

**ELABORATION REPORT**

**WEB BASED PROGRAM Q&A AND INFORMATION SYSTEM**

|  |  |
| --- | --- |
| **Prepared by:** |  |
| Consulting Team Name: | **TeeFour** |
| Project Team Leader: | Chang Xing Li |
| Team Members: | Keith Sandy  Gurbachan Singh  Paul Trudel  Juan Antonio Villamizar  Matthew Zigman |

***<2016-2-18>***

# Acknowledgements

**The Client**

Linda Crane

**The Tee-Four Technologies Team**

Chang Xing (Calvin) Li - Team Leader

Keith Sandy - Team Member

Gurbachan Singh - Team Member

Paul Trudel - Team Member

Juan Antonio Villamizar - Team Member

Matthew Zigman - Team Member

**The Project Professor**

Patricia Murphy - Algonquin College

**The Project Advisor**

Todd Kelley - Algonquin College

# Executive Summary

Tee-Four Technologies has started a new project that was authorized by our client, Linda Crane. This project will pertain to each department in the school. This will help future students at Algonquin College decide what program they should choose based on answering multiple choice questions. This project was started because Algonquin College’s website does not help students narrow down the programs they have to choose from. It can take a long time to search through each program that the college provides. In doing this project students can choose a program much faster by just answering some multiple choice questions. Our project will focus on building frontend and backend software for the project. This report will be focused on the design aspect of the project. There will be three designs that will be discussed in this report which are data design, architectural design, and program design.

From our findings, we deem that this project is feasible. The design constructs for this project can be clearly displayed from the scope of the project. In this report, it will focus on the design aspect of the project. This report will cover 3 design types which are data, architectural, and program design.

**Table of Contents**

Acknowledgements i

Executive Summary i

Introduction 2

Data Design 3

Architectural Design 4

Program Design 6

Conclusion 14

Recommendations 14

References 15

Appendices 16

**List of Tables and Figures**

Table 1: Cross Reference Table 6

Figure 1: ERD Diagram of Program Selection System 4

Figure 2: Physical Architecture of the Program Selection Selection 5

Figure 3: Use Case Diagram 13

Figure 4: Sequence Diagram 13

# Introduction

Tee-Four is an innovative software development team that is eager to jump start our latest project presented to us by our client Linda Crane. At 1:00pm Tuesday February 9, 2016, Tee-Four met with Ms. Crane to discuss project ideas and possible approaches to developing our algorithms. During the meeting Linda Crane picked the approach for the next phase of the project.

The idea for the project is going to be a questionnaire system that will help future students determine what program would be appropriate for them. This system will be an add-on to the Algonquin College website. When users explore the programs tab on the website there will be a button they can click, that will bring them to a page consisting of multiple choice questions that will assist them with choosing a potential program. This system will simplify the process of choosing a program by helping students choose a program pertaining to their skills and interests. Implementing this system will give Algonquin College a competitive edge over other schools because it will help students narrow down their choices and start to specialize in their field of interest.

This system will allow department administrators to enter in questions relevant to their department and choose specific programs within that department based on the answers the students have provided. For this project we will focus on making one questionnaire system for the technology's department specifically. This system will give students who have a general interest in technology, a better understanding of what programs are suitable for them to pursue. This means that each department will have its own questionnaire system, instead of having one generalized system that considers all departments. Generalizing the system so that any program offered at Algonquin could be a potential answer would deem this project as too complex and infeasible. The scope would become too broad, considering that we only have one term to complete this project. After we have implemented a program that is fully functioning and tested, we can start to consider the expandability of this system to other departments.

So far we have finished the administrative application of the project. The next phase will be to finish up the students side and display questions and answers. After that we will be working on adding and retrieving from a database for display. We determined that the final product will be finish during the month of April.

