

Bachelor's Degree Report (BSc)

AUN SENIOR PROJECT REPOSITORY

A Project Report

By

**Miebaka Glory Worika
(A00016415)**

Supervisor: Alexey Vedischev

School of Information, Technology & Computing

American University of Nigeria

Spring 2018

AUN SENIOR PROJECT REPOSITORY

Miebaka Glory WORIKA

BSc Report

**Project Report submitted in
Partial fulfillment of the requirements for the
Degree of Bachelor in Computer Science**

Supervisor: Alexey Vedishchev

School of Information, Technology & Computing

American University of Nigeria

Spring 2018

DECLARATION

I declare that this project work was carried out by myself and has not been previously submitted for the degree. Furthermore, the report was written unaided in our own words, apart from any quoted material which we identified clearly in the correct manner and fully acknowledged as work by others. The work and the report were carried out under the guidance of Mr. Alexey Vedischev.

Student's Name:
Miebaka Glory Worika

Signature.....

Supervisor's Name:
Mr. Alexey Vedischev

Signature.....

Program Chair's Name:
Dr. Narasimha Rao Vajjhala

Signature.....

Dean's Name:
Dr. Mathias Fonkam

Signature.....

SDP Coordinator's Name:
Dr. Abubakar Sadiq Hussaini

Signature.....

ABSTRACT

Every semester, students from each school in AUN take on their Senior Project. This project is expected to be a culmination of their work at the University and represent what they have learned. One of the biggest issues students have is what project they should take on. From the inception of AUN, there have undoubtedly been a great number of projects that have been done. Some could have been expanded upon or even improved; there has however seldom been an opportunity for this to take place. The reason being a lack of an easy to access means to seeing what projects have been taken up in the past and how they went.

To resolve this dilemma, I am proposing a web application capable of functioning in this much needed capacity. The project will be titled SDP Repository. The resulting application will be built to be able to house existing projects and provide means for students to be able to search and go through the previously done work. Through this application, the students may find inspiration or even a project they can improve or expand upon. The project hopes to reduce the amount of repetition among Senior Design Projects that has often plagued SITC.

ACKNOWLEDGEMENT

I am grateful for the guidance rendered to me by Mr. Vedishchev. Additionally, I acknowledge the advice given to me by Dr. Vajjhala on my presentation. My gratitude is particularly expressed towards Mr Ifeoluwa Funsho and Ms. Fiyinfoluwa Owoeye-wyse for all the mental support offered to me during this process, Mr Olaoluwa Nwobodo for his patient review of my ‘progress’ with my User Interface design. Furthermore, I wish to extend my gratitude to Dr Hussaini for the successful organization of the process and without whose guidance this entire project would have been unable to happen. Finally, I wish to thank my parents for their financial support. Without any of the aforementioned people, I would not have been able to complete this project.

Table of Contents

DECLARATION	i
ABSTRACT.....	ii
ACKNOWLEDGEMENT	1
LIST OF FIGURES	3
ACRONYMS	4
CHAPTER ONE	5
1.0 INTRODUCTION	5
1.1 BACKGROUND AND MOTIVATION	5
1.2 PROBLEM STATEMENT	6
1.3 PROJECT OBJECTIVE	6
1.4 OUTLINE OF THE REPORT	7
CHAPTER TWO	8
2.0 REQUIREMENTS	8
2.1 FUNCTIONAL REQUIREMENTS	8
2.2 NON-FUNCTIONAL REQUIREMENTS	9
2.3 TECHNOLOGIES USED	9
2.4 USE-CASE DIAGRAM	10
CHAPTER THREE	14
3.0 DESIGN	14
3.1 ARCHITECTURAL PATTERN	14
3.2 INTERFACE DESIGN	16

LIST OF FIGURES

Figure 1: Use Case Diagram	11
Figure 2 ER Diagram	12
Figure 3 Rails MVC	15
Figure 4 Home Page.....	16
Figure 5 Project Index Page	17
Figure 6 Project Information Page	17

ACRONYMS

MVC Model View Controller

RoR Ruby on Rails

CHAPTER ONE

1.0 INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

The Senior Project be it research or design, represents the culmination of a students learning at university. Through this project, the student is expected to be able to showcase the knowledge they have gained over the course of their learning. In an ideal situation it should be possible to build upon the work done by previous students as opposed to starting from scratch; this unfortunately is not the case. Currently, students start new projects every semester some of which are almost exact replicas of previously done projects. What instead should happen, is the development of those previous ideas. For this development to take place – a preference particularly for the school of SITC, it is required that there be a tool for students to be able to see and access what has been done in previous semesters by their fellow students. Under current circumstance, the most likely place for the records to be kept is with the Digital Department Services under the AUN library. Unfortunately, the resource is poorly managed and only the final report for any uploaded project is available. This project is intended to be able to serve in this role more wholesomely. It will allow students to be able to see what has been done before, even with the code for the application in the case of a design project. This will easily allow for the continuation of previous projects.

