# IT 315 Final Project Part II

**Name:** Carlos Ramirez

**Date:** 4/11/2021

**CRC Card 1:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Front:** | | | |
| **Class Name:**  Course Records | **ID:**  001 | | **Type:**  Concrete Domain |
| **Description:**  Will allow the enrollment staff to add, modify, and/or delete any courses offered by the college. | | | **Associated Use Cases:**  00002 |
| **Responsibilities:**   * Create a new course * Read an existing course * Update an existing Course * Delete and existing Course | | **Collaborators:**   * Maintain Class Records | |
| **Back:** | | | |
| **Attributes:**   * Course Identification : String * Course Name : String * Credit Hours : int * Description : String | | | |
| **Relationships**   * Generalization (a-kind-of): * Aggregation (has-parts): * Other Associations: Associated with Maintain Class Records | | | |

**CRC Card 2:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Front:** | | | |
| **Class Name:**  Class Records | **ID:**  002 | | **Type:**  Concrete Domain |
| **Description:**  Will allow the creation, viewing, modification and deletion of a class. | | | **Associated Use Cases:**  0003 |
| **Responsibilities:**   * Validate that class is associated to an existing course. * Create Class * Read Class * Update Class * Delete Class | | **Collaborators:**   * Maintain Course Records * Maintain Online Class * Maintain Face-to-Face Class | |
| **Back:** | | | |
| **Attributes:**   * Course Identification : String * Class Begin Date : Date * Class End Date : Date | | | |
| Relationships   * Generalization (a-kind-of): * Aggregation (has-parts): * Other Associations: Associated with Maintain Course Records and Register Student for class. | | | |

**CRC Card 3:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Front:** | | | |
| **Class Name:**  Online Class Records | **ID:**  003 | | **Type:**  Concrete Domain |
| **Description:**  Will allow the enrollment staff to add, modify, and/or delete attributes that are specific to an online class. | | | **Associated Use Cases:**  00004 |
| **Responsibilities:**   * Add class URL and browser to class records. * Read class URL and browser to class records. * update class URL and browser to class records. * Delete class URL and browser to class records. | | **Collaborators:**   * Maintain Class Records | |
| **Back:** | | | |
| **Attributes:**   * Class URL : String * Class Browser : String | | | |
| **Relationships**   * Generalization (a-kind-of): * Aggregation (has-parts): Maintain Class Records * Other Associations: | | | |

**CRC Card 4:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Front:** | | | |
| **Class Name:**  Face-to-face Class Records | **ID:**  004 | | **Type:**  Concrete Domain |
| Description:  Will allow the enrollment staff to add, modify, and/or delete attributes that specifically belong to a fate-to-face class. | | | **Associated Use Cases:**  00005 |
| Responsibilities:   * Add class building and classroom to class records. * Read class building and classroom to class records. * Update class building and classroom to class records. * Delete class building and classroom to class records. | | Collaborators:   * Maintain Class Records | |
| **Back:** | | | |
| Attributes:   * Class Building : String * Classroom : String | | | |
| Relationships   * Generalization (a-kind-of): * Aggregation (has-parts): Maintain Class Records * Other Associations: | | | |

**CRC Card 5:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Front:** | | | |
| **Class Name:**  Class Registration | **ID:**  005 | | **Type:**  Concrete Domain |
| **Description:**  will allow the enrollment staff and/or students to register for classes. | | | **Associated Use Cases:**  00006 |
| **Responsibilities:**   * Validate user privileges * Validate if student exist * Validate if Course exist * Validate if Class exist * Register student to class | | **Collaborators:**   * Maintain Student Records * Maintain Course Records * Maintain Class Records * Maintain Face-to-face Class * Maintain Online Class | |
| **Back:** | | | |
| **Attributes:**   * N/A | | | |
| **Relationships**   * Generalization (a-kind-of): * Aggregation (has-parts): * Other Associations: Associated with Maintain Class Records and Maintain Student Records | | | |

**CRC Card 6:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Front:** | | | |
| **Class Name:**  Student Records | **ID:**  006 | | **Type:**  Concrete Domain |
| **Description:**  will allow the enrollment staff to add, modify, and/or delete any courses offered by the college. | | | **Associated Use Cases:**  00001 |
| **Responsibilities:**   * Add a new student to record * Modify a student record * Delete a student record | | **Collaborators:**   * N/A | |
| **Back:** | | | |
| **Attributes:**   * First Name : String * Middle Initials : Char * Last Name : String * Date of Birth : String * Student ID : Int * Department : String * Cumulative GPA : Double | | | |
| **Relationships**   * Generalization (a-kind-of): * Aggregation (has-parts): * Other Associations: Associated with register Student for class | | | |

Generate your SIS class diagram:

Timeline

Description automatically generated

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

To verify and validate my CRC card I just made sure that my cards were filled will all the necessary information that was provided, specifically responsibilities, collaborators, attributes, and relationships. Lastly, I compare the information of my CRC and my use case to validate the all the necessary information was captured. Additionally, when implementing my class diagram, I made sure the no information got lost in transition. I compared my use-case, CRC, and class diagram and verified that although the look different visually, they all show the same information.

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

When approaching this problem my main goal was to use all the information I had already collected in previous weeks and ensure that that same information is reflected on my class diagram. First, I identified all the information that I need by using the use-case diagram I made in week 2. This allowed me to identify the classes that I needed to make on my class diagram. Second, I went back to the SIS requirements definition and identified all the attributes and functionalities that will need to be part of each class. After identifying all of the required information I started implementing my class diagram and placing the information in their respective sections.

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

By looking at my previous work, I was able to quickly identify all the required information that was needed and implement them efficiently in my class diagram.

When I came the class diagram itself, I this as of now the part that I find more challenging is how properly determine the relationships between the classes. There are some classes that I can quickly identified their relationship, but other take a little more time to determine. Overall, being introduce to the creation of class diagram and the entire process we have covered has really allows me to better understand the importance of preparation before coding. Although, there are many missing pieces to make this a functional program, just by looking at how organized everything is I feel confident that I would easily be able to translate this into a functional code.