BOSCARD

Background	**Our Client has provided a music library web application, Choonz. The team has been tasked with augmenting this website, which should present information about songs, albums, artists, playlists, and genres. There are various features which the Client has requested; some are "essential" while others are "desirable". Our Client has insisted that a full test suite for the application must be implemented.**
Objective	To improve upon, and provide a full test suite for, a full-stack Web application suitable for a given client specification, implementing a full test suite to industry standards.
Scope	• The team must demonstrate full commitment to an Agile approach throughout the project, including **daily stand-up meetings**. The Product Owner may be present during these. • Currently, the entire application is not imported into version control. Our Client requests that the existing codebase is **fully integrated into a central repository** prior to the start of any sprints. • The team is required to utilise the **feature-branch model**. • It is recommended that the team regularly builds **project releases** and implements **hotfixes**. • Our Client requires a paper-trail, and therefore expects to see a **project management board** with full expansion on user stories, acceptance criteria and prioritisation. • Our Client also requests that a **risk assessment** is undertaken prior to the first sprint. • Our client would like to see **significant augmenting of the existing website back-end**, provided that most existing functionality remains intact. • Our Client would like to see a **complete oversite front-end**. Any changes to the existing format are permitted, including starting completely afresh. • The training team recommends periodically **running the codebase through a static analysis tool**, with relevant refactoring of your code accordingly to reduce code smells, bugs, and vulnerabilities. • Fully designed test suites – **unit, integration, user-acceptance, functional, and non-functional** – for the improved website. You should aim to reach the industry-standard of **80%** test coverage in the backend.

Constraints	• **Version Control System**: Git
	• **Source Code Management:** GitHub
	• **Kanban Board**: Jira
	• **Back-End Programming Language**: Java
	• **API Development Platform**: Spring
	• **Front-End Web Technologies:** HTML, CSS, 'vanilla' JavaScript
	• **Build Tool:** Maven
	• **Static Analysis:** SonarQube
	• **Testing Frameworks:**
	  • JUnit
	  • Mockito
	  • Selenium
	  • Cucumber
	  • Gherkin
	  • JMeter

Assumptions	• The original source requires refactoring. • A remote, cloud-based database is to be used in production, rather than a local, in-memory database.
Risks	• COVID-19 • Lack of time • Cloud service issues, namely GCP and Github • Loss of data due to versioning faults • Unreliable application

Deliverables	• Full working application
	• Version control with central repository
	• Full test suite:
	  • Unit, integration, user-acceptance, functional, and non-functional
	• Documentation:
	  • Project management board
	  • Risk assessment
	  • BOSCARD
	  • Diagrams
	• Presentation (30 minutes)