COSC 4346/5346

Software Requirements Specification Template

Vaq-Paq

Software Requirements Specification

Version 1.0

Nov 19, 2016

Team Leader

Jessie Reyna

Team Members

Gonzalo Delgado

Adrian Sierra

Kevin Cavazos

Prepared for Software Engineering

Instructor: MK Quweider, Ph.D. Fall 2016

# Revision History

## Date Description Author Comments

|  |  |  |  |
| --- | --- | --- | --- |
| Nov-7-16 | Version 1.0 | Jessie Reyna | First revision |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

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

## Signature Printed Name Title Date

|  |  |  |  |
| --- | --- | --- | --- |
|  | Jessie Reyna | Project Manager | Nov – 19 - 2016 |
|  | MK Quweider | Instructor, CptS 322 |  |
|  |  |  |  |

# Table of Contents

**REVISION HISTORY ...............................................................................................................................................II** **DOCUMENT APPROVAL .......................................................................................................................................II**

1. **INTRODUCTION ................................................................................................................................................... 1**
   1. PURPOSE .............................................................................................................................................................. 1 1.2 SCOPE .................................................................................................................................................................. 1 1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS ................................................................................................ 1 1.4 REFERENCES ........................................................................................................................................................ 1 1.5 OVERVIEW ........................................................................................................................................................... 2
2. **GENERAL DESCRIPTION ................................................................................................................................... 2**
   1. PRODUCT PERSPECTIVE ....................................................................................................................................... 2 2.2 PRODUCT FUNCTIONS .......................................................................................................................................... 2 2.3 USER CHARACTERISTICS ..................................................................................................................................... 2 2.4 GENERAL CONSTRAINTS ...................................................................................................................................... 2 2.5 ASSUMPTIONS AND DEPENDENCIES ..................................................................................................................... 2
3. **SPECIFIC REQUIREMENTS ............................................................................................................................... 2**

## 3.1 EXTERNAL INTERFACE REQUIREMENTS ............................................................................................................... 2

3.1.1 User Interfaces ............................................................................................................................................ 3

3.1.2 Hardware Interfaces ................................................................................................................................... 3

3.1.3 Software Interfaces...................................................................................................................................... 3

3.1.4 Communications Interfaces ......................................................................................................................... 3

## 3.2 FUNCTIONAL REQUIREMENTS .............................................................................................................................. 3

3.2.1 <Functional Requirement or Feature #1> ................................................................................................. 3

3.2.2 <Functional Requirement or Feature #2> ................................................................................................. 4

## 3.3 USE CASES ........................................................................................................................................................... 4

3.3.1 Use Case #1 ................................................................................................................................................ 4

3.3.2 Use Case #2 ................................................................................................................................................ 5

## 3.4 CLASSES / OBJECTS .............................................................................................................................................. 5

3.4.1 <Class / Object #1> .................................................................................................................................... 5

3.4.2 <Class / Object #2> .................................................................................... **Error! Bookmark not defined.**

## 3.5 NON-FUNCTIONAL REQUIREMENTS ..................................................................................................................... 6

3.5.1 Performance ................................................................................................................................................ 6 3.5.2 Reliability .................................................................................................................................................... 6

3.5.3 Availability .................................................................................................................................................. 6 3.5.4 Security ....................................................................................................................................................... 6

3.5.5 Maintainability ............................................................................................................................................ 6

3.5.6 Portability ................................................................................................................................................... 6

3.6 INVERSE REQUIREMENTS ..................................................................................................................................... 6 3.7 DESIGN CONSTRAINTS ......................................................................................................................................... 6 3.8 LOGICAL DATABASE REQUIREMENTS .................................................................................................................. 6 3.9 OTHER REQUIREMENTS ....................................................................................................................................... 6

**4. ANALYSIS MODELS ............................................................................................................................................. 6**

## 4.1 SEQUENCE DIAGRAMS ......................................................................................................................................... 6 4.3 DATA FLOW DIAGRAMS (DFD) ........................................................................................................................... 7 4.2 STATE-TRANSITION DIAGRAMS (STD) ................................................................................................................ 7

**5. CHANGE MANAGEMENT PROCESS ............................................................................................................... 7**

1. **APPENDICES ......................................................................................................................................................... 7**
   1. APPENDIX 1......................................................................................................................................................... 7
   2. APPENDIX 2......................................................................................................................................................... 7

# 1. Introduction

## 1.1 Purpose

The purpose of this Software Requirements Specification SRS is to give a detailed description and use of services and the functionality of the Vaq-Paq, which is a piece of Software for UTRGV. This document is to serve as a manual for the clients, developers, and other stakeholders.

