SRS - Class Registration

# 1 Introduction

## 1.1 Purpose

The purpose of this document is to define the requirements for the system that manages classes at Brigham Young University - Idaho. The system will be used by students, technical administrators, and faculty members.

## 1.2 Scope

This document contains a complete description of the functions to be used for the registration system at Brigham Young University - Idaho. This document also includes functions necessary to work with the Brigham Young University - Idaho *Gradplanner* tool.

## 1.3 Definitions, Acronyms, and Abbreviation

### 1.3.1 Administrator

A user that has administrative control over the entire system, not just specific classes or courses. Administrators can create and modify courses and class offerings, whereas professors may only be able to modify their own class offerings

### 1.3.2 Class

A specific offering of a course, tied to a professor, weekday(s) and time(s), location, semester, number of seats, etc. An example of a class is CS364-01 Winter Semester 2015. Classes can be online.

### 1.3.3 Course

A curriculum for a course, with a course ID, title, description, etc. Not to be confused with a “class” which is a specific offering of a course. An example of a course is CS364 Software Engineering I.

### 1.3.4 Professor

A professor may have some degree of control of class offerings they teach, but probably do not have full administrative access in most cases.

### 1.3.5 Student

### 1.3.6 Gradplan

A gradplan is a sequence of courses defined by a student. A gradplan associates courses with specified semesters to act as a guide in the student’s academic career towards graduation. Furthermore, a gradplan is used by the school in conjunction with a student’s transcript to determine how well the student is meeting academic requirements.

### 1.3.7 Outstanding Charges

Any financial credits a student has been charged to their school account that have not been resolved or paid for.

### 1.3.8 Prerequisite

A course, series of courses, declared major, or declared minor required to be completed by prior to registering for a desired course.

### 1.3.9 Pre-Registration

A student’s status prior to registration opening. The student is not allowed to register for classes under this status.

### 1.3.10 Tentative Plan (TP)

A student’s proposed schedule for classes. The student’s TP includes: course and class.

### 1.3.11 Class Information

The meeting schedule, class length, meeting location, instructor, credit count, and text description of a given class.

### 1.3.12 Academic Deadline

Two weeks after the first day of a semester. Used as a cutoff for registration.

## 1.4 References

## 1.5 Overview

# 2 Overall description

## 2.1 Product Perspective

## 2.2 Product Functions

## 2.3 User Characteristics

## 2.4 Constraints

## 2.5 Assumptions and Dependencies

# 3 Specific Requirements

## 3.1 External interface requirements

### 3.1.1 User interfaces

### 3.1.2 Hardware interfaces

### 3.1.3 Software interfaces

### 3.1.4 Communications interfaces

## 3.2 Functional Requirements

### 3.2.1 Student

#### 3.2.1.1 Add

##### 3.2.1.1.1 The student shall be notified when they are able to register.

This requirement exists because classes fill up quickly and students need to be notified when it is their time to register.

##### 3.2.1.1.2 The student shall be able to register for a class.

This requirement exists because this is the purpose of the system.

##### 3.2.1.1.3 The student shall be allowed to register for 21 credits at most.

This requirement exists because university policy has set credit limit for how many credits a student can register for in one semester.

##### 3.2.1.1.4 The student shall be allowed to register for one class per course.

This requirement exists because students are not allowed to be in two or more classes of the same course simultaneously.

3.2.1.1.5 The system shall automatically register a student for classes using data from the pre-registration system.

This requirement exists because

##### 3.2.1.1.7 The student shall be required to meet all course prerequisites before registering for a class.

This requirement exists because students cannot take classes they are unqualified for.

##### 3.2.1.1.8 The student shall be notified if they do not meet course prerequisites.

This requirement exists because students need to know if they do not qualify for a class.

##### 3.2.1.1.9 The student shall register for classes they have not taken previously.

This requirement exists because students need to register for classes

##### 3.2.1.1.10 The student shall have two weeks to add classes after the semester begins.

This requirement exists because of the add deadline given by the university.

##### 3.2.1.1.11 The system shall display classes a student is already registered for.

This requirement exists because students need to see what they have already taken.

##### 3.2.1.1.12 The student shall choose a search criteria to display classes to register for.

This requirement exists because students do not want to see all available class all the time.

##### 3.2.1.1.13 The student shall be able to add a classes during the first week of the term.

This requirement exists because of the university policies.

##### 3.2.1.1.14 The student shall be able to add second block class up to 10 days after the class begins.

This requirement exists because of university policies.

#### 3.2.1.2 Drop

##### 3.2.1.2.1 The student shall be able to withdraw from a class of their choosing.

This requirement exists because students need be able to withdraw themselves from classes.

##### 3.2.1.2.2 The system shall warn a student that is dropping a class after the academic deadline.

This requirement exists because students receive specific letter grades depending specific academic deadlines.

##### 3.2.1.2.3 The student shall receive a ‘W’ grade upon dropping a class 22 days after the beginning of the semester.

This requirement exists because of university policies.

##### 3.2.1.2.4 The system shall remove a class from the students registered class list.

This requirement exists because students should not see classes they have dropped.

