Milestone 5 (Part2) – Architecture Report

In this report, we want you to document the architecture of your system with a justification.
In your Milestone 3 report (Requirement Documentation), you have defined and ordered *five* most important quality attributes within the context of your system. Since different designs might be appropriate for different quality requirements, you must *summarize the key quality attributes* for which you design your architecture. If the quality attributes to achieved in the architectural design phase are different from what you planned before, we need you to justify.

Due on Dec 19, 2020 11:59pm EST.
Deliverables:
- Group report:
  - Naming format: GroupNumber_ArchReport.pdf [*e.g,. Group21_ArchReport.pdf*]
- Individual report
  - Naming format: GroupNumber_yourName_ArchReport.pdf [*e.g,. Group21_ShuruiZhou_ArchReport.pdf*]

Group report (4pt)

1. Architecture documentation.

In both diagrams and text, document the architecture you propose for the system. Create diagrams that show *at least* two (2) views with which it is possible to reason about all the quality requirements you considered in your design. It is likely a good idea to create multiple diagrams for different purposes. Select suitable architectural views; include a legend for each; choose appropriate levels of abstraction for the components in the diagram; if necessary, use color/shape/text to differentiate between types of components and connectors. In addition to diagrams, include a brief *textual description* for each diagram describing what kind of views you used and about how you can reason about the requirements with it (<0.5 pages each, soft limit). You may reuse documentation from the midterm presentation, but you are not required to do so.

You may find it appropriate to merge more than one view into a single diagram. If you do this, you must be explicit about what views you are merging, and why. Otherwise, diagrams should clearly represent legitimate architectural views. Make sure that multiple views of the architecture are consistent with each other and the links are clear; if necessary provide a mapping in additional text.

2. Architecture justification.

The justification should include at least these subsections:

- **A concise, prioritized list of the overall *quality* requirements** you considered in designing the system (<0.5 pages, soft limit). You should consider the requirements you elicited during

Milestone3. Rank them in decreasing order of importance. We are asking you to include this so that readers performing an evaluation can easily look at the requirements for which the architecture was designed.

*You need to have *at least* N quality requirement justification (N = number of students in the team).

- **textual justification of your design** (1-3 pages, soft limit). Justify your design decisions and why your design is adequate for the desired quality attributes. The justification will typically take positions regarding questions such as: Which requirements were affected by the designs? Which system qualities were promoted or inhibited? What styles or tactics were considered? What are the tradeoffs between the alternative solutions? Which one is better in this context and why? What assumptions did you make in your design? A good justification will refer back to the architecture documentation to substantiate the argument.

Hints:

- It may take several iterations to get your architectural views right. Appoint someone to track the accuracy and completeness of architectural representations throughout this assignment. Do not just divide up the views among your team members and assume they show everything needed.
- The architecture report should include meaningful legends for each diagram included in the report.
- Pay close attention to the downsides of each design: advantages are often very visible, while disadvantages may be tricky to understand and explain.

Perfection. Do not try to design a "perfect" system; this way lies systems that are flawed in their complexity. Instead, choose a simple approach that balances the above concerns in a reasonable way.

Tactics. *Software Architecture in Practice, Third Edition* is a book on software architecture that is available (on Quercus Files/Books/Architecture). We have taught some of the tactics in the lecture. Note that the book is *not* a reading assignment; do not try to read it thoroughly. Instead, *use it as reference material* and select particular bits of advice that are relevant to your situation.

Individual report (8pt)

In this report, we want you to explore the alternatives of the architecture of your system. All the descriptions and hints above could be applied to this individual report.

- **For groups that picked Monolithic:** Decompose your existing prototype into at least three (3) services that can be deployed separately. Decide on a reasonable decomposition and use remote procedure calls (e.g., REST API) to communicate between services
- **For groups that picked Microservices**: Redesign your system in a Monolithic way.

Deliverables:

In both diagrams and text, document the alternative architecture of your system. Create diagrams that show ONE (1) views with which it is possible to reason about the quality requirements you considered in your design. Compare the trade-off between your design with the current group design, in terms of TWO (2) ***quality*** **requirements**.

Evaluation

(For both Group and individual report)

To receive full credit for the architectural design, we expect:

- A precise and complete description of your proposed architecture (diagrams and text, with diagrams showing at least two views). Clarity is paramount. Choose a consistent level of abstraction to document; do not provide extensive detail on one part of the architecture but gloss over important high-level design decisions on another.
- At least two views of your design.
- A list of all quality requirements you considered, ranked in decreasing order of importance.
- The explicit identification of the assumptions you made and the tradeoffs you considered in your design. Be sure to mention specific quality attributes that your design promoted and inhibited.
- An explicit comparison between your architecture and an alternative design. The comparison should be supported with technical arguments.
- A discussion that avoids handwavy, incomplete, or non-sequitur arguments: “Approach X is more reliable because there are fewer messages sent.” (How do few messages translate to higher reliability? More reliable compared to what: the original architecture or approach Y?)
- A substantive justification for your choice of architecture, grounded in the prior comparison. “We agreed it was better” is not a substantive explanation.