1.2 PROBLEM STATEMENT

Every semester, students from each school in AUN take on their Senior Project. This project is expected to be a culmination of their work at the University and represent what they have learned. One of the biggest issues students have is what project they should take on. From the inception of AUN, there have undoubtedly been a number of projects that have been done. Some could have been expanded upon or even improved; there has however seldom been an opportunity for this to take place. This is due to the lack of an accessible means to find out what projects have been done. Furthermore, for projects that are recorded within the university, usually one can only access the final report which is of little use as the student has to build the project to the level reached and then add to the project. Continuation of prior projects by students looking to do their projects, would be made much easier if all the project details were stored. The lack of means has meant that consequently, several projects that have passed as Senior Design Projects have been clones of previous projects in all but name.

1.3 PROJECT OBJECTIVE

This project aims to produce a web application that can be used to store previous and future senior projects regardless of whether they are design or research. This application will be designed to be easily allow the user to get a description of the projects and the option to download the relevant file for the project. This project will be able to store projects for all the schools and senior project courses, alongside providing a search feature to filter projects by supervisor(s), name of project, student name or the particular course. It will be intuitive to use with a minimalistic user interface.

1.4 OUTLINE OF THE REPORT

Chapter 1 clearly introduces the system to be developed as well as its problems and the objectives.

Chapter 2 will state the requirements for the system. This will be done with the provision of a use case diagram and textual description.

Chapter 3 will elaborate upon the development of the system. This chapter will cover the methodology, analysis of the framework used; noting challenges during development.

Chapter 4 describes plausible future implementations for the project and will conclude the report.

CHAPTER TWO

2.0 REQUIREMENTS

The objective of defining the requirements are to define the functions the system must be able to perform. These requirements can either be functional or non-functional.

Functional requirements were split into two categories.

- Essential
- Beneficial

2.1 FUNCTIONAL REQUIREMENTS

The functional requirements refer to the functions the system must be able to perform.

The essential functions are below:

- Only the administrators are able to log in to the system
- The system must provide adequate measures for security of credentials
- Administrators are able to Upload, Modify and Delete Senior Projects stored in the application.
- The Administrators must be able to create and modify information concerning project courses and the schools they fall under.
- The Database for the application must contain the necessary measures to ensure integrity of information stored on the system.
- All users must be able to see information concerning Senior Projects
- Only Administrators may see the information on 'Schools' or 'Project Courses' and are able to modify those entities.

- Regular users should not be able to access functions intended for Administrators.

Beneficial Requirements are below:

- The Administrators are able to have profile pages and modify their information
- Regular users can have accounts that grant them extra features
- Users are able to indicate their interest in Projects

2.2 NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements are those that the application must meet but are not related directly to the functions of the system. They are as follows:

Price: The application is intended to be non-profit and therefore cannot host any goods that do not suit that description

Interface: The application must be clearly visible with an uncluttered interface – it must not use harsh colours or colours that can be mixed by the colour blind.

Errors: There must be error messages that clearly inform the administrator of any errors that have occurred to ensure the integrity of the database.

Security: Using Ruby on Rails as the framework, the system is able to meet necessary security standards to prevent SQL injection, Cross Site Scripting and other means of infiltrating the system to cause harm.

