International Journal of Pure and Applied Mathematics
Volume 118 No. 8 2018, 559-566
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

ijpam.eu

Database Testing using Selenium Web Driver – A Case Study

V.Neethidevan
AP(SLG) - MCA Department
Mepco Schlenk Engg. college
Sivakasi
neethidevan@mepcoeng.ac.in

G.Chandrasekaran
Director-MCA Department
Mepco Schlenk Engg. college
Sivakasi
gchandra@mepcoeng.ac.in

ABSTRACT

Selenium is an open source software testing tool used to perform automated testing to test web based applications. It has lot of software components used to perform various types of software testing. It is also used to perform cross browser testing so that tester can ensure that the given application can be executed in different types of browsers without any problem. Using web driver component of selenium, web applications can be automatically tested. The various forms of database testing include primary key, secondary key, stored procedures and various constraints. The present study focuses on a case study, in which database testing is carried out with Selenium Web Driver.

Keywords—Software testing, Automation testing, Testing tools, Database testing , Web Driver.

I. INTRODUCTION

Database is very much helpful to store data and it is vital for every organization. Each application has to store the data, in a database. Moreover, it has to store only relevant data and the tester has to ensure that it contains valid data. To confirm this, database testing is carried out with various options like duplicate data checking by means of primary key. To check for data validity, check constraints are created. The stored procedures and triggers are tested whether they are properly working or not.

There are two ways to test a database:

1. Manual
2. Automation

In manual testing, tester has to check each table manually and can report if any problem arises. But it is time consuming and a complex task if it has more tables and more data. To save time and be reliable, automated tools can be used for database testing.

Selenium is used to perform various types of testing methods, like,

- Functional Testing.
- Regression Testing.
- Sanity Testing.
- Smoke Testing.
- Responsive Testing.
- Cross Browser Testing.
- UI testing (black box)
- Integration Testing.

Selenium Web driver is an automated tool used for automating web and mobile applications. It provides various in-built APIs which helps in writing user-friendly and easy to understand code that contains Java code. It interacts with various elements of a web page and performs the expected function. This tool also supports various programming languages like Java, C#, Ruby, etc. to write Selenium based test scripts which will be executed on different browsers. The present study focuses on a case study, in which database testing is carried out with Selenium Web Driver[1].

II. LITERATURE SURVEY

In [2], the author deals with **t**he main benefit of using automated tools to avoid manual effort. Manual testing is time consuming, tedious and requires heavy investment in human resources. Automation tools enable us to record the test suite and re-play it, if required. Once the test suite is automated, no human intervention is required. In automation testing the initial investments are bigger than manual testing and we cannot automate everything but automatable test cases, determine which ones (manual or automated) would provide the biggest return on investment. In [3], Akalanka Mailewa et al , while the most basic form of combinatorial testing - pairwise testing - is well established, and adoption by software testing practitioners continues to increase, the software industry is yet to fully adopt higher strength combinatorial testing. Test automation paper will demonstrate a hands-on sample industrial web based software project involving functional test automation with SELENIUM tool. In [4], Monika Sharma et al, had discussed about various web automation testing tools which will help us to understand the automation testing as well as the tools available for automation testing. A variety of web based systems and applications are tested by automation testing tools. The automation testing script is used in test automation. To choose the best tool for a task, various issues like ease of integration should be considered and weighed against the cost and performance. Also the tool needs to be compatible with the design and implementation of an application.

In [5], Taranpreet Kaur et al, had discussed about a tool enable user to perform all the operations from the beginning of data definition which includes Creating, Altering, Dropping and Truncate of table to the data manipulation that includes Selecting, Inserting, Updating and deleting of data, which is followed by controlling them that is commit and reset. In Data Definition we have commands of Creating and dropping of databases followed by creation of tables within a database. And it also allows altering the contents of table either by adding column(s) or removing them. Along with this, user can truncate the table that is removing the whole content of a table. In data manipulation, user has the provision of selecting a specific amount of data from a table by using "where" clause in the command of view data. User can also view data from different table(s) of databases by using sub – queries and join commands according to the requirement.

