# IT 315 Final Project Part I Solution Submission Template

**Name:** Byron LaFerriere

**Date:** 22-Nov-2020

1. **Creation:** Generate your student information system (SIS) use case diagram. Refer to textbook pages 121–129.

The diagram below is used to map out how the classes assigned from use cases will interact with each other and, therefore how the system will be designed. The Functional Model is merely made to provide a brief overview of this interaction and will not go into detail about the attributes or methods each class will use. Each corresponding ID can be traced back to a Requirements Traceability and Verification Matrix (RTVM) that I will provide later in the document. This corresponding information will be available to view in Column H of the RTVM.



*Using the* ***use case description template*** *(refer to textbook pages 141–148), provide a description for* ***each*** *use case in your use case diagram:*

The following Use Cases Descriptions are used to provide further detail into the previously displayed Functional Model. While this model is a basic overview, these Use Case Descriptions provide more detail into how each case should interact with the system, system triggers, and the normal flow of events once the Use Case has been triggered by the actor.

**USE CASE #1:**

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Maintain Course Records | ID:  FM-002 | Importance Level:  HIGH |
| Primary Actor:  Staff Member | Use Case Type:  Detail, Essential | |
| Stakeholders and Interests:  Staff Member- Modify, update, and change Course attributes as needed | | |
| Brief Description:  Provide Staff Member CRUD capabilities and other course attribute updates. | | |
| Trigger:  Staff Member logs in to begin making adjustments to Course attributes and pulls up option to Maintain Course Records.  Type:  External | | |
| Relationships   * Association:   Staff Member   * Include:   n/a   * Extend:   n/a   * Generalization:   n/a | | |
| Normal Flow of Events:   1. Staff Member logs in with their username and password. 2. Staff Member is prompted to answer question about intention of login. 3. Staff Member selects to Maintain Course Records. 4. Staff Member is then provided with 8 options that derive from Maintain Course Records. | | |
| SubFlows:  n/a | | |
| Alternate/Exceptional Flows:  n/a | | |

**USE CASE #2:**

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Maintain Class Records | ID:  FM-003 | Importance Level:  HIGH |
| Primary Actor:  Staff Member | Use Case Type:  Detail, Essential | |
| Stakeholders and Interests:  Staff Member- wants to edit Class Records | | |
| Brief Description:  Provide Staff Member with basic CRUD capabilities for Class Records | | |
| Trigger:  Staff Member logs in with supplied username and password and selects the option to edit Class Records.  Type:  External | | |
| Relationships   * Association:   Staff Member, Maintain Course Records   * Include:   n/a   * Extend:   n/a   * Generalization:   n/a | | |
| Normal Flow of Events:   1. Staff Member logs on to website using supplied username and password. 2. Staff Member selects option that leads to Class Records 3. Staff Member now has access to Class Records of classes they teach only. 4. Staff Member can use CRUD capabilities to edit classes. | | |
| SubFlows:  n/a | | |
| Alternate/Exceptional Flows:  n/a | | |

**USE CASE #3:**

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Online Class | ID:  FM-004 | Importance Level:  MEDIUM |
| Primary Actor:  Staff Member | Use Case Type:  Detail, Essential | |
| Stakeholders and Interests:  Staff Member- need access to providing access to online hub. | | |
| Brief Description:  Basic information related to where to find the Class URL and Class Browser. | | |
| Trigger:  Staff Member is looking through class files and clicks on link to take her to Class Records homepage online.  Type:  External | | |
| Relationships   * Association:   n/a   * Include:   n/a   * Extend:   Maintain Class Records   * Generalization:   n/a | | |
| Normal Flow of Events:   1. Staff Member gains entry to Course and Classes Record System. 2. Staff Member selects to modify an Online Course they are teaching. 3. Staff Member is granted entry through URL or Browser. | | |
| SubFlows:  n/a | | |
| Alternate/Exceptional Flows:  n/a | | |

