TEST CASE 1 : FRD-1

ABSTRACT

* This usecase is to develop a system to manage the information when a client pucrchase products. This usecase involves the administrator of the system and the client who bought the product. Furthermore, this usecase requires the client to log in or sign up if they want to place their orders. Therefore, our team make this usecase for make sure that happen.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

Introduction of the entire plan

According to the administrator, this usecase is planned to help the agents to place their order the correct ways. Furthermore, the usecase requires investigating the agents’ insights to perform suitable functions on the B2C E-commerce website.

Purpose and scope of the plan

This plan aims to develop the system to ensure the agents’ satisfaction and their information security. The system is required to back up every single piece of informations of the agents and when they log in back.The scope of this is data entry and security.

REQUIREMENTS/SPECIFICATIONS-BASED SYSTEM LEVEL TEST CASES

1. State Transition
2. When users need to log in to the system, they open login page and enter their account details. If the user enter a valid account, user will be logged in successfully. If user enter an invalid account in the first four try, user can re-try. In the fifth try, if user still enter an invalid account, that account will be blocked temporarily.
3. State 1: Login and enter correct

This state will manage by the database system whether or not the user is valid.

State 2: Login and enter incorrectly

This state also manage by the database system to check whether that user is valid or not. If the user enters 5 times but is still not correct their account will be banned for 1 minute and go on.

1. The administrator will test an account with the function log-in to check whether the function is working correctly. Then they will initialize the state where they want to test and then trigger that event
2. Lastly, they will observe the system to see whether the transitions are correct or not.
3. Decision Table
4. The administrator will decide and identify if the user is logged in with the right information that they register or not.
5. Then they will create a test case for the decision point they want to test. Example: They find that the user log in with the right information that they register but can not log in
6. Lastly, the administrator will test the input to see if the system response or not.
7. All pair Testing

* This method will be used when the system is maintained, we will make sure that all the possible log in information is valid.

1. Use Case Testing
2. When the user log in but does not correct five times the system will be banned for 1 minute and go on.
3. Then the system will notification you to reset your password or maybe the information you provide is not correct.
4. Lastly, the system will required you to contact the administrator if you still can not log in

* Because of the account that the agents use is required by the administrator so perhaps there will be some error:

1. The account does not exist
2. Wrong information
3. Trying to login to the other account using script.

TRACEABILITY OF TEST CASES TO USE CASES

1. The account does not exist

* Traceability of this test cases is really simple. The administrator will have the code to check if that account already exist on the database or not. If not they will notificate for you. We will use backward traceability, it will provide the wrong information then requirements with short description.

1. Wrong information

* Traceability of this test cases is to ask the agents again with the notification of the wrong information which is check from the database. The administrator will have the code to check if that account already exist on the database or not. If not they will notificate for you.

1. Trying to login to the other account using script.

TECHNIQUES FOR TEST GENERATION

Techniques used:

* Manual test genenaraion

We use this techniques because it will scan all the case that have no database

* White box based testing:

We use this because white box based testing is on code statements, branches, paths or conditions. It’s a low-level testing but it still can use for this use case

* Criteria used:

EVIDENCE THE TEST CASES, DOCUMENT HAVE BEEN PLACED UNDER

CONFIGURATION MANAGEMENT

REFERENCES