# Section 1 - Darwin Test Automation Framework

## 1.1 Project

Darwin Test Automation Framework

## 1.2 Description

This design document captures the high level design of Darwin Test automation Framework used of testing the Restful services consumed Darwin’s applications exposed by JsonPlaceholder service.

## 1.3 Revision History

|  |  |  |
| --- | --- | --- |
| **Date** | **Comment** | **Author** |
| 17/12/2021 | Darwin Test Automation Framework – Design V 0.1 | Munuswamy C |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Contents**

[Section 1 - Darwin Test Automation Framework](#_Toc90635021)

[1.1 Project](#_Toc90635022)

[1.2 Description](#_Toc90635023)

[1.3 Revision History](#_Toc90635024)

[Section 2 - Overview](#_Toc90635025)

[2.1 Purpose](#_Toc90635026)

[2.2 Scope](#_Toc90635027)

[2.3 Requirements](#_Toc90635028)

[2.3.1 Estimates](#_Toc90635029)

[2.3.2 Traceability Matrix](#_Toc90635030)

[Section 5 - Software Domain Design](#_Toc90635031)

[5.1 Module Design](#_Toc90635032)

[Section 6 – Data Design](#_Toc90635033)

[Section 8 - Other Interfaces](#_Toc90635034)

[8.1 Interface X](#_Toc90635035)

[Section 9 - Extra Design Features / Outstanding Issues](#_Toc90635036)

[Section 10 – References](#_Toc90635037)

[Section 11 – Glossary](#_Toc90635038)

# Section 2 - Overview

## 2.1 Purpose

Darwin test automation framework is designed to test the Restful services exposed by JsonPlaceholder service provider for building the User information details

## 2.2 Scope

1. Following are the different REST actions that are in scope of this design document .
2. GET
3. POST
4. PUT
5. Test data for testing the testing the Rest API is provided through Microsoft Excel sheet .

## 2.3 Requirements

Following are the High Level Requirements

1. Build a Test automation framework so that I can consume the service available at <https://jsonplaceholder.typicode.com/>
2. Following are the Test Scenarios are in scope of design
   1. Get a list of the following attributes of an existing user:
      1. Name
      2. Email
      3. Address
      4. Phone
      5. Website
   2. Create a new user
   3. Update user details
   4. Search for users by email

### 2.3.1 Estimates

|  |  |  |
| --- | --- | --- |
| **#** | **Description** | **Hrs. Est.** |
| 1 | Utility Module- Excel Reader | X hours |
| 2 | Utility module – Extent Report Generation : | X hours |
| 3 | Utility module Test Data Provider | X hours |
| 4 | POJO classes | X hours |
| 5 | Test Case Module – CreatUser | X hours |
| 6 | Test Case Module- GetAndUpdateUserDetails | X hours |
| 7 | Test Case Module- -SearchUserByEmail | X hours |

### 2.3.2 Traceability Matrix

Cross reference this document with your requirements document and link where you satisfy each requirement

|  |  |
| --- | --- |
| **SRS Requirement** | **SDD Module** |
| Darwin SDET test - Instruction | Section 5.1 , Section 6 |
| Instruction -Get User Attributes | Section 5.1, Section 6 |
| Instruction - Create a new user | Section 5.1 , Section 6 |
| Instruction - Create a new user | Section 5.1 , Section 6 |
| Instruction – Update user | Section 5.1 , Section 6 |
| Instruction – Search User by Email | Section 5.1 , Section 6 |

# Section 5 - Software Domain Design

## 5.1 Module Design

1. **Base Class** : Base Class methods are executed at the beginning of the test execution , it sets up the prerequisites need for running the test . Following are initial setups to be done to run the test cases
   1. Setup the Config data path
   2. Setup the Test data path
   3. initialize the logging
   4. Initialize and Create the Extent Report
   5. Testgn Annotation are used where ever applicable
2. **CreateUser :** 
   1. This class is used to build the User attributes from the given Test data . It Sends user object as body of the POST request and Verifies the Original user Object with the Response body of POST request. If the comparison is not successful it will fail the test case else it will mark the test as passed .
   2. **Comparison Design** : It creates a POJO user Object and send to the server. Response received from the server is converted to POJO user object and compared with the Original POJO object .
3. **GetAndUpdateUserDetailsUserDetails :** 
   1. **GetUserDetails** :This module is used to retrieve the user detail for the given record . GET method is used along with Path Param to specify the record ID. Record ID is provide through Test Data and Data provider object.
   2. **update** : Get Method is used to retrieve the Existing User Record from the server . Once Json messages are retrieved, they are Converted to POJO object . The POJO is updated with user Data from the Excel and PUT Request is sent with Updated POJO Data .

PUT Response is converted to POJO object and PUT Request body and Response Body are compared to verify if the details are updated correctly .

1. **SearchUserByEmail** :
   1. Retrieve the user details through GET Request .
   2. Parse the Json Response .
   3. Retrieve the Email attribute to a List .
   4. Compare the Email list with the Email from the Test Data and Verify the Test scenarios
2. **XLutils** : This is a utility module to read the Test data from the excel sheet . IT has utility methods like Getcelldata , Setcelldata , Getcolcount and Getrowcount to work with Excel sheets
3. **ExtentReprothandler** : This is a uitlity method to initialize and create the reports through the test execution .
4. **TestDataprovider :** This is a data provider class , that gets the data from excel sheet and feeds it to the test methods .
5. **Log4J :** It is used for providing the logging capability to the test framework .
6. **Runner** : It contains a Main method for easy execution of the test cases from command line .

# Section 6 – Data Design

1. Microsoft Excel sheet is used to provide the test date to the automation test suite through Testng data provider .
2. Excel sheet is composed of following sheets
   1. TestDriver Sheet : It provides scenario with TestCase numbers . Test Case Numbers should following sequential number , Numbering cannot be random as these TC numbers are used in other sheets . It also captured the Test Result and Response body for a Quick glance of the report .
   2. Test Case Sheet : Test data for each test scenario is capture is separate Test data sheet . There is Data provider method available to retrieve the data from each Test Data sheet corresponding to a Test scenario .

# Section 8 - Other Interfaces

1. Maven is used for Project Dependency Management through POM
2. Java 8 is as Programming language
3. RestAssured libraries are used for working with REST operation .
4. Log4J is used for logging functionality
5. TestNG is used a Test Frame work .
6. Apache POI is used for Excel operations

## 8.1 Interface X

1. Json Message are used Restful Transactions .

# Section 9 - Extra Design Features / Outstanding Issues

None

# Section 10 – References

None

# Section 11 – Glossary

None