SOFE3650U Software Design and Architecture
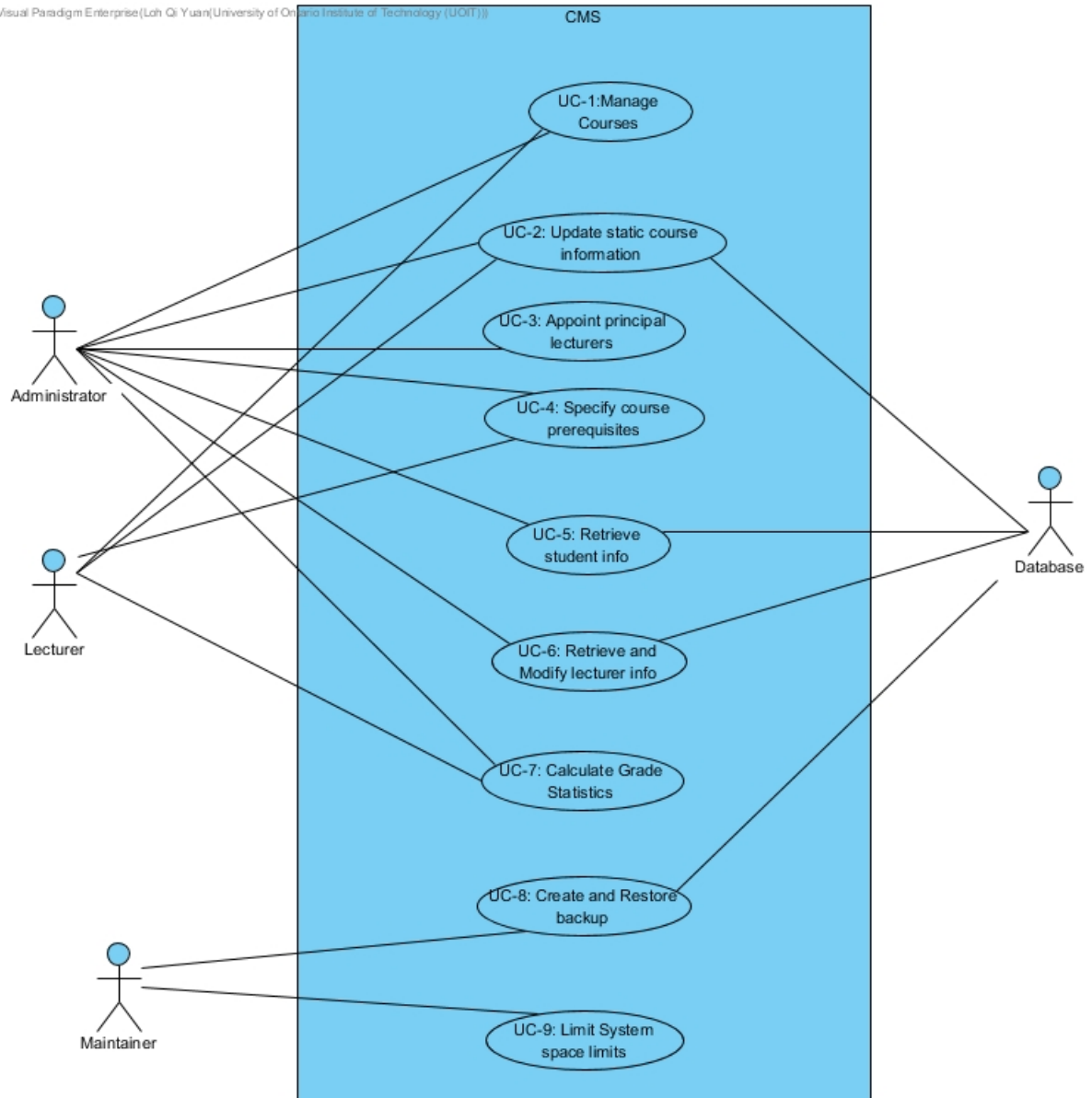
