QsARC:
An accreditation Software for UTB

Software Requirements Specification

Version 1.0

October 30, 2014

Blanca J Garcia
Lead Software Engineer

Prepared for
Software Engineering
Instructor: MK Quweider, Ph.D.
Fall 2014

Revision History

Date	Description	Author	Comments
10/30/2014	Version 1.0	Blanca J Garcia	Initial draft

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	Blanca J Garcia	Lead Software Eng.	10/30/2014

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this Software Requirements Specification document is to provide all of the requirements for the QsARC software. It will describe the purpose and features of the application, the interfaces of the application and what the application will do. This document is intended for both the customer and for the software engineers.

1.2 Scope

The QsARC software system will be an application for accreditation. This system will be designed to automate the process of accreditation, which will otherwise take longer and require a more complex process.
More specifically, the software will consist of two parts; the first one will be a user interface to allow for data entry and a second one will provide a simple GUI interface to allow the user to navigate all the way from the University level to the program level, subsequently permitting the generation of the accreditation report. The system will also comprise a relational database to hold all the necessary data.
The software will be distributed at the department level and the administrator or secretary in charge of each department will be responsible for their data entry.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
SRS	Software Requirement Specifications
QsARC	Academic Repository for Curriculum Accreditation
GUI	graphical user interface
X-Ref tables	A cross reference table is a database table that links records together

1.4 References

SRS_SoftwareRequirmentSpecifications_Template. Based upon the IEEE Guide to Software Requirements Specification (ANSI/IEEE Std. 830-1984).

1.5 Overview

The SRS contains the details of what is required for the QsARC application. It is organized from general to specific. It provides all the detailed functionality and characteristics of the software.

2. General Description

2.1 Product Perspective

The system active users are the administrators of each department. The system will be dependent of a database.

2.2 Product Functions

- Database access to allow users to enter/edit all the information about their department
    - Read
    - Insert
    - Update
    - Delete
- The program will be used at a department level, so the University, the Colleges and the courses will already be available to the users to begin data entry of their department
    - Users will enter data in a form and should create their department under their corresponding College
- Users will be able to navigate through the whole hierarchy to:
    - Enter the programs offered in their department
    - Generate reports
- Reports will be generated in PDF and users will be able to:
    - View report in the application
    - Email report

2.3 User Characteristics

The user is expected to be computer literate and to have the sufficient knowledge of the accreditation process as well as how the courses relate to the learning outcomes.

2.4 General Constraints

The QsARC software system shall be implemented in the Java language using the NetBeans IDE with Git/GitHub for Windows (Distributed Version Control System (VCS)).

2.5 Assumptions and Dependencies

We make the assumption that a server will be available to store the database.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 Software Interfaces

QsARC database:

- The database shall contain all the tables about the University's Colleges, departments and courses as well as all necessary X-Ref tables.
- The software will have access to the database to:
    - Read
    - Insert
    - Update
    - Delete

3.1.2 Communications Interfaces

E-mail provider to send reports.

3.2 Functional Requirements

3.2.1 Data Entry

3.2.1.1 Introduction

Every department shall be able to enter their own information under their corresponding College. A TreeView will be displayed for users to be able to navigate through the hierarchy.
A form/s will be provided to input all information. Users will be able to add new items under their department as well as updating the existing ones.

3.2.1.2 Inputs

- Enter the mission and the vision of the College
- Enter the mission and the vision of the department
- Enter the Department faculty information
    - Number of faculty
    - Personal/academic Information of each faculty
    - CV file for each faculty
- Enter the number of programs offered by the department
- For each Program
    - Program information
        - Name of the program
        - Degree offered
        - Labs
        - Equipment
    - Student statistics
        - Number of students in the program
        - Rank of the students
        - Gender of the students
        - Ethnic make-up
        - Ages
        - Etc.
    - Student learning outcomes
    - POS- Program of study
        - Administrator of Program
        - Course-to-learning outcomes map
        - Courses required for the program
        - For each course:
            - Syllabus for each course
            - Course learning outcomes
            - Collected artifact for each outcome

3.2.1.3 Processing

A save button will be available to save to the database.

3.2.1.4 Outputs

The TreeView will be refreshed to reflect the changes.

3.2.1.5 Error Handling

No duplicates will be allowed for Departments or Programs

3.2.2 Generate Report

3.2.2.1 Introduction
Users will be able to generate, view and e-mail reports. They will use the same TreeView to navigate through the hierarchy to obtain the desired report.
3.2.2.2 Inputs
ID of the selected TreeView node as well as the node Type.
3.2.2.3 Processing
Program will then communicate with the database to pass both parameters to a “select” query/stored procedure.
3.2.2.4 Outputs
The result of the query/stored procedure. All information regarding the selected TreeView node.
3.2.2.5 Error Handling
Make sure that the query/stored procedure does not time out.

3.4 Classes / Objects

- College
- Department
- Program
- Course

3.5 Non-Functional Requirements

3.5.1 Performance

Queries to the QsARC database should take less than 10 seconds to run.

3.5.2 Reliability

The QsARC system shall experience no more than two unhandled errors per month.

3.5.3 Availability

The system shall support 99% availability.

3.5.4 Security

Inconclusive

3.5.5 Maintainability

The average time to repair a defect in the system shall be no greater than 8 person hours.

3.5.6 Portability

Since the system will be created in java, it will be able to run in both Windows and Linux operating systems.

3.7 Design Constraints

The QsARC software system shall be implemented in the Java language using the NetBeans IDE with Git/GitHub for Windows (Distributed Version Control System (VCS)).

3.8 Logical Database Requirements

Research is needed as to what database management system will be used. There will have to be foreign keys and XRef tables.

3.9 Other Requirements

Research is needed as to what e-mail provider to use.