An Accreditation Software for UTRGV

Software Requirements Specification

Version 1.0

October 30, 2014

Miguel Garza

Hulices Lopez(Lead Software Engineer)

Prepared for

Software Engineering

Instructor: MK Quweider, Ph.D.

Fall 2014

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 10-27-14 | Version 1 | Miguel Garza & Hulices Lopez | Need to complete sections. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# **Document Approval**

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Hulices Lopez | Lead Software Eng. |  |
|  | Dr. Mahmoud Quweider | Instructor, COSC 4346 |  |
|  |  |  |  |

**Table of Contents**

**REVISION HISTORY**

**DOCUMENT APPROVAL**

**1. INTRODUCTION**

1.1 Purpose**……………………………………………………………………………………………....…....5**

1.2 Scope**……………………………………………………………………………………………….……..5**

1.3 Definitions, Acronyms, and Abbreviations**…………………………………………………........................5**

1.4 References**…………………………………………………………………………………………….…..6**

1.5 Overview**…………………………………………………………………………………….......................6**

**2. GENERAL DESCRIPTION**

2.1 Product Perspective**…………………………………………………………………………………….....6**

2.2 Product Functions**………………………………………………………………………………………....6,7**

2.3 User Characteristics**………………………………………………………………………………...……7**

2.4 General Constraints**………………………………………………………………………………………7**

2.5 Assumptions and Dependencies**………………………………………………………………………..……7**

**3. SPECIFIC REQUIREMENTS**

3.1 External Interface Requirements

3.1.1 User Interfaces**…………………………………………………………………………………….…**7

3.1.2 Hardware Interfaces**……………………………………………………………………………….…**7

3.1.3 Software Interfaces**………………………………………………………………………………..…**8

3.1.4 Communications Interfaces**………………………………………………………………………....**8

3.2 Functional Requirements

3.2.1 <Functional Requirement or Feature #1>**.................................................................................................**8

3.2.2 <Functional Requirement or Feature #2>**.................................................................................................**8

3.3 Use Cases

3.3.1 Use Case #1**………………………………………………………………………………………......**8

3.3.2 Use Case #2**……………………………………………………………………………………….….**8

3.4 Classes / Objects

3.4.1 <Class / Object #1>**..............................................................................................................................**8

3.4.2 <Class / Object #2>**..............................................................................................................................**8

3.5 Non-Functional Requirements

3.5.1 Performance**………………………………………………………………………………….…….…8,9**

3.5.2 Reliability**………………………………………………………………………………………....…..9**

3.5.3 Availability**………………………………………………………………………………………....…9**

3.5.4 Security**……………………………………………………………………………………………..…9**

3.5.5 Maintainability**………………………………………………………………………………...........…9**

3.5.6 Portability**……………………………………………………………………………………….......…9**

3.6 Inverse Requirements**………………………………………………………………………………....……9**

3.7 Design Constraints**………………………………………………………………………………...…….…9**

3.8 Logical Database Requirements**……………………………………………………………………...….…9**

3.9 Other Requirements**………………………………………………………………………………….…..…9**

**4. ANALYSIS MODELS………………………………………………………………………………….…....….10**

4.1 Sequence Diagrams**……………………………………………………………………………………….…10**

4.3 Data Flow Diagrams (DFD)**……………………………………………………………………………...…10**

4.2 State-Transition Diagrams (STD)**……………………………………………………………………….…10**

**5. CHANGE MANAGEMENT PROCESS……………………………………………………………………....10**

**A. APPENDICES**

A.1 Appendix 1………………………………………………………………………………………………10….

A.2 Appendix 2……………………………………………………………………………………………….10

# **1. Introduction**

## **1.1 Purpose**

The purpose of this Software Requirements Specification is to give a detailed description of Accreditation Software for UTRGV. It will cover the features, interfaces, constraints, dependencies, functionality, and attributes. This document will be used for the implementation of the software and will help the customer understand the description and the direction it is headed. This software will be designed for UTRGV and the engineering team responsible for creating it.

## **1.2 Scope**

The software described in this document is An Accreditation Software for UTRGV. The application will allow users to store accreditation data and generate given reports, such as web, pdf or text. It will also display University information, department information and accommodates network connectivity for a final report to be submitted via email or website. This software will be distributed to UTRGV administrators. Instead of administrator having to add accreditation information every year, this app would allow for automation of the process, and would work for every department. This application will be user friendly and would save the university valuable time.

## **1.3 Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| GUI | Graphical User Interface |
| IDE | Interactive Development Environment |
| Database; DB | An organized collection of data |
| CV | Curriculum Vitae |
| QsARC | ASK DR Q ABOUT THIS |
| UML | Unified Modeling Language |
| UT-RGV | University of Texas at Rio Grande Valley |
| PDF | Portable Document Format |
| MB | Megabytes |
| GB | Gigabytes |

## **1.4 References**

None at this time.