##### 3.2.1.2.5 The system shall require the student to give a reason why all classes are being dropped.

This requirement exists because of university policy.

#### 3.2.1.3 Search

##### 3.2.1.3.1 The student shall be able to search by term.

This requirement exists because students do not want to see all available class all the time.

##### 3.2.1.3.2 The student shall be able to search by discipline.

This requirement exists because students do not want to see all available class all the time.

##### 3.2.1.3.3 The student shall be able to search by department.

This requirement exists because students do not want to see all available class all the time.

##### 3.2.1.3.4 The student shall be able to search by sub program.

This requirement exists because students do not want to see all available class all the time.

##### 3.2.1.3.5 The student shall be able to search by course code.

This requirement exists because students do not want to see all available class all the time.

##### 3.2.1.3.6 The student shall be able to search by instructor.

This requirement exists because students do not want to see all available class all the time.

##### 3.2.1.3.7 The student shall be able to search by course title.

This requirement exists because students do not want to see all available class all the time.

##### 3.2.1.3.8 The student shall be able to search based on the day the class is offered.

This requirement exists because students do not want to see all available class all the time.

##### 3.2.1.3.9 The student shall be able to search based on the time of day the class is offered.

This requirement exists because students do not want to see all available class all the time.

##### 3.2.1.3.10 The student shall be able to search based on class availability.

This requirement exists because students need to see if a class is filled or not.

##### 3.2.1.3.11 The system shall display each course class based on search criteria.

This requirement exists because students need to see what classes they are registering for.

##### 3.2.1.3.12 The system shall display each course title of a class based on search criteria.

This requirement exists because students need to see what classes they are registering for.

##### 3.2.1.3.13 The system shall display each course credit value of a class based on search criteria.

This requirement exists because students need to see how many credits each class is.

##### 3.2.1.3.14 The system shall display each instructor of a class based on search criteria.

This requirement exists because students register for specific classes depending on the instructor.

##### 3.2.1.3.15 The system shall display the number of seats available in a class based on search criteria.

This requirement exists because students need to see how full a class is.

##### 3.2.1.3.16 The system shall display the status of a class based on search criteria.

This requirement exists because students need to see whether a course class opens.

##### 3.2.1.3.17 The system shall display the schedule of a class based on search criteria.

This requirement exists because students need to see class schedules.

##### 3.2.1.3.18 The system shall display the subprogram of a class based on a search criteria.

This requirement exists because students often search for classes through subprograms.

##### 3.2.1.3.19 The system shall display the method of a class based on search criteria.

This requirement exists because students do not want to see all available class all the time.

##### 3.2.1.3.20 The student shall choose which semester to display classes for.

This requirement exists because students do not want to see all available class all the time.

#### 3.2.1.4 Waitlist

##### 3.2.1.4.1 If a student adds a class that is full the student shall be asked by the system to be added to the class waitlist.

#### 3.2.1.5 Pre-registration

##### 3.2.1.5.1 The pre-registration system shall use the registration search features to retrieve the classes to show to the student.

##### 3.2.1.5.2 Look Up

##### 3.2.1.5.3

##### 3.2.1.5.4 Find

##### 3.2.1.5.5 A student shall be able to access the pre-registration system.

##### 3.2.1.5.6 A student will be able to view classes for the next upcoming semester at the start of the current semester.

Students will need adequate time to plan their schedules prior to class registration opening for their credit bracket.

##### 3.2.1.5.7 A student’s Tentative Plan shall be stored

##### 3.2.1.5.8 The student shall be able to pre-register.

This requirement exists because it will facilitate the registration process

##### 3.2.1.5.9 The student shall be able to add tentative classes to a pre-registration plan.

##### 3.2.1.5.10 The system shall store all changes made to a pre-registration plan.

##### 3.2.1.5.11 The student shall be able to retrieve their tentative schedule from a stored pre-registration plan.

##### 3.2.1.5.12 The student shall be allowed to register for all tentative classes from the pre-registration plan when allowed to register.

##### 3.2.1.5.13 The system shall allow for changes to be made to a student’s pre-registration plan.

#### 3.2.1.6 Schedule

##### 3.2.1.6.1 The system will use a student’s gradplan to recommend classes for the current semester.

##### 3.2.1.6.2 The system will use a student’s gradplan to recommend classes for the upcoming semester.

### 3.2.2 Admin

#### 3.2.2.1 API

##### 3.2.2.1.1 The system shall be able to access student information from Jenzabar.

This allow students to be added to a class

##### 3.2.2.1.2 The system shall be able to access hold information from Jenzabar.

If the student has a hold, registration needs to be blocked

##### 3.2.2.1.3 The system shall be able to access students’ gradplan courses from Jenzabar.

Needed for use in pre-registration

#### 3.2.2.2 Waitlist

##### 3.2.2.2.1 The system shall automatically add students to a waitlist if class is full.

When the system doesn’t automatically add students it creates unnecessary work for students The students need to be added to a waitlist in case of an opening due to another student dropping the class.

##### 3.2.2.2.2 An administrator shall be able to drop students from the waitlist.

Useful in case of errors

##### 3.2.2.2.3 The system will be able to automatically add students to the class they are waitlisted for as seats become available.

