Design and Development of Software Test Automation Framework

Submitted By

Shah Rasesh

13MCEC21

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

December 2014

Design and Development of Software Test Automation Framework

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Shah Rasesh

(13MCEC21)

Guided By

Prof. Jigna Patel

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

December 2014

Certificate

This is to certify that the major project entitled **"Design and Development of Software Test Automation Framework"** submitted by **Shah Rasesh (Roll No: 13MCEC21)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Institute of Technology, Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Jigna Patel
Guide & Assistant Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Prof. Vijay Ukani
Associate Professor,
Coordinator M.Tech - CSE
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

ORACLE®

Oracle India Pvt. Ltd.
India Development Center

Oracle Technology Park
3, Bannerghatta Road
Bangalore - 560 029, India.

CIN:U74899DL1993PTC051764
Phone +91 80 4107 0000
Fax +91 80 2552 6124

May 15, 2015

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr. **Shah Rasesh Dipakbhai,** student of M.Tech (pursuing Institute of Technology, Nirma University) has completed his project with Oracle India Private Limited.

The project was undertaken from June 2, 2014 to May 15, 2015. He worked on "Oracle Retail - Unified Automation Program"

We wish him all the best in his future endeavours.

Yours sincerely
For Oracle India Private Limited.

Ganti V S Sarma
Group Manager - HR Operations

Statement of Originality

I, **Shah Rasesh**, Roll. No. **13MCEC21**, give undertaking that the Major Project entitled "**Design and Development of Software Test Automation Framework**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Prof. Jigna Patel

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Jigna Patel**, Assistant Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr K Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

See that you acknowledge each one who have helped you in the project directly or indirectly.

- Shah Rasesh

13MCEC21

Abstract

Software testing plays a crucial role in assuring software quality. As the complexity and size of software grows, the time and effort required to do manual testing increase. Test automation is becoming more and more popular as the software complexity is increasing and the need to test for each scenario is becoming more and more crucial due to the current rising demand for efficient and fast products.

The objective of this paper is to describe, design and develop of test automation framework. A test automation framework is an integrated system that sets the rules of automation for a specific product. This integrated system contains object detail of application, business function library, general function library, various test data sources and reporting module. These components can be use as building block for developing testscripts. This would in turn increase the efficiency of the testing process and simplifies the automation effort.

Abbreviations

MOM	Merchandise Operations Management
SCM	Supply Chain Management
ISO	Integrated Store Operations
RMS	Retail Merchandising System
WMS	Warehouse Management System
ReSA	Retail Sales Audit
POS	Point Of Service
SIM	Store Inventory Management
RIB	Retail Integration Bus
IT	Integration Testing
SUT	Software Under Test

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Oracle Retail System

Oracle Retail provides a open, complete and integrated suite of business applications, server and storage solutions. Oracle retail solutions empower retailers to optimize operations across mobile, traditional and online commerce channels. Oracle retail has wide range of applications to perform the retail business operation. Oracle Retail Merchandising System is an integrated merchandising solution which enables retailers to better manage, control and perform crucial day-to-day merchandising activities from new product introduction to automated replenishment to financial inventory evaluation. The major applications among this wide range can be classified into 3 categories.

- Merchandise Operations Management
- Supply Chain Management
- Integrated Store Operations

Oracle retail products and there functionalities are as listed below:

- **RMS**

  Retail Merchandising System manages the fundamental processes of a retail business. Helps retailers efficiently maintain foundation Data, organization Hierarchy, merchandise hierarchy and items.

- **SIM**

  Store Inventory Management expedites in-store operations, including item searching, receiving, adjustments, count of stock, transfers between stores. Improves

control of inventory management for reduced carrying costs. Helps save the sale by using real-time cross-store inventory access.

- **WMS**

  Warehouse Management System coordinates the movement of merchandise and information throughout the distribution process. Apply sophisticated, flexible configurations and built-in best practices to efficiently utilize distribution resources and increase visibility across your extended retail supply chain.

- **POS**

  Point Of Service improves customer service and increase profitability. Provides the flexibility, responsiveness, and scalability to meet the needs of the largest retailers, improving customer service and increasing profitability.

- **ReSA**

  Retail Sales Audit ensures reliable and consistent sales data. ReSA validates, processes, clean data and ensure consistent sales information from all sales transactions and from all channels.

- **RIB**

  Retail Integration Bus supports near-real-time messaging with following features:

  - With guaranteed once-only delivery
  - Guaranteed sequential delivery within a family of message
  - Automatic retry regardless of errors
  - Support for fast delivery of retail volumes
  - Multi-threading

  RIB provides infrastructure tuned for the high availability, high volume and high reliability needs of retail.

- **Allocation**

  Oracle retail allocation allow retailer to accurate allocate there product. Having right product at right store is very important for retailer perspective because this lead to increased sales, service level and reduce the inventory cost. Oracle retail

allocation enables retailer to achieve high turnover and increase profitability[5]. Task of Allocation is to prepare allocation plan which contain information regarding which store each item should be kept which lead to increase turnover.

1.2 Testing

Importance of software is increasing and mean while complexity as well as size of software is also increased. If quality of software is not upto satisfying level than it is going to increase human effort as well cost for maintenance of software. Aim of software testing is to assess the quality of the product. Demand for good quality software makes more pressure on testing process.

1.2.1 Type Of Software Testing

Software can be effectively tested by two testing techniques which are described below:
Manual testing is more traditional approach, In which manually execute the testcases which relate with all functional area of product to identify any unexpected behavior or bug and at the end submit the report with the defects which are found during testing process.
Automation testing is very popular technique to reduce time and cost relate with testing activity[1]. This technique take support of automation testing tool for testcase execution and validation process.

1.2.2 Benefits Of Automation Testing

- **Perform tests which would be difficult**

  Some software testing activities are not time and cost effective if we perform it manually. Manual testing is also not able to find some type of bugs which are present in software in effective manner. This tasks can be very well handle by test automation.[2].

- **Regression tests**

  When ever new patch is ready to support new feature or remove old bugs, its very probable that this changes can affect previous functionality of software. To overcome this we need to test all functionality after producing new patches for system. This testing is know as regression testing. For this type of task test

automation can provide very effective solution because we can re-run each test scenario as many time as we want.

- **Consistency & Repeatability**

  Automation testing provide consistency because every time its going to run same way and produce consistence output[2]. If we doing same work with manual testing than it might not be as consistence as automation because manual testing involves human error. Human can miss out one testcase or validation rule which might lead to unable detecting bug.Due to consistence performance of test automation, we can able to re-run every test scenario as many time as we want.

