**Software Requirements Specification: SRS Document Development**

Bill Maggs

The University of Arizona Global Campus

CST499: Capstone for Computer Software Technology

Dr. Charmelia Butler

October 30th, 2024

A Software Requirements Specification (SRS) is a comprehensive document that outlines the functional and non-functional requirements of a software system. It serves as a blueprint for developers, testers, and stakeholders to understand what the software will do and how it will operate. The purpose of writing an SRS is to ensure clear communication between all parties involved in the development process and to minimize the risk of misinterpretation or missing key requirements by allowing stakeholders to accurately describe their needs and for developers meet the requirements of the stakeholders.

An SRS provides a clear, agreed-upon understanding of what the software will accomplish. It ensures that the client's expectations are met by documenting all functionalities and constraints. The SRS serves as a foundation for project planning, design decisions, coding guidelines, and testing strategies. By detailing every aspect of the software, the SRS reduces the likelihood of misunderstandings between stakeholders (e.g., users, developers, project managers). It helps prevent scope creep by clearly outlining what is included in the project and what is not. The SRS serves as an action plan for all stages of the project from development to maintenance (Perforce, 2023).

There are multiple templates available for the SRS. Outlines and components from Weigers (Weigers, 2002), Perforce (Perforce, 2023), and IEEE (IEEE, 1984) will be integrated where appropriate. Sections one and two will be adopted from Weigers and Perforce, Section three will be the specific requirements section adopted from IEEE.

The key requirements as given in the assignment (Butler, 2024) can be categorized in the SRS for the University Enrollment System into four categories: Functional Requirements, Non-Functional Requirements, Interface Requirements, and Constraint Requirements. These will be translated into concrete requirements in the respective sections of the SRS, with the following summaries. Other sections considered to be relevant for context of the SRS, primarily sections one and two, will also be maintained.

The Functional Requirements/System Features will consist of a new user registration, including profile information and process for assigning a user ID. There will also be requirements for Login Functionality, Course Listing by Semester, and Course Enrollment options. The system must allow new users to register by creating an account and profile. Each new user must have a unique ID associated with a password. The system must prevent two users from using the same ID during registration. Profiles must include name, phone number, email, and other necessary information such as address and date of birth. Post-registration, users must be able to log in to the system using their unique ID and password. Users must be able to list the courses offered during any of the three semesters (spring, summer, and fall), as not all courses are available in every semester. Each course should have a maximum enrollment capacity that may vary depending on the course. Users must be able to enroll in courses based on availability. If a course is full, users must be able to add themselves to a waiting list. When a seat becomes available, the system must notify the first user on the waiting list that they can enroll. Users must be able to cancel their enrollment in any course they are registered for. Upon cancellation, the system must notify the first person on the waiting list (if applicable) about the availability of the seat.

Non-functional requirements will define security, performance, and scalability. The system must securely manage user accounts and prevent unauthorized access using encryption for sensitive data like passwords. The system must handle high traffic during peak times (such as course registration periods) without significant delays. The system must scale to support thousands of simultaneous users during enrollment periods.

The External Interface/User Interfaces will consist of a web interface, with consideration being given to a mobile interface from a phone or tablet. Users must interact with the system through an intuitive web interface for actions like registration, course enrollment, and account management. Users should also have access to the system via a mobile-friendly interface for course registration and account management.

For Constraints, the system must ensure that all user IDs are unique and cannot be duplicated. Each course must have a predetermined maximum enrollment limit that can vary based on the course. Not all courses will be offered every semester, and this constraint must be considered during course listings and enrollment.

These requirements help ensure that the system is well-organized, meets user needs, and complies with operational constraints. The following formal SRS for this system will clearly outline all these requirements to ensure the system is developed correctly, fulfills its intended purpose, and aligns with the university's operational needs.

