



UUM
Universiti Utara Malaysia

**STIW3034 SOFTWARE TESTING & QUALITY ASSURANCE
GROUP A
SEMESTER A211**

**GROUP PROJECT 2
IKEA WEBSITE**

**Submitted to:
Prof. Madya Ts. Dr. Nor Laily Binti Hashim**

Prepared by:

| No. | Name | Matric No. |
|------------|------------------------|-------------------|
| 1 | Wong Fang Man | 271221 |
| 2 | Yeoh Suan Ming | 271490 |
| 3 | Brandon Jude Mahendran | 272202 |

DATE OF SUBMISSION:

03 FEB 2022

**SCHOOL OF COMPUTING
UNIVERSITI UTARA MALAYSIA (UUM)
SINTOK, KEDAH, MALAYSIA**

Test Plan

1.0 Introduction

This project sets for a testing IKEA website. IKEA is a website that sells furniture and home accessories. IKEA has been developed in JavaScript with a backend programming language written using ASP.NET. The equivalence partitioning method will be used to achieve functional testing with the black box testing technique. Using this technique, test cases may be created that validate the system's accuracy in terms of testing requirements specifications and use cases.

1.1 Scope

1.1.1 In Scope

The project Test Plan defines the functional testing approach. The test scope includes the following:

- a. Testing of functional requirements listed according to IKEA features.

1.1.2 Out of Scope

Non-functional test namely:

- a. Error handling
- b. Performance
- c. Load test
- d. Stress tests
- e. Response time

1.2 Quality Objective

1.2.1 Primary Objective

The primary objective of testing IKEA website is to assure that the shopping gateway system meets the full requirements and satisfies the IKEA as well as market requirements needs of the website. At the end of the project development cycle, the users of the system can expect to use the IKEA website with proper function according to their expected requirement.

1.2.2 Secondary Objective

The secondary objective of testing IKEA website is to detect and expose all issues and potential problems, with the goal of ensuring that all concerns are properly handled before release. As a goal, this necessitates thorough and systematic testing to guarantee that all system functionality relevant to the user class is examined and, as a result, all issues (bugs) discovered are addressed effectively.

1.3 Roles and Responsibilities

Tester

The tester is responsible to:

- a. Develop test case
- b. Conduct functional testing

1.4 Assumptions

The tester is expected to have a thorough knowledge of the website content process, as well as test processes and document a defect and upgrade. The tester will be given a data set.

1.5 Definitions

Bugs: Any fault or flaw in the software/application or hardware that causes it to malfunction.

2.0 Features to be Tested

The following are some of the main features of the IKEA website:

- Register
- Login
- Search
- Shopping list
- Shopping cart
- Profile

3.0 Features not to be Tested

- Payments
- Track order
- FAQ
- Change language
- Delete profile
- Logout

4.0 Test Methodology

4.1 Purpose

4.1.1 Overview

The purpose of this **Test Plan** is to achieve the following goals:

1. To verify that all functional and design criteria are met in accordance with the documentation.
2. To elaborate on an approach.
3. To figure out what the documentation process takes.
4. To determine the testing methods.

4.1.2 Functional Testing

A defined set of objectives is set to ensure that defects discovered during testing are addressed effectively, maximising the iteration of test cycles. It is important to verify that the system satisfies the functional requirements. The system will be tested via black-box testing. The tester is responsible for creating test cases that force execution through various paths and sequences.

4.2 Item Pass/Fail Criteria

The test plan will describe the pass/fail criteria for each item in the test project. Every test case has an expected result. The results of each test item will be compared to the expected test results recorded in the test plan. If the actual result matches the expected result, the test case will be marked as a passing project. A test case is considered to have failed if the actual result produced by the execution of the test case does not match the expected result, in which case the actual result will be recorded in the test result. The source of the failure may be the application under test, the test case, the expected result, or the data in the test environment. Test case failures will be logged regardless of the source of the failure.

4.3 Suspension Criteria and Resumption Requirements

Suspension criteria specify the criteria to be used to suspend all or a portion of the testing activities while resumption criteria specify when testing can resume after it has been suspended.

- Unavailability of external dependent systems during execution.
- When a defect is introduced that cannot allow any further testing.
- Critical path deadline is missed so that the client will not accept delivery even if all testing is completed.

System Integration Testing in the Integration environment may be resumed under the following circumstances:

- When the external dependent systems become available again.
- When a fix is successfully implemented and the Testing Team is notified to continue testing.
- The contract is renegotiated with the client to extend delivery.

Suspension criteria assumes that testing cannot go forward and that going backward is also not possible. A failed build would not suffice as you could generally continue to use the previous build. Most major or critical defects would also not constitute suspension criteria as other areas of the system could continue to be tested.

4.4 Test Deliverables

The following are the deliverables of the test plan:

| No. | Deliverables |
|-----|--------------|
| 1 | Test plan |
| 2 | Test script |
| 3 | Test result |
| 4 | Code |

This document will list out the test cases to be tested and the test results to be executed. When the bugs are found during testing, the bugs will be marked, test status and the reason for failure will be listed in this document. Refer to 8.0 and 9.0 below for the test case and test result template used in the test.

4.5 Testing Tasks

4.5.1 Software Testing Process

The following test procedures will be used:

- a. Testing process
 - Testers will visit IKEA website and run predefined test cases to verify that the software complies with all normal, boundary and illegal test specifications. In the case of the link test and the html validation test, the testers will apply the Google Chrome driver tool to execute the IKEA website online link test.
 - Document any new problems discovered during testing.
- b. Problem Resolution
 - The bugs will be addressed and fixed according to the defect reporting mechanism described in Section 4.5.2 after they have been detected and documented.

4.5.2 Defect Reporting

Defect Tracking enables early discovery and subsequent monitoring of issues that may threaten a project's success. The following are included in Defect Tracking:

- Recognizing project issues
- Tracking and logging individual defects
- Reporting defect status
- Ongoing defect resolution

4.5.2.1 Severity of Defect

A severity code is assigned to each defect. The severity of a defect will impact how resources are assigned for resolution and what steps should be done if there are any issues that prevent the defect from being addressed before the deadline. The impact on the system/interface determines this priority code. The severity of a defect is classified as “Critical”, “High”, “Medium” or “Low”.

In software testing, the severity types of bugs/defects can be divided into four parts:

| Types of Severity | Characteristics |
|-------------------|--|
| 1 – Critical | This defect implies that the process has been completely shut down, and no further action can be taken. |
| 2 – High | A critical defect that causes a major crash where the system fails. However, some elements of the system still function. |
| 3 – Medium | It leads to some undesired behaviour, but the system remains functioning. |
| 4 – Low | It won't create any major system failures. |

4.5.2.2 Type of Defect

Defects are differences between the actual and expected results of a system or software application. A defect may also be defined as any variation or irregularity from the specifications. Developers are the ones who generate defects throughout the software development process. When a developer or programmer makes an error during the development phase, a defect is created. Each defect is allocated to one or more categories, which are collections of events that are tracked for trends and implications on current and future works. When a tester discovers a defect, he or she classifies it into one or more categories. The developer will then fix that problem.

4.5.2.3 Commutation of Defects

The process of logging and resolving defects involves several steps. The process starts with the execution of a test case and the finding that the actual results differ from all those expected. The tester will create a list of the defects and evaluate their severity and priority. The tester will then record the defect into the defect tracking database using the above categories. The tester will keep records of the procedures required to reproduce the error in a database. During the formal system-testing process, the tester will identify a solution to the defect and retest the issue. The timetable for resolution will be determined by the severity. Any defects with a severity rating of 1 are addressed right immediately. Defects with a severity rating of 2 will be addressed the same day. Defects of severity 3 and 4 will be addressed later.

5.0 Environmental Needs

5.1 Hardware

The hardware used to run test cases:

- Intel Core i5, 5-th Generation based PCs

5.2 Software

The software used to run test cases:

- Browser: Google Chrome version 97.0.4692.71

5.3 Testing Tool & Environments

Testing will be conducted in the following environments:

| Priority | Browser/Environment | Version | Comments |
|----------|---------------------|--------------|----------|
| 1 | Google Chrome | 97.0.4692.71 | |
| 2 | Firefox | 96.0 | |

Windows 10 PC, Eclipse software version 2019-06, Java JDK version 8 and Selenium web driver version 4.0.0.

6.0 Schedule

| No. | Deliverable | Time (Days) | Planned Date | Date Completed |
|-----|----------------------------|-------------|--------------|----------------|
| 1 | Test Plan | 14 | 30/12/2021 | 12/01/2022 |
| 2 | Functional Testing Results | 10 | 13/01/2022 | 22/01/2022 |
| 3 | Final Report | 4 | 23/01/2022 | 26/01/2022 |

7.0 Risks and Contingencies

N/A

8.0 Test Cases

Available in the Test Case document

9.0 Test Result Template

A. Test Results by Test Case

[illegible]

B. Overall Test Result 1

| Summary | Test Status | | # of Steps | # of Steps Executed | # of Steps Passed | # of Steps Failed |
|---------|-------------|--------|------------|---------------------|-------------------|-------------------|
| | Passed | Failed | | | | |
| Number | | | | | | |
| Percent | | | | | | |

C. Overall Test Result 2

[illegible]

Software Test Result (STR)

1.0 Scope of the Tests

The website to be tested:

Website: www.ikea.com

2.0 Test Environments (for each test)

2.1 Test Identification

The test details are documented in the STP.

2.2 Detailed Description of the Operating System, Hardware Configuration and the Required Switch Settings for the Tests:

- Operating system: Windows 10
- Browser: Google Chrome version 97.0.4692.71
- Hardware configuration:
 - Intel Core i5 7-th Generation Processor
 - RAM 8GB
 - Solid State Disk 500GB

2.3 Instructions for Software Loading

1. Access the Internet
2. Write the IKEA URL: www.ikea.com

3.0 Testing Process

1. Short Description: **TC01** - Test the register requirement
 - Click on the “Create Account” button.
 - Enter first name “John”, last name “Doe”, email “group6softwaretesting@gmail.com”, password “Welcome01\$” and tick the privacy policy box.
 - Click the “Create profile” button.

2. Short Description: **TC02** - Test the login requirement
 - Click on the “Login” button.
 - Enter Email “abc763188@gmail.com” and Password “Abc123456” and click the “Login” button.

3. Short Description: **TC03** - Test the search requirement
 - Click on the Search Box.
 - Enter product names such as wardrobe, double-beds, and so on.
 - Click on the “Search” icon.

4. Short Description: **TC04** - Test the shopping list requirement
 - Click on the Search Box.
 - Enter product name “RUMPLING” in the search box.
 - Click on the product (carrier bag).
 - Click on the “Shopping List” icon.

5. Short Description: **TC05** - Test the shopping cart requirement
 - Click on the Search Box.
 - Enter product name “BACKTRAV” in the search box.
 - Click on the product (bolster).
 - Click on the “Add to cart” button

6. Short Description: **TC06** - Test the profile requirement
 - Click on the “My profile” icon.
 - Click on the “Edit” button of Personal Information.
 - Enter First Name “Test” and Last Name “Test” and click on the “Save changes” button.