2.3 TECHNOLOGIES USED

The technologies used to build this application are as follows:

- Ruby Programming Language

- Ruby on Rails Framework for building Web Applications
- Ruby Gems
 - Devise for authentication and session control
 - Mysql2 for database connection
 - Paperclip for file upload and download
 - Simpleform for form creation
 - Bootstrap for Bootstrap connection
- Bootstrap Front End Framework
- MySQL Database Management System

2.4 USE-CASE DIAGRAM

Figure 1 below shows how the users interact with the system. The use cases represent the functionality the system is expected to have

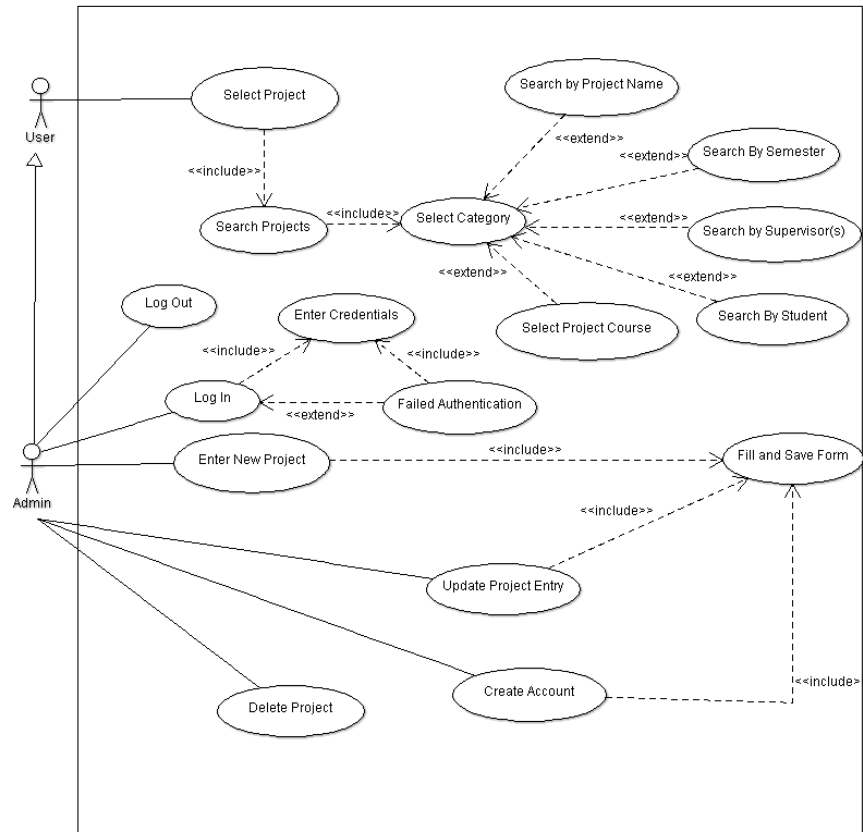


Figure 1: Use Case Diagram

2.5 ENTITY RELATIONSHIP DIAGRAM

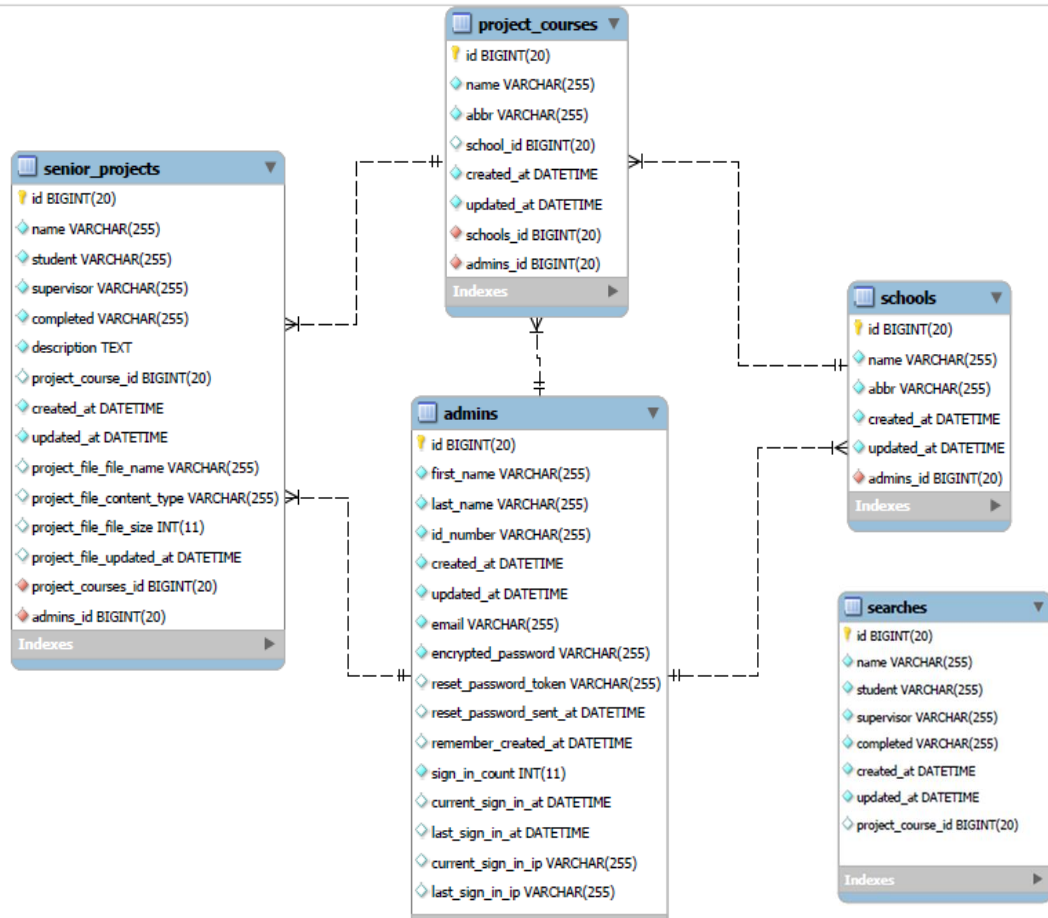


Figure 2 ER Diagram

Figure 2 above depicts the ER Diagram for the database of the Project. As depicted there are five entities:

- Admins
- Schools
- Project Courses
- Senior Projects
- Searches

Schools has a one-to-many non-identifying relationship with project courses which also has the same relationship with Senior Projects. The relationships are non-identifying because each of the entities are uniquely identifiable without the parent table. The Admin table has a one-to-many non-identifying relationship with each other entity with the exception of Searches because only 1 administrator can create each of those entities but several members of those respective entities can be created by a single administrator.

The Searches entity has no relationship to any other entity as it is used to store the variables used for the Advanced Search filter.

CHAPTER THREE

3.0 DESIGN

This application made use of the Ruby on Rails (RoR), therefore made use of the MVC architecture for software development. The User Interface was designed by myself making use of Bootstrap classes and some custom CSS but no template.

3.1 ARCHITECTURAL PATTERN

