# CS301 – IT Solution Architecture: Project Video Proposal (5% of total marks)

The objective of this project proposal is for the team to **understand the project functional and non-functional requirements and propose your designs that you will be implementing in your solution architecture that address the sponsor’s requirements.** This term onwards, you are also required to deliver the video proposal and no longer needs the proposal report.

## General grading criteria for project deliverables:

* 1. *For diagrams:*
     1. Clear and concise
     2. Follows recognized standards or include an unambiguous legend.
     3. The system depicted meets requirements.
  2. *For analysis:*
     1. Clarity, completeness, and conciseness of the analysis
     2. Rigour and accuracy of the analysis
     3. Correctness and impact of recommendations
     4. The degree to which the tradeoff space is discussed and understood.
     5. The specificity of argument to the given context.

Your team should work together. You may consult other teams or sources without penalty, but **any input beyond the course materials**—whether from individuals or documents — **must be acknowledged** in your submission.

## Instructions – Video Proposal

Assuming you are the architect, and you are proposing your architecture design of the solution. Your audience includes the management, development, and operation teams. This presentation is meant to solicit inputs from the sponsor and your final solution can still differ.

1. Keep your video presentation to **10 minutes**.
2. Explain your understanding of the business need. Keep this less than 2 mins.
3. Your proposal should describe the **proposed solution architecture of the project that fulfil both the functional and non-function requirements of the case study.** The team can attempt to show screens that fulfil the functional requirements too. For example, the team may choose to implement the web pages for the use cases. If time is a constraint to code, using prototyping tools (e.g., Figma) is allowed.
4. Submit your deliverables via **eLearn->Assignment-> G{SectionNo} - Project Submissions** by the deadline due. Name your video as **G1-{TeamNo}**-**VideoProposal**.**mp4** **(e.g., G1-T1-Video.mp4).**
5. Late submission or failure to comply with the instructions will be penalized.

## Grading Criteria - Video Presentation

1. [5%] Conformance to the instructions.

[-] Missing, incomplete or late submission.

1. [10%] Quality and completeness of the use cases, views, and diagramming notations.

[-] Ambiguity or inconsistency among views

[+] Clear descriptions, explanations, readable figures and tables, scannable text

[+] Good video presentation

1. [50%] Solution proposed for **sponsor’s functional requirements**.

[-] Missing designs for functional requirements.

[+] Bonus features or innovative designs proposed by the team provided basic features are already covered.

1. [35%] Solution proposed for **sponsor’s nonfunctional requirements**.

[-] Missing design for non-functional requirements.

[-] Wrong conceptual understanding of solution designs.

[-] Alternative designs not compared or not well-supported by evidence or reputable references.

[+] Bonus features or innovative designs proposed by the team provided basic features are already covered.

## Points to note

1. If you are using existing open-source codes, you and your team members must be competent to understand the codebase, able to modify and deploy the codebase.
2. Java programming language is preferred (as most class examples are covered in Java) but other programming languages are also acceptable. (e.g., Python)
3. You are required to implement this solution either entirely on the AWS cloud.
4. The solution architecture must demonstrate your proposed designs to fulfil the functional requirements and non-functional requirements according to the project requirements.
5. You can also propose to implement for other areas of qualities found in [ISO 25010](http://iso25000.com/index.php/en/iso-25000-standards/iso-25010) if appropriate.

## Required Information for the Video Proposal

### GitHub Team name, team members (name and matriculation).

### Background and Business Needs

Provide the background context of the project. Include a summary of the business needs and drivers for this application. A short summary (at most half a page) is sufficient, design should be the focus of your proposal.

### Stakeholders

List the key business and IT stakeholders for this application. If the stakeholder is interacting with the system, you will also need to describe their permissions in the system. Include your team members and their roles which you would be configuring as IAM user/roles.

|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Stakeholder Description** | **Permissions  (if not applicable, write N.A.)** |
| <Title of this stakeholder> | <short description of this stakeholder> | <what is/are the component(s) or service(s) this stakeholder needs to have access to and access level (read/write)> |

### Use Cases for Architectural Significant Requirements (ASRs)

Provide up to **three** architectural significant requirements in the form of **use cases** for the system. Your use cases should clearly demonstrate how the functional and non-functional requirements are being addressed.

Remember your role as an architect. Write the use case to involve not just the business process but also the architectural components of your proposed system. Use the below template for documenting the use cases.

|  |  |
| --- | --- |
| **Use Case Title -** <short title of your architectural significant requirement> | |
| **Use Case ID** | <any numbering format> |
| **Description** | <short description of this use case and why is it significance> |
| **Actors** | <who are involved. Can be the stakeholders or software systems> |
| **Main Flow of events** | <step by step of events how the actors use or interact with your system and its architectural components> |
| **Alternative Flow of events** | <scenarios that deviate from the main flow, e.g., exceptions flow> |
| **Pre-conditions** | <Required conditions before the start of the main flow> |
| **Post-conditions** | <Expected conditions after the end of the main flow> |

### Solution Views

Draw diagram(s) to show the “intended” structure of your system. You can draw network diagrams or AWS architecture diagrams (preferred, refer to week 3 “Web Application Architecture on AWS”).

Solution View

Draw the **sequence diagram(s) for the above use cases of the ASRs to show key end-to-end functionality of your system**. The aim of this section is for you to understand your system. Besides showing the high-level components (e.g., between applications), you can also show key modules/packages in your application.

Design View

### Quality Attributes

Provide a list of the key quality attributes designed in your solution architecture. Explain in words how your solution architecture achieves these qualities. You can present your designs in the views in the final report.

### Proposed Budgets

You are required to estimate **how much your solution costs for developing the prototype solution in this project and estimated production costs**.

**Development budget** are costs of your **software development efforts and infrastructure total costs to develop the prototype solution.** In software development lifecycle, your effort to implement the solution involves not just the code development but also other activities such as analysis, testing and management. This estimate should include efforts from all team members across the course weeks to implement this solution. If the team wish to, this estimate can also be broken down by member. You should also include the cost to deploy your prototype solution till week 14 for comparison between the planned versus actual infrastructure costs.

**Production budget** includes the monthly and annual costs to run the production environment of the IT solution. This cost should include more components than your prototype solution to address the needs in production.

In real projects, there is also development environment which is a scaled-down version of your production environment and requires budgeting too. For our project, we keep it simple, and you can leave it out for the budget. There are also other environments besides development and production (e.g., SIT, UAT etc.) which are out-of-scope for this project.

**Development Budget**

|  |  |  |
| --- | --- | --- |
| **Activity / Infrastructure** | **Description** | **Cost for Prototype Solution** |
| < e.g., Analysis and Design,  Develop web pages, Implement microservices,  System setup, Testing (unit, integration),  project management, AWS components to deploy the prototype for demo > | <Description of this activity or component> | <e.g., man-hours effort to implement service logic, [AWS calculator](https://calculator.aws/" \l "/) > |

**Production Budget**

|  |  |  |
| --- | --- | --- |
| **Hardware/Software/ Service** | **Description** | **Cost for Production Environment** |
| < e.g., AWS components to deploy the prototype in an actual production environment> | <Description of  this item> | <e.g., [AWS calculator](https://calculator.aws/" \l "/)  > |