International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 4 April 2023, pp: 108-111 www.ijaem.net ISSN: 2395-5252

User Interface Test Environment

Kamini Achari, Pratiksha Nagode, Apoorva Dusane, Prof. Rahul. M. Raut

Information Technology Engineering, Sandip Institute of Technology and Research Center Nashik- 422213, India

Date of Submission: 28-03-2023 Date of Acceptance: 07-04-2023

ABSTRACT

Now a days, Software Development is rapidly growing. Various innovative applications are getting developed and deployed with quality. Before deploying a software, a software is first tested as per SDLC (Software Development Life Cycle). The Testing is in categorized as Automation and Manual. The world is getting autonomous, so Automation Testing will be more focused for testing the application, as it can be result in Accuracy as well as Time Consumption. When an application is in testing, every application must be compared and matched with the expectations defined. So, the same we are trying to develop is called User Interface Test Environment.
This UITE Environment, helps in testing the UI of a websites. User Interface is the core part of the website and is the only thing which attracts the user to the application or websites. So, to attract the user with UI, the developed UI must be 99% Accurate as per the Expected UI.

General Terms

UI are the most important part and can be considered as the attractive part of the application or website. So, to look more attractive the website's UI must be Quality Tested and should be deployed as per Expectations. So, we are developing the UITE tool, which will test and validate the User Interface as per expectations, and of the expectations doesn't meet it will be report top developer.

Keywords: Automatic testing, programming environments, programming theory, software quality, software testing.

I. INTRODUCTION

today's world, almost all commercial software has a Graphic User Interface (GUI), either web based or console based, and an intuitive and correctly functioning UI is often valued more important than the actual feature set from a customer's perspective. According to [1], 45- 60% of most software code is GUI code. Testing GUI code has unique and challenging requirements for Quality Assurance (QA) Engineers.

First, we must be open to GUI changes during the course of the development cycle. Usually customers provide feedback regarding usability after Beta releases, and from the development perspective it is comparably cheap to make GUI related changes to address this feedback. However, these changes pose great challenges for a test team to absorb the added test cost and minimize regression risk of the altered code[2]. Second, it is difficult to manage the many potential GUI related test cases, especially when you factor in the potential for a single, seemingly simple, change in a GUI's design surface to impact hundreds of previously authored test cases. Based on our experience, 25% or more of a product's test cases might be GUI related and of these 80% of them can be automated. But, because it might not be uncommon for a single dialog to change several times during the development life cycle, the maintenance cost associated with managing deltas in all these test cases can be very large due to the sheer number of associated test cases. Lastly, traditional test metrics might not apply very well to GUI based testing. For example, a QA Engineer might view 70%+ code coverage as acceptable test coverage for a given feature.

1.1 Software and Hardware Requirements:

Hardware:

- Ram- 4 GB
- Core i5 and onwards Software:
- Operating System-Windows 10 and onwards
- Programming Language- Python

