Specialism Project Specification (SDET):

Client Web Application Testing & Deployment

**Last revision:** October 2020

CONTENTS

[Introduction 3](#_Toc54189872)

[Objective 3](#_Toc54189873)

[Scope 4](#_Toc54189874)

[Constraints 5](#_Toc54189875)

[Deliverable 5](#_Toc54189876)

[Client Requirements 6](#_Toc54189877)

[General 6](#_Toc54189878)

[User Home Screen 6](#_Toc54189879)

[Albums 6](#_Toc54189880)

[Artists 6](#_Toc54189881)

[Tracks 6](#_Toc54189882)

[Genres 6](#_Toc54189883)

[Playlists 6](#_Toc54189884)

[Non-Functional Testing 6](#_Toc54189885)

# Introduction

The purpose of this document is to outline the group project specification that you will be working on during the final weeks of training. This project will encapsulate concepts from all core training modules, with a significant focus on testing.

# Objective

The overall objective of the project is the following:

**To improve upon, and provide a full test suite for, a full-stack Web application suitable for a given client specification – with utilisation of supporting tools, methodologies and technologies that encapsulate all modules covered during training.**

Specifically, you are required to augment a pre-existing Web application according to our Client’s specification:

* Our Client has provided a music library web application, **Choonz**.
* Your 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”. Prioritisation of these features is key to the success of your team.
* Our Client has insisted that a full test suite for the application must be implemented, covering the following key areas:
  + **Unit Testing** of the website back-end
  + **Integration Testing** of the website back-end
  + **User-Acceptance Testing** of the website front-end
  + A **Non-Functional Testing** suite
  + A **Functional Testing** suite
* Our Client has set the deadline to be **Friday Week 12**. We have time for:
  + A **sprint zero** to be undertaken at the end of **Week 10**
  + **Two 5-day Sprints** to be undertaken during **Weeks 11 and 12**

As with the individual projects, you are broadly expected to use the languages and technologies covered during training to complete this project.

**If you wish to use any technologies which have not been covered as part of your training, you must consult your trainer first.**

You must plan the approach you will take to complete this project using the design techniques you have learned.

# Scope

The scope of the project is as follows:

* The team must demonstrate full commitment to an Agile approach throughout the project, including **daily stand-up meetings**. A trainer, acting as 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.
* The training team would like to see **significant augmenting of the existing website back-end**. Provided that you keep most existing functionality intact, you may refactor, edit, add, or remove parts of the existing codebase as you see fit.
* Our Client would like to see a **complete overhaul of the existing website 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.

Our Client is mercurial, and thus will likely change their mind over various requirements for the project over time. As such, **the team should ensure that features worked on are prioritised** according to scope, likelihood of completion, and overall importance to the project, **while avoiding time-sinks**.

**Our Client also places significant emphasis on team collaboration. Therefore, it is expected that each team member regularly contributes to all aspects of the project.**

# Constraints

Our Client has set the following **strict** technological constraints, as encountered during the course of your training:

* **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

# Deliverable

The final deliverable for this project is the completed application with full documentation around utilisation of supporting tools. This will require a fully functional application.

You will be required to present your work to our Client, the Product Owner, and your Lead Trainer (these may all be the same person). In addition, at least one other trainer will be present for the final presentation. This will take the form of a presentation of work lasting **approximately 30 minutes**, including a Q&A session.

It is possible that other colleagues from the business will be present for this presentation, including members of the Operations team, departmental heads, other delegates, and, occasionally, real-life clients. If this is scheduled to occur, you will be informed well in advance, so that you may prepare your presentation strategy accordingly.

You will be required to include all supporting documentation for your project within your remote repository at close-of-business (17:30) on the day of presenting your project.

**This is your opportunity to shine in the eyes of potential clients – accordingly, the team is expected to maintain a professional approach.**

# Client Requirements

These requirements have been determined by our Client. You may wish to ask your Product Owner to clarify these:

## General

* Multiple users can sign up to the system
* The styling of the entire site should be consistent
* Users can browse the system without logging in, but won’t be able to CRUD
* It would be nice to be able to search for a specific track, album, or artist

## User Home Screen

* Users can CRUD albums, artists, tracks and genres
* Users should see cards for each playlist on their home screen

## Albums

* Users should view each album on its own page
* The album should contain a list of songs
* The albums page should contain a link to the artist page
* Each track should link to a track page

## Artists

* Users should view each artist on its own page
* The artist should contain a list of albums
* Each album should link to an album page

## Tracks

* Users should view each track on its own page
* The track page should show the name, lyrics, and genre
* The track page should contain links to the relevant album and artist

## Genres

* Users should view each genre on its own page
* The genre should contain a list of tracks
* Each track should link to a track page

## Playlists

* A user can CRUD as many playlists as they like on their home screen
* Users should CRUD their own playlists either by song id, name, or genre

## Non-Functional Testing

* Response times should be <10 milliseconds
* Latency should be <2 seconds at 10000 concurrent users
* Throughput rate should be >300/s
* RAM allocation should be minimal, with few (if any) memory leaks
* The application should be spike-, load-, stress-, and soak-tested