Software Requirements Specification

for

Class Enrollment System

Version 2.0 approved

Prepared by Edmon Adams, Joshua Jones, and Trevor Fellbaum

Team 1

08/08/2017

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Team 1	6/7/2017	Original Version	1.0
Team 1	8/8/2017	Final Updated Version	2.0

1. Introduction

1.1 Purpose

This SRS describes the entire system of mainly increment 1 with mentions of increments 2 and 3 of this Class Enrollment Software.

1.2 Document Conventions

Every feature has its own stated priority in the description and priority subsection in section 3. Each requirement is to have its priority assumed by the priority of the feature and scope of the requirement.

1.3 Intended Audience and Reading Suggestions

This document is intended for developers and potential clients who wish to use this software and anyone involved in the sale process. We recommend both developers and non-developers to start with **1. Introduction**, and then the developers should read the section in order starting with **1. Introduction** and ending with **6. Other Requirements**. Non-developers, however, should proceed from **1. Introduction** out of order. The recommended sequence is **1. Introduction**, **4. System Features**, **3. External Interface Requirements**, **5. Other Nonfunctional Requirements**, **2. Overall** Description, and then **6. Other Requirements**. This order saves the more technical material for the last few readings and prioritizes the actual non-technical functionality of the software first.

1.4 Product Scope

This software is a class enrollment software intended for post-secondary institutions. The goal of this software is to automate the work of academic advisors and provide students with a reliable source of information about the status of their degree and an intuitive and easy to navigate user interface. This software will help the student navigate the directory of available classes by prioritizing their credit requirements for their specific majors, minors, and colleges. The classes that are displayed first are also recommended for that student based on factors such as class availability, classes previously taken, and what credits they need for graduation. Again, the goal of this software is to automate academic advising and provide students with the resources needed to complete their requirements for graduation.

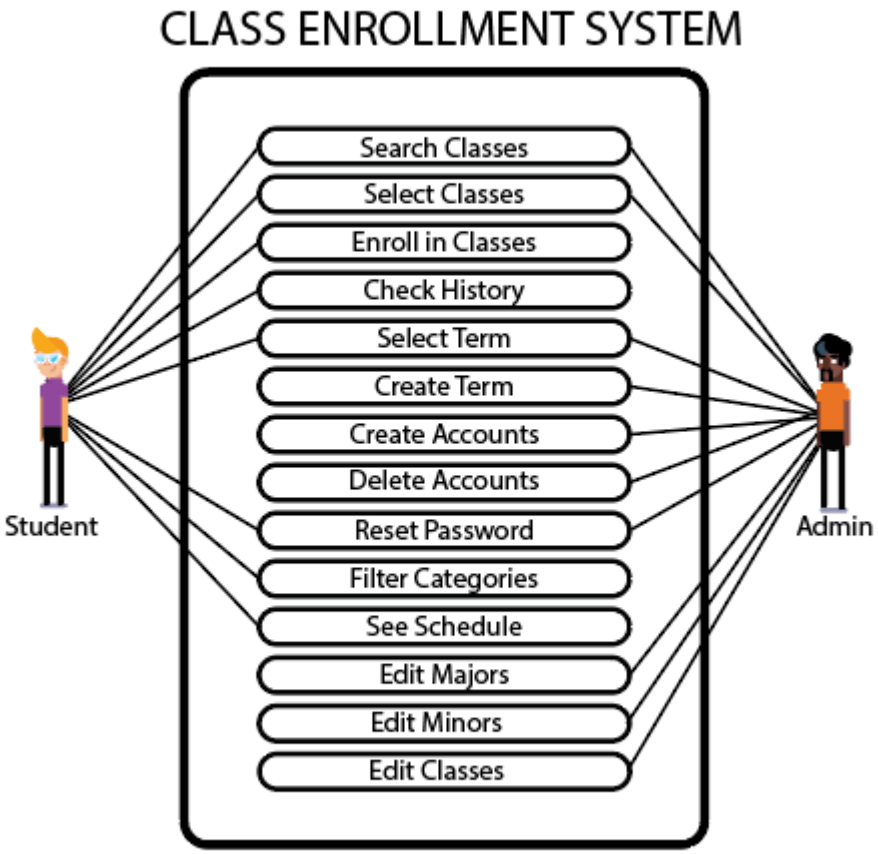
2. Overall Description

2.1 Product Perspective

This product is a replacement for MyRed's enrollment software. The current software is inefficient and sends users to different websites to find all relevant information for enrolling in classes. Our

software will be an improvement on the current application and correct the inefficiencies mentioned above.

