Software Requirements Specification Template

Software Engineering

The following annotated template shall be used to complete the Software Requirements Specification (SRS) assignment of WSU-TC CptS 322. The instructor must approve any modifications to the overall structure of this document.

**Template Usage:**

Text contained within angle brackets (‘<’, ‘>’) shall be replaced by your project-specific information and/or details. For example, <Project Name> will be replaced with either ‘Smart Home’ or ‘Sensor Network’.

Italicized text is included to briefly annotate the purpose of each section within this template. This text should not appear in the final version of your submitted SRS.

This cover page is not a part of the final template and should be removed before your SRS is submitted.

**Acknowledgements:**

Sections of this document are based upon the IEEE Guide to Software Requirements Specification (ANSI/IEEE Std. 830-1984). The SRS templates of Dr. Orest Pilskalns (WSU, Vancover) and Jack Hagemeister (WSU, Pullman) have also be used as guides in developing this template for the WSU-TC Spring 2005 CptS 322 course.

Software Requirements Specification

VaqPaq

Computer Science

Program of Study Organizer Pack

Version 1.0

November 5th, 2016

Lead Software Engineer

Deborah De Leon

Project Team

Jose Ballesteros

David Ramirez

Jesus Sanchez

Prepared for

Software Engineering

Instructor: MK Quweider, Ph.D.

Fall 2014

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| 11/5/2016 | Version 1.0 | Deborah De Leon | First revision |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  | Deborah De Leon | Lead Software Eng. |  |
|  | Dr. M. K. Quweider | Instructor, CSCI-3340 |  |
|  |  |  |  |

**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 2

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 3

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> 3

3.3 Use Cases 3

3.3.1 Use Case #1 3

3.3.2 Use Case #2 3

3.4 Classes / Objects 3

3.4.1 <Class / Object #1> 3

3.4.2 <Class / Object #2> 3

3.5 Non-Functional Requirements 4

3.5.1 Performance 4

3.5.2 Reliability 4

3.5.3 Availability 4

3.5.4 Security 4

3.5.5 Maintainability 4

3.5.6 Portability 4

3.6 Inverse Requirements 4

3.7 Design Constraints 4

3.8 Logical Database Requirements 4

3.9 Other Requirements 4

4. Analysis Models 4

4.1 Sequence Diagrams 5

4.3 Data Flow Diagrams (DFD) 5

4.2 State-Transition Diagrams (STD) 5

5. Change Management Process 5

A. Appendices 5

A.1 Appendix 1 5

A.2 Appendix 2 5

# 1. Introduction

## 1.1 Purpose

## The purpose of the Software Requirements Specification is to provide a detailed description of the VaqPack Computer Science Program of Study Organizer Pack application. The intention of the SRS is to articulate the purpose and features of the application, along with its user and external interfaces, constraints, dependencies, functionality, and attributes. This artifact provides the guidelines for the design and implementation of the software, and clarifies the description of the software for the customer. Therefore, the intended audience of this document includes the client, users, and developers.

## 1.2 Scope

## The software application described throughout this SRS document is the VaqPack Computer Science Program of Study Organizer Pack, or simply VaqPack. While this free desktop application can be used within any institution, it is primarily designed for the incoming Computer Science students of the University of Texas Rio Grande Valley with the purpose of aiding their transition into the Computer Science Program of Study through the use of course scheduling and management.

## Using a graphical user interface, registered users of VaqPack can select courses they are interested in taking from the Program of Study and can read the course description about each course. In addition, users can view their schedule of selected courses, including upcoming courses, remaining courses, and users will be able to calculate their current and future GPA. Users will be able to select their preferred courses and generate a PDF file with the course information and syllabus. Users will be able to see department and faculty information and will be able to send e-mails to faculty through a form on this interface, and users can also set reminders for upcoming events. The data for the course, faculty, and department information is stored in a pre-existing MySQL database.

## 1.3 Definitions, Acronyms, and Abbreviations

## The following terms, acronyms, and abbreviations are used throughout this document and are presented in the table below by order of appearance.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| SRS | Software Requirement Specification |
| VaqPack | VaqPack Computer Science – Program of Study, in short |
| GUI | Graphical User Interface; provides a visual, interactive means for a software user to manipulate the controls, commands, or features of that software. |
| Wizard | A sequential set of prompts for input, assisting in data collection and organized such that its implementation increases ease of use. |
| Database | A structured collection of data that can be efficiently and conveniently accessed. |
| PDF | Portable Document Format; a popular electronic document file type particularly used with rich-text or styled text. |
| HTML | Hyper Text Markup Language; the web standard language used in the delivery of online content, interpreted and rendered by web browsers. |
| IDE | Integrated Development Environment; software that provides tools for the development and organization of programming code. |
| Git | A version control system for the development of software. |
| GitHub | A web-based Git repository used by software development teams. |
| Java Virtual Machine | Provides the necessary links allowing a java program to run on a machine using a particular operating system. |
| Java Runtime Environment | Including the Java Virtual Machine, all necessary components for a system to establish the environment in which Java programs will run. |
| SQL | Structured Query Language; the standard relational database query language |
| JDBC | Java Database Connectivity; a Java API developed by Oracle Corporation which provides methods for querying and updated a database. |

