**Week 2 Assignment – UML Design Modeling**

Vicki Kelm

The University of Arizona Global Campus

CST 499: Capstone for Computer Software Technology

Dr. Amjad Alkilani

January 07, 2024

**Week 2 Assignment – UML Design Modeling**

“Unified Modeling Language (UML) is an object-oriented modeling language that provides the elements and relationships to model software requirements and design” (Tsui, Karam, & Bernal, sect. 4.3.1). This paper will discuss the details for the different levels of testing that follow component testing, integration testing, system testing, and acceptance testing. The paper will also include different UML models to capture the key design and requirement aspects of the system based on the SRS document developed in Week 1. The models will include the following: Class diagram, Sequence diagram, Activity diagram, State diagram, and Use Case diagram.

**Discuss the details for the different levels of testing that follow: Component testing, Integration testing, System testing, and Acceptance testing**

“Within each test level, the tester must make sure the outcomes of development meet the requirements that are relevant or specified on this level of abstraction” (Spillner, Linz, & Schaefer, p. 41). Testing provides several benefits to include reduced costs in post-development, supports maintainability, discovers defects and errors earlier in the development cycle, and provides a product to the stakeholders that meets their requirements. The different levels of testing include Component Testing, Integration Testing, System Testing, and Acceptance Testing.

**Component Testing**

Component testing is the first level of testing where the software units are tested systematically for the first time. “In object-oriented programming, the software units are called classes” (Spillner, Linz, & Schaefer, sect. 3.2.1). In relation to unit testing, component testing is performed after the development team completes the unit testing and the build has been released for the testing team. “Component testing involves the testing of each object or software parts separately with or without isolation of other objects” (Hamilton, 2023). This type of testing plays a crucial role in finding bugs and is completed before integration testing.

**Integration Testing**

The second level of testing is integration testing where “units or individual components of the software are testing in a group” (javaTpoint, n.d.) In this level of testing defects are exposed at the time of interaction between integrated components or units. Once the individual components are tested and pass the testing in their solo capacity, then test cases will be generated in order to evaluate the components when they work together or rather integrated. According to GeeksforGeeks, integration testing focuses on determining the correctness of the interface and its purpose it to expose faults in the interaction between integrated units. This otherwise tests the flow of data between the dependent modules.

**System Testing**

The next level of testing is system testing is also the third level of testing in the software development process. This form of testing is directly associated with the system design phase. System testing is used to check the entire system functionality and the communication of the system under development with external systems. Most of the software and hardware issues of compatibility can be uncovered during this system test execution. The testing verifies that the application performs tasks as designed and intended. “System testing is imperative for inter-connected systems because any system or software defect can cause extreme complications for the consumers” (Yasar, 2023). There are several benefits of system testing which includes:

* Improved product quality
* Error reduction
* Cost savings
* Security
* Customer satisfaction

**Acceptance Testing**

The last phase of software testing is acceptance testing and is performed before making the system available for utilization. “It is a formal testing according to user needs, requirements and business processes conducted to determine whether a system satisfies the acceptance criteria or not and to enable the users, customers or other authorized entities to determine whether to accept the system or not” (GeeksforGeeks, 2022). Advantage of acceptance testing include:

* Helps the project team to know the further requirements from the users directly as it involves the users for testing.
* Automated test execution
* It brings confidence and satisfaction to the clients as they are directly involved in the testing process.
* It is easier for the user to describe their requirement.
* It covers only the Black-Box testing process and hence the entire functionality of the product with be tested.

The designing of UML models provides the tools needed for the development team to provide validation that the developed software meets the needs of the customer. The testing ensures a high-quality product and allows for a quick and efficient development process. This paper discussed the details for the different levels of testing that follow component testing, integration testing, system testing, and acceptance testing. The paper also included different UML models to capture the key design and requirement aspects of the system based on the SRS document developed in Week 1. The models included are a Class diagram, Sequence diagram, Activity diagram, State diagram, and a Use Case diagram.