2.2 Product Functions



2.3 User Classes and Characteristics

Since the software will have two sides, an admin and a student side, when the University installs the software they will be provided with two manuals. The first manual will be for their admins. It will detail how to do anything that can possibly be done in our admin portal of the software. The second manual will detail how to use the student portal on MyRed/the standalone web application for the first and second increment.

Thus, this software will need no user tutorials as the University already familiarizes students with the registration process during NSE days. Also, upperclassmen have had experience with the old system, picking up this new system will be easy. Since the student portal on MyRed will be built with ease of use in mind.

Also, the admin portal for this software will need no user tutorials since the University will distribute the admin manuals to their designated admins. Students will also have the ability to download and view the student portal manual if they wish to read it. Students will have to read the manual for the first and second increments.

2.4 Operating Environment

The software must work well with the MyRed system in Increment #3. For the first and second increment, it will be a standalone web application. Which means it will have to run on Safari, chrome, Firefox and Edge. Also, users will need a strong active internet connection for the software to function correctly.

2.5 Design and Implementation Constraints

The software will have to be designed and implemented to work well with the MyRed system in increment #3. This will bring up issues of designing the database to the University's standards and making sure to meet security requirements of the whole software as well.

For all increments, the software will need to be secure in the matter that students can only perform changes to the database through the student portal.

2.6 User Documentation

The two manuals, as discussed in 2.3, that come with the product will be provided to the University to disburse.

2.7 Assumptions and Dependencies

This software will be dependent on the MyRed system. This is because it will be housed within the MyRed system for the third increment. We also will assume that if the web application works on the web clients we listed in 2.4 that it will work on any computer that those web clients can function on.

3. External Interface Requirements

3.1 User Interfaces

3.1.1 Student Portal

3.1.1.1 Home Screen: There will be two displays for classes enrolled during the Selected term on this screen. The first display will be a list view while the other is a calendar view. There will be buttons to toggle academic terms, go to the browse classes screen, and logout.

3.1.1.2 Browse Classes Screen: When this screen is first visited, it will have a drop down menu that allows students to choose what category of classes they want to choose. After they select an option from the drop

down menu their screen will be populated with the relevant classes.

Classes will appear as pill-shaped elements and will immediately start populating the search display when the user selects their category. There will be buttons to search keywords, choose a new category, and logout. By clicking on a class, the pill element will expand revealing more class information as well as the option to preference time or professor when selecting a section to enroll in. The interface will cycle through time select, professor select, and section selects screens where the user chooses from a list of times, professors, and section based off their preferences and actions on prior screens. The user will then proceed to the enrollment screen.

3.1.1.3 Enrollment Screen: There will be a button for cancelling and confirming the enrollment in a course. This screen will display all the information the user has chosen so they can confirm their selected courses and enroll in them.

3.1.2 Admin Portal

3.1.2.1 Home Screen: There will be buttons for editing classes, editing majors, editing minors, and logging out. The first three buttons will take the user to their respective screens.

3.1.2.2 Edit Students Screen: The edit student screen will have a display for students and a sidebar for filtering admin actions. Admins can select to either add, edit, or remove a student. Adding a student will require them to complete all input fields, editing a student will require them to edit existing information in the input fields after they select a student, and removing a student will prompt them with a confirmation dialog box. The adding a student and editing a student screens will have a save/create, in addition, the editing a student screen will have a cancel button. All three screens will have a logout button.

3.1.2.3 Edit Classes Screen: The edit classes screen will have a display for classes and a sidebar for filtering admin actions. Admins can select to either add, edit, or remove a class. Adding a class will require them to complete all input fields, editing a class will require them to edit existing information in the input fields after they select a class, and removing a class will prompt them with a confirmation dialog box. The adding a class and editing a class screens will have a save/create, in addition, the editing a class screen will have a cancel button. All three screens will have a logout button.