- **Speed**

  Testcasaes can run concurrently on different environments which have different configuration.

- **Reusability**

  Less time require to run automated testcase than manual run. Because of this we are able to get faster feedback.

1.2.3 Automation Testing Process

Automation requires considerable amount of human effort and its also costly to setup initially.Automation aim to reduce manual test execution. To achieve this goal with limited resources, we need to follow specific process while automating testing process. Following are the phases/stages of it[3]. Each one of the stages corresponds to a particular activity.

- **Feasibility Analysis**

  First step is to check if the application can be automated or not. Not all applications can be automated due to its limitations.

- **Appropriate Tool Selection**

  The Next most important step is the selection of tools. It depends on product technology over which its build.Automation provide more promise than manual testing thats why organization adopting automate testing process. Automation test tools can be categories in two way as follow[4].

  - Open source
  - Commercial

- **Select Automation Technique**

  Upon selecting the tool the next activity is to select a suitable technique and develop it. There are various kinds of technique and each technique has its own significance. All automation testing technique and methodologies are explain with its key features in chapter 2.

- **Develop, Execute and Report**

  After Scripts are developed, we needs to execute scripts and at the end needs to submit results with defects.

Chapter 2

Test Automation Framework

Chapter 1 introduced different types to testing, benefits of automation and process of automation testing. The aim of the this chapter is to define requirements for test automation framework and also described various automation framework design with it's merit and demerit.

Set of rules for automation of specific product is set by test automation framework. This integrated system contain object detail of application,reusable components, business function library, general function library, testscript and various test data sources. These component can be use as build block while we are testing application business process. This will simplifies the automation process.

The main benefits of a automation framework of tools, concepts and assumptions that provide support for automation process is very low maintenance cost. If we required to modify the testcase than only testcase should be updated and changes in any other portion of framework are not required. Ideally, No change is required in testscript if application changes.

2.1 Requirements

This section explains required capabilities of automation Framework [2]. These properties can be seen as functional requirements for test automation framework.

- **Executing Tests**

  Framework should set the environment, check all precondition and start execute testscript after execute command is given to it.

- **Control Over Test Execution**

  Framework should be able to support scheduling testscript at specific time or after certain events/precondition are met.

- **Error Handling**

  Test framework should be able to recover from error occur during execution and complete it's given task without any manual intervention.

- **Result Verification**

  Framework should able to provide way which identify deviation between actual and expected output.

- **Debugging**

  Framework should be able to support Debugging by providing pause functionality while testscript are running so that tester can see the state of testscript.

- **Status Of Test**

  Framework should able to provide status after test execution complete. If all the verification step are clear than its should give pass otherwise fail status with proper reason behind failure. Various status with there significant meaning are as listed below[2]:

  - **Pass** This indicates that testcase is passed.
  - **fail** This status indicates that testcase is failed.
  - **Warning** After executing recover scenario if testcase able to pass than we need to log message that after recovering this failure state its passed. These messages are known warning.
  - **Info** These type of message are needed to describe various checkpoint in test-cases like start testcase, validate values and mismatch.

- **Reporting**

  Detail test report contains all the above message. But some time this type of report when we need to glance result like how many testcase are failed and how many are passed. We are not looking for reason behind failure but just statistics than we need to generate another report which summarize the all testcases results.

2.2 Techniques

For reducing cost of overall automation testing process we need to choose right automation framework. Major cost is due to implementation of framework and developing testscripts. For choosing right framework we need to see the merit and demerit of different automation framework and choose which best match for your working environment. Various framework methodologies are as listed below:

- Record and Playback
- Structured & Modular
- Data-driven
- Keyword-driven
- Hybrid

2.2.1 Record and Playback

This is simple technique to automate the testcase. Tester directly interact with Software under test. All the activity perform by tester gets recorded by automation tool. This action stored into testscript and can be replay to perform exact same actions. The generated recorded script consists of a series of testing instructions using the programming language supported by the tool[1].

As shown in Figure 2.1, simple login validation perform by tester on the product and every activity gets recorded by automation tool. These scripts are also known as linear scripts. The problem regarding this technique is its get more complicate and harder to maintain as testcases are get bigger. Linear script generate large code which even harder to maintain[2].

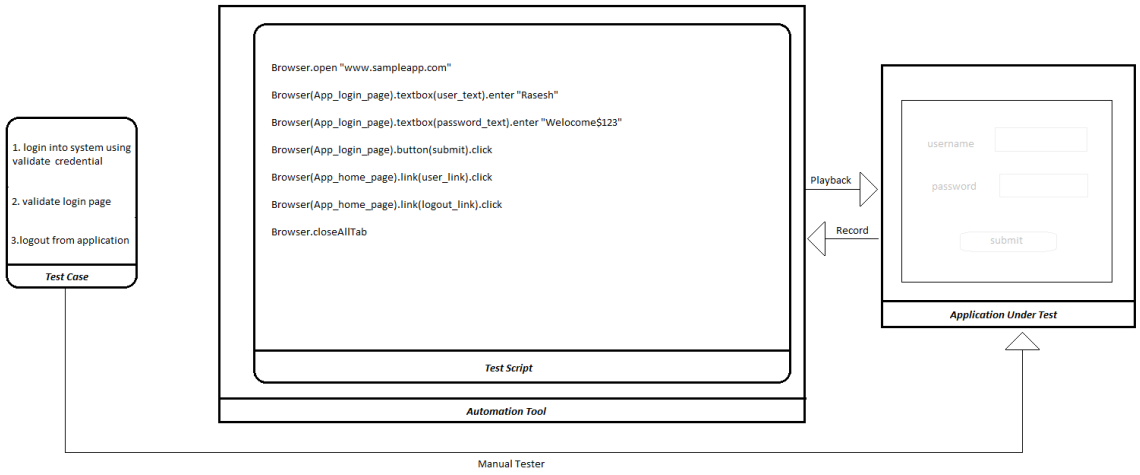

Figure 2.1: Record And Playback

2.2.2 Data Driven

When we recorded testscripts all the data which we have enter were also embedded in scripts. In this technique, data are decoupled at stored in different file by putting placeholder into testscripts. This placeholder are nothing but variable and by changing variable value we are able to run same testscase multiple time. Values for variable can be taken from different datasource.