4.0 Test Cases

| | | | | |
|--|--|---|-----------|---------|
| Test Case #: TC01.01 | | Test Case Name: Register | | |
| System: IKEA | | | | |
| Designed By: Brandon Jude Mahendran | | Design Date: 11/01/2022 | | |
| Executed By: Wong Fang Man | | Execution Date: 19/01/2022 | | |
| Short Description: Test the register requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user should have access to the IKEA website's Sign-Up page. | | | | |
| Overall Pass Criteria: The user is able to sign in into the IKEA account with a valid email address. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the "Create Account" button. | IKEA will redirect to the Sign-Up page. | Pass | |
| 2. | Enter first name "John", last name "Doe", email "group6softwaretesting@gmail.com", password "Welcome01\$" and tick the privacy policy box. | IKEA will verify all credentials entered. | Pass | |
| 3. | Click the "Create profile" button. | IKEA will redirect to the Dashboard page. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully created an IKEA account. | | | | |

| | | | | |
|--|--|---|-----------|---------|
| Test Case #: TC01.02 | | Test Case Name: Register | | |
| System: IKEA | | | | |
| Designed By: Brandon Jude Mahendran | | Design Date: 11/01/2022 | | |
| Executed By: Wong Fang Man | | Execution Date: 19/01/2022 | | |
| Short Description: Test the register requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user should have access to the IKEA website’s Sign-Up page. | | | | |
| Overall Pass Criteria: The user is able to sign in into the IKEA account with a dummy email address. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Create Account” button. | IKEA will redirect to the Sign-Up page. | Pass | |
| 2. | Enter first name “John”, last name “Doe”, email “blablablaj998@dummy.com”, password “Welcome01\$” and tick the privacy policy box. | IKEA will verify all credentials entered. | Pass | |
| 3. | Click the “Create profile” button. | IKEA will redirect to the Dashboard page. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully created an IKEA account. | | | | |

| | | | | |
|---|---|---|-----------|---------|
| Test Case #: TC01.03 | | Test Case Name: Register | | |
| System: IKEA | | | | |
| Designed By: Brandon Jude Mahendran | | Design Date: 11/01/2022 | | |
| Executed By: Wong Fang Man | | Execution Date: 19/01/2022 | | |
| Short Description: Test the register requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user should have access to the IKEA website's Sign-Up page. | | | | |
| Overall Pass Criteria: The user is unable to create an account without providing the first name. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the "Create Account" button. | IKEA will redirect to the Sign-Up page. | Pass | |
| 2. | Enter first name " ", last name "Doe", email "group6softwaretesting@gmail.com", password "Welcome01\$" and tick the privacy policy box. | IKEA will verify all credentials entered. | Pass | |
| 3. | Click the "Create profile" button. | IKEA will prompt users to enter their first name. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user is unable to create an account and IKEA will prompt the user to enter the first name. | | | | |

| | | | | |
|--|--|--|-----------|---------|
| Test Case #: TC01.04 | | Test Case Name: Register | | |
| System: IKEA | | | | |
| Designed By: Brandon Jude Mahendran | | Design Date: 11/01/2022 | | |
| Executed By: Wong Fang Man | | Execution Date: 19/01/2022 | | |
| Short Description: Test the register requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user should have access to the IKEA website's Sign-Up page. | | | | |
| Overall Pass Criteria: The user is unable to create an account without providing the last name. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Create Account” button. | IKEA will redirect to the Sign-Up page. | Pass | |
| 2. | Enter first name “John”, last name “ ”, email “group6softwaretesting@gmail.com”, password “Welcome01\$” and tick the privacy policy box. | IKEA will verify all credentials entered. | Pass | |
| 3. | Click the “Create profile” button. | IKEA will prompt users to enter their last name. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user is unable to create an account and IKEA will prompt the user to enter the last name. | | | | |

| | | | | |
|--|---|---|-----------|---------|
| Test Case #: TC01.05 | | Test Case Name: Register | | |
| System: IKEA | | | | |
| Designed By: Brandon Jude Mahendran | | Design Date: 11/01/2022 | | |
| Executed By: Wong Fang Man | | Execution Date: 19/01/2022 | | |
| Short Description: Test the register requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user should have access to the IKEA website’s Sign-Up page. | | | | |
| Overall Pass Criteria: The user is unable to create an account when input first name using digits or special characters. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Create Account” button. | IKEA will redirect to the Sign-Up page. | Pass | |
| 2. | Enter first name “12345\$”, last name “Doe”, email “group6softwaretesting@gmail.com”, password “Welcome01\$” and tick the privacy policy box. | IKEA will verify all credentials entered. | Pass | |
| 3. | Click the “Create profile” button. | IKEA will prompt users to enter their proper first name using characters. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user is unable to create an account and IKEA will prompt the user to enter proper first name using characters. | | | | |

| | | | | |
|---|--|--|-----------|---------|
| Test Case #: TC01.06 | | Test Case Name: Register | | |
| System: IKEA | | | | |
| Designed By: Brandon Jude Mahendran | | Design Date: 11/01/2022 | | |
| Executed By: Wong Fang Man | | Execution Date: 19/01/2022 | | |
| Short Description: Test the register requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user should have access to the IKEA website's Sign-Up page. | | | | |
| Overall Pass Criteria: The user is unable to create an account when input last name using digits or special characters. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the "Create Account" button. | IKEA will redirect to the Sign-Up page. | Pass | |
| 2. | Enter first name "John", last name "12345\$", email "group6softwaretesting@gmail.com", password "Welcome01\$" and tick the privacy policy box. | IKEA will verify all credentials entered. | Pass | |
| 3. | Click the "Create profile" button. | IKEA will prompt users to enter their proper last name using characters. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user is unable to create an account and IKEA will prompt the user to enter a proper last name using characters. | | | | |

| | | | | |
|---|--|---|-----------|---------|
| Test Case #: TC02.01 | | Test Case Name: Login | | |
| System: IKEA | | | | |
| Designed By: Brandon Jude Mahendran | | Design Date: 11/01/2022 | | |
| Executed By: Wong Fang Man | | Execution Date: 20/01/2022 | | |
| Short Description: Test the login requirement. | | | | |
| Pre-conditions: | | | | |
| 1. User should have successfully created an IKEA account. | | | | |
| Overall Pass Criteria: The user is unable to login into IKEA account with a wrong format email address. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Login” link. | IKEA will redirect to the Login page. | Pass | |
| 2. | Enter email “group6softwaretesting” and password “Welcome01\$” | IKEA will verify all credentials entered. | Pass | |
| 3. | Click on the “Login” button. | IKEA will display error message. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user is unable to log in into the IKEA website. | | | | |

| | | | | |
|---|--|---|-----------|---------|
| Test Case #: TC02.02 | | Test Case Name: Login | | |
| System: IKEA | | | | |
| Designed By: Brandon Jude Mahendran | | Design Date: 11/01/2022 | | |
| Executed By: Wong Fang Man | | Execution Date: 20/01/2022 | | |
| Short Description: Test the login requirement. | | | | |
| Pre-conditions: | | | | |
| 1. User should have successfully created an IKEA account. | | | | |
| Overall Pass Criteria: The user is unable to login into IKEA account with a wrong password. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Login” link. | IKEA will redirect to the Login page. | Pass | |
| 2. | Enter email “group6softwaretesting@gmail.com” and password “wrongpassword” | IKEA will verify all credentials entered. | Pass | |
| 3. | Click on the “Login” button. | IKEA will display error message. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user is unable to log in into the IKEA website. | | | | |

| | | | | |
|--|--|--|-----------|---------|
| Test Case #: TC02.03 | | Test Case Name: Login | | |
| System: IKEA | | | | |
| Designed By: Brandon Jude Mahendran | | Design Date: 11/01/2022 | | |
| Executed By: Wong Fang Man | | Execution Date: 20/01/2022 | | |
| Short Description: Test the login requirement when the user clicks on the forget password using a valid email address. | | | | |
| Pre-conditions: | | | | |
| 1. User should have successfully created an IKEA account. | | | | |
| Overall Pass Criteria: The user successfully requests to reset the password using a valid email address. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Forgot your password?” link. | IKEA will redirect to the reset page. | Pass | |
| 2. | Enter email address “group6softwaretesting@gmail.com”. | IKEA will verify credentials entered. | Pass | |
| 3. | Click on the “Send new password” button. | IKEA will redirect to the reset success page | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully requests to reset the password using a valid email address. | | | | |

| | | | | |
|--|---|--|-----------|---------|
| Test Case #: TC02.04 | | Test Case Name: Login | | |
| System: IKEA | | | | |
| Designed By: Brandon Jude Mahendran | | Design Date: 11/01/2022 | | |
| Executed By: Wong Fang Man | | Execution Date: 20/01/2022 | | |
| Short Description: Test the login requirement when the user clicks on the forget password using invalid email address. | | | | |
| Pre-conditions: | | | | |
| 1. User should have successfully created an IKEA account. | | | | |
| Overall Pass Criteria: The user successfully requests to reset the password using a dummy email address. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Forgot your password?” link. | IKEA will redirect to the reset page. | Pass | |
| 2. | Enter email address ”blablabla998@dummy.com”. | IKEA will verify credentials entered. | Pass | |
| 3. | Click on the “Send new password” button. | IKEA will redirect to the reset success page | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully requests to reset the password using a dummy email address. | | | | |

| | | | | |
|---|---|---|-----------|---------|
| Test Case #: TC02.05 | | Test Case Name: Login | | |
| System: IKEA | | | | |
| Designed By: Yeoh Suan Ming | | Design Date: 11/01/2022 | | |
| Executed By: Brandon Jude Mahendran | | Execution Date: 17/01/2022 | | |
| Short Description: Test the login requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user should have access to the IKEA website's login page. | | | | |
| Overall Pass Criteria: The user is able to login into IKEA account. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the "Login" button. | IKEA will redirect to the login page. | Pass | |
| 2. | Enter email "abc763188@ gmail.com" and password "Abc123456" and click the "Login" button. | IKEA shall authenticate the user and login. | Pass | |
| 3. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully accessed the IKEA dashboard. | | | | |

| | | | | |
|---|---|---|-----------|---------|
| Test Case #: TC03.01 | | Test Case Name: Search | | |
| System: IKEA | | Design Date: 11/01/2022 | | |
| Designed By: Yeoh Suan Ming | | Execution Date: 17/01/2022 | | |
| Executed By: Brandon Jude Mahendran | | | | |
| Short Description: Test the search requirement. | | | | |
| Pre-conditions: | | | | |
| 1. Not applicable. | | | | |
| Overall Pass Criteria: The user is able to search products with text. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the Search Box. | IKEA will display the search history and the product that they recommend. | Pass | |
| 2. | Enter product names such as wardrobe, double-beds, and so on. | IKEA will display the products that are related. | Pass | |
| 3. | Click on the “Search” icon. | IKEA will display the products that users search for. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully searched the product. | | | | |

| | | | | |
|---|--|---|-----------|-------------------------------------|
| Test Case #: TC03.02 | | Test Case Name: Search | | |
| System: IKEA | | | | |
| Designed By: Yeoh Suan Ming | | Design Date: 11/01/2022 | | |
| Executed By: Brandon Jude Mahendran | | Execution Date: 17/01/2022 | | |
| Short Description: Test the search requirement. | | | | |
| Pre-conditions: | | | | |
| 1. Not applicable. | | | | |
| Overall Pass Criteria: The user is able to search products with pictures. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the Search Box. | IKEA will display the search history and the product that they recommend. | Pass | |
| 2. | Click on the “Camera” icon. | IKEA will display a search column for photos. | Pass | |
| 3. | Click on the “Take or update a photo” button. | IKEA will display the file from the laptop. | Pass | |
| 4. | Select the “wardrobe.JPG” and click on the “Open” button to get the photo from the laptop. | IKEA will display similar products. | Fail | |
| 5. | Check the post-condition. | | Fail | Can only be perform by manual test. |
| Post-conditions: | | | | |
| 1. The user successfully searched the product. | | | | |

| | | | | |
|---|--|---|-----------|---------|
| Test Case #: TC03.03 | | Test Case Name: Search | | |
| System: IKEA | | | | |
| Designed By: Yeoh Suan Ming | | Design Date: 11/01/2022 | | |
| Executed By: Brandon Jude Mahendran | | Execution Date: 17/01/2022 | | |
| Short Description: Test the search requirement. | | | | |
| Pre-conditions: | | | | |
| 1. Not applicable. | | | | |
| Overall Pass Criteria: The user is able to search for a store with valid postal code. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Select Store” | IKEA will display a search column for stores. | Pass | |
| 2. | Enter valid Postal Code (example: 14000) and click on the “Find” button. | IKEA will display a searched store. | Pass | |
| 3. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully searched the closest store. | | | | |

| | | | | |
|---|--|---|-----------|---------|
| Test Case #: TC03.04 | | Test Case Name: Search | | |
| System: IKEA | | | | |
| Designed By: Yeoh Suan Ming | | Design Date: 11/01/2022 | | |
| Executed By: Brandon Jude Mahendran | | Execution Date: 18/01/2022 | | |
| Short Description: Test the search requirement. | | | | |
| Pre-conditions: | | | | |
| 1. Not applicable. | | | | |
| Overall Pass Criteria: The user is able to search for a store with invalid postal code. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Select Store” | IKEA will display a search column for stores. | Pass | |
| 2. | Enter invalid Postal Code (example: 14999) and click on the “Find” button. | IKEA will display an error message “We're sorry, but it appears we were unable to find the store closest to you.” | Pass | |
| 3. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user is unable to search the closest store. | | | | |

