

# CS3213 Project – Week 2

Requirement Elicitation | 19-01-2022

- ☐ Customer Introduction
- ☐ Customer Statement
- ☐ Open Floor for Questions
- ☐ Assignment 2

# Group & Project Selection Deadlines

**Deadline** for group registration<sup>1</sup>:  
**Friday, Jan 21, 10 am**

**Deadline** for project selection<sup>1</sup>:  
**Friday, Jan 28, 10 am**

---

<sup>1</sup> [https://docs.google.com/spreadsheets/d/15sk6WnvQHTjClhMi\\_TUuyDkylow6lOmMHVhD5n3Qu-l/edit?usp=sharing](https://docs.google.com/spreadsheets/d/15sk6WnvQHTjClhMi_TUuyDkylow6lOmMHVhD5n3Qu-l/edit?usp=sharing)

# Lab Assignments and Timeline

Week	Lab Session	Assignment	Assignment Due
1	Introduction	A1 – Requirements Analysis & Elicitation	Week 2
2	Requirements Elicitation with Customer	A2 – Requirements Modeling	Week 3
3	Requirements Modeling	A3 – Behavioral Modeling & Architectural Drivers	Week 4
4	<i>ITS Architecture &amp; Projects (CNY)</i>	A4 – Module Design / Strategy Plan	Week 5
5	Module Structure and Behavior Design	A5 – Project Planning	Week 6
6	Implementation	A6 – Intermediate Deliverable	<b>Week 7</b>
<i>R</i>	<i>Recess Week</i>	-	-
7	Recap – Requirements & Design	-	-
8	Static Code Analysis	-	-
9	Unit Testing	A7 – Test Case Design	Week 10
10	Fault Localization and Debugging	A8 – Presentation + Final Code	<b>Week 12</b>
11	Integration testing	-	-
12	Lab Closing Session	A9 – Final Report	<b>Week RD</b>
13	Student Presentations	-	-
<i>RD</i>	<i>Reading Week</i>	-	-
<i>E</i>	<i>Examination</i>	-	-

# Protocol for today's session

- ❑ Introduction of each **available stakeholder**
- ❑ Brief statement by each available stakeholder (→ motivation)
- ❑ Start of the actual requirement **elicitation** with your questions!
  - ❑ Yannic will act as **moderator**
  - ❑ Raise your hand, and speak up if it is your turn
  - ❑ ...
- ❑ Closing statement by stakeholders (→ most important aspect)
- ❑ Outlook to Assignment 2

# Present Stakeholders

- ❑ **Wei Tsang Ooi** (Associate Professor):  
CS1010 - Programming Methodology
- ❑ **Alan, Cheng Ho-lun** (Senior Lecturer):  
CS 1102S - Data Structures and Algorithms



# Question Session

# Questions & Requirements

The next slides show more questions along with some of the responses by our stakeholders.

To get a complete picture, please refer to the recording of our [requirement elicitation session](#).

# Current Pain Point for Tutors

Grading, particularly finding bugs in the student's program, takes time.



# Current Pain Point for Students

- ❑ Could be frustrating for beginners to debug their code.
- ❑ Rely on messaging with tutors/friends to help is slow.
- ❑ Might not be “helped” in the correct way that helps them think and learn.

# 1. Goal and Tasks (1/5)

- ❑ What is the general purpose of the intelligent tutoring system?

→ *Given a set of a test inputs and test outputs, and a program that is expected to produce the output from the corresponding input, the system should help a person identify bugs in the program. Rely on messaging with tutors/friends to help is slow.*

→ *Grading of Assignments and PE.*

- ❑ Who are the expected users?

