi rozwiązaniami . . . . .	8
1.1.1. Allegro . . . . .	8
1.1.2. OLX . . . . .	8
1.2. Możliwości zastosowania praktycznego . . . . .	9
2. Projekt i analiza . . . . .	10
2.1. Diagram ERD . . . . .	10
2.2. Aktorzy i przypadki użycia . . . . .	11
2.3. Diagram przepływu danych . . . . .	12
3. Implementacja . . . . .	13
3.1. Technologie wykorzystane do stworzenia aplikacji . . . . .	13
3.1.1. Ruby on Rails . . . . .	13
3.1.2. Twitter Bootstrap . . . . .	13
3.1.3. Devise . . . . .	13
3.1.4. Pozostałe rozwiązania użyte przy realizacji projektu . . . . .	14
3.2. Struktura aplikacji . . . . .	15
3.2.1. Strona główna . . . . .	15
3.2.2. Aukcje . . . . .	15
3.2.3. Kategorie . . . . .	16
3.2.4. Obserwowane . . . . .	16
3.2.5. Nawigacja . . . . .	17
Zakończenie . . . . .	18
Bibliografia . . . . .	19
A. Płyta CD zawierająca kod projektu . . . . .	20

wersja wstępna [2017.5.11]	5
Spis rysunków . . . . .	21
Oświadczenie . . . . .	22

# Wprowadzenie

Chęć stworzenia czegoś od podstaw oraz własne doświadczenie w korzystaniu z portali aukcyjnych sprawiło, że dla mnie jako programisty stworzenie takiej aplikacji, byłoby niezwykle satysfakcjonujące i pozwoliłoby rozwinąć umiejętności. Ponadto stworzenie takiego portalu może ułatwić mi w przyszłości implementację różnego rodzaju sklepów internetowych i innych serwisów związanych z handlem przez Internet.

Mając więc gotowy pomysł na projekt pozostało mi jedynie wybrać odpowiednią technologię, która pomoże mi go zrealizować. W trakcie studiów poznałem kilka różnych języków programowania i frameworków, które sprawdziłyby się w tego typu projekcie. Po rozważeniu wszystkich za i przeciw, mój wybór padł na język Ruby, wspierany przez framework Rails, który udało mi się poznać już wcześniej. Największą zaletą wybranej przeze mnie technologii jest przede wszystkim fakt, iż Ruby on Rails posiada szeroką gamę dostępnych rozwiązań w postaci gemów, które w dużym stopniu ułatwiają pracę nad projektem. Kolejną kwestią jest fakt, że technologia ta wykorzystuje architekturę MVC, co znacznie ułatwia pracę. Poza tym Ruby on Rails świetnie współpracuje z platformami takimi jak Heroku, czy DigitalOcean.

Aplikacja Let's Bid It jest wersją rozwojową portalu aukcyjnego, dlatego też nie posiada jeszcze wszystkich funkcjonalności, które pozwoliłyby w pełni wykorzystać potencjał serwisu.

Spośród rzeczy, które udało się zrealizować, mogę wyróżnić przede wszystkim obsługę aukcji. Ponadto zaimplementowano ich kategoryzację przy pomocy gemu Act as tree. Kolejną rzeczą jest wdrożenie zaawansowanego edytora tekstowego CKEditor, dzięki któremu dodając aukcję, użytkownik może swobodnie manipulować jej opisem. Dodatkowo zaimplementowałem zabezpieczenie reCaptcha przy logowaniu i rejestracji, a także statystyki strony, które dostępne są dla administratorów. Ważnym elementem Let's Bid It jest również grafika dołączana do każdej aukcji. W związku z tym, że aplikacja znajduje się na platformie Heroku, głównym problemem było przetrzymywanie plików graficznych, które na Heroku usuwane są automatycznie.

Rozwiązaniem tego problemu okazało się skorzystanie z serwisu Cloudinary, który pozwala bezpłatnie przechowywać pliki w chmurze, a co najważniejsze, możliwe jest skonfigurowanie go tak, by współpracował z aplikacją napisaną w Ruby on Rails, w czym pomocny okazał się gem cloudinary.

Najważniejszą rzeczą, której nie udało się zrealizować jest wdrożenie systemu mającego na celu symulację płatności internetowych, a także zaawansowana wyszukiwarka opierająca się na silniku Elasticsearch.

