CSCM64 – Software Testing – Coursework 3

Pallav Shukla - 2154638

Preeti Yadav – 2147070

Shilpi Yadav – 2150456

Testing internet applications

Abstract

The following report is about Internet application testing, in which we will discuss about the architecture, methods, tools, various challenges and issues faced during testing.

- Keywords: Web engineering; Web application testing; Software testing; Automated testing Coverage testing; Testing techniques; Testing tools; Vulnerabilities.

Introduction

In the present world software development process is an important part in which each company spends enormous amount of time and effort [2]. Due to the increasing demand of web application testing deals with great challenge of faster and quicker response [2], [5]. Because of the complex problems web application testing needs to adapt the query according to the specific environment a new approach should be used [1]. According to the research conducted we see several challenges and different tools to conquer the issues created during the testing [3], [2]. We will also define the architecture of Internet application testing [4].

Aims and Objective

Our soul aim is to understand Internet based application testing to find the errors at any given point of time, so we will understand different testing approaches those are functional and non-functional requirements, various stages and strategies, methods and tools, challenges, issues, and architecture.

Literature Review

Internet application must be approached from two different aspects i.e. non-functional and function requirements[1] . Following are the non-functional testing requirements: Performance testing: To ensure that specified system performances are fulfilled e.g. response time, service availability. Load testing: Its goal is to determine the time required to accomplish a variety of tasks and functions under predetermined conditions. Stress testing: It is used to test a system beyond its requirements to see if it will crash or recover from such situations. Compatibility testing: It must identify errors caused by using different web server platforms or client browsers, or different releases or configurations of them. Usability testing: It is mainly concerned with the user interface: issues such as proper information presentation e.g. graphics, text editing format, message clarity, prompts, and commands must be verified. Accessibility testing: It is a form of usability testing whose goal is to ensure that users with physical disabilities, such as blind individuals, have access to the application's content. Security testing: To check the efficiency of the overall application actions against unauthorised user access and improper usage [1] , [5].

A software testing process involves several stages and testing activities, each of which takes into account a different level of testing. First stages is Unit Testing: It is used to validate the individual components of an application. The scope of a unit test cannot be defined uniquely due to the presence of different types of components in both the client and server sides of the programme, such as web pages, script functions, and embedded objects. Integration testing: It evaluates how

different components interact with one another. It focuses on creating efficient methods for merging software components that will be tested as a whole. For example, the various mechanisms used to integrate the heterogeneous and distributed components, which result in different amounts of coupling and data flow between the components. System testing: It aims to identify flaws that are present in the system as a whole rather than in its individual components. Black box testing is widely used to undertake system testing to find flaws in the application's externally observable behaviour. Acceptance testing: User acceptance testing is the final step, and it determines whether the system is ready for release [1].

The approach for designing test cases are defined by testing strategies. They can be responsibility-based (black box), implementation-based (white box), or hybrid (gray box) [1]. Black box testing strategies do not require knowledge of the implementation of the software items under test, but rather generate test cases based on the item's specification or expected functionality. White box testing strategies produce test cases based on the source code analysis of the component under the test model and a coverage model that specifies which sections of the representation must be tested by a test suite. Dataflow testing of internet applications is the white box strategy. Gray box testing strategies combine black box and white box testing approaches to develop test cases: they aim to test a piece of software against its specifications even while understanding its internal workings. Gray-box strategies are suitable for internet application testing since it considers high-level design, environment, and compatibility. [1].

Methodology going through the different research papers we found that there are numerous amounts of testing techniques but they all depend on the type of environment they are having so to summarise we have listed a table 1 [1], [5] as mentioned below.