.

In [6], Prasanth Yalla et al, had proposed a method for an integration testing tool called Selenium that tests the automated web applications. In our proposed framework, Selenium under integration testing tool tests several web applications and generated the code in different languages that include JAVA, C# etc.

In [7], M.Y. Chan et al, had proposed to complement white box techniques with the inclusion of the SQL semantics. This approach is to transform the embedded SQL statements to procedures in some general-purpose programming language and thereby, generate test cases using conventional white box testing techniques. Additional test cases that are not covered in traditional white box testing are generated to improve the effectiveness of database application testing. The steps of both SQL statements transformation and test cases generation were explained using illustrative examples adapted from a course registration system.

.

In [8], Klaus Haller, had discussed about test database state generation methods and the problem of scheduling test cases efficiently. Thereby, the author provided a road map for the emerging domain of testing database-driven applications and for making such testing useful for commercial software development. In [14], Mirza Mahmood Baig et al, designed a computer aided formula for test suites which make "decision", that the test requirements be executed hypothetically or traditionally. The authors had also designed pseudo code for hypothetical inserting and deleting tuple in the database without changing the originality of the database.

III. SIGNIFICANCE OF DATABASE TESTING

The data that are stored in the database are verified for their validity, then it is called database testing. The data entered by the user in a UI screen must match the record stored in the database. The database testing is more important as it is visible to the end-user and it can't be avoided by the testing team. As more applications are developed with more complexity, there is a need for testing the database to make it more reliable and robust. To perform database testing, check the routine work like, addition, deletion, updation operations which involves data about all the entities related to the application under test. In todays' era of application development, the complexity of database is more, due to the business logic which plays an important role for the applications. Tester should ensure that values have been added correctly after the implementation of the business rules.

The database has to be validated by various aspects of testing like the following.

a) Data Mapping:

In any software based applications, data normally present in the database. The user retrieves data from the database whenever there is a need. Also he could store large amount of data in the database. The tester has to ensure the following.

- Often the tester has to ensure fields in the UI/frontend forms are mapped consistently with the corresponding fields in the DB table. Naturally this mapping information is defined in the requirements documents.
- When a certain action is performed by the user in the front end of an application, a corresponding CRUD

(Create, Retrieve, Update and Delete) action gets invoked at the back end.

b) ACID properties validation

Atomicity, Consistency, Isolation and Durability. Every transaction carried out in a DB must obey to these four properties. The tester has to ensure these properties by choosing suitable test cases, to test each transaction.

c) Data integrity

It means exactness of the data. Once we have stored data, at any cost it should not be changed. Once a value is updated in one UI screen, then it can't display any older value on another screen. So DB test cases must be created in a way to ensure consistency of data.

d) Business rule conformity

If the application has more complex data, then more complicated components like relational constraints, triggers, stored procedures, packages etc. have to be checked. So testers use most appropriate SQL queries in order to validate these complex objects.

In [9], Taranpreet Kaur et al , describe about a new tool for database testing, Swing, is the primary Java GUI widget kit. It is a part of Oracle's Java Foundation Classes (JFC)—an API (Application Programming Interface) which is used for providing a graphical user interface (GUI) to the Java programs. Swings were developed in order to provide a more sophisticated set of GUI components as compared to the earlier Abstract Window Toolkit (AWT). Swings provides a native look and feel that emulates the feel and look of several different platforms, and also supports a plug-in look which allows applications to have a look and feel unrelated to the underlying platform.

In [10], Chandraprabha et al, proposed a framework to perform automation testing using "Selenium WebDriver". With Selenium WebDriver authors developed data driven framework which means separating data to code for reusability purpose. In this framework we abstracted the data which would be used in excel file and the program is written to access data from that excel files.