| | | | | |
|---|--|---|-----------|---------|
| Test Case #: TC04.01 | | Test Case Name: Shopping List | | |
| System: IKEA | | | | |
| Designed By: Yeoh Suan Ming | | Design Date: 11/01/2022 | | |
| Executed By: Brandon Jude Mahendran | | Execution Date: 18/01/2022 | | |
| Short Description: Tests the shopping list requirement. | | | | |
| Pre-conditions: | | | | |
| 1. Not Applicable. | | | | |
| Overall Pass Criteria: The user is able to add products to the shopping list. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the Search Box. | IKEA will display the search history and the product that they recommend. | Pass | |
| 2. | Enter product name “RUMPLING” in the search box. | IKEA will display the searched product. | Pass | |
| 3. | Click on the product (carrier bag). | IKEA will display the product details of the product. | Pass | |
| 4. | Click on the “Shopping List” icon. | IKEA will display a successful message “RUMPLING was added to your wishlist”. | Pass | |
| 5. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully added a product into the shopping list. | | | | |

| | | | | |
|--|---|---|-----------|---------|
| Test Case #: TC04.02 | | Test Case Name: Shopping List | | |
| System: IKEA | | | | |
| Designed By: Yeoh Suan Ming | | Design Date: 11/01/2022 | | |
| Executed By: Brandon Jude Mahendran | | Execution Date: 18/01/2022 | | |
| Short Description: Tests the shopping list requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user should add some products into the shopping list. | | | | |
| Overall Pass Criteria: The user is able to delete products from the shopping list. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Shopping List” icon. | IKEA will redirect to the shopping list page. | Pass | |
| 2. | Click on the “Delete” icon next to the product that you want to delete. | IKEA will delete the product. | Pass | |
| 3. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully deleted a product in the shopping list. | | | | |

| | | | | |
|--|--|--|-----------|---------|
| Test Case #: TC04.03 | | Test Case Name: Shopping List | | |
| System: IKEA | | | | |
| Designed By: Yeoh Suan Ming | | Design Date: 11/01/2022 | | |
| Executed By: Brandon Jude Mahendran | | Execution Date: 18/01/2022 | | |
| Short Description: Tests the shopping list requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user should add some products into the shopping list. | | | | |
| Overall Pass Criteria: The user is able to change the quantity of the product. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Shopping List” icon. | IKEA will redirect to the shopping list page. | Pass | |
| 2. | Click on the drop-down menu from the quantity. | IKEA will display a list of numbers “1-999”. | Pass | |
| 3. | Click on the number “3”. | IKEA will change the number and auto calculate the total price of the order. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully changed the quantity of a product in the shopping list. | | | | |

| | | | | |
|---|--|--|-----------|---------|
| Test Case #: TC04.04 | | Test Case Name: Shopping List | | |
| System: IKEA | | | | |
| Designed By: Yeoh Suan Ming | | Design Date: 11/01/2022 | | |
| Executed By: Brandon Jude Mahendran | | Execution Date: 18/01/2022 | | |
| Short Description: Tests the shopping list requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user should add some products into the shopping list. | | | | |
| Overall Pass Criteria: The user is able to add the products into the shopping list. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Shopping List” icon. | IKEA will redirect to the shopping list page. | Pass | |
| 2. | Click on the “Add to shopping cart” button next to the product that you want to add. | IKEA will display a successful message “Item added”. | Pass | |
| 3. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully added a product to the shopping cart. | | | | |

| | | | | |
|--|---|---|-----------|--|
| Test Case #: TC04.05 | | Test Case Name: Shopping List | | |
| System: IKEA | | | | |
| Designed By: Yeoh Suan Ming | | Design Date: 11/01/2022 | | |
| Executed By: Brandon Jude Mahendran | | Execution Date: 18/01/2022 | | |
| Short Description: Tests the shopping list requirement. | | | | |
| Pre-conditions: | | | | |
| 1. Not applicable. | | | | |
| Overall Pass Criteria: The user is able to print the shopping list page in a pdf file. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Shopping List” icon. | IKEA will redirect to the shopping list page. | Pass | |
| 2. | Click on the “Printer” button and click on the “Save” button. | IKEA will display the print dialog and save it into the laptop. | Fail | |
| 3. | Check the post-condition. | | Fail | Browsers use native dialogs for it which cannot be controlled by JavaScript. |
| Post-conditions: | | | | |
| 1. The user successfully saved the shopping list in a pdf file. | | | | |

| | | | | |
|---|--|---|-----------|---------|
| Test Case #: TC04.06 | | Test Case Name: Shopping List | | |
| System: IKEA | | | | |
| Designed By: Wong Fang Man | | Design Date: 11/01/2022 | | |
| Executed By: Yeoh Suan Ming | | Execution Date: 18/01/2022 | | |
| Short Description: Tests the shopping list requirement. | | | | |
| Pre-conditions: | | | | |
| 1. Not applicable. | | | | |
| Overall Pass Criteria: The user is able to create a new list. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Shopping List” icon. | IKEA will redirect to the shopping list page. | Pass | |
| 2. | Click on the drop-down menu from the shopping list. | IKEA will display the lists. | Pass | |
| 3. | Click on the “Create a new list”. | IKEA will display a text field for users to enter the name of the list. | Pass | |
| 4. | Enter the name of the list “Toy List” and click on the “Create list” button. | IKEA will add a new list to “Your Lists”. | Pass | |
| 5. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully creates a new list. | | | | |

| | | | | |
|--|---|--|-----------|---------|
| Test Case #: TC04.07 | | Test Case Name: Shopping List | | |
| System: IKEA | | Design Date: 11/01/2022 | | |
| Designed By: Wong Fang Man | | Execution Date: 18/01/2022 | | |
| Executed By: Yeoh Suan Ming | | | | |
| Short Description: Tests the shopping list requirement. | | | | |
| Pre-conditions: | | | | |
| 1. Not applicable. | | | | |
| Overall Pass Criteria: The user is able to delete the selected list. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Shopping List” icon. | IKEA will redirect to the shopping list page. | Pass | |
| 2. | Click on the drop-down menu from the shopping list. | IKEA will display the lists. | Pass | |
| 3. | Click on the “Delete” icon next to the “Toy List”. | IKEA will display a confirmation message “Are you sure you want to delete this list?”. | Pass | |
| 4. | Click on the “Delete” button. | IKEA will remove the list from “Your Lists”. | Pass | |
| 5. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully deleted a list. | | | | |

| | | | | |
|--|--|--|-----------|---------|
| Test Case #: TC05.01 | | Test Case Name: Shopping Cart | | |
| System: IKEA | | | | |
| Designed By: Wong Fang Man | | Design Date: 11/01/2022 | | |
| Executed By: Yeoh Suan Ming | | Execution Date: 20/01/2022 | | |
| Short Description: Tests the shopping cart requirement. | | | | |
| Pre-conditions: | | | | |
| 1. Not applicable. | | | | |
| Overall Pass Criteria: The user is able to add a product into the shopping cart. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the Search Box. | IKEA will display the search history and the product that they recommend. | Pass | |
| 2. | Enter product name “BACKTRAV” in the search box. | IKEA will display the searched product. | Pass | |
| 3. | Click on the product (bolster). | IKEA will display the product details of the product. | Pass | |
| 4. | Click on the “Add to cart” button | IKEA will display a successful message “BACKTRAV was added to your shopping cart”. | Pass | |
| 5. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully added a product into the shopping cart. | | | | |

| | | | | |
|---|--|--|-----------|---------|
| Test Case #: TC05.02 | | Test Case Name: Shopping Cart | | |
| System: IKEA | | | | |
| Designed By: Wong Fang Man | | Design Date: 11/01/2022 | | |
| Executed By: Yeoh Suan Ming | | Execution Date: 20/01/2022 | | |
| Short Description: Tests the shopping cart requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user should add some products into the shopping cart. | | | | |
| Overall Pass Criteria: The user is able to change the quantity of a product in the shopping cart. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Shopping Cart” icon. | IKEA will redirect to the shopping cart page. | Pass | |
| 2. | Click on the drop-down menu from the quantity. | IKEA will display a list of numbers “1-999”. | Pass | |
| 3. | Click on the number “2”. | IKEA will change the number and auto calculate the total price of the order. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully changed the quantity of a product in the shopping cart. | | | | |

| | | | | |
|---|---|---|-----------|---------|
| Test Case #: TC05.03 | | Test Case Name: Shopping Cart | | |
| System: IKEA | | | | |
| Designed By: Wong Fang Man | | Design Date: 11/01/2022 | | |
| Executed By: Yeoh Suan Ming | | Execution Date: 20/01/2022 | | |
| Short Description: Tests the shopping cart requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user should add some products into the shopping cart. | | | | |
| Overall Pass Criteria: The user is able to delete the product in the shopping cart. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Shopping Cart” icon. | IKEA will redirect to the shopping cart page. | Pass | |
| 2. | Click on the “Delete” icon next to the product that you want to delete. | IKEA will delete the product. | Pass | |
| 3. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully deleted the product in the shopping cart. | | | | |

| | | | | |
|--|---|---|-----------|---------|
| Test Case #: TC05.04 | | Test Case Name: Shopping Cart | | |
| System: IKEA | | | | |
| Designed By: Wong Fang Man | | Design Date: 11/01/2022 | | |
| Executed By: Yeoh Suan Ming | | Execution Date: 20/01/2022 | | |
| Short Description: Tests the shopping cart requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user should add some products into the shopping cart. | | | | |
| Overall Pass Criteria: The user is able to delete all the products in the shopping cart. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “Shopping Cart” icon. | IKEA will redirect to the shopping cart page. | Pass | |
| 2. | Click on the “Remove all products” button at the bottom of the products in the shopping cart. | IKEA will display a confirmation message “Do you really want to remove all items from your shopping cart?”. | Pass | |
| 3. | Click on the “Remove” button. | IKEA will remove all the items from the shopping cart. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully deleted all the products in the shopping cart. | | | | |

| | | | | |
|---|--|---|-----------|---------|
| Test Case #: TC06.01 | | Test Case Name: Profile | | |
| System: IKEA | | | | |
| Designed By: Wong Fang Man | | Design Date: 11/01/2022 | | |
| Executed By: Yeoh Suan Ming | | Execution Date: 21/01/2022 | | |
| Short Description: Tests the profile requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user is already logged in to the account. | | | | |
| Overall Pass Criteria: The user is able to change the personal information. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “My profile” icon. | IKEA will redirect to the profile page. | Pass | |
| 2. | Click on the “Edit” button of Personal Information. | IKEA will display the personal information which are first name and last name. | Pass | |
| 3. | Enter First Name “Test” and Last Name “Test” and click on the “Save changes” button. | IKEA will save the first name and last name and display the name at the Personal Information. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user successfully changed the personal information. | | | | |

| | | | | |
|---|---|---|-----------|---------|
| Test Case #: TC06.02 | | Test Case Name: Profile | | |
| System: IKEA | | Design Date: 11/01/2022 | | |
| Designed By: Wong Fang Man | | Execution Date: 21/01/2022 | | |
| Executed By: Yeoh Suan Ming | | | | |
| Short Description: Tests the profile requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user is already logged in to the account. | | | | |
| Overall Pass Criteria: The user is unable to change the personal information. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “My profile” icon. | IKEA will redirect to the profile page. | Pass | |
| 2. | Click on the “Edit” button of Personal Information. | IKEA will display the personal information which are first name and last name. | Pass | |
| 3. | Enter First Name “Test” and Last Name “123” and click on the “Save changes” button. | IKEA will display the error message “Your last name must not contain <>{}[]()*=*; or be only spaces, numbers or special characters.”. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user is unable to change the personal information. | | | | |

