Black Box Testing

Software Testing Technique

Prepared By

Ottor Mills	ID#: 180917	Email: 180917@gist.edu.cn
David Thomas	ID#: 180912	Email: 180912@gist.edu.cn
Nicoy Smith	ID#: 180902	Email: 180902@gist.edu.cn
Kenneth Anglin	ID#: 180907	Email: 180907@gist.edu.cn

Course Instructor: Thomas Canhao Xu
Course: SWEN3010
Date: April 30, 2019

Table of Contents

Menu Tingz Application

The MenuTingz Application is an application that provides English translations to Chinese menus in the Global Institute of Software Technology canteens. Additionally, the application describes each meal on the menu based on their ingredients and allow students to order directly from their phone requesting a specific meal to be delivered. The stakeholders of the system are the International University of the West Indies (UWI) Students & Personnel, the Restaurant owners in the GIST College Canteen. The UWI Students are the main target audience of the system as the application was designed to cater to their day to day interactions with the chinese menus in the cafeteria.

Testing Method

The testing method that was used in testing the MenuTingz Application was Black Box testing. BLACK BOX TESTING, also known as Behavioral Testing, is a software testing method in which the internal structure, design and implementation of the item being tested is not known to the tester. This method was used to test and determine the application's functionality. Functional testing is a software testing process used within software development in which software is tested to ensure that it conforms with all requirements. Functional testing is a way of checking software to ensure that it has all the required functionality that's specified within its functional requirements. Advantages of Black Box testing are: Unbiased tests because the designer and tester work independently, tester is free from any pressure of knowledge of specific programming languages to test the reliability and functionality of an application / software, facilitates identification of contradictions and vagueness in functional specifications, test is performed from a user’s point-of-view and not of the designer’s and test cases can be designed immediately after the completion of specifications.

Equivalence Partitioning is a common black box testing technique and aims to reduce the number of redundant test cases by eliminating those that generate the same output and do not necessarily reveal defects in a program functionality.

Boundary value testing is the process of testing between extreme ends or boundaries between partitions of the input values.

The testing was done using the automated testing framework Calabash. Calabash is an open source Acceptance Testing framework that allows you to write and execute tests for iOS and Android Apps. It's an Automated User Interface Framework that allows tests to be written in Ruby using Cucumber. Calabash works by enabling automatic UI interactions within a Mobile application such as pressing buttons, inputting text, validating responses, etc.

Test Details

Testing the Registration page:

Testing the vulnerability of the Registration Page.

Input	Expected Result	Actual Result
username: otboss password: password	Pass	Pass
username: \"$ password: \"$	Fail	Pass
username: ! password: !	Fail	Pass
username: “ !” password: “ !”	Fail	Pass
username: password:	Fail	Pass

Testing the Login page:

Testing the vulnerability of the Login page.

Input	Expected Result	Actual Result
username: otboss password: password	Pass	Pass
username: \"$ password: \"$	Fail	Pass
username: admin password: 123456	Pass	Pass
username: “ ” password: “ ”	Fail	Pass
Username: admin Password: 1234567	Fail	Fail

Testing Adding Items to Orders:

Testing if the application can successfully add items to user's the order list.

Input	Expected Result	Actual Result
Touching the "Live Love Home" text	Live Love Home	Live Love Home
Touching the "Pasta with Chicken" text	Pasta with Chicken added to the order list	Pasta with Chicken added to the order list
ADD MORE	List of Items	List of Items
Touching the "Fresh Mush" text	Fresh Mush added to the order list	Fresh Mush added to the order list
Touching the "Rice With Pork" text	Rice With Pork added to the order list	Rice With Pork added to the order list

Testing Removing Items from Orders:

Testing if the application can successfully remove items from the user's order list.

Input	Expected Result	Actual Result
Touching the "REMOVE" text with Pasta with Chicken in order list	Item removed from the order list	Item removed from the order list
Touching the "REMOVE" text with Fresh Mush in order list	Item removed from the order list	Item removed from the order list
Touching the "REMOVE" text with Rice With Pork in order list	Item removed from the order list	Item removed from the order list

Testing Order Placement:

Testing if the application can successfully place a user's order.

