QsARC: Academic Repository for Curriculum Accreditation

Software Requirements Specification

Version 1.0

October 29, 2014

Heather R. Sanders

Lead Software Engineer

Prepared for

Software Engineering

Instructor: MK Quweider, Ph.D.

Fall 2014

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 10/29/2014 | Version 1.0 | Heather R. Sanders | First revision |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Heather R. Sanders | Lead Software Eng. |  |
|  | MK Quweider, Ph.D. | Instructor, COSC 5346 |  |
|  |  |  |  |

**Table of Contents**

Revision History ii

Document Approval ii

1. Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 1

1.5 Overview 1

2. General Description 1

2.1 Product Perspective 1

2.1.1 System Interfaces 2

2.1.2 User Interfaces 2

2.1.3 Hardware Interfaces 2

2.1.4 Software Interfaces 2

2.1.5 Communications Interfaces 2

2.1.6 Memory Constraints 2

2.2 Product Functions 2

2.3 User Characteristics 3

2.4 General Constraints 3

2.5 Assumptions and Dependencies 3

3. Specific Requirements 3

3.1 External Interface Requirements 3

3.2 Functional Requirements 4

3.2.1 New Accreditation Process 4

3.2.2 Continue Accreditation Process 4

3.2.3 Create PDF Report 5

3.3 Use Cases 5

3.3.1 Create 5

3.3.2 Access/Modify Existing 6

3.3.3 Create Report 6

3.4 Classes / Objects 7

3.5 Non-Functional Requirements 7

3.5.1 Performance 7

3.5.2 Reliability 7

3.5.3 Availability 7

3.5.4 Security 7

3.5.5 Maintainability 7

3.5.6 Portability 7

3.6 Inverse Requirements 7

3.7 Design Constraints 7

3.8 Logical Database Requirements 7

3.9 Other Requirements 8

4. Analysis Models 8

4.1 Sequence Diagrams 8

4.3 Data Flow Diagrams (DFD) 8

4.2 State-Transition Diagrams (STD) 8

5. Change Management Process 8

# 1. Introduction

## 1.1 Purpose

The intention of this Software Requirements Specification is to provide detailed information involving the QsARC application. Furthermore, details of the interfaces, constraints, operations, functions, performance, and attributes for the QsARC application can be found in this document.

This Software Requirements Specification will serve as a manual for the design and implementation of the application. Through this document, clarification of the application will be made for the customer.

The document’s intended audience is the client, UTRGV, and the software engineer implementing the QsARC application.

## 1.2 Scope

The scope of this document will cover the internals that make up the QsARC application. This application will consist of a graphical user interface to gather accreditation information about the university from the user and output an organized report in the PDF format.

The interface will act as a wizard tool in order to step the user through manageable levels of information input. Once all data has been gathered, the user will be able to create a PDF report with a title page, table of contents, university information, college information, department information, and an appendix of all the courses listed.

The goal of the QsARC application is to automate the accreditation process, can be used by every department in the university, and output a customized report based on the data given through e-mail.

## 1.3 Definitions, Acronyms, and Abbreviations

To be updated at a later time.

## 1.4 References

## 1.5 Overview

The QsARC Software Requirements Specification will go into detail about external interface requirements, provide sequence diagrams, performance requirements, and design constraints.

# 2. General Description

## 2.1 Product Perspective

The application could be considered a data management system in that it gathers related data about the university and outputs a report in an easy to follow format. It will store the data externally from the application in a cloud database making it dependent to an internet connection.

### 2.1.1 System Interfaces

### 2.1.2 User Interfaces

QsARC

University Staff/Faculty

PDF Accreditation Report

Accreditation Data

QsARC will be a GUI based application. The initial interface will allow the user to initiate an accreditation report wizard which will break down into four sub-interfaces. The sub-interfaces are university information, college information, department information, and program information.

### 2.1.3 Hardware Interfaces

QsARC will require common hardware supplied with a typical PC from the past five years including a network interface.