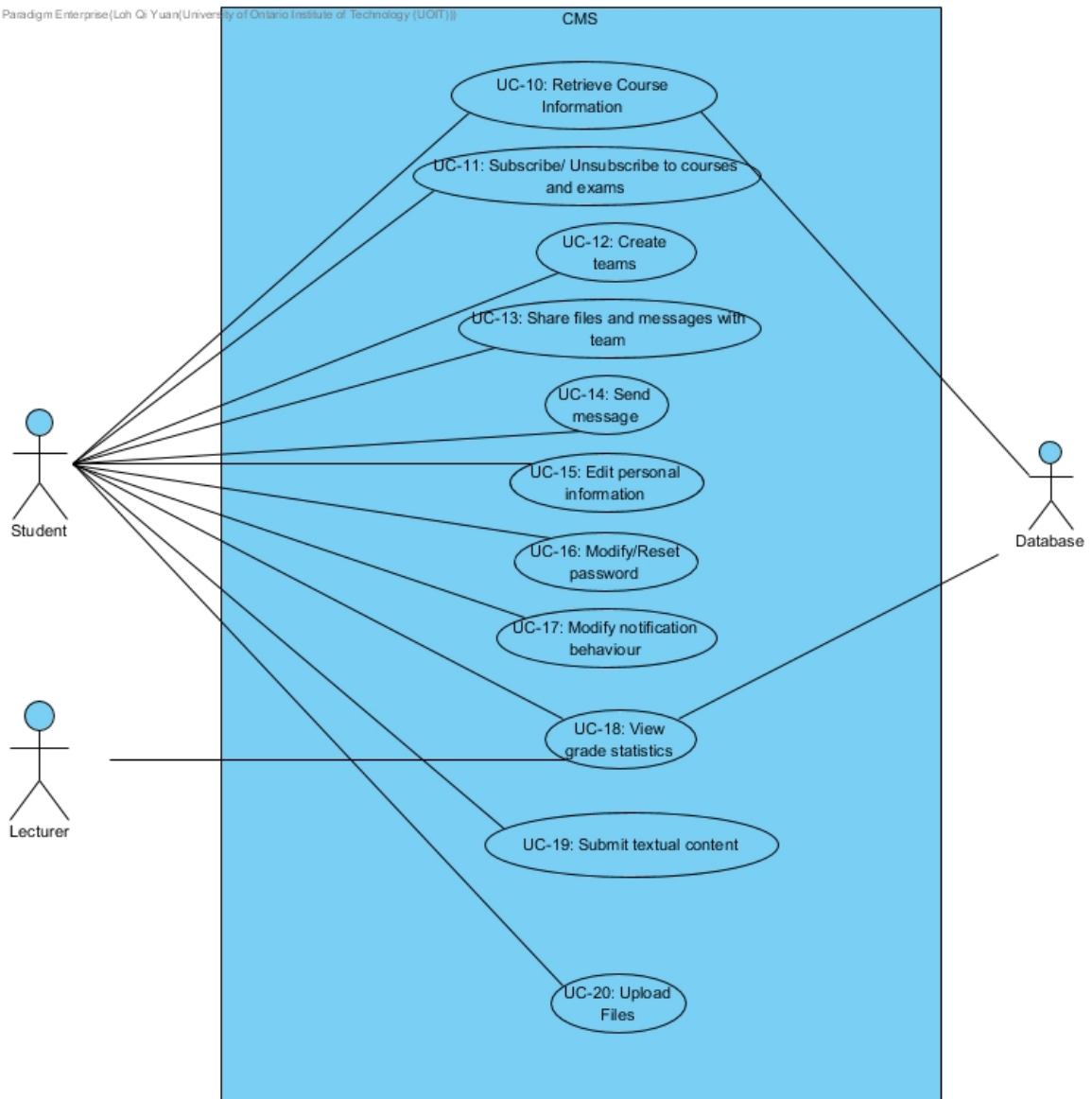**Assignment 2**

