Software Requirements Specification

for

Online College Enrollment System

Version 1.0 approved

Prepared by Jay Figueroa

Bacas Development (Contracted by UBGC)

August 29, 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** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

This system is designed to meet the enrollment needs of the online school UBGC – version 1.0.0. The scope of this project is only to provide necessary functionality for a complete enrollment system within the larger UBGC school site.

## Intended Audience and Reading Suggestions

This document is submitted for review by the ABGC development team responsible for the creation of the school site and all associated management. Sections relevant for management include the Overall Description (Section 2) and System Features (Section 4). The development team is expected to read and understand the rest of the document in order to grant final approval.

## Product Scope

The scope of this project is limited to the enrollment system for the school. This includes auxiliary systems necessary to allow for this such as student profile creation and sign-in, course schedules, and waitlists. This system must allow students to sign up for courses while automatically managing class size maximums.

## References

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

“Website Security.” (May 10, 2022). Mozilla. Retrieved from https://developer.mozilla.org/en-US/docs/Learn/Server-side/First\_steps/Website\_security

# Overall Description

## Product Perspective

UBGC has no existing site being a new University, and this system will provide necessary functionality in its integration into the larger software. The information submitted by the user in the new student sign-up page will be accessed by many other systems within the site such as transcript, billing, advising, and classroom. This means that the student table within the database is the primary table that other systems will interface with in order to obtain relevant information—requiring a complete record of information on each student. This is account creation is one of the major subsystems of Enrollment along with Course Catalog detailing options for the next three semesters, and Classroom, which lists students enrolled or on the waitlist for each class. Finally, there is a small automation responsible for notifying waitlisted students that they have been added to a course if an existing student cancels their enrollment in said course.

## Product Functions

There are several primary functions of the outlined system:

* Allow user to create student account
  + Must include details such as full name, date of birth, address, phone, email, and a selected password
  + Students are automatically assigned a student ID
* Students must be able to view courses available in the upcoming semesters
* Students choose the course numbers they are interested in enroll
  + If there is availability for selected course, they are added to the classroom
  + If the course has reached maximum occupancy, the student is added to waitlist
* Students will have the option to cancel an enrollment at any point leading up to the first day of class.
  + Any cancellation will automatically move the first waitlisted student into the class
    - Notification of said change is sent to newly added student

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

As this is part of a larger school site, the planned environment for main functionality will be within a browser and therefore requires operability on all platforms. Additionally, UBGC will maintain a database on its campus for all students and associated records, including all of those within this system. The hardware requirements for this database are beyond the scope of this project.

## User Documentation

The site should contain instructions hyperlinked to the enrollment homepage that guide the course selection process. This will include images of the interface and a list of the courses required for each major (not provided within this project).

# External Interface Requirements

## User Interfaces

Each page of the site will contain a running header of options for the user including the site’s landing page and the sign-in/registration page. When The user is signed in, the sign-in page link will be replaced with the user profile page link. Additionally, the enrollment page will only be available when the student is signed in.

**Landing Page** – This is the site’s primary page providing some background and current events surrounding the school.

**Registration Page** – This page will contain a simple form where the user will provide the information listed in 2.2. Upon completion, the user will hit the submit button to confirm their choices.

**Sign-in Page** – This page will contain the assigned student ID/password fields along with a submit button. Additionally, it will also contain a link to the Registration Page.

**Enrollment Page** – This page will provide an overview of the enrollment instructions and will contain a link to the full instructions. It will also contain a link to the list of upcoming courses and their descriptions (descriptions fall outside the scope of this document). Below this is the registration window (html form) with several fields. The Semester button will contain a dropdown list of the three semesters in the current year. The course name and number field will contain a dropdown list of all courses offered in the selected semester. Finally, the submit button will allow a user to confirm their choice, which will be added to their schedule in their profile page.

**Profile Page** – This page will display the student’s information provided when they signed up. It will also list their enrolled and waitlisted courses, with the waitlist number next to applicable ones.

## Software Interfaces

This system will rely on cooperation with the school’s database system (MySQL). The database will have interfaces with many other systems outside the scope of this document, but its creation and communicability are one of the key responsibilities of this project. Primary concern within the database is the student table, which contains all student’s ID and personal information. The Student ID will be the foreign key in most of the systems for the software as whole. The final scripting language for the web development end of the project is yet to be determined, but likely PHP.

# System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

## Registration

4.1.1 Description and Priority

Allows users to register as students and assigns them a student ID used throughout the site. Priority 9

4.1.2 Stimulus/Response Sequences

User must navigate to sign-in/registration page and fill out required information before submitting to database.

4.1.3 Functional Requirements

REQ-1: Connection between Database and site.

## Course Enrollment

4.1.1 Description and Priority

Allows users to select courses which they would like to participate in and automatically adds them to a waitlist if class is at capacity. Priority 9

4.1.2 Stimulus/Response Sequences

User selects course name and semester from dropdown options and presses submit when prepared.

4.1.3 Functional Requirements

REQ-1: database connection (particularly to student table)

REQ-2: course catalog

# Other Nonfunctional Requirements

## Security Requirements

Primary security concern is centered around personal data stored in database when students register their accounts. The protection of this information is private, so it is imperative to eliminate risks where possible. In order to avoid SQL injection attacks, the enrollment page only had dropdown options for classes and no fields in which students can supply their own text. Additionally, character limits are placed on forms for registration and sign-in pages (Website Security, 2022).

## Software Quality Attributes

This system will remain as adaptable as possible by only modularizing each portion of its functionality into separate classes. The student information contained in the database is its own class, as are the courses in the catalog. Availability is outside the scope of this project. Correctness is ensured by imposing chartype restrictions on information supplied by students and confining input to format restrictions. Usability is maintained through simplicity of the system. The separation of classes allows for reusability, testability, and maintainability by reducing the dependence of modules throughout the system and allowing each to be modified in future iterations of the system.