WM393 Software Development Life Cycle

Assignment 1

1. Introduction

The purpose of this assignment is to assess the students with knowledge of software engineering methodology and the skills to apply it. This is a team-based assignment. You belong to a team into 5 or 6 people randomly selected by a tutor to work on the assignment. Each team needs to submit one project report before the specified deadline. The project report mark will equally apply to each member in the team, meaning that each member in the team should share efforts and responsibility for the assignment. Remember that the project report will require not only contributions from a team but also from each team member as a collection of individual member works. Thus, each member in the team should have responsibility for own contribution, otherwise it may cause burden to other members in the team and even the team mark.

2. Scenario

The COVID-19 has results in schools and universities shut down across national level and even the world. As a result, education has dramatically changed with the distinctive rise of e-learning, whereby online teaching is being undertaken remotely through digital platforms such as Teams, Zoom, Meet, etc. With such a shift from offline teaching to online teaching, many people believe that the adaption of online teaching will persist even after COVID-19 ends. However, these digital platforms are mainly focusing on video conferencing and business communication. Their additional features provided by the platforms such as sharing or noticing information continuously may be limited for online teaching. For this reason, it is necessary to seek other online services from another digital platforms and use several digital platforms together for online teaching unfortunately.

Suppose that, despite Moodle learning management system, WMG is planning to develop a tailor-made system called **WMG Teaching Support System (WMGTSS)** to provide useful functions to effectively support online teaching for each teaching module. The basic functions for the WMGTSS are designed but the full requirement specification has not been clearly defined yet. Your team is assigned to this development and is responsible for creating this system to serve the purpose of the WMGTSS. There is the description for each basic function designed in the WMGTSS below:

- **Feedback board**

  The feedback board is a board where students can make any feedback during a lecture. There are conditions for this board. A tutor only can create one or more feedback boards in accordance with request. For example, a tutor likes to have feedback on a daily basis or on a teaching block basis (one board per 5 days). A student can leave any feedback as many as possible. Comments on a feedback is optional.

- **Notice board**

  The notice board is a board where a tutor can make any notice to students. There are conditions for this board. A tutor only can create one or more notice boards in accordance with request. A tutor only can make a notice and also assign a priority to a notice (such as a pin functionality). If a notice has a high priority, it should appear on the top of the notice list. Comments on a notice is optional.

- **Q&A board**

  The Q&A board is a board where a student can ask a question to a tutor and a tutor can give an answer for the question. Sometimes, a tutor can make a Q&A on behalf of students. A tutor only can create one more Q&A boards in accordance with request. There are conditions for this board. A tutor or student only can create a question and a tutor only can leave an answer for the question. Comments on a Q&A is optional.

- **Discussion board**

  The discussion board is a board where a tutor can make a discussion topic for students. There are conditions for this board. A tutor only can create one discussion board. A tutor only can create a discussion topic on the board. A tutor or student can leave an opinion on each topic. There is no comments option.

- **Resource board**

  The resource board is a board where a tutor can upload any useful information for students. There are conditions for this board. A tutor only can create one or more resource board in accordance with request. A tutor only can upload any resource such as image, video, text, link, etc. Depending on resources, some are downloadable via a link and some have to be displayed such as image and video. Comments on a resource is optional.

- **Lecture board**

  The lecture board is a board where a tutor can upload the slides of a lecture note and a student can navigate the slides. There are conditions for this board. A tutor only can create one lecture board. A lecture only can create a lecture on a daily basis such as Day 1, Day 2, etc. and upload lecture slide images on each day. A student can navigate the lecture slides using navigation buttons. Uploading and downloading the original lecture slide such as PPT is optional.

- **Data file board**

  The data file board is a board where a tutor can upload file(s) and a student can download some or all. There are conditions for this board. A tutor only can create one or more data file board in accordance with request. A tutor can upload one or many data files with a description on each item. A student can download the data file via a link. Showing download counter on each file is optional.

