**Software Requirements Specification**

**for**

**EasyEnroll Online Course Software**

**Version 1.2 approved**

**Prepared by Jeremy Roosa**

**The University of Arizona Global Campus**

**January 18, 2022**

**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** |
| Jeremy Roosa | December 11, 2021 | Initial creation of Requirements Document | 1.1 |
| Jeremy Roosa | January 18, 2022 | First Revision of Requirements Document (No Significant Changes) |  |

**Introduction**

**Purpose**

The EasyEnroll Online Course Software requirements are specified in this document. This SRS is the first revision to the document and is identified as version 1.1. The functional and non-functional requirements are determined to help users enroll in online classrooms via a database developed through phpMyAdmin, run on XAMPP and WordPress. The database and cloud storage capabilities are discussed throughout this SRS, but the detailed level of coding indexes and entity relationships are not discussed.

**Document Conventions**

**Number Bolded Fonts** used throughout this document break down each section by title and subsection headings. All other font is standardized throughout the form, and highlights are not used. Every requirement statement is to have its own priority. Outside references will be annotated in APA format.

**Intended Audience and Reading Suggestions**

This SRS is intended for developers, project managers, users, testers, and documentation writers. Overall, the SRS should be read in chronological order to interpret the document better. Project managers and documentation writers should read the entirety of the document to understand the full scope of the software system. General users should focus on sections 2.1 through 2.7 to understand the high-level, overall description of the software system. Developers and testers should focus on sections 3 through 5 to understand the software system’s detailed functional and non-functional requirements.

**Product Scope**

The software is intended to provide users with an electronic platform called EasyEnroll to enroll in classes. Users will be able to register a new account in which the software will protect against duplicate user IDs. Profiles will include basic information about the user, such as their name, phone number, e-mail, and any other information that may be necessary. Users will be able to log in at any time after making their account and enroll in courses that run through three semesters per year (spring/summer/fall). Each course will have a maximum number of students who can enroll, depending on the course. Students should be able to be added to a waiting list if the course they want to enroll in is full. Finally, users should be able to cancel their enrollment from any course they are enrolled in, and the system should inform the first person on the waiting list that the course is available. This platform will benefit the user and the business strategy in that a plethora of students and faculty will utilize the platform, and possible revenue could be generated for the university.

**References**

Faiza, J., (2021). The 12 Do’s and Don’ts of Web Design | Adobe XD Ideas. Ideas. https://xd.adobe.com/ideas/principles/web-design/12-dos-donts-web-design-2/

Lyubov, T., (2020). Non-Functional Requirements: A Guide With Concrete Examples - Plutora. Plutora. https://www.plutora.com/blog/non-functional-requirements-guide

Paol, F., (2019). IONOS Digital Guide. XAMPP tutorial: installation and first steps. IONOS Digitalguide. https://www.ionos.com/digitalguide/server/tools/xampp-tutorial-create-your-own-local-test-server/

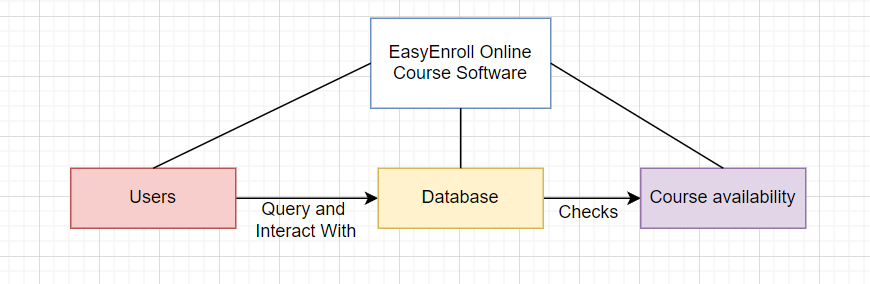
Tsui, F., Karam, O., & Bernal, B. (2018). Essentials of software engineering (4th ed.). Jones & Bartlett Learning.

**Overall Description**

**Product Perspective**

This system is a new, self-contained product. The overall EasyEnroll Online Course Software database will have multiple tables with data that contains triggers and relationships. The database will also be interoperable with outside entities and posted to GitHub.

Figure 1 EasyEnroll Online Course Software Overview



**Product Functions**

* Agnostically Coded - Users should access the system from any Windows, Macintosh, and Linux desktop operating systems.
* Users should query a database to enroll in classes and register a new account/profile. Each new user should have a unique ID associated with a password. The system should guard against two users using the same ID for registration. Profiles must include some key information about the applicant including name, phone, e-mail, and any other information you may see necessary. Post registration, users can login to the system at any time using the ID and the password created during the registration process.
* The system should allow for online courses to run through three semesters per year (spring/summer/fall), and students can list the course that will be offered during any semester, as not all courses will be offered in every semester. Each course should have a maximum number of enrollments that may be different depending on the course.
* If a user wants to enroll into a course and the course is full, the student can add themselves onto a waiting list. A user can also cancel the enrollment from any course that they are enrolled in, and the system should inform the first in the waiting list (if any) that they can enroll into the class.

