Java-FLP parallel project

(Project along with regular learning sessions)

EMS – Employee Management System

INDEX **Page Number**

APPROACH AND EXECUTION PROCESS

Learning project for FLP associate in parallel to their training - Basically participant will work on this after daily's regular session hours, and on Saturdays. So time for EMS work will be evening 5:30 to 7.

Initially, two full days will be dedicated for EMS; to understand the project; to decide the package structure, nomenclature, data model (domain objects) and start the work. After two days, FLP participant will keep working on it till the final project comes.

Project tile - EMS (Employee Management System)

APPROACH - Associate will develop the application in 4 phases, in all 4 phases complete requirement should be achieved.

Phases -

- **First**: immediately after completion of Core Java sessions (after collection API) – Associate will develop application using core Java only.
- **Second**: after RDBMS (SQL) and JDBC session – use of real DB for the application
- **Third**: after the Servlet & JSP session – revamping of the application using the JSP and web technologies like JS, HTML.

It is desirable that in all phase – complete functionality (create, modify, remove, view and search) should be achieved; but it should not be developed every time. It should follow a good package structuring (presentation, business and persistent layer), so that in phase 2 (when JDBC and database will be introduce) only persistent layer should be change. Use of proper interfaces and nomenclature is desirable. Similarly in case of conversion from plain Java application to web base application, only presentation layer should change and no change should be required on business (service) or persistent layer.

Purpose of this exercise is to understand benefit of proper MVC structure, layer based design, reusability and ease of development (realization of requirement) with advancement of technology.

In other word they will EXPERIENCE the power of technology and its real usage.

A BRIEF ON EMS (Employee Management System)

ABOUT EMS

EMS is an application, which helps to manage an employee details like – personal and professional info of employee. Personal info like Name, Date of birth, email id, address etc. And professional info like – Role, project, departments, etc.

EMS aims to manage complete life cycle of employee viz **add** (create an employee – where all info will be captured for the first time), **edit** (modify employee info) – like roles, department, address etc. **Remove** (delete employee) employee data from system. Along with add, edit and remove – system should display all employee summaries (**View**) and must support **search** facility.

The system will first develop using core java only – where employee data will be store as a Collection (List, set or Map). For user interaction, system will use Scanner API.

Later on data will be store on My SQL database; system will use JDBC for the same.

In Phase three –EMS will become web based application, following MVC design pattern. Here the application will be develop in JSP – Servlet.

Macro level Operations/offerings:

1. Create employee (capture employee personal and profession info)
2. Modify employee info
3. Remove employee
4. View all employee (employee summary)
5. Search employee by kin Id, email Id (user id), name

MODULE LIST and MODULE DETAILS

CREATE EMPLOYEE

Following info need to capture

- Name
- Kin Id (must be unique)
- Email id (must be unique and valid)
- Phone no
- Address
- Date of birth
- Date of Joining
- Department (to be select from given list)
- Project (to be select from given list of projects of selected departments)
- Roles (to be select from given list)

SEARCH EMPLOYEE

User should be able to search an employee by kin id, email id and name. User may use any one, two or all the three parameters to search an employee.

MODIFY EMPLOYEE

Search (by kin id, user id or name) or select (from employee summary) an employee and modify the info of employee. System should show existing data/info of employee and should support modify of one, more or all info.

REMOVE EMPLOYEE

Search (by kin id, user id or name) or select (from employee summary) an employee and remove the employee. System should ask for confirmation and on confirmation the data will be removed.

EMPLOYEE SUMMARY (VIEW)

System should show (display) employee list in a tabular format (one row for each employee, and columns for employee info). It is not required to show all the data of employee in table; only important info likes – Kin id, email id, name, role, department, project is needs to show in table.

Constrains

- Proper validation is required (specially on Kin id, email id, name, and on dates) System must check uniqueness on kin id and email id
- System must show appropriate massage on all activity (whether activity is successful or failure)
- User must have proper menu to select the activity (create, modify, search, view, remove) that user want to perform.

