**Software Requirements Specification for Registration System Project**

Steven On

The University of Arizona Global Campus

CST 499: Capstone for Computer Software

Professor Amjad Alkilani

May 9, 2022

# Table of Contents

[Table of Contents 2](#_Toc100695305)

[Revision History 3](#_Toc100695306)

[1. Introduction 4](#_Toc100695307)

[1.1 Purpose 4](#_Toc100695308)

[1.2 Document Conventions 4](#_Toc100695309)

[1.3 Intended Audience and Reading Suggestions 4](#_Toc100695310)

[1.4 Product Scope 4](#_Toc100695311)

[1.5 References 4](#_Toc100695312)

[2. Overall Description 5](#_Toc100695313)

[2.1 Product Perspective 5](#_Toc100695314)

[2.2 Product Functions 6](#_Toc100695315)

[2.3 User Classes and Characteristics 6](#_Toc100695316)

[2.4 Operating Environment 7](#_Toc100695317)

[2.5 Design and Implementation Constraints 7](#_Toc100695318)

[2.6 User Documentation 7](#_Toc100695319)

[2.7 Assumptions and Dependencies 7](#_Toc100695320)

[3. External Interface Requirements 8](#_Toc100695321)

[3.1 User Interfaces 8](#_Toc100695322)

[3.2 Hardware Interfaces 8](#_Toc100695323)

[3.3 Software Interfaces 8](#_Toc100695324)

[3.4 Communications Interfaces 9](#_Toc100695325)

[4. System Features 9](#_Toc100695326)

[4.1 System Feature 1 9](#_Toc100695327)

[4.1.1 Description and Priority 9](#_Toc100695328)

[4.1.2 Stimulus/Response Sequences 9](#_Toc100695329)

[4.1.3 Functional Requirements 10](#_Toc100695330)

[4.2 System Feature 2 (and so on) 11](#_Toc100695331)

[5. Other Nonfunctional Requirements 11](#_Toc100695332)

[5.1 Performance Requirements 11](#_Toc100695333)

[5.2 Safety Requirements 11](#_Toc100695334)

[5.3 Security Requirements 11](#_Toc100695335)

[5.4 Software Quality Attributes 11](#_Toc100695336)

[5.5 Business Rules 12](#_Toc100695337)

[6. Other Requirements 12](#_Toc100695338)

[Appendix A: Glossary 12](#_Toc100695339)

[Appendix B: Analysis Models 12](#_Toc100695340)

[Appendix C: To Be Determined List 12](#_Toc100695341)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| Steven On | April 11, 2022 | First Draft | 1.0 |
| Steven On | May 9, 2022 | Updates After Development | 2.0 |

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to define the Software Requirements Specification (SRS) for an Enrollment Registration System (ERS) for higher education.

## 1.2 Document Conventions

Higher-level requirements are assumed to be inherited by detailed requirements.

## 1.3 Intended Audience and Reading Suggestions

The intended audience for this SRS is for college use, specifically for the CST 499 capstone course for Computer Software Technology program. According to Tsui et al. (2018), “more-detailed functional requirements must be elicited as these high-level statements put the proper customer expectations in place” (p. 111). As the CST 499 course progresses, the intended audience may change as requirements start to unravel that could be helpful for industry professionals.

## 1.4 Product Scope

The ERS system allows students to create a profile for course registration. Students will be able to select available courses while providing perinate contact information with the potential of integrating with a Learning Management System (LMS). The university intends to use the ERS as an automated system to assist with course registration and scheduling which is not limited to the availability of faculty and/or administrative staff. Additionally, students should be able to remove any scheduled course from their registered course.

## 1.5 References

Plateaux, A., Lacharme, P., Murty, K., & Rosenberger, C. (2013). Online user’s registration respecting privacy. *2013 World Congress on Computer and Information Technology (WCCIT), Computer and Information Technology (WCCIT), 2013 World Congress On*, 1–5. <https://doi.org/10.1109/WCCIT.2013.6618718>

Tsui, F., Karam, O., & Bernal, B. (2018). [*Essentials of software engineering*](https://ashford.instructure.com/courses/100130/modules/items/5077278) (4th ed.). Jones & Bartlett Learning.

