Software Requirements Specification

VaqPack

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 2016

# 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. |
| GPA | Grade Point Average |

## 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 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 a 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 Display Log-In Screen

3.2.1.1 Introduction

VaqPack displays a log-in screen when it is started.

3.2.1.2 Inputs

If the user is a first-time user, they will be prompted to register with an e-mail address and 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.

**3.2.2 Display Program of Study Degree Plan – Tab 1 GUI**

3.2.2.1 Introduction

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

3.2.2.2 Inputs

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

3.2.2.3 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.

3.2.2.4 Outputs

Output is a window with course description and information.

3.2.2.5 Error Handling

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

**3.2.3 Display GPA Widget – Tab 1 GUI**

3.2.3.1 Introduction

VaqPack allows the user to interact with a GPA widget that can calculate their GPA based on current and/or future courses.

3.2.3.2 Inputs

The user must select their courses.

3.2.3.3 Processing

VaqPack calculates the GPA using the number of courses selected and the hours from each course.

3.2.3.4 Outputs

VaqPack updates the meter in the GPA widget with the calculated GPA.

3.2.3.5 Error Handling

VaqPack verifies that each course has a number of hours and sanitizes the course fields after user is done using widget.

**3.2.4 Display Remaining Courses List – Tab 1 GUI**

3.2.4.1 Introduction

VaqPack will display the remaining courses list based on the user’s selection and update a pie chart to reflect their selection.

3.2.4.2 Inputs

The user must select their courses from the Program of Study.

3.2.4.3 Processing

VaqPack determines the remaining courses in the students’ Program of Study based on their selection.

3.2.4.4 Outputs

VaqPack displays a list of the remaining courses and updates the pie chart.

3.2.4.5 Error Handling

VaqPack verifies that there are still courses remaining in their program of study.

**3.2.5 Display Faculty/Department Information – Tab 2 GUI**

3.2.5.1 Introduction

VaqPack displays the faculty and department information.

3.2.5.2 Inputs

The user may click on a faculty member to view their information and may choose to e-mail them.

3.2.5.3 Processing

VaqPack opens a network connection to allow the user to e-mail a faculty member.

3.2.5.4 Outputs

VaqPack displays faculty/department information and an e-mail form if the user chooses to send an e-mail.

3.2.5.5 Error Handling

If user is sending an e-mail, VaqPack verifies whether e-mail valid.

**3.2.6 Reminder Mechanism for Upcoming Events – Tab 2 GUI**

3.2.6.1 Introduction

VaqPack allows the user to set reminders for upcoming events.

3.2.6.2 Inputs

The user sets time, date, and description of reminder.

3.2.6.3 Processing

VaqPack stores this information in the database for future log-ins.

3.2.6.4 Outputs

VaqPack displays reminder of events every time user logs in.

3.2.6.5 Error Handling

VaqPack verifies that time and date are valid.

**3.2.7 Report Generation – Tab 3 GUI**

3.2.7.1 Introduction

VaqPack allows the user to select courses and generate reports from that selection that include the course description, schedule, and syllabus.

3.2.7.2 Inputs

The user selects the courses they want a report for or they may elect to generate a report for all courses.

3.2.7.3 Processing

VaqPack takes selected courses and adds them to a document which is then converted to PDF format.

3.2.7.4 Outputs

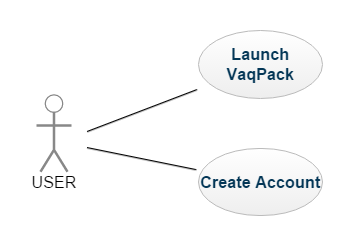
VaqPack displays the PDF report for the user who may then elect to save it to their hard drive or print it out.

3.2.7.5 Error Handling

VaqPack makes sure there are no duplicates in the user’s selection.