### 2.1.4 Software Interfaces

The programming language used to create the QsARC application is Java, and therefore, is platform-independent.

### 2.1.5 Communications Interfaces

The accreditation data will be stored in an online database. Internet access is necessary for the QsARC application to run.

### 2.1.6 Memory Constraints

The application itself is not expected to be larger than 2 MB.

## 2.2 Product Functions

The QsARC application will allow the user to enter specific data necessary for accreditation. The functions necessary are:

* Database connection through the internet
* Store and retrieve data in the database, including files
* Provide links in the GUI of completed areas that can be revisited
* Ability to import CSV data into the database
* Display university, college, department, or program information and allow the user to add, change, or delete information
* Produce a PDF report and send it as an attachment through an electronic message

## 2.3 User Characteristics

Users of the QsARC application should have fundamental computer skills such as file uploading and data entry skills.

## 2.4 General Constraints

No known constraints.

## 2.5 Assumptions and Dependencies

Internet access is assumed, if it becomes required that the application must work with or without internet access, then the data storage method would need to be revised.

# 3. Specific Requirements

## 3.1 External Interface Requirements

The online database and the data and file input from the user are the only external interfaces for the application.

Attributes of the online database:

1. Object name: QsARC Database
2. Object purpose: To store related data about the university, it’s colleges, departments, programs, courses, faculty, and students. Through the application, the user will be able to add, change, and delete information in the database.
3. Source of input: The application will add, change, and delete data stored in the database through SQL statements and queries based on user interactions.
4. Destination of output: The application will retrieve data from the database and display information to the monitor to provide the user feedback. A PDF report will be created and sent via electronic mail at the user’s request.
5. Timing: Data addition/modification/retrieval/deletion is expected to not take more than 5 seconds for non-file data. File addition/retrieval/deletion is expected to vary depending on the file size.
6. Data formats: The majority of the database is text-based. However, a few fields are dedicated for image and PDF files.

## 3.2 Functional Requirements

### 3.2.1 New Accreditation Process

#### 3.2.1.1 Introduction

When the application is open, the user is able to start a new accreditation process.

#### 3.2.1.2 Inputs

The application requests data/files from the user in order to store the pertinent data needed to create the accreditation report.

#### 3.2.1.3 Processing

The application sends the data/files from the user to the database through appropriate add/modify/delete SQL statements and queries.

#### 3.2.1.4 Outputs

If the update to the database was successful, a confirmation will be given to the user through the application.

#### 3.2.1.5 Error Handling

Ensuring the correct data types are input by the user will be handled prior to sending the data to the database. If incorrect data types are given, the application will alert the user and allow the user to input the correct data type or cancel the request. The application will not be able to verify if the user is inputting the correct data, however. It is up to the user to understand what the data is they are inputting and whether it is correct.

### 3.2.2 Continue Accreditation Process

#### 3.2.2.1 Introduction

When the application is open, the user is able to click on an accreditation process that has already been created.

#### 3.2.2.2 Inputs

The application requests data/files from the user in order to store the pertinent data needed to create the accreditation report.

#### 3.2.2.3 Processing

The application sends the data/files from the user to the database through appropriate add/modify/delete SQL statements and queries.

#### 3.2.2.4 Outputs

If the update to the database was successful, a confirmation will be given to the user through the application.

#### 3.2.2.5 Error Handling

Ensuring the correct data types are input by the user will be handled prior to sending the data to the database. If incorrect data types are given, the application will alert the user and allow the user to input the correct data type or cancel the request. The application will not be able to verify if the user is inputting the correct data, however. It is up to the user to understand what the data is they are inputting and whether it is correct.3.3 Use Cases

### 3.2.3 Create PDF Report

#### 3.2.3.1 Introduction

When the user has finished adding the accreditation data, the application will generate a PDF report and send it to the user given e-mail address.

#### 3.2.3.2 Inputs

The application will request an e-mail address to send the report to from the user.

#### 3.2.3.3 Processing

The application will generate the PDF and send the e-mail.