→ *Students learning to program; tutors/TAs grading students' program. Grading of Assignments and PE.*

# 1. Goal and Tasks (2/5)

- ❑ How does the current tutoring process look like? Please describe the typical workflow

→ *Students are given a programming problem to solve. Tutors may give hints at the beginning on the approach to solve the problem. Students then solve on their own. Students may contact tutors for help if they are stuck or need guidance to get started. After submission, tutors will test the student programs, and manually identify the bugs. Test data are given to students. Additional test data may be introduced during grading.*

# 1. Goal and Tasks (3/5)

- ❑ What should be the main tasks of the envisioned intelligent tutoring system?
  - *1. Help students learn basic programming constructs (first 2-3 weeks).*
  - 2. Save tutors time when grading.*
  - *Primary Grading. We may have appeals and secondary grading after it.*
  
- ❑ Are you using/heard of any existing automated software system for tutoring?
  - *We are not using any intelligent tutoring system.*

# 1. Goal and Tasks (4/5)

- ☐ What is the goal of the software system?
- ☐ How does the current tutoring process look like? Please describe the typical workflow(s).
- ☐ Does the current process use any sort of automation?
- ☐ What do you like about the current tutoring process?
- ☐ What do you dislike about the current tutoring process?
- ☐ What is the general purpose of the software system?
- ☐ What should be the main tasks of the envisioned software system?
- ☐ What manual tasks should it support?
- ☐ Who are the expected users?

# 1. Goal and Tasks (5/5)

- ☐ Are there different user groups that need exposure to different parts of the software system?
- ☐ What are the typical characteristics of the users? Can you give examples?
- ☐ What is the scope of the software system?
- ☐ Are there existing software systems that you like, or would like to be included? Please provide references.
- ☐ What do you like about the existing software systems?
- ☐ What do you dislike about the existing software systems?
- ☐ Which limitations are expected from the envisioned software system?

## 2. Functional Requirements

- ❑ What are the main functionalities that you expect the intelligent tutoring system to have?
  - *Automated Error Localization*
  - *Automated Plagiarism Detection*
  - *Automated Error Fixing*
  - *Automated Programming Functionality Evaluation*
- ❑ What is the most urgent functionality you need?
- ❑ In which way should these functionalities be offered to the user?
- ❑ Do you have any particular requirements for the user interface?
- ❑ Do you have any accessibility requirements?

### 3. Constraints

- ❑ Are there any design constraints with regard to the software architecture?

→ *It should be command line based following Unix software / tools philosophy.*

- ❑ Are there any functionalities of the overall process, which should explicitly \*not\* be implemented by the software system?
- ❑ Is there any dependency regarding existing software systems? Please provide references.
- ❑ Are there any external dependencies that need to be taken care of? Please provide references.



## 4. Non-functional Requirements (1/2)

- ❑ Do you have any safety or security requirements?

→ *We should be able to limit the number of times a student can use the system for each programming problem.*

- ❑ Do you have any requirements on the maintainability of the intellignet tutoring system?

→ *We should be able to add new features easily; we should be able to update/upgrade the core "intelligence" easily.*

## 4. Non-functional Requirements (2/2)

- ☐ What are the quality requirements for the software system?
- ☐ Do you have any performance requirements?
- ☐ Do you have any particular requirement for the user interface?
- ☐ Where should the system be deployed? Is there any existing technical infrastructure?
- ☐ How many users are expected in one week / one semester?
- ☐ Do you have any safety or security requirements?
- ☐ Do you have any requirements on the maintainability of the software system?
- ☐ Do you have any specific requirements on the usability of the system?

## 5. Product Increment (1/2)

❑ Can you describe the functionality of a minimal viable product (MVP)?

→ *it should be a command line program running on Linux platform*

→ *it should take in as arguments: the name of the program to check, an input file, and an output file.*

→ *it should output either: (i) nothing if the output matches the input; or (ii) the (wrong) output of the program; the expected output; and the line or lines where the identified bug(s) occurs; or (iii) goes into infinite loop if the student's program goes into infinite loop.*

→ *it should mark assignments.*

## 5. Product Increment (2/2)

- ❑ To what extent do you expect any further extension ability?

