# Courses and Classes Records System Requirements Definitions

**Nonfunctional Requirements**

* The system should be accessible over the internet sentence.
* Only staff members are the users of the system.
* The system should ensure that only allowed users are able to use the system.

**Functional Requirements**

* Maintain Course Records
* The system should maintain CRUD operations on (i.e., provide CREATE, READ, UPDATE, and DELETE operations on) course records with the following attributes:

1. Course identification (e.g., IT 315)
2. Course name (e.g., Object-Oriented Analysis and Design)
3. Credit hours
4. Description

* Maintain Class Records
* Of the above courses, classes are offered each semester. There are two kinds of classes. Online classes are offered over the internet and accessible via a web browser; face-to-face classes that are offered on campus at the college. A class cannot exist without an associated course in the system
* The system should maintain CRUD operations on (i.e., provide CREATE, READ, UPDATE, and DELETE operations on) class records with the following attributes:

1. Course identification
2. Class begin date
3. Class end date

* In addition, online classes maintain the following additional attributes:

1. Class URL
2. Class browser

* Similarly, face-to-face classes maintain the following additional attributes:

1. Class building
2. Class room

# Student Information System Requirements Definitions

**Nonfunctional Requirements:**

* The student information system (SIS) should be accessible as both a website and a mobile app.
* The SIS users are students and enrollment staff.
* The SIS should enforce appropriate security privileges to its uses.

**Functional Requirements:**

* Maintain Student Records
* The SIS should maintain records of basic information about students such as:

1. First name
2. Middle initials
3. Last name
4. Date of birth
5. Student ID
6. Department (e.g., psychology)
7. Cumulative GPA (on a scale of 1.0 to 4.0)

* These records are updated by the enrollment staff who has the permissions to add, modify, and/or delete any student record.
* Maintain Course Records
* The SIS should maintain records of all the courses offered by the college.
* The system should maintain the following fields for each course:

1. Course ID
2. Course name
3. Credit hours
4. Description
5. Prerequisite course

* These records are updated by the enrollment staff who has the permissions to add, modify, and/or delete any course record.
* Maintain Class Records
* Classes are offered each semester of the above courses.
* There are two kinds of classes: Online classes offered over the internet are accessible via a web browser and face-to-face classes are offered on campus at the college.
* A class cannot exist without an associated course in the system.
* The basic information to maintain for all kinds of classes include the following:

1. Course ID
2. Class begin date
3. Class end date

* In addition, online classes also maintain the following information:

1. Class URL
2. Class browser

* Similarly, face-to-face classes should maintain this additional information about them:

1. Class building
2. Class room

* These records are updated by the enrollment staff who has the permissions to add, modify, and/or delete any class record (either online or face-to-face).
* Register a Student for Classes
* Both students and enrollment staff should be able to register a student (either online or face-to-face) for a class offering for a course. The registration process should proceed as follows:

1. The student information is entered to identify the registering student.
2. The course information for which to register is entered to identify the course.
3. All future classes with start dates after the current date for this course are displayed.
4. The enrollment staff enter future classes for only one semester out.
5. A selection of one class for which to register is then made.
6. The selected class registration is then validated against the following registration rules:
7. There is no duplicate registration for the same class in one semester.
8. Online class registration requires students to acknowledge that they have the required hardware and software to access online classes.
9. No more than three class registrations are allowed for one student in one semester unless the student’s GPA is greater than 3.5. (In this case, there is no limit on number of class registrations.)

* When the class registration is validated against the above rules, a confirmation is then displayed to confirm registration. Otherwise, a message explaining the violation of the associated rule is displayed.

The process continues, starting from item b), until the use

1. Generate your student information system (SIS) sequence diagram for the Register a Student for Classes use case.

A close up of a map

Description automatically generated

Generate your SIS communication diagram for the Register a Student for Classes use case.