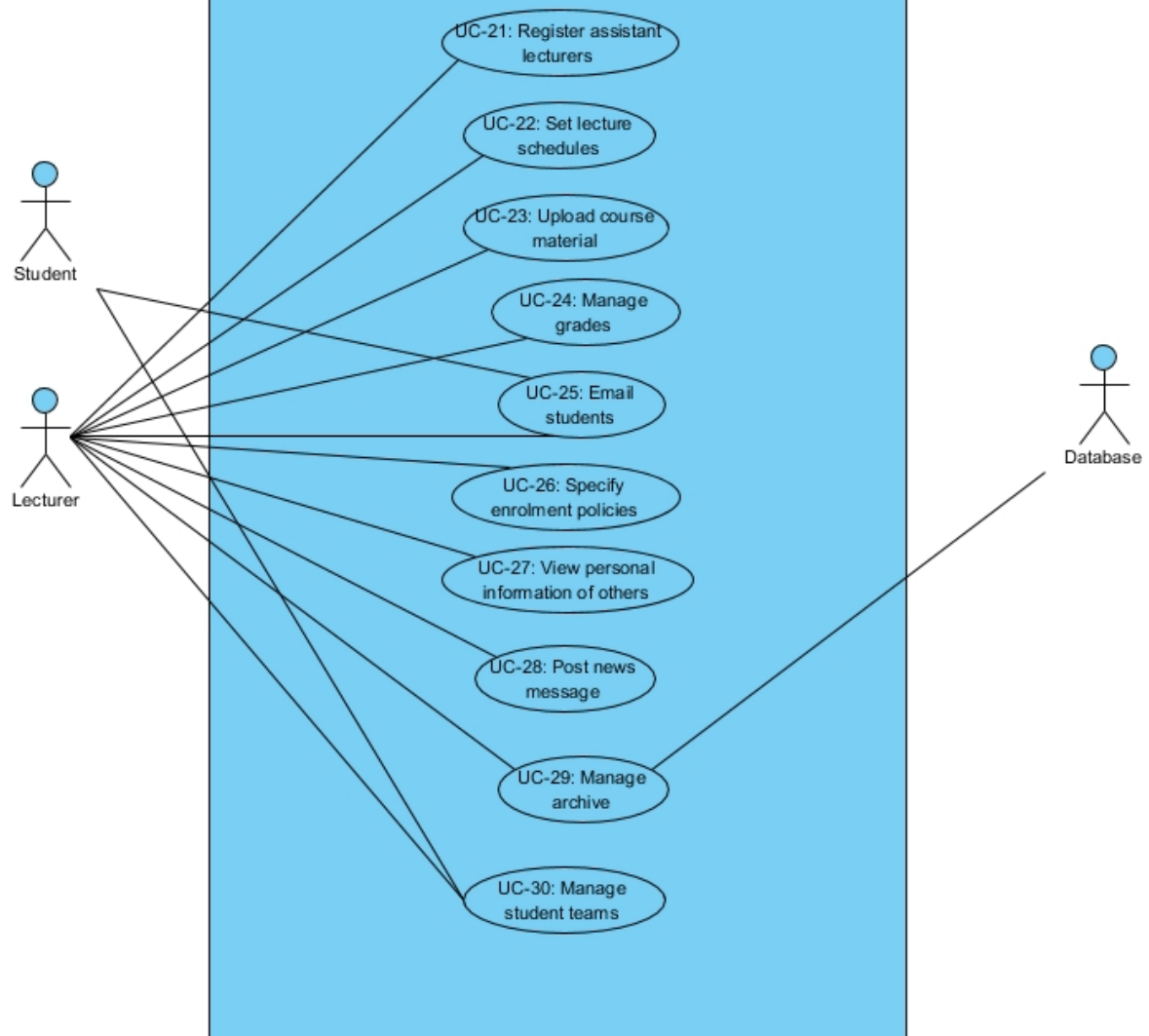**HEW, GUO WEI 100720466**
LOH, QI YUAN 100720442