## Let's Bid It w praktyce

### 1.1. Porównanie z dostępnymi rozwiązaniami

#### 1.1.1. Allegro

Jest to największa, a także najpopularniejsza platforma transakcyjna on-line w Polsce, oferująca setki produktów w niezwykle konkurencyjnych cenach. Serwis udostępnia możliwość zakupu natychmiastowego, licytacji, a także wystawiania ogłoszeń. Typ danej aukcji, zależy tylko i wyłącznie od jej właściciela.

Jeśli chodzi o wystawianie aukcji, to proces wygląda podobnie do Let's Bid It. Tak jak w mojej aplikacji użytkownik może skorzystać z zaawansowanego edytora tekstowego, dodaje zdjęcia i przyporządkowuje ją do konkretnej kategorii. Główną różnicą jest fakt, że Allegro pozwala na wystawienie przedmiotu z użyciem opcji „Kup Teraz”, czego Let's Bid It nie ma.

#### 1.1.2. OLX

Serwis ten jest jednym z najpopularniejszych portali ogłoszeniowych na świecie. Użytkownicy mogą udostępniać lokalne ogłoszenia dotyczące sprzedaży oraz usług.

Główną różnicą pomiędzy OLX, a Let's Bid It jest fakt, że wymieniony wyżej portal nie udostępnia funkcji licytacji towaru, natomiast moja aplikacja posiada ją, a także tak jak OLX daje użytkownikowi prawo wystawienia takiej technologii jakowaru, czy też usługi jako zwykłe ogłoszenie. Ponadto stworzony przeze mnie portal aukcyjny, dzięki zaawansowanemu edytorowi tekstowemu pozwala stworzyć użytkownikom dużo bardziej oryginalne opisy, co nie jest możliwe przy korzystaniu z wyżej wymienionego portalu.



## 1.2. Możliwości zastosowania praktycznego

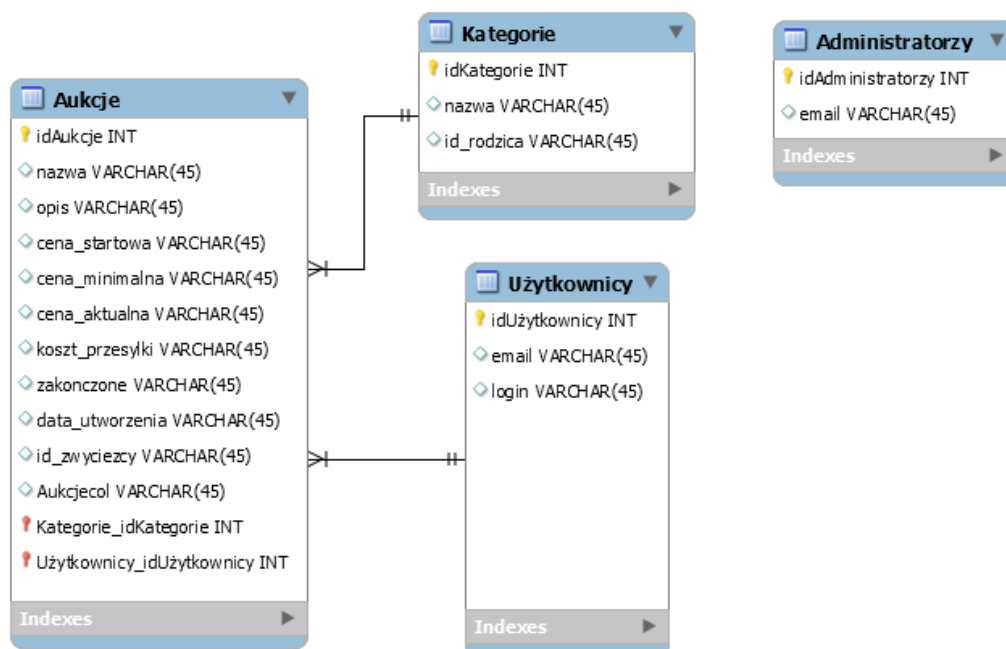
Głównym założeniem serwisu Let's Bid It jest przede wszystkim portal aukcyjny i jest to główna możliwość zastosowania napisanej przeze mnie aplikacji w praktyce. Poza tym portal ten użyty może zostać również do innych celów:

- Portal ogłoszeniowy – jest jedną z możliwości wykorzystania Let's Bid It. Potrzeby takiej aplikacji zaspokajają praktycznie wszystkie wdrożone w mojej aplikacji funkcjonalności, takie jak kategoryzacja, a także prosta obsługa dodanych ogłoszeń. Ponadto użytkownik takiego serwisu będzie miał do dyspozycji bardzo prosty interfejs ułatwiający poruszanie się po stronie, a także zaawansowany edytor, który pozwoli na to, by jego ogłoszenie wyglądało wyjątkowo.
- Sklep internetowy – jest to kolejna możliwa opcja zastosowania mojej aplikacji w praktyce, co można osiągnąć przy niedużym nakładzie pracy. Tak jak przy portalu ogłoszeniowym, dostępne są przede wszystkim kategoryzacja, a także obsługa dodawanych aukcji oraz zaawansowane narzędzia do zarządzania wyglądem aukcji.

## ROZDZIAŁ 2

# Projekt i analiza

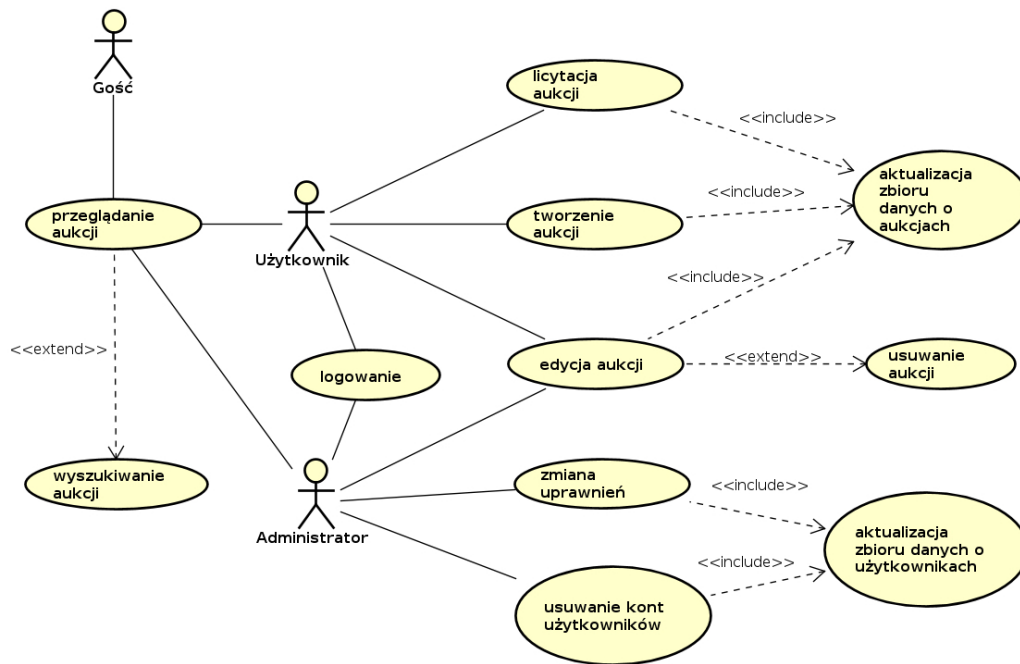
### 2.1. Diagram ERD



Rysunek 2.1. Diagram związków encji

Źródło: Opracowanie własne za pomocą MySQL Workbench

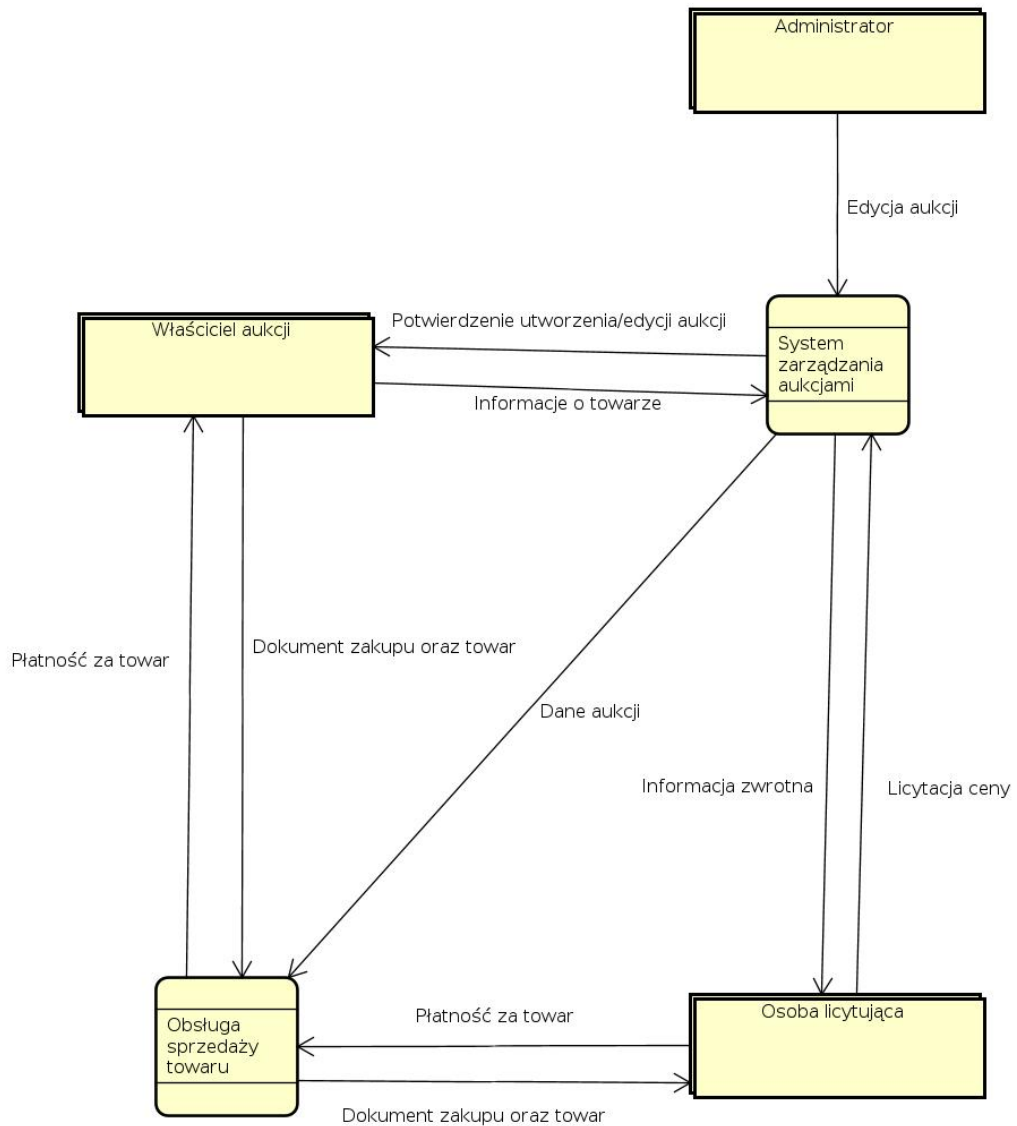
## 2.2. Aktorzy i przypadki użycia



Rysunek 2.2. Diagram Przypadków Użycia.

Źródło: Opracowanie własne za pomocą programu Astah [?].