Yu Peng, NaiJia Liu, YanXia Li, & ZhengLong Shao. (2012). Design and implementation of the online course registration system at Tsinghua University. *2012 International Conference on Systems and Informatics (ICSAI2012), Systems and Informatics (ICSAI), 2012 International Conference On*, 1179–1182. <https://doi.org/10.1109/ICSAI.2012.6223244>

# 2. Overall Description

## 2.1 Product Perspective

The ERS is a component to an overall university portal and not intended to be used in a different format such as a document repository or university communications. Furthermore, an ERS is not to be confused with an LMS. Because an LMS would contain the course content and curriculum to fulfill student learning upon course completion. Respectively, the intended use of the ERS is a microservice to handle student contact profile and course registration only.

**Figure 1**

*High-level ERS system architecture.*

Diagram

Description automatically generated

Figure 1 demonstrates the high-level ERS design where integration endpoints can be initiated at the User Profile level. Branding could also be integrated at the user interface level upon login. Note. The “Edit User Profile” functionality will not be reflected in the ERS design

## 2.2 Product Functions

* Cancel Course Registration
* ~~Cancel Course Waitlist Registration~~
* Course Enrollment Status
* Course Registration
* User profile creation
* ~~User profile editing~~
* User sign-in page.

## 2.3 User Classes and Characteristics

The most important class is the “User” class as it contains the most properties and the most methods to perform transactions such as creating a new user, editing user profiles, and registering for courses. According to Peng et al. (2012), the users are separated in three groups: (1) Student, (2) Teacher, (3) Administrator. The student will register, check, and cancel courses. Teachers can check course roster and allow students from the waitlist to be added to the course. Administrators controls the entire registration process, updates detailed information, and performs all support of the entire ERS system.

**Figure 2**

*ERS UML Class Diagram*

Diagram, schematic

Description automatically generated

Figure 2 demonstrates the “users” class along with the derived classes of student, faculty, and administrator. The intended use of Figure 2 is for reference to the design of the ERS model, as system development and implementation continues, the class diagram should be modified to reflect system changes for all technical and non-technical parties involved in the ERS project.

## 2.4 Operating Environment

For best performance, the ERS application should be used on the most current or first previous major release of Chrome, Firefox, Edge, or Safari. Because it’s built using web standards, the ERS application can be ran on Windows, Mac, Linux, iOS, Android, or any other device with a modern web browser.

## 2.5 Design and Implementation Constraints

The following listed items are considered out of scope for this project:

* Third-party integrations such as the setup of OTP for authentication and password reset emails.
* The integration of linking a student’s degree plan with the available course register for course registration.
* The integration of linking course pre-requisites with the available course register for course registration.
* LMS building is prohibited, and course content or user’s ability to view a course roster list will not be developed in this project.

## 2.6 User Documentation

All user documents including operational support, integration mapping, and regression test plans will be documented in Confluence.

## 2.7 Assumptions and Dependencies

The ERS is dependent on course availability and student enrollment. Therefore, the management of course availability must be updated prior to the registration date of all courses. Additionally, a possible assumption is to develop a mechanism to notify registered users about course cancellation due to low enrollment and third-party integration for student advisor assistance.

# 3. External Interface Requirements

## 3.1 User Interfaces

For the first implementation, the ERS will not use any third-party external integrations until functionality has been officially approved by the university director. Once requirements fulfillment has been approved, the actual product backlog for the ERS can be filled in forms of user stories for the software development team to start implementation for a real use case.

## 3.2 Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## 3.3 Software Interfaces

Internally, the following software interfaces will be achieved with the ERS:

* MySQL Database – Backend
* Apache - Backend
* PHP, HTML, CSS, and JavaScript - Frontend

## 3.4 Communications Interfaces

The preferred method of communications is https in a secure tunnel to the backend MySQL database. Internal calls to the MySQL database will be handled by PHP server-side code locally on the webserver itself. Additionally, all security protocols must follow the most recent AES encryption.