## 1.2 Scope

This piece of Software(s), libraries, and packages are for the Vaq-Pac – including all its services and implementations. The software will provide a GUI (Graphical User Interface) for the clients to interact with and to deploy underlying services the software provides. Services include: file formatting (pdf, html and xml), DBMS (Database Management Service) - for storing said files, University Information (such as courses, program of studies, and transcripts). It will also implement network to provide said software to connect to the DBMS with login access to a personal account. Networking for email .

## 1.3 Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| XML | – Extensible Markup Language |
| TXT | – File to save simple text characters |
| SRS | – Software Requirements Specification |
| PDF | – Portable Document File |
| POS | – Program of Study |
| IDE | – Interactive Development Environment |
| Java | – OOP language used to implement said software |
| JavaFX | – GUI views for Java |
| MySQL | – Local Database |
| HTML | – Hypertext Markup Language |
| GUI | – Graphical User Interface |
| SRS | – Software Requirement Specification. |
| EventHandlers | – Reading user inputs into the software. |
| OS | – Operating System |
| Vaq-Paq | – This Piece of Software |
| DBMS | – Database Management Systems |
| UTRGV | – University of Texas at Rio Grande Valley |
| XSL | – language for expressing style sheets |
| Network | – Transfer of data between Database and Vaq-Paq |

## 1.4 References

Netbeans -<https://netbeans.org/>

MySQL -<https://www.mysql.com/>

Github –<https://www.github.com/>

JavaFX -<http://docs.oracle.com/javase/8/javase-clienttechnologies.htm>

**1.5 Overview**

The sections are divided into 3 categories – General Description, Requirements, and Analysis.

# 2. General Description

## 2.1 Product Perspective

The software will have a friendly GUI that the client interacts with to view courses and POS for UTRGV. It will store the data on a Database and connect with it through a secure network. The data will be course, transcript, POS files which will be in PDF, XML, and HTML formats – these files can be generated on the fly in the software as a service.

## 2.2 Product Functions

* User Friendly GUI
  + Log in securely with username and password
  + View entire list of courses in the Computer Science Program of Study
  + Select courses to view course descriptions and other pertinent information.
  + Enter courses and grades to calculate present and future GPA
  + View department and faculty information
  + Set reminders with time and date
* Database Connection
  + Store XML
  + Temp HTML storage
  + Temp PDF files
  + User Information Storage
  + Confirmation of Username and Password
* Network Connection
  + Email a member of the faculty.
* Services
  + File Conversion
  + File Manipulation
  + Generate an HTML object of all Computer Science Program of Study courses.
  + Generate a PDF object of selected courses with course description and syllabus information.
  + Generate printable, distributable PDF files of the objects.

## 2.3 User Characteristics

The user should be able to read and write Course, Faculty, Transcript information. Also, basic understanding of computer interfaces and hardware – i.e. button controls and mouse w/ keyboard.

## 2.4 General Constraints

* JavaFX for GUI
* MySQL for Database
* XML to hold data
* Github for Version Control
* VaqPack must be developed using the NetBeans IDE.

## 2.5 Assumptions and Dependencies

* VaqPack will run on a system with an operating system that has a compatible Java Virtual Machine and up to date Java Runtime Environment.
* VaqPack will run on or connect to a system with an existing MySQL server.
* In the event of remote MySQL server connectivity, the system on which VaqPack will run has the networking capabilities to connect to said database server.
* User has a UTRGV email with Username and password
* User is familiar with XML and HTML structures

# 3. Specific Requirements

## 3.1 External Interface Requirements

The Database will have an external interface for the maintenance of said database. If the user chooses to view a certain PDF, HTML of XML outside of the software, a browser may be viewed as an external interface, but isn’t necessary.

### 3.1.1 User Interfaces

1. **User Interface:** Each part of the user interface intends to be as user friendly as possible. VaqPack provides a GUI for the user to interface with all the functionality necessary to accomplish the user’s goals in a visual manner. When the user opens VaqPack, if they are a new user, they will be prompted to register a username and password for future logins. Then, the user will be able to view and navigate through different tabs to view course descriptions, faculty and department information, course management, and e-mail.
2. **Admin Interface:** The administrator will be able to log in and manage the database, as well as back-up and restore content, create and manage security configurations, and tune system performance.