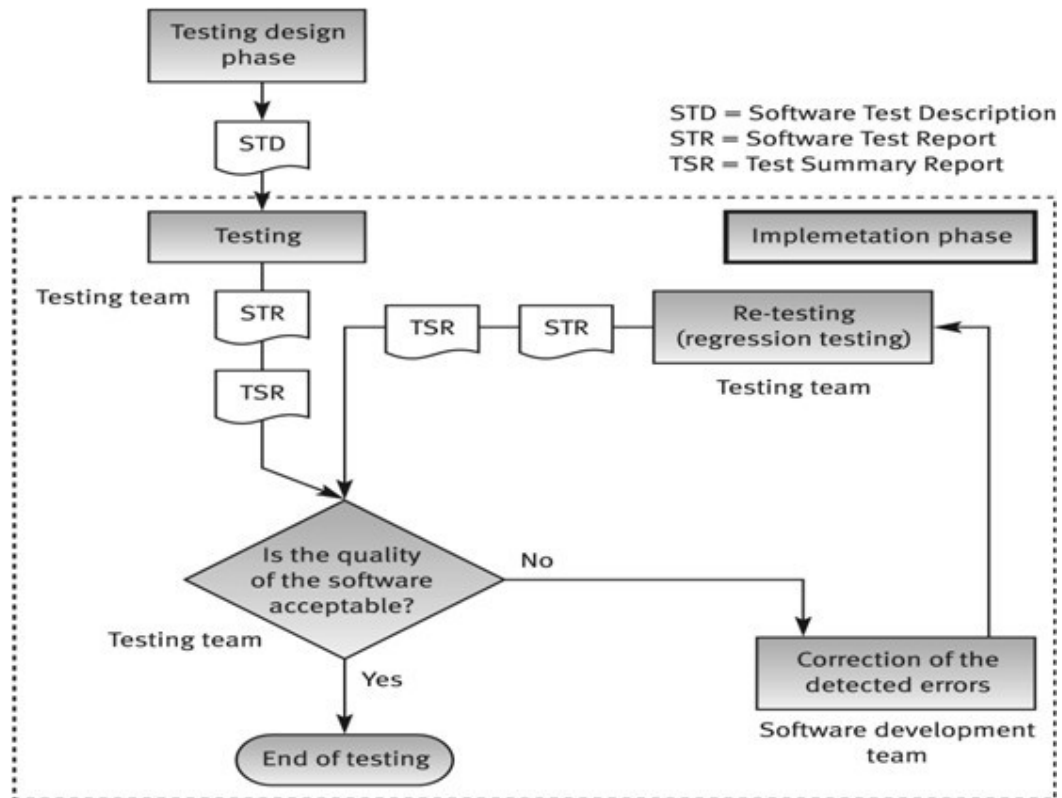
| | | | | |
|--|--|--|-----------|---------|
| Test Case #: TC06.03 | | Test Case Name: Profile | | |
| System: IKEA | | Design Date: 11/01/2022 | | |
| Designed By: Wong Fang Man | | Execution Date: 21/01/2022 | | |
| Executed By: Yeoh Suan Ming | | | | |
| Short Description: Tests the profile requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user is already logged in to the account. | | | | |
| Overall Pass Criteria: The user is unable to change the email. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “My profile” icon. | IKEA will redirect to the profile page. | Pass | |
| 2. | Click on the “Edit” button of Contact. | IKEA will display the text field of the email. | Pass | |
| 3. | Enter Email “test1234@gmail.com” and Password “Test1234” and click on the “Save changes” button. | IKEA will display an error message “Password is incorrect.”. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user is unable to change the email. | | | | |

| | | | | |
|---|--|--|-----------|---------|
| Test Case #: TC06.04 | | Test Case Name: Profile | | |
| System: IKEA | | | | |
| Designed By: Wong Fang Man | | Design Date: 11/01/2022 | | |
| Executed By: Yeoh Suan Ming | | Execution Date: 21/01/2022 | | |
| Short Description: Tests the profile requirement. | | | | |
| Pre-conditions: | | | | |
| 1. The user is already logged in to the account. | | | | |
| Overall Pass Criteria: The user is unable to change the password. | | | | |
| Steps | Action | Expected System Response | Pass/Fail | Comment |
| 1. | Click on the “My profile” icon. | IKEA will redirect to the profile page. | Pass | |
| 2. | Click on the “Edit” button of Password. | IKEA will display the text field of current password, new password and confirm password. | Pass | |
| 3. | Enter Current Password “Test1234”, New Password “Test12345” and Confirm New Password “Test12345” and click on the “Save changes” button. | IKEA will display an error message “Password is incorrect.”. | Pass | |
| 4. | Check the post-condition. | | Pass | |
| Post-conditions: | | | | |
| 1. The user is unable to change the password. | | | | |

5.0 Actions to be taken in case of program failure/cessation

The testers should refer the report on program failure/cessation to the software developer.

6.0 Procedures to be applied according to the test results summary



7.0 Test Result

A. Test Results by Test Case

| | Test Case | Test Status | Reason for Failure | Passed | Failed |
|----|-----------|-------------|--------------------|--------|--------|
| 1. | TC01.01 | Pass | | 1 | |
| 2. | TC01.02 | Pass | | 1 | |
| 3. | TC01.03 | Pass | | 1 | |
| 4. | TC01.04 | Pass | | 1 | |

| | | | | | |
|-----|---------|--------|--|---|---|
| 5. | TC01.05 | Pass | | 1 | |
| 6. | TC01.06 | Pass | | 1 | |
| 7. | TC02.01 | Pass | | 1 | |
| 8. | TC02.02 | Pass | | 1 | |
| 9. | TC02.03 | Pass | | 1 | |
| 10. | TC02.04 | Pass | | 1 | |
| 11. | TC02.05 | Pass | | 1 | |
| 12. | TC03.01 | Pass | | 1 | |
| 13. | TC03.02 | Failed | Can only be perform by manual test. | | 1 |
| 14. | TC03.03 | Pass | | 1 | |
| 15. | TC03.04 | Pass | | 1 | |
| 16. | TC04.01 | Pass | | 1 | |
| 17. | TC04.02 | Pass | | 1 | |
| 18. | TC04.03 | Pass | | 1 | |
| 19. | TC04.04 | Pass | | 1 | |
| 20. | TC04.05 | Failed | Browsers use native dialogs for it which cannot be controlled by JavaScript. | | 1 |
| 21. | TC04.06 | Pass | | 1 | |
| 22. | TC04.07 | Pass | | 1 | |
| 23. | TC05.01 | Pass | | 1 | |
| 24. | TC05.02 | Failed | Browsers use native dialogs for it which cannot be controlled by JavaScript. | | 1 |
| 25. | TC05.03 | Failed | Browsers use native dialogs for it which cannot be controlled by JavaScript. | | 1 |
| 26. | TC05.04 | Failed | Browsers use native dialogs for it which cannot be controlled by JavaScript. | | 1 |
| 27. | TC06.01 | Pass | | 1 | |
| 28. | TC06.02 | Pass | | 1 | |
| 29. | TC06.03 | Pass | | 1 | |

| | | | | | |
|-------|---------|------|--|----|---|
| 30. | TC06.04 | Pass | | 1 | |
| Total | | | | 25 | 5 |

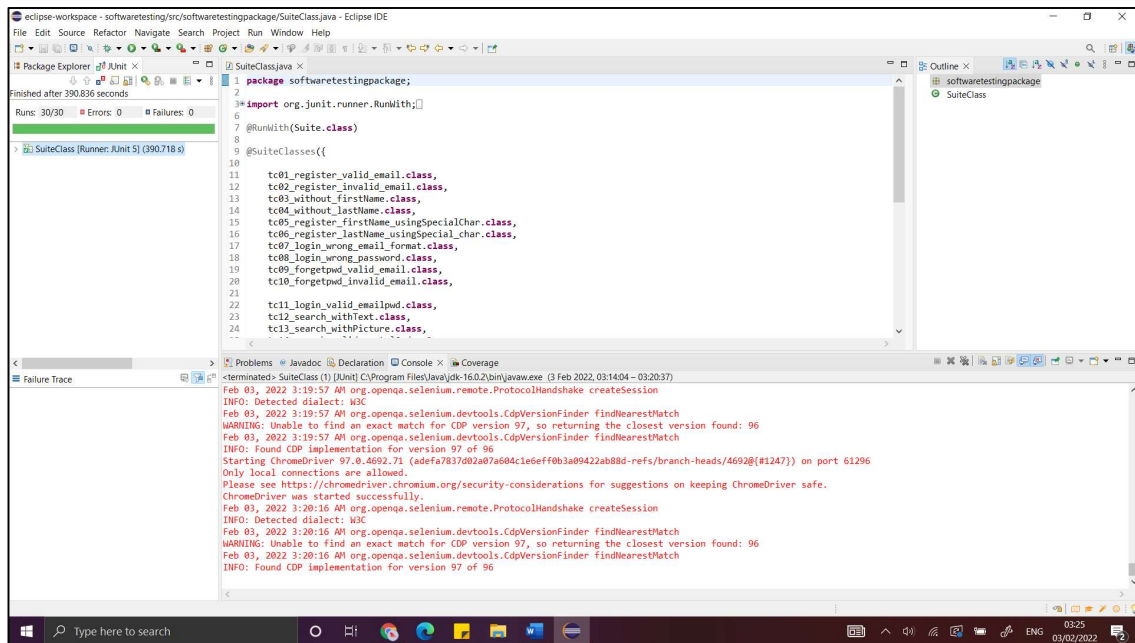
B. Overall Test Result 1

| Summary | Test Status | | # of Steps | # of Steps Executed | # of Steps Passed | # of Steps Failed |
|---------|-------------|--------|------------|---------------------|-------------------|-------------------|
| | Passed | Failed | | | | |
| Number | 25 | 5 | 118 | 118 | 111 | 7 |
| Percent | 83.33% | 16.67% | 100% | 100% | 94% | 6% |

C. Overall Test Result 2

| Summary | # of Test Cases Passed / Total # of Test Cases | # of Test Cases Failed / Total # of Test Cases | Total # of Steps / Total # of Steps Executed | # of Steps Passed / # of Steps Executed | # of Steps Failed / # of Steps Executed | Notes | | | |
|------------|--|--|--|---|---|-------------|------|---|--------------|
| Proportion | 5/6 | 1/6 | 1 | 111/118 | 7/118 | Date Tested | Ver. | Tested By | Notes |
| Percent | 83.33 | 16.67 % | 100% | 94% | 6% | 22/01/2022 | 97.0 | Wong Fang Man, Yeoh Suan Ming & Brandon | Satisfactory |

Selenium Source Code for Web Testing IKEA Website



```
package softwaretestingpackage;
```

```
import org.junit.runner.RunWith;
```

```
import org.junit.runners.Suite;
```

```
import org.junit.runners.Suite.SuiteClasses;
```

```
@RunWith(Suite.class)
```

```
@SuiteClasses({
```

```
    tc01_register_valid_email.class,
```

```
    tc02_register_invalid_email.class,
```

```
    tc03_without_firstName.class,
```

```
    tc04_without_lastName.class,
```

```
    tc05_register_firstName_usingSpecialChar.class,
```

```
    tc06_register_lastName_usingSpecial_char.class,
```

tc07_login_wrong_email_format.class,
tc08_login_wrong_password.class,
tc09_forgetpwd_valid_email.class,
tc10_forgetpwd_invalid_email.class,

tc11_login_valid_emailpwd.class,
tc12_search_withText.class,
tc13_search_withPicture.class,
tc14_search_valid_postalCode.class,
tc15_search_invalid_postalCode.class,
tc16_addToShoppingList.class,
tc17_deletefrmShoppingList.class,
tc18_changeQuantity.class,
tc19_addToShoppingCart_from_ShoppingList.class,
tc20_printShoppingList.class,

tc21_createshoppinglist.class,
tc22_deleteshoppinglist.class,
tc23_addProductToCart.class,
tc24_changeProductQuantity.class,
tc25_deleteProduct.class,
tc26_removeAllProduct.class,
tc27_editProfile.class,
tc28_failedToEditProfile.class,
tc29_failedToChangeEmail.class,
tc30_failedToChangePassword.class,