- **Quiz board**

  The quiz board is a board where a tutor can make a quiz to check students' progress and a student can answer the quiz. There are conditions for this board. A tutor only can create one quiz board. A tutor can make a quiz with a number of questions. A student only can make the answers for the questions. A lecture can see the overall scores with good representation such as a graph, but a student can see own score. Deciding multiple attempts is optional.

- **Calendar board**

  The calendar board is a board where a tutor can inform any schedule or notice to students. There are conditions for this board. A tutor only can create one calendar board. A tutor only can create any event information on a day or days. A student can see the event information on monthly view or weekly view.

3. Task

First, your team needs to capture the functional requirements and non-functional requirements based on the basic functions described above. However, due to the limited number of team members allocated in a team, each team member must select only one of the functions exclusively and should contribute the selected function to the team report, particularly in the Functional Requirement and Interface Requirement section. The team report should also include the management of these functions such as how to add, update and delete a function in the WMGTSS, and the management of users including tutors and students such as how to give a privilege to users.

Second, your team also needs to design the system architecture of the WMGTSS which addresses the structure, behaviour, and more views of the system. However, due to the limited time, your team only considers logical and structural view of the system such as using simple block diagram and component diagram in UML when designing the system architecture.

Finally, your team needs to produce a team project report with the requirements and architecture design. Also, the team report should include 2 or 3 meeting summary notes in order to prove a team holding efficient agile meetings during the team project. In addition, it is noticed that the assignment 2 will be an individual task based on own selected function in the assignment 1. Thus, when you select a function for your team report, please keep in mind of the assignment 2. The detailed instructions of this assignment as a team and also team member are:

[TEAM]

- A team should produce a team project report including the requirements and system design.
- A team should include the meeting summary notes for the selected team meetings (2 or 3 recommended) but it is not applicable to the total number of word counts.
- A team should adhere to the report format below:
    - Introduction
    - Overall Description
    - Functional Requirement
    - Interface Requirement
    - Non-Functional Requirement
    - System Architecture
    - Appendixes (if necessary)
    - Meeting Summary Notes (for 2 – 3 selected team meetings)
- A team should gather each section information by team discussion.
- A team should discuss and specify non-functional requirements based on the scenarios.
- A team should discuss and include functional requirements for how to effectively include the management of functions and users in the report.
- A team should arrange contributions between team members in the report.
- A team should discuss and design the system architecture based on the team's functional and non-functional requirements.
- A team should organise a team meeting and keep the short meeting summery as the track of the meetings.

[TEAM MEMBER]

- Each team member must select one of the functions but should not overlap the selected function with other team members' selection.
- Each team member should capture the functional requirement for the selected function.

- Each team member should contribute at least one use case for the selected function in the team report.
- Each team member should contribute at least one use case diagram for the selected function in the team report.
- Each team member should contribute at least one user interfaces for a prototype implementation in the team report.

4. Submission

The team report should follow the constraints below:

Submission requirements

- Team leader must submit a team report on behalf of the team members indicating the Group number in the title of the submission. i.e., Team_A_Report.
- The team report must be in PDF format.
- The team report must be submitted via Tabula and must not be Zipped.
- Students must check if a team report has been uploaded successfully.
- Students must include the assessment front sheet.

Report Requirements

- The team report should be no more than **3000 words**.
- The report template will be available to a team.
- The report should include a title page, table of contents in the report.
- The report should include all the team members ID and name.
- There is no page limit as long as it fits the total number of words for the report.
- The report should follow a logical and well-defined structure with headings and subheadings.
- Diagrams in the report should be clearly labelled and well-presented.
- Appendices will not normally be marked but they must not include material essential to the argument developed in the main body of the work.

Late submission policy

If work is submitted late, penalties will be applied at the rate of **5 marks per University working day** after the due date, up to a **maximum of 10 working days** late. After this period the mark for the work will be reduced to 0 (which is the maximum penalty). "Late" means **after the submission deadline time as well as the date** – work submitted after the given time even on the same day is counted as 1 day late.

