UML Design Modeling

Anders Berg

University of Arizona Global (Formally Ashford University)

CST 499–Capstone for Computer Software Technology

Amjad Alkilani

11/11/2021

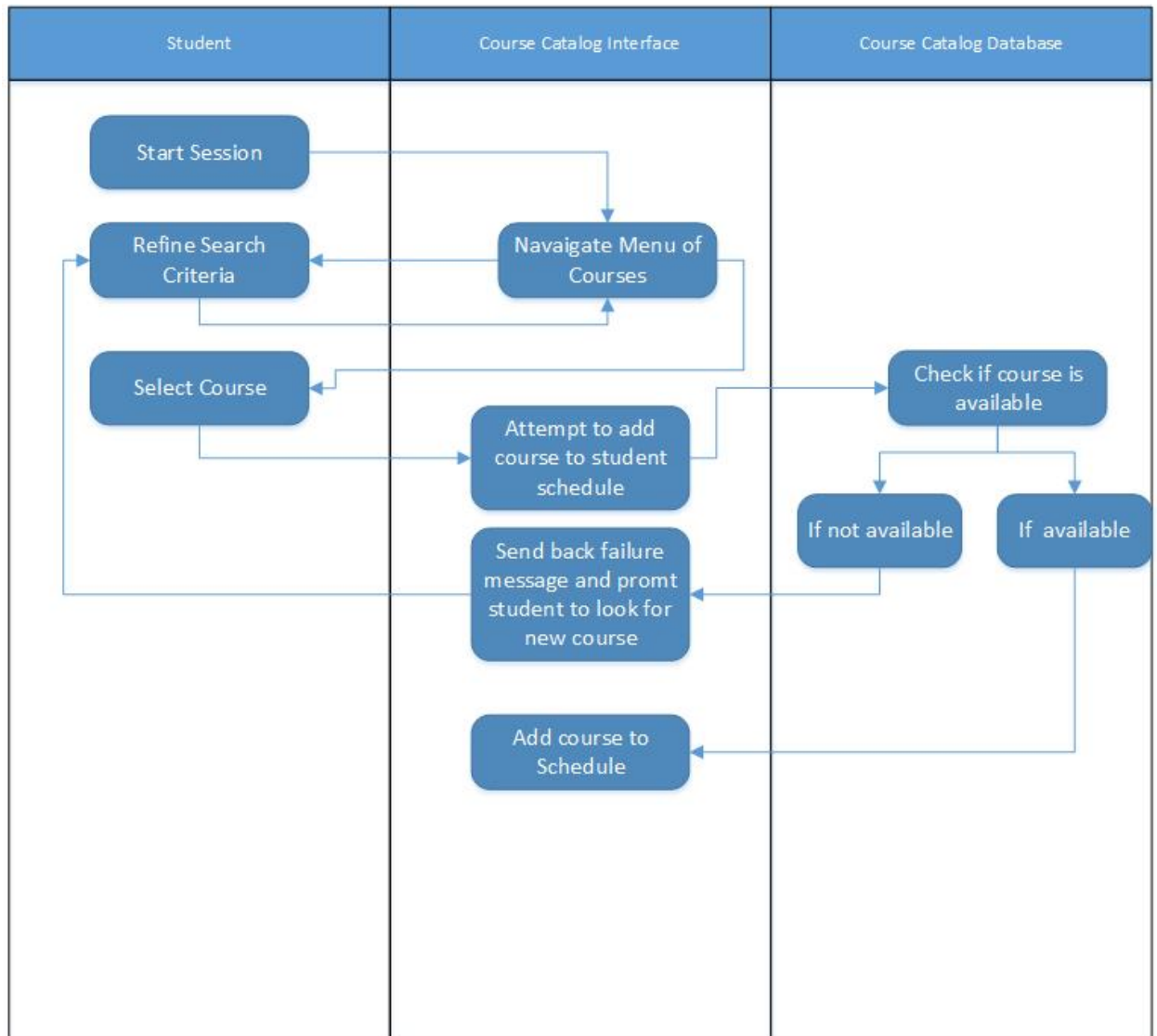# UML Design Modeling

Scope of Development

The scope of this development is to create a system for user and course registration for learning institutions. The system will need to be able to register users that are new and provide the ability for them to create an account. In addition, users need to be assigned a unique id and password. User ID's need to be unique and not be duplicated. User profiles need to include name, phone, and email. Once a user completes a registration, they need to be able to login with their unique id and password that they created. Users will need to be able to register for course in spring/summer/fall semesters. Courses will need to be listed out and available for registration per semester. Course need to have a cap for max registration and if it is full users will be waitlisted. Finally, users need to be able to cancel their registration and automatically allow for waitlisted users to register.

