Icon

Description automatically generated

**Spring, 2022-2023**

**SE2226 - Software Quality Assurance and Testing**

**Term Project Report**

|  |  |
| --- | --- |
| Project Name | Amazon Website Testing |
| Student Name | Pelin Sude Kırlı, Hande Hazan Keskin, Mert Kahraman |
| Student No | 20070006023, 21070006014, 20070006020 |
| Department | Software Engineering |
| Course Section No | 1 |

|  |  |
| --- | --- |
| **Contents** | **Page** |
| Introduction | 2 |
| Black-Box Testing Methods | 2-6 |
| White-Box Testing Method | 7 |
| Usability Testing | 7 |
| Performance Testing | 7-8 |

**Introduction:**

Amazon is a website (https://www.amazon.com.tr) which is used for online shopping world-wide. It is found in 1994 and it is one of the biggest sites of the world. On the website, there are nearly 300 million members and more than 353 million products in multiple categories such as fashion, consumer electronics, home & furniture, grocery and lifestyle products.

In our project, We tested Amazon website in these categories:

1. Black-Box Testing
   1. Testing Report
   2. Decision Table
   3. Equivalence Table
   4. Boundary Analysis
   5. Use Case Testing
2. White-Box Technique

2.1 Test for Java Selenium, ChromeDriver

1. Usability Testing
2. Performance Testing

4.1 Load Testing

**Testing:**

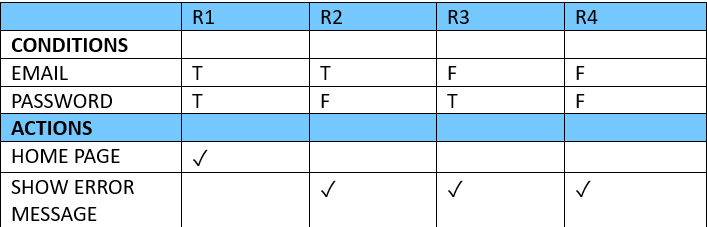
1. **Black-Box Testing:**

**1.1 Testing Report for Sign up Action:**

**Table

Description automatically generated-**Checkbox is necessary

**1.2 Decision Table for Log in Action:**

**Explanation:** In this decision table we have two conditions which are: Email and Password. Therefore, we have 22= 4 different rules. T symbolizes true, F symbolizes false. The tick symbol shows us the action as known as output.

* 1. **Equivalence Table:**

We decided to use equivalence partitioning on price filtering of Amazon’s Dizüstü Bilgisayar Category. Our goal is to find valid prices according to price boundaries, to make our item search easier. Price is computer’s price that placed in the given range. User click the range that he/she want to choose, and system displays the items.

**Table

Description automatically generated**

A screenshot of a computer

Description automatically generated with medium confidence

* 1. **Boundary Analysis:**

In this boundary analysis we have tested every boundary that takes place in our equivalence table.

Table

Description automatically generated

**1.5 Use Case Testing:**

Use Case

Use Case Name: Select and add item to the cart and delete it

Actor: Customer

Trigger: Customer wants to purchase items

Precondition: Customer must have an existing account

Normal Flow of Events

1. Customer enters to the Amazon’s sites login page.
2. Customer enters his/her e-mail.
3. Customer clicks to the “Devam et” button.
4. Customer enters his/her password.
5. Customer clicks on “Giriş Yap” button.
6. System verifies the e-mail and password and directs the customer to the home page.
7. Customer clicks on “Tümü” category.
8. Customer clicks on the displayed “Bilgisayar” button.
9. System directs the customer to the Bilgisayar page.
10. Customer chooses “Dizüstü Bilgisayarlar” button.
11. Customer chooses “Windows 11 Home” filter.
12. Customer sets a filter for price fields between 10000-32000.
13. Customer changes the filter to “Düşükten Yükseğe”.
14. System displays all the items ordered by ascending order.
15. Customer goes to second page.
16. Customer selects an item that he/she wants to purchase.
17. Customer looks every photo of the product.
18. Customer clicks the “Sepete Ekle” button to add an item to his/her cart.
19. System displays a screen to the customer which has “Şimdi Al” and “Sepete Ekle” options.
20. Customer clicks on the option “Sepete Ekle”.
21. Customer clicks on “Sepete Git” button.
22. System displays the purchase page.
23. Customer clicks on the “Sil” button.
24. System displays a confirmation message.

Alternate Flows:

6A1: Customer enters incorrect e-mail to login.

1. System displays an error message about wrong password or e-mail.
2. Customer enters the correct e-mail.
3. Use case continues.

6A2: Customer enters incorrect password to login.

1. System displays an error message about wrong password or e-mail.
2. Customer enters the correct password.
3. Use case continues.

7A1: Customer clicks on the “Moda” category button.

1. System displays the “Moda” category page.
2. Use case continues.

12A1: Customer sets a filter for price fields between 100-1000.

1. System filters the prices for 100-1000.
2. System displays the page that informs the customer there are no product according to their filters.
3. Customer clicks on home page button.
4. Use case ends.

12A2: Customer sets a filter for Sabit Sürücü Kapasitesi between 200-499 GB.

1. System filters by the Sabit Sürücü Kapasitesi and displays the items.
2. Use case continues.

20A1: Customer clicks on the option “Şimdi Al”.

1. Customer enter his/her billing information.
2. System confirms the payment.
3. Purchase has been made.
4. Use case ends.

Table, calendar

Description automatically generated

**A picture containing text, screenshot, number, line

Description automatically generated**TC1, TC3, and TC5 are automated by Selenium Web driver.

**2. White-Box Technique**

**2.1** Test for Java Selenium, ChromeDriver

We have used the Selenium library and ChromeDriver while automating out test cases. Scenarios that we have tested are given above the use case table TC1, TC3 and TC5. TC1 is the normal flow where customer logs in to the system successfully and after the necessary selections and filters, customer selects the computer, adds to the cart and then deletes from the cart. TC3 is an alternative flow where customer enters the incorrect password in the first try then in the second try enters it correctly then continues as the normal flow. TC5 is an alternative flow when customer selects different filtering options (price filter between 100 and 1000) and web application could not display any items according to the filters. We have started our automation in the login page of Amazon and we have continued according to the steps on the use case form. Here is the code of the normal flow, the rest of the code is in JavaSeleniumCode file.

A screenshot of a computer program

Description automatically generated with medium confidence

A screen shot of a computer program

Description automatically generated with low confidence



**3. Usability Testing:**

Website is tested for these browsers and seen that the website works correctly in these browsers:

- Google Chrome

- Mozilla Firefox

- Internet Explorer/Microsoft Edge

**4. Performance Testing**

Load Testing: We used Jmeter load test for Amazon Website. These screenshots are belong to 1000 number of samples.

metin, ekran görüntüsü, multimedya yazılımı, grafik yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

ekran görüntüsü, metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

A screenshot of a computer

Description automatically generated with medium confidence

Server Hostname: [www.amazon.com.tr](http://www.amazon.com.tr)

Time taken for tests: 27.4 seconds

Completed request: 1000

Failed request: 345

Average bytes: 99221.7 bytes

Longest request: 15850

Shortest request: 113