CMPS310 Software Engineering

- Group Project -

Milestone-2: Design and Implementation (9%)

This is a compulsory project. If you do not submit this, your grade will be 0 in this course.

Submission Due: 18/11/2023 (mid-night) on Blackboard

Fall 2023

Qatar University has received your *Milestone-1: Requirements analysis and Design*. It has decided that your team should be assigned this project to develop it further. This time QU wants that your team develops sequence diagrams of two use cases, identifies the important system properties (constraints and quality requirements), implements one use case, and tests the program. This document contains the following items:

(1) Your tasks, grade distribution to tasks, and the submission requirements

(2) The same system description as in Milestone-1, including **additional information** about **constraints and quality requirements** is available from **Appendix A**

Your Tasks:

This project concentrates on the technical aspects of the software engineering process, and hence emphasizes on the quality properties and the architectural design issues of the system. The body of your Milestone 2: Design and Implementation will consist of the following deliverables:

1) Prepare two design sequence diagrams for any two use cases proposed in your Milestone-1 submission. You must also include the use case specifications with **normal scenarios of those two use cases** and **design class diagram** in your Milestone-2 submission. However, you can update/improve those if you want to (20%).

2) Identify Constraints and group them into different categories; and recognize quality requirements/non-functional requirements (NFRs) of the system. You can propose more constraints and NFRs in addition to the requirements of the user. Most of the information is available from the systems requirements in **Appendix-A** (20%). Note: Desired quality attributes should be precisely specified with exact scenarios. Instead of just saying the system is scalable. You should be more specific by stating the scenario such as, "The system can handle more than 50 thousands orders within next 3 years."

3) Implement any one use case of your choice using Java programming language and provide screen shots demonstrating that your program works along with some sample output of the programs. Use nice user interface to give your program a professional look. You must provide the normal scenario of the use case (20%).

4) Test your program developed in Task 3. You must present test cases and test results. Explain how you have tested the program. (20%)

5) Test at least two quality requirements (NFRs) identified in Task (2). You must show how you have tested those, and explain your test results. **Note**: *You may use any testing technique(s) outlined in Lecture 10, slide no. 22; or any other testing approaches of your choice to complete this task* (20%).

Provide justifications for any assumptions you made in this milestone. Your submitted components will be evaluated for **accuracy**, **clarity**, **relevancy, justification and completeness** (especially among components and among artifacts) of your document.

Grading scheme for Milestone-2: Requirements Analysis

Deliverables	Grading %
Task (1) Two sequence diagrams Submit *normal scenarios of two use cases* and *design class diagram*	20
Task (2) Constraints and quality properties	20
Task (3) Implementation of one use case with nice interface	20
Task (4) Testing the program developed in Task (3)	20
Task (5) Testing two NFRs	20
Total	100

Submission requirements

Your assignment group/team MUST comply with the following submission requirements; **otherwise grade will be deducted**:

1. **Make sure that each team member must submit the identical electronic copy individually on Blackboard**.

2. The cover page of your submission MUST contain the following items in four bullets:

   [Each member of the group will be required to submit a peer group evaluation with an estimate of the contribution from each member to the project. Write the percentage contribution made by each team member so that it adds up to 100%. This evaluation *may* be used to adjust the marks awarded to each team member]

   - Effort distribution of the student:

     QUID:_____STUDENT NAME: ___ _____Effort given_____%
     QUID:_____STUDENT NAME: ___ _____Effort given_____%
     QUID:_____STUDENT NAME: ___ _____Effort given_____%
     QUID:_____STUDENT NAME: ___ _____Effort given_____%

   - Class section: **L01/ L02 / L51 / L53** (select one)

   - Course number_____

   - Submission date_____

   - DECLARATION: *We hereby certify that no part of this project or product has been copied from any other student's work or from any other sources except where due acknowledgement is made in the project. No part of this project/product has been written/produced for us by any other persons.*

3. Be aware of
   - *Submitted work must be students' own work*
   - *You cannot copy the project from other groups.*

Appendix-A

Conference Management System (CMS).

The systems requirements are the same as in Milestone-1, but the "Additional Information" on page 4 is new.

Systems Requirements

" Assume Qatar University (QU) wants to develop a software to manage various local and international conferences. Your team has been contracted by QU to advise regarding the development of this Conference Management System (CMS). QU has prepared the following requirements for you on how the CMS should work:

An organizer can register for a conference. The organizer first logs in, and the system finds the organizer details. It is assumed that the organizer has already registered and has login details. The organizer then provides the conference details such as conference name, dates, a list of reviewers, their expertise, and venue. The venue has address. A conference may have multiple venues. The system records the conference if no other conference with the same name exists. If exists, the system asks the organizer to re-enter new conference details with a message, "The conference already exists." The system registers the new conference details and attaches this with the information of the organizer. An organizer can register any number of conferences, but a conference can be organized by only one organizer. All reviewers' information with their expertise is also recorded if not found in the system and is associated with the conference. The reviewer is informed about his/her association with the conference. The system finally generates a conference code and makes a confirmation to the organizer once the conference has been successfully registered.

