|  |  |
| --- | --- |
| Background | Explore the range of technologies, methodologies and supporting tools that encapsulate all core modules covered during training. |
| Objectives | Create a fully functioning to-do list web application. |
| Scope | * Code fully integrated into a Version Control System using the feature-branch model: **master/dev/multiple features.** * A project management board with full expansion on user stories, acceptance criteria and tasks needed to complete the project. • A risk assessment which outlines the issues and risks faced during the project timeframe. * A relational database, locally or within the Cloud, which is used to persist data for the project. This database must contain at least **two entities**, with their relationships modelled using an ERD. * A functional application ‘back-end’, written in a suitable framework of the language covered in training, which meets the requirements set on your Kanban board. * A build of your application, including any dependencies it might need, produced using an integrated build tool. * Code ran through a static analysis tool, with relevant refactoring of your code accordingly to reduce code smells, bugs, and vulnerabilities. * A functional ‘front-end’ website which connects to your back-end API. * Fully designed test suites for the application you are creating, as well as automated tests for validation of the application. You should aim to reach the industry-standard of **80%** test coverage through **unit, integration, and user-acceptance** testing. |
| Constraints | * **Version Control System:** Git * **Source Code Management:** GitHub * **Kanban Board:** Jira * **Database Management System:** MySQL Server 5.7 (local or GCP instance) * **Back-End Programming Language:** Java * **API Development Platform:** Spring * **Front-End Web Technologies:** HTML, CSS, JavaScript * **Build Tool:** Maven * **Static Analysis:** SonarQube * **Unit & Integration Testing:** JUnit, Mockito * **User-Acceptance Testing:** Selenium |
| Assumptions |  |
| Risks | * Shortage of time. * Self-management issues. * Maintenance of cloud services. * Required line coverage is not met. |
| Deliverables | * Codebase * Unit tests (80% coverage) * Project management board * Git repository * Working application * Documentation * Presentation (15 mins) |