SE322: Software Intensive Systems Engineering	**Semester Project**
Semester 2 5785	**Due: 30 June 2025**
Lecturer: Michael J. May	**Kinneret College**

Final Project Submission

This is the final submission of the semester project. You've submitted parts of the project documents before and received intermediate feedback. This is the final iteration, so you'll submit the entire project report and all supporting documents.

1 What to do: Documents

In this iteration you'll finish the project documents: the requirements list, the *Software Requirement Specification (SRS)*, the *Systems Analysis and Design (SAD)* for the project.

1.1 Requirements

A final requirements table for the complete system. Fill in all columns in the template:

1. Header information: Project name, Student names
2. Requirement ID #
3. Requirement text
4. Source (user story, derived)
5. Functional category (FR)
6. Non-Functional category (NFR)
7. System Use Case (SUC) trackback
8. *new* Hardware (HW) trackback - which hardware nodes support the requirement
9. *new* Class trackback - which classes in the class diagram support the requirement

Also fix any issues based on the feedback from the previous design review.

1.2 SRS

Complete all sections of the SRS:

1. Header information (project name, authors, sources)
2. Section 1: System Use Case Diagram for the complete system
3. Section 2: List of actors and stakeholders
4. Section 3: List of System Use Cases for the complete system
5. Section 4:
   (a) System Use Case details (all parts) for all use cases.
      - If there are more than 10 use cases, you may put details for the 10 most important ones.

   (b) Activity diagram for 3 System Use Cases, including swim lanes divisions for actors and system elements
6. Section 5:

(a) Screen shots for all GUIs needed for the use cases.

Fix any parts of the SRS based on requests for changes or updates from the previous design review.

1.3 SAD

Complete all sections of the SAD:

1. Header information (project name, authors, sources)
2. Section 1: System Physical Architecture. All sections.
3. Section 2: System Logical Architecture. All sections.
4. Section 3: Sequence diagrams. For all System Use Cases in the SRS. Create the sequence diagrams using Enterprise Architect.
5. Section 4: Database diagrams. All sections.

Fix any parts of the SAD based on requests for changes or updates from the previous design review.

1.4 *new* SDD

Fill in the SDD document with the following elements:

1. Header information (project name, authors, sources)
2. Section 1: PDOM:
   (a) A *Problem Domain Object Model (PDOM)*
   (b) A PDOM with classes for each fundamental concept.
3. Section 2: Class Diagrams
   (a) Class diagrams for one important software component. Design the classes according to good object oriented software design principles. Include any proxy classes that you will use to model external systems or components.
4. Section 3: Sequence Diagrams
   (a) 1 class level sequence diagram for 1 complete important use case that uses the classes detailed in the previous section.
5. Section 4: Code GitHub Link
   (a) Link to the code repository on GitHub

1.5 *new* Source Code and Tests

Create the following source code elements:

1. Code for all GUIs shown in the SRS. Include code that shows basic user interactions.
2. Static source code for all of the classes included in Section 2 of the SDD.
3. Complete method implementations for all methods shown in the sequence diagram included in Section 3 of the SDD. Connect the component to the GUIs that interact with it.
4. Any external objects or components that are shown in the sequence diagram in Section 3 of the SDD should be replaced with "proxy" methods that call the required method and output a single line of output when run (*e.g.* "Received credit card payment approval", "User clicked on OK button").

In addition, include a DOCX or PDF file with output from running the sequence diagram on the compiled version of the code as a test of its output. The output must include all potential execution paths for the use case, including all branches and exceptions. Take screen shots of the stages of execution of the program. If necessary, you may include output from multiple runs to cover all potential executions. Explain any non-obvious parts of the output with explanatory text.

Place all code in a GitHub repository and put a link to the repository in the appropriate section of the SDD.

2 What to submit by 30 June 2025

Submit the following documents via Moodle:

1. User story. If anything changed since the previous submission, note the changes.
2. Complete and final requirements file (XLSX) based on the temple provided.
3. Complete and final SRS document (DOCX or PDF).
4. Complete and final SAD document (DOCX or PDF).
5. Complete and final SDD document (DOCX or PDF).
6. Output file(s) from code test runs (DOCX or PDF).

3 Feedback and Evaluation

Each team will receive written feedback from the instructor and a final grade for the project.

3.1 Grading

Grading will be assigned as follows:

- User story: 5%
- Requirements: 15%
- SRS: 30%
  - Header: 5%
  - SUC diagram: 15%
  - List of SUC and actors: 10%
  - SUC details: 40%
  - Activity diagrams (3): 15%
  - Screen shots: 15%
  - SRS Total: 100% - weighted to 30 points for the total project
- SAD: 30%
  - Header: 5%
  - Physical architecture:
    - Deployment diagram: 15%
    - Nodes and computers table: 5%

    - ∗ Component breakdown: 10%
  - Logical architecture:
    - ∗ Component diagram: 15%
    - ∗ Detail tables: 5%
    - ∗ Composite diagram: 10%
    - ∗ Database diagrams: 5%
  - Sequence diagrams: 30%
  - SAD Total: 100% - weighted to 30 points for the total project
- SDD: 15%
  - Header: 5%
  - PDOM:
    - ∗ PDOM concepts: 20%
    - ∗ PDOM as classes: 15%
  - Component class diagram:
    - ∗ Class diagram: 30%
  - Class level sequence diagram: 30%
  - SDD Total: 100% - weighted to 15 points for the total project
- Code: 5%