### 3.1.2 Hardware Interfaces

* **Windows**
  + Windows 10 (8u51 and above)
  + Windows 8.x (Desktop)
  + Windows 7 SP1
  + RAM: 128 MB
  + Disk space: 124 MB for JRE; 2 MB for Java Update
  + Processor: Minimum Pentium 2 266 MHz processor
* **Mac OS X**
  + Intel-based Mac running Mac OS X 10.8.3+, 10.9+
  + Administrator privileges for installation
  + 64-bit browser
* **Other Specs**
  + Since VaqPack has e-mail abilities, the hardware shall require to connect to the internet via some medium, e.g. Modem, WAN – LAN, Ethernet Cross-Cable.
  + Client Side: Java and the most minimum requirements for the home PC, MAC, and small consumer computers will suffice.
  + Server Side: Sufficient RAM and Storage.
  + Network: Network Connection.

### 3.1.3 Software Interfaces

* Database Server (MySQL) – Operating System (any)
* Client Application – Vaq-Paq GUI.
* Java Runtime – Run program/JAR.

**3.1.4 Communications Interfaces**

Client will be using HTTP/HTTPS and the GUI, but the connectivity will all be achieved within the Vaq-Paq.

## 3.2 Functional Requirements

* User Friendly GUI
  + Software will have all the needed controls to operate the included functions.
  + GUI will have initiate network and database functions w/out without the client noticing.
* Database Connection
  + The DB will have a connection to the Vaq-Paq.
  + The DBMS will be able to modify, insert, and delete records.
  + The DB will have attributes that will make accessing it viable.
* Network Connection
  + The network will connect to the Vaq-Paq.
  + The network will connect to the DB.
  + The network will be secure.
  + Email connections
* Services
  + The file conversion will operate decupled from all other components of the VaqPaq.
  + The file conversion will be fast – 2 to 3 seconds is the limit.
  + The file conversion will convert XML to HTML and HTML to PDF o The file conversion will send the output files into the database for storage.
  + File Manipulation
  + File manipulation will be able to merge documents together.

### 3.2.1 User Friendly GUI

|  |  |
| --- | --- |
| Purpose | To assist the user in handling Events and using internal functions. |
| Inputs | Button clicks and Text inputs. |
| Processing | Use of EventHandlers to bring to the forefront of the GUI the users desired results by |
|  | starting internal functions. |
| Outputs | Starting network transfer to database, converting/merging files, or displaying other GUI components. |
| Error  Handling | Account for chaotic button clicks (threads) and invalid user input (try catch). |

### 3.2.2 Database connection

|  |  |
| --- | --- |
| Purpose | To store persistent data across application runs. |
| Inputs | Records – user info, courses, and POS. |
| Processing | Saving tuples into a table. |
| Outputs | Send info to the GUI. |
| Error  Handling | Tell GUI data cannot be inserted, deleted, or modified (try catch). |

### 3.2.3 Network connection

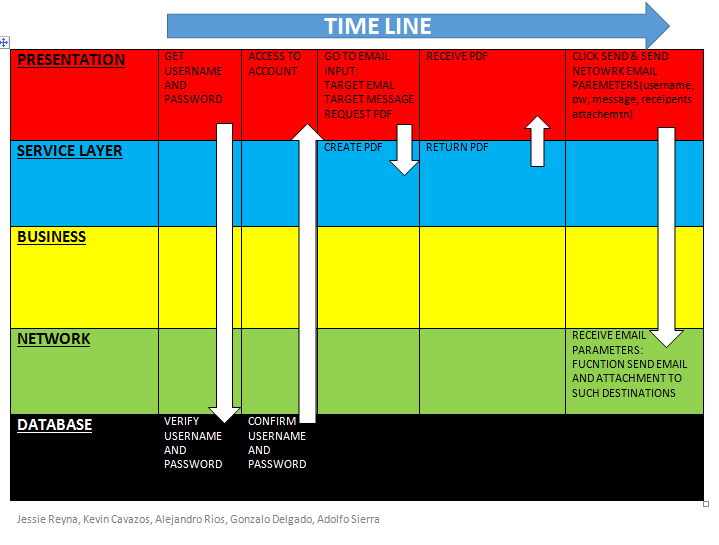
|  |  |
| --- | --- |
| Purpose | To send data between application and DB server. |
| Inputs | Packets. |
| Processing | Through sockets. |
| Outputs | Connection established. |
| Error  Handling | Send error to GUI to be seen be the user (try-catch that the connection was not established, lost or not secure). |