In [11], J.Tuya et al, developed an approach to reduce a database with respect to a set of SQL queries and a coverage criterion. The reduction procedures search the rows in the initial database that contribute to the coverage in order to find a representative subset that satisfies the same coverage as the initial database. The approach is automated and efficiently executed against large databases and complex queries. The evaluation is carried out over two real life applications and a well-known database benchmark

In [12], ***Neha Dubey*** to carry out a comparing and studying the concepts, builds and features of automated tools such as the Ranorex and the Automated QA TestComplete based on criteria such as the hard work involved with generating test scripts, capacity to playback the scripts, end result reports, and expenditure

In [13], Tarik Sheth et al, developed an approach to establish a mechanism used evaluate testing tools effectiveness, at the moment, there are many complex systems are built across the platforms and it is very complex problem to establish one common criterion to evaluate the testing tools.

IV. CHALLENGES IN DATABASE TESTING

Various challenges for the tester during database testing

1. Large scope of testing

First thing is to find out the various items to be tested in database. Once various items to be tested are identified for database testing, testing can be carried out in an effective manner and quality of the application is getting improved. Also total effort required to design the tests and execute test for each test item is estimated. If more time is required to test all these test items, select only important test items for the testing purpose.

2. *Incorrect/ scaled-down test databases*

The database to be tested contains only small amount of test data. But it is not sufficient, since testing has to be done with more realistic data.

3. *Changes in database schema and data*

It is more challenging, since we must be aware of changes made to the database during testing. We must analyse the impact of changes and modify the impacted tests.

4. *Messy testing*

Database contains more data. Create a test plan and prepare a set of test cases and proceed testing as per schedule.

5. *Lack of skills*

Assigning best personnel to testing process will be a big challenge for the testing team. The tester who wants to perform database testing, must have proficiency with SQL commands and SQL queries.

V. PERFORMING DATABASE TESTING

To perform database testing, tester must have more experience about the knowledge related to database design in addition to skills in database testing tool.

Different forms of database testing involves web application or desktop and mobile based applications. Different types of applications like banking, finance, health insurance which contain more confidential data requires extensive testing.

The following points should be more useful to perform this type of testing.

First of all, tester should make sure that he/she understands all the applications totally and which database is being used with the testing application.

- Figure out all the tables which exist for the application and try to write all the database queries for the tables to execute since there are many things which are really complex, so you can take the assistance of developers and figure out the queries. Test each and every table carefully for the data added. This is the best process for the testers to perform the DB testing, it can be done for any application and it does not matter application is small or big.
- If things are really complex, then tester can obtain the query from the developer to test the appropriate functionality. Database is the spine of the application and tester should make sure to test very carefully. It requires skill, proficiency and sound knowledge.

VI. DEVELOPMENT AND TESTING OF SCRIPTS

In this study, we have taken an application like Examination registration for students. The application contains a web page, in which user to enter various details like, name, college name, course name etc. Once entered by the user, it is stored in a database (MySQL). After that, using Web Driver concept we have developed Selenium scripts, that connect with MySQL database and retrieved the last record entered by the user. Now the testing script will check whether each data entered by the user is correctly stored in the database.

If they are correct, the test is successfully passed and corresponding message is displayed otherwise an error message should be given to the user.
In this way, once the user has entered the data, it must be validated with the data available in the database. Also using selenise commands title pages are also checked. MySQL is the backend for storing registration details about each student.
The following figure 1, shows a web page developed using PHP and various details are entered by the user and those details are entered into the MySQL database.

What is WebDriver?
WebDriver is a web automation framework that allows you to execute your tests against different browsers, not just Firefox (unlike Selenium IDE). Gecko Driver is an executable file that you need to have in one of the system path before starting your tests. Firefox browser implements the WebDriver protocol using an executable called GeckoDriver.exe. This executable starts a server on your system. All your tests communicate to this server to run your tests. It translates calls into the Marionette automation protocol by acting as a proxy between the local and remote ends.

