

## C-SW311: Software Design and Development Spring 2025

### Submission and Discussion guidelines

#### Guidelines for project deliverables

1. For all diagrams, be **creative and rationale** on your assumptions about the information required, and try to include everything that is *important* for your model to be explanatory.
2. For EACH deliverable, each group must submit a well-written and organized **Technical Report document** containing ALL steps, tables and diagrams and also describing your solutions and rationales to the assignment (based on the deliverable requirements, preferably zipped or otherwise compressed). **Any assumptions you made during your work must be explicitly mentioned either in the Technical Report.**
3. Submit your **Technical Report document** in PDF format and No Handwriting will be accepted.
4. All submissions will be on Canvas and a submission link will be created for each deliverable.
5. **Plagiarism will be treated strictly.**
6. **NO LATE Submission will be accepted, and NO EXCEUSES**
7. **Any late submission will take ZERO.**
8. Please bear in mind that submission at the last minute might cause a network problem, and that would not be taken as an excuse. Therefore, you need to submit as early as possible on the submission day.
9. Discussions will be scheduled after the submission. Eng. Shereen will return the submitted reports to each group with feedback highlighted inline (in the form of embedded comments).

## Guidelines for Deliverable#5: Software Architecture & REST web services

**Deadline of submission:** Sat 24<sup>nd</sup> of May, 2025 at 11:59 pm.

### Architectural Design:

1. Decide on the architectural pattern/style of your system, by justifying your answer.  
You may use more than one architectural style based on your system's requirements
2. Draw a component diagram that represents the architecture of your system

### REST Web services/APIs

3. Identify two GET APIs based on your project requirements. Implement them in your working project and test them using postman
4. Identify two POST APIs based on your project requirements. Implement them in your working project and test them using postman
5. Document your developed APIs. You may follow [these guidelines](#) for best practices in API documentation

### Expected Outcomes:

1. A report that:
  - a. Answers point#1 above
  - b. Contains the component diagram of your system (point#2 above)
  - c. Document the requirements of your APIs (point#3,4 above)
  - d. Screenshots of testing your developed APIs using Postman
  - e. Documentation for your developed APIs following the guidelines stated above. The documentation should be self-contained to assist anybody who wants to use your APIs
2. The UML file of your component diagram
3. The implementation of 3&4 within your system/project



جامعة مصر للمعلوماتية  
EGYPT UNIVERSITY  
OF INFORMATICS



FACULTY OF COMPUTING  
& INFORMATION SCIENCES