**References:**

Butler, C. (2024). *SRS Document Development and Github*. UAGC. Retrieved from [Week 1 - Assignment (instructure.com)](https://uagc.instructure.com/courses/137971/assignments/2667775?module_item_id=7033166)

Weigers, K. (2002). *SRS Template*. Process Impact. Retrieved from <https://exinfm.com/training/M2C3/srs_template.doc>

Perforce (2023) *How to Write an SRS Document (Software Requirements Specification Document).* Perforce. Retrieved from [How to Write an SRS (Software Requirements Specification Document) (perforce.com)](https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document)

IEEE (1984). *IEEE Guide for Software Requirements Specifications.* *IEEE Std 830-1984* , vol., no., pp.1-26, 10 Feb. 1984, doi: 10.1109/IEEESTD.1984.119205.

**Software Requirements Specification**

**For**

**University Enrollment System**

Version 1.1 Approved

Prepared by William Maggs

On behalf of UAGC

10/30/2024

**Table of Contents**

Table of Contents ……………………………………………………………………………………………………. 2

Revision History ……………………………………………………………………………………………………. 3

1. Introduction ……………………………………………………………………………………………………. 4

1.1 Purpose ……………………………………………………………………………………………………. 4

1.2 Document Conventions and Organization …………………………………………………………. 4

1.3 Intended Audience and Reading Suggestions …………………………………………………………. 4

1.4 Project Scope ……………………………………………………………………………………………………. 4

1.5 References ……………………………………………………………………………………………………. 4

2. Overall Description …………………………………………………………………………………………..…. 4

2.1 Product Perspective …………………………………………………………………………………………….. 4

2.2 Product Features …………………………………………………………………………………………………. 4

2.3 User Classes and Characteristics …………………………………………………………………………… 5

2.4 Operating Environment …………………………………………………………………………………………. 7

2.5 Design and Implementation Constraints …………………………………………………………………. 7

2.6 User Documentation ……………………………………………………………………………………………. 7

3. Specific Requirements …………………………………………………………………………………………. 7

3.1 Functional Requirements ……………………………………………………………………………………… 7

3.1.1 New User Registration ………………………………………………………………………………………….. 7

3.1.2 Login Functionality ………………………………………………………………………………………………. 8

3.1.3 Course Listing by Semester …………………………………………………………………………………... 8

3.1.4 Course Enrollment ………………………………………………………………………………………..…….. 8

3.2 Non-Functional Requirements ………………………………………………………………………………. 9

3.2.1 Security ……………………………………………………………………………………………………………… 9

3.2.2 Performance ………………………………………………………………………………………………………. 9

3.2.3 Scalability ………………………………………………………………………………………………………….. 9

3.3 Interface Requirements ……………………………………………………………………………………….. 9

3.3.1 Web Interface …………………………………………………………………………………………………….. 9

3.3.2 Mobile Interface …………………………………………………………………………………………………… 9

3.4 Constraint Requirements …………………………..…………………………………………………………. 9

3.4.1 User Identification ……………………………………………………………………………………………….. 9

3.4.2 Enrollment Limits ………………………………………………………………………………………………… 9

3.4.3 Course Offerings per Semester ……………………………………………………………………………… 9

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| William Maggs | 10/7/24 | Original Release | 1.00 |
| William Maggs | 10/30/24 | References updated to match citations. Functional Requirements of SRS updated to match academic paper content. | 1.10 |

**1. Introduction**

**1.1 Purpose**

This document will identify the concrete requirements for a university enrollment system. The initial release will be version 1.00, with subsequent releases identified in the revision history block.

**1.2 Document Conventions and Organization**

Following IEEE std 830-1984 (IEEE, 1984), the remainder of this document is organized in two following sections. Section 2 will cover the Overall Description to include product perspective, product features, user classes and characteristics, operating system, design and implementation constraints, and user documentation.

Figures where used will be annotated following IEEE guidelines.

**1.3 Intended Audience and Reading Suggestions**

This document is intended for University Staff, Project Managers, Developers, Marketing Staff, Users, Testers, and Documentation writers. This document is initially intended to be read in full.

**1.4 Project Scope**

The software being specified is a system for university student enrollment, with a focus on improving operational efficiency, user satisfaction, and the alignment of the system with broader institutional goals. From a business perspective, the benefits will be higher operational efficiency, cost reductions in manual services, data-driven decision making, and improved resource utilization while increasing enrollment capacity. Users will gain an enhanced user experience, accessibility, real-time notifications, and self-service capabilities.

