System Test Plan

1. Introduction

The goal of this Test Plan is to inform all who are involved in the test process about the approach, the activities and the deliverables for the project "Pizza Ordering System". This test plan describes a concrete and detailed elaboration of what has been described in the master test plan for this project.

2. Test Cases

2.1 Initialize the system.

2.1.1 Testing object: UI module.

2.1.2 Importance: High

2.1.3 Instructions:

Tester executes the project.

2.1.4 Expected result:

The system should present a default main UI provides entries to other possible UIs.

2.1.5 Cleanup: No need.

2.2 Add employees.

2.2.1 Testing object: System, UI module

2.2.2 Importance: High

2.2.3 Instructions:

Step 1: Tester login the system as Manager.

Step 2: Tester enters UI provided for adding employees (Manage Employee).

Step 3: Tester will try to creates 3 employees sequentially by the following information:

a. Name: Shimon; LoginID: Shimon ;Role: Manager; Password: shimon001;

b. Name: Nathan; LoginID: Nathan ; Role: Cashier; Password: nathan002;

c. Name: Caleb; LoginID: Caleb; Role: Chef; Password: caleb003;

d. Name: Peter; LoginID: Peter; Role: Cashier; Password: peter004;

e. Name: Wrong; LoginID: Peter; Role: Manager; Password: peter005;

2.2.4 Expected result:

a. The system should present successful messages for the first three adding employee cases.

b. The system should present an unsuccessful message for the fifth adding employee case, which indicates that another user with same login ID already exists.

c. The changes should be saved in the database.

2.2.5 Cleanup: Tester should logout the administrator account.

2.3 Edit employee

2.3.1 Testing object: System, UI module

2.3.2 Importance: Medium

2.3.3 Instructions:

Step 1: Tester login to the system as Manager

Step 2: Tester enters the editing employee UI (Manage Employee).

Step 3: Tester selects the employee he wants to edit.

Step 4: Tester edit Caleb's role from chef to cashier, keeping all other info like name, loginID and password the same.

Step 5: Tester edit Nathan's name from Nathan to Peter , keeping all other info same.

Step 6. Tester edit Caleb’s Login ID from Caleb to Peter, keeping all other info same.

2.3.4 Expected result:

a. The role of employee with name Caleb and loginID Caleb is "cashier".

b. The name of employee with loginID Nathan is "Peter".

c. The system presents an error , indicating that there is another user with the same loginID.

d. The changes should be saved in the database.

2.3.5 Cleanup:

a. Tester need to change the role of employee with loginID Caleb from cashier to chef.

b. Tester need to change the name of employee with loginID Nathan from Peter to Nathan.

2.4 Login attempt

2.4.1 Testing object: System, UI module

2.4.2 Importance: High

2.4.3 Instructions:

Step 1: Tester enters the login UI.

Step 2: Tester login to the system by the following information:

a. ID: Shimon; Password: shimon001.

b. ID: Nathan; Password: nathan002.

c. ID: Caleb; Password: caleb003.

d. ID: Caleb; Password: caleb004.

Step 3: Tester logout the system after each login attempt in Step 2.

2.4.4 Expected result:

a. System will present UI for manager in Step 2 - (a).

b. System will present UI for cashier in Step 2 - (b).

c. System will present UI for chef in Step 2 - (c).

d. System will present error message indicates the ID and password are not matched in Step 2 - (d).

2.4.5 Cleanup: No need.

2.5 Create menu

2.5.1 Testing object: System, UI module

2.5.2 Importance: High

2.5.3 Instructions:

Step 1: Tester login to the system as manager, which is the one created with ID 001, password: shimon001.

Step 2: Tester enters the UI for adding menu (Edit Menu).

Step 3: Tester selects Create Menu and try to create the following menus sequentially:

a. Name: "Pizza Menu"; Description: "Menu for Pizza"

b. Name: "Drink Menu"; Description: "Menu for Drink"

c. Name: "Pizza Menu"; Description: "Another menu for Pizza"

2.5.4 Expected result:

a. There will be two menus in the system: "Pizza Menu" and "Drink Menu".

b. The system will present an error message at the third create menu attempt, indicating that name "Pizza Menu" already exists.

c. The changes should be saved in the database.

2.5.5 Cleanup: Tester logout the current account.

2.6 Add menu item

2.6.1 Testing object: System, UI module

2.6.2 Importance: High

2.6.3 Instructions:

Step 1: Tester login to the system as manager, which is the one created with ID 001, password: shimon001.

Step 2: Tester enters the UI for managing menu item (Manage Menu Item)".

Step 3: Tester enters the UI for adding menu item. (Create)

Step 4: Tester try to create the following menu items sequentially:

a. Name: "Cheese Pizza"; Description: "Pizza with cheese"; Price: 5;

b. Name: "Pepperoni Pizza"; Description: "Pizza with pepperoni"; Price: 6;

c. Name: "Cheese Pizza"; Description: "Another pizza with cheese"; Price: 6;

2.6.4 Expected result:

a. There will be two menu items "Cheese Pizza" and "Pepperoni Pizza".

b. The system will present an error message at the third create menu item attempt, indicating that name "Cheese Pizza" already exists.

c. The changes should be saved in the database.