Input	Expected Result	Actual Result
Touching the "Make Order" text with Pasta with Chicken in order list	SMS sent to the user about order	SMS sent to the user about order
Touching the "Make Order" text with Rice With Pork in order list	SMS sent to the user about order	SMS sent to the user about order
Touching the "Make Order" with empty list	Error	

Test Results

Testing was done on the vulnerability of the login and sign up requirements and the ability to carry out requirements defined in the MenuTingz Software Requirements Specification Document. Firstly testing the login and sign up features show that the application allows for users to have the same username and allows the user to enter weak passwords. Weak Passwords can make your account information vulnerable to hackers. Another issue was detected where the form fields have no max length value, meaning the form fields allow the user to input passwords and usernames of any length.

The MenuTingz application also does not have an option for users to sign out which made testing the application more difficult. However the application allows users to successfully: Signup , Login into the application, View Restaurant Listings meaning the system allows the user to view a list of restaurants available, View Menus meaning the system allows the user to view the menus from restaurants, Request Order meaning the system allows the user make order requests from their phones, Request Delivery meaning the system provides a delivery option, Send SMS meaning the system compiles the order information into an SMS to be sent to the restaurant manager, Remove orders meaning the system allows the user to remove an order that they have created.

Application Improvements

Based on testing carried out on the MenuTingz application can be improved by setting a minimum password input length of at least 6 characters so that users accounts are less susceptible to hackers, also by making mandatory that users have unique usernames and make users register with their emails for validity purposes or another thing that can also be done is to implement a dual email and username login approach also validating the users email is also a necessity and placing a hold on the account establishing until it is validated happens.

An example is Facebook. You can login to Facebook with your username, your email or even your phone number. The form fields can also be improved by limiting the length of user's inputs. Adding a signing out option for the application as well as displaying an error message when trying to make an order with an empty item list will also improve the application.

Calabash Test on Menu Things Application

```
Feature: All Tests for Menu Tingz app

    Scenario: As a valid user I can log into my app

        When I see "Login"

        Then I press "Login"

        When I see "Username"

        Then I enter text "admin" into field with id "lusername"

        * I go back

        Then I enter text "123456" into field with id "lpassword"
# INCORRECT PASSWORD RESULTS IN AN ERROR TOAST

        * I go back

        * I press "Login Now"

        * I wait for 5 seconds

        * I should see "Restaurants List"


# NOW TESTING ADD ITEMS TO ORDERS

    Scenario: The app should allow the user to add items to their order

        When I see "Live Love Home"

        Then I touch the "Live Love Home" text

        Then I should see "Live Love Home"

        When I see "Pasta with Chicken"

        Then I touch the "Pasta with Chicken" text

        When I see "My Orders"

        Then I touch the "ADD MORE" text

        When I see "Fresh Mush"

        Then I touch the "Fresh Mush" text

        When I see "Rice With Pork"
```

Then I touch the "Rice With Pork" text

#THE CART SHOULD NOW HAVE MULTIPLE ITEMS

NOW TESTING REMOVE ITEM FROM ORDERS

Scenario: The app should allow the user to remove items from their order

When I see "Live Love Home"

Then I touch the "Live Love Home" text

Then I should see "Live Love Home"

When I see "Pasta with Chicken"

Then I touch the "Pasta with Chicken" text

When I see "My Orders"

Then I touch the "REMOVE" text

NOW TESTING ORDER PLACEMENT

Scenario: Users should be allowed to place orders within the app

When I see "Live Love Home"

Then I touch the "Live Love Home" text

Then I should see "Live Love Home"

When I see "Pasta with Chicken"

Then I touch the "Pasta with Chicken" text

When I see "MAKE ORDER"

Then I touch the "MAKE ORDER" text

When I see "Are you sure you want to place this order"

Then I touch the "YES" text

Scenario: A complete test

When I see "Live Love Home"

