# Week 3—Class Diagram and CRC Cards

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Generate a class diagram for the SRS system. The diagram must be generated by a UML drawing tool. Copy and paste your diagram here:  **SRS Class Diagram**    Use this CRC template and complete a CRC card for each class you designed in your SRS class diagram.  **Class1 CRC Card**   |  |  |  |  | | --- | --- | --- | --- | | **Front:** | | | | | **Class Name:** School Staff | **ID:** 1 | | **Type:** Concrete, System | | **Description:** The School Staff that control over Students, Classes, Courses, and Registrations. | | | **Associated Use Cases:** School Staff | | **Responsibilities**  Register Student, Manage Student Records, register students for the classes, manage classes, Manage Courses | | **Collaborators**  School Staff | | | **Back:** | | | | | **Attributes:**  School Staff ID, School Staff Name | | | | | **Relationships:**  School Staff has control over Students One to Many. School Staff has control over Courses One to Many, School Staff has control over Class One to Many, School Staff has control over Registration One to many  **Generalization (a-kind-of): None**  **Aggregation (has-parts):** None  **Other Associations:**  Every association with each class is One to Many regards that School Staff has control over any System in the school | | | |   **Class2 CRC Card**   |  |  |  |  | | --- | --- | --- | --- | | **Front:** | | | | | **Class Name:** Student | **ID:** 1 | | **Type:** Concrete, System | | **Description:** Student has a right to register for the classes and check his information in the system | | | **Associated Use Cases:** Student | | **Responsibilities**  Providing Student Information, Getting Student GPA, (Updating Deleting and Adding) Student Information, register for the Class, Updating or Deleting the registration | | **Collaborators**  Student | | | **Back:** | | | | | **Attributes:**  First Name, Middle Initials, Last Name, Student ID, department (e.g. Grad Law, Undergrad IT), and Date of Birth, GPA (scale of 1.0 - 4.0) | | | | | **Relationships:**  Student can register for many classes when it associates with his GPA and requirements. Student has a right to search for the course and search for the class that he/she need to register for.  **Generalization (a-kind-of):** None  **Aggregation (has-parts):** None  **Other Associations:** Student has limited roles and he/she can only manage the registration and his/her own account or view personal information. | | | |   **Class3 CRC Card**   |  |  |  |  | | --- | --- | --- | --- | | **Front:** | | | | | **Class Name:** Course | **ID:** 1 | | **Type:** Concrete, System | | **Description:**  The collection of the classes in the course. Course contain the list of classes that student can take to accomplish his/her degree. | | | **Associated Use Cases:**  Course | | **Responsibilities**  School Staff can Update/Create/Delete Course, the course contains prerequisite courses, student/school staff can search for the course name | | **Collaborators**  Course, Student, School Staff | | | **Back:** | | | | | **Attributes:**  Course ID, Course Name, Credit Hours, Description, Prerequisite Course | | | | | **Relationships:** Each course can contain many classes. Each student/school staff can search for the courses. Each course can have multiple registrations with many classes. School staff can manage many courses  **Generalization (a-kind-of):** none  **Aggregation (has-parts):** none  **Other Associations:** none | | | |   **Class4 CRC Card**   |  |  |  |  | | --- | --- | --- | --- | | **Front:** | | | | | **Class Name:** Class | **ID:** 1 | | **Type:** Concrete, System | | **Description:**  The class that student can register for to proceed with their degree. | | | **Associated Use Cases:**  Class | | **Responsibilities**  Get type of the class and set it to be online class or face-to-face class. If the class is online then it will add class URL and class browser, if the class is face-to-face then it will add class building and class room. Each class can be deleted/added/updated. Each class display information about the class | | **Collaborators**  Student, School Staff, Class | | | **Back:** | | | | | **Attributes:**  Course ID, Class Begin Date, and Class End Date. In addition, online classes also should add or maintain this additional information: Class URL, Class Browser. Similarly, face-to-face classes should add or maintain this additional information about them: Class Building, Class Room. | | | | | **Relationships:**  Each class can maintain multiple registration, each class can be assigning to multiple courses because each course can be different but share same class. Classes can be register many to many. Class can be updated/deleted/added by School Staff. Student/School Staff can view a list of classes  **Generalization (a-kind-of):** none  **Aggregation (has-parts):** none  **Other Associations:** | | | |   **Class 5 CRC Card**   |  |  |  |  | | --- | --- | --- | --- | | **Front:** | | | | | **Class Name:** Registration | **ID:** 1 | | **Type:** Concrete, System | | **Description:**  Registration that associate with classes, course, and students | | | **Associated Use Cases:**  Registration | | **Responsibilities**  Each registration can contain the cours, class, and students list. School Staff can update/delete/add the registration. Student can view his/her registration | | **Collaborators**  Student, School Staff, Registration | | | **Back:** | | | | | **Attributes:**  Registration ID, Course ID, Class ID, Student ID | | | | | **Relationships:**  Each registration can contain multiple classes and course many to many. School staff can update/delete/add registration for the student many to many. Student can view his registrations many to many.  **Generalization (a-kind-of):** none  **Aggregation (has-parts):** none  **Other Associations:** | | | | |
| 2 | Validate and verify your class diagram and CRC cards against the SRS use case diagram and use case descriptions. Due to limitation of the student access, he/she can only register and search for course or class for their needs. On the other hand, School staff has a full control over the data like student, registration, class, and course. Each class has his own Setter and Getter which helps to define and get information for the School Staff/Student. On them behave of access they can delete/update/add the information for each class attributes. Each attribute follows with autogenerated ID which helps to create a unique field for the user. |
| 3 | Explain how you completed your work, the decisions you made to arrive at your conclusions, and the lessons you learned.  The following week for me was actually very interesting, I have learned how to create a simple UML class diagram and used to create my own system for example a SOS system that helps students to registers for needed class. On the other hand, it really helps to understand how the data will be displayed in the database to store or retrieve some of the information. Each class has a relationship which is associate with many to many or one to many relationship. |