Distributed Systems

Final Project

Introduction

The final project SHOULD cover most of the topics and concepts introduced in the course and the exercise sessions, including (but not limited to):

- Design (challenges) of distributed systems
- Threading
- RPC
- SOAP/REST
- Microservices

It is a larger project than weekly assignments with a higher correlation to course grade (35%). You should already be familiar with the topics from lectures **as innovative ideas and deep consideration of the project requirements** give you the best points.

Task

Come up with the idea of a distributed system, discuss the requirements within groups (you may take roles as developers and clients to dig deep), and implement it. You can choose what software, hardware, services, and technology you use. Please note considering the challenges and the topics mentioned earlier help you get higher scores.

You can use existing cloud service providers (such as Azure, Amazon, Google, Heroku, etc.) or you can create your local servers.

Basic Requirements (10-25 points)

1. A **written project report** on system requirements and design (10 points)

- Must include:
  - Project's functional and non-functional requirements
  - Representation of the database system architecture
    - With justifications for choices
  - Use case diagram and the description of actors & the use cases
  - A deployment diagram with descriptions
    - Why are the nodes relevant?
    - How are the nodes connected?
- Should follow the template available on Moodle

2. A running **distributed system** that solves the problem (10 points)

- Must be distributed in nature
- Has the basic characteristics of a distributed system as discussed in the lecture (please see Lectures 1 and 2)

3. A **presentation** including code execution, and a short run-down on the codebase (5 points)

- Max 10 minutes
- A full day to be booked where groups can make presentations and demos

Advanced Requirements (25-35 points)

1. **System Quality** (5 points)

- The system is efficient, execution time is reasonable
  - Execution time might be minutes depending on the input
- The system provides a good user experience
  - Text-based / Terminal UI can provide a great user experience for the system
- The system is fairly fault-tolerant

2. **Quality of reporting** (5 points)

- The written report goes through system design thoroughly and realistically
  - It's better to write what things went wrong during design than argue that the system is perfect
  - Mention the task taken up by each group member
  - Clarity of the presentation

Return Instructions

- Submit source code files, project report, and presentation on Moodle by the deadline.
- Mention the sources you use in the report and presentation. LUT policy for the use of generative AI applies.