**USE CASE #4:**

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Face-to-Face Class | ID:  FM-005 | Importance Level:  MEDIUM |
| Primary Actor:  Staff Member | Use Case Type:  Detail, Essential | |
| Stakeholders and Interests:  Staff Members- ability to use CRUD capabilities for Classes being taught by Staff Member. | | |
| Brief Description:  Staff Members that are teaching Face-to-Face or On-Campus Classes, will have the ability to CRUD for all Classes they are teaching that semester. | | |
| Trigger:  Staff Member joins domain and requests entry to the Class they are teaching by clicking on it.  Type:  External | | |
| Relationships   * Association:   n/a   * Include:   n/a   * Extend:   Maintain Class Records   * Generalization:   n/a | | |
| Normal Flow of Events:   1. Staff Member gains entry to the Courses and Classes Records System 2. Staff Member selects option to edit a Face-to-Face Class that they are teaching this semester. 3. Staff Member can CRUD all options within Class, as well as ClassBuilding and ClassRoom. | | |
| SubFlows:  n/a | | |
| Alternate/Exceptional Flows:  n/a | | |

**USE CASE #5:**

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Maintain Student Records | ID:  FM-008 | Importance Level:  MEDIUM |
| Primary Actor:  Enrollment Staff | Use Case Type:  Detail, Essential | |
| Stakeholders and Interests:  Enrollment Staff- Keeping student files up to date and ensuring all attributes are correct. | | |
| Brief Description:  Enrollment Staff needs the ability to CRUD all attributes listed in Student Records. | | |
| Trigger:  Enrollment Staff selects option to gain entry into Student Records Portal.  Type:  External | | |
| Relationships   * Association:   Enrollment Staff   * Include:   n/a   * Extend:   n/a   * Generalization:   n/a | | |
| Normal Flow of Events:   1. Enrollment Staff logs in with supplied username and password to hub. 2. Enrollment Staff selects option to maintain Student Records. 3. Enrollment Staff is provided options pertaining to attributes in Student Records. | | |
| SubFlows:  n/a | | |
| Alternate/Exceptional Flows:  n/a | | |

**USE CASE #6:**

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Register Students for Classes | ID:  FM-009 | Importance Level:  MEDIUM |
| Primary Actor:  Student/ Enrollment Staff | Use Case Type:  Detail, Essential | |
| Stakeholders and Interests:  Student- Ability to personally register for upcoming classes, without needing the help of advisor  Enrollment Staff- Ability to help Student, if requested, to sign up for upcoming classes. | | |
| Brief Description:  Portal that provides Students and Enrollment Staff the capability to register for Classes, both Online and Face-to-Face. | | |
| Trigger:  Either Student or Enrollment Staff requests entry to the portal to register for Classes.  Type:  External | | |
| Relationships   * Association:   Student/ Enrollment Staff   * Include:   n/a   * Extend:   n/a   * Generalization:   n/a | | |
| Normal Flow of Events:   1. Either actor logs in to portal with supplied credentials. 2. The actor selects to register for classes 3. The actor selects a class to register for and then the Enrollment Staff will update Student Records, as necessary. | | |
| SubFlows:  n/a | | |
| Alternate/Exceptional Flows:  n/a | | |

**Use Case 1 description template:**

This class can be considered a ‘stand-alone’ class in our Structural Model. While the only association it has is with the Login Authentication System, the rest of the system relies on this to operate. The class will provide Staff Members with CRUD abilities for all attributes listed.

**Use Case 2 description template:**

This class inherits from MaintainCourseRecords, as no Class can exist without the Course. This class is also a ‘parent’ class to Online Classes and Face-to-Face Classes. This class also needs to provide Staff Members with CRUD abilities for the inherited attributes, as well as the independent attributes specific to the class.

**Use Case 3 description template:**

This class inherits from MaintainClassRecords, as no Online Class can exist without the Class. This class is considered a ‘child’ class because of this and inherits all attributes and responsibilities from MaintainClassRecords. It also introduces new independent attributes that are specific to Online Classes only.

