Class Assignment 1 Software Architecture: Logical Layers

1. **Choose a layer model**
   What types of logic (according to the Logic In Layers reference model) must be separated in order to meet the requirements?

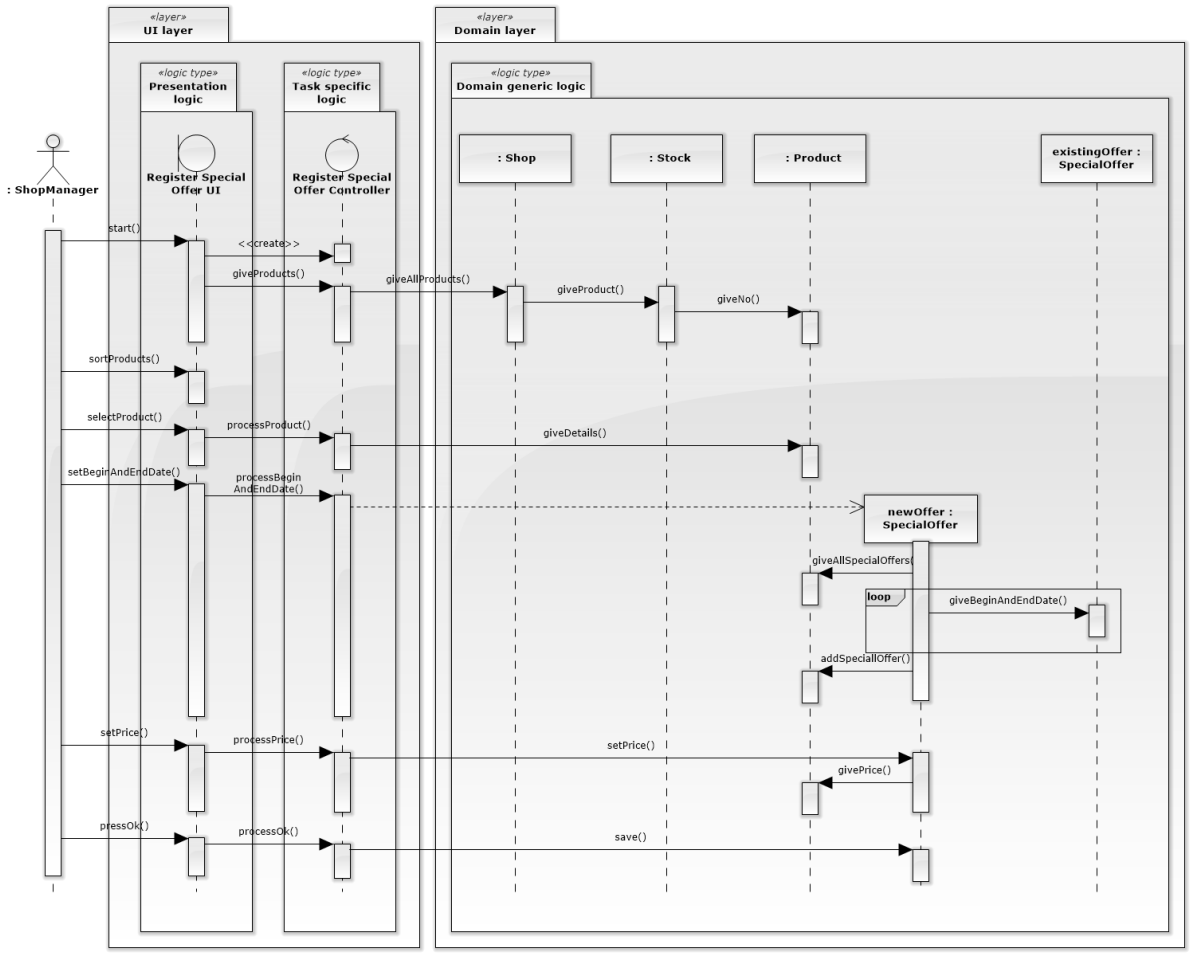
Non-functional requirement	**Logic to separate**
The new system should last a long time. At least 10 years. Therefore, the system must be easy and cheap to expand with new functions (use cases).	Task specific logic separated from domain generic logic
Possibly, in the future, sales will also be launched from mobile devices instead of only with the internet sales. The functions that will be developed, must therefore not only work with Microsoft Edge, but also with Safari (iPhone) and Chrome (Android) browser.	Presentation logic separated from domain Task specific logic
The system will be developed first with a MySQL database, because it currently used in the company. But one must keep in mind that MySQL may be replaced within a few years by a different type of DBMS (e.g. Oracle or MS SQL).	Domain generic logic separated from domain Infrastructure abstraction logic

The following questions concern the use case Register Special Offer, which has the following User Interface design.