In this technique rather hardcoding the value which are recorded, we are using placeholder and suppling this placeholder values from external data sources or from datatable. Data is decoupled from code so code can be rerun for different test data. As shown in Figure 2.2 external user datasheet contain users credential. For every user we can test the login scenario by calling same login subscript/function provided different input value.

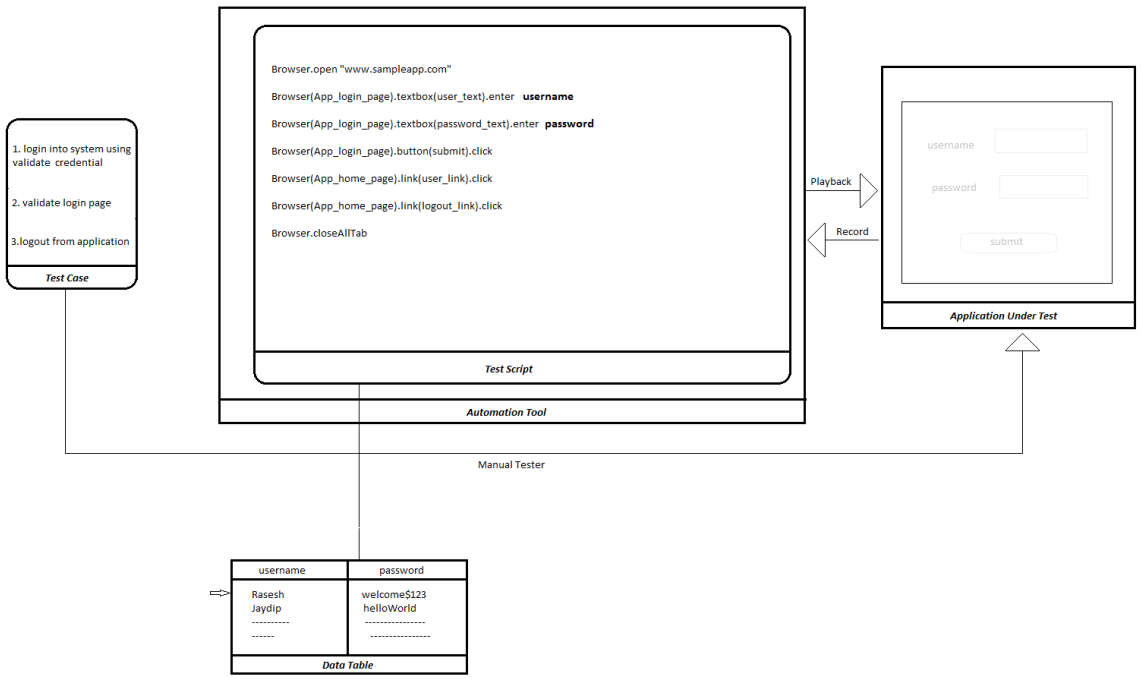

Figure 2.2: Data Driven Framework

2.2.3 Structured & Modular

Script which use structure programming to control the flow of execution are known as structured scripts[1]. In modular scripting technique we are using small script, subscript, action or function to create testscript. Combined of above two approach is know as structure & modular framework. Small subscript which also use structure programing can be reuse to create large testscript.

As shown in Figure 2.3, instead of repeating login function in every testscript we can create one library in which this function are defined and in testscript we just calling this function. This approach not only increase the reusalblity but also reduce the maintenance effort. If application function change than we only need to update functions which relate with updated application functionality. So only small portion of code is updated instead of changing number of different testcases[2].

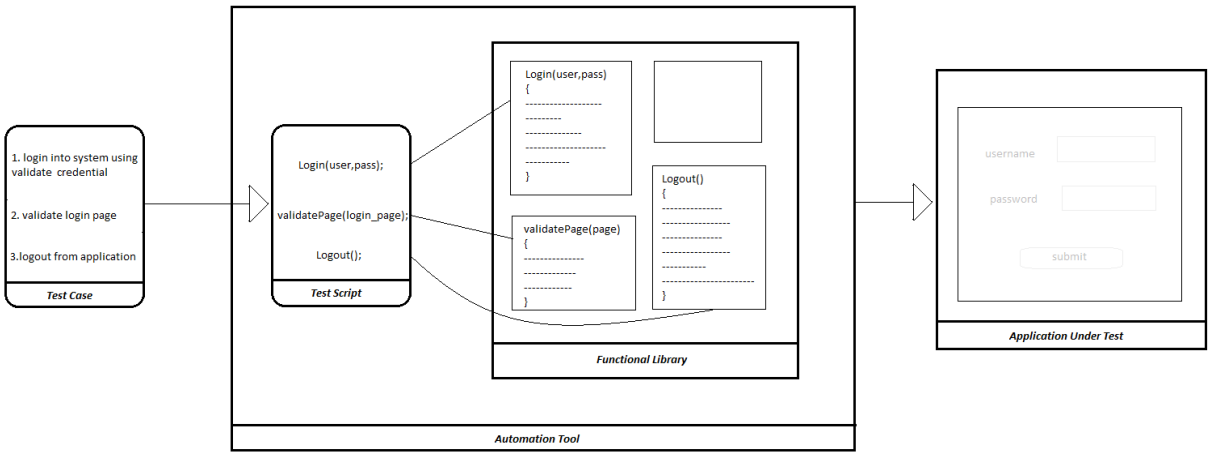

Figure 2.3: Structured Framework

2.2.4 Keyword Driven

Keyword-driven testing is a framework that separates much of the programming work from the actual test steps. In this framework we need to create a set of Keywords and Data Sheets. Keywords perform the generic tasks that can be used across the application and data sheets contain the detailed steps to be carried out for the execution of test case. The keyword concepts of this framework, separates control data from test scripts and hide complexities from users and improve usability. Main advantage of keyword driven testing is that it reduce/eliminate the maintenance activity due change in software under test. It might possible that application object are change or modified but due to this we don't need to modify our testscript because keyword are not change at all, only function relate with that object need to change. This type of technique is very effective when system is migrate to another operating system or changes application object because framework need hardly any changes. As Shown in Figure 2.4 all the keyword is available to tester and tester create testscript or test action table which contains list of actions, object on which action to be perform and relate data which is need to perform this action. Parser interpret every action and find related function from function library and execute the function.