## 3.3 Use Cases

**3.3.1 Create a VaqPack Account**



**Goal:** User Creates an Account

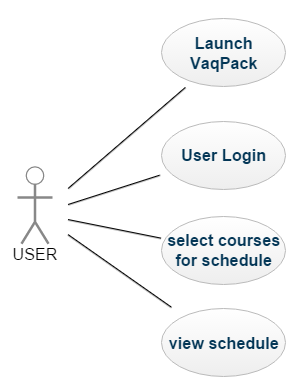
**Input:** Email/Password

**Output:** none.

**Main Scenario:** User creates an account to store user login credentials

**Pre-Condition:** VaqPack must be running.

**3.3.2 User creates schedule from selected courses**



**Goal:** User creates schedule from selected courses

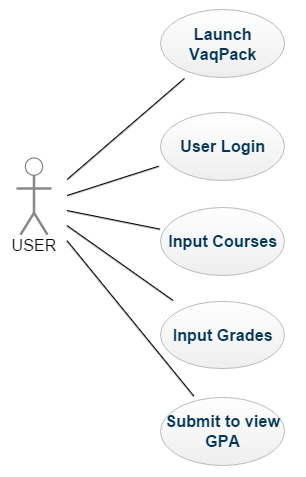
**Input:** Email/Password, click on selected courses

**Output:** user view schedule of selected courses

**Main Scenario:** user creates schedule from Program of Study courses

**Pre-Condition:** VaqPack must be running.

**3.3.3 User Calculates GPA**



**Goal:** User views their calculated GPA

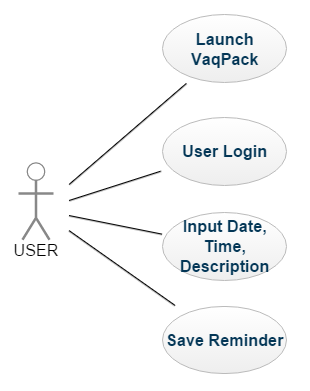
**Input:** Email/Password, input courses, input grades for those courses, click submit

**Output:** calculated GPA, displayed through text and GPA meter.

**Main Scenario:** User inputs their grades and views their calculated GPA.

**Pre-Condition:** VaqPack must be running.

**3.3.3 User sets reminder**



**Goal:** User sets reminder for an event

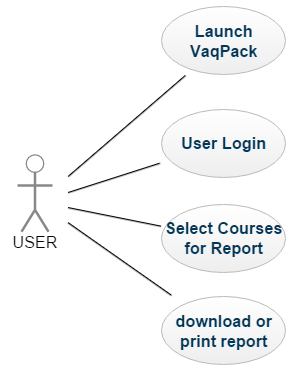
**Input:** Email/Password, input date, time, and description of event

**Output:** reminder viewable and accessible every time user logs in.

**Main Scenario:** User sets reminder for upcoming events.

**Pre-Condition:** VaqPack must be running.

**3.3.4 Download or Print Report of Selected Courses**



**Goal:** User prints or downloads a PDF file of their selected courses

**Input:** Email/Password, select courses, click generate report

**Output:** a PDF file in a viewer window

**Main Scenario:** User selects their courses and prints or downloads a report of the course descriptions and schedule.

**Pre-Condition:** VaqPack must be running.

**3.4 Classes / Objects**

**3.4.1 VaqPack**

**3.4.1.1 Description:**

This class contains the main method entry point for the program. The JavaFX Stage is launched, the window is given a title, and a POS\_GUIController object is created which begins the construction of the user interface. Upon completion of the construction, the stage is shown to the user.

**3.4.1.2 Attributes:**

This class has no attributes.

**3.4.1.3 Functions:**

* main – launches JavaFX Stage
* start – creates a POS\_GUIController object, names the window, shows the stage.

**3.4.1.4 Use Case Reference:**

This class is inherently referenced by all use-cases since it is the main entry point for the program.