→ *Beyond MVP, we should provide options to (a) limit the number of times a user run the program within a certain period; (b) configure the verbosity of the output (identify the bugs or just the lines containing the bugs or suggest how to fix the bugs)*

→ *Easy marking scheme input by users*

- ❑ Which features are less important and can be provided with some later product increment?
- ❑ Which functionality may require extension in the future?

## 6. Verification

- ❑ How much would you like to be involved in the testing process? For example, can you test a beta version?

→ *I can "try" it out and provide feedback during the development, but probably can't do extensive QA. I can deploy a beta version if I am still teaching CS1010 next year.*

→ *Discussion of functionality?*

- ❑ Do you have any requirements for the test process of the software system?
- ❑ Do you have available test data that can be used to validate the software system?

## 7. Others

- ❑ Do you have any other comments, any more requirements or concerns regarding the software system?

→ *Fixing indentations of Python.*

- ❑ Do you have any other supporting information for the design and implementation of the envisioned software system?

# Assignment 2: Requirements Modelling



## Assignment 02: Requirements Modeling

CS3213 Foundations of Software Engineering (AY21/22 Sem2)

Submission Deadline: **Week 2**

Discussion: Week 3

- You need to solve and submit this sheet in the assigned/formed groups via LumiNUS.
- Please appropriate tools to create your solutions (e.g., LibreOffice/Word/LaTeX for textual submissions, or `draw.io` for graphical solutions). Handwritten solutions are accepted only in exceptional cases and if they are very legible.
- Please create a **PDF document** from the solution including a **title sheet** with the exercise sheet number, group number and the names/matriculation numbers of the students in the group.
- Please use this scheme as the file name for the PDF document: `assignment.XX_group.YY.pdf`, where `XX` is the exercise number and `YY` is the group number.
- Please submit this PDF document via LumiNUS. In case of any discrepancies regarding the submission date, the date given in LumiNUS will count. If a submission in LumiNUS is not possible for *good* reasons, send your solution by e-mail *timely* to `yannic.noller@comp.nus.edu.sg`. The date of receipt of the email counts in this case.
- There are **60 points** to be scored for this assignment sheet. The worst score for any assignment sheet is 0 points.
- Note that achieving a total of **at least 60% of the points in each assignment sheet is required to pass the course.**

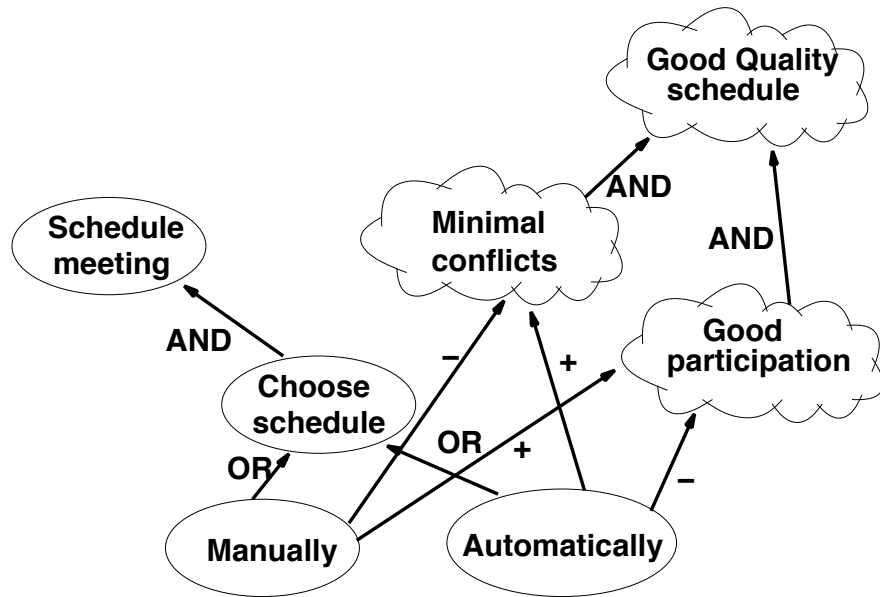
### Overview

The objective of this assignment is to exercise the documentation and modeling of requirements. In the last week, you have performed a requirements elicitation session with *real* stakeholders for the planned system. In this session, you have interviewed the stakeholders, which allows *them* to have a clearer view of what they need and to narrow the scope of the project. However, they will need documentation to back up what is being agreed upon. Now your task is to organize/structure your notes and collected ideas, and document them as *models* and apply appropriate *abstractions*. Your initial findings need to be documented using two Software Engineering standard models for requirement elicitation: one goal-