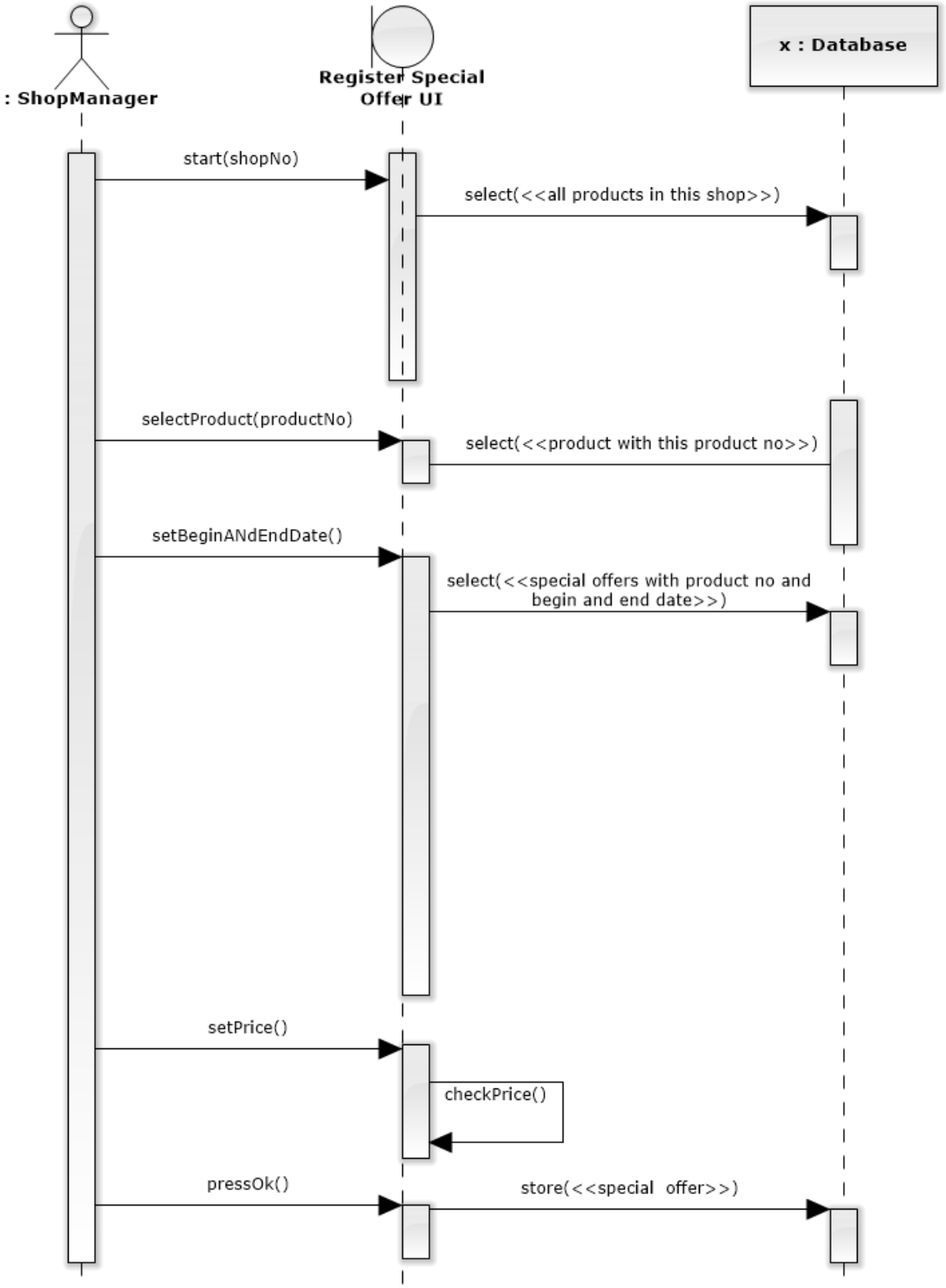
2. **What functionality belongs to which type of logic (according to the Logic In Layers reference model)?**

Functionality	**Type of Logic**
Generate a list of all product numbers of all products that are carried by a given store.	Domain generic
Order the database to return on all product properties.	Infrastructure abstraction
Create product objects based on the product properties.	Domain generic
Sort the product numbers in a different way (e.g. descending instead of ascending) if the user select that option with the mouse.	Presentation (Task specific)
Control that after the selection of a product, the product properties will be retrieved.	Task specific
Knowing that after the start and end dates have been entered using the keyboard, a new Special Offer must be created.	Task specific
Checking that for the selected product no other Special Offer is valid in the same period.	Domain generic
Being responsible for making sure that a new Special Offer is linked to a product, and vice versa.	Domain generic
Checking that SpecialOffer.price < Product.price	Domain generic
Storing the new Special Offer on disk.	Infrastructure abstraction (Infrastructure)