**User Classes and Characteristics**

The various user classes that will utilize the system include administrative, faculty, general student, and guest users. Administrative users will be able to manipulate the system and update the databases based on requirements. Developers and other users that require administrative privileges will fall under this category. Faculty users will be able to add and remove students from courses, add and remove courses, and alter course sizes and availability. General student users will have limited rights to search for courses, request to be added to courses, request to be added to course waiting lists, and request to be removed from courses. General student user classes will be the most frequently used class by the general student population. Before signing in to their respective accounts, guest users will have limited access to register and create profiles. Guest users will have the option to create a general student user account, create a faculty level account, create an administrative account, or log in to the system. All types of user classes are of equal importance during product development.

**Operating Environment**

The product will operate in a web-based environment only accessible through desktop functionalities. This application will not be accessible from mobile devices (i.e., cell phone/tablet). Standalone web database application software will have to be developed. Users can interact with the HTML-based web browsing software in a cloud-based environment. The product will be agnostically coded to operate on Windows 10, Windows 11, ArchLinux, MacOS 11, and MacOS 12.

**Design and Implementation Constraints**

Hosting limitations and associated costs will require users to set the EasyEnroll Online Course Software database up locally on every machine they need. This software and the connected database will be built using XAMPP, and the various PHP code will be placed in the users’ htdocs folder to enable the different functionalities. According to Paol (2019) in the article “IONOS Digital Guide. XAMPP tutorial: installation and first steps,” “you can create a new PHP page easily by using the following content in your editor and storing it as test.php in your ‘ test’ folder (C:\xampp\htdocs\test):” (para. 13). Along with local hosting, multiple language requirements and accessibility features must also be considered throughout development. Security considerations are also a significant constraint as personal information is saved within the system. Data leaks could result in personal data being maliciously used by unwanted entities.

**User Documentation**

There are many online help and tutorials associated with this software system. Many user guides for creating HTML web pages through XAMPP are available. A WordPress instructions document will be developed to ensure users can properly set up the software on their local machine. A frequently asked questions section and online help features should be delivered along with the software.

**Assumptions and Dependencies**

Gaining cloud hosting services for the website could cause issues during the development and the operating environment. This software system is being built for an online course taken by a student at the University of Arizona Global Campus. Therefore, the student will not be issued funds or provided a monetary incentive to a functional application hosted on the internet. Users will have to download zip files, place them in the correct location on their local machine, and run the XAMPP application to access the website. This website will not be interfaced with external applications other than being hosted on GitHub.

**External Interface Requirements**

**User Interfaces**

There are various user interfaces and website screens that will be exhibited in the EasyEnroll Online Course software. The software’s home screen will show the landing page and allow users to navigate the system as a guest user.Users should interact with a sign-in option when loading the software. After logging in, the graphical user interface (GUI) should exhibit an account section where users can update information, view courses, enroll in courses, access waiting lists, drop classes, and view their courses. Different user interfaces will have to be developed for varying user classes as Faculty Users will be able to accomplish varying functionalities. Administrative users will have a more encompassing user interface and have to ability to edit the website. A standard search bar will allow users to look up courses with suggested searches populate as the user starts typing. If a user encounters an error, an error message will be displayed on the screen. Screen layout constraints must be tailored to meet standard computer monitors.

**Hardware Interfaces**

Software components will have to incorporate interaction with a user keyboard and the hardware within the device. The software will have to be tailored to be agnostically coded for Microsoft, Linux, and Apple products. The software should be able to be accessed from Windows, Linux, and Macintosh devices. The applicable Kernal for each operating system should be coded to the software system.

**Software Interfaces**

As previously described, the software should interface with Macintosh, Linux, and Windows operating systems. The system will also be interfaced with the EasyEnroll database to send and receive data about the various courses, user account information, and enrollments. Users’ search queries will send data to the database in which course availability will be determined. Data will then be sent out to the student user to decide if they will enroll, disenroll in, add themselves to a waiting list, or view various courses. Data will also be sent to faculty and administrative users when they interact with the system.

