Week 1 – Assignment

John Grier

CST499 - Capstone for Computer Software Technology

University of Arizona Global Campus

Instructor: Joseph Rangitsch

May 20, 2024

**Software Requirements Specification**

**for**

**SRS Document Development**

**Version 1.0 approved.**

**Prepared by John Grier**

**UAGC**

**May 20, 2024**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1. Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

**3. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

**Software Requirements Specification**

**1. Introduction**

**1.1 Purpose**

This document lists the requirements for the Easy Enroll Online Course Software. This SRS is the very first revision to the document and version 1.1. The functional and non-functional are the development needs for the user to enroll online in the class cum photography service directory. The users can acquire information regarding the classrooms offered via a database driven by phpMyAdmin powering XAMPP, running in WordPress. The SRS explains database capabilities and cloud parameters, but the technical coding sequences for the indexes and entity relationships should be discussed in detail.

**1.2 Document Conventions**

Bolded Fonts These outline each section by subsection and title headings for this Form. All the other Fonts are Standardized in the Form, and highlighting is prohibited. Each Requirement Statement is to have a Standalone priority. Outside references are to be cited in APA style.

**1.3 Intended Audience and Reading Suggestions**

This SRS addresses various stakeholders, including developers, project managers, users, testers, and documentation writers. In general, following the sequence of these sections is recommended to interpret this document. Specifically, to understand the overall scope of the software system, project managers and documentation writers should focus on reading the whole document. General users should narrow down to sections from 2.1 through 2.7, namely the overall description of the system from the user’s perspective. Meanwhile, developers and testers should focus on parts II and III, referring to sections 3 through 5, namely the functional and non-functional requirements of the software system, respectively.

.

**1.4 Product Scope**

The software provides an electronic podium, " Easy Enroll," so users can connect to the website. Users have the right to make a new account for themselves. The software will protect users from making the same user ID. If the user needs anything else, there needs to be more information about users put in their profile, such as their name, phone number, e-mail, etc. The users can log in any time they want. After they make their account, they can enroll in courses. There are three semesters per year(spring, summer, and fall). The classes are limited to a maximum number of people to start the class; that is, the teacher's decision. If the course is complete, there is a quiet waiting period, and the students who want to take the class can be put on the waiting list. Users should be able to cancel the class from any course they are set in the class, and the software will contact the first one on the waiting list and tell them the class is available. This will help the user and the business strategy because there will be a lot of students and faculty using polymorphic software platforms, and there is a possibility of providing funds for universities.

**2. Overall Description**

**2.1 Product Perspective**

We are developing a standalone product with multiple tables with primary and foreign keys, triggers, and relational dependencies. The database will be connected to external entities and published on GitHub.

**2.2 Product Functions**

* Agnostic To OS (Operating System) — The system must be able to be accessed by users from Windows, Macintosh, and Linux desktop operating systems. Each user has to go through a registration process to join classes and set up a new account/ profile.
* Each new user gets an ID with a password linked to them. The system should block two users from having the same ID when applying for the registration process. A profile includes significant information about the applicant that one can enter, such as name, phone, e-mail, and any other information you might think of. After registering, a user can log in to the system at any time using an ID and a password from the registration process.
* The system should support three semesters for online courses throughout the year: Spring/summer/fall. Three options have been given for the course choice, and all the courses are only provided in some semesters. Instead, a student can list the course they like to enroll in only if such a course has been created for any of the semesters. There is a fixed limit for enrollment in each course, which can differ from course to course.
* If the user wants to enroll in a course and the class is complete, the student can be placed on a waiting list.
* A user can unenroll from any course they are enrolled in, and the system can tell the next person on the waiting list (if any) that the class is now open.

**2.3 User Classes and Characteristics**

It is the responsibility of the administrative user class to administrate the system and to update the database for system requirements. For example, all developers are part of the administrative user class. Faculty users can add and remove students from courses, add and remove courses, and change the size and availability of courses. For general students, these roles will encompass all academic stuff. Although they may not have many administrative rights, general students can search for courses, request an add-on course, request an addition to a waiting list, and so on. All of the rights given to general students fall under the General Student User Class. General student populations will use the majority of general student users. An individual wishing to access a specific account (e.g., a student, faculty, or administrative account) should have sufficient background to register and create a profile before signing in. A guest user can register as a general student, faculty member, or administrative member. A guest user can register either for one of these accounts or logs into the system. Product development depends equally on all user classes. General student users play a crucial role in the system as they perform various academic tasks such as searching for courses, requesting add-on courses, and adding themselves to waiting lists. While they may not have extensive administrative rights, their active participation in the system contributes to the overall functionality and success of the product development process.

**2.4 Operating Environment**

Cross-platform functionality as well as mobile responsiveness are key characteristics of an e-commerce website. Besides functioning properly on all major web browsers, the website should also work on iOS, Android, Bing, and Yahoo as well. Additionally, it should work well on both Mac and Windows operating systems.