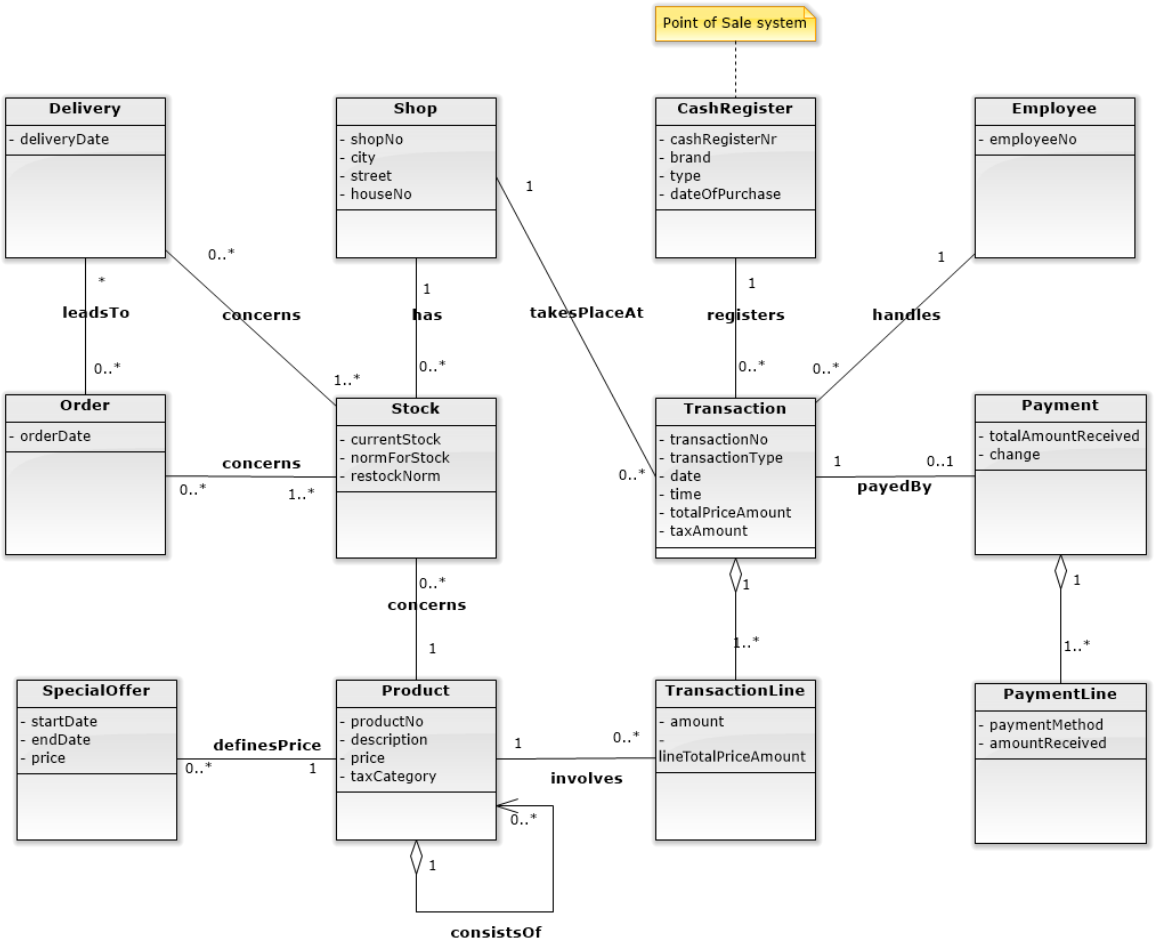
3. **Make a UML sequence diagram to carry out the use case "Register Special Offer". Assume an implementation in an object-oriented language like e.g. Java**.
   The requirements are as follows:
   - Use a 2-layer model that both aims both for Analyzability as well as Reuse of generic business logic.
     - Apply all communication rules between the layers, without exceptions.
   - Implement all the functionality of the use casem which can be found above.
     - The supplied UI-design.
     - The functionality specified in the table above.
       Ensure the following events from the actor are dealt with according to the specifications:
       - The user starts the use case 'Registrer Special Offer'.
       - The user wants to sort the product numbers in a different manner.
       - The user has selected a product number.
       - The user has entered a start and end date (for the Special Offer).
       - The user filled in the price of the special offer.
       - The user click on the OK button
   - Make a UML sequence diagram, on the logical level (as discussed in class)
     - All UI-objects of this use case may together be represented by a single object.
     - The task specific logic may be represented for this use case as a single controller object.
     - Analyse which (four) domain classes are involved from the class diagram below. Show these domain classes in the sequence diagram
   - Show which objects belong to which layer!



4. **Suppose you want to apply a two-layer model, but you want to implement the domain layer in the form of stored procedures within a relational database.**
   Which stored procedures must be programmed?



A stored procedure should be created for each database call.



Based on Leo Pruijt, *Course on Software Architecture*. Hogeschool Utrecht, 2010-2013.