Authors can submit their papers/articles to any conference before the deadline. An author first provides his/her details if not provided earlier. The system records the author details if not exist and creates a login name and password for the author. If the author already exists, he/she simply logs in, and the system finds author details. It then finds all conferences and displays them. The author selects the conference, and the system records the selection. The author then provides paper details such as the title of the paper, names of all authors, abstract of the paper, key words, and finally uploads the paper. The system stores the paper details. It includes the paper in the selected conference and attaches the paper with the author(s). It then produces a paper number to the author. An author can submit maximum three papers to a conference. If the total number of papers for a conference exceeds 3, the system terminates the session without registering the paper. A paper can have any number of authors. However, a paper can only be submitted to only one conference.

Once the deadline passes of a conference, the organizer starts assigning reviewers to the submitted papers. The organizer selects his/her conference. The system retrieves the conference details and the list of submitted papers. For each paper, the system searches for three reviewers who are assigned to less than 3 papers to review. It assigns three such reviewers to each paper. The system knows which paper has been assigned to which reviewers. A reviewer can review papers for any number of conferences, but for a particular conference, a reviewer can review at least 2 and at most 3 papers. The reviewer is given a deadline to submit the comments/feedback and a review result in terms of *accept*, *reject*, *conditional accept*. The system notifies the reviewer about the review task. The reviewer sends acknowledgement, and the system records the acknowledgement with the reviewer details.

The reviewers can select a conference and see which papers have been assigned to them, and their deadline, assumed the reviewer is logged in. The reviewer can select a paper, the system shows the abstract of the paper. The reviewer has an option to decline to review the paper, in that case, the system then releases the reviewer from the paper, and finds another reviewer for the paper. If the reviewer accepts a paper to review, he/she submits the comments and review rating (1, 5, or 10) about the selected paper online. The system records all comments and review rating sent by the reviewer with the paper and the reviewer details. It then updates the system by marking the paper as already reviewed by the reviewer with the review date.

Once all three reviewers' comments are received, the organizer asks the system to make a decision (either accept or reject) on every paper automatically based on the comments and the review rating. The decision is recorded, attached with the paper, and the authors are notified about the decision of their paper. The accepted papers are classified separately from the rejected papers. For the accepted

paper, the author is asked to submit the revised paper and to register for the conference by a submission date. In order to register, the author provides the paper number, and credit card details by the final submission deadline. The system finds out if the paper was accepted. If the paper number does not exist, it generates an error message and asks the author to enter correct paper number. The author has a choice to select the meal preferences (Vegetarian, Halal, or Vegan) for the conference. The system stores the choice. The card details and the paper number are forwarded to a payment system called QPay that is not a part of the CMS. QPay immediately sends back the outcome of the payment. If the payment gets accepted by QPay, it sends a receipt of the payment and the paper number to the system which stores this information, otherwise it sends an error message. The system informs the author about the payment outcome.
The organizer selects a conference after the final submission deadline passed. The organizer requests for a conference proceeding. The system creates the proceedings and selects only those accepted papers with at least one author registered for the conference. If not, the paper will be removed from the accepted paper list. It prepares a table of contents of the proceedings and includes all selected papers and sends the final proceedings to the press for printing. The press notifies the system once the printing is complete. The final version of the proceedings is available to all registered authors of the conference.
The CMS can also book hotels for the authors who have already registered for their papers. In that case the author first provides their paper number along with their last name. The system asks for the date of check-in/check-out and the name of the city. The system finds a list of hotels that satisfies the author's preference. The author selects the preferred hotel. The system then contacts the hotel. The hotel makes reservation if the room is available and sends back a reservation number with the author's name, the date of check-in, check-out, and the price. Otherwise, the hotel sends a message that the room is not available. The system then forwards this to the customer. The author can later find the reservation details from the system just entering the reservation number and the guest name."

The administrator of Qatar University can generate various summary reports of the system such as how many conferences are registered in one month; which conferences are scheduled in a particular month, week, or day; how many papers submitted for a conference; what is the average acceptance rate, etc.

Additional Information

Qatar University now describes additional information about the Conference Management System (CMS). It describes that time to time, some functions may need upgrading and modifications, but the interfaces to the rest of the system may remain same. The software could be interoperable, that means, it can be integrated with varieties of other software systems. The system must by highly portable, meaning it can run on different operating systems such as Android, Linux, iOS, Unix, etc. QU mentions that the CMS system does not have budget to recruit more than 6 new technical staff for the project. Similarly, it has budget for only 10 new servers, not more than that. QU further adds that more new functionalities need to be added to the system in the future, some existing ones might be modified or deleted to enhance further capabilities. The account information of the authors is confidential; these should not be disclosed to unauthorized entities. QU also points out that several backup modules and servers will be managed in order to support availability of the systems most of the time.