Lab09	IT618 Enterprise Computing
	PM Jat, DA-IICT, Winter'2017

In this lab, we implement few tasks from web based inventory management system. We use here classes InventoryItem (Lab04) and InventoryDAO (Lab07).

Implement following use cases -

a. Show all items in the stock
b. Show all items that are under stock
c. Show items under a specific category
d. Add new Items: view should use drop down box for selecting category of the item
e. Update details of existing item
f. Delete an Item
g. Delete multiple items (by selecting from check boxes)

While implementing this follow MVC and Front Controller Strategy.

For storing in database you should be creating DAO classes. A skeleton DAO for **`InventoryItem`** is given at the end of the document.

Suggested Relation schema for is -
Category(CateID, CateName)
Item(icode, idescription, istock, min_ stock, cost, category)
//icode is PK; assume that category is FK referring to another relation Category.

Interface of **InventoryDAO**

```
import pjava.inventory.InSufficientStock;
import pjava.inventory.InventoryItem;
import pjava.inventory.ItemNotFound;
import pjava.inventory.ItemExists;

public class InventoryDAO {
    InventoryDAO() {
    }
    public void InsertItem(InventoryItem items)
                              throws ItemExists, DAOException {
    }
    public void InsertItems(InventoryItem[] items)
                              throws ItemExists, DAOException {
          //should commit only if all items are successfully inserted otherwise should rollback
    }
    public void updateItem(InventoryItem items)
                             throws ItemNotFound, DAOException {
    }
```

Continues...

```
    public void updateItems(InventoryItem[] items)
                              throws ItemNotFound, DAOException {
        //should commit only if all items are updated otherwise rollback
    }
    public void deleteItems(int[] items_codes)
                             throws ItemNotFound, DAOException {
        //should commit only if all items are deleted otherwise rollback
    }
    public void deleteItem(int item_code)
                             throws ItemNotFound, DAOException {
    }
    public InventoryItem getItem(long item_code)
                              throws ItemNotFound, DAOException {
    }
    public InventoryItem[] getItems() throws DAOException {
        //returns all items from the database
    }
    public void setPageLength(int length) {
    }
    public InventoryItem[] getPaginatedItems(int page_no)
                               throws DAOException {
    }
    public InventoryItem[] getItemsCategory(int category)
                               throws DAOException {
        //returns all items for a category
    }
    public void addStock(int item_code, int qty)
                              throws ItemNotFound, DAOException {
    }
    public void withdrawStock(int item_code, int qty)
            throws ItemNotFound, InSufficientStock, DAOException {
    }
    public InventoryItem[] itemsUnderStock() throws DAOException{
    }
    public double totalInventoryCost() throws DAOException {
    }
    private int page_length = 20;
    //more fields go here, if needed
}
```