## Objectives:

- ☐ exercise the **documentation** and **modeling** of requirements
- ☐ get together as **submission group**

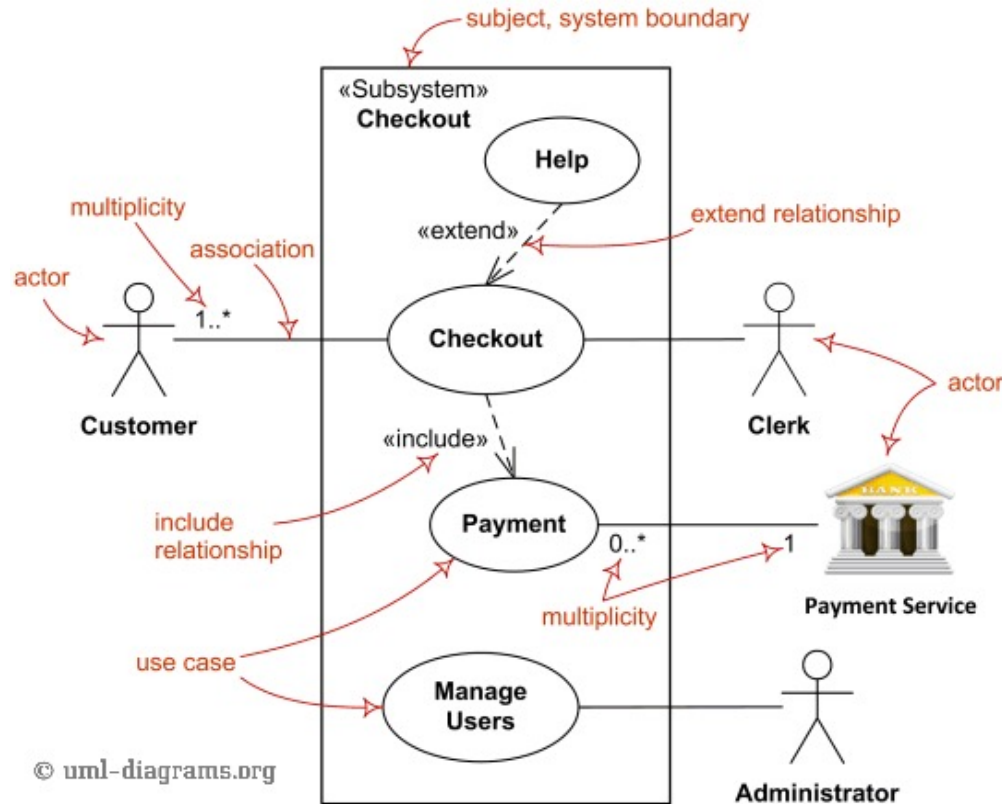
# Task 1: GORE modeling



- ❑ **make yourself familiar** with goal modeling (a simple introduction was provided in Lab 1)
- ❑ **decompose** the goals into sub-goals and model their dependencies
- ❑ think about **hard-goals** and **soft-goals**



# Task 2: UML Use Case Model



- ❑ create a **use case diagram** to summarize the functional requirements
- ❑ create a **textual description** for two use cases

# Conclusion

- ❑ **Thank you** customers for attending our session!
- ❑ Next step: **documentation** of the requirements

## Next Week: **Requirements Modeling**

- Discussion of Analysis, Elicitation, Models
- Architectural Drivers