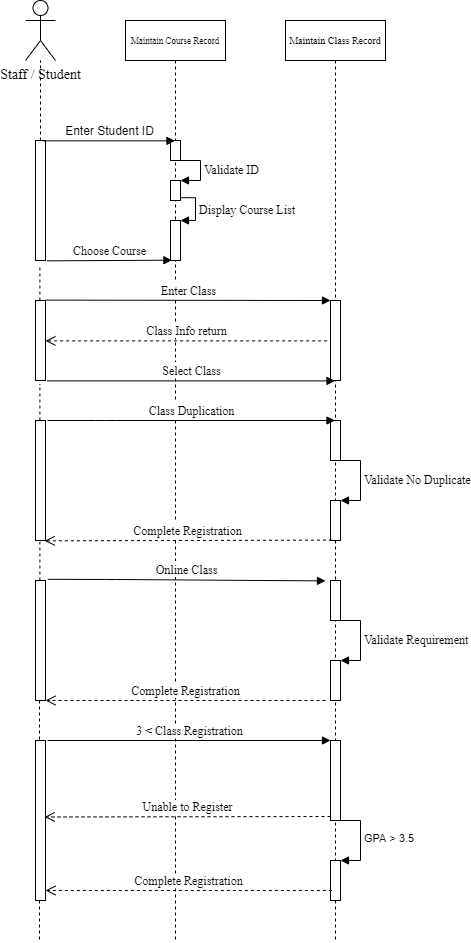
# IT 315 Final Project Part III Solution Submission Template

This template is a guide for you to organize your information. To complete it, **replace the bracketed text with the relevant information.** Some areas may be too large or too small for the information you’re inserting. Adjust the size of the areas as necessary.

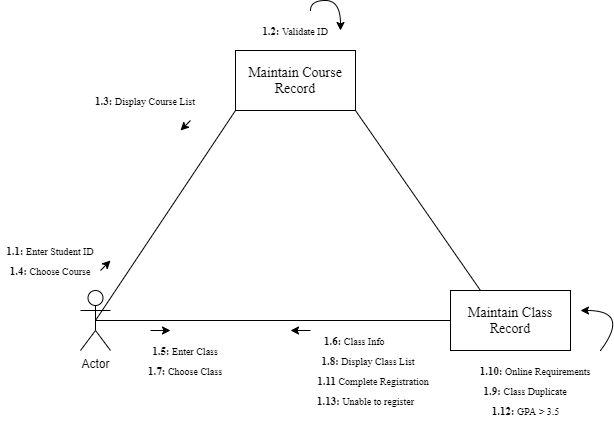
**Name:** [Joseph Silva Jr.]

**Date:** [02/28/21]

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



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

**n**

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

|  |  |  |
| --- | --- | --- |
| Method Name:  [Validate Student ID] | Class Name:  [Course Record] | ID:  [1] |
| Clients (Consumers):  [Staff & Student] | | |
| Associated Use Cases:  [Student Registration Record, Maintain Student Record & Course Record] | | |
| Description of Responsibilities:  [The Student Registration Record sends the entered Student ID by the actor to the Maintain Student Record to check if the entered Student ID is valid or invalid.] | | |
| Arguments Received:  [Student ID - Validate] | | |
| Type of Value Returned:  [void] | | |
| Pre-Conditions:  [An actor entered student ID.] | | |
| Post-Conditions:  [Student ID is validated to obtain information to deny or give access to course list.] | | |

**SIS Method Contract 2 template:**

|  |  |  |
| --- | --- | --- |
| Method Name:  [Enter Class] | Class Name:  [Class Record] | ID:  [2] |
| Clients (Consumers):  [Student & Staff] | | |
| Associated Use Cases:  [Class Record] | | |
| Description of Responsibilities:  [Actor enters class and Class Record returns class info.] | | |
| Arguments Received:  [Enter Class – Class request info returned] | | |
| Type of Value Returned:  [void] | | |
| Pre-Conditions:  [An actor puts in class name] | | |
| Post-Conditions:  [Class record sends Actor Class Info.] | | |

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

|  |  |  |
| --- | --- | --- |
| Method Name:  [Validate Student ID] | Class Name:  [Course Record] | ID:  [3] |
| Contract ID:  [100] | Programmer:  [J. Silva] | Date Due:  [02/28/2021] |
| Programming Language:  C++ | | |
| Triggers/Events:  [Actor enters Student ID and ID is verified before course list is sent back to actor.] | | |

| **Arguments Received:**  **Data Type:** | **Notes:** |
| --- | --- |
| [Validate Student ID] | [The entered Student ID validated.] |

| **Messages Sent & Arguments Passed:**  **ClassName.MethodName:** | **Argument Data Type:** | **Notes:** |
| --- | --- | --- |
| [validateStudentID.getStudentID()] | [student ID] | [] |
| [] | [] |
| [] | [] |

| **Argument Returned:**  **Data Type:** | **Notes:** |
| --- | --- |
| [void] | [] |
| Algorithm Specification:  [Sequence & Communication diagram] | |
| Misc. Notes: | |

**SIS Method Specification 2 template:**

|  |  |  |
| --- | --- | --- |
| Method Name:  [Enter Class] | Class Name:  [Class Record] | ID:  [4] |
| Contract ID:  [115] | Programmer:  [J. Silva] | Date Due:  [02/28/21] |
| Programming Language:  C++ | | |
| Triggers/Events:  [An actor enters class name into the class record to return class info] | | |

| **Arguments Received:**  **Data Type:** | **Notes:** |
| --- | --- |
| [void] | [] |

| **Messages Sent & Arguments Passed:**  **ClassName.MethodName:** | **Argument Data Type:** | **Notes:** |
| --- | --- | --- |
| [enterClass.getClass()] | [Class] | [] |
| [] | [] |
| [] | [] |

| **Argument Returned:**  **Data Type:** | **Notes:** |
| --- | --- |
| [void] | [] |
| Algorithm Specification:  [sequence & communication diagram] | |
| Misc. Notes:  [none] | |

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

I compared the sequence and communication diagrams to the SIS functional model structure that was given to us in the beginning of this course. I used the SIS guideline to make sure to stay within the rules of the system for the use case. I conducted an entire overhaul of my diagram and templates due to the feedback. As you can see in the diagrams, I made sure to show how each action had a reaction such as in the sequence diagram, I showed how when registering for a class a process in conducted and multiple validation needs to be made in order to continue certain processes. The multiple validations were shown in both diagrams and one of those examples were entering a Student ID. The actor has to enter the student ID and the system validates the ID before a course list is returned to the actor.

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

I used my previously created sequence diagram as a template to create this sequence diagram. For the communication diagram, contracts, and specifications, I followed the reading material in order to get as close as possible to the example shown in the book. The contacts and specifications were difficult for me and I am not positive I created them perfectly, but the book helped me get as close as possible.

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

The experience and lessons I obtained from this assignment were more multi-tasking then the previous milestones. This assignment showed how much work goes into software developments when we can see even just one-use case takes a lot of time to add the information for the diagrams, cards, specifications, and contracts. I also learned from each assignment that each diagram gets smaller and more detailed when it comes to showing the process from major use cases to communication diagrams.