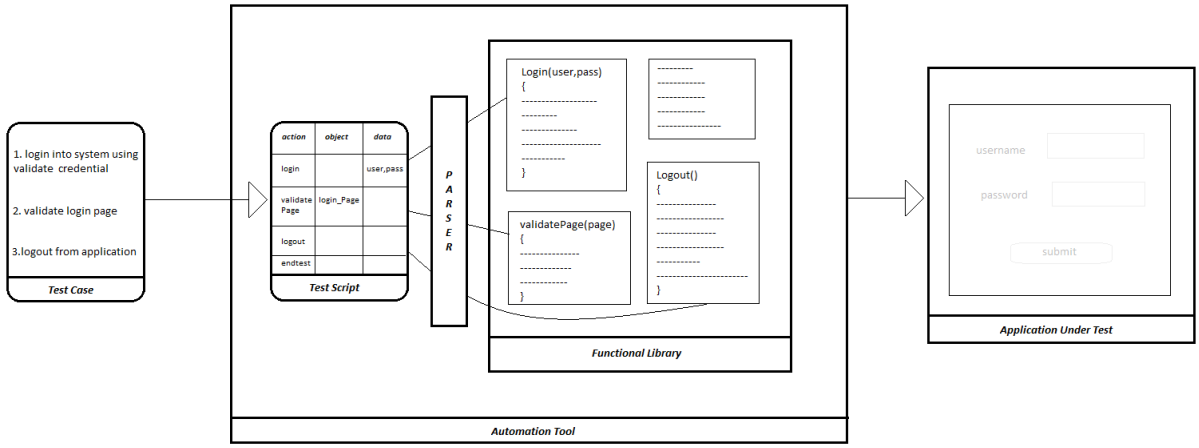

Figure 2.4: Keyword Driven Framework

2.2.5 Hybrid

This Technique combined more than one techniques that's why it named as hybrid automation framework. As show in figure 2.5 data-driven, modular and keyword combined to create new framework. In this framework for each data from data table testscript . Each testscript contain list of actions. This actions are parsed by parser and execute relate function to it. This type of framework takes advantages of all combined frameworks.

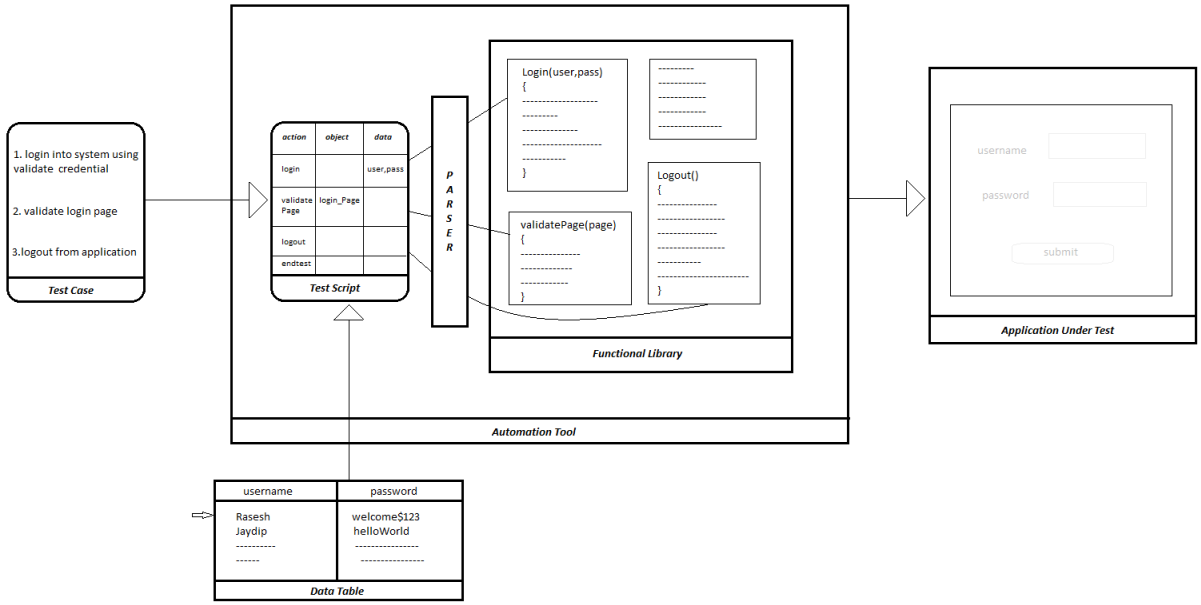

Figure 2.5: Hybrid Framework

2.3 Comparison

Table 2.1: Comparison Of Test Automation Techniques

Feature	**Linear**	**Structure & Modular**	**Data-Driven**	**Keyword-Driven**
Reuse Function	No	No	Yes	Yes
Data Decoupled From scripts	No	No	Yes	Yes
Logical Step separated from scripts	No	No	No	Yes
Use Control Structure	No	Yes	Yes	Yes
Need Programming skills	1	2	3	4
Effort in testscript creation	1	2	3	4
Maintenance Cost	4	3	2	1
Reusablity	1	2	3	4

Chapter 3

Design and Development of Test Automation

In this chapter we are going to cover overview of SUT and also design of various key components of automation framework.

3.1 Software Under Test

Oracle Retail Merchandising System is an integrated merchandising solution which enables retailers to better manage, control and perform crucial day-to-day merchandising activities.

Allocation is one product which help retailer to prepare plan for moving different item from one warehouse to different store such that it maximize the sale as described in Figure 3.1.We are trying to design and develop test automation framework for Allocation application.

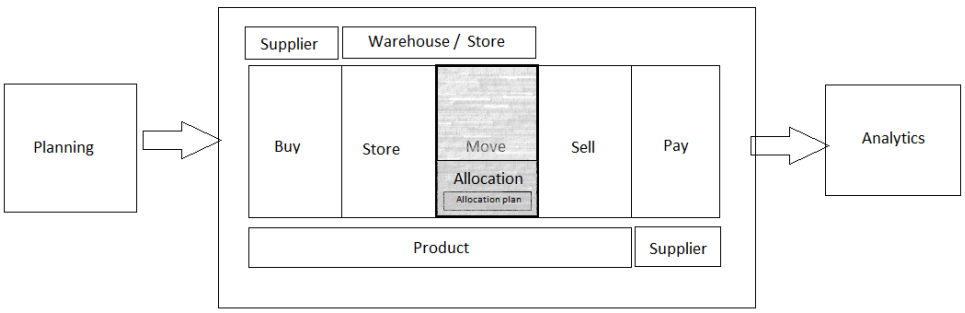

Figure 3.1: Allocation

3.2 Key Components Of Automation Framework

We are using hybrid framework to automate testing process for allocation application.we are trying to build modular framework which also going to support data driven testing..we are going to support data driven testing by allowing tester to run same testscript multiple time with different data provided by datatable or external datasource.In this section,All the component to develop hybrid automation framework are described in Figure 3.2.