**UML Diagrams**

**Figure 1**

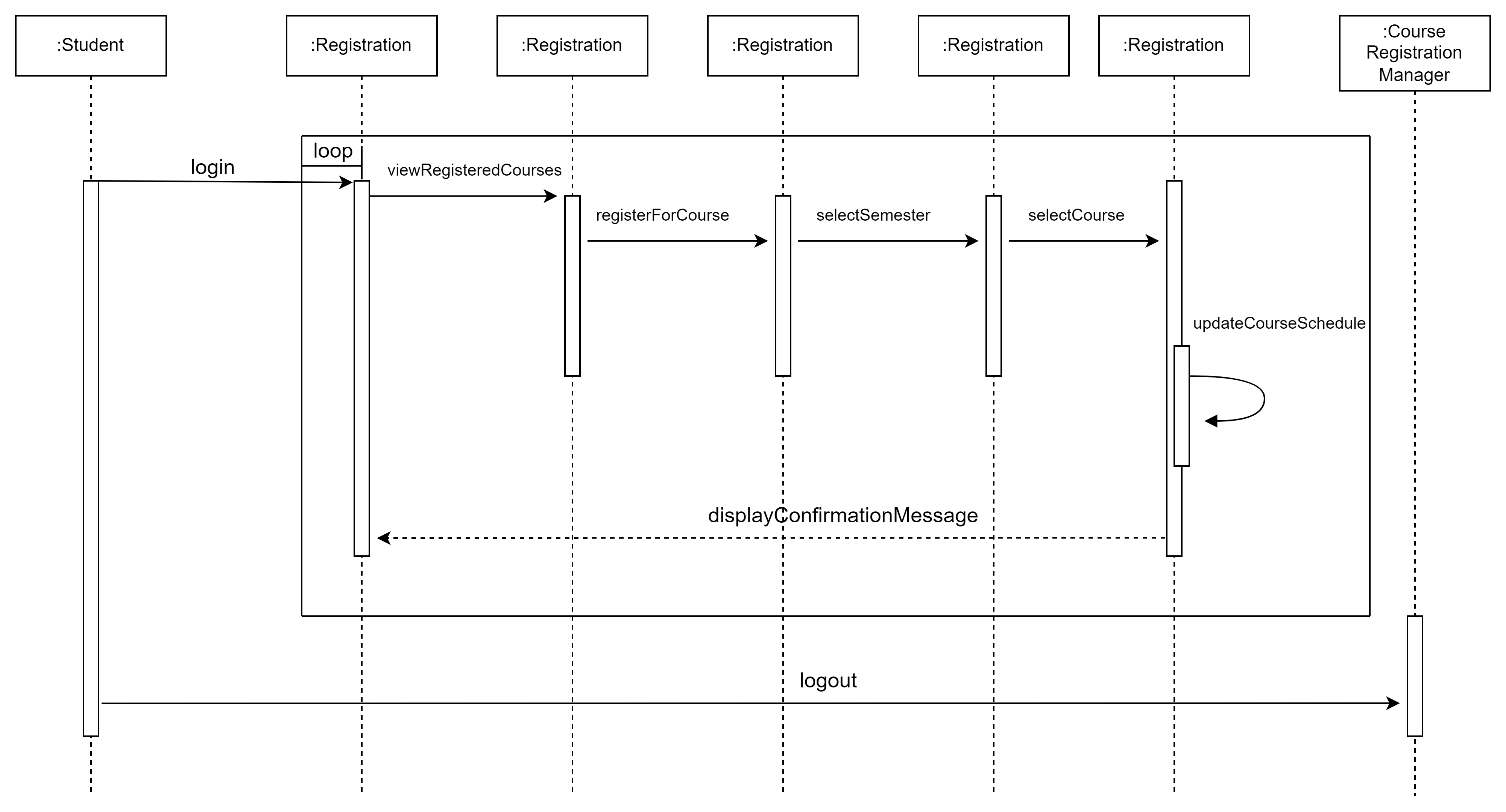
Class Diagram



Note. UML Class Diagram for Student Registration System

**Figure 2**

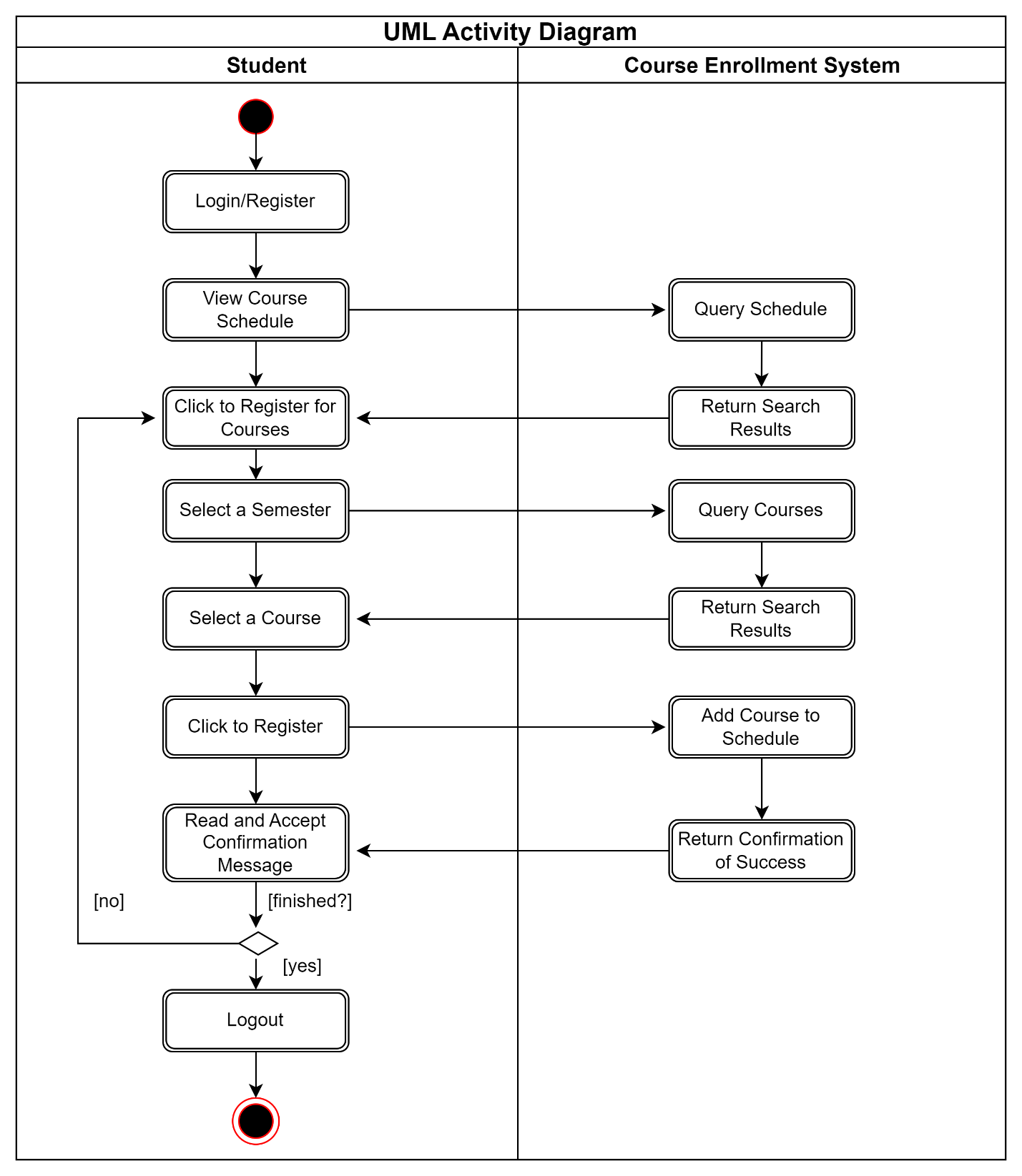
Sequence Diagram



Note. UML Sequence Diagram for Student Registration System

**Figure 3**

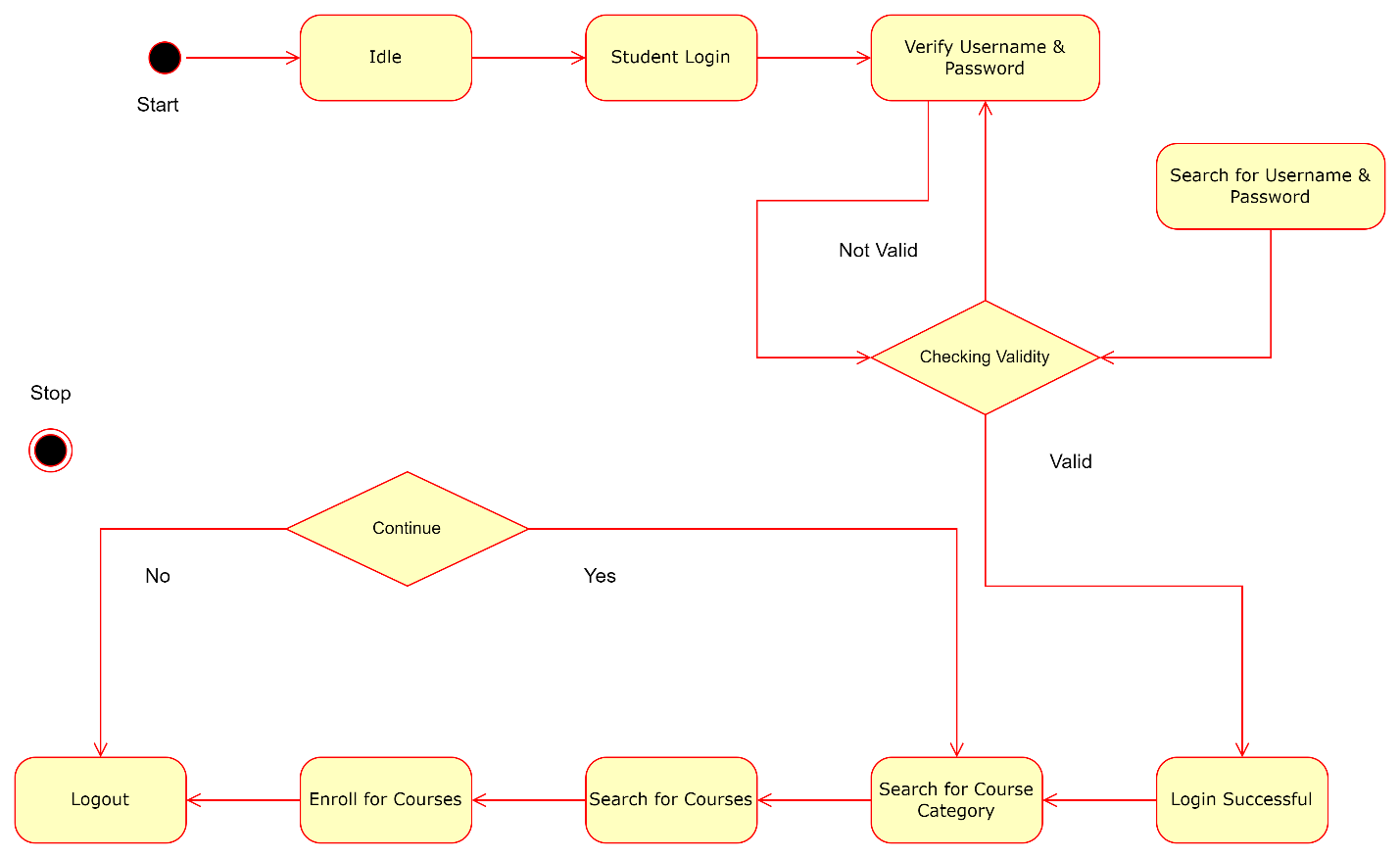
Activity Diagram



Note. UML Activity Diagram for Student Registration System