2.6.5 Cleanup: Tester logout the current account.

2.7 Edit menu

2.7.1 Testing object: System, UI module

2.7.2 Importance: Medium

2.7.3 Instructions:

Step 1: Tester login to the system as manager, which is the one created with ID 001, password: shimon001.

Step 2: Tester enters the "Manage menu" UI.

Step 3: Tester selects edits "Pizza Menu":

a. Name: "New Pizza Menu"

b. Description: "New Pizza Menu"

Step 4: Tester edits "Drink Menu":

a. Name: "New Pizza Menu"

b. Description: "Another menu"

2.7.4 Expected result:

a. In Step 2, the system will present all menus: "Pizza Menu" and "Drink Menu".

b. The system will present an error message indicating that "New Pizza Menu" already exists.

c. The system will have menus "New Pizza Menu" and "Drink Menu".

d. The changes should be saved in the database.

2.7.5 Cleanup:

a. Tester edit the "New Pizza Menu" back to:

Name: "Pizza Menu";

Description: "Menu for Pizza"

b. Tester logout the current account.

2.8 Edit menu item

2.8.1 Testing object: System, UI module.

2.8.2 Importance: Medium.

2.8.3 Instructions:

Step 1: Tester login to the system , which is the one created with loginID Shimon, password: shimon001.

Step 2: Tester enters the "Manage menu item" UI.

Step 3: Tester selects and edit the menu item name of "Cheese Pizza":

a. Name: "New Cheese Pizza"

Step 4: Tester edit the menu item name of "Pepperoni Pizza":

a. Name: "New Cheese Pizza"

Step 5: Tester edit the menu item description of “Pepperoni Pizza”

a. Description: “A new kind of cheese pizza”.

Step 6: Tester edit the menu item description of "Pepperoni Pizza":

a. Description: “A new kind of cheese pizza”.

2.8.4 Expected result:

a. After Step 2, the system will present all menu items which are present in the system. These are "Cheese Pizza" and "Pepperoni Pizza".

b. After Step 3, the "Cheese Pizza" is changed to "New Cheese Pizza" . The rest of the fields are the same.

c. In Step 4, the system will present an error message indicating that the name "New Cheese Pizza" already exists.

d. After step 5, The “Pepperoni pizza” description is changed to “A new kind of cheese Pizza. “

e. After step 6, The “Cheese pizza” description is changed to “A new kind of cheese Pizza. “

d. The changes should be saved in the database.

2.8.5 Cleanup:

a. Tester edits "New Cheese Pizza" back to:

Name: "Cheese Pizza";

Description: "Pizza with cheese";

b. Tester logout the current account.

2.9 Add Item in menu:

2.9.1 Testing object: System, UI module.

2.9.2 Importance: Medium.

2.9.3 Instructions:

Step 1: Tester login to the system , which is the one created with LoginID: Shimon, password: shimon001.

Step 2: Tester enters the "Manage menu item" UI.

Step 3: Tester selects menu "Pizza Menu" and then selects add item.

Step 4: Tester selects the menu item name of "Cheese Pizza" and clicks ok.

Step 5: Tester selects Pizza menu and add item again, and then the menu item name of "Pepperoni Pizza" and selects cancel.

Step 6: Tester selects Pizza menu and add item again, and then the menu item name of "Pepperoni Pizza" and selects ok.

Step 7: Tester selects Pizza menu again , then selects add item.

Step 8: Tester selects Drink menu and then selects add item.

2.9.4 Expected result:

a. After Step 3, the system will present all menu items which are present in the system , and not currently added to the current menu. These are "Cheese Pizza" and "Pepperoni Pizza".

b. After Step 4, the "Cheese Pizza" is added to the current menu . System goes back to the previous (Manage Menu). If

c. In Step 5, after Pizza menu is selected, only those menu which are not already added to the current menu will be shown. This is “Pepperoni pizza”. After hitting cancel, the item will not be added to the current menu and system goes back to previous screen.

d. After step 6, after Pizza menu is selected, only those menu which are not already added to the current menu will be shown. This is “Pepperoni pizza”. After hitting ok, the item will be added to the current menu and system goes back to previous screen.

e. After Step 7, no item will be shown as all available items in the store are added to the current menu.

f. After Step 8, the system will present all menu items which are not added to drink menu and currently is in the system. These are “Cheese Pizza” and “Pepperoni Pizza”.

g. The changes should be saved in the database.

2.9.5 Cleanup:

a. Tester logout the current account.

2.10 Delete Item in menu:

2.10.1 Testing object: System, UI module.

2.10.2 Importance: Medium.

2.10.3 Instructions:

Step 1: Tester login to the system , which is the one created with loginID Shimon, password: shimon001.

Step 2: Tester enters the "Manage menu" UI.

Step 3: Tester selects menu "Pizza Menu".

