# IT 315 Final Project Part II 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:** Walter Hernandez

**Date:** June 9th, 2023

1. Using this class responsibility collaboration (CRC) card template, document each class you identified from your student information system (SIS) functional model.

**CRC Card 1:**

| Front: Initial Login | | | |
| --- | --- | --- | --- |
| Class Name:  Student Login | ID:  1001 | | Type:  Concrete, Domain |
| Description:  Initial Login for SIS access | | | Associated Use Cases:  Student Enrollment |
| Responsibilities:  To allow student to SIS. | | Collaborators:  Enrollment Staff, Student3 | |
| Back:  Access to SIS | | | |
| Attributes:  -Username:  -Password: | | | |
| Relationships:  #LoginVerification()  Generalization (a-kind-of):  Aggregation (has-parts):  Other Associations: | | | |

**CRC Card 2:**

| Front: Main Menu Landing Page | | | |
| --- | --- | --- | --- |
| Class Name:  Main Menu | ID:  1002 | | Type:  Concrete, Domain |
| Description:  Landing Page after login | | | Associated Use Cases:  Student Login |
| Responsibilities:  To display all options after login | | Collaborators:  Student | |
| Back:  Take Student to next page after selection. | | | |
| Attributes:  -Register:  -StudentInformation: | | | |
| Relationships:  Generalization (a-kind-of):  Aggregation (has-parts):  -StudentLogin()  Other Associations: | | | |

**CRC Card 3:**

| Front: Course Search Option | | | |
| --- | --- | --- | --- |
| Class Name:  Course Search | ID:  1003 | | Type:  Concrete, Domain |
| Description:  Give student access to the course search | | | Associated Use Cases:  Main Menu |
| Responsibilities:  Display options to search for courses | | Collaborators:  Enrollment Staff, Student | |
| Back:  Display available courses | | | |
| Attributes:  -BySubject:  -ByCourseNumber: | | | |
| Relationships:  Generalization (a-kind-of):  Aggregation (has-parts):  -MainMenu()  Other Associations: | | | |

**CRC Card 4:**

| Front: Display Courses | | | |
| --- | --- | --- | --- |
| Class Name:  Course Courses | ID:  1004 | | Type:  Concrete, Domain |
| Description:  Show student available courses | | | Associated Use Cases:  Student Enrollment |
| Responsibilities:  To allow student to SIS. | | Collaborators:  Enrollment Staff, Student | |
| Back:  Give student access to classes | | | |
| Attributes:  -CourseID:  -CourseName:  -CreditHours:  -Description: | | | |
| Relationships:  Generalization (a-kind-of):  Aggregation (has-parts):  -Student()  -EnrollmentStaff()  Other Associations: | | | |

**CRC Card 5:**

| Front: Class Options | | | |
| --- | --- | --- | --- |
| Class Name:  Classes | ID:  1005 | | Type:  Concrete, Domain |
| Description:  Show student available classes | | | Associated Use Cases:  Course Display |
| Responsibilities:  Display all available classes | | Collaborators:  Enrollment Staff, Student | |
| Back:  Take user to physical or online | | | |
| Attributes:  -ClassID:  -CourseBegin:  -CourseStart:  -CourseEnd:  -Prerequisites:  -GPA: | | | |
| Relationships:  Generalization (a-kind-of):  Aggregation (has-parts):  -CourseDisplay()  Other Associations: | | | |

**CRC Card 6:**

| Front: Student elects an option | | | |
| --- | --- | --- | --- |
| Class Name:  Physical or Online | ID:  1006 | | Type:  Concrete, Domain |
| Description:  Give student option of online or physical | | | Associated Use Cases:  Class Options |
| Responsibilities:  Register student to an online or physical class | | Collaborators:  Student, Enrollment Staff | |
| Back:  Displays Class info for designated choice. The gives option of register for another course, or logout. | | | |
| Attributes:  -Class Building:  -Classroom number:  -Browser:  -Classurl: | | | |
| Relationships:  Generalization (a-kind-of):  -GPA()  -Prerequisites()  Aggregation (has-parts):  -ClassDisplay()  -RegisterToAnotherCourse()  -Logout()  Other Associations: | | | |

**CRC Card 7:**

| Front: Student gets a registration error or registration verification | | | |
| --- | --- | --- | --- |
| Class Name:  Reg Confirmation | ID:  1007 | | Type:  Concrete, Domain |
| Description:  To alert student of registration confirmation or error | | | Associated Use Cases:  Physical or Online |
| Responsibilities:  To alert student of error or registration success. | | Collaborators:  Student, Enrollment Staff | |
| Back:  Student then gets taken back to Class Options or gets option to register for another course. | | | |
| Attributes:  -GPA:  -Prerequisites:  -RuntimeError() | | | |
| Relationships:  Generalization (a-kind-of):  -  Aggregation (has-parts):  -Logout()  -RegisterForAnotherCourse()  Other Associations: | | | |

**CRC Card 8:**

| Front: Student click on Register for another course | | | |
| --- | --- | --- | --- |
| Class Name:  Registration\_Loopback | ID:  1008 | | Type:  Concrete, Domain |
| Description:  Takes student back to course search. | | | Associated Use Cases:  Reg Confirmation |
| Responsibilities:  Take student back to registration page | | Collaborators:  Student, SIS | |
| Back:  Course Search is displayed. | | | |
| Attributes:  -BySubject:  -ByCourseNumber: | | | |
| Relationships:  Generalization (a-kind-of):  Aggregation (has-parts):  -MainMenu()  Other Associations: | | | |

Generate your SIS class diagram:



1. Verify and validate your CRC cards and class diagram against your SIS functional model:

The Class Diagram works very well with the SIS.

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

The approach to the problem was to try to be linear as possible. I also researched a lot of the reading and tutorials to understand what is going on. I am still having trouble; however, I feel that I get the idea of it now. The approach was to understand what the SIS was doing. Then I looked at other videos to see how it correlates.

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

The class diagram visually helps the programmers or designer see what is going on with the software. The viewer can see what is supposed to happen, and with higher details see the operations that should align with the class. It also show where the next step is after each class, which gives the designer an idea of where to go next.