Q1a)

CMS

Administrator

Lecturer

Maintainer

Database

UC-1:Manage Courses

UC-2: Update static course information

UC-3: Appoint principal lecturers

UC-4: Specify course prerequisites

UC-5: Retrieve student info

UC-6: Retrieve and Modify lecturer info

UC-7: Calculate Grade Statistics

UC-8: Create and Restore backup

UC-9: Limit System space limits

CMS

UC-10: Retrieve Course Information

UC-11: Subscribe/ Unsubscribe to courses and exams

UC-12: Create teams

UC-13: Share files and messages with team

UC-14: Send message

UC-15: Edit personal information

UC-16: Modify/Reset password

UC-17: Modify notification behaviour

UC-18: View grade statistics

UC-19: Submit textual content

UC-20: Upload Files

Student

Lecturer

Database

CMS

Student

Lecturer

Database

UC-21: Register assistant lecturers

UC-22: Set lecture schedules

UC-23: Upload course material

UC-24: Manage grades

UC-25: Email students

UC-26: Specify enrolment policies

UC-27: View personal information of others

UC-28: Post news message

UC-29: Manage archive

UC-30: Manage student teams

Q1b) Specify the quality attribute scenarios associated to the Use Cases.

ID	Quality Attribute	Scenario	Associated Use Case
QA1	Privacy	Prevent students from viewing grades of others. Provide a user-customizable visibility policy for personal information	UC15 ,16, 27
QA2	Availability	No unexpected downtime/ expected downtime to be announced at 48 hours in advance. Downtime only during low-intensity hours, at most 4hours/month	ALL
QA3	User friendliness	Bilingual support. Max 3 click to reach content. Single login to access all content. Intuitive, descriptive, consistent UI.	ALL
QA4	Accessibility	Accessible by blind users.	ALL
QA5	Security	Allow only students to change study information of others. Allow lecturers to view dynamic course information by other lecturers, manage dynamic content visibility. Allow lecturers to view grade of all students. Students can only view own grades.	UC1, 5, 18, 24
QA6	Interoperability	Export to commonly used calendar formats. Import BOZ roster info into course roster. System to be	UC1,2, 11

		interoperable with secondary uni systems	
QA7	Extensibility/Evolvability	Easily extensible, evolvable. Allow admin to make exceptions to student enrolment	UC5,6, 8, 9
QA8	Testability	Easily testable	UC8, 9
QA9	Scalability	scalable	UC8, 9
QA10	Maintainability	Easily maintainable	UC 8,9

Q1c) Specify the constraints imposed on the CMS.

Constraint 1: System must be accessed through a web browser (Chrome, Firefox, IE, Safari) on different platforms: Windows, OSX and Linux

Constraint 2: A minimum of 200 simultaneous users must be supported.

Constraint 3: System must support different forms of accessibility options such as Text-to-speech.

Constraint 4: All course information since the start must be stored.

Constraint 5: System must be backed up every 24hours

Q2a) In what step of the ADD would one leverage the ATAM?

Step 7: Where one performs analysis on the current architecture design.

Q2b) What is the main assumption that ATAM makes concerning quality attributes and the software architecture?

We assume that quality attributes are measurable and observable.

Q2c) What are the 4 main sections of ATAM?

Presentation, Investigation and analysis, testing, reporting

Q2d) Within those 4 main sections what are the ATAM steps?

1. Presentation: Present the ATAM, present business drivers, present architecture.

2. Investigation and analysis: Identify architectural approaches, generate quality attribute utility tree, analyse approaches.

3. Testing: brainstorm and prioritise scenarios, analyse architectural approaches

4. Reporting: resent results

e) What are the 2 variants of ATAM and which one was adopted by the game architectural design paper?