**2.5 Design and Implementation Constraints**

As a result, Easy Enroll Online Course Software has limitations in terms of the host. An increase in the host and its cost may require setting up the Easy Enroll Online Course Software database locally on every machine the user wants. A PHP code injected into the user's htdocs folder will incorporate the various functionalities, which will be organized using XAMPP and the interconnected database. Paul (2019), in the article 'IONOS Digital Guide.' According to the 'XAMPP tutorial: installation and first steps,' you can set up a PHP page in seconds by putting the following text in your editor and saving it in a new file. For example, test.php in the "test" folder (C:/xampp/htdocs/test): (to 13). Across the site, multiple language requirements and accessibility add-ons limit the site's accessibility. There is also the possibility of a security breach occurring during all stages of the development process, as personal information/details are stored in the system, which can be compromised under circumstances where it might be manipulated to be used for unauthorized purposes.

**2.6 User Documentation**

It is anticipated that the website will not only provide users with the requested functionalities, but also provide documentation to assist them in troubleshooting problems or creating an account if they require assistance. The site will also include a chat bot, which allows users to request assistance whenever necessary.

**2.7 Assumptions and Dependencies**

Sometimes, the website may not function after migration to the cloud, the application might not run during the development phase, and the operating system might be affected. Consequently, here is the simple software system developed by the University of Arizona Global Campus for a university student taking online classes. Neither the University nor a monetary incentive can be provided to students in return for an accessible web application on the internet that works. Users may need to extract the zip files, copy-paste them into the root directory on their local computer and run the XAMPP program to view the website. Communicating with or interacting with other applications will not be possible unless the website is hosted on GitHub. However, hosting the website on GitHub may have its limitations. For example, GitHub is primarily designed for version control and code collaboration rather than hosting live websites with dynamic functionality. It may not support certain server-side technologies or databases required for the website to function fully, and there may be limitations on storage space and bandwidth for hosting larger websites or handling high-traffic volumes.

**3. External Interface Requirements**

**3.1 User Interfaces**

For consistency, the header, footer, and navigation bar should be consistent across each page of the website. The icons should all be selected from Font Awesome. An arrow should appear on every page that enables users to navigate back to the previous page in the same location. Each image should include an alt tag and the user should be able to navigate through the page. Furthermore, colors should be selected for ease of use and visual appeal, as well as fonts which are easily readable and easy to read. In addition, advertisements should not distract attention from the page's main content, and the layout of the page should be consistent throughout. Besides clearly displaying the home button on each page, forms should provide users with clear instructions as to how to correct errors when they occur.

**3.2 Hardware Interfaces**

The specification also states that software must interact with the user device keyboard to explain the entry command. The software shall also interact with the hardware present on this device. Developed for both citizens, this software is agnostically coded for use in Microsoft, Linux, and Apple products. The software must be accessible by or capable of allowing users to use it on Windows, Linux, and Mac OS devices. The kernel should also be provided or coded for the software debugging process to communicate with the system.

**3.3 Software Interfaces**

There should also be an API that talks to a relational database so that data about customers, products, and transactions can be stored and retrieved. The website should be compatible with iPhones, Androids, Macs, and Windows operating systems. Additionally, the site should offer single sign-on and options to use Google and Facebook accounts.

**3.4 Communications Interfaces**

This product serves users by mailing notifications, involves web browsers and network servers communicating using HTTP, and involves registration of user accounts and information regarding course enrollment. To edit its own functions, a user is not required to verify the creation of an e-mail account. For the specified purpose of a mail service, the HTTP-based data transfer with asynchronous connections between distributed clients and a server is necessary and essential. The communication security of personally identifiable information must be considered when executing general transactions in such a system. The encryption of specific user personal information (names, ssn, and addresses­ – all personally identifiable information in user account information) must be considered for a training course. To sync data queries and training course availability, the university course catalog will be filled as enrollment capacity reaches a maximum and will be depleted as a particular course is filled.

**4. System Features**

**The system is built to be compatible with Windows, Macintosh, and Linux desktop operating systems.**

**4.1.1 Description and Priority**

The software is written in an agnostic manner. The system works in any windows, Macintosh and Linux desktop Operating System. Priority level 9.

**4.1.2** Stimulus/Response Sequences. Buyer should open System web site or platform by loading applicable desktop application for the system to respond by opening the page.

**4.1.3** Functional Requirements.

**REQ-1:** There is a Customizable Graphical User Interface for each one of the user types (Student, Faculty, Administrative, Guest) to access the system. The size of the screen is customized for the size of the GUI.

**REQ-2:** GUIs should be interactive. Users should be able to navigate the site using keyboard and mouse. Touch and pen-input devices are included here. ‘One of the top rules of UX is to keep the interface consistent. The look and feel of your website should be the same across every page of your site.’ ‘The functionality should match’ (para 2): ‘The 12 Do’s and Don’ts of Web Design | Adobe XD Ideas’ (2021), by Faiza.

**An approved user must select an entry from the table to register a new account/profile. Each user is assigned a unique identification number ID linked to a password. Hence, it is not possible to register several users using the same identification number or ID. In other words, duplicates are not possible. Some information must appear on the user's profile regarding the candidate in the system, such as their name, telephone number, and e-mail address. The remaining data can be represented using your imagination. Upon registration, each time a user wishes to access the system, they must provide their ID and password to begin using the system.**

**4.2.1** Description and Priority Users should be able to write accounts with per-user privilege levels and unique user IDs Priority level 8.