3.1.2.4 Edit Majors Screen: The edit majors screen will have a display for Majors and a sidebar for filtering admin actions. Admins can select to either add, edit, or remove a major. Adding a major will require them to complete all input fields, editing a major will require them to edit existing information in the input fields after they select a major, and removing a major will prompt them with a confirmation dialog box. The adding a major and editing a major screens will have a save/create, in addition, the editing a major screen will have a cancel button. cancel button. All three screens will have a logout button.

3.1.2.5 Edit Minors Screen: The edit minors screen will have a display for minors and a sidebar for filtering admin actions. Admins can select to either add, edit, or remove a minor. Adding a minor will require them to complete all input fields, editing a minor will require them to edit existing information in the input fields after they select a minor, and removing a minor will prompt them with a confirmation dialog box. The adding a minor and editing a minor screens will

have a save/create, in addition, the editing a minor screen will have a cancel button. All three screens will have a logout button.

3.2 Hardware Interfaces

This web application should be supported on any internet capable mobile device and desktop computer. It will be optimized for desktop and mobile use. There are no special interactions between this application, the different hardware components, and the user.

3.3 Software Interfaces

3.3.1 External Software Connections: The inspiration for this software comes from the current version of the University of Nebraska's MyRed enrollment web application. This application will be optimized for desktop internet browsers including, but not limited to, Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari. Additionally, the application should work on most operating systems that have current versions of the above listed internet browsers. The data will be stored in a secured MySQL database where it will update, upload, and retrieve information based on the user's account and inputs.

3.3.2 Student Portal User Inputs

3.3.2.1 Account Information: The user will provide a username and password via text fields to access their account including all of their stored data so they know which classes they have taken and which classes they need to take.

3.3.2.2 Enrollment Selections: The user will input the classes they select via buttons for enrollment and will be enrolled in those classes in their selected term if they meet the requirements.

3.3.2.3 Class Search

3.3.2.3.1 Specific Search: The user will input a class code to find a specific class via a text field.

3.3.2.3.2 Keyword Search: The user will input a keyword into a text field to find classes associated with the provided word(s).

3.3.2.3.3 Group Search: The user will select filters or groups to find classes organized by a given characteristic.

3.3.3 Student Portal Software Outputs

3.3.3.1 Enrollment: The software will display the enrollment of the selected term in schedule form as well as list form.

3.3.3.2 Class Search: The software will display classes based on the user's search input in list form.

3.3.3.3 Graduation Requirements: The software will provide a degree like audit which lets the user know which classes they have completed and which classes they have left to take. By suggesting which college requirements they need to fulfill still based on what they have taken and need to take for their degree. This will be shown in the browse screen's drop-down menu.

3.3.3.4 Database Communication: The software will write to the database the classes the user enrolls in and update the related fields appropriately.

3.3.4 Admin Portal User Inputs

3.3.4.1 Account Information: The admin will provide their username and password via text fields to gain access to view and modify database records.

3.3.4.2 Class Information: The admin will specify pertinent information about a class in order to add it to the database. Possible pertinent Information is broken down in 3.3.4.2.1.

3.3.4.2.1 Active Term: The term in which this class is available.
3.3.4.2.2 Class Lecture Time: The time in which this class is to be held.
3.3.4.2.3 Class Lecture Location: The location in which this class's lecture will be held.
3.3.4.2.4 Class Lecture Instructor: The person's name who is lecturing.
3.3.4.2.2 Class Recitation Time: The time in which this class's recitation will be held. Class recitations are treated as a normal class, but denoted with an 'R' in their class code when created.
3.3.4.2.3 Class Recitation Location: The location in which this class's recitation will be held. Class recitations are treated as a normal class, but denoted with an 'R' in their class code when created.
3.3.4.2.4 Class Recitation Instructor: The person's name who is instructing the recitation. Class recitations are treated as a normal class, but denoted with an 'R' in their class code when created.
3.3.4.2.2 Class Lab Time: The time in which this class's lab will be Held. Class labs are treated as a normal class, but denoted with an 'L' in their class code when created.
3.3.4.2.3 Class Lab Location: The location in which this class's lab will be held. Class labs are treated as a normal class, but denoted with an 'L' in their class code when created.
3.3.4.2.4 Class Lab Instructor: The person's name who is instructing the lab. Class labs are treated as a normal class, but denoted with an 'L' in their class code when created.
3.3.4.2.5 Class Code: A unique identifier for this class. Ex. CS101
3.3.4.2.6 Class Capacity: The maximum number of students that can take the class.
3.3.4.2.2 Class Prerequisites: The classes that a student must complete so they are prepared for the course. The admin will input a comma separated list of class codes for this section.
3.3.4.2.2 Class Description: A brief description about the class and what topics may be covered.