Model View Controller or MVC is the architectural pattern used by the Ruby on Rails web framework. This is a way of organising your code so that the different parts – Models, Views, Controllers, handle different aspects of the application.

The M which stands for Models refers to the aspect of the application which refers to your database. It handles validations and associations between entities in the database.

V for Views is where the code for the application visuals are written. The Controller C is responsible for receiving the request from the server and collecting information from the Model that will be in the Views. It is the bridge between the Views and Models.

“Controller code acts as a liaison between the Model and the View, receiving user input and deciding what to do with it. It’s the brains of the application, and ties together the model and the view” [1].

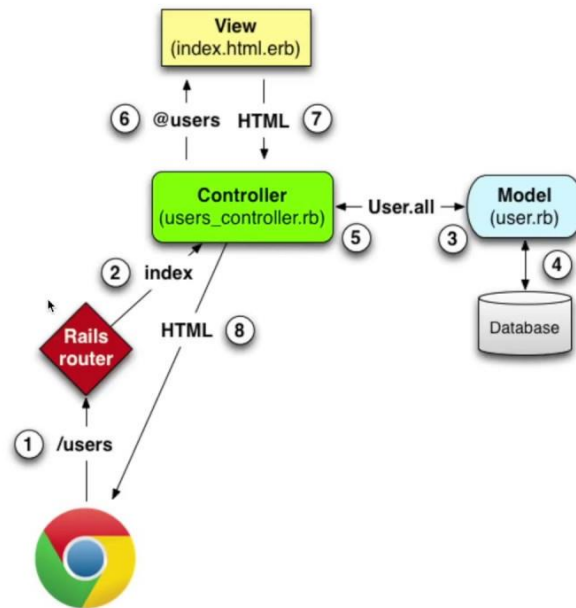


Figure 3 Rails MVC

Figure 3 above, shows how the MVC architectural design is used in the Rails framework. This picture utilises a resource called users, therefore there is a view, controller and model all named using rails conventions. The numbers are interpreted as follows:

1. The browser makes a request `/users`. This request is sent to router.
2. The rails router sends the request to the corresponding method in the controller based on the rails configuration and conventions.
3. The controller matches the request from the router to the appropriate method and executes the necessary commands, if the model is necessary, the necessary call is made to the Model.
4. The model fetches the appropriate information from the database.
5. The information is sent back to the controller from the Model.
6. The controller passes the information it has received as parameters to the views.
7. The view passes the appropriate information to the controller.