})

```
public class SuiteClass {
```

```
}
```

```
package softwaretestingpackage;
```

```
import static org.junit.Assert.*;
```

```
import java.time.Duration;
```

```
import org.junit.Test;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.JavascriptExecutor;
```

```
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import org.openqa.selenium.support.ui.ExpectedConditions;
```

```
import org.openqa.selenium.support.ui.WebDriverWait;
```

```
public class tc01_register_valid_email {
```

```
    @Test
```

```
    public void register_valid_email() {
```

```
        // 1. Register with valid email
```

```
        System.setProperty("webdriver.chrome.driver",  
"D:\\Common\\drivers\\chromedriver.exe");
```

```
        ChromeDriver cd = new ChromeDriver();
```

```

cd.get("https://www.ikea.com/my/en/profile/login/");

cd.manage().window().maximize();

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));

WebDriverWait wait = new WebDriverWait(cd, Duration.ofSeconds(15));

JavascriptExecutor js = (JavascriptExecutor) cd;

js.executeScript("window.scrollTo(0,1000)");

wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//span[(text()='Create account')]]")));

    wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//span[(text()='Create account')]]")));

        cd.findElement(By.xpath("//span[(text()='Create account')]]")).click(); //click create account

            cd.findElement(By.id("regular-signup-form-firstName")).sendKeys("John"); //insert first name

            cd.findElement(By.id("regular-signup-form-lastName")).sendKeys("Doe"); //insert last name

            cd.findElement(By.id("regular-signup-form-username")).sendKeys("group6softwaretesting@gmail.com"); //insert valid email

            cd.findElement(By.id("regular-signup-form-password")).sendKeys("Welcome01$"); //insert password

        js.executeScript("window.scrollTo(0,1000)");

        WebElement acceptPrivacyPolicy1 = cd.findElement(By.id("regular-signup-form-acceptPrivacyPolicy"));

        for (int i = 0; i < 2; i++) {

            acceptPrivacyPolicy1.click(); //tick privacy policy

        }

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        cd.findElement(By.xpath("//span[(text()='Create profile')]]")).click(); //click create profile

        cd.quit();

    }

```

```
}
```

```
package softwaretestingpackage;
```

```
import static org.junit.Assert.*;
```

```
import java.time.Duration;
```

```
import org.junit.Test;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.JavascriptExecutor;
```

```
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import org.openqa.selenium.support.ui.ExpectedConditions;
```

```
import org.openqa.selenium.support.ui.WebDriverWait;
```

```
public class tc02_register_invalid_email {
```

```
    @Test
```

```
    public void register_invalid_email() {
```

```
        // 2. Register with invalid email
```

```
        System.setProperty("webdriver.chrome.driver",  
"D:\\Common\\drivers\\chromedriver.exe");
```

```
        ChromeDriver cd = new ChromeDriver();
```

```
        cd.get("https://www.ikea.com/my/en/profile/login/");
```

```
        cd.manage().window().maximize();
```



```

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));

WebDriverWait wait = new WebDriverWait(cd, Duration.ofSeconds(15));

JavascriptExecutor js = (JavascriptExecutor) cd;

js.executeScript("window.scrollBy(0,1000)");

wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//span[(text()='Create account')]"")));

wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//span[(text()='Create account')]"")));

cd.findElement(By.xpath("//span[(text()='Create account')]"")).click(); //click create account

cd.findElement(By.id("regular-signup-form-firstName")).sendKeys("John"); //insert first name

cd.findElement(By.id("regular-signup-form-lastName")).sendKeys("Doe"); //insert last name

cd.findElement(By.id("regular-signup-form-username")).sendKeys("blablabla998@dummy.com"); //insert invalid email

cd.findElement(By.id("regular-signup-form-password")).sendKeys("Welcome01$");
//insert password

WebElement acceptPrivacyPolicy1 = cd.findElement(By.id("regular-signup-form-acceptPrivacyPolicy"));

js.executeScript("window.scrollBy(0,1000)");

for (int i = 0; i < 2; i++) {

    acceptPrivacyPolicy1.click(); //tick privacy policy

}

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

cd.findElement(By.xpath("//span[(text()='Create profile')]"")).click(); //click create profile

cd.quit();

}

}

```

```

package softwaretestingpackage;

import static org.junit.Assert.*;

import java.time.Duration;

import org.junit.Assert;
import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class tc03_without_firstName {

    @Test
    public void without_firstName() {

        // 3. Register without insert First Name

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        cd.get("https://www.ikea.com/my/en/profile/login/");

        cd.manage().window().maximize();

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));

```

```

WebDriverWait wait = new WebDriverWait(cd, Duration.ofSeconds(15));

JavascriptExecutor js = (JavascriptExecutor) cd;

js.executeScript("window.scrollTo(0,1000)");

wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//span[(text()='Create
account')]"))));

wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//span[(text()='Create
account')]"))));

cd.findElement(By.xpath("//span[(text()='Create account')]")).click(); //click create
account

cd.findElement(By.id("regular-signup-form-firstName")).sendKeys(""); //insert first
name

cd.findElement(By.id("regular-signup-form-lastName")).sendKeys("Doe"); //insert last
name

cd.findElement(By.id("regular-signup-form-
username")).sendKeys("group6softwaretesting@gmail.com"); //insert valid email

cd.findElement(By.id("regular-signup-form-password")).sendKeys("Welcome01$");
//insert password

// Verify error msg with assert

WebElement verifyErrorMsg = cd.findElement(By.id("regular-signup-form-
firstName-error"));

String str = verifyErrorMsg.getText();

Assert.assertTrue(str.contains("Your first name is required."));

cd.quit();

}

}

package softwaretestingpackage;

import static org.junit.Assert.*;

```

```

import java.time.Duration;

import org.junit.Assert;
import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class tc04_without_lastName {

    @Test
    public void without_lastName() {

        // 4. Register without insert Last Name

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        cd.get("https://www.ikea.com/my/en/profile/login/");

        cd.manage().window().maximize();

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));

        WebDriverWait wait = new WebDriverWait(cd, Duration.ofSeconds(15));

        JavascriptExecutor js = (JavascriptExecutor) cd;

        js.executeScript("window.scrollTo(0,1000)");

```

```

wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//span[(text()='Create
account')]"))));

    wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//span[(text()='Create
account')]"))));

    cd.findElement(By.xpath("//span[(text()='Create account')]")).click(); //click create
account

    cd.findElement(By.id("regular-signup-form-firstName")).sendKeys("John"); //insert first
name

    cd.findElement(By.id("regular-signup-form-lastName")).sendKeys(""); //insert last
name

    cd.findElement(By.id("regular-signup-form-
username")).sendKeys("group6softwaretesting@gmail.com"); //insert valid email

    cd.findElement(By.id("regular-signup-form-password")).sendKeys("Welcome01$");
//insert password

    // Verify error msg with assert

    WebElement verifyErrorMsg = cd.findElement(By.id("regular-signup-form-lastName-
error"));

    String str = verifyErrorMsg.getText();

    Assert.assertTrue(str.contains("Your last name is required.));

cd.quit();

}

}

package softwaretestingpackage;

import static org.junit.Assert.*;

import java.time.Duration;

```

```

import org.junit.Assert;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

public class tc05_register_firstName_usingSpecialChar {

    @Test

    public void register_firstName_usingSpecialChar() {

        // 5. Register (insert the First Name using numbers or special characters)

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        cd.get("https://www.ikea.com/my/en/profile/login/");

        cd.manage().window().maximize();

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));

        WebDriverWait wait = new WebDriverWait(cd, Duration.ofSeconds(15));

        JavascriptExecutor js = (JavascriptExecutor) cd;

        js.executeScript("window.scrollTo(0,1000)");

        wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//span[(text()='Create
account')]"))));

        wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//span[(text()='Create
account')]"))));

```

```

        cd.findElement(By.xpath("//span[(text()='Create account')]")).click(); //click create
account

        cd.findElement(By.id("regular-signup-form-firstName")).sendKeys("12345$"); //insert
first name

        cd.findElement(By.id("regular-signup-form-lastName")).sendKeys("Doe"); //insert last
name

        cd.findElement(By.id("regular-signup-form-
username")).sendKeys("group6softwaretesting@gmail.com"); //insert valid email

        cd.findElement(By.id("regular-signup-form-password")).sendKeys("Welcome01$");
//insert password

        // Verify error msg with assert

        WebElement verifyErrorMsg = cd.findElement(By.id("regular-signup-form-
firstName-error"));

        String str = verifyErrorMsg.getText();

        Assert.assertTrue(str.contains("Your first name must not contain <>{}[]()*=*; or be
only spaces, numbers or special characters."));

        cd.quit();

    }

}

package softwaretestingpackage;

import static org.junit.Assert.*;

import java.time.Duration;

import org.junit.Assert;

import org.junit.Test;

import org.openqa.selenium.By;

```

```

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;


public class tc06_register_lastName_usingSpecial_char {

    @Test

    public void test() {

        // 6. Register (insert the Last Name using numbers or special characters)

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        cd.get("https://www.ikea.com/my/en/profile/login/");

        cd.manage().window().maximize();

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));

        WebDriverWait wait = new WebDriverWait(cd, Duration.ofSeconds(15));

        JavascriptExecutor js = (JavascriptExecutor) cd;

        js.executeScript("window.scrollTo(0,1000)");


        wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//span[(text()='Create
account')]"))));

        wait.until(ExpectedConditions.elementToBeClickable(By.xpath("//span[(text()='Create
account')]"))));

        cd.findElement(By.xpath("//span[(text()='Create account')]")).click(); //click create
account

        cd.findElement(By.id("regular-signup-form-firstName")).sendKeys("John"); //insert first
name

```



```
        cd.findElement(By.id("regular-signup-form-lastName")).sendKeys("12345$"); //insert  
last name
```

```
        cd.findElement(By.id("regular-signup-form-  
username")).sendKeys("group6softwaretesting@gmail.com"); //insert valid email
```

```
        cd.findElement(By.id("regular-signup-form-password")).sendKeys("Welcome01$");  
//insert password
```

```
    // Verify error msg with assert
```

```
        WebElement verifyErrorMsg = cd.findElement(By.id("regular-signup-form-  
lastName-error"));
```

```
        String str = verifyErrorMsg.getText();
```

```
        Assert.assertTrue(str.contains("Your last name must not contain <>{}[]()*=*; or be  
only spaces, numbers or special characters."));
```

```
        cd.quit();
```

```
    }
```

```
}
```

```
package softwaretestingpackage;
```

```
import static org.junit.Assert.*;
```

```
import java.time.Duration;
```

```
import org.junit.Test;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.JavascriptExecutor;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import org.openqa.selenium.support.ui.WebDriverWait;
```

```

public class tc07_login_wrong_email_format {

    @Test

    public void login_wrong_email_format() {

        // 7. Login with invalid email

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        cd.get("https://www.ikea.com/my/en/profile/login/");

        cd.manage().window().maximize();

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));

        cd.findElement(By.id("login-form-username")).sendKeys("group6softwaretesting");
//insert invalid email

        cd.findElement(By.id("login-form-password")).sendKeys("Welcome01$"); //insert
password

        JavascriptExecutor js = (JavascriptExecutor) cd;

        js.executeScript("window.scrollTo(0,1000)");

        cd.findElement((By.xpath("//span[(text()='Login')]"))).click(); //click login

        cd.quit();

    }

}

package softwaretestingpackage;

import static org.junit.Assert.*;

import java.time.Duration;

```

```

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.WebDriverWait;

public class tc08_login_wrong_password {

    @Test

    public void login_wrong_password() {

        // 8. Login with invalid password

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        cd.get("https://www.ikea.com/my/en/profile/login/");

        cd.manage().window().maximize();

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));

        WebDriverWait wait = new WebDriverWait(cd, Duration.ofSeconds(15));

        cd.findElement(By.id("login-form-username")).sendKeys("group6softwaretesting@gmail.com"); //insert invalid email

        cd.findElement(By.id("login-form-password")).sendKeys("wrongpassword"); //insert password

        JavascriptExecutor js = (JavascriptExecutor) cd;

        js.executeScript("window.scrollTo(0,1000)");

        cd.findElement((By.xpath("//span[(text()='Login')]"))).click(); //click login

        cd.quit();

    }
}

```

```
}
```

```
package softwaretestingpackage;
```

```
import static org.junit.Assert.*;
```

```
import java.time.Duration;
```

```
import org.junit.Test;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.JavascriptExecutor;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import org.openqa.selenium.support.ui.WebDriverWait;
```

```
public class tc09_forgetpwd_valid_email {
```

```
    @Test
```

```
    public void forgetpwd_valid_email() {
```

```
        // 9. Forget password (insert valid email)
```

```
        System.setProperty("webdriver.chrome.driver",  
"D:\\Common\\drivers\\chromedriver.exe");
```

```
        ChromeDriver cd = new ChromeDriver();
```

```
        cd.get("https://www.ikea.com/my/en/profile/login/");
```

```
        cd.manage().window().maximize();
```

```
        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));
```

```
        WebDriverWait wait = new WebDriverWait(cd, Duration.ofSeconds(15));
```

```

        JavascriptExecutor js = (JavascriptExecutor) cd;

        js.executeScript("window.scrollTo(0,1000)");

        cd.findElement(By.id("reset-password-link")).click();

        cd.findElement(By.id("reset-password-form-username")).sendKeys("group6softwaretesting@gmail.com");

        cd.findElement((By.xpath("//span[(text()='Send new password')]"))).click();

        cd.quit();

    }

}