1.2 UML Diagrams

1. Data Flow Diagram

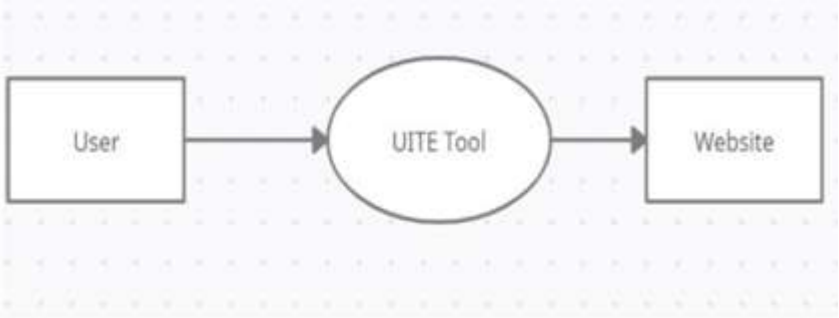

Figure 2.1: Data Flow Diagram Level -0

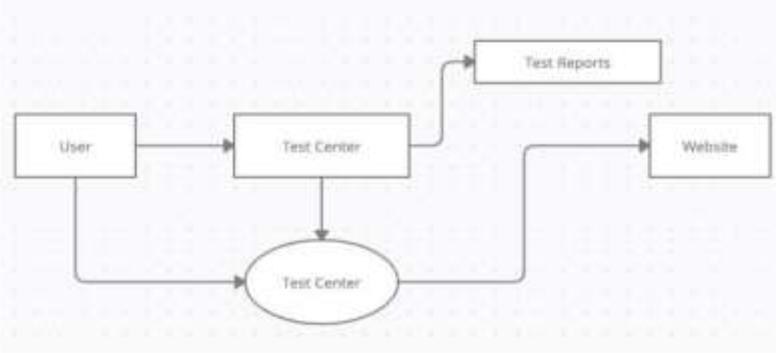

Figure 2.2: Data Flow Diagram Level -1

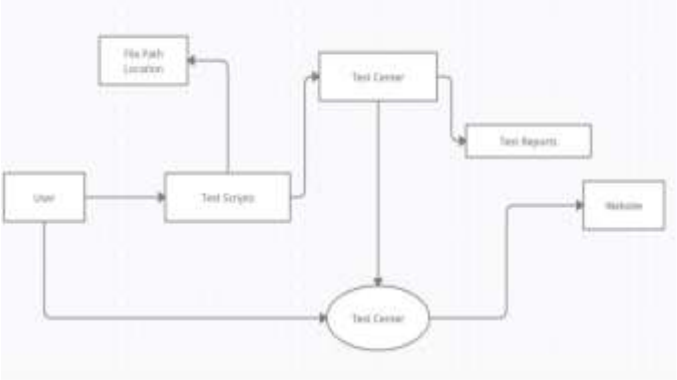

Figure 2.2: Data Flow Diagram Level -2

2. Entity Relationship Diagram

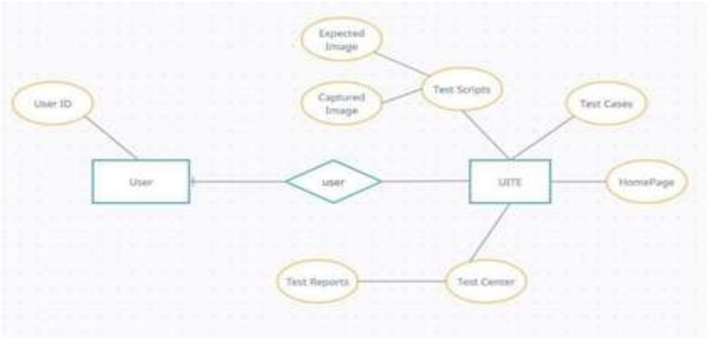

Figure 3: E-R Diagram

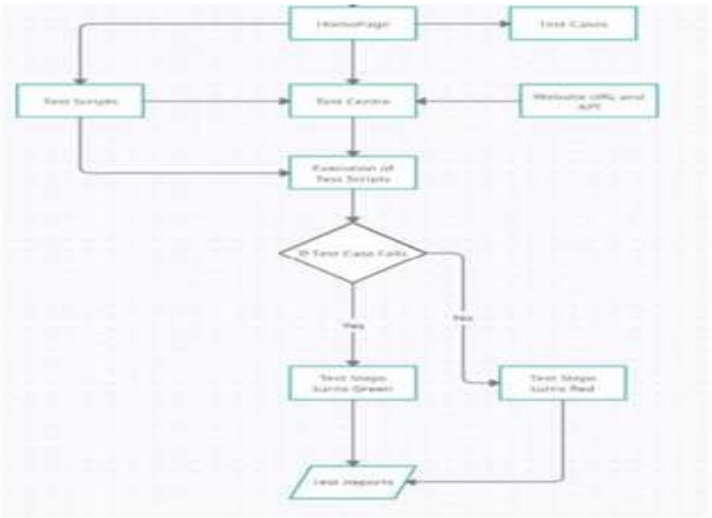

Figure 4.1: Activity Diagram

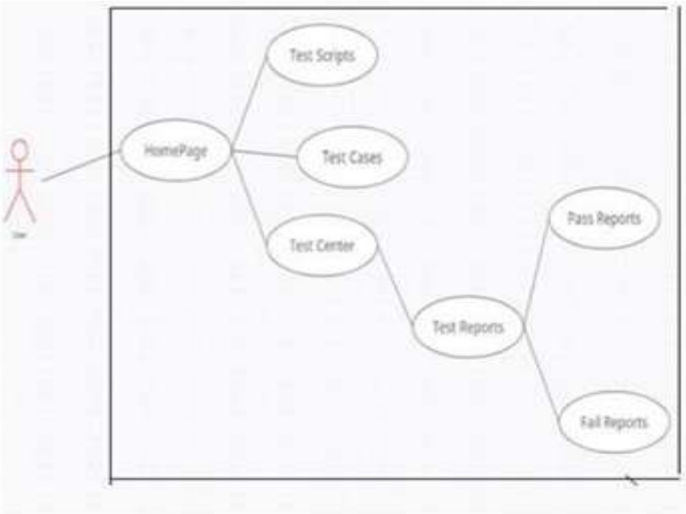

Figure 4.2: Use Case Diagram

II. ADVANTAGES

Acquisition of new customers. A successful UI design contributes to a positive user experience, which is a competitive advantage.

It is easier to incorporate changes discovered through user testing. There can be multiple user interfaces for the same application. interface since the tool will be used with many different applications. Created using the same user interface tool.

III. APPLICATIONS

- Automating web application testing.
- It enables QAs to verify the cross browser compatibility of web application using web driver.

3. Software Quality Attributes

- Adaptability – This software is adaptable by all users.
- Availability – This software is freely available to all users. The availability of this software is easy for everyone.
- Integrity- Integrity refers to the extent to which access to software or data by unauthorized persons can be controlled.
- Security – Users are authenticated using many security phases so reliable security is provided.
- Testability – The software will be tested considering all the aspects.

IV. ACKNOWLEDGEMENT

We are profoundly grateful to **Mr. Rahul M. Raut**, our Project Guide for her expert guidance and continuous encouragement all the time since the projects commencement to its completion. We express deepest appreciation towards **Prof. Rahul M. Raut**, our Project Coordinator, for continuously letting us know about the upcoming project competitions, improvement and additions of the modules in the project We must express sincere heartfelt gratitude towards **Dr. Vivek N. Waghmare**, Head of Department of Information Technology and **Dr. M. M. Patil,** Principal, Sandip Institute of Technology and Research Center, and to all the staff members of Information Technology Department who saw our growth and helped us in every way possible.

