**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.

## **Struts 1 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.

# **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.

## **Struts Web Application**

### **Duration: Total: 6 days.**

* Half of 1st day: introduce project assignment.
* 1st – 5th day: implement project assigment.
* 6th day: Present about project assignment.

### **Objectives**

* Understand about MVC architect of Struts.
* Know how to use Expression Language.
* Can implment validation.

### **Contents**

1. Action classes.
2. Threading Model.
3. Servlet Dependency.
4. Testability.
5. Harvesting Input.
6. Expression Language.
7. Binding values into views.
8. Type Conversion.
9. Validation.
10. Control Of Action Execution.
11. Give some solutions for:
    1. Change location of the struts.xml file.
    2. How do we set checkboxes false (on uncheck)?
    3. How can we set the focus on a form field?
    4. What is the analogy to ForwardAction?
    5. How do we emulate the defaul=”true” attribute of a Struts 1.x Action mapping?
    6. How do we extend an action mapping in struts.xml?
    7. Can we use DynaBeans?
    8. How do we set a token to track duplicate submits?

### **Assessment**

* Propose one web application using struts1:

+ Bidding System

+ Shopping Cart

+ Human Resource Management

🡺 Choose only one project, or choose your self-defined project.

* Functional requirements: (at most 3 businesses)
  + Login/Logout.
  + ...
  + ...
* Non-functional requirements:
  + Use Expression Language
  + Validation
  + Error Handling
  + Logging
  + Logging method execution duration
  + Authentication/Authorization
  + Store data by HashMap

### **Objectives**

* Understand the reason why we need to use Spring.
* Understand some design pattern in Spring.
* Distinguish Spring Web Application v/s Spring Web Service.
* Understand the process of receiving request and producing response.
* Can use some functionalities of Spring.
* Deploy RESTful WS by Spring-WebMVC or Jersey into Spring container.
* Can use Logging framework.
* Trainee can use XML-based and code-based configuration.
* Trainee can use deployment tool: Maven, Gradle.
* Can use Hibernate in spring & understand Hibernate technologies.

## **Spring – RESTful WS - Hibernate**

### **Duration: Total: 6 days.**

* Half of 1st day: section 1,2,3,4,5,6,7.
* 1st – 5th day: self-implement project assignment.
* 6th day: present and interview.

### **Contents**

1. Spring introduction:
   * Inversion of Control: Dependency Injection (DI), Autowire.
   * Spring Bean: scope & usage.
   * Spring MVC (\*: example) : **front controller (RequestDispatcher)**, **response handler**, **message mapping**, **view resolver**.
   * Message Auto Binding: **Jackson, JAXB**.
2. Design Pattern in Spring:
   1. Inversion of Control.
   2. Model View Controller.
   3. Front Controller.
   4. Singleton.
   5. Template method.
   6. Proxy.
   7. Service Locator Pattern.
   8. Factory.
3. Spring functionalities:
   1. Spring AOP: logging, time measuring feature.
   2. Spring Security:
      1. Spring security component.
      2. Token based authentication.
   3. Spring JdbcTemplate.
   4. Spring RestTemplate.
4. RESTful Web Service:
   1. **Implement RESTful WS by Spring-WebMVC.**
   2. Implement RESTful WS by Jersey.
   3. CRUD with **POST, GET**, PUT, DELETE method.
5. Hibernate:
   1. Hibernate Configuration: connect to MySQL or Oracle.
   2. Hibernate Mapping (XML or Annotation):
      1. One-To-One.
      2. One-To-Many.
      3. Many-To-One.
      4. Many-To-Many.
   3. Hibernate Query: Criteria, Hibernate Query Language.
   4. Hibernate Caching:
      1. First-level cache.
      2. Second-level cacheehcache: ehcache.
      3. Query-level cache.
6. Logging framework:
   1. Log4J framework.
   2. SLF4J & LogBack.
7. Change configuration style: from XML-based to code-based configuration.
8. Change deployment tool: from Maven to Gradle.

### **Assessment**

* Project assignment: Bidding System Project (BSP).
* Functional Requirements:
  + User: can view products and current price in real time and bid a new price.
  + Admin: can create new product, view-update and delete existing product.
* Non-functional Requirements:
  + Database: RBDMS (Oracle, MySQL, PostgreSQL,
  + Implement 3 layer model:
    - Presentation layer:
      * Deploy RESTful web service using Spring WebMVC.
      * Validate user request parameters.
    - Business layer:
      * Use @Service to configure Service Bean.
      * Use @Transactional to control transaction.
    - Persistent layer: use Hibernate.
  + Log duration of each processing by using Spring AOP.
  + Transaction management is implemented by Spring AOP.
  + Security management is implemented by Spring AOP + Spring Security.
    - User: can view and bid only.
    - Admin: can create/read/update/delete product. Admin can’t bid.
* Trainee need to submit:
  + Source code.
  + Binary file: (war file) can deploy into web server.
  + Release Note: mention all features and functionalities.
  + Interface Control Document (ICD): define interface to consume backend services.
  + Presentation (Power Point or equivalent): 15 minutes to present.
* Face to face interview: 30 minutes.

## **Enterprise Java Bean (EJB)**

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

* Half of 1st day: introduce about EJB and give assignment project.
* 1st – 3rd day: self-implement assigned project.
* 3rd day: presentation.

### **Objectives**

* Understand EJB architecture.
* Can implement and deploy EJB Session Bean (stateless, statefull).
* Understand and can declare/implement EJB Message Driven Bean.
* Understand and can implment EJB Entiy Bean.

### **Contents**

* EJB3 architecture.
* EJB Session Beans:
  + Stateless session bean.
  + Stateful session bean.
* EJB Message Driven Bean.
* EJB Entiy Bean.

### **Assessment**

* Move all business logic of previous project to using EJB bean.

## **Hibernate (ORM)**

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

* Half of 1st day: give requirements to transfer data storage to RBDMS and Hibernate.
* 1st – 3rd day: self-implement.

### **Objectives**

* Know how to configure Hibernate.
* Know how to integrate Hibernate into Struts/Spring.
* Know Hibernate Query Language.

### **Contents**

* Change data storage of last project to DBRMS
* Use Hibernate for ORM: use HQL, Criteria, Projections, Aggregation, Grouping, Restrictions
* Configure caching in Hibernate.

### **Assessment**

* Change data storage of all previous project to RDBMS and Hibernate for ORM.

# **Training Documents**

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