In use case analysis, we perform certain operations like, indentifying classes (which are performing within use case), realization (distribute behavior among these classes), then identifying attributes and associations.

And for the process of Analysis model generation, we may follow the following process.

1- Pick a use-case for the process

User case realization can be termed as a process of validating the behavior as intended to be designed in use case. It is achieved with several diagrams, mainly Class diagrams, and the interaction diagrams (Sequence or/and collaboration diagrams).

2- Supplement the use-case descriptions

Input: use-case description

Output: refined use-case

3- Find analysis classes from use-case behavior

Input: use-case description or refined use case

In use case, behaviors can be described as a work that system will be performed in the system, in order to facilitate the actor's desire. And that helps in discovering the Analysis Classes. There are many ways to discover it, but appropriate recommended one is to see if it complies with four rules:

1- Candidate inside system boundary, 2- it has identifiable behavior, 3- it has identifiable structure, lastly have the relationship with other candidates.

Further, there are three approaches for defining details in Analysis Classis;

1-data-driven approach: This helps in defining on the basis of database or procedural details. Which is important, but this alone is not enough.

2-A behavior-driven approach: This helps in defining operations in the Classes, and then which data will be used inside. But with this approach, it’s difficult to identify which operation belongs to which class and filtration is difficult.

3-A responsibility-driven approach: In this approach, problem domain is the main focus. And responsibilities are divided, and classes are formed in the context of mission statement for the problem. So in involves declaration of the services which class will provide to one who will request.

Output: Analysis class diagram with only classes

4-Describe the class's responsibilities

A responsibility of a class describes the services that this class will provide in our system, and that no other class will provide. Responsibilities in different classes must not overlap.

Input: Analysis class diagram with only classes

Output: Refined Analysis class with no overlap details

5-Establish associations between analysis classes

Input: Refined Analysis class with no overlap details

Output: Analysis class diagram with relationships applied

Another important step. In this step, we'll have to define relationships among classes. And there are three types of relationships in UML.

1- Association

Process: Draw solid line between classes

2- Composition

Process: line with filled-in diamond between classes

3- Aggregation

Process: unfilled diamond on the line between classes

6-Distribute behavior to analysis classes

Input: Analysis class diagram with relationships applied

Output: Sequence diagram (Interactions between objects of our analysis classes)

Two diagrams can be created in this phase of Analysis Model. Either Sequence or Collaboration diagrams.

In other reference, it was suggested that sequence diagram is not required in all cases; this should only be created where use-case is most important in system. And scale of interaction is high.