Practical Assignment #1

PART A:

1. Requirements table:

ID	Module	Functional / Non-Functional	Description	Priority	Risk	Status
1	Inventory	Functional	The system MUST support adding, removing, and updating inventory items.	High	Low	Done
2	Inventory	Functional	The system SHOULD allow tracking of stock levels and trigger low stock warnings based on thresholds.	High	Medium	In Progress
3	Inventory	Functional	The system CAN allow generating reports of low stock items by category.	Medium	Low	Not Started
4	Inventory	Functional	The system MUST integrate with suppliers for reordering stock when levels are low.	High	High	Not Started
5	Inventory	Non-Functional	The system SHOULD support multi-user access with roles and permissions.	Medium	Medium	In Progress
6	Inventory	Functional	The system SHOULD allow setting minimum quantities for automatic restocking.	High	Low	Done
7	Reporting	Functional	The system MUST allow generating periodic inventory reports for management.	High	Low	Done

Why can't it be implemented?

3. This will be implemented after the core inventory management features are fully implemented. It requires more time and integration.

4. Integration with external suppliers will require API access and testing which is scheduled for a later phase of the project.

5. Permissions and roles management requires additional user management infrastructure, which will be built in later phases.

1.2

Open Questions:

#	Topic	Open Question
1	Supplier Integration	Should the system automatically create orders for suppliers when stock is low or manually?
2	Reporting	Should the system allow custom date ranges for the inventory reports?
3	Low Stock Threshold	Should there be different alert thresholds for different categories of products?

Presentation Layer

SampleDataInitializer

+ inventory
  reportGeneerator
  scanner

InverntoryConsole

+ inventory Inventory
+ reportGenerator
  scanner Scanner

initializes

Domain Layer

uses

1

Inventory

− items: List<
− reports: ListLst>

manages

0..*

Item

− id Integer
− hame: String

manages

1

categorized by

0.1

Category

+ id Integer
+ hame: String
+ quentity:
  minQuantity
+ location:
  costPrio-uble
+ sellPriceo-ble
+ expireDate
  category

owes

1

Category

− canegory

Discount

+ item Item
+ reportDate
+ issuetype
+ comments

0.1

discounted by

ItemRepōtt

+ date
+ itemsList>

contains

0..*

Report

+ date: Date
+ items: .Item>

generates from

ReportGenerator

+ inventory

Domain Layer