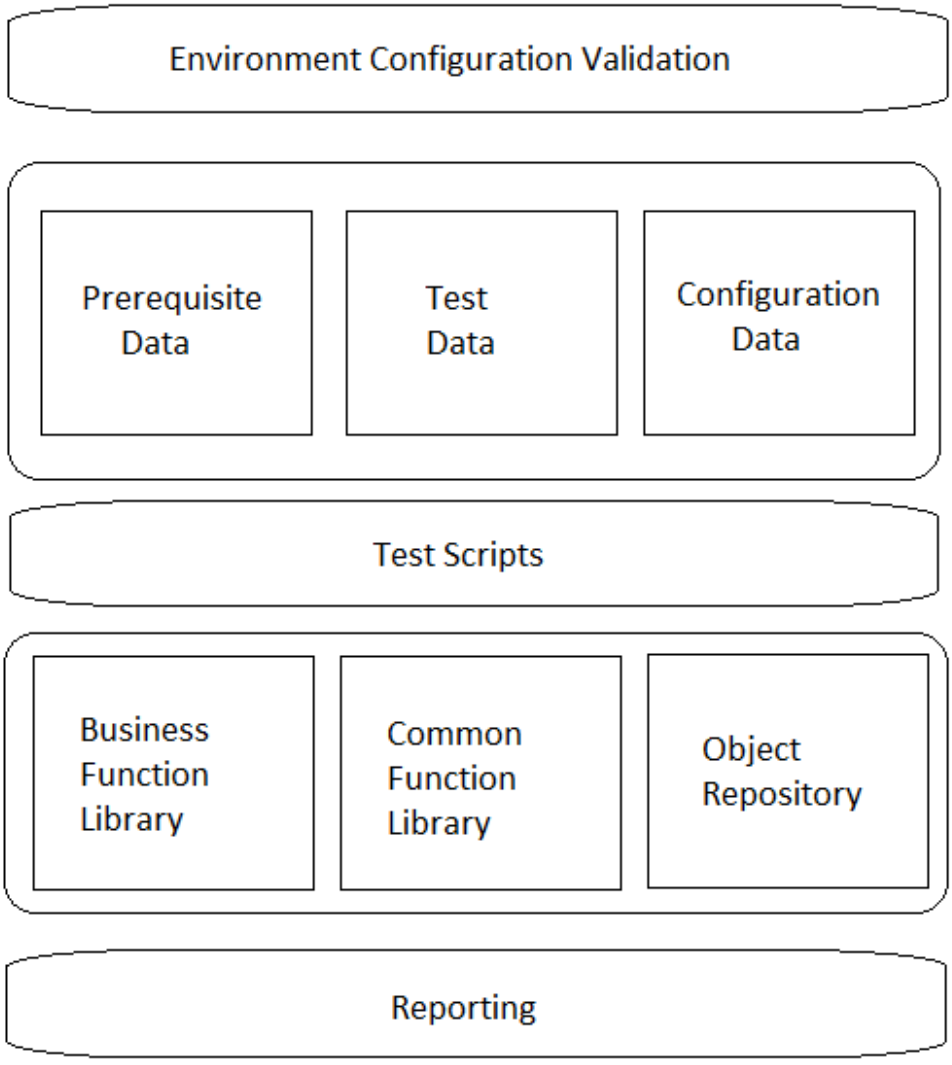

Figure 3.2: Component Of Hybrid Test Automation Framework

3.2.1 Prerequisite Data

As Shown in Figure 3.3 Allocation product is tightly integrate among other retail products.Oracle Retail Allocation calculates the allocation based on the information it has

received from the merchandising system. Once the retailer reviews and approves the allocation, Oracle Retail Allocation sends the following information back to the merchandising system.

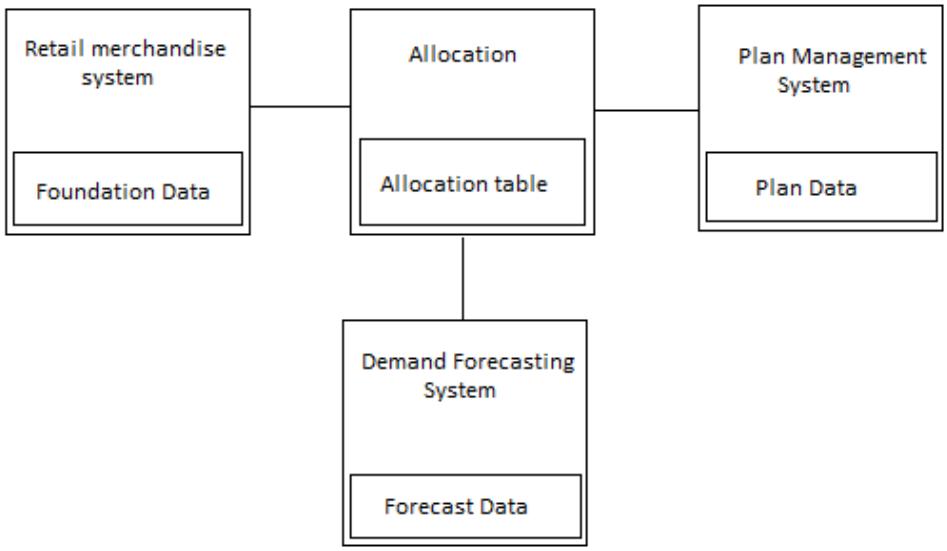

Figure 3.3: Allocation Integration With Other Systems

To test whether all the functionality of allocation product is working correctly,we need setup & configure other product on which allocation is dependent.

Allocation prepare its allocation plan after getting following data from three other retail product.These data are Prerequisite for allocation:

- **Foundation Data**

  Information which is related to fundamental element of retail business process are know as foundation data.Fundamental element of business process like locations,supplier,item and organization hierarchy are well manage by oracle merchandising system.Items are classified within the Merchandise Hierarchy.company,area, region,location,store,department, item,supplier and order are the data which require by allocation in prior.

- **Forecast Data**

  Forecasting data Oracle Retail Allocation accesses forecasting data that originates in the Oracle Retail Demand Forecasting (RDF) system.

- **Plan Data**

Oracle Retail Allocation accesses plan data that originates in the planning application.

