# Technologies to be Used for Testing

## Server Side

### Postman

Postman is a tool used for API Testing with HTTP requests. It is heavily customisable to create a Test Framework to test APIs, which in turn test the Server.

## Web App

### Selenium Web Driver in Java

Selenium will be used to drive the automated tests of the Front End of the web application. This has the advantage the it is OpenSource meaning that it is free to use. It works across a range of web browsers and is heavily customisable using Java.

### TestNG

TestNG is to be used to drive the Selenium tests in Java. It’s a testing Framework that is inspired by Junit but provides more functionality in terms of annotating tests.

### Cucumber

Cucumber can be implemented to provide BDD functionality. This enables the tests to be written in Plain English that would aid the client in understanding what we are testing.

# Design Process

## Server Side

To test the Server Side, the tests will be created in Postman. The server tests run independently from the client side. Postman will send SOAP requests to the server in the same manner that the client would. There’re four key areas that need to be supported: Create, Read, Update and Delete (CRUD).

|  |  |  |
| --- | --- | --- |
|  | SOAP Request Type | Description |
| Create | POST | Used to send requests to the Server to create objects. |
| Read | GET | Used to send requests to the Server to retrieve objects. |
| Update | POST | Used to send requests to the Server to update existing objects. |
| Delete | DELETE | Used to send requests to the Server to delete existing objects. |

A full breakdown of the Tests which Postman will run can be found later in this document.

## Web App

Each webpage on the Application will have a Page Object created for it. The Page Object Model is provided by Selenium to create an Object Repository for Web UI elements. It means that the Web Elements and Methods for them are kept in a separate place to where the Test Methods will be located. This has the advantage of only one reference to where an object is rather than in the Test Methods. This is ideal in case the layout changes, only one reference needs to be updated in one class.

The test Methods will call support functions within the Page Object to enable Selenium to navigate the web page, enter data and retrieve information. The Test Methods will be written in the TestNG framework.

On top of this, Cucumber will be used to provide a platform to define the tests in Plain English. Its support methods will then link in with the Test Methods file. Cucumber is used primarily to liaise with the client, so they can easily see what we are testing.

Cucumber hasn’t been implemented with my code. An example for test\_searchEmployeeByIDValid, the scenario would look like:

***Feature****: Admin Employee UI  
As a Manager  
I want to be able to add, remove and edit employees  
So that I can manage Performance Reviews*

***Scenario****: User wants to Search for an Employee by their ID****When*** *I go to the Admin Employee UI****Then*** *I should see a Search Bar****And*** *be able to enter an Employee ID in the field*

# Postman High Level Tests

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Request Type | Functional Area | Description |
| 1 | POST | Admin | Create a new Employee. |
| 2 | DELETE | Admin | Remove an existing Employee. |
| 3 | POST | Admin | Update an existing Employee. |
| 4 | GET | Admin | View an existing Employee. |
| 5 | POST | Admin | Create a Performance Review. |
| 6 | POST | Admin | Update a Performance Review. |
| 7 | GET | Admin | View a Performance Review. |
| 8 | POST | Admin | Assign Employee to participate in a Performance Review. |
| 9 | GET | Employee | Get a List of Performance Reviews requiring feedback. |
| 10 | POST | Employee | Submit Feedback for a Performance Review. |

# Assumptions

|  |  |
| --- | --- |
| ID | Assumption |
| 1 | The Web App is written in HTML. |
| 2 | The Web App is accessible in Firefox, Chrome and Internet Explorer. |
| 3 | SOAP is used for communication between the Client and the Server. |
| 4 | A SQL Database exists to store the data for this application. |
| 5 | Employees can be searched for either by Name or Employee Number. |
| 6 | The Admin Employee UI is a single window which updates for different contexts: View/Edit. Edit adding additional buttons to Save or Cancel the edit. Also, the text fields will become editable in the Edit view and Read Only in the View context. |
| 7 | The Admin Employee UI has validation on the Text Fields. Any invalid characters or lengths cause the field to be highlighted in red. Hovering over them displays an error message. |
| 8 | The Admin Assign Performance Review UI uses a similar Search Bar to the Admin Employee UI, whereby an Employee can be searched for by their Name or ID. |
| 9 | The Admin Assign Performance Review UI has two fields (Employee to perform review, Employee who the review is for) and a dropdown box (Selects what performance review to do. Finally, there is a button to send out the review. |
| 10 | The Admin Assign Performance Review UI sends out an email to the Employee doing the Review when Submit is clicked. |
| 11 | The Admin Assign Performance Review UI leaves the Submit button greyed out if all the fields haven’t been populated. |
| 12 | When entering an Employee ID on the Admin Assign Performance Review UI, it gets updated to display the Employees Name after clicking out of the box. |
| 13 | A Manager has access to the Admin area, a User does not. |