#### 3.2.3.4 Outputs

The accreditation report will output as a PDF.

#### 3.2.3.5 Error Handling

There will need to be a check to ensure the user input an e-mail address in the correct format. However, it is up to the user to verify the e-mail address is correct.

## 3.3 Use Cases

### 3.3.1 Create

#### 3.3.1.1

Application begins a new accreditation process by requesting university information

#### 3.3.1.2

User enters the requested information about the university.

#### 3.3.1.3

Application confirms to the user the information was added successfully.

#### 3.3.1.4

Application requests user to enter information for each of the colleges at the university.

#### 3.3.1.5

User enters the requested information for each of the colleges at the university.

#### 3.3.1.6

Application confirms to the user the information was added successfully.

#### 3.3.1.7

Application requests user to enter information for each department in the colleges.

#### 3.3.1.8

User enters the requested information for each department in the colleges.

#### 3.3.1.9

Application confirms to the user the information was added successfully.

#### 3.3.1.10

Application requests user to enter information for each program offered by the departments.

#### 3.3.1.11

User enters the requested information for each program offered by the departments.

#### 3.3.1.12

Application confirms to the user the information was added successfully.

#### 3.3.1.13

Application requests user to enter the e-mail address to send the accreditation report to.

#### 3.3.1.14

User enters the desired e-mail address to receive the report.

#### 3.3.1.15

Application generates the accreditation process report in PDF format and sends to e-mail address entered by user.

#### 3.3.1.16

Application takes the user back to the initial screen

### 3.3.2 Access/Modify Existing

#### 3.3.2.1

Application lists existing accreditation processes for the user to choose.

#### 3.3.2.2

User selects an existing accreditation process they would like to access or modify.

#### 3.3.2.3

Application asks the user if they would like to modify the university, college, department, or program information.

#### 3.3.2.4

User clicks on selection.

#### 3.3.2.5

Application requests user to enter information for the desired area.

#### 3.3.2.6

User enters the requested information.

#### 3.3.2.7

Application confirms to the user the information was added/modified successfully.

### 3.3.3 Create Report

#### 3.3.3.1

Application offers to generate the accreditation report.

#### 3.3.3.2

User enters the e-mail address to send the report to.

#### 3.3.3.3

Application generates the accreditation report in PDF format and sends to the input e-mail address.

#### 3.3.3.4

Application confirms the e-mail was sent.

#### 3.3.3.5

Application returns to initial screen.

## 3.4 Classes / Objects

This section is under construction.

## 3.5 Non-Functional Requirements

### 3.5.1 Performance

The QsARC application shall generate the accreditation process report in PDF format in less than 3 minutes.

### 3.5.2 Reliability

The QsARC application shall encounter no more than two unresponsive faults per month.

### 3.5.3 Availability

The QsARC application shall be available 98% of the time.

### 3.5.4 Security

The QsARC application does not currently use security features.

### 3.5.5 Maintainability

The average time to repair a defect shall be no greater than 12 hours.

### 3.5.6 Portability

The QsARC application will be platform-independent.

## 3.6 Inverse Requirements

The application will not verify the correctness of the data entered by the user. It is assumed the user is inputting the correct information requested by the application.

## 3.7 Design Constraints

No known design constraints.

## 3.8 Logical Database Requirements

The database used by the application is hosted by Amazon Web Services Free Tier. The AWS Free Tier provides 20 GB of database storage and 20 GB of database backup storage. AWS offers a Virtual Private Cloud (VPC) service for security purposes, but it will not be used in this version of the application. The database is capable of storing text and files.

## 3.9 Other Requirements

None.

# 4. Analysis Models

## 4.1 Sequence Diagrams

To be determined.

## 4.3 Data Flow Diagrams (DFD)

To be determined.

## 4.2 State-Transition Diagrams (STD)

To be determined.

# 5. Change Management Process

As the project becomes more clear or if requirements change, the Software Requirements Specification will be updated by the Lead Engineer.