3.2.2 Data Creation Process

Foundation data creation process take very long time if created sequential.To overcome this issue we need to find way to parallel insert data into system so overall time require to insert prerequisite foundation data can be reduce.Following is way how we create the foundation data parallel:

- list all data element which need to created
- Create dependency list among data element
- Non dependent elements are categories among four sublist
- each sublist data are created parallel from different virtual machine

3.2.3 Environment Configuration

Allocation follows three tier architecture as shown in figure 3.4.business process are reside in application server and data are reside at data server.

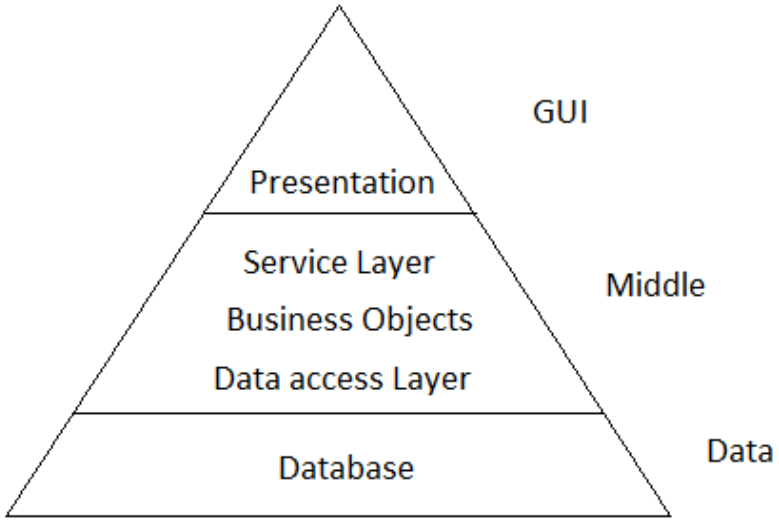

Figure 3.4: Allocation Architecture

During testing process we need to verify that actual data are properly inserted or not into database.For this reason we need detail relate to application as well as database

server and also database access credentials.this all information are fetch during initial automation environment setup time.So before running any script we need to fetch server detail which are store into environment configuration file.

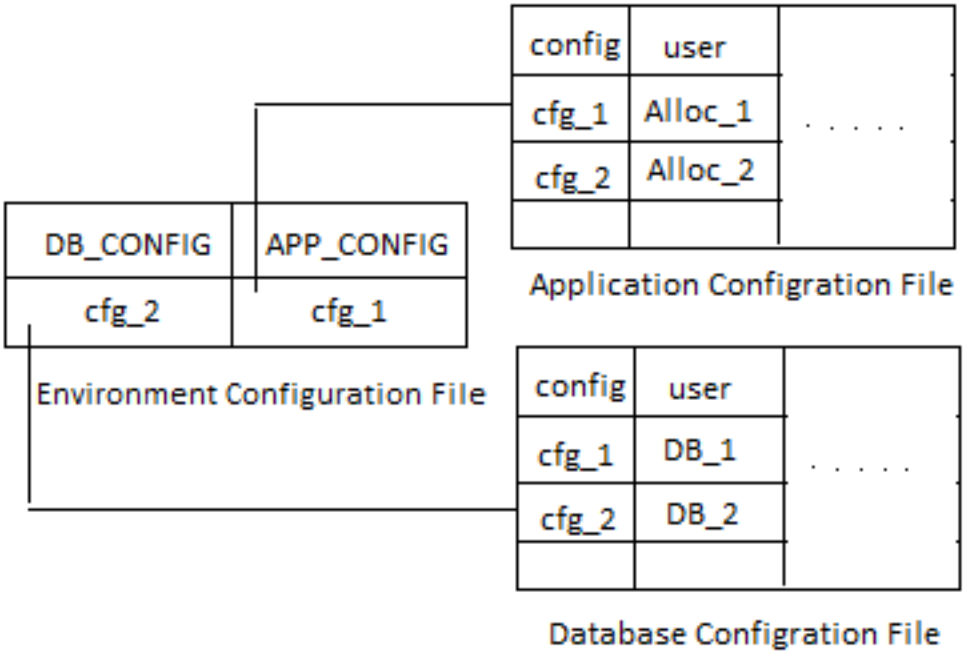

Figure 3.5: Environment Configuration

In configuration file we need to specify the which application or database configuration we are going to use during whole automation process.Configuration file structure is as shown in figure3.5.Environment configuration process set the application and data server according to configuration.

3.2.4 Test Data

For testing system we required data for each test scenario.In every test scenario we need to interact with application and need to provide different test data as input.Any such specifically identified data which is used in tests is known as test data.Test data can be maintain in separate data sheet which manual or automatically enter during execution of testcase.Test data can be store into various datasource like file,database and also can be store in different data format like json,xml,table.More part of test data contain inputs and expect results.We are getting testdata for each testcase from quality assessment team.This testcases and testdata available through quality assurance tool provide by automation test tool.

3.2.5 Environment Configuration Validation

The Environment configuration helps the tester to create an environment which is seamless to automate and provides functionalities to handle automation test reports.Environment configuration is a baseline for the testing to commence as it provides the necessary skeleton for manual testing or automation process to initiate. Aim of this component is to check whether Environment is correctly configure or not.Two major checks we are performing in this component are listed as below:

- **Prerequisite Data Check**

  Oracle retail have large collection of product to cover every business activity of retail industry.Automation team run testcase of every product to measure there quality.For running script one environment/server is given where every software is installed and configure according to requirement.After Getting Environment, automation team generate prerequisite data for each application.This component helps to identify that prerequisite data properly inserted or not.After performing datacheck for each product it will return report.

  We are maintaining one workbook where each product datacheck are mention in different sheet.We are fetching each sheet,performing database check operation and generating pipe separate file as result which give you data check status,success or failure with reason code as described in Figure 3.6.