8. The controller passes the information to the browser where it is rendered as
Html for the user

3.2 INTERFACE DESIGN

For the interface, the objective was to make a simple yet effective design. I also wanted to utilise a darker mix of colours akin to a 'night theme'. This notion stemmed from the preference users have shown on Twitter for the dark theme. It is less bright and glaring to the eyes. Additionally, I wanted a very uncluttered home page unlike that of Jstor.org. Bootstrap made this easy to see with the easy classification of background colours accessible via classes. The conclusion is seen in the home page depicted in Figure 4 seen below.

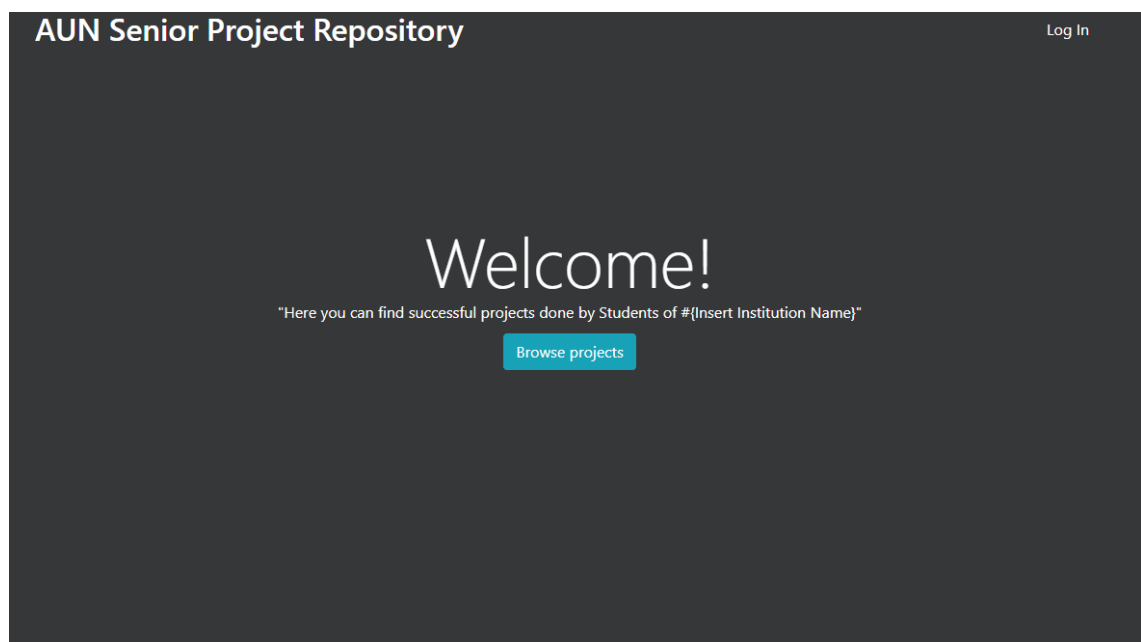


Figure 4 Home Page

AUN Senior Project Repository

Home

Schools

Project Courses

Senior Projects

LOG OUT!

Senior Projects

Search Project Name

Search

Advanced Search

Project Name: [AUN Safe Shop](#)

By: Olaoluwa Nwobodo

Supervised By: Dr Rao

Description: Auction Site for AUN...

Completed: Spring 2018

[Download Project](#)

[Edit Project](#)

Project Name: [Micro RNA](#)

By: Reuben Ikape

Supervised By: Dr Hayatu Raji

Description: Micro RNA can be used for...

Completed: Fall 2018

[Download Project](#)

[Edit Project](#)

Project Name: [AUN SDP - The Second](#)

By: Miebaka Worika

Supervised By: Alexey Vedischev

Description: Lorem Ipsum.....

Completed: Spring 2019

[Download Project](#)

[Edit Project](#)

Project Name: [Effective Data Management](#)

By: Enyetere Ambituni

Supervised By: Dr Mathias Fonkam

Description: Using a case study of flo...

Completed: Spring 2018

[Download Project](#)

[Edit Project](#)

Project Name: [CRM App for Yola Stores](#)

By: Maryam Usman

Supervised By: Dr Rajesh Prasad

Description: An application to help sh...