**Communications Interfaces**

This product will require e-mail, web browsers, and internet network server communication. E-mail verification is not needed to create a user account and provide information about course enrollment. The HTTP communication standard will be utilized, and communication security of personally identifiable information must be considered. Encryption of users’ personal information like names, social security numbers, and addresses should protect user account information. Database queries and course availability should be synchronized as the university course catalog is filled and depleted.

**System Features**

These features are organized in a hierarchical priority list, with the first system feature being the highest priority.

**Agnostically Coded – Users should access the system from and Windows, Macintosh, and Linux desktop Operating Systems**

4.1.1 Description and Priority

The software must be agnostically coded. The system should be available on any Windows, Macintosh, and Linux desktop Operating System. Priority level 9.

4.1.2 Stimulus/Response Sequences

Users should navigate to the system website or access the platform via loading the applicable desktop application. The system will respond by granting them access and loading the relevant information.

4.1.3 Functional Requirements

REQ-1: Individualized graphical user interfaces should be developed for each type of user accessing the system (Student, Faculty, Administrative, and Guest). The screen size and layout of the GUI will be tailored based on screen size.

REQ-2: GUIs should be interactive to allow users to navigate the application with a keyboard and mouse or touch screen device if applicable. According to Faiza (2021), in the article “The 12 Do’s and Don’ts of Web Design | Adobe XD Ideas,” “one of the top principles of good UX is to keep the interface consistent throughout the entire product. The overall look and feel of your website should be consistent across all of your site’s pages” (para. 2). The functionality should reflect this requirement.

**Users should query a database to enroll in classes and register a new account/profile. Each new user should have a unique ID associated with a password. The system should guard against two users using the same ID for registration. Profiles must include some key information about the applicant including name, phone, e-mail, and any other information you may see necessary. Post registration, users can login to the system at any time using the ID and the password created during the registration process.**

4.2.1 Description and Priority

Users should be able to create accounts with varying privilege levels and unique user IDs. Priority level 8.

4.2.2 Stimulus/Response Sequences

Users create an account, and the system responds without requiring a verification method (e-mail). The system will provide the user with account status and other applicable information.

4.2.3 Functional Requirements

REQ-1: Users automatically are assigned to a guest account when accessing the system for the first time. Users will have the option to create an account or log in to an existing account. The system must protect against duplicate user IDs.

REQ-2: General student users, administrative users, and faculty users who have already created an account can save and edit profile information on the system.

REQ-3: Administrative level accounts must be provided for developers, testers, and other users requiring administrative privileges.

REQ-4: Faculty-level user accounts must be able to edit course information, add students, drop students, and manage course content.

**The system should allow for online courses to run through three semesters per year (spring/summer/fall), and students can list the course that will be offered during any semester, as not all courses will be offered in every semester. Each course should have a maximum number of enrollments that may be different depending on the course.**

4.3.1 Description and Priority

The software functions should allow users to perform queries to a database for the EasyEnroll Online Course Software System. Priority level 7.

4.3.2 Stimulus/Response Sequences

Users should be able to input queries into the EasyEnroll Online Course database functions. The system will respond by providing them with course descriptions and enrollment capabilities. Users can list courses over the three semesters based on course relevance and availability. The database will respond by investigating the available “seats” in the course and allowing the users to enroll in the courses.

4.3.3 Functional Requirements

REQ-1: The database should populate and remove courses based on availability to each user.

REQ-2: Query language and indexing methods will be used to generate relevant course search results. Users can only enroll in courses and drop courses as they wish.

**If a user wants to enroll into a course and the course is full, the student can add themselves onto a waiting list. A user can also cancel the enrollment from any course that they are enrolled in, and the system should inform the first in the waiting list (if any) that they can enroll into the class.**

4.4.1 Description and Priority

The software should provide users with unique information based on if course “seats” are available. If courses are full, a functionality to be added to a waiting list should be available. Priority level 6.

4.4.2 Stimulus/Response Sequences

Users regularly use the system to search for courses and enroll in them. Through table tagging, the system should respond by generating course waiting list results and providing applicable course availability notifications if students drop from a full course.

4.4.3 Functional Requirements

REQ-1: The system will utilize item tagging and indexing to provide waiting lists for students if a course is full.

REQ-2: Push notifications should be sent to the user to alert them of courses that become available that they are currently to open for availability.

REQ-3: General student users should be able to apply to waiting lists and drop courses that correlate to waiting lists.

REQ-4: If students drop from a course that has another student on its applicable waiting lists, the system will notify them that the course is available.

**Other Non-functional Requirements**

**Performance Requirements**

The user should experience limited processing and loading times when utilizing the system through regular use. The speed and performance of the system should be highly reliable and exceptionally fast for the user. According to Lyubov (2020), in the article “Non-Functional Requirements: A Guide With Concrete Examples,” “performance requirements determine how well a system should perform and whether or not it meets expectations. A system’s performance is mainly determined by its throughput and response time” (para. 4). Requests made throughout the enrollment and account creation functions should happen within seconds, if not instantaneously. The user should not notice any lag or loss of bandwidth when utilizing the system.