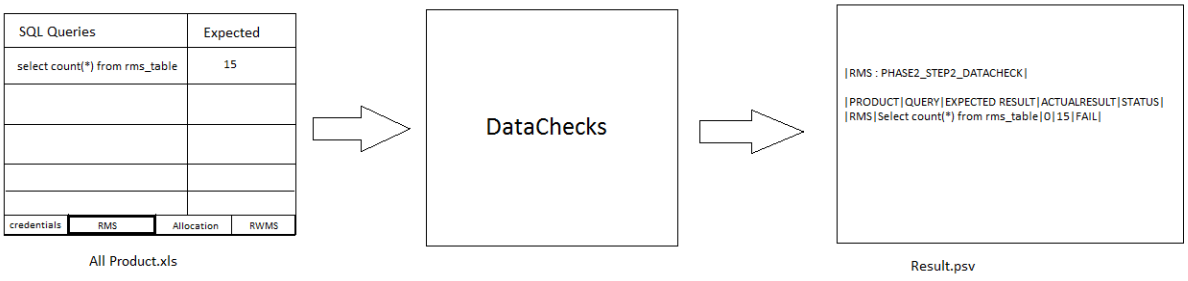

Figure 3.6: Prerequisite Data Check

- **RIB Configuration**

RIB is the main interface for data to flow between any two products, RIB uses JMS (Java messaging services) internally.Aim of this component is to check whether new environment consists of all the adapters as old environment or not and also make sure there status should be up and running. Adapters are required because these are product specific methods which actually let data to flow from one product to another product. Any data which we want to send, that data is packaged as an XML message and then RIB is used to transfer this message to another product by passing it to the corresponding adapter which in turn converts the XML message to other product format as described in Figure 3.7.

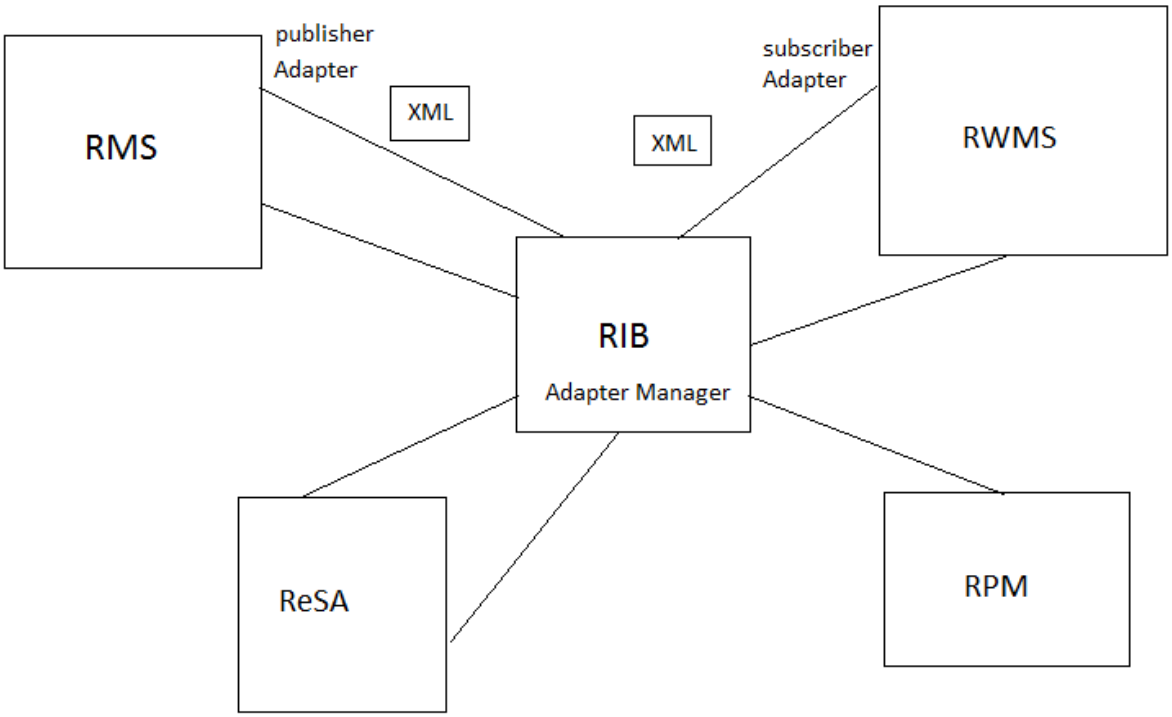

Figure 3.7: RIB Configuration

- **Automation of SVN Checkout**

  To perform Datacheck for every product we are maintaining workbook which have credential and datacheck in separate sheets.This component help to directly fetching from credential from credential page and also write directly result to SVN.This will avoid manually getting credential and also putting result in SVN repository as described in Figure 3.8.

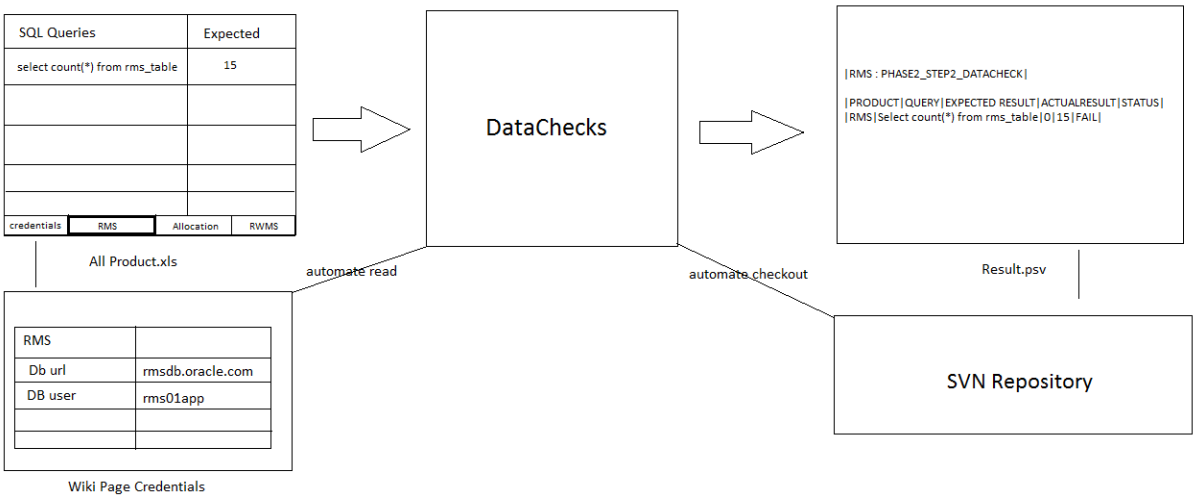

Figure 3.8: Automation Of SVN Checkout

3.2.6 Object Repository

Object Repository is a collection of object and properties with which automation tool will be able to recognize the objects and act on it. When a user records a test, the objects and its properties are captured by default. We are managing object repository for every library file in separate way.