The waitlist process will be faster and more efficient.

##### 3.2.2.2.4 The system will be able to automatically send text notifications to students on a waitlist when a spot becomes available.

Student needs to be aware that he/she has been added to the class

#### 3.2.2.3 Database

##### 3.2.2.3.1 An administrator shall be able to add classes to the database

New classes need to be added each semester to meet the needs of the students.

##### 3.2.2.3.2 An administrator shall be able to remove classes from the database.

Some classes are no longer needed and must be removed.

##### 3.2.2.3.3 An administrator shall be able to edit classes in the database.

There may be a class that only needs some minor adjustments rather than being dropped completely

#### 3.2.2.4 Holds

##### 3.2.2.4.1 The system shall place holds on students automatically

The administrator should not have to place holds on each student manually.

##### 3.2.2.4.2 The system shall not allow students with holds to register for classes

Holds need to be resolved before allowing registration.

##### 3.2.2.4.3 An administrator shall be able to manually resolve registration holds.

This would be for exceptions that cannot be resolved automatically.

#### 3.2.2.5 Permissions

##### 3.2.2.5.1 An administrator shall be able to grant permission for a student to register for a class that has restrictions.

This would be for students who take a class and its prerequisite at the same time. This would also be for students who get teacher permission to bypass the registration system for a specific class.

#### 3.2.2.6 Priorities

##### 3.2.2.6.1 The system shall be able to track the number of student credits.

This allows for priority registration, which separates the students into registration groups with staggered starting times. This prevents all students from registering at once and bogging down the system.

### 3.2.3 Teacher

#### 3.2.3.1 Add

##### 3.2.3.1.1 Teachers shall be able to add students to classes.

Teachers need to be able to add students to classes without regard to the waitlist or to class pre-requisites. This requirement simplifies the current system’s convoluted associated processes.

#### 3.2.3.2 Student Information

##### 3.2.3.2.1 Teachers shall be able to view student information for students registered in their classes.

The current system doesn’t provide teachers with enough student information about students who are registered in their classes. Teachers want this information.

##### 3.2.3.2.3 Teachers shall be notified of all students who drop the class after the semester starts.

The current system doesn’t provide this information to teachers, and teachers want to know it.

## 3.3 Performance requirements

## 3.4 Design constraints

## 3.5 Software system attributes

## 3.6 Other requirements

# 4 Appendixes

## 4.1 Elicitation Documentation

### 4.1.1 User Stories

4.1.1.1 Emily is a student in her junior year. Emily’s *gradplan* has completed her *gradplan* for the upcoming semester. She has recently been notified by email that registration is now open for her credit bracket, and that she can now register for the upcoming semester. She signs into her student account and has navigated to the place to search for classes.

Emily is prompted whether she wants to find classes on her own or use her *gradplan* to find classes. Emily indicates that she wants to use her *gradplan*. Emily chooses a 300 level course from her *gradplan*. The system shows her the available classes with their associated days, times, and instructors. She clicks one among the few that are listed; as Emily clicks on the class, she is notified that the class has been saved but not registered yet. The system now shows a schedule (in addition to the course and class information) which indicates the days and times of the class Emily has selected. Emily notices a different class option that she thinks may be better suited for her but she is uncertain, so she indicates the different class option as well. The system notifies her that the selected class has been saved and that two of [course name] are now saved.

Emily clicks on another course that is listed from her *gradplan*; the system hides the classes from the previously selected course and shows the current class options for the new course Emily has indicated. She clicks on a class that is listed. The system notifies Emily that the class has been saved but not added, but that there is a conflict of schedule. Emily notices that the schedule view clearly indicates where the conflicts are at; she clicks on another class and clears the problem. Emily continues this approach for the remainder of her classes.

When Emily has found and saved all of the classes for the upcoming semester and there are no conflicts, Emily clicks on the “Add Classes” button. She is prompted to indicate which classes (among the saved classes) she would like to register for. Emily clicks “Select All” and then clicks “Register”.

The system notifies Emily that each class has been successfully added. A link is provided to let her see and download her schedule for the upcoming semester. She is then assured that the classes have been successfully registered.

4.1.1.2 Troy is a student in his freshman year at Brigham Young University - Idaho. Troy knows that registration would be taking place soon, so he decided to visit the Academic Discovery Center (ADC) to declare his major and minor. While at the ADC, he made a *gradplan* using his student account.

Troy logs into his my.byui.edu account and begins his search for classes. As he searches for classes using the list from his *gradplan*, Troy indicates a class that he believes will work with his schedule. The system then notifies Troy that the class has been saved but not registered, and that he can begin registering on [Freshman Date] because he is in the freshman credit bracket.

Troy continues to add classes that will fit with his *gradplan* and ideal schedule.

The day prior to [Freshman Date], Troy receives an email and text notification from the school indicating that he can register for classes the following day. Troy clicks on the link provided in the email, taking him to where he can register for the classes. He is shown the list of saved courses. Troy indicates which courses (assuming the classes are tied to them as well) he would like to register for; when Troy is done deciding, he clicks “Register”. Troy is taken to a confirmation page.