## **1.5 Overview**

This SRS is organized into two sections. The first is the Overall Description and is intended for the customers. The Overall Description provides the high-level requirements and is an insight into the the Specific Requirements, which is the next section. The Specific Requirements details the functions of the software and is intended for the developer. Lastly, we have the Analysis Models which will describe the basic UML used and provides a logical model of the software.

# **2. General Description**

## **2.1 Product Perspective**

This product is strictly for web portal where the administrator will have read and write privileges and the student will only have read privileges. Also, since this product is data-based driven we must also create a database. The hard drive space designated for the database has yet to be determined.

## **2.2 Product Functions**

The main functions of the software are to provide administrators with an easy-to-use GUI to automate the process of accreditation. The administrator(s) will be able to do the following:

* Store Accreditation Data and Generate Requested reports
* View the mission and the vision of the university
* View number and name of each college
* View all the courses offered at the university
* For each department:
  + View the mission and vision of the department.
  + View number of faculty
  + View personal/academic information of each faculty
  + View CV file for each faculty
* View number of programs offered by the department
* For each program:
  + View name of the program
  + View administrator of the program
  + View courses-to-learning outcomes map
  + View courses required for the program
  + View degrees offered
  + View labs
  + View equipment
* For each course:
  + View syllabus
  + Course learning outcomes
* View student statistics:
  + View number of students in the program
  + View rank of the student(s)
  + View gender of the student(s)
  + View ethnic make-up
  + View age(s) of student(s)

## **2.3 User Characteristics**

The general characteristics of the users of this intended product are those that have finished their compulsory education levels, with a basic understanding and experience with computers or other technical devices. Some training might be needed

## **2.4 General Constraints**

Time, resources.

## **2.5 Assumptions and Dependencies**

We make the assumption that the memory available in the dedicated server is largely sufficient for all the data to store.

# **3. Specific Requirements**

## **3.1 External Interface Requirements**

The external interface would be the database. Here is an overview:

* UTRGV database
* Purpose: To contain all the university and department information.
* Data: The information in the database will be text, documents.

### **3.1.1 User Interfaces**

* For university employees such as administrators or advisors:
  + Update data such as university and department information.
  + Upload faculty CV.
  + Generate report(s) in either PDF, Web, or text format.
* For students:
  + Only have the ability to view data; no modifying or manipulation.
  + Have the ability to print generated report in either PDF, Web, or text format.

### **3.1.2 Hardware Interfaces**

No designated hardware, as the desktop based software may be run on any system.

### **3.1.3 Software Interfaces**

The software grabs information where it then stores to a database. Information is pulled from database as html file and converted into xml then to pdf,text or web.

### **3.1.4 Communications Interfaces**

## **3.2 Functional Requirements**

*This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.*

### **3.2.1 <Functional Requirement or Feature #1>**

3.2.1.1 Introduction

3.2.1.2 Inputs

3.2.1.3 Processing

3.2.1.4 Outputs

3.2.1.5 Error Handling

### **3.2.2 <Functional Requirement or Feature #2>**

…

## **3.3 Use Cases**

### **3.3.1 Use Case #1**

### **3.3.2 Use Case #2**

…

## **3.4 Classes / Objects**

### **3.4.1 <Class / Object #1>**

3.4.1.1 Attributes

3.4.1.2 Functions

<Reference to functional requirements and/or use cases>

### **3.4.2 <Class / Object #2>**

…

## **3.5 Non-Functional Requirements**

*Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).*

### **3.5.1 Performance**

Since this is not a real-time application, perfomance will not be a major issue.

The main functions described previously should take more than 5 seconds.

The database must support simultaneous requests and must have sufficient bandwidth to quickly deliver user generated reports

### **3.5.2 Reliability**

The customer imposes no reliability attribute for the moment.

### **3.5.3 Availability**

The software and the server should be available to use at any moment.

### **3.5.4 Security**

No security to be implemented for the moment.

### **3.5.5 Maintainability**

The administrators should be able to make a backup of the content of the server and database.

### **3.5.6 Portability**

The software will be platform-independent.

## **3.6 Inverse Requirements**

The software shall not allow students to modify and manipulate any data. The software shall not be able to work offline; internet connectivity is a must. The software shall not allow the administrator to edit faculty CV.

## **3.7 Design Constraints**

* Software will run using a database and memory usage has yet to be determined.
* The amount of Operating System memory occupied by the application.
  + MUST: yet to be determined.
  + PLAN: yet to be determined.
  + WISH: yet to be determined.

## **3.8 Logical Database Requirements**

Database will be used and have html. Storage capabilities yet to be determined. Database will store information and only administrators will have access to modify the information as new faculty are added.

## **3.9 Other Requirements**

Convert HTML files to text, pdf, or web files.

# **4. Analysis Models**

*List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS’s requirements.*

## **4.1 Sequence Diagrams**

## **4.3 Data Flow Diagrams (DFD)**

## **4.2 State-Transition Diagrams (STD)**

# **5. Change Management Process**

*Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.*

# **A. Appendices**

*Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS’s overall set of requirements.*

*Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.*

## **A.1 Appendix 1**

## **A.2 Appendix 2**