Name: SARWEEN A/L ALAGES

Matrics: 207460

Quality Attribute Specification Assignment

1. You are working in company XYZ required you to develop ticketing system for Cinema, the owner of Cinema is focusing on two things:
   1. Security system for purchase ticket using Credit Cards.
   2. The performance of booking tickets.

Write scenarios for each requirement to get you client signature.

1. You are working in governmental sector and your boss required you to develop architecture to remove the overhead of official stamping for the documents without losing security.

Propose a proper architecture tactic to achieve this feature.

1. You are working in a starting company with limited budget and your boss required you suggest some tactics to reduce the maintainability cost of the software.

Propose three tactics for this objective.

* Submit your answers at in your github host.
* We will pick up randomly 5 students next session to present their answers.

1. A. Security

Source : Customers

Stimulus : Attempt to purchase ticket using Credit card

Environment : Under normal operations

Artifact : System

Response : System requests for the One-Time Password (OTP) that was sent

to customer’s phone number linked to the credit card for

authentication purposes.

Response Measure : OTP is authenticated or rejected within 1 minute.

B. Performance

Source : Customers

Stimulus : Payment is successful.

Environment : Under normal or overloaded operations.

Artifact : System

Response : System generates booking ticket using First-in/first-out FIFO queue.

Response Measure : The latency of booking ticket is within 10 seconds.

1. To reduce overhead use intermediaries to increase the resources consumed in processing an event stream. Hence, removing them improves latency.

Co-locating resources also reduce overhead. Co-location may mean hosting cooperating components on the same processor to avoid the time delay of network communication; perform a periodic cleanup of resources that have become inefficient.

1. Reduce the size of the module in the system. By splitting a module into smaller modules, the modification cost is reduced.

Increase cohesion from one module to another in the system. This is to reduce the

likelihood of side effects affecting other responsibilities in the original module. To increase the semantic coherence, place responsibilities with different purpose in separate modules. This will remove unnecessary responsibilities in the module so that they are not affected by changes meant for a responsibility.

Reduce coupling between modules of the system. Encapsulation introduces an explicit interface to a module. It reduces the probability that a change to one module propagates to other modules. Restrict dependencies restricts the modules that a given module interacts with or depends on by restricting a module’s visibility and by authorization to restrict access to only authorized modules.