# Week 2—Use Case Diagram and Use Cases Descriptions

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | Generate a use case diagram for all of your use cases, including their actors. The diagram must be generated by a UML drawing tool. Copy and paste your diagram here:  **SRS Use Case Diagram**            Use this use case description template and complete the two SRS use cases of (1) Maintain Class Records and (2) Register a Student for Classes.  **Maintain Class Records**   |  |  |  |  | | --- | --- | --- | --- | | Use Case Name: Maintain Class Records System | | ID:123-123 | Importance Level: Normal | | Primary Actor: Staff | Use Case Type: | | | | Stakeholders and Interests: School Staff – maintain (create, edit, modify, and delete) class records for a given course in Student Online System <<system>> Class – store all classes in the database. Actor (School Staff) can only update or delete class per one selection. Relation ship Many classes to one selected class. | | | | | Brief Description: This use case maintains the school records of the classes it offers. By class in course. It has an option to Create, Read, Update, and Delete. It provides the School Staff actor with complete control over the creation, deletion, modification and the reading of class records for a given course. | | | | | Trigger: A School Staff wanting to maintain the class records for a given course  Type: | | | | | Relationships:  Association: with the School Staff actor  Include: relationship to the Login into SRS use case  Extend: Internet connection and user verification, Search Course ID  Generalization: - | | | | | Normal Flow of Events:   * 1. User getting verification token   2. The school staff enters the course whose classes are to be maintained   3. The system responds with a list of all classes for the entered course   4. The School Staff then select one of the classes and commands the system to perform one of the following operations on the selected class:      1. Delete the selected class      2. Modify certain fields in the selected class   5. Alternative way, the School Staff actor can create a new class offering for the course, where he/she cam select between online or face-to-face class, and then enter the required infmation for all kinds of classes and for each classification as fallow      1. For all classes: Class ID, Class Begin Date and End Date      2. For Online classes: Class URL, Class Browser      3. For Face-to-Face classes: Class Building, Class Room | | | | | SubFlows: - | | | | | Alternate/Exceptional Flows:  Alternative flow: each one of the CRUD operations is considered an alternative flow through the use case.  Exceptional flow: when the course entered, it might not exist in the system. | | | |   **Register a Student for Classes**   |  |  |  |  | | --- | --- | --- | --- | | Use Case Name: Register Student for Classes System | | ID: 123 | Importance Level: Normal | | Primary Actor: School Staff, Student | Use Case Type: Detail, Essential | | | | Stakeholders and Interests:  Student – self-register for classes School Staff – register on behalf of students <<system>> Classes – store all classes in database <<system>> Courses – store all courses that related to the classes in database <<system>> Registration – store all registration that was made by School Staff or Student in the database | | | | | Brief Description: This use case enables the actor to select a class (online or face-to-face) and then registers a student for the selected class provided that the student meets the registration requirements for the class. | | | | | Trigger: an actor (Student, School Staff) who wants to register a student for a class.  Type: | | | | | Relationships:  Association: with an actor (Student, School staff)  Include: relationship to the Login into SRS use case, Validation of registration, list of available courses, confirmation registration, violation registration, select class of selected course.  Extend: Verification of the user, Internet connection,  Generalization: <system>> Course is set of <<system>> Classes, All students are related to the <<system>> Registration database. | | | | | Normal Flow of Events:   1. User getting verification token 2. The system identifies the student implicitly from the login (if the actor is a Student) or by explicitly entering the student ID (if the actor is a School Staff) and verifying it. 3. The actor enters the course for which classes registration is thought 4. The system responds with a list of classes (pulls the list from <<system>> Class database) offered of the selected course (<<system>> Course list from database) 5. The actor then selects a class from the list of classes 6. The actor then commands the system to register the identified student for the selected class 7. The system verifies the registration rules as applied to the student and class (<<system>> Registration verify all available course from being duplicated). 8. If registration rules are verified, then the student is registered for the class and the registration is confirmed (<<system>> Registration send record to save to database). If not, then an error message is printed to the actor with the reason why registration cannot be competed at this time. 9. The system allows the actor to re-start the use case again beginning with event to register for another class 10. System will automatically update registration after class is completed and all other records like grades will be send to the Student. 11. Student on his/her behalf will maintain the course dropping if it considers using in the future. | | | | | SubFlows: - | | | | | Alternate/Exceptional Flows:  Alternative flow: same as above normal flow except validation of registration rules fails and the system prints out a message for the reason why the registration cannot be completed.  Exceptional flow: then the entered student ID is might not found in the system | | | | |
| 2 | Validate and verify your use case diagram and use case descriptions against the SRS Requirement Definition and the SRS System Request.  The area of this system is to save time for actor (School Staff, Student) Each actor has a right to control the system, however, the limitations of the Student actor is only allowed register and drop classes, when School Staff has a full control over the Courses, Classes, Registration and Student Records. Each access requires a validation token in order to take control over the system. |
| 3 | Explain how you completed your work, the decisions you made to arrive at your conclusions, and the lessons you learned.  This week assignment is simple but use of SRS system took me a while to understand. This week I learn how to create UML Use Case diagram by only using Activity Diagram. Also, I learn how to create a subsystem items in order to control database and user behavior. Include and extract the use case for each behavior and associate the records between use case and actor. |