# Data Design

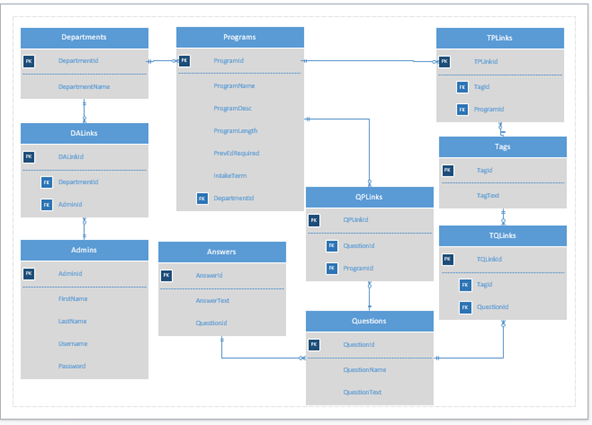
The data design for this Questionnaire will implement the standards used for a web-based survey. Where at the back end data will be inserted, updated and manipulated by SQL through the use of PHP in order to be displayed in a web environment.

MySQL integrated with PHP will be used for the development and production of this questionnaire. Tables will be created according to questionnaires and users will be created to grant access to the existing tables and also to enable the user to create questionnaires that will use fixed tables for specific parameters that pertain to the questionnaire structure.

Requirements are available since the Client’s server uses Tomcat thus making it possible for the team to integrate the back-end MySQL with the website. Data space has to be discussed with the client.

For the ERD diagram we have created a set of tables for department, questions, tags and program. In order to avoid possible many-to-many relationships, we have created linking tables for department-admin (DALinks), question-program(QPLinks), tag-program links (TPLinks) and finally, tag-question links(TQLinks).

Figure 1: ERD Diagram of Program Selection System

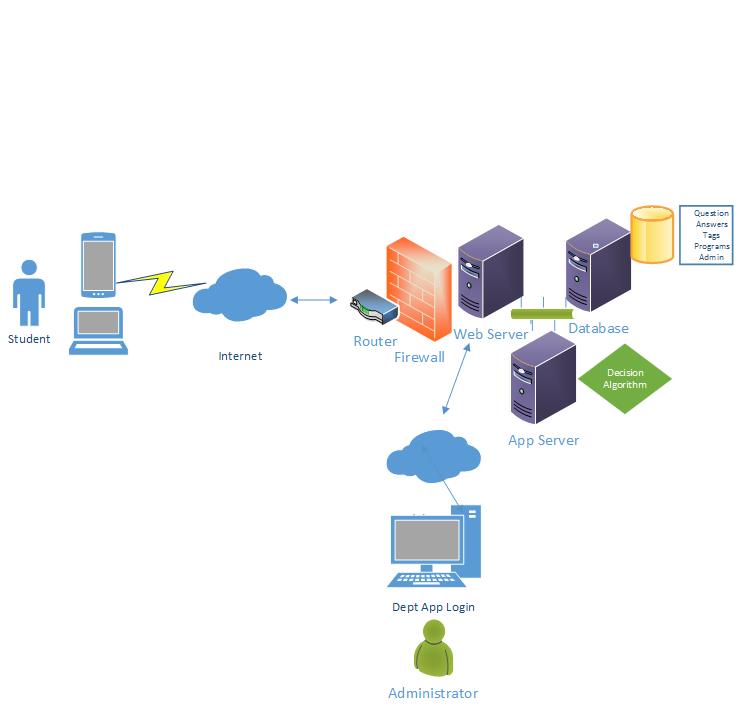


All databases for this project will be created by us and by the user, there is no existing databases that need to be integrated with the ones created by this project.

# Architectural Design

The proposed student program selection system physical architecture comprise of a web-based frontend access by students and a server-based application administrative backend deployed on an existing wordpress content management system of the college. The diagram below depicts the physical architecture of the program selection system.

Figure 2: Physical Architecture of the Program Selection Selection



When the program is up and running the students using their mobile phones or laptops will click on a special button at the department’s program main web page to start the process of getting help in selecting a suitable IT program. The students will be asked to respond to a series of questions and based on their answers that will be stored in the database, the application will automatically recommends the most suitable programs.

