Nurul Izzah Nasuha binti Mohd Faith 207394

Quality Attribute Specification

1. You are working in company XYZ required you to develop ticketing system for Cinema, the owner of Cinema is focusing on two things:

    a. Security system for purchase ticket using Credit Cards.

    b. The performance of booking tickets.

    Write scenarios for each requirement to get you client signature.

    a. The customer purchases the ticket by inserting his/her credit card under normal operation, then the system validates the credit card and allows access to the system within 10 seconds.

    b. The customer wants to initiate 500 transactions under normal operation, the system will process the transactions in average of 100ms latency.

2. 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.

    The architecture tactic to achieve this feature is authenticate the user to ensure only authorize user can access the official stamp. This can be achieved by providing some access control patterns within the system. Next, maintain data confidentiality by applying some form of encryption to dat. Encryption provides extra protection to persistently maintained data beyond that available from authorization. Thus, the documents will be protected from unauthorized access.

3. 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.

    Availability tactic

    - Fault prevention by using transaction. A transaction is the bundling of several sequential steps such that the entire bundle can be undone at once. Transactions are used to prevent any data from being affected if one step in a process fails and also to prevent collisions among several simultaneous threads accessing the same data.

    Modifiability tactic

    - Restricting modifications to a small set of modules. The goal is to assign responsibilities to modules during design such that anticipated changes will be limited in scope. The tactic that can be used is generalize the module. Making a module more general allows it to compute a broader range of functions based on input. The input can be thought of as defining a language for the module, which can be as simple as making constants input parameters or as complicated as implementing the module as an interpreter and making the input parameters be a program in the

interpreter's language. The more general a module, the more likely that requested changes can be made by adjusting the input language rather than by modifying the module.

Testability tactic

- Tactic to manage input and output for testing is by separating interface from implementation. Separating the interface from the implementation allows substitution of implementations for various testing purposes. Stubbing implementations allows the remainder of the system to be tested in the absence of the component being stubbed. Substituting a specialized component allows the component being replaced to act as a test harness for the remainder of the system.