Completed: Spring 2018

[Download Project](#)

[Edit Project](#)

Figure 5 Project Index Page

Figure 5 above depicts the index page for the projects currently stored. As can be seen the colours are very similar to the 'Twitter Dark Theme' seen on the app. The dark background contrasts with the white text and the blue links are still contrasting enough and bright enough to be seen. Each Project is demarcated by the horizontal line, imitating a tabular format.

AUN Senior Project Repository		Home	Schools	Project Courses	Senior Projects	LOG OUT!
Project Title: CRM App for Yola Stores						
Student Maryam Usman Supervisor Dr Rajesh Prasad Completed Spring 2018		Description An application to help shop owners understand the shopping behavior of their clients enabling them to build a better relationship with them. With this application complaints and suggestions from customers can be recorded and progress towards implementation can be indicated. Download Project				
		Edit Project Back to Index Delete Project				

Figure 6 Project Information Page

Figure 6 above depicts the format for how Project information is shown. The use of lines to demarcate certain aspects remains constant throughout the application in addition to the theme.

A deliberate albeit unusual decision was taken to not design an admin panel for this application instead, the features of the administrator are revealed in tandem with the regular users' access – only appearing when the administrator is logged in. In Figure 5 above the Administrator only options are the Edit Project and Delete Project links at the bottom of the page. These links only appear if an Administrator is logged in.

CHAPTER FOUR

4.0 CONCLUSION AND FUTURE WORK

The application as it stands is still in development; having been submitted as requirement for my degree in Bachelor of Computer Science. As a result, it is not yet ready for implementation and some changes will need to be made such the handling of concurrent users, hosting alongside the optimisation of the features in the application.

4.1 CONCLUSION

This application is capable of fulfilling a role that is considered as very important within the school and will be of great use to students. It will ensure that projects that have large value to the school can be continuously developed and improved by students from the school. It will also encourage the school to implement more projects that students have done. Students will also be able to find more unique ideas or contribute their skills towards correcting flaws in previous ideas or adding more features to them. It would reduce the level of repetition in projects done as fulfilment of the Senior Project.

4.2 FUTURE IMPLEMENTATION

This project has a lot of work that can be added to it to make it much better for its intended role. There are two categories to the additions the project needed. These categories are:

- Production/Hosting

- Future Additions

4.2.1 PRODUCTION/HOSTING

For this application to be ready for hosting, as it stands it will need to be moved from development to production. This is because Ruby on Rails has specific production dependencies. Part of these dependencies is the acquisition of a web/server that can run rails, notably it can't be shared hosting because rails/ruby is not supported by shared hosting. The server would need to be similar to AWS or Google Cloud. Heroku, is not a cost-effective option for hosting rails application at a professional level so there is a need to set up a "VPS (virtual private server). This allows us to set up the OS & dependencies how we need." [2]

Additionally, it is imperative that the application be made capable of allowing for concurrent users to be working on the same time. Further changes will need to be made to the database to allow for scalability as the application will be used by the whole university at the very least.

4.2.2 FUTURE IMPLEMENTATIONS

There are some features that could be added to the application They are listed below:

1. Advanced Search: Currently, while the advanced search feature 'works' it is a far cry from optimal. The database will have to be modified to index the criteria upon which the projects are filtered. Should the system remain the same then a script will need to be written to wipe the search database frequently so it does not slow down the system while it is in use.
2. Interactivity: Currently, only administrators log in to the system. Future implementation can allow for users to be able to have accounts with controlled

access levels to allow them to show ‘interest’ in projects or even commenting. It would better enable the development of ideas.

- [1] Codecademy, “MVC: Model, View, Controller,” [Online]. Available: <https://www.codecademy.com/articles/mvc>. [Accessed 29 April 2018].
- [2] D. Kehoe, Learn Ruby on Rails: Book One and Two, 2016.
- [3] Frontline Utilities LTD, “How To Deploy Ruby on Rails Apps To The Internet (Production/Staging),” Medium, 20 August 2017. [Online]. Available: <https://medium.com/ruby-on-rails-web-application-development/how-to-deploy-ruby-on-rails-apps-to-the-internet-production-staging-49efc503c91d>. [Accessed 29 April 2018].
- [4] M. Hartl, Ruby On Rails Tutorial, Addison-Wesley.

REFERENCES