A close up of text on a white background

Description automatically generated

**SIS Method Contract 1 template** (refer to textbook pages 306–314):

|  |  |  |
| --- | --- | --- |
| Method Name:  Course Records | Class Name:  Course Records | ID:  2 |
| Clients (Consumers):  Enrollment staff | | |
| Associated Use Cases:  Class records, student records | | |
| Description of Responsibilities:  Maintain update and delete course records | | |
| Arguments Received:  none | | |
| Type of Value Returned:  confirmed | | |
| Pre-Conditions:  Maintain course records and create new courses | | |
| Post-Conditions:  none | | |

**SIS Method Contract 2 template:**

|  |  |  |
| --- | --- | --- |
| Method Name:  Enrollment | Class Name:  Student records | ID:  1 |
| Clients (Consumers):  students | | |
| Associated Use Cases:  Student records course enrollment | | |
| Description of Responsibilities:  Maintain update and change enrollment information and student records | | |
| Arguments Received:  none | | |
| Type of Value Returned:  confirmation | | |
| Pre-Conditions:  Student must already be enrolled then they ca edit their information | | |
| Post-Conditions:  none | | |

**SIS Method Specification 1 template** (refer to textbook pages 314–318):

|  |  |  |
| --- | --- | --- |
| Method Name:  Class records | Class Name:  Maintain class records | ID:  3 |
| Contract ID:  1 | Programmer:  NA | Date Due:  12/9/2019 |
| Programming Language:  Java | | |
| Triggers/Events:  Maintain create and delete class records and student’s information | | |

| **Arguments Received:**  **Data Type:** | **Notes:** |
| --- | --- |
| Class is already registered  Online requirements when f2f  Class numbers are registered | It isn’t wanting to register, or the requirements are not meant. |

| **Messages Sent & Arguments Passed:**  **ClassName.MethodName:** | **Argument Data Type:** | **Notes:** |
| --- | --- | --- |
| Maintain class records  Class records begin and end date  Maintain f2f or online info  Maintain student records | int | Find course ID |
| int | Find class begin and end date |
| int | Edit student info |

| **Argument Returned:**  **Data Type:** | **Notes:** |
| --- | --- |
| Class duplicate or registered requirements | Verify the class is not a duplicate and the requirements are meant |
| Algorithm Specification:  Call class requirements model to perform class verification against class registration | |
| Misc. Notes:  This method class uses a class requirements model to verify the class registration against registration rules | |

**SIS Method Specification 2 template:**

|  |  |  |
| --- | --- | --- |
| Method Name:  Select courses | Class Name:  courses | ID:  4 |
| Contract ID:  2 | Programmer:  NA | Date Due:  12/9/2019 |
| Programming Language:  Java | | |
| Triggers/Events:  Actor needs to input student information and course information to register | | |

| **Arguments Received:**  **Data Type:** | **Notes:** |
| --- | --- |
| Student identification  Course information | Student information is inputted into the system  Course information is inputted in the system |

| **Messages Sent & Arguments Passed:**  **ClassName.MethodName:** | **Argument Data Type:** | **Notes:** |
| --- | --- | --- |
| coursesdisplaycourses | String | Course info |
| string | Registration info |
| string | Student |

| **Argument Returned:**  **Data Type:** | **Notes:** |
| --- | --- |
| string | The course records are displayed for actor to make selection |
| Algorithm Specification: display courses to display list of the course catalog  Call | |
| Misc. Notes:  This method class uses student and course information that is displayed | |

1. Verify and validate your sequence diagram and communication diagram against your SIS functional model and structural model.

I was able to verify and validate the sequence diagram and communication diagram by structural and functional models. The student can access everything besides setting up that class information where as the enrollment staff is able to access all the course info and student records. You access the registration then the form would take you to the course catalog to retrieve the class offerings and catalog information then you choose if you want face to face or online courses, then you register for said class. When the student makes all the selections the functional model is complete.

1. Explain your approach to the problem, the decisions you made to arrive at your solution, and how you completed it.

Using the functional model in the second assignment I was able to validate the sis cases and find the best way to approach the assignment. I then gathered the information for the functional and structural models to find that the registration of the classes for the students was the best course of action for the functional model.

1. Reflect on this experience and the lessons you learned from it.

The main problem I had with this assignment and with this class is I seem to think to broadly or to much in detail. This class and using UML are about keeping it basic but with enough detail to accomplish whatever you are working towards. Luckily using extra sources like YouTube and Reddit to understand what I’m supposed to do helps so much. But this also helps to code, I am taking a game design class this semester and you have to map out things like moving doors, this class actually helped me by building the uml and functional models. This made me break down things in a way I’m not use to and it helped me to build out blueprints in the game engine.