LAB221 Assignment

Type:	Long Assignment
Code:	J2.L.P0002
LOC:	300
Slot(s):	N/A

Title

Armor Management.

Background

N/A

Program Specifications

In this assignment, you are required to build an items management application, in the form of a desktop application. The program has basic functions: login, add-delete-edit item and armor's information. You are required to use the basic components to design interfaces, use the tabbed pane to organize item management and supplier management on the one screen. File is used to store information.

Program organization must clearly separate functions according to MVC model.

USING RMI WRITE INFORMATION TO FILE

Create an application using java swing to create frames like description below

- **ArmorServer.java**

```
public interface ArmorInterface extends Remote {
    boolean createArmor(ArmorDTO dto);
    ArmorDTO findByArmorID(String id);
    List<ArmorDTO> findAllArmor();
    boolean removeArmor(String id);
    boolean updateArmor(ArmorDTO dto);
}
```

```
public class ArmorServer extends UnicastRemoteObject implements ArmorInterface{

    @Override
    public boolean createArmor(ArmorDTO dto) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }
    @Override
    public ArmorDTO findByArmorID(String id) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }
    @Override
    public List<ArmorDTO> findAllArmor() {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }
    @Override
    public boolean removeArmor(String id) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }
    @Override
    public boolean updateArmor(ArmorDTO dto) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.
    }
}
```

- File Name: ArmorData.txt

  + Employee **information is constructed as following table**

Field Name	Type
ArmorId	String

Classification	String
Description	String
Status	String
TimeOfCreate	java.util.Date
Defense	int

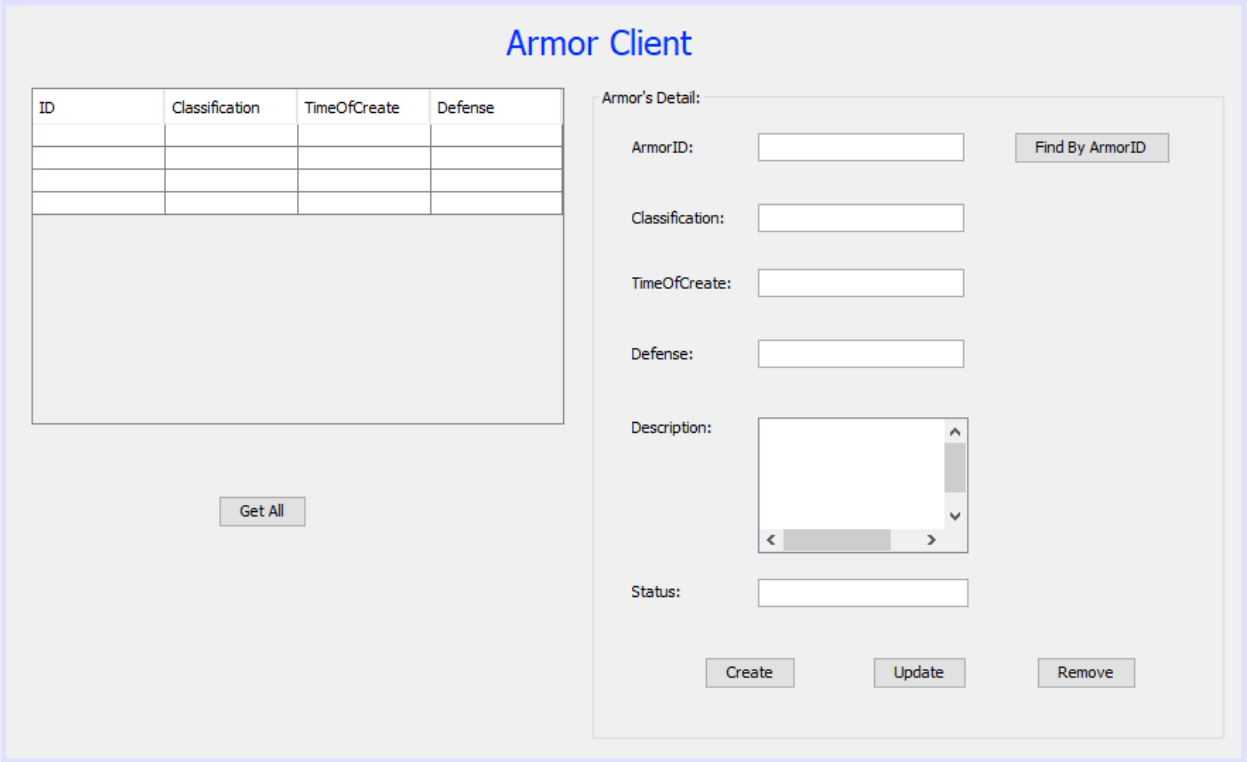
Features:

This system contains the following functions:

- **Verify constraint of following data type fields - 50 LOC**
  - ArmorId: max length is 10, not contains special characters (@, #, $)
  - Classification: max length is 30
  - Description: max length is 300
  - TimeOfCreate: The current time is automatically filled
  - Defense > 0

- **Function 1: Create Armor - 50 LOC**
  - When a user fills all information and click Create, the application starts accepting all the details of the new Armor and store it into a FILE.
  - The current time is automatically filled to timeOfCreate and not displayed at Create Armor page
  - The program checks the validity of data, if data is not valid then display an error message, otherwise insert new armor into the database.
  - The armor table must be refreshed after new data has been successfully inserted.

- **Function 2: Show All Armors - 100 LOC**
  - The screen is divided into 2 parts: Main information and Detailed information.
  - Main part: this part lists all available Armor with their information (ID, Classification, TimeOfCreate, Defense) in the system. (50 LOC)
  - Detail part: When you click a row on the table, system call findByArmorID method (50 LOC), If the application finds a match ArmorID. All information of Armor will be shown. The details of the respective armor are displayed with information as ArmorID(disabled), Classification, TimeOfCreate, Defense, Description, Status.
  - 04 buttons Get All, Create, Update and Remove button are added to detail part.



- **Function 4: Edit Armor - 50 LOC**
  - The user clicks on the Armor that he/she wants to modify on the Armor table.
  - The details of the respective armor are displayed.
  - The user changes the information of the Armor (not allow modify the EmpID). Then user clicks the Update button.
  - The program checks the validity of data, if data is not valid then display an error message, otherwise system will update Armor information.
  - The Armor table must be refreshed after new data had been successfully updated.

- **Function 5: Remove Armor - 50 LOC**
  - The user clicks on the Armor he wants to delete on the Armor table. Then users click the Remove button.
  - The program must display a message to confirm the deletion. If the user confirms, system will delete the selected Armor.
  - The Armor table must be refreshed after data has been successfully deleted.

- The above specifications are only basic information; you must perform a requirements analysis step and build the application according to real requirements.
- The lecturer will explain the requirement only once on the first slot of the assignment.