## 1.4 References

Git - <https://git-scm.com/>

GitHub - <https://github.com/>

Java Virtual Machine - <https://java.com/en/download/>

Java Runtime Environment - <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

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

MySQL - <http://dev.mysql.com/downloads/mysql/>

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

## 1.5 Overview

The remaining content of this SRS is organized in 5 sections: General Description, Specific Requirements, Analysis Models, Change Management Process, and the Appendices. The General Description section aims to make the requirements for the VaqPack application more easily understood from a high-level point of view, especially from the perspective of typical end-users. However, the Specific Requirements will define and describe the details of these requirements with the technical information needed by the developers. The Analysis Models section lists all the models used in developing the specific requirements that are outlined in the previous section. Since the first version of VaqPack is currently being constructed, and since many requirements still need to be met, this section is subject to much change and many additions. The Change Management Process section outlines the procedures that must be followed when such changes occur throughout the development of VaqPack, including the updates to this SRS document. The Appendices include conceptual documents such as the initially provided high-level requirements and any conceptual diagrams or documents used by the developers. The documents in the Appendix may or may not be used in requirements definitions, but this is clearly specified for each document.

# 2. General Description

## 2.1 Product Perspective

The VaqPack product is independent in that it does not augment any existing product and is not intended to encapsulate another product. However, VaqPack must operate along with a MySQL server. Connectivity to a MySQL server is included with the software. VaqPack intends to provide students with an all-in-one application to assist their curriculum needs in the Computer Science Program of Study from a full course management service to scheduling and communication activities.

## 2.2 Product Functions

In a general high-level point of view, the VaqPack application will perform the following functions:

* Store user login credentials and privileges in a database.
* Store collected user information in a database.
* Store user’s schedule, reminders, and GPA in a database.
* Provide a system for an admin user to connect to a MySQL database server.
* Provide a system for an admin user to initialize the database on first run.
* Calculate GPA based on current courses and/or future courses.
* Retrieve individual collected user information for viewing or editing.
* Retrieve individual collected course list, schedule, and reminders for viewing or editing.
* 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.
* Store generated files in the database for fast access.
* Email a member of the faculty.
* Provide a system to reset a forgotten user password.
* Provide a system for an admin user to migrate the database.

## 2.3 User Characteristics

The VaqPack product is primarily designed for incoming Computer Science students of the University of Texas Rio Grande Valley. However, the user can also be an existing Computer Science student to organize and manage their classes or a faculty member who seeks to utilize the course descriptions or report generator.

## 2.4 General Constraints

In a general high-level point of view, the developers of the VaqPack application will have the following constraints:

* VaqPack must be developed using the NetBeans IDE.
* VaqPack must be MySQL database-driven
* Choice of local or remote database server must be present.
* The VaqPack GUI must be JavaFX-based.
* VaqPack must be desktop-based.
* Git and GitHub must be used for version control.

## 2.5 Assumptions and Dependencies

In a general high-level point of view, the developers of the VaqPack application requirements are currently influenced by the following assumptions and dependencies:

* It is assumed that VaqPack will run on a system with an operating system that has a compatible Java Virtual Machine and up to date Java Runtime Environment.
* It is assumed that VaqPack will run on or connect to a system with an existing MySQL server.
* It is assumed, 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.

# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interfaces

The two interface types found in the VaqPack application are as follows:

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

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.

### 3.1.3 Software Interfaces

VaqPack is Java program and therefore interfaces with the Java Runtime Environment and Java Virtual Machine for whichever platform the program must run. VaqPack must operate along with a MySQL server. Connectivity to a MySQL server is included with the software.

### 3.1.4 Communications Interfaces

VaqPack can connect to a remote MySQL database which may require Internet connectivity. VaqPack can also send files as email attachments which requires Internet connectivity. The product communicates with the MySQL server via a driver that is embedded in the software and uses SQL standards.

## 3.2 Functional Requirements

### 3.2.1 Store User Login Credentials

3.2.1.1 Introduction

VaqPack stores the user’s chosen email and hashed password in a database.

3.2.1.2 Inputs

The user provides their email address and a password of their choosing.

3.2.1.3 Processing

VaqPack stores the email and hashed password in the ‘user’ database table.

3.2.1.4 Outputs

Upon entering the proper credentials, the user is allowed access to their stored data.

3.2.1.5 Error Handling

VaqPack verifies the correct email and password are used to login and that valid inputs are provided at the time of account creation.

**3.2.3 Display Degree Plan**

3.2.3.1 Introduction

The user can view degree plan with all available classes and view course description/scheduling.

3.2.3.2 Inputs

The user must click on a class to view information about it.

3.2.3.3 Processing

Each class in the degree plan is a hot link. VaqPack displays the course information and scheduling for that class when a user clicks on a hot link.

3.2.3.4 Outputs

No output unless retrieved via database query.

3.2.3.5 Error Handling

VaqPack keeps tracks of selected courses to catch possible duplicate selections.

## 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

### 3.5.2 Reliability

### 3.5.3 Availability

### 3.5.4 Security

### 3.5.5 Maintainability

### 3.5.6 Portability

## 3.6 Inverse Requirements

State any \*useful\* inverse requirements.

## 3.7 Design Constraints

Specify design constrains imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.

## 3.8 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.

## 3.9 Other Requirements

Catchall section for any additional requirements.

# 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