



Black Box Testing on Website-Based Guestbook Registration Applications

Yasir Salih¹, Rifki Saefullah^{2*}

¹*Department of Mathematics, Faculty of Education, Red Sea University, SUDAN*

²*Master's Program of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Jatinangor, West Java, Indonesia*

**Corresponding author email: rifki23008@mail.unpad.ac.id*

Abstract

In the rapidly developing digital era, website-based applications have become an important means of meeting the needs of various uses. One application that has received attention is the guest book registration application, which aims to record and manage guest information efficiently. In this context, application testing becomes crucial to ensure service quality, security and availability. This research focuses on black box testing, a software testing method that emphasizes the functionality and reliability of applications from the user's perspective. Testing covers key aspects such as data input, and key functionality. Through this approach, it is hoped that the research results can contribute to the development of a website-based guestbook registration application that is more reliable and suits user needs.

Keywords: Guest Book List Application, Black Box Testing, Functionality, Testing Methods, Input.

1. Introduction

In this digital era, the development of information technology is increasingly rapid. One form of implementation is through a website-based application. The website application has many uses, one of which is as a guest book list. In this case, it is important to test the application, especially using the black box testing method.

Black box testing is a software testing method that is carried out without paying attention to the structure or internal details of the application (Sholeh et al, 2021). This method aims to test the functionality and reliability of the application from the user's perspective. In the context of a website-based guestbook application, black box testing will help ensure that the application runs well and meets user needs (Itkin et al, 2020).

Black box testing on website-based guestbook registration applications is very important because it involves interaction between the user and the system (Gao et al, 2020). In this case, testing will cover various important aspects such as input validation, user interface navigation, and testing the main functionality of the application (Anagandula & Zavarsky, 2020).

Apart from that, black box testing can also help identify potential problems or bugs that may occur in the application. By conducting thorough testing, we can reduce the risk of errors or failures in using the guestbook registration application (Gustivo et al, 2023). This will improve the quality and reliability of the application.

Black box testing can also help in optimizing the performance of the guestbook application. In this test, load testing will be carried out to test the application's ability to handle a large number of users simultaneously. This is very important because the guest book registration application is usually used at events or public places that have many visitors.

Apart from that, black box testing on website-based guestbook registration applications will also involve security testing. This testing aims to ensure that the data entered by the user is safe and protected from security threats such as hacker attacks or data theft (Aliero et al, 2020).

In black box testing, commonly used testing methods include functional testing, non-functional testing, and general error testing (Zhou & Hastie, 2021). Functional testing will involve testing key features of the app such as guest registration, guest search, and guest deletion. Non-functional testing will involve testing the performance, security, and availability of the application (Turner et al, 2021). While general error testing will include input validation testing, navigation testing, and compatibility testing.

Black box testing on website-based guestbook registration applications, it is important to pay attention to variations in testing conditions (Talib & Doh, 2021). Testing should cover a variety of possible usage scenarios, such as usage

on different devices, usage with a slow internet connection, or usage with invalid data input. This will help ensure that the application can run well under various conditions that may occur (Zednik, 2021).

In this research, we will carry out black box testing on a website-based guestbook registration application using the testing method mentioned previously. We will identify various possible test scenarios and perform thorough testing to ensure the quality and reliability of the application. It is hoped that the results of this research can contribute to the development of a better and more reliable website-based guestbook registration application in the future.

2. Methods

2.1. Testing Needs Analysis

The first step in this methodology is to analyze testing requirements. We will identify the main features of the guest book registration application that need to be tested, such as registration, login and guest list view menu.

2.2. Test Planning

2.2.1. Test Plan on from Register

Figure 1: interface from register

In figure 1 is the registration form for users who do not have an account who are directed from login to register

Table 1: Test plan in from register

Id	Testing	Desired Results
R01	Register an account with "email: aku@gmail.com, and password: 1234"	Can be saved in the database and displays a message when registration is successful
R02	Register an account with the same email	Failed to register and the email message has been used

2.2.2. Test Plan on from login

Figure 2: interface from login

When the user has registered or has an account, then enter from login to enter the main page.

Table 2: Test plan table from login

Id	Testing	Desired Results
L01	Enter an email and password that is not yet registered	A login failed warning appears, because the email or password has not been registered
L02	Register an account with the same email	Failed to register and the email message has been used
L03	Enter the email and password that matches the one you have registered	Login is successful and goes to the main page

2.2.3. Test Plan on the contents of the guest list

Figure 3: interface from guest list contents

In figure 3 is from inputting the guest list.

Table 3: Test plan on the contents of the guest list

Id	Testing	Desired Results
I01	Enter guest list data in the contents of the guest list	The data is stored in databases and appears in the guest list view menu
I02	Enter empty guest list data in the contents of the guest list	A message will appear that must be filled in first.

2.2.4. Test Plan on from edit guest list

Figure 4: interface from edit

In this section, users can edit names, emails and comments on the guest list.

Table 4: Test plan on from edit guest list

Id	Testing	Desired Results
E01	Make changes to guest list data such as names, emails, and comments	Successfully edited and updated data on the guest list view menu

2.2.5. Test Plan on from guest list

Guest list

Date	Name	Email	Comment	Action
2024-01-14	slytherin	fina@gmail.com	hai	Edit Wipe
2024-01-14	jelle	aku@gmail.com	Hallo	Edit Wipe

[Fill in the Guest Book](#)
[Return to Home Page](#)

Figure 5: guest list interface

Figure 5 shows the guest list for each user who fills in the guest list contents menu.

