The Assignment 2 Specification and Marking Criteria

In this assignment, you are to implement a 3-tier enterprise application based on the following scenario.

The application scenario

An online car sale company needs to implement an e-business system. The system is a typical 3-tier enterprise application that integrates a presentation tier, a business tier and a persistence tier. After user requirement and budget analysis, the executive of IT Department of the company has decided to use Java EE open source technology and related platforms and tools for the implementation of the system. As an officer of the IT Department of the company, you are designated as the developer for the system core. To accomplish this task, you will need to design, implement, test and document the e-business system.

The assignment specification

1. The persistence tier

   a. The company sells both *brand new cars* and *used cars*, and may extend to more products in the future through this general framework. A major assumption is that brand new car and used car share some common properties. For example, both brand new cars and used cars have the properties of Make, Model, Drive Type, Colour, Transmission, Engine, Fuel Type, Doors and Seats etc. However, a brand new car has speciality properties such as Warranty, Extending Warranty and Roadside Assistance Packages; a used car has speciality properties such as Odometer, Rego No., Rego Expiry, Service History and VIN (Vehicle Identification Number) and Car History etc.

   b. The *car* information must be persisted into a Derby database. You are required to use Java Persistence API (JPA) to persist the car information into a Derby database. In order to reduce code redundancy, you should use **inheritance mapping**. That is, in your design and implementation, the Java entity classes should be at least **3** with **1** class as the super class to abstract the common properties and operations for both brand new car and used car and the other **2** classes as the sub-class for the specialty of a brand new car or a used car. You are required to use the **Joined-Subclass Strategy** (detailed from page 170 of the textbook) in your design and implementation.

   c. The company needs to store **customer** details and customer **orders**. You are required to use entities to persist customer details and customer orders. The relationship between a customer and his/her orders must be **one-to-many** (detailed from page 156 of the textbook), i.e. a customer can have multiple orders.

2. The business tier

   Develop the business tier by using Enterprise Java Beans (EJB). The business tier will process the data persistence or retrieval requests from users and interact with the persistence tier for accomplishing the requests.

Note: to interact with the persistence tier, the EJBs need to use Java Persistence Query Language (JPQL) to query entities and return the processed results to the presentation tier.

3. The presentation tier

Develop the presentation tier by using JavaServer Faces (JSF). The presentation tier will provide a web-based user interface, which will allow users to enter product (brand new cars or used cars) details, customer details and order details and retrieve these persisted information when needed. Presenting exception messages, e.g. a user doesn't enter data for a required textbox, is necessary and required for this tier.

Note: you need to review Backing Beans (also called Managed Beans) as the core component of presentation tier and its functions for JSF pages' navigation (detailed from page 354 of the textbook) and do some personal research if necessary.

To demonstrate success, you are required to implement all the required entities, EJBs and JSF pages (.xhtml files). To submit your assignment, you need to provide:

Part 1: Documentation

The diagram of architecture: it should include major components from all 3 tiers, which are depicted by using any formal method e.g. UML.

The typical workflow: you need to use an example workflow, e.g. creating a customer's order, to describe the tier components interaction and JSF pages navigation.

The end user manual and test instruction: you are required to provide detailed instructions about how a user would compile and deploy the software application and use the software application to persist at least 2 brand new cars and 2 used cars, **2** customers, where each customer puts in at least **2** orders, one for a brand new car and the other for a used car.

The above must be detailed in a Word document.

Part 2: Software

You will have 2 options to provide the complete implementation code of the 3 tiers, including Java source code and executable files, persistence files and JavaServer Faces xhtml pages. You can provide your implementation by either a Maven project or a NetBeans project, which must satisfy:

1. The project can be compiled directly without any further revision/re-development or debugging.

2. The generated .war application from the project can be deployed on a separate GlassFish server (not the NetBeans embedded one) to interact with a separate Derby database (not the NetBeans embedded one).

Submission

You must submit all the required files in a zip file. You must submit your assignment via the unit web site. **Any hardcopy or email submission will not be accepted. After the marked assignments are returned, any late submissions will not be accepted.**

Demonstration of the e-business system

To help you understand and develop the application, a number of screenshots and related descriptions have been provided to demonstrate the e-business system's functionality of a reference implementation. You will need to use this demonstration as a guide to ensure that your implementation complies with the assignment specification and is complete in functionality. The demonstration can be accessed on the unit web site.

The marking criteria of this assignment are detailed in the following table.

Marking Criteria	Available Marks
Design Documentation	8
1. Diagram of Architecture	4
2. Description of tier component interaction and page navigation	4
Test Instruction	10
1. User's manual for compiling and deploying the project	2
2. Inputs and expected outputs of each test	6
3. Coverage of the application scenario	2
Software	22
1. Being compilable and runnable	2
2. Conformance between the implementation of entities, EJBs, JSF pages and the given assignment specification	4
3. Functionality of data persistence	6
4. Functionality of data retrieval	6
5. Functionality of presentation (necessary exception messaging is required)	2
6. Source code in-line comments and class readability	2
Subtotal for Assignment 2	40
Late Penalty	-2 (5%) for each calendar day (either full or partial)
Plagiarism Related Penalty	
Total for Assignment 2	