

Case Study Rubrics

Modules	Deliverables	Estimated % Weight
Database	<ul style="list-style-type: none">• Database (RDBMS) – This deliverable includes creating a database that reflects the website you have decided to create• Schema diagram: a visual representation of your database• RDBMS file (DDL.sql): data definition language used to create the database	11.00%
Core Java	<ul style="list-style-type: none">• Models: Java classes that represent an entity and are used to transfer data related to an entity, create multiple queries, and represent the database as an object-oriented model• Service Class or Data Access Objects (DAO): Java classes composed of one or more functions and have direct access to the database by using JPA persistent object. Each function in a DAO class interacts with the database differently.• Custom Exceptions: Java classes that allow you to describe an error while the application is running.• Utilizes Java classes with constant variables (Variables that never change from their initial value). The value of these variables can be requested parameters, SQLs queries used in the DAO, name of HTML pages, or URL patterns to forward a request to.• Utilizes when appropriate Lambda Expressions	25.00%

JPA	<ul style="list-style-type: none"> • Persistence.xml/hibernate.cfg.xml: This file configures the Java classes that are going to be using JPA to interact with the database. • Persistence Java Class: A static class that allows the application to create a persistent object which can be used to interact with the database. 	12.00%
Front-end Development	<ul style="list-style-type: none"> • HTML5: Use HTML for static and dynamic pages and markup the structure of every page. • CSS3: Use CSS3 to style your HTML pages and take into consideration the knowledge acquired from the visual design lessons. • Use Bootstrap for typography, forms, buttons, tables, navigation, modals, image carousels • JavaScript: Use JavaScript to give behavior to your site according to the users' actions. 	10.00%
Junit	<ul style="list-style-type: none"> • Junit: A Java framework responsible for performing unit testing against every DAO class available. There should be a test class for every DAO, and inside the test classes, there should be at least one test case for every function inside the DAO classes. When using JUnit, use the following functionalities: Suite classes, Runner, Feature life cycle, Test, Parameterized classes, Java Hamcrest library. 	12.00%
Spring MVC/Spring Boot	<ul style="list-style-type: none"> • Spring MVC/Spring Boot: Responsible for responding to a request made by the user. This can be login in, registration, etc. When using Spring MVC/Spring Boot, use at least the following functionalities: different types of session management, annotation-based controller, exception handling, input field validation, models, model attributes. • Use of server-side Java template engine, i.e., Thymeleaf or JSP • Use of Spring Data JPA • Use of Spring Security • Use of Transaction logging and request/response logging (write log to a file) 	25.00%

Presentation	<ul style="list-style-type: none"> • Create a short presentation about your application • Highlight the business use cases of your presentation • Highlight how your application works from the technical perspective (high level) • Highlight what you have learned from this case study development • what additional features could be added in the future 	5.00%
Project Management (Extra Credit)	<ul style="list-style-type: none"> • Make use of SDLC/STLC (V-Model) • Perform Requirements Analysis • Adhere Agile Principles, Scrum Framework • Perform stand-ups sessions (with an instructor or teammates) and other Agile frameworks when possible • Successfully track project using JIRA 	5.00%