

Assignment 3: System Design

Report Format: table of contents should be as follows (Ensure the page number listed here are correct):

- Section 1: Interaction Diagrams
- Section 2: Class Diagrams and Interface Specifications
 - Class Diagram
 - Data Types and Operation Signatures
 - Traceability Matrix
- Section 3: System Architecture and System Design
 - Architectural Styles
 - Identifying Subsystems
 - Mapping Subsystems to Hardware
 - Persistent Data Storage
 - Network Protocol
 - Global Control Flow
 - Hardware Requirements
- Section 4: Algorithms and Data Structures
 - Algorithms
 - Data Structure
- Section 5: User Interface Design and Implementation
- Section 6: Design of Tests
- Section 7: Project Management and Plan of Work
 - Project Coordination and Progress Report
 - Plan of Work
- References

Section 1:

Interaction Diagrams

Create interaction diagrams for the use cases you elaborated ("fully-dressed") in Report #1. You need to draw **the sequence diagrams** and **other UML interaction diagrams**.

Describe the design principles that you used in the process of assigning responsibilities to objects. You can use annotations or captions to explain.

- Design principles: You use design principles to diagnose problems.
- Use these references to help
 - [SOLID design principles \(Links to an external site.\)](#)
 - [Composition over inheritance \(Links to an external site.\)](#)
 - [Don't Repeat Yourself \(Links to an external site.\)](#)
 - [Inversion of Control \(Links to an external site.\)](#)
 - [You Aren't Gonna Need It \(Links to an external site.\)](#)
 - [Law of Demeter \(Links to an external site.\)](#)
 - [Principle of Least Astonishment \(Links to an external site.\)](#)
 - [Minimum Viable Product \(Links to an external site.\)](#)

Section 2:

Class Diagram and Interface Specification

a. Class Diagram

Display all classes and their relationships. Indicate attributes and operations without details.

b. Data Types and Operation Signatures

Write the class specification in UML notation. For each class, specify data types of attributes and operation signatures. Define each class, operation, and attributes in plain English.

c. Traceability Matrix

Display how your classes created from your domain concepts. Provide explanations for revised names or several classes that evolved from a single concept.

Use traceability matrix like the example shown below mapping the system requirements to use cases. If a use case is derived from a requirement, then you check the corresponding entry in the matrix. The Max PW (priority weight) shows the maximum priority of the checked requirement in the column. The Total PW shows the value summing up the priority weights of the checked requirements in the column. These values are references to schedule the work on implementing the use cases. You need to elaborate, implement, and deliver the use cases with the highest priority.

Req't	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8
REQ1	5	X	X						
REQ2	2		X						
REQ3	5	X						X	
REQ4	4	X						X	
REQ5	2	X	X						
REQ6	1			X	X				X
REQ7	2						X		X
REQ8	1					X			X
REQ9	1					X			X
Max PW		5	2	2	2	1	5	2	1
Total PW		15	3	2	2	3	9	2	3

Section 3:

System Architecture and System Design

a. Architectural Styles

Describe the architectural styles in your design ([Architectural \(Links to an external site.\)](#)).

b. Identifying Subsystems

Draw and describe **UML package diagram** of subsystems of your system.

c. Mapping Subsystems to Hardware

Will your system run on multiple computers? For example, whether you will have a client (web browser) and a server (webserver) subsystems running on different computers?

If it will **not**, you can skip this step.

If it will, show what subsystems run on which machine.

d. Persistent Data Storage

Whether your system require to save data outliving a single execution of the system?

If it does **not**, you can skip this step

If it does, identify the persistent objects and select the storage management strategy, e.g., flat files, relational databases, etc.

e. Network Protocol

If your system will only run on one machine, you can skip this step.

If not, state which communication protocol you will use, e.g., plain Java sockets, Java RMI, Java JDBC, HTTP, etc. Explain why you choose this protocol.