DATA BASE

Tables (domain object – for phase 1, since there is no data base)

Table Name – EMPLOYEE		
Field name	**Constrains**	**Data type**
Employee id	Primary key (system generated)	Number
Name		Text
Kin Id	Unique	Text
Email id	Unique	Text
Phone no		Number
Date of birth		Date
Date of Joining		Date
Address		Text
Department id	Foreign Key	Number
Project id	Foreign Key	Number
Roles id	Foreign Key	Number

Table Name – DEPARTMENT		
Field name	**Constrains**	**Data type**
Department id	Primary key (system generated)	Number
Name		Text
Description		Text

Table Name – PROJECT		
Field name	**Constrains**	**Data type**
Project id	Primary key (system generated)	Number
Name		Text
Description		Text
Department id	Foreign Key	Number

Table Name – ROLE		
Field name	**Constrains**	**Data type**
Role id	Primary key (system generated)	Number
Name		Text
Description		Text

PACKAGES AND THEIR REQUIREMENT

In Phase 1 (completely developed in core Java)

Layer	Package Name	Class/interfaces	Purpose
View	com.flp.ems.view	BootClass Will have two method – p.s.v.main (starting point of application – it will call other method) menuSelection	Main method start point of application; System will show menu to user and user will select menu (using **Scanner API**) menuSelection method – will use scanner to interact with user and switch to direct the user instruction to appropriate method of UserInteraction class
		UserInteraction (with 5 method AddEmployee, ModifyEmployee, RemoveEmployee, SearchEmployee, getAllEmployee)	User will enter/view the data based on activity type (using **Scanner API**). For Create and Modify - System will capture date (validate it) and create a HashMap and pass it to next Layer For Search, View and Remove – capture the employee id on which activity needs to perform and pass it to next layer
Service	com.flp.ems.service	Interface - IEmployeeService (with 5 method AddEmployee, ModifyEmployee, RemoveEmployee, SearchEmployee, getAllEmployee) Classes – EmployeeServiceImpl Which implements the interface	Receive data from UserInteraction class (view layer). For create – instantiate a new employee object; populate it and pass it to next layer (DAO) - here system must check employee uniqueness (kin id, email id) and must avoid duplicity of data For other activity – act appropriately on employee object and pass it to DAO

Dao	com.flp.ems.dao	Interface – IemployeeDao (with 5 method AddEmployee, ModifyEmployee, RemoveEmployee, SearchEmployee, getAllEmployee) Classes – EmployeeDaoImplForList Which implements the interface	System will receive data from service layer and perform appropriate action In case of create – add new employee object into Array List In case of remove – remove the selected employee object from the list In case of modify – replace the selected employee object
N/A	com.flp.ems.domain	This package will have following Classes Employee Role Department Project	POJO (bean) classes – having required properties and getter and setter methods. For Employee class – it will be required to override equals and hashCode method corresponds to Kin Id and email id properties. To avoid duplicity of data.
N/A	com.flp.ems.util	This package is to keep all utility classes – for now there will be only one class Validate	Validate class will have static method with return type boolean, to validate – Name, Kin Id, Phone no, email id, date, etc. User regEx to validate date

In Phase 2 (introduction of JDBC and My-SQL database)

Changes

Only change will be – Now IEmployeeDao interface will be implemented by class EmployeeDaoImplForDB (instead of EmployeeDaoImplForList).

Since, now system will be storing data into database, in place of array list.

In Phase 3 (Convert EMS as web application – using Servlet and JSP)

Changes

- No change in business and persistent layer – but view layer will be changed completely
- No need of BootClass – as now web.xml will be the starting point of application and menu will be available on the index page (welcome page (jsp) of application)
- No need of UserInteraction class – as now Servlet will capture data from user (request.getparameter("parameterName"). The Servlet class will send data to next layer in a similar way (as userInteraction class), i.e. in case of For Create and Modify - System will capture date (validate it) and create a HashMap and pass it to next Layer
- Also now the Validate class will not be in use; all fields will be validated at client side using JavaScript

TECHNOLOGY AND ARCHITECTURE

TECHNICAL SPECIFICATIONS - Software Requirement:

- Spring Source Framework 2.5.2
- Tomcat Server 6
- My SQL 5.0 (From phase 2)
- JDK 1.6

PHASE WISE TECHNOLOGY FOCUS -

Phase		
1	Only Core Java	Special attention on – Collection API Scanner Calender RegEx Proper package structuring
2	Database	JDBC My SQL
3	Web based	HTML (front end) Java Script (Specially for validation) JSP Servlet
4	Spring MVC based	Spring MVC IoC Spring JDBC

ARCHITECTURE – Layers