**1.5 References**

IEEE (1984). *IEEE Guide for Software Requirements Specifications.* *IEEE Std 830-1984* , vol., no., pp.1-26, 10 Feb. 1984, doi: 10.1109/IEEESTD.1984.119205.

**2. Overall Description**

**2.1 Product Perspective**

This system is being designed as a new system for a university enrollment program. This section does not state the specific requirements. It is intended to give clarity. This system is intended to be used as a stand-alone system.

**2.2 Product Features**

In the University Enrollment System, students provide their information during registration, search for courses, and enroll or cancel courses. The registration system handles new user creation, profile management, and login information. The course management system provides course listings, manages course availability by semester, and tracks enrollment limits. The enrollment and waitlist system handles the core functionality of student course enrollment, waitlist management, and reallocation of seats. The notification system sends real-time updates to students (e.g., enrollment confirmation, waitlist status, course availability). University administrators will be able to access data reports and analytics for decision-making.

A top-level data flow diagram is provided in Fig 1.

A diagram of a course management system

Description automatically generated

Fig 1

A Top-Level Data flow diagram

**2.3 User Classes and Characteristics**

There are six distinct user classes for the system, each with their own characteristic needs to be fulfilled.

2.3.1 Students

This class will consist of undergraduate, graduate, and part-time students with varying technical proficiencies. The primary objective of students is to register for courses, manage their schedules, and access academic information (e.g., grades, course catalogs). Students expect a seamless, intuitive interface that allows them to perform tasks like course registration, profile updates, and schedule management independently. A significant portion of students may prefer using mobile devices, so a responsive and mobile-friendly interface is essential. Students rely on timely notifications for course availability, deadlines, waitlist status, and other important updates.

2.3.2 Faculty/Instructors

Faculty members generally have a higher level of comfort with online systems, especially for managing course-related tasks. Faculty need to view and manage course rosters, handle student enrollment requests, and post grades. Instructors often have limited time to navigate the system, so a straightforward interface with clear functionalities is crucial. Faculty may require access to reports on student enrollments, waitlists, and course performance data. Some faculty may also have roles as department chairs or program coordinators, needing higher-level system access for curriculum planning or course scheduling.

* + 1. Administrative Staff

Administrative users are typically well-versed in managing institutional systems and handling complex workflows. Staff members have extensive access to the system and are responsible for managing course offerings, handling special enrollment cases, and troubleshooting user issues. Administrators often need to generate and analyze reports on student enrollment patterns, course demand, and waitlists to assist with academic planning. They must ensure that the system complies with university policies and external regulations (e.g., FERPA), particularly concerning data privacy and student records. Administrative staff assist users (students and faculty) with account issues, enrollment errors, or other system-related challenges.

* + 1. IT/Technical Support

IT staff have access to the technical backend of the system and are responsible for maintaining its uptime, security, and performance. They handle technical aspects such as server management, software updates, bug fixes, and integrations with other university systems (e.g., student information systems, financial aid systems). IT personnel may need to address system errors or performance issues reported by students, faculty, or administrative staff. This user class is responsible for ensuring the security of the system, safeguarding against data breaches, and maintaining compliance with security protocols and standards.

* + 1. Department Heads/Program Administrators

Department heads and program coordinators may not use the system as frequently as administrative staff but require access for curriculum planning and course approvals. These users oversee course offerings within their departments and may need to approve or adjust course capacities, scheduling, and instructor assignments. Department heads may require access to detailed reports on student enrollments, course demand, and academic performance for strategic planning. These users rely on system data to make informed decisions regarding faculty workload distribution and course availability across semesters.

2.3.6. University Administrators

These high-level administrators need access to system-wide reports to inform institutional planning and resource allocation. While university administrators may not use the system daily, they require insights from the system for high-level decision-making. They rely heavily on reports and analytics for monitoring overall enrollment trends, course demand, and institutional performance. This user class ensures that the system supports university-wide policies, including academic guidelines, student services, and resource planning.

**2.4 Operating Environment**

The system is intended to be operated from any standard browser hosted by either a personal computer or mobile device, with the initial release requiring operation on a personal computer running a Windows operating system utilizing an Edge browser.