If you use plain sockets, attach the description of the communication protocol (types of message, message format, etc.).

f. Global Control Flow

- i. Execution orderness: Is your system procedure-driven and execute in a linear fashion, where every user every time has to go through the same procedures, or is it an event-driven system that waits in a loop for events, and every user can generate actions in a different order?
- ii. Time dependency: Do you have timers for your system? Is your system of event-response type, with no concern for real-time (periodic), or is it a real-time system? If it is periodic, what are the time constraints for each period?
- iii. Concurrency: Does your system use multiple threads?

If No, you can skip this step.

If Yes, identify the objects that have separate threads of control and describe any synchronization between the threads.

g. Hardware Requirements

- i. What system resources your system depend on? For example, screen display, communication network, or whether it will need any specific sensor or instrument.
- ii. Describe the exact requirements for these resources in order for your system to run. For example, you might need color display, with minimum resolution of 640 × 480 pixels; minimum of 1 Gbyte hard disk space; minimum network bandwidth 28 Kbps.

Section 4:

Algorithms and Data Structures

a. Algorithms

Describe the algorithms that implement mathematical models from your Report #1. If you didn't use any algorithm, you can skip this item.

If Yes, describe your algorithms using activity diagrams.

b. Data Structure

Does your system use any complex data structures, such as arrays, linked lists, hash tables, or trees?

If No, you can skip this step.

If Yes, what criteria you used in deciding what data structure to use, e.g., performance vs. flexibility?

Section 5:

User Interface Design and Implementation

Describe whether and how you revised and implemented the initial screen mock-ups created in report #1. Comment on significant changes in your user interface, that increase or reduce the user effort. Changes of colors or styles are not important which you should not mention in your report.

Section 6:

Design of Tests

- For this report, you just need to design your tests. You will program and run those tests as part of work for your first demo.
- List and describe the test cases that will be programmed and used for unit testing of your system.
- Discuss the test coverage of your tests.
- Describe the Integration Testing strategy and plans on how you will conduct it.
- Describe your plans for testing algorithms, non-functional requirements, or user interface requirements that you stated in report #1.

Section 7:

Project Management and Plan of Work

a. Project Coordination and Progress Report

- i. List use cases that have been implemented.
- ii. What is already functional, what is currently being tackled?
- iii. List and describe other related project management activities.

b. Plan of Work

- i. List the project milestones and dates by which you plan to accomplish them using Gantt charts.

Section 8:

References

The list of references should contain exact references and URLs of any material that is used in the project and does not come from the textbook.

Notes:

1. You should firstly provide generic, language-independent design diagrams in UML
2. You will not submit the source code with this report, but it will be required with your first demo materials.
3. Please comment on your diagrams. Commenting on all design decisions and other things that are not obvious from the diagrams. There is no limit on the number of pages for the report.
4. Drawing interaction diagrams. Generally, you need to draw a single interaction diagram for each system function. But some system functions might be unimportant, thus, you can start drawing one interaction diagram for each use case. If your diagram overflows to the next page, split your interaction diagram and draw one diagram per system function.
5. The tests in section 6 are preliminary designs that may be revised when you will do actual testing for your first demo. You do not need to write any program code for unit testing for this report. Test code writing and running the tests is expected for the first demo. For report #2, you just need to describe which classes/ functions should be tested and how you would go about doing it.

Grading

Item	Sub-item	Points Possible
Interaction Diagrams	UML diagrams	15
	Prose description of diagram	15
Classes + Interface Specification	Class diagram & Description	10
	Signatures	5
System Architecture & Design	Styles	5
	Package diagrams	8
	Map hardware	3
	Database	4
Algorithms & Data Structure		4

Item	Sub-item	Points Possible
User Interface	Appearance	6
	Prose description	5
Testing design		12
Project Management	Document consistency	2
	Project progress	4
	Plan of work	2
Ref's		(-5)
TOTAL		100