**4.2.2** Stimulus/Response Sequences User The creates an account; Verification need not occur (e-mail reply can be sent). The system will send the status of the account and any other relevant information to the user.

**4.2.3** Functional Requirements

**REQ-1:** The system should automatically assign a guest account while accessing it for the first time. The user can create an account or update his existing account. The system should be efficient enough to prevent duplicate system user IDs.

**REQ-2:** Students (General users), administrative users, and faculty users (who will have administrative privileges and likely be involved in creating user accounts) may 10 save and edit profile information on the system. R

**EQ-3:** Users seeking higher levels of administrative control must have separate accounts; developers, testers, and other power users must be separated from ordinary users.

**REQ-4:** A faculty-level user account must be able to edit course information, add students, drop students, and manage course content.

**As online courses are not offered in all semesters, the system should be able to provide 3 semesters per year (spring, summer, and fall) and the students could specify the courses they wish to take during each semester. There will be an enrolment capacity for each course that may vary from one course to another.**

**4.1** Description: Easy Enroll Online Course Software System allows members to perform search queries on the database. Priority level 7.

**4.3.2** Stimulus/Response Sequences Course availability functions will enable users to enter queries into the easy enroll Online Course database. Users can find out which courses are available by typing in their questions. The functions will reply to their queries with course descriptions and enrollment information. Courses can be listed over three semesters based on the availability and needs of the user. The database functions will respond to the input by checking the number of "seats" in the course, and the users can enroll.

**4.3.3** Requirements for Functionality

**REQ-1:** Each user should be able to add and remove courses according to their availability in the database.

**REQ-2:** In generating pertinent courses, we utilize query language and indexing, particularly course search. Courses can be enrolled in or dropped at any time.

**Users can add themselves to a waiting list if a course is already entire or if they wish to enroll in a course that is already full. Alternatively, a student can cancel all the classes they are enrolled in. The system must inform the first person on the waiting list (if any) that they can enroll.**

**4.4.1** Description and Priority The software needs to give users priority information about whether course ‘seats’ are available, allowing users to use another functionality that will need to be added in case the courses are already complete.

**4.4.2**  Priority level 6. Stimulus/Response Sequences Users frequently access the system by using search commands to find courses and course availability, entering table tagging, and generating course waiting list results and notifications about course acceptances/denials if students drop out of a whole class.

**4.4.3** Functional Requirements

REQ-1: The system will tag and index items to provide student waiting lists for courses if they are complete.

**REQ-2:** Notify users when new courses become available that they are eligible to open.

REQ-3: Students, as users in general, should be able to apply to waitlists and drop courses that are related to waitlists.

**REQ-4:** If a student is on any of the waiting lists for a course and this student drops from the course, the system shall issue warnings to all students belonging to the resulting applicable waiting lists.

**5.1 Performance Requirements**

Eventually, the process and loading times are minimal for users. It should feel highly reliable and quick, which are examples of positive system performance. The following statement is provided by Lyubov (2020), "Non-Functional Requirements: A Guide with Concrete Examples," para. 4, which states that performance requirements determine whether a system is acceptable or not. When a user decides to use the system, the requests documented in enrollment and account creation must occur in seconds, if not instantly, when the system's performance is primarily determined by its response time and throughput. Using the system should not result in any bandwidth loss or lag for the user.

**5.2 Safety Requirements**

It is imperative that the website verify a user's age before confirming their order. The website should protect all personally identifiable information, and all passwords should be encrypted, and sessions should end after eight minutes of inactivity, unless the user confirms they wish to continue.

**5.3 Security Requirements**

Credit card information should not be stored, and transactions should be processed in a secure manner. Credentials should be verified before accessing account information, and passwords should be encrypted.

**5.4 Software Quality Attributes**

User identity is not authorized by an email message attached to the account. This software system will run on local machines that do not pose any substantial security risk to user information. All information entered into the system will be fictitious and joined at no risk to the user. Suppose information about the user (such as social security numbers, addresses, etc) is entered into a local machine. In that case, any information about a real user should have a user identity masked from harmful hacks. Information about a user should not be publicly accessible; therefore, information security and user privacy certification must be satisfied to ensure the safety of the data.

**5.5 Business Rules**

In addition to ensuring that users are logged in, administrators should be able to assist them in resetting their credentials if they require them to add items to their carts, checkout, or view account information. In addition to searching for order details using a reference number for administrative purposes, administrators should be able to obtain a list of all past and present orders to conduct reports.

**Appendix A: Glossary**

GUI –GUI stands for graphical user interface

HTML –markup language for hypertext

HTTP –Protocol for sending and receiving text

REQ -The requirement

SRS –Specifications for Software Requirements

TBD –To Be Decided

**Appendix C: To Be Determined List**

**Reference:**

Lyubov, T., (2020). Non-Functional Requirements: A Guide With Concrete Examples - Plutora.

Plutora. https://www.plutora.com/blog/non-functional-requirements-guide