**Use Case 4 description template:**

This class also inherits from MaintainClassRecords, as no Face-To-Face Class can exist with the Class. This class is also considered to be a ‘child’ class because of this and inherits all attributes and responsibilities from MaintainClassRecords. It also introduces new independent variables that are specific to Face-To-Face Classes only.

**Use Case 5 description template:**

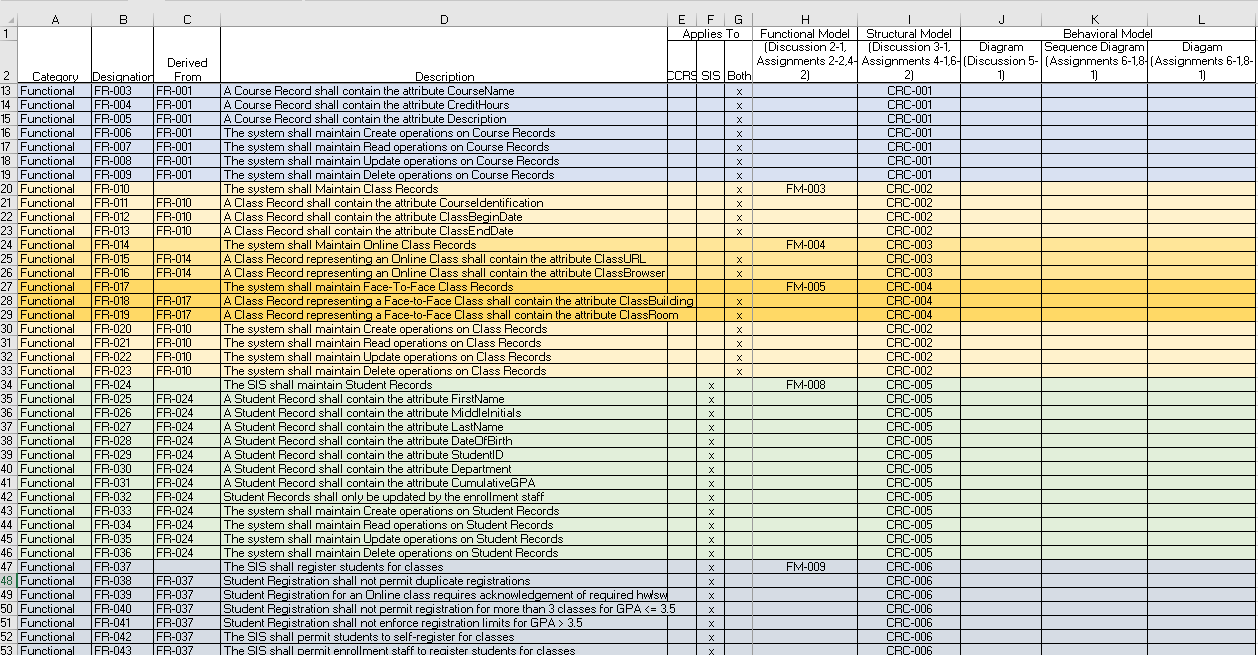
This class breaks from the previous inherited classes to create a new class that will provide necessary information for its dependently related class RegisterStudent4Classes. It is provided with its own CRUD capabilities, as well as independent attributes related to individual student information.

**Use Case 6 description template:**

This class is meant to act alongside MaintainStudentRecords and carries many of the doing responsibilities required for student registration. It has a dependency relationship with MaintainStudentRecords which means that it requires the class for instantiation and cannot operate without it.

1. **Testing:** Verify and validate your use case diagram and use case descriptions against the SIS requirements definition.

The best way to validate and verify that the Use Cases contained all required functionality from the Courses and Classes Requirements Definitions page, was to supply a spreadsheet containing the requested use cases. The attached Requirements Traceability Verification Matrix (RTVM) contains these use cases, as well as additional information about where these needs were met and, on which model they were satisfied within. In column H, the RTVM displays where the use cases can be found on the Functional Model. When viewing the Functional Model, look for the corresponding values assigned in this column, which can be matched back to use cases being defined and constructed on the Functional Model. In column I on the RTVM, the use cases have similar corresponding values that align with the Structural Model, which does not apply to this Model. This was merely provided here to create a symbiotic presence between the two models. Each use case identified in the Course and Classes Records System has been met and assigned to a Use Case within this model and the Use Case templates.