**<< THE FOLLOWING SPACE IN 3.4 IS RESERVED FOR THE CLASSES AND OBJECTS THAT ARE ANTICIPATED TO BE USED FOR THIS PROJECT. The descriptions and details for these classes and objects will be updated accordingly during the design process >>**

**3.4.2 POS\_GUIController**

**3.4.2.1 Description:**

**3.4.2.2 Attributes:**

**3.4.2.3 Functions:**

**3.4.2.4 Use Case Reference:**

**3.4.3 POS\_GUIBuilder**

**3.4.3.1 Description:**

**3.4.3.2 Attributes:**

**3.4.3.3 Functions:**

**3.4.3.4 Use Case Reference:**

**3.4.4 POS\_Header**

**3.4.4.1 Description:**

**3.4.4.2 Attributes:**

**3.4.4.3 Functions:**

**3.4.4.4 Use Case Reference:**

**3.4.5 POS\_Tree**

**3.4.5.1 Description:**

**3.4.5.2 Attributes:**

**3.4.5.3 Functions:**

**3.4.5.4 Use Case Reference:**

**3.4.6 POS\_Center**

**3.4.6.1 Description:**

**3.4.6.2 Attributes:**

**3.4.6.3 Functions:**

**3.4.6.4 Use Case Reference:**

**3.4.7 POS\_Loader**

**3.4.7.1 Description:**

**3.4.7.2 Attributes:**

**3.4.7.3 Functions:**

**3.4.7.4 Use Case Reference:**

**3.4.8 POS\_DataManager**

**3.4.8.1 Description:**

**3.4.8.2 Attributes:**

**3.4.8.3 Functions:**

**3.4.8.4 Use Case Reference:**

**3.4.9 POS\_DatabaseManager**

**3.4.9.1 Description:**

**3.4.9.2 Attributes:**

**3.4.9.3 Functions:**

**3.4.9.4 Use Case Reference:**

**3.4.10 POS\_FileManager**

**3.4.10.1 Description:**

**3.4.10.2 Attributes:**

**3.4.10.3 Functions:**

**3.4.10.4 Use Case Reference:**

**3.4.11 DataToHtml**

**3.4.11.1 Description:**

**3.4.11.2 Attributes:**

**3.4.11.3 Functions:**

**3.4.11.4 Use Case Reference:**

**3.4.12 POS\_** **HtmlToPdf**

**3.4.12.1 Description:**

**3.4.12.2 Attributes:**

**3.4.12.3 Functions:**

**3.4.12.4 Use Case Reference:**

**3.4.13 POS\_Mail**

**3.4.13.1 Description:**

**3.4.13.2 Attributes:**

**3.4.13.3 Functions:**

**3.4.13.4 Use Case Reference:**

**3.4.14 POS\_ErrorHandler**

**3.4.14.1 Description:**

**3.4.14.2 Attributes:**

**3.4.14.3 Functions:**

**3.4.14.4 Use Case Reference:**

**3.4.15 POS\_User**

**3.4.15.1 Description:**

**3.4.15.2 Attributes:**

**3.4.15.3 Functions:**

**3.4.15.4 Use Case Reference:**

**3.4.16 POS\_Template**

**3.4.16.1 Description:**

**3.4.16.2 Attributes:**

**3.4.16.3 Functions:**

**3.4.16.4 Use Case Reference:**

**3.4.17 POS\_GPAWidget**

**3.4.17.1 Description:**

**3.4.17.2 Attributes:**

**3.4.17.3 Functions:**

**3.4.17.4 Use Case Reference:**

**3.4.18 POS\_Reminder**

**3.4.18.1 Description:**

**3.4.18.2 Attributes:**

**3.4.18.3 Functions:**

**3.4.18.4 Use Case Reference:**

**3.4.19 POS\_Courses**

**3.4.19.1 Description:**

**3.4.19.2 Attributes:**

**3.4.19.3 Functions:**

**3.4.19.4 Use Case Reference:**

**3.4.20 POS\_Faculty**

**3.4.20.1 Description:**

**3.4.20.2 Attributes:**

**3.4.20.3 Functions:**

**3.4.20.4 Use Case Reference:**

**3.4.21 POS\_Department**

**3.4.20.1 Description:**

**3.4.20.2 Attributes:**

**3.4.20.3 Functions:**

**3.4.20.4 Use Case Reference:**

**3.4.20 POS\_Schedule**

**3.4.20.1 Description:**

**3.4.20.2 Attributes:**

**3.4.20.3 Functions:**

**3.4.20.4 Use Case Reference:**

**3.5 Non-Functional Requirements**

**3.5.1 Performance**

There are no performance requirements at this stage of the development of VaqPack. However, the product performance shall perform under typical standards.