### 2.3. Diagram przepływu danych



Rysunek 2.3. Diagram Przepływu Danych.

Źródło: Opracowanie własne za pomocą programu Astah [?].

## Implementacja

W celu stworzenia aplikacji, która ma w odpowiedni sposób realizować postawione przed nią założenia, potrzebne są odpowiednio dobrane technologie oraz narzędzia, dzięki którym uda się spełnić postawione przed nią wymagania. Dobór odpowiedniej technologii, a także frameworków jest kluczowy do skutecznej implementacji aplikacji.

### 3.1. Technologie wykorzystane do stworzenia aplikacji

#### 3.1.1. Ruby on Rails

Do stworzenia projektu wykorzystana została technologia Ruby on Rails bazująca na języku Ruby. W aplikacji wykorzystano Ruby w wersji 2.3, a także framework Rails w wersji 5.0.1. Całość stworzona została przy pomocy architektury MVC (ang. Model-View-Controller).

#### 3.1.2. Twitter Bootstrap

Widoki utworzone zostały przy pomocy frameworka Twitter Bootstrap, który pozwala na proste tworzenie graficznego interfejsu stron internetowych z wykorzystaniem gotowych rozwiązań bazujących na językach HTML i CSS. Główną zaletą tego frameworka jest responsywność, czyli zapewnienie poprawnego wyświetlania stron WWW na różnych urządzeniach.