In the use case diagram below, the intended user interactions with the site are illustrated. Based on the scope of development, the users are anticipated to login into the system, search for courses and register for them based on semester. Admins of the system are anticipated to be able to update the course catalog and update student profile settings if needed. This interaction model is important for depicting the anticipated interaction that will occur from the system, including inputs and outputs to internal system connections or external user operations. This type of modeling is important because it helps outline and illustrate the requirements of the system. (Sommerville, 2018)

University Course Registration Site
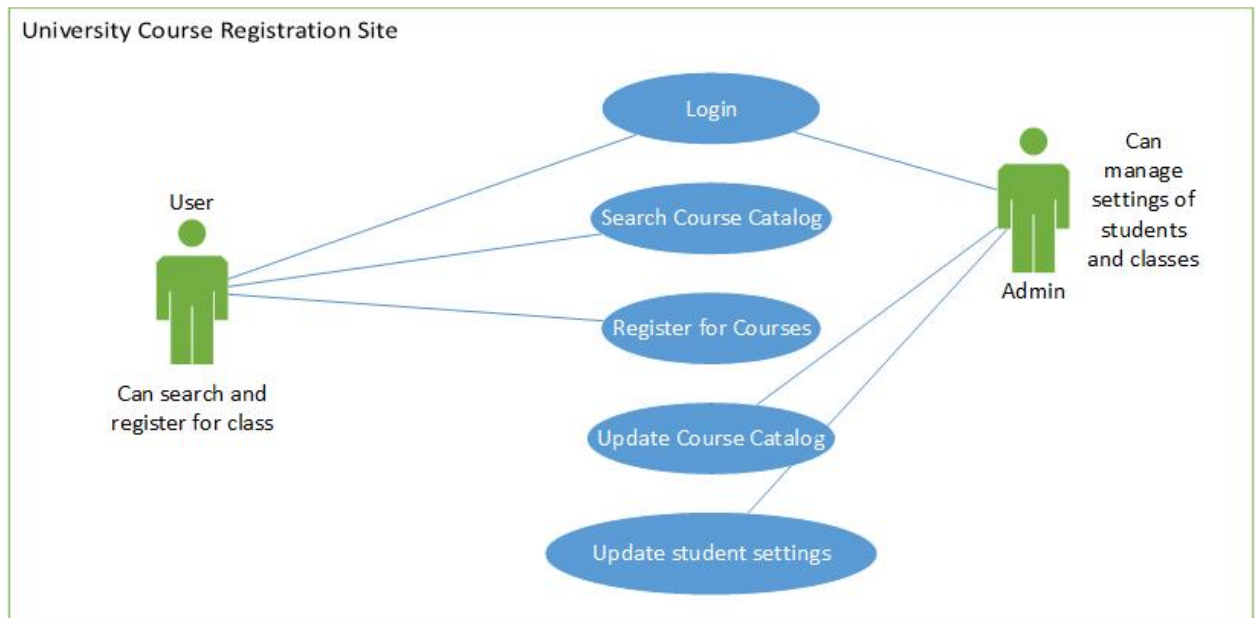
A sequence diagram helps outline how the system will operate and the behavior that the user can expect. Below is a basic diagram that depicts the intended interaction that the user should expect when browsing the inventory on the ecommerce website. The user should be able to search for a course, when they find it, they then should be able to register for it. While the user is doing this they are indirectly, interacting with the database. When the user attempts to add and courses to their schedule, the database records are checked to validate if that course is available for registration. If it is not available a message will be returned to the user, letting them know that the course is not available for registration, and they can elect to be added to the waitlist. (Sommerville, 2018)