**3.5.2 Reliability**

There are no reliability requirements at this stage of the development of VaqPack. However, it is the goal of the development team to ensure maximum reliability.

**3.5.3 Availability**

The stored data for any particular course, faculty or department must be available upon request, provided that there are no other issues external to the VaqPack software such as hardware failure, the database server being inactive, or lack of or poor Internet connectivity preventing proper access to a remote MySQL database server.

**3.5.4 Security**

The following system is in place to avoid the compromise of user data. Users of VaqPack must login with a user and password. The password must be hashed before being stored in the database. A user may request to change his or her password, upon which an email with a confirmation code will be sent to the email associated with the user. The user must enter the code into the form provided in VaqPack within 60 minutes to successfully change the password.

**3.5.5 Maintainability**

No requirements concerning maintainability have been outlined by the customer.

**3.5.6 Portability**

No portability requirements have been defined by the customer. However, the requirement of Java as the programming language for the VaqPack project implies that the product is portable to any machine with an operating system for which some Java Runtime Environment with corresponding Java Virtual Machine exists.

**3.6 Inverse Requirements**

The VaqPack application will not provide any functionality without a connection to a MySQL database. Upon such an event, the user is to be notified, followed by an immediate closing of the program.

Besides any administrator user account, the typical user shall not ever be granted the privilege of accessing another user’s data.

**3.7 Design Constraints**

The VaqPack design must incorporate in some way a wizard, with which the program collects the data from the user in sequential, organized manner.

**3.8 Logical Database Requirements**

VaqPack must access and store data in a MySQL database. Functionality must be in place to allow the migration of data from a database server at one location to another. The only requirements for the types of data and capabilities are that they align with the completion of a working product that meets all other requirements. Additionally, the security requirements, as outlined in section 3.5.4 must be implemented.

**3.9 Other Requirements**

No other requirements exist at this stage of development.

**4.** **Analysis Models**

There are no models to present at this stage of development.

**4.1 Sequence Diagrams**

There are no sequence diagrams related to meeting requirements at this stage of development. This section is to be updated soon.

**4.2 Data Flow Diagrams (DFD)**

There are no data flow diagrams related to meeting requirements at this stage of development. This section is to be updated soon.

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

There are no state-transition diagrams related to meeting requirements at this stage of development. This section is to be updated soon.