### 3.2.4 Services

|  |  |
| --- | --- |
| Purpose | To convert files for user’s viewing, storage or manipulation. |
| Inputs | Files: XML, XSL, HTML, and PDF |
| Processing | Converts (Factory Libraries) to output desired file. |
| Outputs | Another file in a different format or a merged one. |
| Error  Handling | File cannot be manipulated (try-catch) |

## 3.3 Use Cases

### 3.3.1 Email

* Goal: Allow user to email course selection to desired target with PDF attachment of such course selection. This also includes a Carbon Copy, Subject Line, and Text Message. User’s UTRGV.edu Email smtphost will be utilized for sending email message and attachment.
* Input:
  + Email password,
  + To (who is being)
  + Subject
  + Text message
* Main Scenario: User desired to email Courses to a target audience.
* Pre-Condition: VaqPack must be running.
* 

### 3.3.2 Generate XML

* Input:
* Main Scenario
* Pre-Condition

## 3.4 Classes / Objects

### 3.4.1 cMail

3.4.1.1 Attributes

* private String smtpHost
* private String port

3.4.1.2 Functions

* sendEmail

**3.4.2 XMLGen**

3.4.2.1 Attributes

* private String [] NodeName
* private String [] NodeText
* private ArrayList<String> NodeNameList
* private ArrayList<String> NodeTextList
* private String newfilename;

3.4.2.2 Functions

* private generateWithArraylist
* private generateWithArrays
* public generateXMLWithArrayList
* public generateXMLWithArrays

## 3.5 Non-Functional Requirements

### 3.5.1 Performance

All functions should not take more than a few seconds. Network connections is at the whim of Client’s own internet connection.

**3.5.2 Reliability**

The software should not crash.

### 3.5.3 Availability

The software should be available 24/7 on local host machine and when an internet connection can be connected, it should be able to connect.

**3.5.4 Security**

Secure connection w/ a secure DB server. Email Password is not stored in DB, and should be prompted by GUI.

### 3.5.5 Maintainability

The Admin will oversee maintenance of the server and the application should require little to no maintenance.

### 3.5.6 Portability

Software will run on desktop like machine, but can be ran on other if the other machines have Java.

**3.6 Inverse Requirements**

User may Email from different UTRGV.edu email address.

**3.7 Design Constraints**

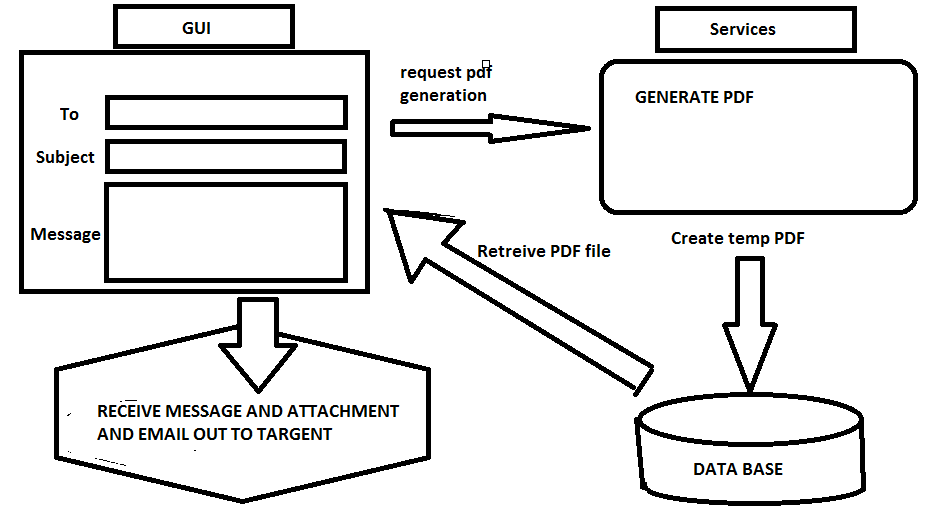
N/A currently

**3.9 Other Requirements**

N/A.

# 4. Analysis Models

## 4.1 Sequence Diagrams



**4.3 Data Flow Diagrams (DFD)**

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

# 5. Change Management Process

Changes can be made by any on the developer teams. But to commit the change to the final SRS, you must seek approval from Instructor.

# 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