Build your domain layer

1. Given in the next slide the problem statement of the POS-Point Of Service system, construct the problem domain model with the following classes we identified for you:

Register	Item	Store	Sale
Sales LineItem	Cashier	Customer	Manager
Cash Payment	Product Catalog	Product Description	

2. Include all the attributes you consider necessary.
3. Write in Java the classes belonging to the POS system domain layer. Group them in meaningful packages and relate packages with each other with dependency relationships.
4. Provide the persistence services of the lower layer that load/save information in the instances of classes in the domain layer from/in a back front relational database .
5. Test your program simulating what happen when the endSale() is issued from the cashier's interface (end of transaction).

The “Point of Service” Application

A POS system is a computerized application used (in part) to record sales and handle payments; it is typically used in a retail store.
POS includes hardware components such as a computer and bar code scanner, and software to run the system.
POS interfaces to various service applications, such as a third-party tax calculator and inventory control. These systems must be relatively fault-tolerant; that is, even if remote services are temporarily unavailable (such as the inventory system), they must still be capable of capturing sales and handling at least cash payments (so that the business is not crippled).
A POS system increasingly must support multiple and varied client-side terminals and interfaces. These include a thin-client Web browser terminal, a regular personal computer with something like a Java Swing graphical user interface, touch screen input, wireless PDAs, and so forth.
Furthermore, we are creating a commercial POS system that we will sell to different clients with disparate needs in terms of business rule processing. Each client will desire a unique set of logic to execute at certain predictable points in scenarios of using the system, such as when a new sale is initiated or when a new line item is added. Therefore, we will need a mechanism to provide this flexibility and customization.