3.3.4.3 Major Information: The admin can specify what classes are needed to satisfy a major. Some major information that is relevant is: major name, major specific classes, college class requirements.
3.3.4.4 Minor Information: The admin can specify what classes are needed to grant a minor. Some minor information that is relevant is: minor name, minor specific classes needed.

3.3.5 Admin Portal Software Outputs

3.3.5.1 Database Communication: The software will write to the database the information the admin had input from the class, major or minor information Sections. The software will then update the related fields appropriately.
3.3.5.2 Current Information: The software will pull from the database information about any specific class/major/minor so that the admin can either edit or read it.

3.4 Communications Interfaces

The web application requires an internet connection and will communicate via HTTP on various web browsers. It also requires communications to a MySQL database where the data will be secured as mentioned previously.

4. System Features

4.1 Term History

4.1.1 Description and Priority

Part of Increment #1. Medium priority: 5 out of 9. This feature will allow users to be able to see their courses for every term via a dropdown menu on the main user screen. Courses will be complete with descriptions and what grade they received in the class.

4.1.2 Stimulus/Response Sequences

On the main user screen, the user can select the 'term' dropdown menu. The user will then be shown all the available terms of their academic career (ranging from their first term at the institution to the term after the current term). When the user selects a term, their enrolled courses will be displayed. The information displayed will include class time, professor, class abbreviation, class information, and final grade received in the class; IP will be displayed if class has not started yet or is in progress.

4.1.3 Functional Requirements

REQ-1: Software must be able to pull class details stored in the database.

REQ-2: Software must be able to recognize what terms the student has been with the institution.

REQ-3: Software must be able to display all pulled database information.

4.2 Personal Advisor Checkout System

4.2.1 Description and Priority

Part of Increment #1. Maximum priority: 9 out of 9. This feature will allow users to receive all the benefits they receive from an advising appointment in a more efficient and convenient way. Users will know what credits they need for graduation and have classes recommended to them based on availability, major(s), minor(s), and their four-year plan created by the university.

4.2.2 Stimulus/Response Sequences

On the browse screen, the user will be prompted with a drop down with several various categories displayed on it. For example, ACE 1, ACE 3, Major Core, Minor Core, and CDR. The user will then click on one of these categories revealing several classes picked by our software. The user will then choose the class they prefer from the list of recommended classes. The user will then be prompted with a popover that will display more information about the class. The user can then choose whether they want to check class availability based on professor or time. If they choose professor, the screen will display all the sections of that course taught by that professor. The user will be able to add that course to their cart. If they choose time, the screen will display all class sections sorted by time. The user can then add that course to their cart. After the student has their classes in their cart, they can 'checkout' and enroll in their classes by clicking the 'enroll' button at the bottom of the cart.

4.2.3 Functional Requirements

REQ-1: Software must be able to communicate with class details stored in the database.
REQ-2: Software must be able to know what classes the student has taken and know what classes to recommend with maximum accuracy.
REQ-3: Majors and minors will need to be stored in the database to find relevant classes.
REQ-4: Users must be able to add courses to their shopping cart, but not able to enroll in more than the university's limit of credit hours.
REQ-5: Enrolled classes should be added to the student information in the database.
REQ-6: Only unfulfilled categories should be displayed in the drop-down menu with an elective category that will never go away.
REQ-7: All class information should be accurate and include class abbreviation, professor, place of instruction, time, and description.
REQ-8: Users must be able to search for a specific class after being provided with recommended classes.

