**TRAINING PLAN**

**Position: BU27 Senior J2EE Developer**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Course** | **Duration**  **(days)** | **Trainer Effort** | **Trainee Effort** | **Trainer** | **Note** |
| 1 | Java Core | 3d | 4h | 3d | VinhNX8 |  |
| 2 | Struts Web Application | 6d | 4h | 6d | VinhNX8 |  |
| 3 | Spring – RESTful WS | 6d | 4h | 6d | VinhNX8 |  |
| 3 | Enterprise Java Bean (EJB) | 3d | 4h | 3d | VinhNX8 |  |
| 4 | Hibernate (ORM) | 3d | 4h | 3d | VinhNX8 |  |
| **Total effort** | | **21d** | **2.5d** | **21d** |  | |

# **Objectives**

* Provide knowledge about backend system, web service and configuration for trainee.
* Trainee has good theory knowledge and practical experience.

# **Covered Knowledge**

## **Java Core**

* OOP & SOLID: characteristics, practical exercises, quiz exam.
* Collection: Set (HashSet, TreeSet), List (ArrayList, LinkedList), Map (HashMap, TreeMap).
* Concurrent collection: CopyOnWriteArrayList, ConcurrentHashMap.
* Compare performance among: single thread collection, synchronized collection, concurrent collection.
* Algorithm Complexity.
* Cost of computing.
* Floating point.
* Serialization & Externalization: usage and performance comparison.
* Garbage Collection: processing, configuration & optimization.
* Class Loader: types, priority, sefl-defined class loader.
* Multithreading: synchronize, deadlock, collection manipulation.

## **Project Assignment**

### **Struts1 Framework**

* Action classes.
* Threading Model.
* Servlet Dependency.
* Testability.
* Harvesting Input.
* Expression Language.
* Binding values into views.
* Type Conversion.
* Validation.
* Control Of Action Execution.

### **Enterprise Java Bean**

* EJB Session Beans.
* EJB Message Driven Bean.
* EJB Entity Bean.

### **Spring - RESTful WS**

* IoC, DI.
* XML configuration vs Code base configuration.
* Spring-MVC: front controller, response handler, message mapping, view resolver.
* Jackson, JAXB auto binding.
* Spring security.
* Sping AOP.
* Spring JdbcTemplate.
* Spring RestTemplate.
* RESTful web service with Spring-MVC.
* RESTful web service with Jersey.

### **Hibernate**

* Hibernate Configuration.
* Hibernate Mappings (XML, Annotation).
* Hibernate Query.
* Hibernate Caching.

### **SOAP**

# **Training Schedule**

## **Java Core**

### **Duration: Total: 3 days.**

* Half of 1st day: section 1,2,3,4,5,6,7,8.
* 1st, 2nd day: present assignments (1,2,3,4,5,6,7).
* 3rd day: Quiz exam and interview questions.

### **Objectives**

* Understand the concept of OOP and best practices.
* Deeply understand about Java Collections.
* Can compute algorithm complexity & cost of computing.
* Can measure and monitor system resource of JVM.
* Understand the process of Java Class Loader.
* Understand floating point presentation in Java.

### **Contents**

1. Object Oriented Principal & SOLID principals
   * Characteristics: understand and give example in Java.
   * Exercises: design class diagram using UML.
2. Collection: understand the process of add, remove, iterate.
   1. Set: HashSet, TreeSet.
   2. List: ArrayList, LinkedList.
   3. Map: HashMap, TreeMap.
3. Concurrent collection: understand process of add, remove, iterate.
   1. Fail-fast iterator vs Fail-safe iterator.
   2. CopyOnWriteArrayList.
   3. ConcurrentHashMap.
4. Collections and performance comparison:
   1. Single Thread collection.
   2. Synchronized collection.
      1. Vector.
      2. HashTable.
      3. Collections.synchronized….();
   3. Concurrent collection.
5. Serialization & Externalization
   1. Default Java Serialization process: serialVersionUID, transient, readObject, writeObject.
   2. Externelization process.
   3. Performance comparision.
6. Garbage Collection:
   1. Memory Allocation.
   2. Garbage Collecting processing.
   3. Optimize Garbage Collector.
7. Class Loader:
   1. Types & priority.
   2. Self-defined class loader.
8. Multithreading:
   1. Synchronize block vs method.
   2. Deadlock.
   3. Thread-safe.
9. Floating point
   1. double
   2. float
   3. strictfp

