Date _____ Day _____

HOMIOGENIZATION

Homogenization in software development is crucial for maintaining consistency, clarity, and correctness accross a system.

Following elements are involved in homogenization.

1. Combining Class:

→ To merge classes with overlapping responsibilities and avoid redundancy.

→ Examine the purpose, attributes and functions of similar classes

→ Identify synonyms and consolidate classes performing identical tasks

→ Align class names with customer terminology for clarity.

Example: Class Customer and Client both store customer details & have similar attributes (Name, Email, Address). Then merge them into a single class.

2. Splitting Classes

→ Ensure each class adheres to single responsibility principle

Example: StudentAge class handles splits

Date _____ Day _____

into Student Info and Transcript to seperate student details and course records

3. Eliminating Classes:

→ Remove unesecccary classes that donot contribute to the system.

Example: A transactionControl class only forwards data from transactionUI to transaction Processor without adding any logic or behaviour

↳ Eliminate the control class

4. Consistency Checking:

→ Ensure static (class Diagram), dynamic (state diagram), and interaction (sequence diagram) views, use consistent terminologies.

Example:

↳ In class diagram, an object is named OrderItem

↳ In sequence diagram, the same obj is reffered to as Item

↳ Rename Item to OrderItem in seq diagram.

5. Seenario Walkthrough:

→ Validate that scenarios align with the system design and no elements are missed.

Date _____ Day _____

Example:

↳ Seq diag shows Customer interacting with PaymentGateway for online payments.

Upon walkthrough:

↳ Reflexive relationship is missed: Paymentgateway interacts with its internal FraudChecker object.

6. Event Tracing:

→ Validate each message in the seq/collab diag

Example:

- In seq diag, OrderManager sends validateOrde to OrderValidation
- Verify:
  1. OrderManager has a validateOrder() oper defined
  2. OrderValidator has a validateOrder() method to recieve & process the message.
  3. An association class exists b/w OrderM and OrderValidator in the Class Diagram.

Date _____ Day _____

7. Documentation Review :

→ Ensures all classes, methods and attributes are clearly documented & std are followed.

Example :

↳ A User class has attributes userId and userName but lacks description

↳ After review : Add description

↳ Ensure consistent formatting, such as camelCase, for attributes & PascalCase for class names.