4.3 Admin Dashboard

4.3.1 Description and Priority
Part of Increment #2. Very high priority: 8.5 out of 9. This feature will allow admins to edit, add, and remove students/classes/majors/minors from the database.
4.3.2 Stimulus/Response Sequences
The first screen of the dashboard will have options for editing students, classes, majors, and minors. If the admin clicks the edit classes button, they can choose to create, edit, or delete a class; creating a class requires the admin to input all relevant information, editing a class requires just to change the information already entered, and deleting a class removes the class from the system permanently. A very similar stimulus and response sequence is present for the students, majors, and minors screens.
4.3.3 Functional Requirements
REQ-1: Software must be able to communicate with class details stored in the database.
REQ-2: Software must be able to alter information already stored in the database.
REQ-3: Software must be able to assign courses as requirements for major(s) or minor(s).

4.4 MyRed Integration

4.4.1 Description and Priority
Part of Increment #3. Very high priority: 8 out of 9. This feature will allow students to access our software's enrollment software through MyRed instead of visiting our standalone web application.
4.4.2 Stimulus/Response Sequences
The student visits MyRed, then logs in. The student then clicks on the enrollment tab. The student then sees the new enrollment system and can login to their accounts. The interaction will remain the same as it was with increments #1 and #2.
4.4.3 Functional Requirements
REQ-1: Software must be able to work correctly within MyRed.

4.5 Admin Dashboard

4.3.1 Description and Priority
Part of Increment #2. Very high priority: 8.5 out of 9. This feature will allow admins to edit, add, and remove students/classes/majors/minors from the database.
4.3.2 Stimulus/Response Sequences

The first screen of the dashboard will have options for editing students, classes, majors, and minors. If the admin clicks the edit classes button, they can choose to create, edit, or delete a class; creating a class requires the admin to input all relevant information, editing a class requires just to change the information already entered, and deleting a class removes the class from the system permanently. A very similar stimulus and response sequence is present for the students, majors, and minors screens.

4.3.3 Functional Requirements

REQ-1: Software must be able to communicate with class details stored in the database.
REQ-2: Software must be able to alter information already stored in the database.
REQ-3: Software must be able to assign courses as requirements for major(s) or minor(s).

4.6 Assumptions for Increment #1

For the first increment, the database will be populated with placeholder values that admins would need to enter to demo functionality.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The software must add classes into a shopping cart or enroll the student in less than five seconds after the student has hit add to shopping cart or enroll. If more than one student attempts to enroll in a class that has one remaining seat open at the exact same time, the student with the most credits gets favored.

The software must also update the database with new information inputted by the admins in less than five seconds after submission.

5.2 Safety Requirements

The only safety requirement is that we recommend administrators, both on the development and client sides, to save backups of their database information in the very rare case of a system failure. The only thing that may be harmed is the academic records of students; the protection of this information is our highest priority.

5.3 Security Requirements

Database connections must be secure to prevent compromising sensitive information such as student academic records. Students should not be able to change their grades, see other students' grades, or alter classes, majors, or minors in any way, shape, or form. Only admins have that privilege.

Admin or student logins must be encrypted as well so that credentials are not sent as plain text over the connection.

5.4 Software Quality Attributes

The student portal must be very easy to use, available year-round 24/7 with minimum downtime for maintenance at low usage points, and must make correct choices when recommending classes. It must be robust enough to handle malformed input for searches, but robustness is not necessary for the enrolling aspect since it will be choice led and require no unexpected inputs.

The admin portal must be easy to use but also a little bit more complicated because of the nature of the information being inputted. It too must be available year-round 24/7 with minimum downtime for maintenance at low usage points, must make sure the information provided by the admin is in acceptable format, and must be robust when encountering malformed input formats.
The student and admin portals must work on current versions of internet web browsers that we listed earlier.

5.5 Business Rules

The only two roles in this system is student and admin. There will be no further hierarchy within these two roles. Every admin will have the same privileges as the other admins and the students likewise. For further functions please consult section **2.2 Product Functions**.

6. Other Requirements

The full first increment is due by July 26, 2017. The full second increment is due by August 8, 2017. The full third increment is due by September 1st, 2020.

Appendix A: Glossary

MyRed -- University of Nebraska's Student Portal.
NSE -- New Student Enrollment day.
Student Portal -- Personal Advisor Checkout System. Allows students to enroll in classes and to view their progress.
Admin Portal -- Admin Dashboard which allows admins to manipulate information in the database.