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

To create this Functional Model, I went back through previously defined documentation to find the information needed. I used the CCRS Requirements Definitions document that was provided and the RTVM. Once the Use Case templates had been filled in with the information provided to me, I was able to better ensure the Functional Model would be consistent throughout. To stay true to this mentality, when designing the model, I included the use case information and the Functional Model ID’s to make cross-referencing easier for the viewer. Starting with our MaintainCourseRecords, this class is the foundation for the system. This means that all other classes require this to exist for instantiation. There is an association or more specifically a composition relationship that was chosen between MaintainCourseRecords and MaintainClassRecords. This was done because MaintainClassRecords is a-part-of MaintainCourseRecords and cannot be instantiated without it (Dennis, Wixom, & Tegarden, 2015). A generalization relationship would not have been appropriate here because MaintainClassRecords was given its own CRUD capabilities and adds new attributes to the system. It does need to inherit from MaintainCourseRecords. There was an <<extend>> relationship chosen between the ClassRecords and OnlineClasses and F2FClasses because it was adding new functionality to the system, that had not previously been included. An <<include>> relationship would not be appropriate in this situation because an include relationship is meant to be a re-use of functionalities previously defined (Dennis, Wixom, & Tegarden, 2015). We would not have used a generalization relationship in this case either because that is used when inheritance is needed between two classes with minimal addition of new functionality (Dennis, Wixom, & Tegarden, 2015). Each class definition was chosen based on the class properties, with only MaintainClassRecords being an abstract (parent) class. This is so, because of the attributes that it passes along to its child classes, MaintainOnlineClassRecords and MaintainF2FClassRecords (Dennis, Wixom, & Tegarden, 2015). The other five classes were all chosen to be standard Domain classes.

When designing the Use Case Templates it was important to keep these attributes and responsibilities of each case in mind. The use case is the building block of the system (Dennis, Wixom, & Tegarden, 2015) and therefore, needs to be done correctly the first time. When assigning the ID to each case, it is important to stay consistent with naming that way one could easily find each case on the Functional Model and vice versa. The level of importance is chosen by the user and used to specify business functions they would like to see on the first releases of the product (Dennis, Wixom, & Tegarden, 2015). The primary actor for each use case can be defined by the user who would instantiate the use case or in other words trigger the action (Dennis, Wixom, & Tegarden, 2015). To fill in the stakeholders and interest section, I used the primary actors identified and provided detail as to just how those actors would be using the function of the system. One of the most important parts of this functional model was to begin describing the flow of events in the last section of the use case templates. In this area, I began to think about exactly what would be done in each function and possible sub-flows of those events as well. Sub-flows are used to help keep the normal flow of events as simple as possible (Dennis, Wixom, & Tegarden, 2015). The completion of these templates will provide a good sense of direction when designing the Structural Model.

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

The most important lesson learned during the process of creating the Functional Model was to think of it as a foundational piece for the rest of the system. Symbiosis seems to be an important element throughout this iterative and incremental process and the Functional Model is an important piece to this. By starting with an accurate overall depiction of what will be required by the system, functionality-wise, it provides a proper layout for the rest of the systems. It is important to establish correct definitions from use cases, that allow for the creation of the Structural Model to become easier. This entails properly establishing use case types, level of importance, and flow of events. I particularly noticed how frustrating it was when I had to correct something on the Functional Model, that it trickled all the way through my Structural Model and CRC Cards. The next system design that I lay out, I will spend more time with the Functional Model and think things through to a deeper level in hopes of avoiding this repetitive process.

References:

Dennis, A., Wixom, B. H., & Tegarden, D. (2015). VitalSource Bookshelf Online. Retrieved November 07, 2020, from <https://mbsdirect.vitalsource.com/>