JEE
Mini Project
Survey Management System

Document Control

Reference No:			
Security Classification:	**Internal**		
Issue Date:			
Author(s):	**Name**		**Title**
	Rathnajothi		Senior Executive – T & D
Reviewer(s):	**Name**		**Title**
	Training Team		
Issuer(s):	**Name**		**Title**
Issuer Signature(s):			
Distribution:	**Name**		**Title**
	Team		All Designations in Training
Document History:	**Date**	**Revision**	**Change**
	01-08-2014	0.01D	Initial draft
	29-06-2015	0.02D	Changes in Miniproject document content as per the upgraded courses

Table of Contents

1 INTRODUCTION

This document outlines a mini project for the J2EE LOT. The project is to develop an Online Survey Management System for an organization. This document contains the work flow of the system and gives guidelines on how to build the functionality gradually in each of the course modules of the J2EE LOT.

1.1 SETUP CHECKLIST FOR MINI PROJECT

Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows 95, 98, or NT 4.0, 2k, XP, Windows 7
- Memory: 32MB of RAM (64MB or more recommended)
- Internet Explorer 6.0 or higher
- Oracle 9i client and access to oracle 9i server
- JDK 8
- Eclipse Luna
- JUnit 4.0, Maven
- WildFly

1.2 INSTRUCTIONS

- The code modules in the mini project should follow all the coding standards.
- Create a directory by your name in drive **<drive>**. In this directory, create a subdirectory **MiniProject**. Store your Project here.
- You can refer to your course material.
- You may also look up the help provided in the java docs and documentation provided with WildFly.
- The total time required to complete this mini project is 50 hrs.
- Since this project work will span over couple of months, you will need to take care of maintaining the code

2 PROBLEM STATEMENT

2.1 OBJECTIVE

Development of an online Survey Management System.

2.2 ABSTRACT OF THE PROJECT

This project is aimed at developing Online Survey Management System. This is an Intranet based application that can be accessed throughout the organization and this is a web based application that can be accessed over the web. This system can be used to create the survey questionnaire, edit old survey (only when not distributed), distribute survey, review survey, display list of responded survey, view the list of pending surveys to be responding, and respond for the survey. This is an integrated system that contains the surveyor, the respondent component and the admin (both surveyor and respondent) component.

2.3 FUNCTIONAL COMPONENTS OF THE PROJECT

Following is a list of functionalities of the system. Wherever, the description of functionality is not adequate, you can make appropriate assumptions and proceed.

There are three categories of people who would access the system viz. Admin (both surveyor and respondent), Surveyor & Respondent. Each one of them would have some exclusive privileges (for e.g. Surveyor can create the survey questionnaire, edit old survey (only when survey is not distributed), distribute survey and review survey.)

1. Surveyor should be able to
    - Login to the system using his/her credentials
    - Create the survey questionnaire by filling the form field's values for survey title, description, questions. Atleast one question should be mentioned. Questions can be of 4 type as mentioned below:
        i. Question that have choices and more than one option.
        ii. Question that have choices and only one correct option.
        iii. Question that has a one line answer (max of 250 characters).
        iv. Question that has a descriptive answer (max of 4000 characters).
    - Edit the survey details like description and questions based on survey title only if survey is not distributed.
    - Distribute an existing survey to a list of existing respondents.
    - Review the distributed survey responses by viewing the number of responds received against the number of distribution & View the list of pending surveys.
2. The Respondents should be able to

- Login into the system using his/her credentials.
- Respond for a survey (from a list) by answering the questionnaires.
- View list of responded survey.

3. The Admin should be able to do all the activities of both surveyor and respondents.

2.4 TECHNOLOGY USED:

- *Front End & Web Components:–*
    1. HTML/JavaScript
    2. Servlets
    3. JSP
- *Business Logic Components and Services :-*
    1. Java Beans
- *Application Servers :-*
    1. WildFly
- *Databases:-*
    1. Oracle 9i

3 IMPLEMENTATION IN J2EE LOT

3.1 SUMMARY OF THE FUNCTIONALITY TO BE BUILT:

The participants need to develop the Online Survey Management System by building the functionality incrementally in each of the course modules of J2EE LOT.