### **Assessment**

* 1st day assignments:
  + Assignment 1: Provide examples of OOP & SOLID principals.
  + Assignment 2: Provide examples of using Set (HashSet, TreeSet), List (ArrayList, LinkedList), Map (HashMap, TreeMap).
  + Assignment 3: Provide examples of using concurrent collections.
  + Assignment 4: Provide examples to compare among: normal collection, synchronized collection, concurrent collection.
* Day 3: Fix assignments.
* 2nd day assignments:
  + Assignment 5: Propose an example that HashMap ALWAYS store only one key-value pair.
  + Assignment 6: Provide examples of using Serialization & Externalization.
  + Assignment 7: Provide examples of synchronize, deadlock, Mutex, Critical Sections, Semaphore.
* 3rd day:
  + Fix assignments (pitch one to present randomly)
  + Quiz examination: quiz exam on paper, duration: 2h (pass score: 80%).
  + Face to face interview: 30 minutes.

## **Bidding Web System**

### **Duration: Total: 10 days.**

* Half of 1st day: introduce project assignment.
* 1st – 9th day: implement project assigment.
* 10th day: Present about project assignment.

### **Objectives**

* Develop/Deploy RESTful WS, SOAP WS, RMI, EJB/JPA.
* Understand about MVC architect of Struts/Spring.
* Understand the reason why we need to use Struts/Spring.
* Understand some design pattern in Struts/Spring.
* Distinguish Spring Web Application v/s Web Service.
* Understand the process of receiving request and producing response.
* Can use Logging framework.
* Trainee can use XML-based and code-based configuration.
* Trainee can use deployment tool: Maven, Gradle.

### **Project Assignment**

* Bidding System is an web system allow people to post their item to bid. Other people can bid an existing item. The highest price will win the item. After end\_bid\_time, this item doesn’t allow to bid any more.
* Bidding System has some tables as below:

**Accounts**

------------

account\_id

email\_address

first\_name

last\_name

**Items**

--------

item\_id

account\_id  -- of the auctioneer

item\_name

item\_description

starting\_bid\_amount

start\_bid\_datetime

end\_bid\_datetime

**Bids**

-------

bid\_id

item\_id

account\_id  -- of the bidder

bid\_amount

transaction\_datetime

* **Functional Requirements:**
  + Authenticated User:
    - Can view item and current price (price by their currency).
    - Can bid a new price.
    - Can post a new item.
    - Can start bidding for an item.
    - Can stop an existing bidding.
  + Public User:
    - Can view list of item and current price (price by their currency).
* **Non-functional Requirements:**
  + **Database: Oracle**
    - Write procedures:
      * Create/update accounts, make sure that email address is unique.
      * Create/update auctioned items, can specify start and end date of auctioned for each item.
      * Create/update bids, make sure that each bidder will only have 1 transaction for every item they bid on.
    - Write views:
      * Show all the auctioned items which are open for

bidding.

* + - * Shows the highest bidder for each auctioned items.
  + **ITEM WS: Items RESTful Web Service:**
    - Deploy RESTful web service using Spring-WebMVC to get information about items from database.
    - Log duration of each processing by using Spring AOP.
    - Implement 3 layers model:
      * Presentation: request/response by JSON, validate requested parameters.
      * Business: use @Transactional.
      * Persistent: user Hibernate JPA.
  + **BIDDING ENGINE: EJB3**
    - Provide Bidding service using statefull session bean:
      * Engine for bidding base on item\_id and account\_id.
      * User can’t bid their own items.
      * User need to login in order to bid an item.
      * Bidding engine will check user credential before allowing bidding by send SOAP request to Authorization WS.
    - Provide Currency Exchane service using stateless session bean:
      * Use dummy data for currencies rate.
      * Provide currency exchange for user from different country.
* **Trainee need to submit:**
  + Source code.
  + Binary file: (war file) can deploy into web server.
  + Presentation (Power Point or equivalent): 15 minutes to present.
* **Face to face interview: 30 minutes.**

**BROWSER**

**FRONTEND**

(Struts1)

**RESTful WS**

**SOAP**

**BIDDING ENGINE**

**Bidding**

(EJB Stateful Bean)

**CurrencyExchange**

(EJB Stateless Bean)

**BiddingDAO**

(EJB Entity Bean)

**ITEM WS**

**Item WS**

(Spring Rest WS)

**Hibernate JPA**

**Oracle**

**AUTHEN SERVICE**

**Authentication WS**

(SOAP WS)

**Authorization WS**

(SOAP WS)

**Session Storage**

# **Training Documents**

* Traing Plan (this file).
* Questions & Answers for Quiz exams.
* Example projects.