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 administrator.

Step 2: Tester enters UI provided for adding employees.

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

a. Name: Shimon; Role: Manager; ID: 001; Password: shimon001;

b. Name: Nathan; Role: Cashier; ID: 002; Password: nathan002;

c. Name: Caleb; Role: Chef; ID: 003; Password: caleb003;

d. Name: Peter; Role: Cashier; ID: 002; Password: peter002;

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 fourth adding employee case, which indicates the employee ID already exists.

c. The system should keep the employee information recorded in a single file "employee\_info".

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 administrator

Step 2: Tester enters the editing employee UI.

Step 3: Tester edit Caleb's (with ID 003) role from chef to cashier.

Step 4: Tester edit Nathan's (with ID 002) name from Nathan to Peter.

2.3.4 Expected result:

a. The role of employee with ID 003 is "cashier".

b. The name of employee with ID 002 is "Peter".

c. The changes should also be applied to the file "employee\_info".

2.3.5 Cleanup:

a. Tester need to change the role of employee with ID 003 from cashier to chef.

b. Tester need to change the name of employee with ID 002 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: 001; Password: shimon001.

b. ID: 002; Password: nathan002.

c. ID: 003; Password: caleb003.

d. ID: 003; 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.

Step 3: Tester 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 system should keep the menu information recorded in a single file "menu\_info".

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 menu "Pizza Menu".

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

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 in under "Pizza Menu": "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 system should keep the item information recorded in the file "menu\_info".

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 "Edit menu" UI.

Step 3: Tester 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 also be applied to the file "menu\_info".

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 ID 001, password: shimon001.

Step 2: Tester enters the UI for menu "Pizza Menu".

Step 3: Tester enters the "edit menu item" UI.

Step 4: Tester edit the menu item "Cheese Pizza":

a. Name: "New Cheese Pizza"

b. Price: 10

c. Description: "A new kind of cheese pizza"

Step 5: Tester edit the menu item "Pepperoni Pizza":

a. Name: "New Cheese Pizza"

b. Price: 9

c. Description: "Another kind of cheese pizza"

2.8.4 Expected result:

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

b. After Step 4, the "Cheese Pizza" is changed to "New Cheese Pizza" with price 10$ and new description as well.

c. In Step 5, the system will present an error message indicating that the name "New Cheese Pizza" already exists.

d. The changes should also be applied to the file "menu\_info".

2.8.5 Cleanup:

a. Tester edits "New Cheese Pizza" backto:

Name: "Cheese Pizza";

Price: 5;

Description: "Pizza with cheese";

b. Tester logout the current account.

2.9 Customer in-house order with cash:

2.9.1 Testing object: System, UI module.

2.9.2 Importance: High.

2.9.3 Instructions:

Step 1: Tester login to the system as cashier, which is the one created with ID 002, 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

c. Quantity: 1

Step 4: Tester enters $10 in the "received cash"

2.9.4 Expected result:

a. After Step 3, the system should switch to checkout UI.

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.

2.9.5 Cleanup: Tester logout current account.

**2.10 Customer home delivery order**

2.10.1 Testing object: System, UI module.

2.10.2 Importance: High

2.10.3 Instructions:

Step 1: Tester login to the system as cashier, which is the one created with ID 002, 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"

Step 4: Tester enters the credit card information:

a. Card Number: 8888 8888 8888 8888

b. Exp Date: 01/2018

c. CVN: 123

2.10.4 Expected result:

a. After Step 3, the system should switch to check-out UI.

b. After Step 4, the system should mark the order as "paid".

c. The system contains an order with ID 002.

2.10.5 Cleanup: Tester logout current account.

**2.11 Chef marks order to complete**

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 chef, which is the one created with ID 003, password: caleb003.

Step 2: Tester enters complete order UI.

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

2.11.4 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".

2.11.5 Cleanup: Tester logout current account.

**2.12 Delete menu items:**

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 "Pizza Menu" UI.

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

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

2.12.4 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.

**2.13 Delete menus.**

2.13.1 Testing object: System, UI module.

2.13.2 Importance: Medium

2.13.3 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"

2.13.4 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".

2.13.5 Clean up: Tester logout current account.

2.14 Delete Employees

2.14.1 Testing object: System, UI module.

2.14.2 Importance: Medium

2.14.3 Instructions:

Step 1: Tester login to the system as administrator.

Step 2: Tester enters "Delete Employee" UI.

Step 3: Tester deletes employees with ID 001, 002, 003.

2.14.4 Expected result:

a. In Step 2, the system should present all employees:

Name: Shimon; Role: Manager; ID: 001; Password: shimon001;

Name: Nathan; Role: Cashier; ID: 002; Password: nathan002;

Name: Caleb; Role: Chef; ID: 003; Password: caleb003;

b. After Step 3, there is no employees in the system.

c. The changes should also be applied to the file "employee\_info".

2.14.5 Cleanup: Tester logout the current account.

3. Final cleanup: After all 14 manual sequential tests. The system will be as same as the initial.