

Assignment 1

Shashwat Shah

60004220126

TYBtech Comp B

Q1 Elaborate task set for creating component level design in OO concepts.

→ Component level design is an important step in OO software development. It involves creating detailed design for individual component of software system based on requirements & specifications. Task Set involved in creating a component level design.

- 1) Identify all design classes that correspond to the problem.
- 2) Identify classes corresponding infrastructure as these classes are not described in requirements model.
- 3) Elaborate design classes that are not covered elaboration requires that all interface, ~~into~~ attributes & operations.
- 4) Specify message details when classes or components collaborate, requirements model makes up of a collaborate diagram.

Although the is optimal it can be used as a precursor to specification

→ Identify appropriate interfaces for each component in essence operation.

→ Elaborate attributes and detail data types and data structures required to implement them if an attribute appears repeatedly across a no. of design.

→ Describe process flow within each operation in detail. This may be accomplished using programming

5) Develop and elaborate behavioral representation for a class on components. State diagrams were used as part of requirement.

6) Elaborate deployment diagrams are represented in descriptor format.

Q2 Explain golden rules of user interface design

The following 3 rules are

① Place user interface

② Reduce user memory load.

③ Make interface consistent.

① Place user interface.

→ Define interface mode in such a way that does not force a user into unnecessary undesired errors.

→ Provide flexible interaction.

→ Allow user interaction to be interruptible and undoable.

→ Hide technical internals from casual users.

→ Design for direct interaction.

② Reduce users memory load

→ Reduce demand on short term memory

→ Establish meaningful defaults

→ Define shortcuts that are intuitive

→ Visual layout of interface should be based on a real world metaphor.

③ Make interface consistent.

→ Allow users to put current task into meaningful context.

→ Maintain consistency across a family of applications.

→ If past interactive model have created user expectations do not make change unless there is a compelling reason.

3 Elaborate with suitable example process of transform mapping.

→ Transform mapping is a technique to translate a data flow diagram with a specific flow characteristic called transform flow into software architecture.

① Identify requirements - Understanding software system, this involves gathering information about what system expected to do its input, output and any other functional requirement.
Eg. For developing payroll management system.

② Analyze input & output - Analyze input & output to determine structure format any constraint or requirements associated

③ Define transformation - Identify transformation needed to convert input into output.

④ Map input to transformation - Map each input to transformation it undergoes. This step involves understanding how each input

⑤ Map transformation to output - Determine which transformation contributes to generate each output. This step involves understanding how results of transformation are combined to produce.

⑥ Refine and validate : Reviews transform mapping to ensure accuracy and completeness. Validate mapping through testing and feedback to ensure that it meets requirements and produces expected results.

Q4 Elaborate with suitable examples the process of transaction mapping.

→ Transaction mapping in software engineering involves mapping flow of transactions through a system. Transaction represents discrete units of work that either complete entirely or not at all.

- ① Identify transaction types: Begin by identifying different types of transaction in system.
- ② Define transaction boundaries - Determine boundary of each transaction including action that define start and end transaction.
- ③ Identify data dependencies associated with transaction involves understanding which data elements are read.
- ④ Map transaction to processes within system that handle it. This step involves identifying modules or functions responsible for executing each transaction.
- ⑤ Define transaction flows through system includes any sequential or parallel execution paths. This step involves understanding order in which transactions are initiated and their interaction with other transactions.