Table 5: Test plan on from guest list

Id	Testing	Desired Results
H01	Delete data on the guest list view menu	Successfully deleted the data and the data was deleted in the view guest list menu

3. Results and Discussion

After carrying out black box testing on the website-based guestbook registration application, we obtained the following results:

3.1. Functionality Testing

In functionality testing, we ensure that the app's key features, such as registration, login, and deleting or editing guest lists, work properly. We carry out tests using a variety of different input combinations, including valid and invalid data input. As a result, the application succeeded in carrying out the expected functionality correctly as in the following test results table:

Table 6: Test results for each from

Id	Testing	Desired Results
R01	Successfully registered the account and a success notification appears	In accordance
R02	Failed to register and a message appears that username is already used	In accordance
L01	A login failed warning appears, because the ID and password have not been registered	In accordance
L02	Login failed and a warning appears. This column must be filled in	
L03	Login is successful and goes to the main page	In accordance

I01	Enter guest list data in the contents of the guest list and save it in the guest list view	In accordance
I02	Enter empty guest list data in the contents of the guest list	It is not in accordance with
E01	Successfully edited and updated data on the guest list view menu	In accordance
H01	Successfully deleted data on the view guest list menu	In accordance

3.2. Problem Finding

During testing, we also discovered some issues or bugs that needed to be fixed. For example, during testing we discovered some issues or bugs that needed to be fixed. Problems such as empty data input in the menu list contents need to be looked at again by the developer to be fixed in the future.

3.3. Improvement Recommendations

Based on the test results, we provide several recommendations for improvements to improve the quality and reliability of the website-based guestbook registration application. These recommendations include improving input validation on guest registration forms to prevent users from entering invalid data. We also recommend developers to improve user interface navigation to make it more intuitive and easy for users to understand.

4. Conclusion

Based on the black box testing that we carried out on the website-based guestbook registration application, it can be concluded that this application has undergone comprehensive and thorough testing. Functionality and performance testing has been carried out well. The test results show that the application is able to carry out the main functionality well.

However, during testing we also discovered some issues or bugs that needed to be fixed. Problems such as empty data input in the menu list contents need to be looked at again by the developer to be fixed in the future.

In this case, we recommend that developers improve input validation on guest registration forms so that users cannot enter invalid data. Apart from that, improvements need to be made to the user interface navigation to make it more intuitive and easy for users to understand.

By making improvements and developments based on the findings and recommendations that have been submitted, it is hoped that the website-based guest book registration application can improve quality and reliability better. This application is expected to provide a better user experience, meet user needs, and provide better security in use and data protection.

Through this research, we hope to contribute to the development and improvement of the quality of the website-based guestbook registration application. With structured and systematic black box testing, it is hoped that this application can become more reliable, effective and efficient in its use.

References

- Aliero, M. S., Ghani, I., Qureshi, K. N., & Rohani, M. F. A. (2020). An algorithm for detecting SQL injection vulnerability using black-box testing. *Journal of Ambient Intelligence and Humanized Computing*, 11, 249-266.
- Anagandula, K., & Zavarsky, P. (2020, June). An analysis of effectiveness of black-box web application scanners in detection of stored SQL injection and stored XSS vulnerabilities. In *2020 3rd International Conference on Data Intelligence and Security (ICDIS)* (pp. 40-48). IEEE.
- Gao, X., Saha, R. K., Prasad, M. R., & Roychoudhury, A. (2020,). Fuzz testing based data augmentation to improve robustness of deep neural networks. In *Proceedings of the acm/ieee 42nd international conference on software engineering* (pp. 1147-1158).
- Gustinov, M. D., Azani, N. W., Al Ghani, R., Auliani, S. N., Maharani, S., Hamzah, M. L., & Rizki, M. (2023). Analysis of Web-Based E-Commerce Testing Using Black Box and White Box Methods. *International Journal of Information System and Innovation Management (IJISIM)*, 1(1), 20-31.
- Itkin, I., Treshcheva, E., Konnova, L., Braslavski, P., & Yavorskiy, R. (2020, December). Black-Box Testing of Financial Virtual

- Assistants. In *2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (pp. 684-685). IEEE.
- Sholeh, M., Gisfas, I., & Fauzi, M. A. (2021). Black Box Testing on ukmbantul. com Page with Boundary Value Analysis and Equivalence Partitioning Methods. In *Journal of Physics: Conference Series* (Vol. 1823, No. 1, p. 012029). IOP Publishing.
- Talib, N. A. A., & Doh, K. G. (2021). Assessment of Dynamic Open-source Cross-site Scripting Filters for Web Application. *KSII Transactions on Internet & Information Systems*, 15(10).
- Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., & Guyon, I. (2021). Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track* (pp. 3-26). PMLR.
- Zednik, C. (2021). Solving the black box problem: A normative framework for explainable artificial intelligence. *Philosophy & technology*, 34(2), 265-288.
- Zhao, Q., & Hastie, T. (2021). Causal interpretations of black-box models. *Journal of Business & Economic Statistics*, 39(1), 272-281.