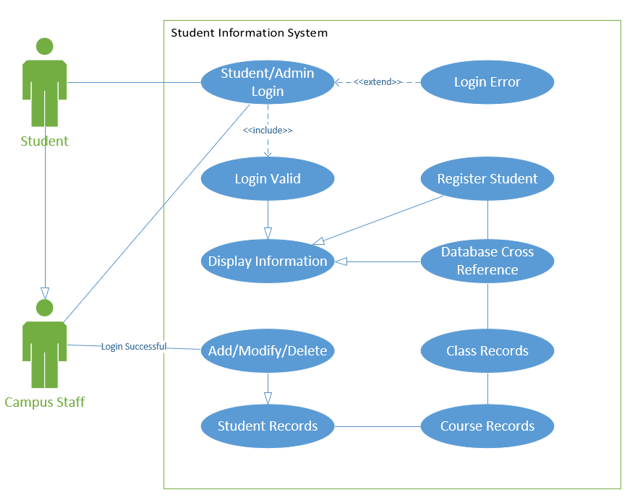
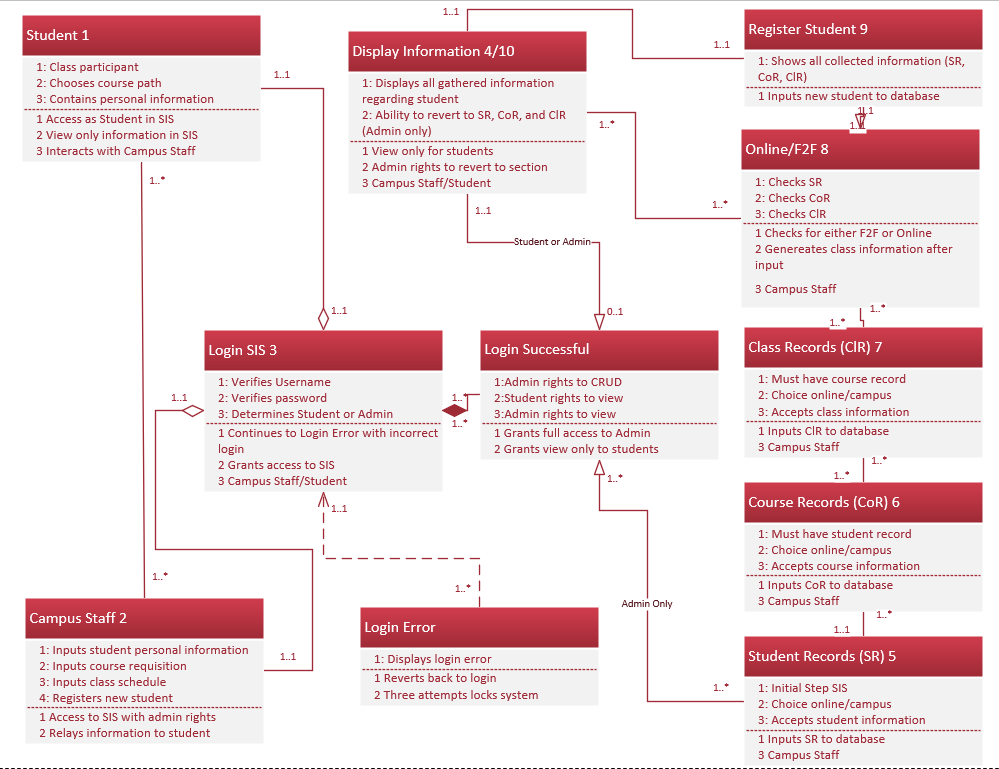
Eric Clancy | 3/20/2021

Southern new hampshire university  | eric.clancy@snhu.edu

Software engineering and design

With the implementation of software engineering design, the initial step was to create the idea without coding. This is defining the structure you want to obtain for the final product. The concept is set with the first draft of the student information system as seen in the diagram below. 

This draft lays out the main functions that are needed to cover for the application to be complete. There is not much detail just an overall layout that gets the team or yourself a pathway to the end of the project. The first example above with this particular diagram is referred to as a use case diagram. The two actors in this use case diagram are the “Student” and the “Campus Staff”. The actors will be the indivudals that interact with the student information system application. Each oval that is displayed are known as a use case. The use cases are the distinct functionality of the system that is wanted in the application. The use case diagram should be as simple as possible with the constuction of the diagram. This should be the most essential properties that are needed to construct.

The next step for the team or yourself is to know that if there are details that need examined you can generate a class diagram of the student information system. This class diagram can break down more detail for the application. The kind of model a class diagram is would be a static model. Which either working with yourself or others will help with the reduction of maintenance that more than likely is going to occur during the projects build. Along side of this diagram you can incorporate CRC (Class, Responsibilites, Collaborators) cards. 

With the class digram created the CRC cards should have more detail about each use case that is significant within the project. This is where we can start to establish variable names that is going to be used in the code and will be a quick reference guide if troubleshooting occurs. As an example the numbering on the “Student” box has a numeric value of one. This is to tie the CRC card directly to that information in that box. The CRC card with have the use case name which is “Student” the identification of that box on the diagram which is the value of one. An importance level tied to that use case. The primary actor that is associated with the use case which is the student accessing the student information system. We can see the example of the CRC card below.

**Use Case 1 description:**

|  |  |  |
| --- | --- | --- |
| Use Case Name:  Student | ID:  1 | Importance Level:  High |
| Primary Actor:  Student | Use Case Type:  Overview | |
| Stakeholders and Interests:  Campus Staff | | |
| Brief Description:  Has student information and requests course path | | |
| Trigger:  Requests course information or views student records (SR, Cor, ClR)  Type:  External | | |
| Relationships   * Association:   Campus Staff   * Include:   Login SIS   * Extend:   N/A   * Generalization:   N/A | | |
| Normal Flow of Events:  1. Talks with campus staff  2. Logs into SIS | | |
| SubFlows:  1. View only SR, CoR, ClR | | |
| Alternate/Exceptional Flows:  1. When online not with campus sends email or requisition | | |

This card ties directly to the class diagram above.

The important thing to keep in mind is the four phases of systems development life cycle. This consists of planning, analysis, design and implementation. While maintaining clarity with the association relationships between the actors and the use cases. This can best describe the communication between them. The extension of the functionality is maintained with the extended relationships between the use cases and the behavior that is wanted. The relationships need that detail to make sure that it is concise with the wanted outcome. Once the planning and anaylsis are incorporated into your design the next push would be to start drafting code for the application. From the use case diagram showing the general layout that is wanted to a more detailed breakdown in the class diagram, you are at a great starting point for the code to begin. Once the code is generated the implementation of that system is going to execute from the team that started this project.