The department administrator will have a login access to the application on the existing wordpress dashboard. The administrator is required to upload a series of questions, tags and all the available programs. The PHP application will store the information in the existing wordpress database with a additional tables catered for the application. The application will include a ‘smart’ decision-making algorithm that will use the answers and tags to determine the most appropriate program for the student. The result will be automatically displayed to the student immediately after he has completed all the questions.

The overall application solution will involve programming with with PHP, JQuery, Javascript and CSS, WordPress CMS using existing web server and database resources. The project will adopt an agile adaptive development strategy where there will be regular engagements with the client and a series of sprint events to iteratively ‘facilitates and embraces changes’ (Subrahmanya & Lakshmanan, 2012).

# Program Design

This section shall describe the functionality that will be present within our system upon its completion. The languages which will be used to achieve these requirements are PHP, SQL, HTML, and Javascript. During development and testing we will be monitoring how information is sent into the program and the database. We will do this to make sure that all information is validated and cleaned as to protect from potential SQL injection, XSS, and other web based attacks.

1. Cross Reference Table - list functional requirements and identify associated use case and sequence diagrams.

**Priority scale:**  1 (optional) - 5 (completely necessary)

|  |  |
| --- | --- |
| **SWF1** | **Administrator Creation and Deletion** |
| **Functional Requirement**  **UC1**  **S1** | The system shall allow administrators which have been given additional privileges to add or remove new administrators. |
| **Description** | Administrators which have been granted special privileges from the program's super user(s) will been given the option to create a new admin. This new admin will be added to one or more departments with the ability to modify aspects of that department which relate to the rest of the system. Tying into this admins may be added to more departments after creation  Administrator which have been granted special privileges from the program’s super user(s) will be given the option to remove admins from the system. When an admin has been removed from the system they will no longer have to ability to login or make anymore changes to the program. Admins which have been assigned to more than one department may be removed from one department thereby losing all privileges relating to that department but will still retain their administrative privileges to all other departments to which they have been assigned. When an admin is completely removed from the system or simply removed from a department any changes which they have made will remain intact. |
| **Priority** | 5 |

|  |  |
| --- | --- |
| **SWF2** | **Tag Creation and Deletion** |
| **Functional Requirement**  **UC2**  **S1** | The system shall allow the creation and deletion of association tags |
| **Description** | Administrators shall have the option to add a tag. Administrators shall provide text into a field that will be saved to a database when the Administrator commits their changes by clicking the add button. Tags will be used later on in the question and answer creation phase to associate questions with their related program.  Administrators shall have the option to delete a tag. Administrators shall be able to view what tags are associated with what programs and remove them by selecting them and clicking the remove button. |
| **Priority** | 4 |

|  |  |
| --- | --- |
| **SWF3** | **Question Creation and Deletion** |
| **Functional Requirement**  **UC3**  **S1** | The system shall allow administrators to add and remove questions to and from the question bank. |
| **Description** | Administrators shall have the option to add new questions and provide answers. Administrators will be provided with text fields, where they can enter questions and choose what program they are related to. After the administrator provides their new question they will have to provide potential answers in text fields directly below the question field. The administrator can then save their questions to a database by clicking the add button.  Administrators shall have the option to delete current questions. All programs within a department will be displayed after the login screen. Administrators can select any program and have the questions pertaining to that program displayed. The administrator shall have the option to simply select any question and delete it by clicking the remove button. |
| **Priority** | 4 |

|  |  |
| --- | --- |
| **SWF4** | **Find Programs from answers** |
| **Functional Requirement** | The system shall be able to find programs based on the answers provided by the user. |
| **Description** | After the administrator provides their new question and answers, they will choose a tag which will relate a question to a specific program. The tag bridges the questions and their answers to their associated programs. This allows the system to gather the results from the user and provide them with a suitable program based on their answers to the question. |
| **Priority** | 5 |

|  |  |
| --- | --- |
| **SWF5** | **Gather questions based on program** |
| **Functional Requirement** | The system shall automatically gather questions based on their linked programs. |
| **Description** | If an administrator wishes to view any questions, they will have to view them by their related programs. The system will be able to determine what department each administrator resides in, after they logged in. Once they have logged in they will be presented with all programs within their department. By clicking on any of the programs they will be able to view all the questions related to that program. |
| **Priority** | 3 |