Sr. No	Course	Duration (in PDs)	No. of Saturdays	Functionality to be built
1	Programming Foundation with Pseudo code	3	1	Analyze the given case study
2	Introduction to Software Engineering	0.5		Analyze the Case study using SDLC phases.
3	Web Basics (HTML 5,CSS 3, JavaScript, XML)	4.5	1	Developing prototype i.e. developing screens/web pages in HTML and client side validation in JavaScript.
4	Oracle Basics	4	1	Creating relevant database tables
5	OOP & UML	1.5	1	Creating relevant Use case and class diagrams
	Programming Foundation with Pseudo code + Web Basics +Oracle Basics +OOP & UML Test	1		
	Core Java 8 & Development Tools (Junit, Log4j)	10	2	Developing Business components (java classes). Coding for test classes & testing the functionality using JUnit
	Core Java 8 + Dev Tools + OOP/UML Test	1		
8	Servlets	3.5	2	Developing the web application using the prototypes. Converting the HTML web pages to jsp pages and java classes (business components) to java beans. Integrating jsp web pages with business components to complete the entire functionality. Building the web applications component using MAVEN build script.
9	JSP	2		
10	Developer Workbench (PMD, MAVEN)	1		
11	Servlets + JSP + Dev Workbench Test	1		
12	Basic Spring 4.0	5	1	

13	Basic Spring Test	1		Prepare document for presentation.
14	Mini Project presentation	1		

3.2 GUIDELINES ON THE FUNCTIONALITY TO BE BUILT:

The functionality and components to be built in each of the course modules of J2EE LOT is as follows:

1. **Course: HTML, JavaScript (Duration: 10 hours)**
   a. Develop the following screens:

      i. Home page screen: Home page for the online Survey Management System which provides a link for the login page.
      ii. Login Screen: Allows the valid admin/surveyor/respondent to logon to the system and display the Main option screen.
      iii. Main option screen: For Surveyor, this screen will display links for create survey questionnaire, edit survey, distribute survey and review survey responses. For Respondent, this screen will display links for respond for a survey and view all the responded survey details. For admin, this screen will display links for all the options viewable for both surveyor & Respondent.
      iv. Create survey questionnaire screen: This screen allows the surveyor to create survey questionnaire about specific topic.
      v. Edit Survey screen: This screen allows the surveyor to edit the survey details based on title.
      vi. Distribute Survey screen: This screen allows the surveyor to distribute the list of surveys to an existing respondents.
      vii. Review Responded survey screen: This screen allows the surveyor to review the responded / view list of pending survey details.
      viii. Respond survey screen: This screen allows the respondent to response for the survey by answering the questions which was distributed for him/her.
      ix. View all responded survey screen: This screen allows the respondents to view all the existing responded survey details.
      x. Navigate to home page at any point of time
      xi. Logoff from the application at any point of time

   b. In this course you need to develop the user interface using HTML and document the flow of your application including the images of html page in a word document. The screens/web pages should include the fields as per the functionality mentioned above. Also, include client-side validations using JavaScript in each of these screens

2. **Course: Oracle (Duration: 5 hours)**

a. Create the following database tables:
   i. User_Master: This will contain the list of valid users.
   ii. Survey_Master: This contains the list of survey details.
   iii. Survey_Distribution: This will contain the details of survey distributed.
   iv. Survey_Questions: This will contain the details of survey questionnaire.
   v. Survey_Respondent: This contains the list of respondents to whom the surveys are distributed.
   vi. Survey_Respondent_answer: This contains the list of respondent's response details for the survey questionnaires.
   vii. Survey_Question_options: This contains the list of respondent's choices for the questions will be stored.

b. The structure of the above listed tables is as follows:

   i. **User_Master:** User_ID NUMBER(6), User_Password VARCHAR2(20), First_Name VARCHAR2(50), Last_Name VARCHAR2(50), User_Type VARCHAR2(2)
   For Surveyor, respondent and admin assume that the users are already added to the system.

   ii. **Survey_Master:** Survey_ID NUMBER(6), Survey_Title VARCHAR2(50), Survey_Description VARCHAR2(100), User_ID NUMBER(6)

   iii. **Survey_Distribution:** Distribution_ID NUMBER(6), Survey_ID NUMBER(6), Distributed_Date TIMESTAMP(6)

   iv. **Survey_Question_Details:** Question_ID NUMBER(6), Survey_ID NUMBER(6),Question_Text VARCHAR2(500), Question_Type NUMBER(1)

   v. **Survey_Question_Options:** Option_ID NUMBER(6), Question_ID NUMBER(6), Option_Description VARCHAR2(100)

   vi. **Survey_Respondent_Relationship:** Distribution_ID NUMBER(6), User_ID NUMBER(6),Response_Status NUMBER(1)

   vii. **Survey_Respondents_Answers:** User_ID NUMBER(6), Distribution_ID NUMBER(6), Question_ID NUMBER(6), Option_ID NUMBER(6), Answer_Text VARCHAR2(4000)

Note: You may add/normalize/denormalize the tables if your application demands it.