```

```

package softwaretestingpackage;

```

```

import static org.junit.Assert.*;

```

```

import java.time.Duration;

```

```

import org.junit.Test;

```

```

import org.openqa.selenium.By;

```

```

import org.openqa.selenium.JavascriptExecutor;

```

```

import org.openqa.selenium.chrome.ChromeDriver;

```

```

import org.openqa.selenium.support.ui.WebDriverWait;

```

```

public class tc10_forgetpwd_invalid_email {

```

```

    @Test

```

```

    public void forgetpwd_invalid_email() {

```

```

        // 10. Forget password (insert invalid email)

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        cd.get("https://www.ikea.com/my/en/profile/login/");

        cd.manage().window().maximize();

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));

        WebDriverWait wait = new WebDriverWait(cd, Duration.ofSeconds(15));

        JavascriptExecutor js = (JavascriptExecutor) cd;

        js.executeScript("window.scrollTo(0,1000)");

        cd.findElement(By.id("reset-password-link")).click();

        cd.findElement(By.id("reset-password-form-username")).sendKeys("blablabla998@dummy.com");

        cd.findElement((By.xpath("//span[(text()='Send new password')]"))).click();

        cd.quit();

        }

}

```

```

package softwaretestingpackage;

```

```

import static org.junit.Assert.*;

```

```

import java.time.Duration;

```

```

import org.junit.Test;

```

```

import org.openqa.selenium.By;

```

```

import org.openqa.selenium.JavascriptExecutor;

```

```

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;


public class tc11_login_valid_emailpwd {

    @Test

    public void login_valid_emailpwd() throws InterruptedException {

        // 11. Login with valid email and valid password

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        cd.manage().window().maximize();

        // Direct to login page

        cd.get("https://www.ikea.com/my/en/profile/login/");

        // Enter valid credentials

        cd.findElement(By.id("login-form-
username")).sendKeys("abc763188@gmail.com");

        cd.findElement(By.id("login-form-password")).sendKeys("Abc123456");

        // Remove pop-up menus

        cd.findElement(By.xpath("/html/body/div[13]/div/div[2]/button")).click();

        cd.findElement(By.xpath("/html/body/div[12]/span")).click();

        Thread.sleep(2000);

        // Click on submit button

        WebElement loginbutton = cd.findElement(By.xpath("//*[@id='login-form-
submit']"));

        JavascriptExecutor executor = (JavascriptExecutor) cd;

        executor.executeScript("arguments[0].click();", loginbutton);

        Thread.sleep(3000);
    }
}

```

```

        cd.quit();
    }

}

package softwaretestingpackage;

import static org.junit.Assert.*;

import java.time.Duration;

import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.chrome.ChromeDriver;

public class tc12_search_withText {

    @Test
    public void search_withText() throws InterruptedException {

        // 12. Search product with text

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        cd.manage().window().maximize();

        // Direct to Dashboard page

        cd.navigate().to("https://www.ikea.com/my/en/profile/dashboard/");

        // Remove pop-up menus

```



```

        cd.findElement(By.xpath("/html/body/div[13]/div/div[2]/button")).click();

        cd.findElement(By.xpath("/html/body/div[12]/span")).click();

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        // Enter product name

        cd.findElement(By.xpath("/html/body/header/div/div/div/div[2]/div/div/form/div/div/
div/form/div[1]/div/input"))

                .sendKeys("Display Cabinet");

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        // Click on search button

        cd.findElement(By.xpath("//*[@id=\"search-box__searchbutton\"]")).click();

        Thread.sleep(3000);

        cd.quit();

    }

}

package softwaretestingpackage;

import static org.junit.Assert.*;

import java.time.Duration;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.chrome.ChromeDriver;

public class tc13_search_withPicture {

```

[illegible]

```

package softwaretestingpackage;

import static org.junit.Assert.*;

import java.time.Duration;

import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class tc14_search_valid_postalCode {

    @Test
    public void search_valid_postalCode() throws InterruptedException {

        // 14. Search store with valid postal code (ex: 14000)

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        cd.manage().window().maximize();

        JavascriptExecutor executor = (JavascriptExecutor) cd;

        // Direct to Dashboard page

        cd.get("https://www.ikea.com/my/en/profile/dashboard/");

        // Remove pop-up menus

        cd.findElement(By.xpath("/html/body/div[13]/div/div[2]/button")).click();

```

```

cd.findElement(By.xpath("/html/body/div[12]/span")).click();

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

Thread.sleep(5000);

// Click on "Select Store" text

WebElement select_store =
cd.findElement(By.xpath("/html/body/div[5]/div/div/div/button/div/strong"));

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

executor.executeScript("arguments[0].click();", select_store);

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

// Enter valid postcode

cd.findElement(By.xpath("//*[@id=\"hnf-txt-storepicker-
postcode\"]")).sendKeys("14000");

Thread.sleep(3000);

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

// Click on Search button

WebElement searchbuttonpostal = cd.findElement(By.xpath("//*[@id=\"hnf-
btn-storepicker-method-postcode\"]"));

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

executor.executeScript("arguments[0].click();", searchbuttonpostal);

Thread.sleep(5000);

// Click on X icon to close the pop-Up menu

WebElement closebuttonpostal =
cd.findElement(By.xpath("/html/body/aside[2]/div[2]/div[3]/button"));

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

executor.executeScript("arguments[0].click();", closebuttonpostal);

Thread.sleep(3000);

cd.quit();

}

```

```
}
```

```
package softwaretestingpackage;
```

```
import static org.junit.Assert.*;
```

```
import java.time.Duration;
```

```
import org.junit.Assert;
```

```
import org.junit.Test;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.JavascriptExecutor;
```

```
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class tc15_search_invalid_postalCode {
```

```
    @Test
```

```
    public void search_invalid_postalCode() throws InterruptedException {
```

```
        // 15. Search store with invalid postal code (ex: 14999)
```

```
        System.setProperty("webdriver.chrome.driver",  
"D:\\Common\\drivers\\chromedriver.exe");
```

```
        ChromeDriver cd = new ChromeDriver();
```

```
        cd.manage().window().maximize();
```

```
        JavascriptExecutor executor = (JavascriptExecutor) cd;
```

```
        // Direct to Dashboard page
```

```
        cd.get("https://www.ikea.com/my/en/profile/dashboard/");
```

```

// Remove pop-Up menus

cd.findElement(By.xpath("/html/body/div[13]/div/div[2]/button")).click();

cd.findElement(By.xpath("/html/body/div[12]/span")).click();

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

Thread.sleep(5000);

// Click on "Select Store" text

WebElement select_store =
cd.findElement(By.xpath("/html/body/div[5]/div/div/div/button/div/strong"));

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

executor.executeScript("arguments[0].click();", select_store);

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

// Enter invalid postcode

cd.findElement(By.xpath("//*[@id=\"hnf-txt-storepicker-
postcode\"]")).sendKeys("14999");

Thread.sleep(3000);

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

// Click on Search button

WebElement searchbuttonpostal = cd.findElement(By.xpath("//*[@id=\"hnf-
btn-storepicker-method-postcode\"]"));

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

executor.executeScript("arguments[0].click();", searchbuttonpostal);

Thread.sleep(5000);

// Verify error msg with assert

WebElement verifyErrorMsg = cd

    .findElement(By.xpath("//*[@id=\"hnf-
storepicker\"]div[2]/div[2]/div[3]/div[1]/div"));

String str = verifyErrorMsg.getText();

Assert.assertTrue(str.contains("We're sorry, but it appears we were unable to
find the store closest to you."));

```

```
        System.out.println("TestCase 15: Search store with invalid postal code  
completed");
```

```
        // Click on X icon to close the pop-up menu
```

```
        WebElement closebuttonpostal =  
cd.findElement(By.xpath("/html/body/aside[2]/div[2]/div[3]/button"));
```

```
        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

```
        executor.executeScript("arguments[0].click();", closebuttonpostal);
```

```
        Thread.sleep(3000);
```

```
        cd.quit();
```

```
    }
```

```
}
```

```
package softwaretestingpackage;
```

```
import static org.junit.Assert.*;
```

```
import java.time.Duration;
```

```
import java.util.concurrent.TimeUnit;
```

```
import org.junit.Assert;
```

```
import org.junit.Test;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.JavascriptExecutor;
```

```
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class tc16_addToShoppingList {
```

@Test

```
public void addToShoppingList() throws InterruptedException {

    System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

    ChromeDriver cd = new ChromeDriver();

    cd.manage().window().maximize();

    // 11. Login with valid email and valid password

    cd.get("https://www.ikea.com/my/en/profile/login/");

    cd.findElement(By.id("login-form-
username")).sendKeys("abc763188@gmail.com");

    cd.findElement(By.id("login-form-password")).sendKeys("Abc123456");

    // Remove pop-up menus

    cd.findElement(By.xpath("/html/body/div[13]/div/div[2]/button")).click();

    cd.findElement(By.xpath("/html/body/div[12]/span")).click();

    Thread.sleep(2000);

    WebElement loginbutton = cd.findElement(By.xpath("//*[@id='login-form-
submit']"));

    JavascriptExecutor executor = (JavascriptExecutor) cd;

    executor.executeScript("arguments[0].click();", loginbutton);

    cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

    // 16. Add to shopping list

    // Navigate to Dashboard page

    cd.navigate().to("https://www.ikea.com/my/en/profile/dashboard/");

    cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

    // Enter word "RUMPLING" in search textfield
```



```
cd.findElement(By.xpath("/html/body/header/div/div/div/div[2]/div/div/form/div/div/div/form/div[1]/div/input"))
```

```
.sendKeys("RUMPLING");
```

```
cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

```
// Click on search button
```

```
cd.findElement(By.xpath("//*[@id=\"search-box__searchbutton\"]")).click();
```

```
// Add Product 1
```

```
cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

```
// Navigate to RUMPLING page
```

```
cd.navigate().to("https://www.ikea.com/my/en/search/products/?q=RUMPLING");
```

```
cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

```
// Click on the product displayed
```

```
WebElement productimage = cd.findElement(By.xpath("//*[@id=\"search-results\"]/div[1]/a/div/span[2]"));
```

```
cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

```
executor.executeScript("arguments[0].click();", productimage);
```

```
Thread.sleep(5000);
```

```
cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

```
// Click on the heart-shape icon
```

```
WebElement shoppinglisticon =  
cd.findElement(By.xpath("//*[@id=\"content\"]/div/div/div/div[2]/div[1]/div/div[2]/div/button"));
```

```
cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

```
executor.executeScript("arguments[0].click();", shoppinglisticon);
```

```
Thread.sleep(5000);
```

```
cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

```

        // Verify product name with assert

        WebElement verifyproductName =
cd.findElement(By.xpath("//*[@id=\"content\"]/div/div/div/div[2]/div[1]/div/div[1]/div[2]/div[1]/h1/div[1]"));

        String str = verifyproductName.getText();

        Assert.assertTrue(str.contains("RUMPLING"));

        System.out.println("TestCase 16: Add to Shopping List completed");


        // Add Product 2

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        cd.navigate().to("https://www.ikea.com/my/en/search/products/?q=Display%20Cabinet");

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        WebElement productimage2 = cd.findElement(By.xpath("//*[@id=\"search-results\"]/div[2]/a/div/span[2]"));

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        Thread.sleep(3000);

        executor.executeScript("arguments[0].click();", productimage2);

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        WebElement shoppinglisticon2 =
cd.findElement(By.xpath("//*[@id=\"content\"]/div/div/div/div[2]/div[1]/div/div[3]/div/button"));

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        Thread.sleep(3000);

        executor.executeScript("arguments[0].click();", shoppinglisticon2);

        Thread.sleep(5000);

        cd.quit();

    }

