# Week 4—Sequence, Communication, and State Machine Diagrams

|  |  |  |
| --- | --- | --- |
| 1 | Generate a sequence diagram for the SRS system Register a Student for Classes use case. The diagram must be generated by a UML drawing tool. Copy and paste your diagram here:  **Sequence Diagram for the Register a Student for Classes Use Case**     |  | | --- | | Generate a communication diagram for the SRS system Register a Student for Classes use case. The diagram must be generated by a UML drawing tool. Copy and paste your diagram here:  **Communication Diagram for the Register a Student for Classes Use Case** |   Generate a state machine diagram for the SRS system Registration Record object/class (the class that maintains the registration of a student in a class). The diagram must be generated by a UML drawing tool. Copy and paste your diagram here:  **State Machine Diagram for the Registration Record Object** |
| 2 | Validate and verify your behavioral diagrams against the SRS Register a Student for Classes use case description and the SRS class diagram.  For the each of the diagram that displayed above we can see that Sequence diagram is built on the school staff actor interaction where he/she login to the system with a right validation, then he/she search for course and class. Which then follow with registration of the student. The community diagram represents similar behavior as sequence diagram, but in the community diagram we must follow the steps to record the school staff interaction via SOS systems. Finally, State Machine diagram represent a one-way state with events that follows each state. Each event represents a proportion of the school staff anchor in the system. In conclusion, the diagrams that was represented above shares same type of Use Case that was build in week 1, and it represent a different type of messages in which school staff has an ability to register a student for the classes. |
| 3 | Explain how you completed your work, the decisions you made to arrive at your conclusions, and the lessons you learned. Due to number of diagrams I had to consider the same use case that I use in week 1 and compile it three different type of diagrams that shares similar behavior. In overall to complete the task I had to understand what each of those diagrams should be look like and how the behave to each situation when we have limited amount of the information. Even though, state machine diagram has different approach of displaying states and events. In conclusion, this week was interesting and well complicated to challenge us. |