**SOFTWARE ARCHITECTURE DOCUMENT**

***Author: Mohamed Salah***

Agenda

1. Introduction
   1. Purpose
   2. Objective
   3. Scope
2. Architecture Diagram & Data Flow
3. Architecture Explanation
4. Used Technologies and Tools
5. Configuration And Run Steps

1. INTRODUCTION

This document provides a high level overview and explains the architecture of the automation done for Facebook application

The document defines an overview on the architecture chosen and the covered scenarios and how it helps in testing these scenarios and maintaining the automation code.

1.1 PURPOSE

The Software Architecture Document (SAD) provides a comprehensive architectural overview of the test automation .

1.2 OBJECTIVE

Our system aims to automate the regression suite and the frequently executed test cases also automation helps in the following

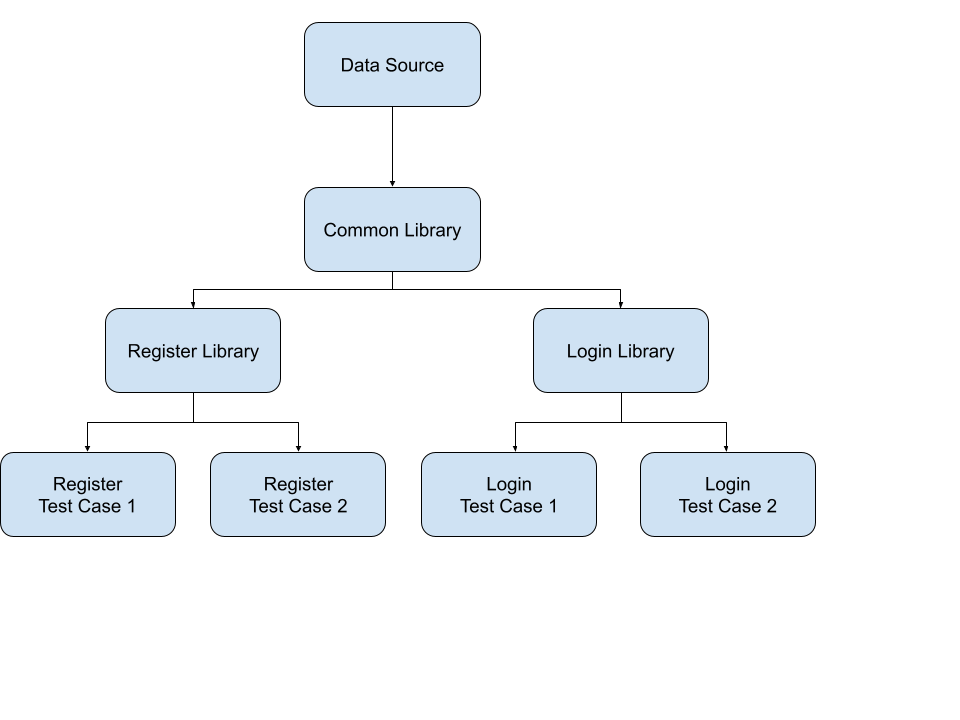
* Reduce Cost
* Save Time
* Foundation for CI and DevOps
* Accuracy and Reliability
* Increased Confidence
* Measure Quality Metrics

1. 3 SCOPE

The scope of this document is to show the following

* Architecture Diagram & Data Flow
* Architecture Explanation
* Used Technologies and Tools
* Configuration And Run Steps

2.0 Architecture Design & Data Flow



In this architecture I have made reusable components to be used in all applications and called it a library. This approach will help me in code maintenance as any change in the application will modify it from my side to only one place and it will be reflected to the whole application automatically. I have made 3 libraries

* **Common Library**: It will be used by all the applications whatever the module being tested also it will feed the other libraries. In this library we will be handling also the data management layer
  + **Capabilities Class**: Setting the application configuration before running it
  + **DataDriven Class**: Read the data from Mysql database and then frommating it to Object array to send it to TestNG
* **Module Library**: It will be used by specific module in the application, now we have 1 Module library
  + **Page Object Module**: It will be containing all the elements ID used for automating Facebook application
* **Test cases Classes**: each class will be testing a test case separately here I am applying the single responsibility principle (SRP)



The data flow Diagram explains how the data flow from the data source till it reaches the test case by the following steps. The importance of this approach is that we can run the same test case with different data so we can try the positive scenario and negative scenario with the same test case which will save a lot of time doubling the test cases. **(It the same way used for Jira Automation Testing the only additional thing is password encryption)**

* Create the data source and fill it
* Read it in the common Library
* prepare the data and do some transformation
* Use it in the test cases through TestNG

4.0 Used Technologies And Tools

**Used Technologies, Libraries And Tools:**

**Selenium Java**: Selenium is a freely distributed open source web application UI Testing framework. Selenium allows web application testing and supports automation testing on different web browsers

**Apache Maven**: Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

**TestNG**: TestNG is an automation testing framework in which NG stands for "Next Generation". TestNG is inspired from JUnit which uses the annotations (@). TestNG overcomes the disadvantages of JUnit and is designed to make end-to-end testing easy. Using TestNG, you can generate a proper report, and you can easily come to know how many test cases are passed, failed, and skipped. You can execute the failed test cases separately.

**Mysql**: MySQL is a relational database management system based on SQL.

**Log4J**: Apache Log4j is a Java-based logging utility.

**Eclipse**: Eclipse is an integrated development environment (IDE) used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment.

6.0 Configuration And Running Steps

You can find here in the [readme.md](https://github.com/mohamedsalah939/Mohamed-Salah/blob/master/Downloads/Mohamed-Salah/E2EProject/readme.md) all the configuration and running steps to run this application on other environments