Design Activity Template

CSYE 6225 – Network Structures and Cloud Computing

Date: 16-09-2024

Activity number/title: A1

Team nickname: ____________________

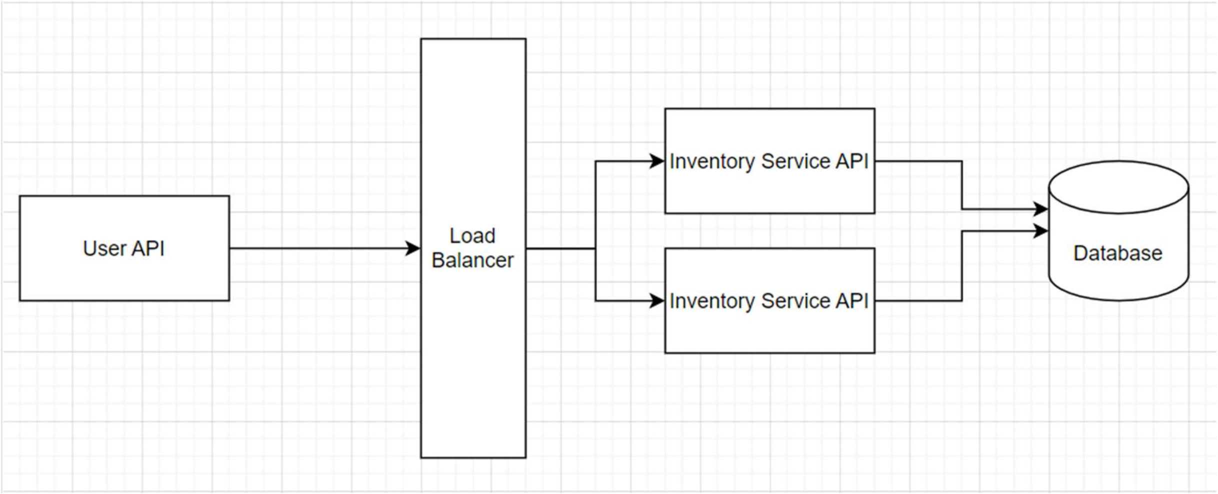
Team members:

1. Praveen Venkateshan

Notes:

1) This template is here as a guidance.
2) Students must fill provide the information required on the first page.
3) Feel free to make changes to the sections below based on relevance.
4) You may also use any tool to generate the design document.
5) **The final submission must be a single PDF document on Canvas per team.**

The design



The pros and cons

Benefits of the design:

- Load balancer distributes traffic, preventing any single instance from being overwhelmed.
- Multiple instances of the Inventory Service API can be added as traffic grows.

Drawbacks of the design:

- Running multiple instances and a load balancer adds infrastructure costs.
- The system requires managing multiple components (load balancer, multiple API instances), which increases operational complexity.

Value analysis

This design provides scalability, fault tolerance, and performance improvements, making it suitable for systems with fluctuating traffic. However, it comes at the cost of increased complexity and infrastructure expense.

Conclusion, future work, and lessons learned.

The system is well-suited for handling large-scale inventory services, providing reliability and performance improvements. Consider adding redundancy for the load balancer and optimizing latency. Balancing complexity and scalability is crucial in designing distributed systems.

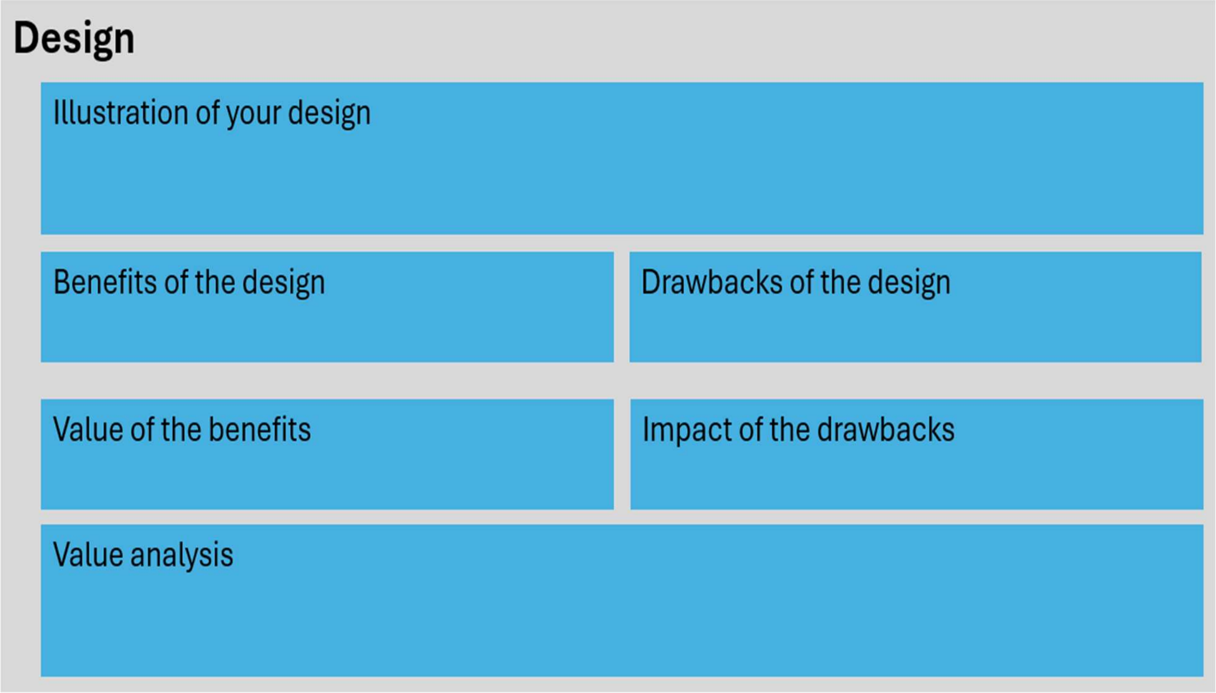
API Implementation:

The API implemented checks the inventory for the quantities of item types. If an item is not found, it throws an error message indicating that the item does not exist. Otherwise, it returns the count of the item type.

Unit Test Cases:

- In test_e2e.py, I have added three-unit test:
    1. **Add Operation**: Verifies the count after adding items.
    2. **Remove Operation**: Verifies the count after removing items.
    3. **Undefined Quantity**: Checks the count when the quantity is undefined.
- In test.py, I have added two unit test:
    1. **Success Case**: Tests the function's ability to correctly display the item type count.
    2. **Not Found Case**: Tests the function's response when an item type does not exist.

Recall:



Design

Illustration of your design

Benefits of the design

Drawbacks of the design

Value of the benefits

Impact of the drawbacks

Value analysis