# Domain

You are required to build an application that an end user can interact with via a CLI (Command Line Interface). The application needs to be an inventory management system that needs to be able to:

* Create/Add a **customer** to the system
* Read/View all **customers** in the system
* Update a **customer** in the system
* Delete a **customer** in the system.
* Add an **item** to the system
* View all **items** in the system
* Update an **item** in the system
* Delete an **item** in the system
* Create an **order** in the system.
* View all **orders** in the system.
* Delete an **order** in the system
* Add an **item** to an **order**.
* Calculate a cost for an **order**.
* Delete an **item** in an **order**

**Sprint 1 – Mandatory**

**Sprint 2 – Nice to have**

**Sprint 3 – Nice to have**

When considering the entities in this domain:

* A **customer** needs to have a name.
* An **item** needs to have a name and a value.
* An **order** needs to have a link to **one customer** and contains a value.

You need to create for me a:

Java application running locally – can be from the IDE, which interacts with a locally hosted DB.

I have changed the order **slightly** – delete is easier to implement than update

**C R U D**

CR – **minimum**

D – Extra

U – Best

As you haven’t worked with the JDBC yet – lets scale down the difficulty of the problem.

You are no longer required to have an order that contains both **customer and items**

A simpler relationship is acceptable (one-to-many) between an order and a customer

**An order needs to have a value and a customer (its id)**

JUnit testing is fine – you will still be marked on the amount of testing you can write.

Mockito is a nice to have.

To reiterate - No CI server required.

For this you will need to know:

* DB design
* Java:
  + Basic OOP
  + JDBC
  + Scanner
* Basic Testing
* Working with a VCS

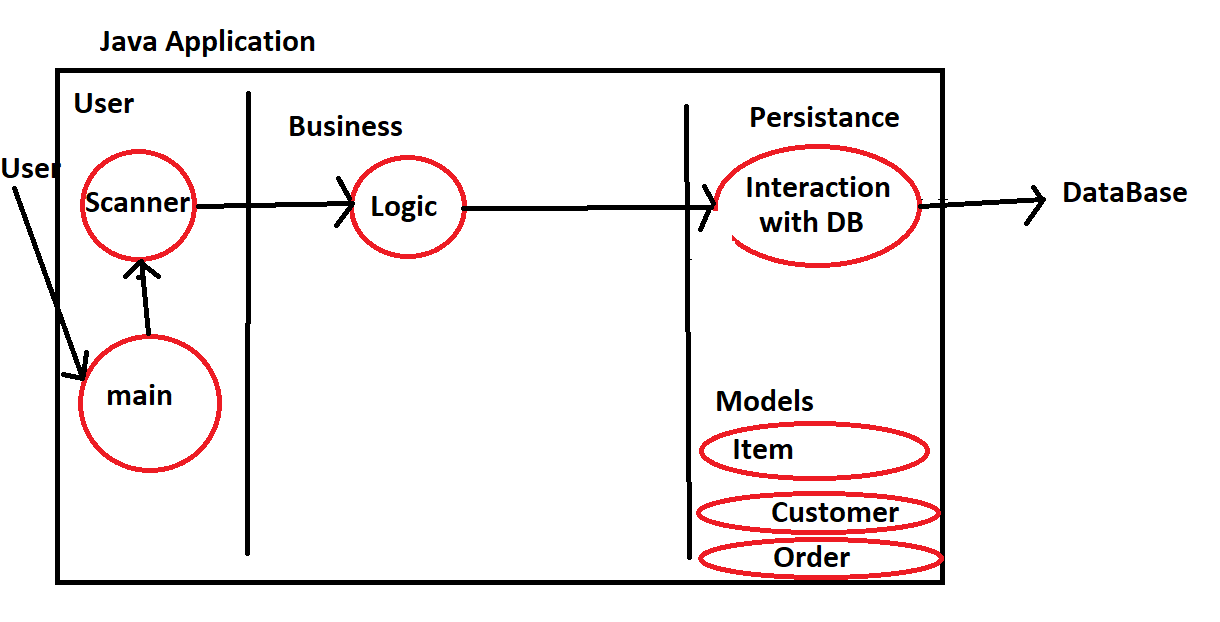
You **do not** need to show

Mockito

A many-to-many relationship

Update or delete Functionality

This is how you could structure it…

****