Aline Koftikian

Project Deliverable 1

13:00 pm – 14:15 pm

Section 7.2 – Artifacts

1. Domain Model

The Domain Model is the model that describes all core concepts and an overview of the system at hand. Its main purpose is to list the objects of interest that form the system, and presents their attributes, constraints, and relationships between them. The model is designed in UML format, with arrows describing relationships from one object to another, and ensuring that there are multiplicity values at both ends of the arrow denoting multiplicity factors of the relationship. The finalized domain model will come with a description that will provide a more detailed explanation of the model as well as each object present in the model.

1. Requirements

Requirements are the tasks that are necessary for the completion of the system. They are the components of the system that must be documented, measured and tested in order to ensure the ideal conclusion to the project in question. The requirements that need to be specified for this system are:

* Functional requirements: The functional requirements are the tasks that define what the system is trying to accomplish. These are the elementary actions that constitute the main goal of the system. They include the user viewing weekly schedule, adding classes, dropping classes, swapping, etc.
* Non-functional requirements: The non-functional requirements are the general properties of the system. They are the constraints and qualities that will make up the system when in use. These include response times, maintenance times, security, etc.

1. Use Cases

Use cases, as the name suggests, is a description of the behavior of a system that explains the process and results obtained through interactions of the user with the system. The functional requirements must be known before writing use cases, because they are the details that compose the main actions and tasks of the system. The use cases are present in both list form and diagram form.

* List Use Cases: The list-based use cases are a full formal document that list all use cases of the system in detail, with each property of the use case explained in full detail. These include the name of the use case, the general description, the actors, preconditions, postconditions, etc.
* Diagram Use Cases: Use case diagrams are used to depict a visual representation of the use cases listed. The actors, which typically includes the user or an external entity, are depicted using stick figures. The use cases themselves are drawn as circles with a name that describes the title of the action that can be initiated by the actors. There are lines connected from users to the use cases called connections, which describe the actor’s interactions with those use cases.

1. Architectural design

The architectural design is a high-level description of the architecture of the system.