**Capabilities tested:**

* **CJCP010 -** Define relational database to build database for an application
* **CJCP011-** Write DDL & DML SQL statements to query the database
* **CJCP014** - Define and understand JDBC to connect to database through JAVA application
* **CJCP003 -** Follow best practices in exception handling and exception funneling to make an application more reliable.
* **CJWP005 -** Build client layer using HTML, Java script and CSS for an enterprise application to implement client side logic
* **CJWP001-** Develop XML files to script/setup configuration details of an application
* **CJCP004-**Define and understand object oriented design principles
* **CJCP018-** Design Java classes which adhere to the object oriented principles
* **CJCP019-**Implement basic Java program for a given basic object oriented design
* **CJCP002-** Define and understand the exception handling in Java to increase reliability of an application
* **CJCP005 -** Define and understand collection framework in JAVA to handle group of objects
* **CJCP020 -** Use appropriate classes from core java collection package to handle group of business objects
* **CJCP021 -** Design & implement Java classes to use advanced object oriented concepts in an application
* **CJCP007 -** Use Java 5 features - Generics & varargs to handle different types of objects in an application
* **CJWP006 -** Define and understand servlet class to build controller class of an application
* **CJWP009 -** Define life cycle of JSP pages to interact with servlet class
* **CJWP003 -** Build persistence layer using JDBC for an enterprise application to establish connection with DB and query the DB
* **CJWP014 -** Define controller class and develop it using Servlet class
* **CJWP007 -** Build web tier using Servlet to develop an enterprise application
* **CJWP010 -** Use Servlet and JSP to build web layer of an web application
* **CJWP013 -** Implement MVC design pattern using Servlet and JSP to build web layer for an application
* **CJWP008 -** Enhance web tier using Ajax and Servlet to make more responsive web application
* **CJCP022 -** Use appropriate classes from java.sql package to query the database from Java application
* **CJCP006 -** Use appropriate classes from java.io package to handle file system of an application
* **CJET007 -** Write unit tests using Junit framework to test business classes of an application

**Domain Allocation System**

Design and implement domain allocation system. It will be used by the project managers for allocating the employees into different domain of projects.

Create following two tables: **Domain** and **Employee**:

**Domain**

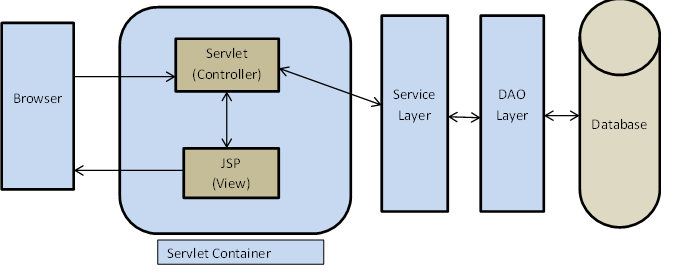
|  |  |
| --- | --- |
| **DNo** | **DomainName** |
| **1** | **Banking** |
| **2** | **SCM** |
| **3** | **Insurance** |

**Employee**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Constraint** | **Description** |
| MID | INT | Primary Key | It must be auto increment |
| Name | Varchar(10) | NOT NULL |  |
| DomainNo | INT | Foreign Key | References to **DNo** column of Domain table |
| Duration | INT | NOT NULL |  |

**Note:**

1. Do not change the table structures.
2. Values for **‘Domain’** table must be inserted manually (as given in above table diagram) before execution of your application.
3. Values for **‘Employee’** table will be inserted through your application using hibernate and spring.
4. **READ THE QUESTION PAPER COMPLETELY BEFORE YOU START CODING.**
5. Use appropriate data types and precisions for the variables
6. Handle all exception and alternate flows
7. The solution must follow below MVC architecture



1. Create appropriate packages and classes
   * com.mindtree.dao
   * com.mindtree.dao.jdbcimpl
   * com.mindtree.entity
   * com.mindtree.exceptions
   * com.mindtree.controller
   * com.mindtree.service
   * com.mintree.test
2. Use JSTL tags in all JSP pages
3. Create necessary tables in the database and **upload the table script along with project solution**. Read the question paper completely to know how the data have to be inserted.
4. Use generics wherever applicable for parameters and return type
5. Write at least three test cases (both positive and negative/exceptional) for service class

Home page contains following links

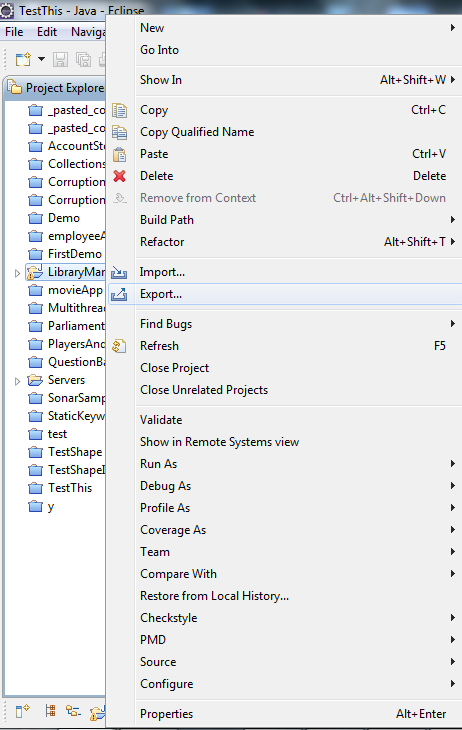
1. **Allocate employees**
2. **View employees and record into a file**