#### 3.1.3. Devise

Budując portal aukcyjny koniecznym wymogiem jest przede wszystkim obsługa użytkowników i administratorów aplikacji. W tym celu wdrożony został gem Devise, który w znacznym stopniu ułatwia autentykację na stronie. Dzięki zastosowaniu

powyższego gemu otrzymujemy niemalże gotowe widoki, kontroler oraz model, co pozwala na bezproblemowe zarządzanie użytkownikami.

#### 3.1.4. Pozostałe rozwiązania użyte przy realizacji projektu

Heroku - to platforma do tworzenia i hostowania aplikacji w chmurze, dzięki której w łatwy sposób możliwe jest wdrożenie jej na serwer zewnętrzny.

Cloudinary - jest to serwis pozwalający na przechowywanie danych w chmurze. Dzięki połączeniu go z aplikacją możliwe było bezproblemowe przechowywanie grafik dołączanych przez użytkowników do aukcji i rozwiązanie problematycznego przechowywania tych plików na serwerze Heroku.

CKEditor - jest to wizualny edytor tekstowy języka HTML, który umożliwia użytkownikowi na wybranie między innymi konkretnej czcionki, jej rozmiaru, koloru liter, czy też ich stylu. Poza tym użytkownik może również ustawić wyrównanie tekstu, a także możliwe jest wstawienie listy, tabeli, odnośników, a także obrazów.

ActsAsTree - gem ten wykorzystany został w celu stworzenia drzewa kategorii, które ułatwia w sposób zdecydowany nawigację po stronie, a także sprawia, że dane na stronie są uporządkowane. Sama implementacja przebiega w środowisku technologii Ruby on Rails, więc nie wymaga bezpośredniej instalacji.

reCaptcha - zabezpieczenie dzięki któremu możliwość wysyłania danych będą mieli jedynie sprawdzeni użytkownicy.

Chartkick - gem pozwalający na tworzenie i wyświetlanie różnego rodzaju wykresów na podstawie danych zawartych w aplikacji.

## 3.2. Struktura aplikacji

### 3.2.1. Strona główna

Strona główna aplikacji zaprojektowana została tak, aby w prosty sposób udostępniać wszystkie funkcjonalności aplikacji. Głównymi elementami strony głównej jest panel z wszystkimi nazwami kategorii nadrzędnych, a także miniatury wraz z tytułami losowo wybranych aukcji z całego ich zbioru. Ponadto po zalogowaniu na konto użytkownika, na stronie głównej informacje o aukcjach, które dany użytkownik dodał do obserwowanych.

### 3.2.2. Aukcje

Głównym elementem Let's Bid It są aukcje. Każda z nich posiada atrybuty takie jak:

- identyfikator autora aukcji
- kategorię, do której należy
- nazwę
- szczegółowy opis
- cenę początkową
- cenę minimalną
- koszt przesyłki
- informację o tym, czy aukcja publikowana jest jako ogłoszenie
- zdjęcie, które będzie publikowane jako główne.

Widok aukcji składa się z czterech stron, które różnią się ze względu na dostęp i wyświetlane na niej dane.

Pierwszym elementem jest strona index dostępna tylko dla zalogowanych użytkowników. Wyświetlane na niej są wszystkie aukcje, które utworzył aktualnie zalogowany użytkownik. Aukcje wyświetlane są w tabeli według identyfikatorów, zaś każdej

aukcji przypisane są akcje pokaż, edytuj oraz usuń. W górze strony wyświetlany jest również odnośnik do utworzenia nowych aukcji przez aktualnie zalogowanego użytkownika.

Następnymi elementami aukcji są strony, które umożliwiają użytkownikom dodawanie oraz edytowanie aukcji. Są to dwa oddzielnie funkcjonujące elementy, jednak łączy je wspólny formularz wypełniany przy tworzeniu/edycji aukcji. Formularz ten zawiera wszystkie atrybuty aukcji, które zostały wymienione powyżej.

Najważniejszym elementem aukcji jest strona, na której każda z nich, jest wyświetlana wraz z wszystkimi informacjami, które zostały zawarte przy jej tworzeniu. Strona oraz wszystkie informacje na temat aukcji jest widoczna dla każdego, jednak niektóre jej funkcjonalności widoczne są jedynie dla zalogowanych użytkowników. Elementami, które widoczne są po zalogowaniu, to przycisk dodający aukcję do obserwowanych, a także panel, za pomocą którego możliwe jest licytowanie aukcji. Sam panel zabezpieczony został w taki sposób, aby niemożliwe było dodanie oferty niższej niż cena aktualna.