They are architecture-centric, and stakeholder-centric.
The architecture-centric face was adopted in the game design paper.

f) What are the 2 business goals that are being addressed by the game architecture systems?

The first goal breaks the conventional practises used in developing games.
The second goal aims to adapt architecture into different types of games.

g) List and briefly describe the 3 principal quality attributes for game architecture system.

The three quality attributes for the game architecture system are portability, modifiability and performance. Portability is measured by the ability to run the same game on multiple different game engines without modifying the game. Modifiability is the ability to make changes to the game easily and is measured by the amount of changes to different components within the architecture. Performance is the ability for the system to respond to

stimuli within an acceptable timeframe and is measured in terms of time, with an acceptable time being less than a second

h) What test is proposed for the portability scenario and what are the 3 dependencies that must be broken to allow portability?

The test proposed is to run the same game on multiple game engines without modifying the game itself. The portability attribute succeeds if the same game can be serviced to multiple game engines without any modification to the game.

• **Dependency 1:** The direct link between the game space's game state and the game engine's game state should be broken.
• **Dependency 2:** The direct link between the game space's game model and the game engine's game model must be broken.
• **Dependency 3:** The game behaviour should not be formatted in the game engine's proprietary format.

i) What 5 architectural decisions are proposed to resolve the portability quality and what is the reason to choose them?

<u>Architectural decisions</u>	<u>Reasons</u>
Model-View-Controller (MVC)	Multiple views can exist for the same model and any modification carried out by one of the views is visible to other views.
Asynchronous Messaging	The messaging mechanism used is asynchronous which aims at limiting the delay that is caused by the synchronous mechanism.
Scripting	Allows for specifying and manipulating the game without hardcoding, making it easier to modify.
Ontologies	Avoids the inflexibility associated with building hard-coded classes. It also provides a common language for describing a domain and machine interpreted language. Additionally, it is easy to understand and maintain
API	API and mid-game scripting are used to aid the game model dependency and set the game behaviour to break the last unwanted dependency

j) How does the ATAM help resolve the ambiguity of stakeholder requirements like "the system should be modifiable"

The use of scenarios provide a platform to concretize these ambiguous statements. They allow stakeholders to come up with concrete requirements as compared to vague statements as these scenarios are stimulated real world events. Also, exploratory scenarios may expose the boundary conditions for the requirements

k) What performance scenario was eventually agreed to for the utility tree?

1) Display Rate (FPS) : The system should aim to achieve 20 FPS
2) Responsiveness : When the user triggers an action in the game the response should be achieved in less than 1 second.
3) Throughput : measure the highest allowed throughput before performance starts to be affected

l) What 2 risks are associated with the MVC architectural style?
- the tight coupling between the controller and the model which is a known liability of using this pattern
- the difficulty to maintain the data integrity between the two states.

m) What is a non-risk? Why is N1 a non-risk rather than a risk?

Non-risks are good decisions that rely on assumptions that are frequently implicit in the architecture. N1 is a non-risk as it is a good decision made to prevent the tight-coupling between the view and controller.

n) The portability scenario analysis is a key table in the ATAM approach. Summarize the intention of the table and what information is captured.

The table shows the outcome of analysing the portability scenario for which a number of key architectural decisions have been identified. For each Architecture Decision (AD) there will be following information on them such as Sensitivity, Tradeoff, Risk and Nonrisk.

o) What is the difference between the scenarios generated in step 7 of the ATAM and those used for the utility tree?

The first looks similar to generating scenarios for the utility tree in phase 2, however the aim here is different. There the stakeholders were asked to generate scenarios based on given quality attributes, here they are asked to ignore that and give general scenarios. The goal of this step is to widen the spectrum from which scenarios can be elicited. This step has been described as a bottom-up approach and the utility tree step as a top-down approach.

p) In the reporting stage of ATAM what was the conclusion about using the MVC pattern.

The separation decision made in MVC affects performance. Due to MVC, the data integrity across different game states is at risk.