Use Case Design if performed to refine use case realizations in terms of interactions, requirements on the operations of design classes, and requirements on the operations of subsystems and/or their interfaces.

Use case design is best described as outcome of a process to transform each business abstraction in a system (like: the analysis classes) into one or more design classes that are implementable representations, taking into account the properties of the target execution environment.

Steps for Use-case design preparation are:

Create use case realization

Describe interactions between design objects

Simplify sequence diagrams using subsystems

Describe persistence-related behavior

Refine the flow of events description

Unify design classes and subsystems, and evaluate your result.

Define design mechanisms

Use case design is best described as outcome of a process to transform each business abstraction in a system (like: the analysis classes) into one or more design classes that are implementable representations, taking into account the properties of the target execution environment.

1-Create use case realization

Input: output of analysis model's Analysis class diagram

In design, our realization will contain design-level information explaining how the steps of a use case will be carried out by collaborating design objects. Class diagrams, interaction diagrams, and description, of derived requirements will populate our design-level realizations.

2-describe interactions between design objects

Here we define exact details of relationship, instead of general configuration of composition, aggregate and association. And which class supports the functionality of data fetching and modification.

Similarly, contrary to defining general sequence of Object's behavior as in use case analysis. In use case design, each and every detail should be applied, like from creation to process termination of a process. So in the case of Class Diagram, because now we'll be having more operations and other changes to apply to it.

And this process will repeat until all parts of the systems are incorporated.

Input: Output of use-case realization

Output: Refined sequence diagram, Class diagram

3- Simplify sequence diagrams using subsystems (optional)

In this step we simplify behaviors by identifying repeated behavior, and grouping into subsystems.

Input: Refined sequence diagram

Output: Input: Refined sequence diagram

4-Describe persistence-related behavior

In this phase we can identify and define factory object, db object, and persistence interfaces.

Input: sequence diagram step 3

Output: sequence diagram with persistence applied

5-Describe design mechanisms

Specify a meaningful name for the persistence store, security mechanism (like secure session handling, and re-evaluation of legacy interface designing (like major interface classes are being utilized, and no duplication exist).

Input: current design model

Output: new/Modification in DB-ware