Step 4: Tester selects the menu item name of "Cheese Pizza" and clicks Remove Item.

Step 5: Tester selects Pizza menu.

Step 6: the menu item name of "Pepperoni Pizza" and selects remove item.

Step 7: Tester selects Pizza menu and selects remove item.

2.10.4 Expected result:

a. After Step 3, the system will present all menu items which are present in the menu item. These are "Cheese Pizza" and "Pepperoni Pizza".

b. After Step 4, the "Cheese Pizza" is removed from the current menu . System shows the menus available under Pizza menu which is only “Pepperoni Pizza”. If cheese pizza exists in any other menu, it will not be deleted from that menu.

c. In Step 5, System shows the menus available under Pizza menu which is only “Pepperoni Pizza”.

d. After step 6, “Pepperoni pizza” will be removed from the current menu, but will not be deleted from the available menu items under the store. Meaning if Pepperoni pizza is also included under any other menu, that menu will be unaffected.

e. After Step 7, no item will be shown in available items . If remove item is selected then the system will show a prompt message asking to select a menu item first.

f. The changes should be saved in the database.

2.10.5 Cleanup:

a. Tester logout the current account.

2.11 Delete menu items:

2.11.1 Testing object: System, UI module

2.11.2 Importance: Medium

2.11.3 Instructions:

Step 1: Tester login to the system as manager, which is the one created with ID 001, password: shimon001.

Step 2: Tester enters "Pizza Menu" UI.

Step 3: Tester enters "Delete menu items" UI.

Step 4: Tester deletes "Cheese Pizza" and "Pepperoni Pizza".

2.11.4 Expected result:

a. In Step 3, the system should present all menu items under "Pizza Menu", i.e. "Cheese Pizza" and "Pepperoni Pizza".

b. After Step 4, the "Pizza Menu" will contain no menu items.

c. The changes should also be applied to the file "menu_info".

2.12.5 Cleanup: Tester logout current account.

2.12 Delete menus.

2.12.1 Testing object: System, UI module.

2.12.2 Importance: Medium

2.12.3 Instructions:

Step 1: Tester login to the system as manager, which is the one created with ID 001, password: shimon001.

Step 2: Tester enters "Delete Menus" UI.

Step 3: Tester deletes "Pizza Menu" and "Drink Menu"

2.12.4 Expected result:

a. In Step 2, the system should present all menus, i.e. "Pizza Menu" and "Drink Menu".

b. After Step 3, the system contains no menus.

c. The changes should also be applied to the file "menu_info".

2.12.5 Clean up: Tester logout current account.

2.13 Customer in-house order with cash:

2.13.1 Testing object: System, UI module.

2.13.2 Importance: High.

2.13.3 Instructions:

Step 1: Tester login to the system as cashier, which is the one created with LoginID Nathan, password: nathan002.

Step 2: Tester enters the UI for creating an order.

Step 3: Tester creates an order by the following information:

a. Order type: in-house

b. Order item: Cheese Pizza

b. Quantity: 1

Tester after entering the order item, selects pay and then enters the order type . When prompted for membership info, selects cancel.

Step 4: Tester enters $10 in the amount to pay

2.13.4 Expected result:

a. After Step 3, the system should go to the main menu, exiting the place order screen.

b. After Step 4, the system should present $5 as "change" on the screen and mark the order as "paid".

c. The system contains an order with order ID 001.

2.13.5 Cleanup: Tester logout current account.

2.14 Customer home delivery order

2.14.1 Testing object: System, UI module.

2.14.2 Importance: High

2.14.3 Instructions:

Step 1: Tester login to the system as cashier, which is the one created with ID Nathan, password: nathan002.

Step 2: Tester enters the UI for creating an order.

Step 3: Tester creates an order by the following information:

a. Order type: home-delivery

b. Order item: Cheese Pizza

c. Quantity: 1

d. Address: "100 Laural St, Apt 123"

Tester after entering the order item, selects pay and then enters the order type and the address. When prompted for membership info, selects cancel.

Step 4: Tester enters the credit card information:

a. Card Number: 8888 8888 8888 8888

b. Exp Date: 01/2018

c. CV2: 123

Step 5: Tester enters the amount to pay ($10).

2.14.4 Expected result:

a. After Step 5, the system should switch to check-out UI and should mark the order as "paid".

c. The system contains an order with ID 002.

2.14.5 Cleanup: Tester logout current account.

2.15 Chef marks order to complete

2.15.1 Testing object: System, UI module

2.15.2 Importance: Medium

2.15.3 Instructions:

Step 1: Tester login to the system as chef, which is the one created with loginID Caleb, password: caleb003.

Step 2: Tester enters complete order UI.

Step 3: Tester marks the order ID 001 as complete.

2.15.4 Expected result:

a. After Step 2, tester should be able to see all incompleted orders, i.e. orders with ID 001 and 002.

b. After Step 3, The status of order with ID 001 is "completed".

2.15.5 Cleanup: Tester logout current account.

3. Final cleanup: After all 15 manual sequential tests. The system will be as same as the initial.