**2.5 Design and Implementation Constraints**

For successful development and deployment there are multiple constraints which must be considered. First, scalability is crucial, as the system must handle high user traffic, especially during peak registration periods, without performance degradation. Security and data privacy are paramount, requiring compliance with regulations like FERPA to protect sensitive student information, which necessitates encryption, secure authentication, and role-based access controls. Additionally, the system must attempt to be mobile-friendly, ensuring compatibility with various devices and browsers for a broad range of users. Lastly, there are time and resource limitations, as development timelines, budget, and personnel must be efficiently managed to meet project goals within university requirements.

**2.6 User Documentation**

User documentation will include a PowerPoint presentation with a recorded video presentation.

**3. Specific Requirements**

Each of the requirements are to be considered concrete, with an equal priority. Release will not be considered until all requirements have been met. The requirements will be organized by functional, non-functional, interface, and constraint requirements.

**3.1 Functional Requirements**

3.1.1 New User Registration

The system must allow new users to register by creating an account and profile. The new user registration system will create and assign a unique user identification number by the mySQL database. A user password will be created by the user during the new user registration.

3.1.1.1 Inputs will be user profile information to include fields for first name, last name, address, phone number, email, and password.

3.1.1.2 Processing will include assigning a unique user ID by the mySQL system as a primary key, storing the user profile and password in a user table.

3.1.1.3 Output will be a verification to the user that registration was successful.

3.1.2 Login Functionality

The system will require a user to be registered and logged in to access the course catalog, and course functionality

3.1.2.1 Inputs will user ID and password.

3.1.2.2 Processing will be verifying credentials and allowing access to the course registration pages.

3.1.2.3 Output will be redirection to the course registration system as a logged in user.

3.1.3 Course Listing by Semester

Courses will be listed in three possible semesters consisting of spring, summer, and fall. Constraints to course offerings will be listed in 3.4.2.

3.1.3.1 Inputs will be the table of available courses for the given semester.

3.1.3.2 Processing will be calculating the number of open seats.

3.1.3.3 Outputs will be a listing by semester of seats available.

3.1.4 Course Enrollment.

Students will be able to enroll in any available course. Constraints to enrollment are listed in 3.4.2. If a desired course if full, students may add themselves to a waiting list. Students may cancel enrollment from any currently enrolled course. The system will then inform the first student on the waiting list of availability.

3.1.4.1 Inputs are the requested course for enrollment or waitlisting, or cancelation of an existing course.

3.1.4.2 Processing will be determining if there is space available in the requested course. If there is space available, the student will be enrolled. If there is not space available, the student will have an option of being placed on a waiting list. If a student cancels a registration or name from a waiting list, they will be removed and the first person on a waiting list notified of an opening.

3.1.4.3 Output will be a confirmation that registration or cancellation is successful, or a waitlist entry has been created.

**3.2 Non-Functional Requirements**

3.2.1 Security

3.2.1.1 The system will require registration and login. Access control will be accomplished through role-based access control (RBAC), with the user table including a field for access rights with additional access for administrators and instructors.

3.2.2 Performance

3.2.1.2 With the lack of metric there is no specific performance requirement.

3.2.3 Scalability

3.2.1.3 The system will be able to scale to thousands of users.

**3.3 Interface Requirements**

3.3.1 Web Interface

3.3.1.1 The system will require compatibility with the Windows operating system and Edge browser for remote students’ access.

3.3.2 Mobile Interface

3.3.2.1 The initial release will give high consideration for mobile interfaces.

**3.4 Constraint Requirements**

3.4.1 User Identification

3.4.1.1 On successful registration, the user will be assigned a unique user ID based on a mySQL table process.

3.4.2 Enrollment Limits

3.4.2.1 Active enrollment will be limited to the number of seats allowed by the faculty/instructors. The system must automatically calculate the available seats from the mySQL table of capacity and filled columns.

3.4.3 Course Offerings per Semester

3.4.3.1 The number of courses offered each semester is not required to be the same. Couse offering constraints will be determined by the administrative staff and reflected in the enrollment system.