**Figure 4**

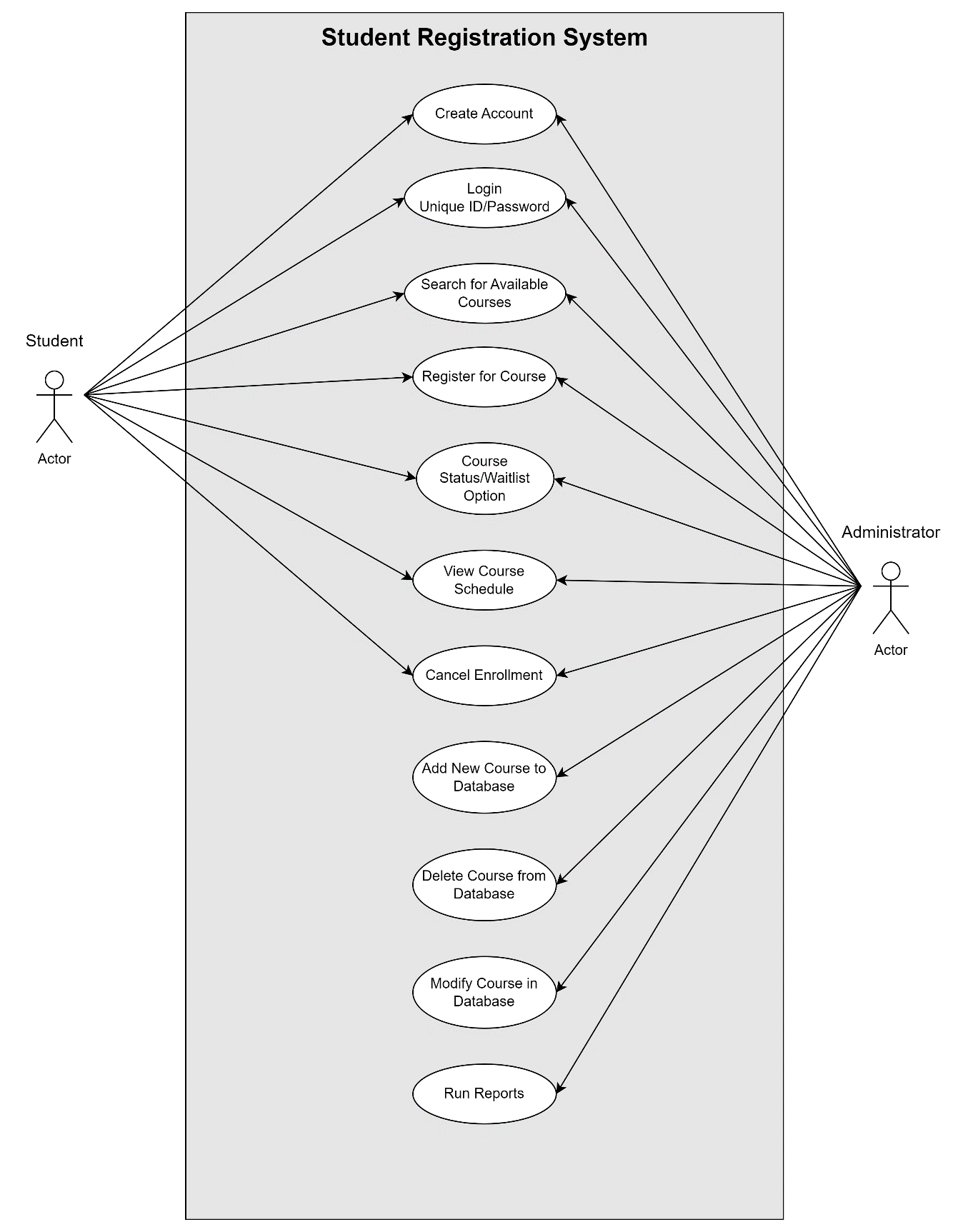
State Diagram



Note. UML State Diagram for Student Registration System

**Figure 5**

Use Case Diagram



Note. UML Use Case Diagram for Student Registration System

**References**

GeeksforGeeks. (2022, June 15). *Acceptance Testing | Software Testing*. <https://www.geeksforgeeks.org/acceptance-testing-software-testing/>

GeeksforGeeks. (2023, November 21). *Integration Testing – Software Engineering*. <https://www.geeksforgeeks.org/software-engineering-integration-testing/>

Hamilton, T. (2023, Decemer 09). *What is Component Testing? Techniques, Example Test Cases*. <https://www.guru99.com/component-testing.html>

javaTpoint. (n.d.). *Integration Testing*. <https://www.javatpoint.com/integration-testing>

Spillner, A., Linz, T., & Schaefer, H. (2014) [Software testing foundations: A study guide for the certified tester exam](https://uagc.instructure.com/courses/126521/modules/items/6439323)(4th ed.). Rocky Nook.

Tsui, F., Karam, O., & Bernal, B. (2018). [Essentials of software engineering](https://uagc.instructure.com/courses/126521/modules/items/6439323)(4th ed.). Jones & Bartlett Learning.

Yasar, K. (2023, March). *System Testing*. <https://www.techtarget.com/searchsoftwarequality/definition/system-testing>