V. CONCLUSION

User Interface is the most important part of a software. A software can be more attractive if the UI of same is attractive. User Experience turns positive due to UI too, for the same UI Development should be done properly and to deliver it with Quality it must be Quality Tested. Hence, In this project we are testing the UI with Automation Testing Algorithm.

Future Scope

- In recent times, Software testing has become has a very flourishing career for many people. The career options available in software testing is very vast and diverse.
- Like you can choose to become a test manager , senior testing manager ,QA manager , and many more.
- Simultaneously , there is an increasing demand for open source automation tools such as selenium , robotium.

REFERENCES

[1] Atif M. Memon, "A Comprehensive Framework for Testing Graphical User Interfaces", Ph.D. Dissertation, Department of Computer Science, University of Pittsburgh, July, 2001.

[2] D. Binkley, "Semantics Guided Regression Test Cost Reduction", IEEE Transactions on Software Engineering, vol. 23, no. 8, pp. 498-516, Aug. 1997.

[3] Daniel Hackner and Atif M. Memon, "Test Case Generator for GUITAR", ICSE '08: Research Demonstration Track: International Conference on Software Engineering, Washington D.C., USA, 2008.

[4] "Microsoft Management Console", Microsoft TechNet.

[5] Lao-Tzu, and Hua Ching Ni, The Complete Works of Lao Tzu: Tao Teh Ching and Hua Hu Ching / Translation and Elucidation by Hua-Ching, Seven Star Communication Group Inc. Los Angeles, CA 1995.

[6] James D. McCaffrey, "The Microsoft UI Automation Library", MSDN Magazine, February 2008, vol. 23, no. 2, pp. 115-121.

[7] McCaffrey, James D., .NET Test Automation Recipes: A Problem Solution Approach, Apress Publishing, New York, 2006.