University of Wollongong

School of Computing and Information Technology

CSIT314 Software Development Methodologies

SIM S2 2020

Group Project (40 marks)

TASKS

Your tasks are to:

1. Carefully read this document;
2. Form and structure your group, allocating roles and responsibilities to your members;
3. Complete the development of a given software system (described in a separate document). This should cover **all software development activities** from requirements elicitation and specification, design to implementation and testing.
4. Choose an agile method (Scrum is highly commended) for your group to follow. Choose a tool which supports that methodology for your group to use. **Taiga https://taiga.io/ for agile method is strongly recommended** (choose the public option which allows multiple team members).
5. Produce a report detailing the group's work.

SUBMISSION

1. Final deliverable (**35 marks**): **22nd May 2020**
    - Final report
    - A 15 minute recorded video of a live demo of your product
    - Source code
2. Final presentation (**5 marks**): **The last lab session**

All submission must be made to Moodle by one member of your group by the deadline.

GUIDLINES

1. The **final report** should cover at least the following:
    - A complete list of user stories.
    - For each user story, a list of complete tasks.
    - A complete and detailed design including UML **use case diagrams**, **use case descriptions, sequence diagrams, class diagrams, data persistence and user-interface aspects.**
    - Test plans, test cases, and test data that is sufficiently large enough to simulate the scale of the developed system.
    - Sufficient evidence to demonstrate that your group has followed an agile methodology and has used a tool which supports the methodology from the beginning to the end of the project. Note: **make**

sure that you record these evidence every week (e.g. taking screenshots of the product backlog, sprints, the work-in-progress software system, etc.).

- **True** group meeting records: copies of agenda, meeting minutes, action plans/items, etc.
- Member contribution for the whole project (with each member's signature)

Notes regarding member contribution report:

- List of the details of **what each team member has done and contributed** to the project.
- Indicate the contribution of each team member, and **everyone in the team should sign**. The individual contribution of each team member is assessed by all the other members. The scale can be a percentage number (e.g. 60%). Alternatively, the scale be in the form of "contributed", "very little", and "almost no contribution". For a team member who has "contributed", he/she will receive 100% of the group mark; for a team member who contributed "very little", he/she will receive 50% of the team mark; for students who made "almost no contribution", he/she will receive 0 marks for the entire group project. Your tutor/lecturer may make adjustment to this marking criterion based on practical situations.

A suggested plan:

- The first week after the lectures: finalize group and start working on the project. Produce the first complete version of the requirements.
- One **iteration/sprint per week** until the end of project. In each iteration:
  - Design, implement, and test a number of functionalities;
  - **Demonstrate the working system to the clients (i.e. tutors) to receive feedback during the second half of weekly labs.** ***Weekly progress will be observed and noted by the tutors and will contribute to the project management component of your project marks. Check the marking scheme in the last page for details.***
  - Continue eliciting and clarifying further requirements.

The project description is in the next page.

Project Description

Important Notes:

- *This Project Description provides only the **high-level goals** of this project. The development team **MUST elicit more <u>detailed and specific requirements</u> AND get feedback from "the client"** (the tutor).*
- ***The tutors will note your group's progress and interaction with the "client" since they are one of the factors contributing to the final marks.***
- ***The requirements may change during the course of the project (this is to simulate real-life projects).***
- ***At least the backend/middleware of your software product (i.e. the main code that controls/runs all application logic and hold data in memory) needs to be <u>object oriented.</u>***
- ***You need to form a group of <u>6-7 people</u> in your same lab ASAP and register your group with your tutor (see below):***
    - ***Full time cohort: Terence Chew (tchew@uow.edu.au)***
    - ***Part time cohort: Kheng Teck Tan (ktan@uow.edu.au)***

Peer-to-peer learning is an effective learning method in which students learn from and with one another. It provides opportunities for students to connect, learning about other students' ideas, views and thoughts on relevant topics. Due to recent events, there is a need for this form of learning to be conducted remotely.

Your team is asked to develop a software system that facilitates remote *peer-to-peer* learning. Using this system, students can ask and answer questions, vote questions and answers, accept an answer, provide comments and gain participation ratings. This rating should take into account all factors such as the number of questions, answers, accepted answers, comments and votes.

The system should support the following key aspects:

- Manages users (e.g. user profiles, their questions, answers, comments and participation ratings)
- Manages questions, (correct) answers, their votes and associated comments.
- Support searching for questions and answers.
- Generate various reports such as the most interesting questions, weekly questions, monthly questions, the top students (in terms of participation ratings), etc.

You must create test data that is sufficiently large enough to simulate the system (e.g. at minimum 50 users, and 500 questions and answers). You could write a script to randomly generate these data.

The marking scheme is in the next page for your reference.

Marking scheme

Component	*Out of*	*Marks*	*Comments*
Final Project Presentation (Recoded demo video + Q&A session)	*5*		
Final Deliverables			
Overall quality of the Final Deliverables	*2*		
User stories and tasks	*8*		
Analysis and Design (use cases, detailed design models, consistency between design models, consistency between design with code, etc.)	*9*		
Implementation (quality of code, functionalities implemented, sophistication of the solutions, consistency with design, etc.)	*6*		
Testing (test plans, test cases and test data)	*4*		
Effective use of methodologies (e.g. evidence of the use of a methodology and tool support, true meeting records, **weekly progress as observed and noted in the labs**, etc.)	*6*		
Total	***40***		