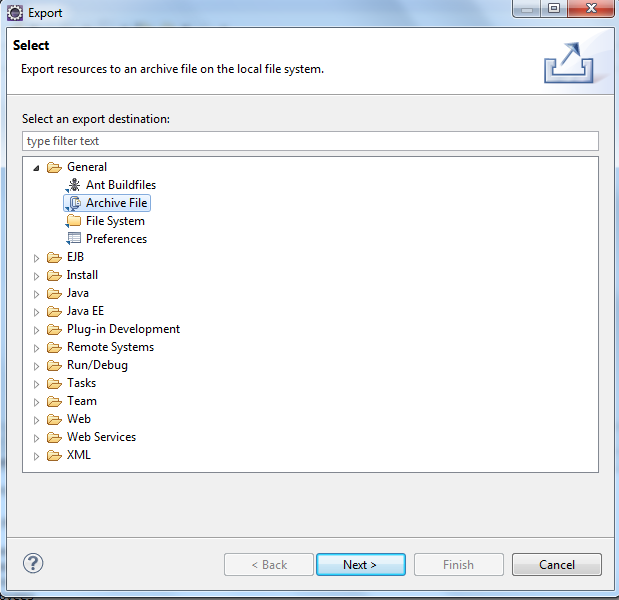
|  |  |
| --- | --- |
| **Use Case#1** | **Allocate Employee** |
| Trigger | Manager clicks on **‘Allocate Employee’** on the home page |
| Pre-Conditions | Domain table with values must exist in the database |
| Post-Conditions | Information about employees should be stored in **‘Employee’** table. |
| UI Screen Details | The input contains following :   * MID : Text box, value must be in alphanumeric * Name : Text box, value must be only alphabets and in uppercase * Domain: Dynamic dropdown combo box containing the stream names populated from the Domain table. * Defaults to **‘--Select—‘** * Duration: Text box, value must be in months (numeric value, for example: 06, 10, 18, etc.) * Allocate : a submit button * Cancel : a button to go back to home page |
| Main Flow | User enters value for MID and Name and selects the Domain from dropdown list then clicks the ‘Allocate’ button. If employee has not allocated to any of domain previously then he/she should be allowed to allocate to a domain and a message – **“Allocation successful”** should be displayed. If Employee had already been allocated, a message - **“This Employee has already been allocated”** should be displayed. |
| Data Validations | * All fields are mandatory, MID start with M followed by 7 numbers; name should be in uppercase. * Duration must be numeric. It should not be zero and it should not more than 24 months. * **All validations should be done using Spring.** |
| Business Rules | Employee can be allocated at most for one domain |
| Alternate Flow | If user clicks on the cancel button (no hyper link) he/she should be redirected to home page |
| Exception handling in DAO layer | If given employee had already been allocated to a domain, then **“EmployeeAlreadyAllocated”** exception must be thrown |

|  |  |
| --- | --- |
| **Use Case#2** | **View Employees** |
| Trigger | User clicks on **‘View Employees’** on the home page |
| Pre-Conditions | Domain table which contains domain details must be pre populated and stored in the database |
| Post-Conditions | Information should be displayed in table format.  The output should be in a tabular format and should contain only the Employee details of the selected domain from the dropdown list.  The columns must include:  A header Which displays the selected Domain Name   * MID * Name * Duration   The report displayed should also be saved into a file for future reference |
| UI Screen Details | * **Domain:** Dynamic dropdown combo box containing the domain names should be populated   + Defaults to **‘--Select—‘** * **View Details:** a submit button (use Ajax to implement this feature) * **Back:** a hyperlink for returning to home page |
| Main Flow | User selects the domain from dropdown list then clicks the ‘View Details’ button. It will display the list of employees who works in respective domain. If none of employees allocated for selected domain, then **‘No employees found’** message should be displayed without any table |
| Alternate Flow | If user clicks on the hyperlink ‘back’ he/she should be redirected to the home page |

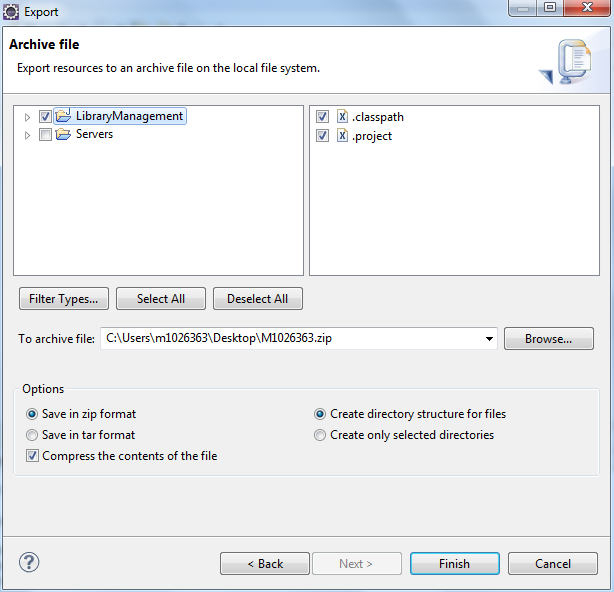
**Follow below instructions to upload the solution document:**

* Remove JAR files from all Java projects before uploading to learning hub (LMS)
* Save all Eclipse projects modified along with the entire source
* If you did not check the ‘Copy project into workspace’ option while importing the project, export these projects by right clicking on the project name in the workspace in Eclipse, click on ‘Export’->General->Archive File click on ‘Next’.





Select your project on the left top pane and the necessary “.classpath” and “.project” files on the top right pane.



Select the location to store the zip file in and click on ‘Finish’.

* Name the ZIP file as MXXXXXX.zip (It will be your employee ID) and add following files
  + Modified/Updated Eclipse projects
  + DB script
  + Upload MXXXXXX.zip as your final deliverable.