### 3.2.3. Kategorie

Uporządkowanie aukcji niemożliwe byłoby bez ich kategoryzacji, dlatego też koniecznym wydawało się wdrożenie kategorii, przy czym niezwykle pomocny był gem ActsAsTree.

Tak jak aukcje, kategorie składają się ze stron, które służą do ich dodawania, wyświetlania oraz edycji. Cechą różniącą kategorie i aukcje jest fakt, że strony oznaczone jako index, new oraz edit widoczne są tylko dla osób zalogowanych na konto administratora.

Ogólnodostępnym elementem kategorii jest widok udostępniający dane o konkretnej kategorii. Na stronie tej widoczne są wszystkie podkategorie danej kategorii, a także aukcję, które do niej należą.

### 3.2.4. Obserwowane

Dodanie aukcji do obserwowanych, jest jedną z funkcjonalności, która umożliwia łatwiejsze ich odnalezienie.

Widok opiera się na prostej tabeli, która wyświetla podstawowe dane na temat



aukcji, które zostały dodane do obserwowanych. Rekordy tabeli wyposażone zostały również w przycisk, dzięki któremu możliwe jest usunięcie elementu z obserwowanych.

### 3.2.5. Nawigacja

Nawigacja na stronie umożliwiona została dzięki paskowi nawigacyjnemu, który umieszczony został na górze strony. Pasek ten w zależności od typu użytkownika wyświetla różne wartości.

Osoba odwiedzająca portal jako gość ujrzy na samej górze odnośnik do strony głównej, a także linki do rejestracji i logowania, które pozwolą na korzystanie z większej ilości funkcjonalności, które zostały wdrożone w Let's Bid It.

Zwykły użytkownik po zalogowaniu, poza odnośnikiem do strony głównej będzie miał również możliwość skorzystania z panelu użytkownika, edycji profilu, czy też po prostu z opcji wylogowania się z aktualnego konta.

Administrator strony, w odróżnieniu do użytkownika będzie mógł skorzystać z panelu administracyjnego, nie mając jednakowoż dostępu do panelu użytkownika. Pozostałe funkcje paska nawigacyjnego pozostaną takie same jak dostępne dla zalogowanego użytkownika.

## Zakończenie

W trakcie pracy nad projektem udało mi się zrealizować praktycznie wszystkie jego założenia. Praca ta pozwoliła mi nabyć nowe doświadczenia, a także poszerzyć posiadane umiejętności. Podczas realizacji aplikacji udało mi się wykorzystać niemal wszystkie umiejętności, które nabyłem w trakcie odbywania studiów.

# Bibliografia

- [1] David A. Black. Ruby. Przewodnik programisty wyd. II. HELION S.A. 2015.
- [2] John Elder. Ruby on Rails - Tworzenie aplikacji WWW Codemy.com 2016.
- [3] Oficjalna dokumentacja - Gem Devise  
<https://github.com/plataformatec/devise/> (dostęp 5.03.2017)
- [4] Oficjalna dokumentacja - Gem ActsAsTree  
[https://github.com/amerine/acts\\_as\\_tree](https://github.com/amerine/acts_as_tree) (dostęp 8.04.2017)
- [5] Oficjalna dokumentacja - Gem reCAPTCHA  
<https://github.com/ambethia/recaptcha/> (dostęp 10.04.2017)
- [6] Oficjalna dokumentacja - Gem CKEditor for Rails  
<https://github.com/galetahub/ckeditor/> (dostęp 10.04.2017)
- [7] Oficjalna dokumentacja - Gem Chartkick  
<https://github.com/ankane/chartkick/> (dostęp 28.04.2017)

DODATEK A

Płyta CD zawierająca kod projektu

## Spis rysunków

2.1. Diagram związków encji . . . . .	10
2.2. Diagram Przypadków Użycia. . . . .	11
2.3. Diagram Przepływu Danych. . . . .	12

# Oświadczenie

Ja, niżej podpisany(a) oświadczam, iż przedłożona praca dyplomowa została wykonana przeze mnie samodzielnie, nie narusza praw autorskich, interesów prawnych i materialnych innych osób.

.....

data

.....

podpis