**5.** **Change Management Process**

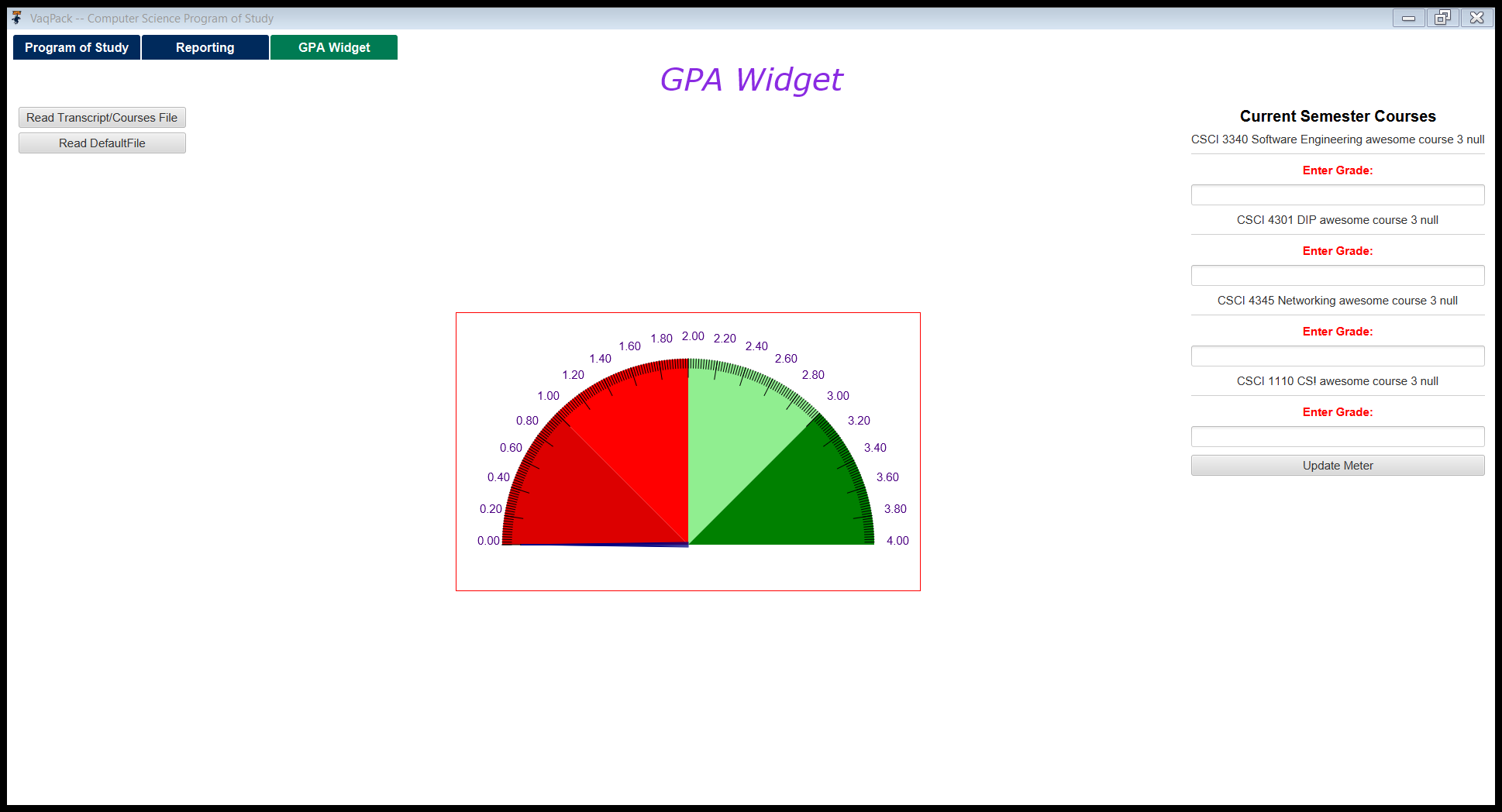
The following procedures are required when changing this SRS document.

* Suggested changes by the development team to the scope or requirements outlined in this SRS document are to be presented independently by creating a branch for the edit in GitHub.
* The project manager will examine the changes, and if approved, will attach a signed document found in page ii to a copy of the updated SRS.
* If changes are dictated by the customer, whether verbally or written, the Project manager will create an updated copy of the SRS and attach a signed document found in page ii.
* The project manager will present the updated SRS to the customer for final approval, upon which the customer will also sign the document found in page ii.
* When both the project manager and the customer approve the updates to the SRS, the project manager will merge the official SRS with the updated SRS via Git.
* The project manager completes the procedure by recording the update in the document found in page ii.

# A. Appendices

## A.1 Appendix 1

The following is a preliminary GUI design concept and is not to be officially considered as part of the requirements.

****

## A.2 Appendix 2