```

```
}
```

```
package softwaretestingpackage;
```

```
import static org.junit.Assert.*;
```

```
import java.time.Duration;
```

```
import org.junit.Test;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.JavascriptExecutor;
```

```
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class tc17_deletefrmShoppingList {
```

```
    @Test
```

```
    public void deletefrmShoppingList() throws InterruptedException {
```

```
        // 11. Login with valid email and valid password
```

```
        System.setProperty("webdriver.chrome.driver",  
"D:\\Common\\drivers\\chromedriver.exe");
```

```
        ChromeDriver cd = new ChromeDriver();
```

```
        cd.manage().window().maximize();
```

```
        // Direct to login page
```

```
        cd.get("https://www.ikea.com/my/en/profile/login/");
```

```
        // enter valid credentials
```

```
        cd.findElement(By.id("login-form-  
username")).sendKeys("abc763188@gmail.com");
```

```

cd.findElement(By.id("login-form-password")).sendKeys("Abc123456");

// Remove pop-up menus

cd.findElement(By.xpath("/html/body/div[13]/div/div[2]/button")).click();

cd.findElement(By.xpath("/html/body/div[12]/span")).click();

Thread.sleep(2000);

// Click on submit button

WebElement loginbutton = cd.findElement(By.xpath("//*[@id=\"login-form-submit\"]"));

JavascriptExecutor executor = (JavascriptExecutor) cd;

executor.executeScript("arguments[0].click();", loginbutton);

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));


// 17. Delete from shopping list

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

// Direct to Shopping List page

cd.navigate().to("https://order.ikea.com/my/en/checkout/shoppinglist/");

// Click on delete icon

Thread.sleep(5000);

WebElement shoppinglistDeleteicon = cd.findElement(
        By.xpath("//*[@id=\"one-checkout\"]/div[2]/div/div[4]/div[1]/div/div[4]/div[2]/div[1]/button"));

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

executor.executeScript("arguments[0].click();", shoppinglistDeleteicon);

Thread.sleep(5000);

// Click on the Confirm to delete button

WebElement shoppinglistDeleteconfirm = cd
        .findElement(By.xpath("//*[@id=\"one-checkout\"]/div[2]/div/div[4]/div[1]/div/div[4]/div/div[3]/button"));

```

```

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        executor.executeScript("arguments[0].click();", shoppinglistDeleteconfirm);

        Thread.sleep(3000);

        cd.quit();
    }

}

```

```

package softwaretestingpackage;

```

```

import static org.junit.Assert.*;

```

```

import java.time.Duration;

```

```

import org.junit.Assert;

```

```

import org.junit.Test;

```

```

import org.openqa.selenium.By;

```

```

import org.openqa.selenium.JavascriptExecutor;

```

```

import org.openqa.selenium.WebElement;

```

```

import org.openqa.selenium.chrome.ChromeDriver;

```

```

import org.openqa.selenium.support.ui.Select;

```

```

public class tc18_changeQuantity {

```

```

    @Test

```

```

    public void changeQuantity() throws InterruptedException {

```

```

        // 11. Login with valid email and valid password

```

```

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        cd.manage().window().maximize();

        // Direct to login page

        cd.get("https://www.ikea.com/my/en/profile/login/");

        // enter valid credentials

        cd.findElement(By.id("login-form-
username")).sendKeys("abc763188@gmail.com");

        cd.findElement(By.id("login-form-password")).sendKeys("Abc123456");

        // Remove pop-up menus

        cd.findElement(By.xpath("/html/body/div[13]/div/div[2]/button")).click();

        cd.findElement(By.xpath("/html/body/div[12]/span")).click();

        Thread.sleep(2000);

        // Click on submit button

        WebElement loginbutton = cd.findElement(By.xpath("//*[@id='login-form-
submit']"));

        JavascriptExecutor executor = (JavascriptExecutor) cd;

        executor.executeScript("arguments[0].click();", loginbutton);

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));


        // 18. Change quantity

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        // Direct to Shopping List page

        cd.navigate().to("https://order.ikea.com/my/en/checkout/shoppinglist/");

        Thread.sleep(5000);

        // Click on quantity dropdown menu

        Select dropdown = new Select(cd.findElement(By.id("js_qty_50511683")));

```

```

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        // Click on the number 3

        dropdown.selectByVisibleText("3");


        // Verify product quantity with assert

        WebElement verifyproductQuantity =
cd.findElement(By.xpath("//*[@id=\"js_qty_50511683\"]/option[3]"));

        String str = verifyproductQuantity.getText();

        Assert.assertTrue(str.contains("3"));

        System.out.println("TestCase 18: Change Quantity completed");

        Thread.sleep(3000);

        cd.quit();

    }
}

```

```

package softwaretestingpackage;

```

```

import static org.junit.Assert.*;

```

```

import java.time.Duration;

```

```

import org.junit.Test;

```

```

import org.openqa.selenium.By;

```

```

import org.openqa.selenium.JavascriptExecutor;

```

```

import org.openqa.selenium.WebElement;

```

```

import org.openqa.selenium.chrome.ChromeDriver;

```

```

public class tc19_addToShoppingCart_from_ShoppingList {

    @Test

    public void addToShoppingCart_from_ShoppingList() throws InterruptedException {

        // 11. Login with valid email and valid password

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        cd.manage().window().maximize();

        // Direct to login page

        cd.get("https://www.ikea.com/my/en/profile/login/");

        // Enter valid credentials

        cd.findElement(By.id("login-form-
username")).sendKeys("abc763188@gmail.com");

        cd.findElement(By.id("login-form-password")).sendKeys("Abc123456");

        // Remove pop-up menus

        cd.findElement(By.xpath("/html/body/div[13]/div/div[2]/button")).click();

        cd.findElement(By.xpath("/html/body/div[12]/span")).click();

        Thread.sleep(2000);

        // Click on submit button

        WebElement loginbutton = cd.findElement(By.xpath("//*[@id='login-form-
submit']"));

        JavascriptExecutor executor = (JavascriptExecutor) cd;

        executor.executeScript("arguments[0].click();", loginbutton);

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        // 19. Add to shopping cart from shopping list

```



```

        // Direct to Shopping List page

        cd.navigate().to("https://order.ikea.com/my/en/checkout/shoppinglist/");

        Thread.sleep(5000);

        // Click on the "Add to shopping cart" button

        WebElement shoppinglisttoCart = cd.findElement(
                By.xpath("//*[@id=\"one-
checkout\"]/div[2]/div/div[4]/div[1]/div/div[4]/div[2]/div[3]/button"));

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        executor.executeScript("arguments[0].click();", shoppinglisttoCart);

        Thread.sleep(5000);

        // Navigate to Shopping Cart page

        cd.navigate().to("https://secure.ikea.com/my/en/mcommerce/shoppingcart");

        Thread.sleep(3000);

        cd.quit();

    }

}

```

```

package softwaretestingpackage;

```

```

import static org.junit.Assert.*;

```

```

import java.time.Duration;

```

```

import org.junit.Test;

```

```

import org.openqa.selenium.By;

```

```

import org.openqa.selenium.JavascriptExecutor;

```

```

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;


public class tc20_printShoppingList {

    @Test

    public void printShoppingList() throws InterruptedException {

        // 11. Login with valid email and valid password

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        cd.manage().window().maximize();

        // Direct to login page

        cd.get("https://www.ikea.com/my/en/profile/login/");

        // enter valid credentials

        cd.findElement(By.id("login-form-
username")).sendKeys("abc763188@gmail.com");

        cd.findElement(By.id("login-form-password")).sendKeys("Abc123456");

        // Remove pop-Up menus

        cd.findElement(By.xpath("/html/body/div[13]/div/div[2]/button")).click();

        cd.findElement(By.xpath("/html/body/div[12]/span")).click();

        Thread.sleep(2000);

        // Click on submit button

        WebElement loginbutton = cd.findElement(By.xpath("//*[@id='login-form-
submit']"));

        JavascriptExecutor executor = (JavascriptExecutor) cd;

        executor.executeScript("arguments[0].click();", loginbutton);

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

```

```

        // 20. Print the shopping list (pdf file)

        // Direct to Shopping List page

        cd.navigate().to("https://order.ikea.com/my/en/checkout/shoppinglist/");

        Thread.sleep(5000);

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        // Click on the printer icon

        cd.findElement(By.xpath("//*[@id=\"one-
checkout\"]/div[2]/div/div[1]/h1/button[2]")).click();

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));

        Thread.sleep(5000);

        cd.quit();

    }
}

```

```

package softwaretestingpackage;

```

```

import static org.junit.Assert.*;

```

```

import java.time.Duration;

```

```

import org.junit.Assert;

```

```

import org.junit.Test;

```

```

import org.openqa.selenium.By;

```

```

import org.openqa.selenium.WebElement;

```

```

import org.openqa.selenium.chrome.ChromeDriver;

```

```

public class tc21_createshoppinglist {

    @Test

    public void createshoppinglist() {

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        String URL = "https://www.ikea.com/my/en/";

        cd.get(URL);

        cd.manage().window().maximize();


        cd.findElement(By.xpath("/html/body/header/div/div/div/ul/li[5]/a")).click(); //click
shopping list

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(15));

        cd.findElement(By.xpath("/html/body/main/div[2]/div/div[1]/h1/button[1]")).click();
//click drop down

        cd.findElement(By.xpath("/html/body/div[13]/div/div/div/div/button")).click(); //click
create new list

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));


        cd.findElement(By.xpath("/html/body/div[13]/div/div/div/form/div[1]/input")).sendKeys("To
y List"); //input list name


        cd.findElement(By.xpath("/html/body/div[13]/div/div/div/form/div[2]/div[2]/button")).click()
; //click create list


        // Verify shopping list name with assert

        WebElement shoppinglistName =
cd.findElement(By.xpath("/html/body/div[13]/div/div/ul/li[2]/div"));

```

```

        String str = shoppinglistName.getText();

        Assert.assertTrue(str.contains("Toy List"));

        System.out.println("TestCase 21: Create a new shopping list completed");
        cd.quit();
    }

}

package softwaretestingpackage;

import static org.junit.Assert.*;

import java.time.Duration;

import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.chrome.ChromeDriver;

public class tc22_deleteshoppinglist {

    @Test
    public void deleteshoppinglist() {

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        String URL = "https://www.ikea.com/my/en/";

```

```

cd.get(URL);

cd.manage().window().maximize();


cd.findElement(By.xpath("/html/body/header/div/div/div/ul/li[5]/a")).click(); //click
shopping list

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(15));

cd.findElement(By.xpath("/html/body/main/div[2]/div/div[1]/h1/button[1]")).click();
//click drop down

cd.findElement(By.xpath("/html/body/div[13]/div/div/div/div/button")).click(); //click
create new list

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));


cd.findElement(By.xpath("/html/body/div[13]/div/div/div/form/div[1]/input")).sendKeys("To
y List"); //input list name


cd.findElement(By.xpath("/html/body/div[13]/div/div/div/form/div[2]/div[2]/button")).click()
; //click create list

cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));


cd.findElement(By.xpath("/html/body/div[13]/div/div/ul/li[2]/div/button[3]")).click();
//click delete icon


cd.findElement(By.xpath("/html/body/div[13]/div/div/ul/li[2]/div[2]/div[3]/button")).click();
//click delete

cd.quit();

}

}

package softwaretestingpackage;

```

```
import static org.junit.Assert.*;
```

```
import java.time.Duration;
```

```
import org.junit.Test;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.Keys;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class tc23_addProductToCart {
```

```
    @Test
```

```
    public void addProductToCart() {
```

```
        System.setProperty("webdriver.chrome.driver",  
"D:\\Common\\drivers\\chromedriver.exe");
```

```
        ChromeDriver cd = new ChromeDriver();
```

```
        String URL = "https://www.ikea.com/my/en/";
```

```
        cd.get(URL);
```

```
        cd.manage().window().maximize();
```

```
        cd.findElement(By.xpath("/html/body/header/div/div/div/div[2]/div/div/form/div/div/div/form/div[1]/div/input")).sendKeys("BACKTRAV");
```

```
        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(15));
```

```
        cd.findElement(By.xpath("/html/body/header/div/div/div/div[2]/div/div/form/div/div/div/form/div[2]/div/div/ol/li[1]/a")).click(); //click backtrav
```

```
cd.findElement(By.xpath("/html/body/main/div/div/div/div[2]/div[1]/div/div[2]/div/div/button")).sendKeys(Keys.ENTER); //click add to cart
```

```
cd.quit();
```

```
}
```

```
}
```

```
package softwaretestingpackage;
```

```
import static org.junit.Assert.*;
```

```
import java.time.Duration;
```

```
import org.junit.Assert;
```

```
import org.junit.Test;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
import org.openqa.selenium.support.ui.Select;
```

```
public class tc24_changeProductQuantity {
```

```
    @Test
```

```
    public void changeProductQuantity() throws InterruptedException {
```

```
        System.setProperty("webdriver.chrome.driver",  
"D:\\Common\\drivers\\chromedriver.exe");
```

```
        ChromeDriver cd = new ChromeDriver();
```



```

String URL = "https://www.ikea.com/my/en/p/backtrav-bolster-10421000/";

cd.get(URL);

cd.manage().window().maximize();


cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));

cd.findElement(By.xpath("/html/body/main/div/div/div/div[2]/div[1]/div/div[2]/div/div/button")).click(); //click add to cart

Thread.sleep(8000);


cd.findElement(By.xpath("/html/body/header/div/div/div/ul/li[6]/a")).click(); //click shopping cart

Select select = new Select(cd.findElement(By.id("quantity_1"))); //click quantity
select.selectByValue("2");


        cd.quit();
    }

}

package softwaretestingpackage;

import static org.junit.Assert.*;

import java.time.Duration;

import org.junit.Test;

import org.openqa.selenium.By;