2) The Use Cases used in the analysis and design are listed as follows:

UC1

**Name:** Creation and Deletion of Administrators

**Actor:** Administrator

**Type:** Create and Delete Administrators

**Description**:

* + 1. Administrators which have been granted special privileges from the program's super user(s) will been given the option to create a new admin.
    2. This new admin will be added to one or more departments with the ability to modify aspects of that department which relate to the rest of the system.
    3. Tying into this admins may be added to more departments after creation
    4. Administrator which have been granted special privileges from the program’s super user(s) will be given the option to remove admins from the system.
    5. When an admin has been removed from the system they will no longer have to ability to login or make anymore changes to the program.
    6. Admins which have been assigned to more than one department may be removed from one department thereby losing all privileges relating to that department but will still retain their administrative privileges to all other departments to which they have been assigned.
    7. When an admin is completely removed from the system or simply removed from a department any changes which they have made will remain intact.

UC2

**Name:** Tag Creation and Deletion

**Actor:** Administrator

**Type:** Create and Delete Tags

**Description**:

* + 1. Administrators shall have the option to add a tag.
    2. Administrators shall provide text into a field that will be saved to a database when the Administrator commits their changes by clicking the add button.
    3. Tags will be used later on in the question and answer creation phase to associate questions with their related program.
    4. Administrators shall have the option to delete a tag.
    5. Administrators shall be able to view what tags are associated with what programs and remove them by selecting them and clicking the remove button.

UC3

**Name**: Question Creation and Deletion

**Actor**: Administrator

**Type**: Create and Delete Questions

**Description:**

i) Administrators shall have the option to add a question/

ii) Administrators shall write a questions which will be associated both with a program and with a set of answers.

iii) Questions will be used to determine a suitable program of study for the users of this system.

iv) Administrators shall have the option to delete a question

v) Administrators shall be able to remove questions from the database which will also remove all answers which were associated with that question.

UC4

**Name:** Find Program form Answers

**Actor:** Administrator

**Type:** Find Program form Answers

**Description**:

* + 1. After the administrator provides their new question and answers, they will choose a tag which will relate a question to a specific program.
    2. The tag bridges the questions and their answers to their associated programs.
    3. This allows the system to gather the results from the user and provide them with a suitable program based on their answers to the question.

UC5

**Name:** Questions Based on Program

**Actor:** Administrator

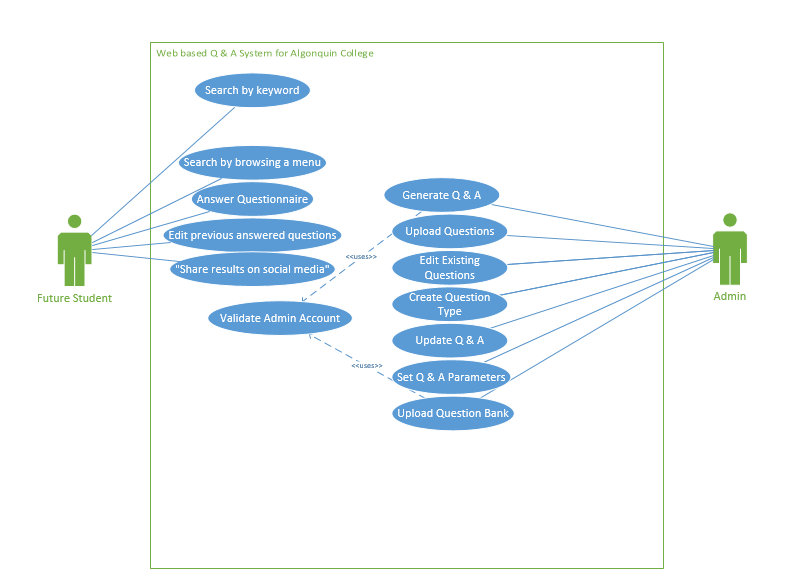
**Type:** Gather Questions based on Program

**Description**:

* + 1. If an administrator wishes to view any questions, they will have to view them by their related programs.
    2. The system will be able to determine what department each administrator resides in, after they logged in.
    3. Once they have logged in they will be presented with all programs within their department.
    4. By clicking on any of the programs they will be able to view all the questions related to that program.