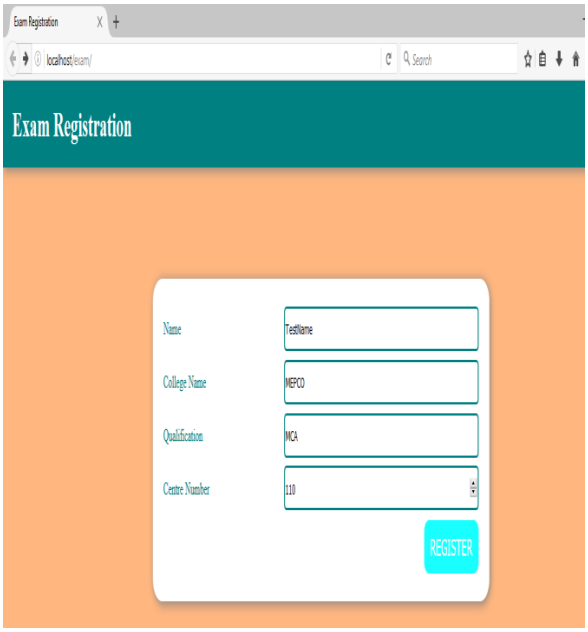

Figure 1– Registration page

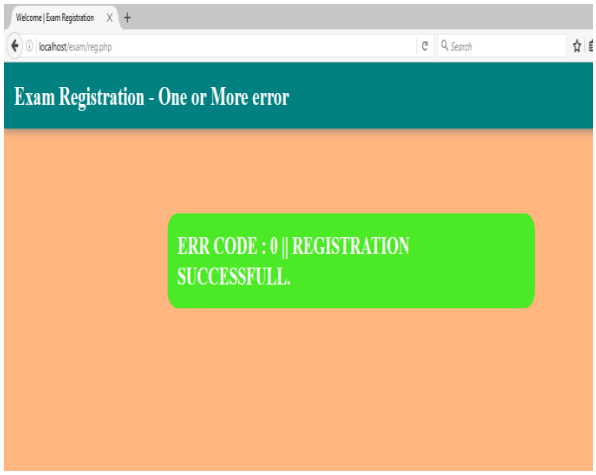

Figure 2 Successful registration page

The figure 2 shows the output, in which the database had the data entered by user. Thus it is verified by the database testing.

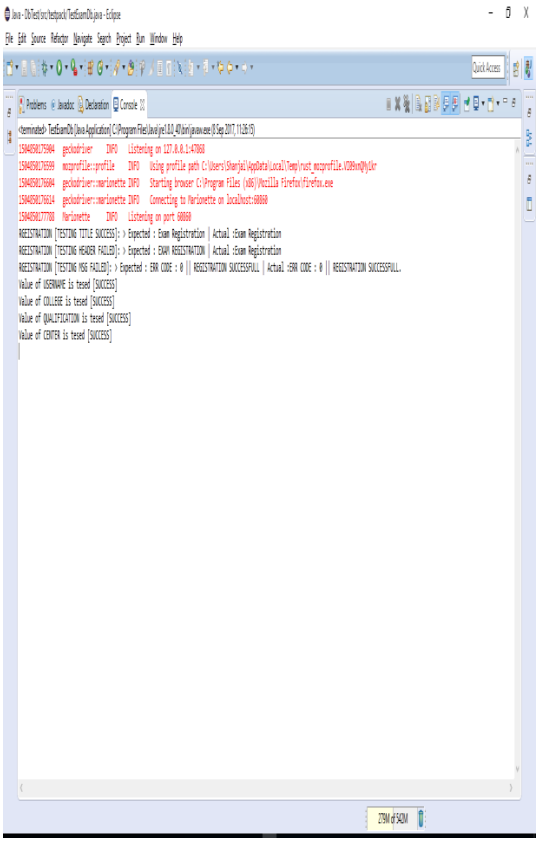

Figure 3 Output that shows page is validated.

Using Selenium Web driver concept, we have developed a testing script (refer Appendix). With JDBC driver concept, we made a connection to the MYSQL database using appropriate jar files. after that last record entered by the user is retrieved and each field value is checked for its correctness. Once they are equal, we ensure that the data entered by user it, getting stored in the database. Thus data is validated and this is one form of Database Testing done automatically.