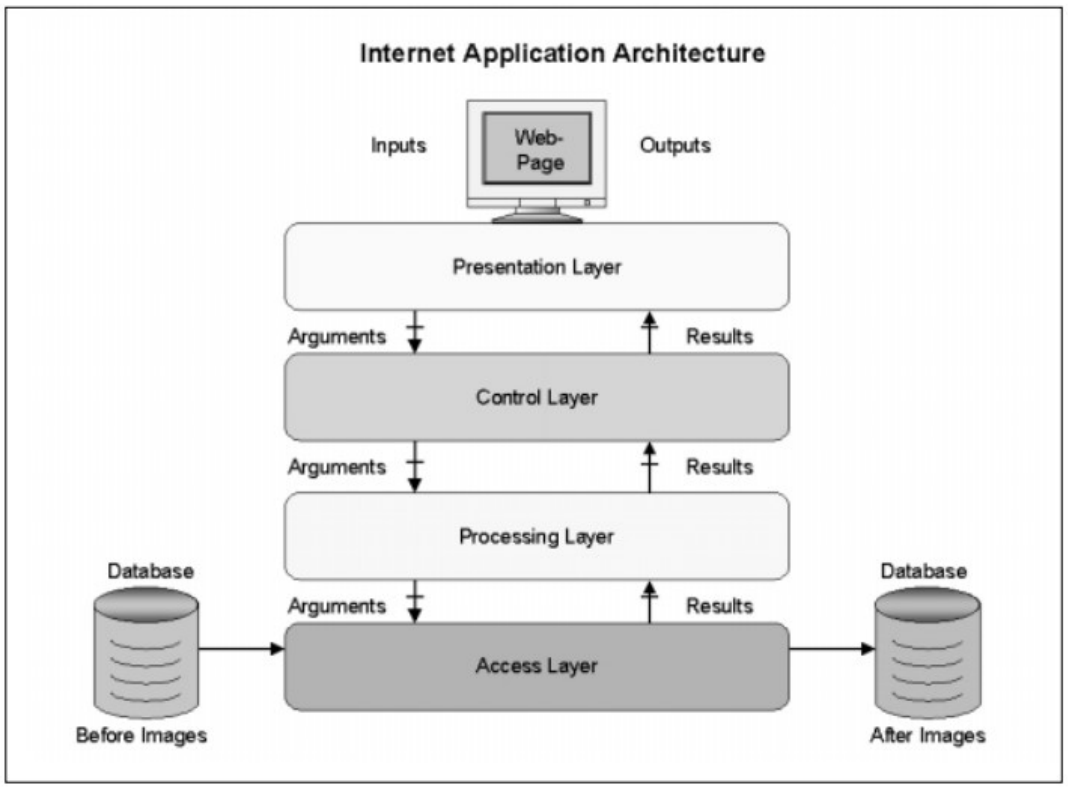
Table 1. Summary of Different Testing Methods/ Techniques available for Web Application Testing

Sr. No	Testing Technique	Central Idea Behind the Technique
1	Structural Testing	Data flow analysis
2	Statistical Testing	Interactions based on a profile use of a web application
3	Mutation Testing	Fault based testing
4	Combinatorial Interaction Testing	Combination of different techniques.
5	Penetration Testing	Active attack simulation
6	Search Based Software Engineering Testing	Branch Coverage of web applications
7	Using UIO and Genetic Algorithms	Path Selection using GA based algorithm
8	GUI Interaction Testing	State based testing of GUI widgets
9	Web Application Slicing	Testing on a slice of web application
10	Cross Browser Compatibility Testing	Testing on different browsers
11	Hierarchical Strategy	Development of an operational profile
12	Bypass Testing	Security testing by avoiding client side validations
13	Leveraging User Session Data	URL testing
14	Browser Fuzzing By Scheduled Mutation	Browser testing in static and dynamic ways
15	Invariant Based Technique	State based testing of interactions
16	Model Based Testing Technique	State based testing based on link navigation

Going through the different methods we found different tools that are applied in the present scenarios. Some of them are listed below [2]:

Table 2 : Various Testing Tools in Web Application Testing

Sr. No.	Tool Name	Type of Testing Supported	Browser Support	Language Supported	Open Source/Licensed
1	WATIR	Functional Testing	IE, Chrome, Safari, Firefox	All	Open source
2	Selenium	Functional Testing	IE, Chrome, Safari,Firefox	Java, .NET, Ruby, Perl, PHP	Open source
3	HP-QTP	Functional Testing	IE, Chrome, Safari,Firefox	VB Script	Licensed
4	Fitnesse	Acceptance Testing	N/A	Java, Python, C#,	Open source



The architecture is defined above is as follows: Presentation Layer – The look and feel of an application comes from the first tier. It provides the visual content to the end user for instance HTML to create the user interface (UI). Control Layer – Here there will be the interface between client and server. Processing Layer – Here all the data get accessed. Access Layer – Here data will get accessed.

Conclusion

In key conclusion we have discussed all testing parts that are directly dependent on the implementation technologies must be carefully adapted to the heterogeneous and 'dynamic' nature of Web applications,

References

1. di Lucca, G. A. (2005). Testing web-based applications: The state of the art and future trends. *Proceedings - International Computer Software and Applications Conference*, *2*, 65. https://doi.org/10.1109/COMPSAC.2005.95

2. Lakshmi, D. R., & Mallika, S. S. (2017). A review on web application testing and its current research directions. *International Journal of Electrical and Computer Engineering*, *7*(4), 2132–2141. https://doi.org/10.11591/ijece.v7i4.pp2132-2141

3. Jaiswal, A., Raj, G., & Singh, D. (2014). Security Testing of Web Applications: Issues and Challenges. *International Journal of Computer Applications*, *88*(3), 26–32. https://doi.org/10.5120/15334-3667

4. *Software Focus - 2000 - Sneed - Testing software for internet applications*. (n.d.).

5. Myers, Sandler, C., & Badgett, T. (2012). The art of software testing (3rd ed.). John Wiley & Sons.