1. **Compare the Actor-System Interaction & Object Interaction Modeling**
   Actor – System Interaction: It is modeling and design of how the system interacts with the actors to carry out the use cases. It is accomplished by constructing a two-column table that describes, for each interaction, the actor input and actor action, and the system response.
   Object Interaction Modeling: It helps the development team understand the existing business processes and design object interaction behaviors to help and improve business.

2. **What is a Nontrivial Step?**
   A nontrivial step is one that requires background processing. It goes beyond the presentation layer (i.e., the user interface).

3. **How to Identify a Nontrivial Step?**
   Identify nontrivial steps in the right column of each use case. One type of actor requires the system to perform background processing. It requires software objects to interact and collaborate with each other to fulfill the request. Those kinds of actor request is a nontrivial actor request or nontrivial request.

4. **What is a scenario table?**
   A scenario table is to facilitate scenario description and facilitate translation into a sequence diagram. It facilitates the automatic generation of sequence diagrams because the mapping from the table to sequence diagram is carried out mechanically.

5. **How to construct a scenario table?**
   Constructing a scenario table involves identifying and highlighting the subject, subject action, data or objects required by the subject action, and the object acted upon. These are entered the scenario table row by row and column by column.

6. **What are the cures to Bloated Controllers?**
   <u>Add more controllers</u>: Various responsibilities should be functionally divided and distributed to the various objects. There by creating more controllers and it exhibits higher cohesion.
   <u>Design the controller to delegate the work</u>: Design the main controller that it can delegate the work according to the functionality of each controller. Distribution of function to different controller enables high cohesion and modularity.

7. **What are the Problems with the Conventional Design?**
   - The database manager must know a lot of database detail.
   - Responsibilities are not correctly assigned.
   - It is designed with a procedural programming
   - It is not an object-oriented design

8. **What is "wrong" with "A Reset Password Sequence Diagram" in slide ch10-37?**
   - It assigns getQuest() and checkAns() to the wrong object – DBMgr, which does not have the attributes to fulfill the requests.
   - It violates the expert pattern.
   - It is designed with a conventional way.

9. **What are the Steps for Deriving a Design Class Diagram?**
   1) Identify all classes used in each of the sequence diagrams and put them down in the DCD
   2) Identify methods belonging to each class and fill them in the DCD
   3) Identify and fill in attributes from sequence diagrams and domain model
   4) Identify and fill in relationships from sequence diagram and domain model

10. **How does the POS system calculate the total? Who is responsible for this?**
    - The Sale object is the obvious expert object to handle this responsibility
    - The information required to compute the total is the sum of the subtotals for each line item
    - The SalesLineItem knows the subtotal, quantity of the line item, and the Sale contains the SalesLineItems
    - Each SalesLineItem is the expert that can compute the subtotal, if it is associated with a ProductDescription to get the price

11. **In the Process Sale usecase of POS system, the last postcondition notes that "the Sale is logged with the Store". What is the type of this postcondition?**
    The Sale is logged with the sore is an association that is formed in the process sale usecase of POS system.

12. **Please draw the Sequence Diagram for the POS start up (Initialization) usecase, using starUML.**
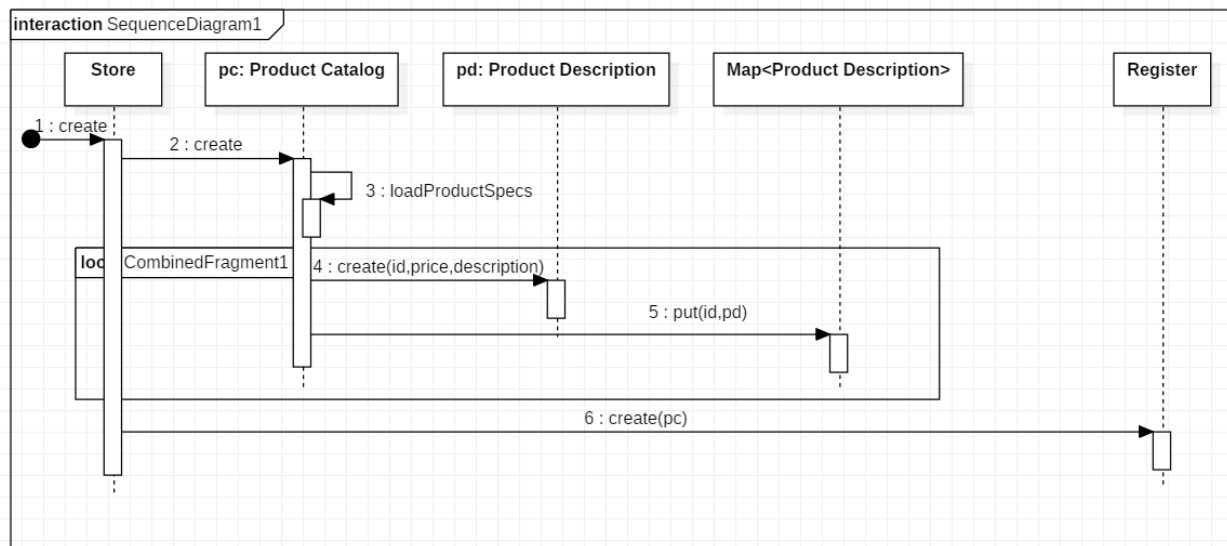
13. **Please describe your POS "start up" Sequence Diagram.**
    The store object invokes a create method which will load products in product catalog. That load product method runs in a loop to create product with respective id, description. These created products are mapped to product description object. The product catalog with all the product details is added to register using create method.