3. **Course: OOP & UML (Duration: 5 hours)**

a. Develop relevant Use case and Class diagrams for the application.

4. **Course: Core Java 8 + Developer Tools (Duration: 14 hours)**

a. Develop business components (java classes) for the following functionality:

i. Authentication Service (on Login): This component will verify if the user who is trying to access the system is a valid user. This verification is as against the valid users listed in the User_Master table.
ii. Surveyor Service: This component will allow the Surveyor to create the survey questionnaire, Edit survey, distribute survey and review the survey responses.
iii. Respondent Service: This component will allow the Respondents to respond for the survey and view the responded surveys list.
iv. XMLAuthentication Service (xml data store): Develop a component to replace the class Authentication Service that does user verification to perform authentication against an XML data store (i.e. the user details are in xml file instead of database). The component should be easily pluggable.

b. Develop test classes for testing the following functionality and Test the application using JUnit.
i. Login
ii. Create the survey questionnaire
iii. Edit survey
c. Configure Logger to log the status of an application

5. **Course: Servlets + JSP + Developer Workbench (Duration: 14 hours)**

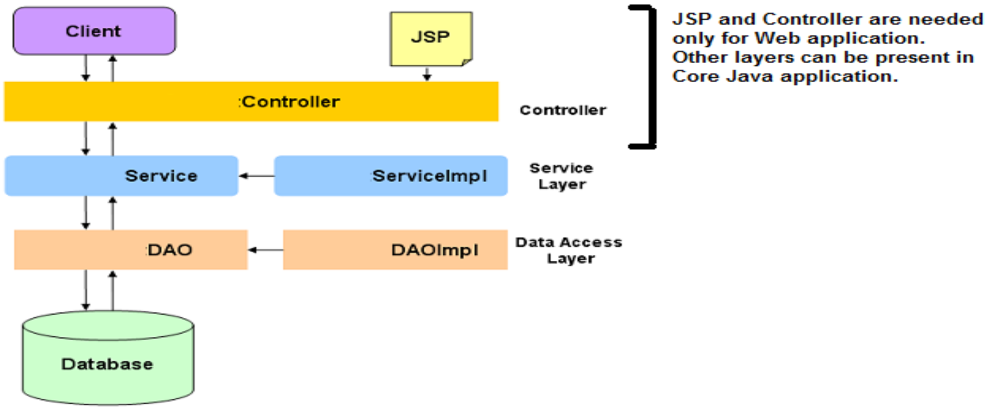
a. Convert all the screens developed in HTML to JSP.
b. Convert all the java classes (business components) created in Java module to Java beans
c. Integrate all screens (JSP pages) with business components (java beans) to complete the entire functionality
d. Configure the DataSource and modify the data access classes to use DataSource object configured.
e. Use https for security throughout the pages so that the valid users can only access the application**.**
f. Develop Logger ServletFilter to log status of an application
g. Build the web component using MAVEN

6. **Documentation (Duration: 2 hours)**

a. Project Documentation: Document your project details (Duration: 1 hour 30 min’s).

b. Project submission: Submit your project with all the artifacts including the test cases & documentation (Duration: 30 min's).

Application Architecture: Discuss this with your mentor on regular basis.


Client
JSP
Controller
Controller
Service
ServiceImpl
Service Layer
DAO
DAOImpl
Data Access Layer
Database
JSP and Controller are needed only for Web application. Other layers can be present in Core Java application.

3.3 EVALUATION AND ASSESSMENT PARAMETERS:

This mini project will be done in groups of 7 members. Each group will identify a Team Lead who will decide which team member will code for which functionality. This project shall be evaluated at the end of spring module.

Evaluation Criteria (out of 100):

Look and Feel of Web pages	**05**
Client-side and server-side validation	**10**
Code Documentation and using coding standards	**10**
Overall Business logic. This includes • Usage of Logging API (log4j)	**25**
Usage of Maven to build project	**5**
Good amount of appropriate dataset to showcase project completely	**5**
Appropriate test cases using Junit 4.0	**5**
Using MVC architecture and clean encapsulation of business logic in appropriate components. Judicious use of java beans, cleaner looks to JSP	**35**