```

```

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;


public class tc25_deleteProduct {

    @Test

    public void deleteProduct() throws InterruptedException {

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        String URL = "https://www.ikea.com/my/en/p/backtrav-bolster-10421000/";

        cd.get(URL);

        cd.manage().window().maximize();


        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));

cd.findElement(By.xpath("/html/body/main/div/div/div/div[2]/div[1]/div/div[2]/div/div/butto
n")).click(); //click add to cart

        Thread.sleep(8000);


        cd.findElement(By.xpath("/html/body/header/div/div/div/ul/li[6]/a")).click(); //click
shopping cart


        WebElement element = cd.findElement(By.id("trashQuantity_1")); // click delete

        JavascriptExecutor js = (JavascriptExecutor) cd;

        js.executeScript("arguments[0].click();", element);

        cd.quit();

    }
}

```

```
}
```

```
package softwaretestingpackage;
```

```
import static org.junit.Assert.*;
```

```
import java.time.Duration;
```

```
import org.junit.Test;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.JavascriptExecutor;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class tc26_removeAllProduct {
```

```
    @Test
```

```
    public void removeAllProduct() throws InterruptedException {
```

```
        System.setProperty("webdriver.chrome.driver",  
"D:\\Common\\drivers\\chromedriver.exe");
```

```
        ChromeDriver cd = new ChromeDriver();
```

```
        String URL = "https://www.ikea.com/my/en/p/backtrav-bolster-10421000/";
```

```
        cd.get(URL);
```

```
        cd.manage().window().maximize();
```

```
        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(15));
```

```
cd.findElement(By.xpath("/html/body/main/div/div/div/div[2]/div[1]/div/div[2]/div/div/button")).click(); //click add to cart
```

```
Thread.sleep(8000);
```

```
cd.findElement(By.xpath("/html/body/header/div/div/div/ul/li[6]/a")).click(); //click shopping cart
```

```
cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(15));
```

```
JavascriptExecutor js = (JavascriptExecutor) cd;
```

```
js.executeScript("window.scrollTo(0,1500)");
```

```
cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
```

```
cd.findElement((By.xpath("/html/body/div[8]/div[5]/ul/li[2]"))).click(); //click remove all product
```

```
cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(15));
```

```
cd.findElement(By.xpath("/html/body/div[8]/div[6]/div/div[2]/div[2]/div")).click();  
//click remove
```

```
cd.quit();
```

```
}
```

```
}
```

```
package softwaretestingpackage;
```

```
import static org.junit.Assert.*;
```

```
import java.time.Duration;
```

```

import org.junit.Assert;

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;


public class tc27_editProfile {

    @Test

    public void editProfile() throws InterruptedException {

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        String URL = "https://www.ikea.com/my/en/profile/login/";

        cd.get(URL);

        cd.manage().window().maximize();


cd.findElement(By.xpath("/html/body/div[9]/div/div/div[3]/div/div[1]/div/form/div[1]/div/div/div/div/div/input")).sendKeys("abc763188@gmail.com"); //input username


cd.findElement(By.xpath("/html/body/div[9]/div/div/div[3]/div/div[1]/div/form/div[2]/div/div/div/div/div/input")).sendKeys("Abc123456"); // input password


        Thread.sleep(5000);


cd.findElement(By.xpath("/html/body/div[9]/div/div/div[3]/div/div[2]/div/button")).sendKeys(Keys.ENTER); //login


        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));

```

```

        cd.findElement(By.xpath("/html/body/header/div/div/div/ul/li[4]/a")).click(); //my
profile

cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[1]/div[1]/a"
)).sendKeys(Keys.ENTER); //edit

        cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));

        WebElement firstN =
cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[1]/div[2]/di
v[1]/div/form/div/div[1]/div/div/div/div/input")); //input first name

        WebElement lastN =
cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[1]/div[2]/di
v[1]/div/form/div/div[2]/div/div/div/div/input")); //input last name

        firstN.sendKeys(Keys.chord(Keys.CONTROL, "a"), "Test");

        lastN.sendKeys(Keys.chord(Keys.CONTROL, "a"), "test");

        Thread.sleep(5000);

cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[1]/div[2]/di
v[2]/div/button[2]")).sendKeys(Keys.ENTER); //save

        // Verify personal information with assert

        WebElement personalInfo =
cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[1]/div[3]"));

        String str2 = personalInfo.getText();

        Assert.assertTrue(str2.contains("Test test"));

        System.out.println("TestCase 27: Edit Personal Information completed");

        cd.quit();

    }

}

```

```
package softwaretestingpackage;

import static org.junit.Assert.*;

import java.time.Duration;

import org.junit.Assert;
import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class tc28_failedToEditProfile {

    @Test
    public void failedToEditProfile() throws InterruptedException {

        System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

        ChromeDriver cd = new ChromeDriver();

        String URL = "https://www.ikea.com/my/en/profile/login/";

        cd.get(URL);

        cd.manage().window().maximize();
    }
}
```

```
cd.findElement(By.xpath("/html/body/div[9]/div/div/div[3]/div/div[1]/div/form/div[1]/div/div/div/input")).sendKeys("abc763188@gmail.com"); //input username
```

```
cd.findElement(By.xpath("/html/body/div[9]/div/div/div[3]/div/div[1]/div/form/div[2]/div/div/div/div/input")).sendKeys("Abc123456"); //input password
```

```
Thread.sleep(5000);
```

```
cd.findElement(By.xpath("/html/body/div[9]/div/div/div[3]/div/div[2]/div/button")).sendKeys(Keys.ENTER); //login
```

```
cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));
```

```
cd.findElement(By.xpath("/html/body/header/div/div/div/ul/li[4]/a")).click(); //my profile
```

```
cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[1]/div[1]/a")).sendKeys(Keys.ENTER); //edit
```

```
cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));
```

```
WebElement firstN =  
cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[1]/div[2]/div[1]/div/form/div/div[1]/div/div/div/div/input")); //input first name
```

```
WebElement lastN =  
cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[1]/div[2]/div[1]/div/form/div/div[2]/div/div/div/div/input")); //input last name
```

```
firstN.sendKeys(Keys.chord(Keys.CONTROL, "a"), "Test");
```

```
lastN.sendKeys(Keys.chord(Keys.CONTROL, "a"), "123");
```

```
JavascriptExecutor js = (JavascriptExecutor) cd;
```

```
js.executeScript("window.scrollTo(0,500)");
```

```
Thread.sleep(5000);
```



```
cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[1]/div[2]/div[2]/div/button[2]")).click(); //save
```

```
Thread.sleep(5000);
```

```
// Verify error msg with assert
```

```
WebElement verifyErrorMsg = cd.findElement(By.id("regular-dashboard-account-personal-form-lastName-error"));
```

```
String str = verifyErrorMsg.getText();
```

```
Assert.assertTrue(str.contains("Your last name must not contain <>{}[]()*=; or be only spaces, numbers or special characters."));
```

```
cd.quit();
```

```
}
```

```
}
```

```
package softwaretestingpackage;
```

```
import static org.junit.Assert.*;
```

```
import java.time.Duration;
```

```
import org.junit.Test;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.Keys;
```

```
import org.openqa.selenium.WebElement;
```

```
import org.openqa.selenium.chrome.ChromeDriver;
```

```
public class tc29_failedToChangeEmail {
```

@Test

```
public void failedToChangeEmail() throws InterruptedException {

    System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

    ChromeDriver cd = new ChromeDriver();

    String URL = "https://www.ikea.com/my/en/profile/login/";

    cd.get(URL);

    cd.manage().window().maximize();


    cd.findElement(By.xpath("/html/body/div[9]/div/div/div[3]/div/div[1]/div/form/div[1]/div/div/div/div/input")).sendKeys("abc763188@gmail.com"); //input username

    cd.findElement(By.xpath("/html/body/div[9]/div/div/div[3]/div/div[1]/div/form/div[2]/div/div/div/div/div/input")).sendKeys("Abc123456"); //input password

    Thread.sleep(5000);


    cd.findElement(By.xpath("/html/body/div[9]/div/div/div[3]/div/div[2]/div/button")).sendKeys(Keys.ENTER); //login

    cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));

    cd.findElement(By.xpath("/html/body/header/div/div/div/ul/li[4]/a")).click(); //my
profile

    cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[2]/div[1]/a"))
    ).sendKeys(Keys.ENTER); //edit

    cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));

    WebElement email =
    cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[2]/div[2]/div[1]/div/form/div/div/div/div/div/div/div[1]/input")); //input email

    email.sendKeys(Keys.chord(Keys.CONTROL, "a"), "test1234@gmail.com");
```

```

        Thread.sleep(3000);

        WebElement pass =
cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[2]/div[2]/div[1]/div/form/div/div/div/div[2]/div/div[2]/div/input")); //input pass

        pass.sendKeys(Keys.chord(Keys.CONTROL, "a"), "Test123");

        Thread.sleep(5000);

cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[2]/div[2]/div[2]/div/button[2]")).sendKeys(Keys.ENTER); //save

        cd.quit();

    }

}

package softwaretestingpackage;

import static org.junit.Assert.*;

import java.time.Duration;

import org.junit.Test;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class tc30_failedToChangePassword {

```

@Test

```
public void failedToChangePassword() throws InterruptedException {

    System.setProperty("webdriver.chrome.driver",
"D:\\Common\\drivers\\chromedriver.exe");

    ChromeDriver cd = new ChromeDriver();

    String URL = "https://www.ikea.com/my/en/profile/login/";

    cd.get(URL);

    cd.manage().window().maximize();


    cd.findElement(By.xpath("/html/body/div[9]/div/div/div[3]/div/div[1]/div/form/div[1]/div/div/div/div/input")).sendKeys("abc763188@gmail.com"); //input username


    cd.findElement(By.xpath("/html/body/div[9]/div/div/div[3]/div/div[1]/div/form/div[2]/div/div/div/div/input")).sendKeys("Abc123456"); //input password


    Thread.sleep(5000);


    cd.findElement(By.xpath("/html/body/div[9]/div/div/div[3]/div/div[2]/div/button")).sendKeys(Keys.ENTER); //login


    cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));


    cd.findElement(By.xpath("/html/body/header/div/div/div/ul/li[4]/a")).click(); //my
profile


    cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[3]/div[1]/a"))
).sendKeys(Keys.ENTER); //edit


    cd.manage().timeouts().implicitlyWait(Duration.ofSeconds(5));


    WebElement pass =
cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[3]/div[2]/div[1]/div/form/div/div[1]/div/div/div/div/input")); //input current pass


    pass.sendKeys(Keys.chord(Keys.CONTROL, "a"), "Test1234");


    Thread.sleep(2000);
```

```

        WebElement newPass =
cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[3]/div[2]/div[1]/div/form/div/div[2]/div/div/div/div/div/div/input")); //input new pass

        newPass.sendKeys(Keys.chord(Keys.CONTROL, "a"), "Test12345");

        Thread.sleep(2000);

        WebElement conPass =
cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[3]/div[2]/div[1]/div/form/div/div[3]/div/div/div/div/input")); //input confirm pass

        conPass.sendKeys(Keys.chord(Keys.CONTROL, "a"), "Test12345");

        Thread.sleep(5000);

cd.findElement(By.xpath("/html/body/div[9]/div[2]/div/div/div[2]/div[2]/div/div[3]/div[2]/div[2]/div/button[2]")).sendKeys(Keys.ENTER); //save

        cd.quit();

    }

}

```