Student	Course Catalog Interface	Course Catalog Database

Start Session

Refine Search Criteria

Navaigate Menu of Courses

Select Course

Check if course is available

Attempt to add course to student schedule

If not available   If available

Send back failure message and promt student to look for new course

Add course to Schedule

Structural models show how a system is to be organized and the relationships between the components that make up the system. Structural models are used to depict system process that are static such as architectural design or they can depict dynamic processes such as processes in execution. An example of a structural model would be the class diagram shown below which is the object-oriented construction of the course registration system. (Sommerville, 2018)
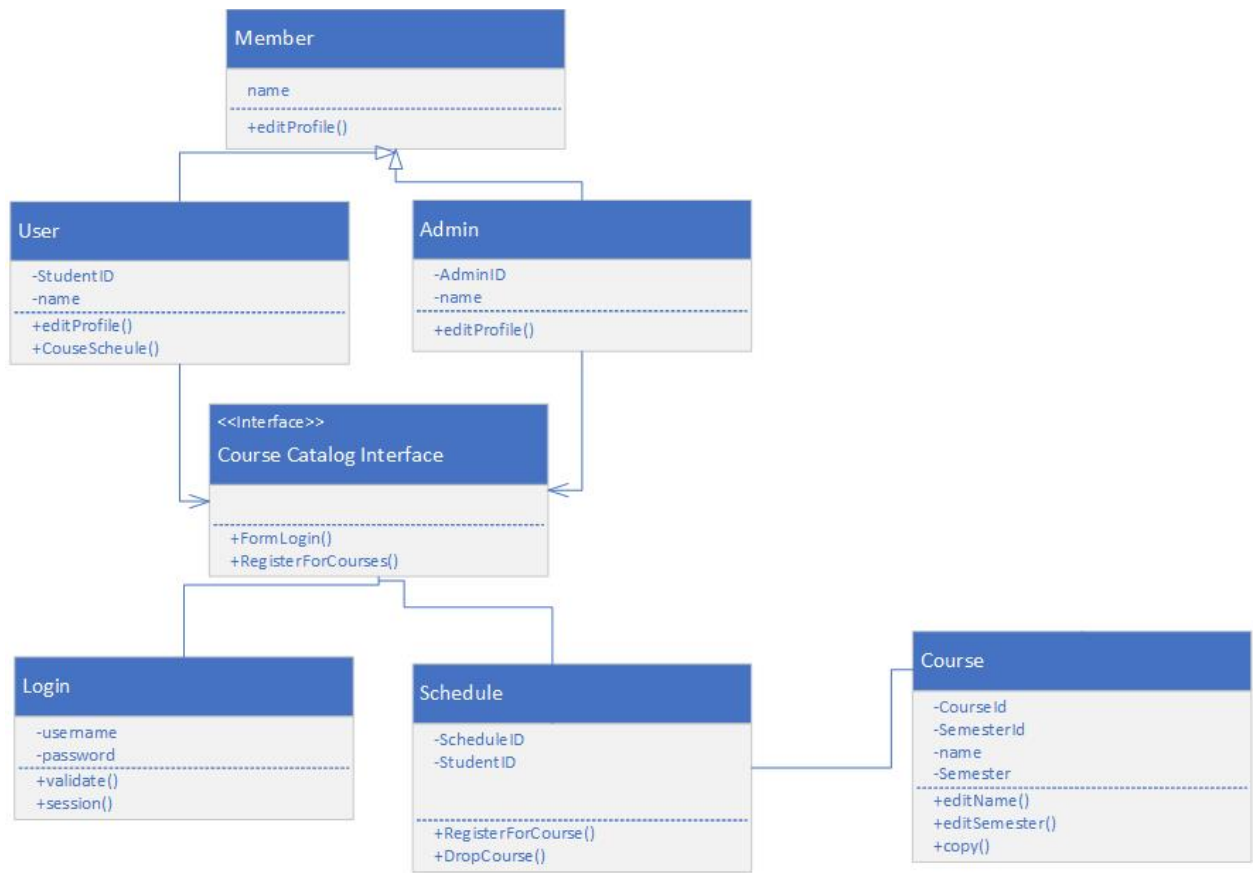
Acceptance Testing

An acceptances test verifies if the developed system meets the requirements and expectations that have been outlined in the scope of development. Acceptance tests also verify that a system will meets requirements and expectations of the end users. In this case, the core requirements of this system are illustrated in the use case diagram above. For example, users are required to be able to login to the system, browse the course catalog and register for course from the catalog. Acceptance testing help verify that the requirements definition process for this software development project have been completed. Requirements definitions is the outline of the needs of the system that are gathered, specified in the detailed scope of work. (Spillner, et al, 2021)

System Testing

A system test will check if the system specifications that are required in the system design have been met. System testing verifies the functional system design process, design process is the decomposition of components from the system design. The decomposition process creates smaller subsystems that are contained within the overall system. Systems test are important to validate that the technical components from the scope of work to ensure the system fulfilling the requirements of the functional system design and overall system. As seen in the sequence diagram, the system is operating with a front-end user interface and a back-end database. These components will need to be validated to ensure that the system is functioning and meeting the overall design requirements. (Spillner, et al, 2021)

Integration Testing

An integration test will check if a group of components are interacting with the system the way that they were designed to in the technical system design. Integration testing verifies that the technical system design process. Technical system design is the development of the building blocks for a system. Integration testing is important and important for validating the systems technical components and have been broken down from the scope of work. (Spillner, et al, 2021)

Component Testing

A component test verifies whether a software component meets it required specification that have been constructed from the scope of work. Component testing verifies each component specification within the developed software. Component testing is important for determining if a task, behavior, structure, interface, or other subsystems is operating to the design specifications.

Component testing is looking at the smaller portions of a system, but these tests are still vitally important. For example, component testing would example all the components created from the class diagram and make sure they are operating to the design specifications listed in the class diagram. (Spillner, et al, 2021)

References

Sommerville, I. (2018). *Software Engineering*. Hallbergmoos/Germany: Pearson.

Spillner, A., & Linz, T. (2021). Software Testing Foundations, 4th Edition: A Study Guide for the Certified Tester Exam (5th ed.)