**Safety Requirements**

Users should agree to avoid prolonged usage of the system to prevent eye strain and other adverse health-related effects from using the product. Users should be aware that the damage or loss of security related to the information systems could leak personal information.

**Security Requirements**

User identity will not be authenticated through an attached e-mail to the account. Because this software system will be utilized on local machines, there are not any significant security concerns related to user information. All user information input into the system is fictional and does not have any associated risk. If personal information is entered into a local machine, user information should be protected from malicious hacking or data breaches. User information such as social security numbers and addresses must be considered sensitive and not publicly available. Security and privacy certifications to ensure the safeguarding of this data must be satisfied.

**Software Quality Attributes**

Systems should be highly available and reliable in that unscheduled downtimes should be limited, and scheduled downtimes for system maintenance should avoid peak usage periods. Because this system will be run on a local machine, the user is responsible for ensuring their equipment is operational. Interfacing with external clients through application programming interfaces (API) or another coding mechanism should enable interoperability. The software development will utilize new and emerging process methodologies throughout design and development. According to Tsui, Karam, & Bernal (2018), in the course textbook “Essentials of software engineering (4th ed.),” “agile processes are a family of software development methodologies that produce software in short iterations and allow for greater changes in design” (p. 84). Along with agile methodology utilization, the system will favor ease of use over ease of learning. The software must exhibit an easy-to-use GUI and be made accessible by disabled users. Features like multiple languages, more extensive font settings, and text-to-speech functions will enable more individuals to access the platform. The software should also be highly testable so functional requirements can be thoroughly and comprehensively implemented.

**Business Rules**

Administrative users will have the right to manipulate the layout, design, testing, and overall system development. General users and guest users will not have extra functionalities under any previously mentioned circumstances.

**Other Requirements**

Database requirements such as indexing and tagging methods are not specified throughout this SRS, but development will be necessary for the system’s functionalities. The product may require Internationalization policies and procedures to be used worldwide. Legal documentation like product copyright requirements will be necessary to validate the actual enrollment of courses on the platform. This product should be highly reusable for future developments and building requirements.

**Appendix A: Glossary**

API – Application Programming Interface

GUI – Graphical User Interface

HTML – HyperText Markup Language

HTTP – Hypertext Transfer Protocol

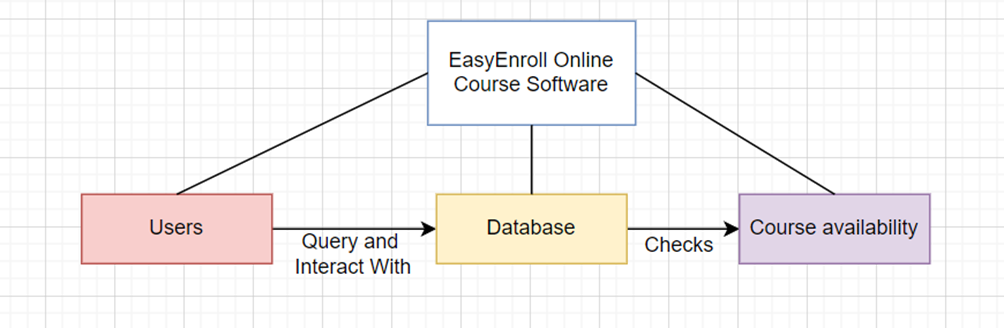
REQ - Requirement

SRS – Software Requirements Specification

TBD – To Be Determined

**Appendix B: Analysis Models**

Figure 1 eCommerce Shopping Software Concept Map



**Appendix C: To Be Determined List**

N/A

**References**

Faiza, J., (2021). The 12 Do’s and Don’ts of Web Design | Adobe XD Ideas. Ideas. https://xd.adobe.com/ideas/principles/web-design/12-dos-donts-web-design-2/

Lyubov, T., (2020). Non-Functional Requirements: A Guide With Concrete Examples - Plutora. Plutora. https://www.plutora.com/blog/non-functional-requirements-guide

Paol, F., (2019). IONOS Digital Guide. XAMPP tutorial: installation and first steps. IONOS Digitalguide. https://www.ionos.com/digitalguide/server/tools/xampp-tutorial-create-your-own-local-test-server/

‌Tsui, F., Karam, O., & Bernal, B. (2018). Essentials of software engineering (4th ed.). Jones & Bartlett Learning.

‌