# 4. System Features

## 4.1 System Feature 1

New user registration and login.

## 4.1.1 Description and Priority

High priority: The ability for a user to create a new profile with a unique user ID.

## 4.1.2 Stimulus/Response Sequences

When a user selects create new account, the user can register for a new account by providing required property values such as first name, email, and phone number.

## 4.1.3 Functional Requirements

REQ-1: New user registration that includes account and profile creation.

REQ-2: 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.

REQ-3: Profiles must include some key information about the applicant including name, phone, email, and any other information you may see necessary.

REQ-4: Post registration, users can login to the system at any time using the ID and the password created during the registration process.

## 4.2 System Feature 2

Course enrollment process.

## 4.2.1 Description and Priority

High priority: Any registered user can request, check, or cancel enrollment for available courses.

## 4.2.2 Stimulus/Response Sequences

A user can search the course catalog for available courses to register for. When the user registers for a course, the number of enrollments to that course increases and displays the maximum number of enrollments. ~~Additionally, users can be asked to be added to the course waitlist pending professor approval to allow the student into the course.~~ At any time, users can cancel course enrollment.

## 4.2.3 Functional Requirements

REQ-5: Online courses run through three semesters per year (spring/summer/fall), and students can list the courses that will be offered during any semester, as not all courses will be offered in every semester.

REQ-6: Each course should have a maximum number of enrollments that may be different depending on the course.

~~REQ-7: If a user wants to enroll into a course and the course is full, the student can add themselves onto a waiting list.~~

REQ-8: 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.~~

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

The system should be able to reserve student enrollments in a first come, first-serve approach. The registration should be able to handle multiple message updates and determine student registration requests in nanoseconds. The database updates should be a quick response for students.

## 5.3 Security and Privacy Requirements

Students or faculty in witness protection or confidential profiles should be able to be hidden with approval. According to Plateaux et al. (2013), “Anonymity and pseudonymity preserve the users’ privacy during a connection to commercial sites… The anonymity ensures the user can access to information without revealing his/her identity, whereas the pseudonymity requires the user is responsible for such use” (p. 2). Therefore, sensitive privacy content even accidentally added in the system should have the ability to be removed by an administrator and by the user themselves to protect student and faculty privacy.

## 5.4 Software Quality Attributes

The overall objective of finding software quality depends on the useability that might be accepted during UAT. Initial testing activities are responsible for meeting requirement acceptance criteria however overall quality depends on the combined persona of an end user, a stakeholder representative, and the project sponsor. All three personas must be one to fulfill maximum software quality specifications defined by the individual that created to user story.

## 5.5 Business Rules

The following business rules must be considered during the ERS development stages:

* Students cannot retake courses they’ve already completed.
* Students should be working toward their degrees therefore, providing only classes toward their degree is essential to their success.
* Students should be able to Sign-off to protect their registration and their identity.
* Students cannot edit their own profiles; they must work with their academic success advisors to edit their profiles.

# 6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

# Appendix A: Glossary

|  |  |
| --- | --- |
| TERM/ACRONYM | DEFINITION |
| ERS | Enrollment Registration System |
| LMS | Learning Management System |
| OTP | One Time Password |
| UAT | User Acceptance Testing |

# Appendix B: Analysis Models

Optionally, the following UML models can be used for the ERS design:

**Figure 3**

*ERS User login Process – UML State Diagram*

Diagram

Description automatically generated

**Figure 4**

*ERS Roles – UML Use Case Diagram*

Diagram, schematic

Description automatically generated

**Figure 5**

*ERS Functionality – UML Sequence Diagram*

Graphical user interface

Description automatically generated with low confidence

**Figure 6**

*ERS Security Embedded Login – UML Activity Diagram*

Diagram

Description automatically generated

# Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>