Then I touch the "Live Love Home" text

```
    Then I should see "Live Love Home"
    When I see "Pasta with Chicken"
    Then I touch the "Pasta with Chicken" text
    When I see "My Orders"
    Then I touch the "ADD MORE" text
    When I see "Fresh Mush"
    Then I touch the "Fresh Mush text
    When I see "Rice With Pork"
    Then I touch the "Rice With Pork" text

# NOW TESTING REMOVE ITEM FROM ORDERS
    Then I touch the "ADD MORE" text
    When I see "Live Love Home"
    Then I touch the "Live Love Home" text
    Then I should see "Live Love Home"
    When I see "Pasta with Chicken"
    Then I touch the "Pasta with Chicken" text
    When I see "My Orders"
    Then I touch the "REMOVE" text

# NOW TESTING ORDER PLACEMENT
    When I see "MAKE ORDER"
    Then I touch the "MAKE ORDER" text
    When I see "Are you sure you want to place this order"
    Then I touch the "YES" text

# It is observed that no error is thrown when checking out with an empty cart
    When I see "Live Love Home"
    Then I touch the "Live Love Home" text
```

Then I should see "Live Love Home"

When I see "Pasta with Chicken"

Then I touch the "Pasta with Chicken" text

When I see "My Orders"

Then I touch the "REMOVE" text

Then I touch the "MAKE ORDER" text

When I see "Are you sure you want to place this order"

Then I touch the "YES" text

6 scenarios (6 passed)
70 steps (70 passed)
1m26.434s

Calabash Test on the vulnerability of MenuTingz Application (First Test)

Feature: Form Vulnerability Test

Scenario: As a new user I should be able to create a new account
When I see "Sign Up"
Then I press "Sign Up"
When I see "Username"
Then I enter text " " into field with id "username"
Then I enter text " " into field with id "password"
* I go back
Then I enter text " " into field with id "confirmpassword"
* I go back
* I wait for 2 seconds
* I press "submit"
* I wait for 5 seconds
* I should see "Restaurants List"
* I go back
When I see "Username"
Then I go back
When I see "Login"
When I see "Username"
Then I enter text " " into field with id "lusername"
* I go back
Then I enter text " " into field with id "lpassword"
* I go back
* I press "Login Now"
* I wait for 5 seconds
* I should see "Restaurants List"

1 scenario (1 passed)
25 steps (25 passed)
1m18.870s

Calabash Test on the vulnerability of MenuTingz Application (Second Test)

Feature: Form Vulnerability Test

Scenario: As a new user I should be able to create a new account
When I see "Sign Up"
Then I press "Sign Up"
When I see "Username"
Then I enter text "!" into field with id "username"
Then I enter text "!" into field with id "password"
* I go back
Then I enter text "!" into field with id "confirmpassword"
* I go back
* I wait for 2 seconds
* I press "submit"
* I wait for 5 seconds
* I should see "Restaurants List"
* I go back
When I see "Username"
Then I go back
When I see "Login"
When I see "Username"
Then I enter text "!" into field with id "lusername"
* I go back
Then I enter text "!" into field with id "lpassword"
* I go back
* I press "Login Now"
* I wait for 5 seconds
* I should see "Restaurants List"

1 scenario (1 passed)
25 steps (25 passed)
1m18.870s

Calabash Test on the vulnerability of MenuTingz Application (Third Test)

Feature: Form Vulnerability Test

Scenario: As a new user I should be able to create a new account
When I see "Sign Up"
Then I press "Sign Up"
When I see "Username"
Then I enter text " !" into field with id "username"
Then I enter text " !" into field with id "password"
* I go back
Then I enter text " " into field with id "confirmpassword"
* I go back
* I wait for 2 seconds
* I press "submit"
* I wait for 5 seconds
* I should see "Restaurants List"
* I go back
When I see "Username"
Then I go back
When I see "Login"
When I see "Username"
Then I enter text " !" into field with id "lusername"
* I go back
Then I enter text " !" into field with id "lpassword"
* I go back
* I press "Login Now"
* I wait for 5 seconds
* I should see "Restaurants List"

1 scenario (1 passed)
25 steps (25 passed)
1m18.870s