The Use Case Diagram below summarises Administrator and Future Student actions performed within the boundary of the system.

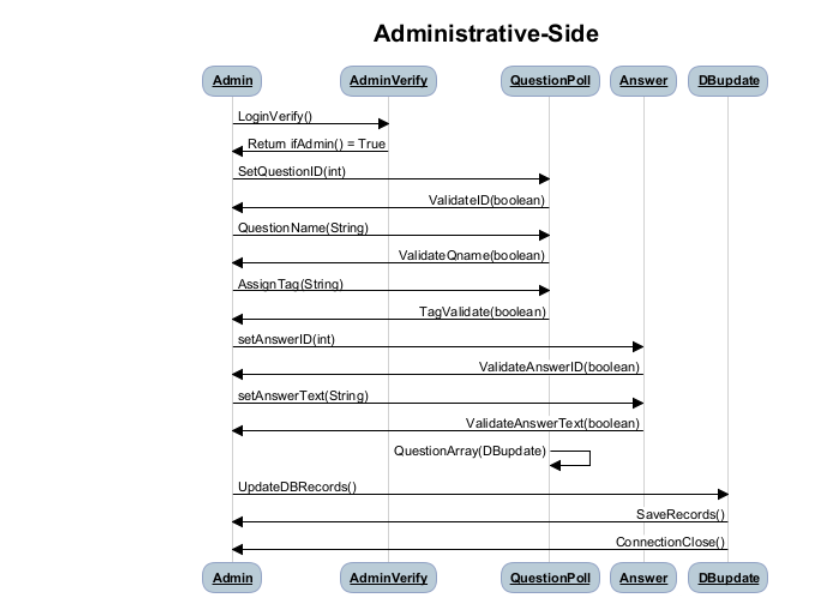
Figure 3: Use Case Diagram



3) Behavioural descriptions:

The sequence diagram is a sample example describing the general overview of the process behind what an administrator would do, the options are limited enough to have this diagram display what process is used. The administrator logs in and is authenticated, then they are prompted with adding question to pool (Removing too) or the option to quit the application

Figure 4: Sequence Diagram



# Conclusion

Equipped with the dedication of our team members along side of their ability to be extremely proficient within the chosen languages we have allowed the idea at hand of creating these helpful features come to life. Tee-four is one hundred percent dedicated to developing a flexible but sturdy back end as well as an intuitive and simple front end application that has the ability to achieve the goal that was locked inside our clients mind. Our project hopes to give upcoming Algonquin students the benefits of easier program selection, in a functional and intuitive manner. This application will also help meet the needs of the administrative staff to be able to enter any new programs upcoming into the college without the worry or requirement of having a computer science degree. Given the information we have available to us at this point we feel that we will be able to meet the requirements of our client with minimal conflicts.

# Recommendations

Tee-four understands the challenges of the current layout and once we currently have a few more specific recommendations to address so we are able to proceed with development of this required framework. These recommendations at a minimum will include the following:

* Discussion of website constraints whether that may be related to word-press, space allocation or compatibility with current plug-ins in use at Algonquin College.
* Further development of Logic of this new framework
* Focusing on WordPress as our Platform
* Further development of Administration UI and looking at meeting clients requirements for this.

# References

Feasibility- Project Management and The inception Phase. Object Oriented Analysis and Design with the Unified Process. Satzinger, Jackson, Burd. Texidium.com. Pg-96-99.

Feasibility- Algonquin College. Sustainable Algonquin

<http://www.algonquincollege.com/sustainability>

Subrahmanya, S. V., & Lakshmanan, G. (2012). *Next Generation IT Architecture.* Delhi, India: Pearson.

# Appendices

* Appendix A - Project Gantt chart

