**Module 3: Requirements Specification with Use Cases**

1. **What is a Use Case?**
   * Describes sequences of events between an **actor** and a **system** that yield a valuable result​(Module 5 - ReqsUseCases…).
   * Focuses on **system behavior**, capturing how a system acts and reacts (scenarios)​(Module 5 - ReqsUseCases…).
   * Each use case is a template for related scenarios (Happy Day case and variants)​(Module 5 - ReqsUseCases…).
2. **Components of a Use Case Model**
   * **Use Case Diagram**: Visual representation showing actors, use cases, and their relationships​(Module 5 - ReqsUseCases…).
   * **Use Case Specification**: Textual description detailing the use case goal, actors, trigger events, and pre/postconditions​(Module 5 - ReqsUseCases…).
   * **Glossary**: Defines specific terminology used in the system (e.g., FURPS: Functionality, Usability, Reliability, Performance, Supportability)​(Module 5 - ReqsUseCases…).
3. **Use Case Development Process**
   * Step 1: Identify and describe actors.
   * Step 2: Identify use cases and write a brief description​(Module 5 - ReqsUseCases…).
   * Step 3: Identify relationships between actors and use cases.
   * Step 4: Outline the individual use cases, focusing on the main success scenario​(Module 5 - ReqsUseCases…).
   * Step 5: Refine use cases by identifying alternate scenarios and failure cases.
   * Step 6: Validate use cases with stakeholders​(Module 5 - ReqsUseCases…).
4. **Best Practices and Pitfalls**
   * Avoid **functional decomposition** (breaking a system into isolated tasks)​(Module 5 - ReqsUseCases…).
   * Focus on **actor goals** and how use cases satisfy them​(Module 5 - ReqsUseCases…).
   * Ensure use cases deliver a valuable result and are not overly granular (e.g., "Insert Card" is too small)​(Module 5 - ReqsUseCases…).

**Module 3: Use Case Analysis**

1. **Purpose of Use Case Analysis**
   * Focuses on understanding how use cases relate to one another after they have been independently developed​(Module 6 - Use Case Ana…).
   * **Two use case models** are suggested: one after elicitation and another at the start of analysis​(Module 6 - Use Case Ana…).
2. **Use Case Relationships**
   * **<<include>>**: Behavior of one use case is included in another. Example: Authentication must include fingerprint verification​(Module 6 - Use Case Ana…).
   * **<<extend>>**: Behavior of one use case may be extended by another use case. Example: Additional logging after 5 PM​(Module 6 - Use Case Ana…).
   * **Generalization**: One use case inherits and extends the behavior of another​(Module 6 - Use Case Ana…).
3. **Steps in Use Case Analysis**
   * Step 1: Identify shared behaviors and refactor into new use cases​(Module 6 - Use Case Ana…).
   * Step 2: Promote visibility of important extensions in the use case diagram​(Module 6 - Use Case Ana…).
   * Step 3: Consider special cases or specialized actors​(Module 6 - Use Case Ana…).
   * Step 4: Partition behaviors into **analysis classes** (boundary, entity, control classes)​(Module 6 - Use Case Ana…).
   * Step 5: Begin thinking about high-level architecture​(Module 6 - Use Case Ana…).
4. **Classes in Use Case Analysis**
   * **Boundary Classes**: Interface with the system (e.g., UI, system interfaces)​(Module 6 - Use Case Ana…).
   * **Entity Classes**: Represent key abstractions in the system (e.g., data)​(Module 6 - Use Case Ana…).
   * **Control Classes**: Coordinate behavior across multiple use cases​(Module 6 - Use Case Ana…).
5. **Pros and Cons of Use Case Modeling**
   * **Pros**:
     + Provides early buy-in from users and domain experts​(Module 6 - Use Case Ana…).
     + Helps identify who interacts with the system and what the system does​(Module 6 - Use Case Ana…).
   * **Cons**:
     + Only captures functional requirements​(Module 6 - Use Case Ana…).
     + Organizing common functionality can be challenging​(Module 6 - Use Case Ana…).