Resubmission policy

If you fail this assignment or module, please be aware that the University allows students to remedy such failure (within certain limits). Decisions to authorise such resubmissions are made by Exam Boards. Normally these will be issued at specific times of the year, depending on your programme of study. More information can be found from your programme office if you are concerned.

5. Grading

The learning outcomes measured by this assignment are the following, which essentially covers the topics of the first two blocks of teaching:

- Demonstrate a sound understanding of a range of software process models and symbolic representations.
- Discriminate scenarios where different design patterns and software testing strategies can be applied, and critically evaluate these patterns and testing solutions.
- Discriminate the key concepts and techniques used in the Agile Manifesto and Scrum, carefully design and critically evaluate project plans using these techniques.

The following rubric will be applied to your submission.

	0 – 40%	41 – 60%	61 – 80%	81 – 100%
Requirements Analysis (10%)	Little evidence of analysing the project scenario and task.	Evidence of analysing the project scenario and task, by providing the overall description with good structure. However, it is not very well justified.	Good analysis of the project scenario and task by providing the overall description with the consideration of system purpose, features, environment, characteristics, etc.	Excellent analysis of the project scenario and task by providing the overall description with the consideration of system features, environment, characteristics, etc. Further, the analysis considers constraints, system process, assumptions, dependencies, etc.
Functional Requirements (15%)	Little evidence of specifying functional requirements either from team or individual team member.	Evidence of specifying functional requirement from team and team members based on the requirement analysis. However, it is not very well justified and also not represented with use of symbolic notations.	Good specification of functional requirement from team and team members, based on the requirement analysis well justified. Also, it includes well representation using symbolic notations.	Excellent specification of functional requirements from team and each team members based on the requirement analysis well justified using good symbolic notations. Further, there is consistency between team members in the requirement contents.
Non-Functional Requirements (15%)	Little evidence of specifying non-functional requirements.	Evidence of specifying non-functional requirement from team works based on the requirement analysis. However, it does not consider many non-functional factors for the system.	Good specification of non-functional requirement from team works the requirement analysis. Also, it includes many factors considering basic performance, security, usability.	Excellent specification of non-functional requirement from team works the requirement analysis. It includes not only basic performance, security, usability but also further non-functional requirements such as scalability, maintainability based on design pattern, and testing, etc.
Interface Requirements (15%)	Little evidence of specifying interface requirement from team and individual team member.	Evidence of designing interface requirement from team and team members based on the functional and non-functional requirements. However, there is less design consistency between individual team members.	Good specification of designing Interface requirement from team and team members based on the functional and non-functional requirements. Also, it considers the design consistency between team members.	Excellent specification of designing Interface requirement from team and team members based on the functional and non-functional requirements. Not only it considers the design consistency between team members but also the design consistency overall the project.
System Architecture (15%)	Little evidence of specifying system architecture design.	Evidence of designing system architecture from team works based on the team's system requirements. However, it is not very well justified and also not represented with use of symbolic notations.	Good design of system architecture from team based on the team's system requirement. Also, it is well justified and presented with use of symbolic notations. It includes at least two diagrams using UML to represent the system architecture.	Excellent design of system architecture from team based on the team's system requirement. Also, it is well justified and presented with use of symbolic notations. It includes not only two or more diagrams using UML to represent the system architecture but also well-defined description of each diagram.
Agile Manifesto (15%)	Little evidence of having regular team meetings and it does not provide the meeting note summary.	Evidence of having regular team meetings by providing the meeting note summary. However, it does not include the perspectives of role, plan, retrospective, backlog, etc.	Evidence of having regular team meetings by providing the meeting note summary. It includes the well-defined perspectives of role, plan, retrospective, backlog, etc.	Evidence of having regular team meetings by providing the meeting note summary. It includes the well-defined perspectives of role, plan, retrospective, backlog, etc. Further, it shows the progress based on the team meetings.
Report (15%)	The report contains lots of grammar mistakes and is difficult to follow.	The report makes use of illustrations and references, however, it contains some errors.	The report is clearly structed and well written with little mistakes.	The report is clearly structed and well written. Also, it includes well-defined visual presentation.