Statement and Confirmation of Own Work

A signed copy of this form must be submitted with every assignment.
If the statement is missing your work may not be marked.

Student Declaration

I confirm the following details:

Candidate Name:	SHIHAB SAJID MIRZA
Candidate ID Number:	202100255
Qualification:	NCC LEVEL 4 DIPLOMA IN COMPUTING (L4DC)
Unit:	DDOOCP (With C#)
Centre:	ZCAS UNIVERSITY. LUSAKA, ZAMBIA.
I have read and understood both NCC Education's *Academic Misconduct Policy* and the *Referencing and Bibliographies* document. To the best of my knowledge my work has been accurately referenced and all sources cited correctly. I confirm that this is my own work and that I have not colluded or plagiarised any part of it.	
Candidate Signature:	
Date:	26/10/2021

CONTENTS-

INTRODUCTION……3

Test plan……3

Data selection and execution……5

CODE BOX SURVEY SYSTEM.

Testing.

Introduction

the following report will have a test plan that would be used on the code box survey system for testing. It will also have an additional discussion on how the data for testing was selected and executed. This system is made up of 3 forms namely:

sign-up form, login form and survey form.

Test plan.

The execution of the test plan will be in different sections. These sections will be basically the different components/forms/layers of the system. We will just be using the core test cases to test all of these layers. Furthermore, we can use other testing strategies such as black-box testing, white-box testing or unit testing.

To start off, we will plan how to test the sign-up form.

Testing the sign-up form:

the sign-up form has 5 text boxes. Each of those text boxes are where the user's credentials go. All 5 of the text boxes should be filled in if the person creating an account chooses to create an admin account, but only 4 of them should be filled in if the person chooses to create a consumer account. So, to test the functionality, we need to come up with examples of test cases such as:

- Leaving all fields empty and clicking the create account button.
- Leaving some fields empty and clicking the button.
- Filling in all the fields correctly.
- Trying to creating an admin account and entering the correct secret key.
- Trying to create an admin account and entering the wrong secret key.

These will be al the core test cases for the sign-up form.

We will then move on to test the log-in form using the accounts we have created here in the sign-up form.

Testing the log-in form:

the log-in form has 3 fields where the user enters their credentials, we will use the accounts created in the log-in form to aid us in testing this form. To start-off obviously we will need examples of test cases like:

- All fields are left empty and log-in is pressed.
- Some fields left empty and log-in is pressed.
- Correct credentials are entered.
- Wrong credentials are entered.
- Wrong credentials are entered for more than 6 times (if consumer account).
- Wrong credentials are entered for more than 3 times (if admin account).

And those will be all the core test cases for the login form.

Testing the survey-form:

the survey form consists of 5 different functionalities, though some may be disabled depending on the user type. The functionalities are:

- Taking a survey.
- Creating a survey.
- Deleting a survey.
- Viewing analysis.
- Logging out.

For all these different functionalities we will have to come up with different white-box testing, black box testing and unit testing scenarios. And to be able to do this, we will need to make use of the two different types of account i.e., admin and consumer.

We will also have to make sure that the functionalities like creating and viewing analysis of surveys are going to be disabled for the consumer accounts and the functionality for taking a survey should be disabled for administrator accounts.

After logging out of the survey page, the current surveys created by the admin should be able to reflect on the consumers options for taking survey also.

Data selection and execution.

The data was selected according to its; extremity, normality, abnormality. These were the core criteria of selection throughout the entire systems components/layers.

For example, to test with extreme data, the login form was tested with correct credentials, but the user was an admin and we chose consumer on purpose to test with extreme data.

Normal data was also used to test the correct functioning of the system and to look for bugs or errors that were hanging around and went unnoticed when the development of the system was underway.

Abnormal data was used for the sole purpose of finding out whether or not the system was breaking if tested with abnormal data and this was happening often. Therefore, many exception handlers were put in place to prevent the termination of the program just because of an insignificant error.

In short, the system was made such compatible that it could handle all types of input of data and would give an output or response without the system terminating or crashing.

The system could also handle all types of events without a problem.

The data selection was all done manually and all types of data i.e., normal, abnormal and extreme data were derived in real time of testing.

It should be noted that no testing scripts were used for the testing of this system.

Testing results-

The results of testing for most of the components and units were coming out as desired. Although some places were giving us unexpected/buggy results. These bugs were thankfully rectified. However, there were some bugs that could not be rectified such as the bug for the analysis part, where if you try to analyze a survey with more than one question type, 2 questions from that survey were missing.

Conclusion

The test plan was executed as mentioned and results came close to the expectations. The system was at the point of being thoroughly tested and was now ready for its intended use.