VII. FUTURE WORK AND ENHANCEMENT

A mobile app is a software application developed specifically for use on small, wireless computing devices, such as smartphones and tablets, rather than desktop or laptop computers. Compatibility testing plays a vital role in software testing and it gives more confidence to ensure the application under test is usable across multiple platforms. Since lot of mobile devices are available, we have to perform testing using mobile simulators. Appium drives various native automation frameworks and provides an API based on Selenium's WebDriver JSON wire protocol [15].

VIII. CONCLUSION

Automation of software testing is not an easy process. It involves a lot effort from the testing team. Selenium is one of the most popular tool used for web based testing and database testing Test engineers with more skills are needed to perform automation of software testing. To conclude this, if automation is performed by more qualified test engineers using any of the Automated Tools like Selenium, it can be an asset to the company.

REFERENCES

[1]. V. Neethidevan , Role of Automated Testing Tools in Software Testing , GRD Journals- Global Research and Development Journal for Engineering | Volume 3 | Issue 1 | pages [28..32], December 2017.

[2]. Preeti Yadav, Ajay Kumar , AN AUTOMATION TESTING USING SELENIUM TOOL, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS). Volume 4, Issue 5(2), September - October 2015.

[3]. Mailewa, Akalanka, Jayantha Herath, and Susantha Herath. "A Survey of Effective and Efficient Software Testing." The Midwest Instruction and Computing Symposium. Retrieved from http://www.micsymposium. org/mics2015/ProceedingsMICS_2015/Mailewa_2D1_4 1. pdf. 2015.

[4]. Monika Sharma, Rigzin Angmo, , Web based Automation Testing and Tools , International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 908-912.

[5]. Taranpreet Kaur, Designing and Development of Database Testing Tool, International Journal of Computer Applications (0975 –8887) Volume 120 No.1 June 2015.

[6]. Prasanth Yalla, Dr. L S S Reddy, M.Srinivas, T.Subha Mastan Rao , Testing tool with respect to Integration Testing, IJCST Vol. 2, Issue 3, September 2011.

[7]. M.Y. Chan and S.C. Cheung, HKUST ,Testing Database Applications with SQL Semantics. Proceedings of 2nd International Symposium on Cooperative Database Systems for Advanced Applications (CODAS'99), Wollongong, Australia, March 1999.

[8]. Klaus Haller COMIT AG, White-Box Testing for Database-driven Applications, A Requirements Analysis, 2009.

[9]. Taranpreet Kaur, Designing and Development of Database Testing Tool, International Journal of

Computer Applications (0975 – 8887) Volume 120 – No.19, June 2015

[10]. Chandraprabha, Ajeet Kumar, Data Driven Testing Framework using Selenium WebDriver, nternational Journal of Computer Applications (0975 -8887) Volume 118–No. 18, May 2015.

[11]. J. Tuya, de la Riva, M.J. Suárez-Cabal, R. Blanco , Coverage-Aware Test Database Reduction IEEE Transactions on Software Engineering.

[12]. *Neha Dubey, Mrs. Savita Shiwani*, Studying and Comparing Automated Testing Tools; Ranorex and TestComplete, *International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 3 Issue 5 may, 2014 Page No. 5916-5923.*

[13]. Tarik Sheth *, Dr. Santosh Kumar Singh, Software Test Automation- Approach on evaluating test automation tools, International Journal of Scientific and Research Publications, Volume 5, Issue 8, August 2015.

[14]. Mirza Mahmood Baig† and Ansar Ahmad Khan, IJCSNS International Journal of Computer Science and Network , S 230 security, VOL.9 No.4, April 2009.