3.2.7 Common Function Library

Common Function Library contains functions which are need during business library implementation. This function are generic function which work as support function during business library implementation. These functions are independent of Test Automation Framework & can be also useful outside the context of the framework. In this framework the Support t Functions. Following are the type of function which is part of common function library:

- **File Handling**

  This module handle i/o active relate to file.reading,writing and matching two file are key functionality.

- **String Handling**

  During validation we need to verify two string are matching or not. This module contain all operation required to manipulate string.

- **Validation**

  During validation we need to validate that presentation view the correct table data or not. For these we need to take data from view and compare it with the datatable which we got from database.These module handle all operation relate to table validation.

- **Database Access**

  test data and other configuration data are reside in database. So we need to provide data access interface. This module is responsible to handling the database connection read, write and update queries.

3.2.8 Reporting

This section describe about how detail report are generate after execution of testcases and aggregate report for each product. This section also describe about email service to send report to respected teams for each product.

- **Detail Reporting**

  Test Framework needs to give enough information to developer and quality engineer so that they can investigate reason behind failure of testcases. It is not enough if only short status of testcase execution is given as failed or passed. While running testscript openscript is automation tool which automatically generate report from log which is generate by testscript. Openscript generate report in three format one can be seen test tool itself, second one is html format which can be seen in Figure 3.9 and third it generate in form of xml which can be exchangeable across different system.

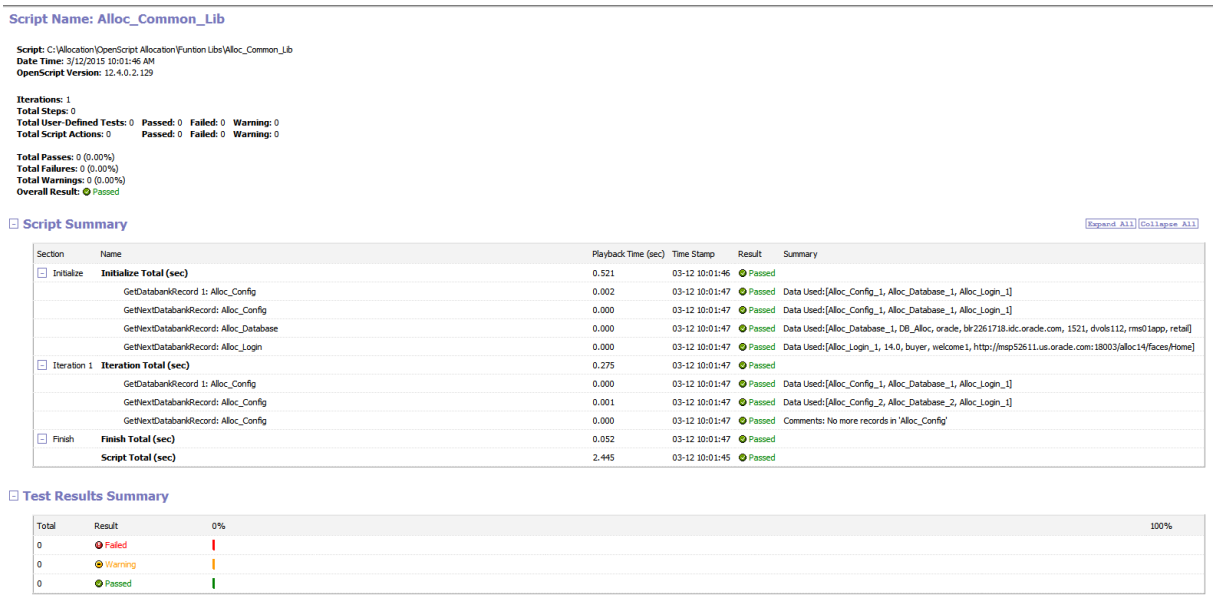

Figure 3.9: Detail TestScript Report

- **Aggregated Report**

  Top level Manager only interested in aggregate report which provide failure vs passed ratio. This aggregate report can be generate from detail report generate by automation tool. We are generating aggregate report which provide total testcases

failure count functional area vice as seen in Figure 3.10.

	A	B	C	D
1	Function Name	Passed	Failed	
2	BO: Bank Deposits	2	1	
3	Reports Preview	3	7	
4	CleanUp	0	1	
5	Bank Deposites	0	1	
6	Gift Receipts	1	0	
7	CO: Data Management	2	3	
8	CO: Transaction Tracker	4	5	
9	CO: Admin Task	13	27	
10	Scan Sheet	2	1	
11	BO: Reports	16	29	
12	CO: Customer	4	0	
13	BO: Employee	13	4	
14	Suspended Transactions	0	2	
15	BO: Daily Operations	39	3	
16	BO: Discount Rule Maintenance	24	1	
17	BO: Item Maintenance	5	1	
18				

Figure 3.10: Aggregate Report

- **Email Service**

  Every Oracle retail products are tested by automation for measuring system quality.At every product release, automation team execute testscripts for each and every product respective. As describe earlier openscript automation tool generate detail report for each and every testscript and also aggregated result is generate for each product.This report need to send to quality assurance team and also developer for improve the software quality. This component provide webservice by which we are able to send report to respective team via mail.This component takes sender detail, recipients detail, reports, subject and message in form of JSON or XML format. After receiving data it send mail respected recipients.

Chapter 4

Conclusion

Achieved aim of this project by Developing software test automation framework for allocation application which consists of functional libraries, test data, configuration scripts, configuration validation scripts, reporting module which simplify automation testing process. Developed framework will be use to automate test scenarios which further use to measure quality of Allocation application.

References

[1] E. R. K. Prof. (dr.) V. N. Maurya, "A review of scripting techniques used in automated software testing," *International Journal of Advanced Computer Science and Applications*, vol. 5, 2014.

[2] P. Laukkanen, *Data-Driven and Keyword-Driven Test Automation Frameworks*. PhD thesis, Helsinki University Of Technology, Feb 2006.

[3] E. R. K. Prof. (dr.) V. N. Maurya, "Analytical study on manual vs. automated testing using with simplistic cost model," *International Journal of Electronics and Electrical Engineering*, vol. 2, JAN 2012.

[4] D. G. Harpreet Kaur, "Comparative study of automated testing tools: Selenium, quick test professional and testcomplete," *Journal of Engineering Research and Applications*, vol. 3, SEPT 2013.

[5] "Oracle retail allocation data sheet," *Oracle*, 2013. http://www.oracle.com/us/products/applications/062048.pdf.