[15].http://blogs.quovantis.com/database-testing-using-selenium/

APPENDIX

```
package testpack;
import java.sql.DriverManager;
import java.sql.ResultSet;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import com.mysql.jdbc.Connection;
import com.mysql.jdbc.Statement;

public class TestExamDb {
private WebDriver driver;
private String targetUrl = "http://localhost:8080/exam";
WebDriverWait wait;
Connection con = null;
Statement statement;
ResultSet rs;
String Url = "";
String database = "regTab";
String dbuser = "root";
String expHeader = "EXAM
REGISTRATION",actualHeader="",expTitle = "Exam
Registration",expSuccessMsg = "ERR CODE : 0 ||
REGISTRATION SUCCESSFULL",actualMsg="";
//FORM TEST DATA
WebElement msg;
String name="TestName";
String college="MEPCO";
String qualification="MCA";
String center="110";

public TestExamDb(){
//init
System.setProperty("webdriver.gecko.driver",
"C:\\Users\\mca\\Downloads\\geckodriver.exe");
driver = new FirefoxDriver();
wait = new WebDriverWait(driver, 15, 100);
}

public static void main(String args[]){
TestExamDb self = new TestExamDb();
self.testReg();
                self.testDb();

        }
public void testReg(){
        //Local Testcases for registration page
        openUrl(targetUrl);
                //testing Title
        if(expTitle.equals(driver.getTitle())){
                System.out.println("RGEISTRATION
[TESTING TITLE SUCCESS]: > Expected : "+expTitle+" |
Actual :"+driver.getTitle());
        }else{
                System.out.println("RGEISTRATION
[TESTING TITLE FAILED]: > Expected : "+expTitle+" |
Actual :"+driver.getTitle());
        }
                //testing header content
        actualHeader =
driver.findElement(By.id("heading")).getText();
        if(expHeader.equals(actualHeader)){
                System.out.println("RGEISTRATION
[TESTING HEADER SUCCESS]: > Expected :
"+expHeader+" | Actual :"+actualHeader);
        }else{
                System.out.println("RGEISTRATION
[TESTING HEADER FAILED]: > Expected : "+expHeader+"
| Actual :"+actualHeader);
        }
//testing registration form
driver.findElement(By.id("nametxt")).sendKeys(name);
driver.findElement(By.id("clgtxt")).sendKeys(college);
driver.findElement(By.id("qualificationtxt")).sendKeys(qualifi
cation);
driver.findElement(By.id("centertxt")).sendKeys(center);
driver.findElement(By.id("regbtn")).click();
//test the messages
```

```
//msg =
wait.until(ExpectedConditions.visibilityOfElementLocated(By
.id("msg")));
		try {
			Thread.sleep(2000);
		} catch (InterruptedException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
		}
		actualMsg =
driver.findElement(By.id("msg")).getText();
		//actualMsg = msg.getText();
		if(actualMsg.equals(expSuccessMsg)){
			System.out.println("RGEISTRATION
[TESTING MSG SUCCESS]: > Expected :
"+expSuccessMsg+" | Actual :"+actualMsg);
		}else{
			System.out.println("RGEISTRATION
[TESTING MSG FAILED]: > Expected :
"+expSuccessMsg+" | Actual :"+actualMsg);
		}
}

		private void openUrl(String url)
{
			driver.get(url);
}

//test DataBase
public void testDb()
{
 try {
Class.forName("com.mysql.jdbc.Driver");
con = (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/e
xamdb",dbuser,"");
String query = "SELECT *FROM regtab ORDER BY id
LIMIT 1";
Statement statement = (Statement) con.createStatement();
rs = statement.executeQuery(query);
rs.next();
 if(rs.getString("uname").equals(name)){
System.out.println("Value of USERNAME is tested
[SUCCESS] ");
 }else{
System.out.println("Value of USERNAME is tested
[FAILED] ");
}

if(rs.getString("ucollege").equals(college)){
System.out.println("Value of COLLEGE is tesed [SUCCESS]
");
}else{
System.out.println("Value of COLLEGE is tesed [FAILED]
");
 }

if(rs.getString("uqualification").equals(qualification)){
System.out.println("Value of QUALIFICATION is tesed
[SUCCESS] ");
 }else{
System.out.println("Value of QUALIFICATION is tesed
[FAILED] ");
 }
if(rs.getString("ucenter").equals(center)){
System.out.println("Value of CENTER is tesed [SUCCESS]
");
}else{
System.out.println("Value of CENTER is tesed [FAILED] ");
 }
} catch (Exception e) {
		e.printStackTrace();
			}

		}

}
```