**Hotel Booking System**

**User Acceptance Test**

☺Andy Huang ☻

♪ Patrick Kevorkian ♫

► Can Peng Cai ◄

◄ Steven Kong ►

11/30/2015

CSC 430

1. **Introduction**

This user acceptance test document is to define the test environment, test data set and test procedures for the user acceptance test (UAT). The objectives of UAT are to test the software system against user/system requirement documents which was agreed on by user and development organizations. Based on the results of UAT, the software system can be either accepted (or accepted with conditions) or rejected which will lead actions governed by contractual agreement documents signed between the two parties.

1. **Test Environment**

The test environment will be PCs with windows 7 or later having the hotel booking system application installed on the machine. Furthermore the database will be localized to the machine for testing purposes for easy corrections of schema and data.

1. **Test Data Set**

Username and passwords in the database. All the hotel rooms and their amenities in the database. All the orders and food items in the database. All records and receipts in the database.

Outside informations (user input) would be their name, credit card number, their email address, and their home address.

1. **Exception Report**

The UAT team should document and report any exceptions (or malfunctions, crash etc…) follow a predefined format and procedures defined in this sections.

1. Login text box was not case sensitive, therefore unauthorized access was given.

* FIXED

1. Food items were misplaced in different orders.

* FIXED

1. Price calculations were incorrect.

* FIXED

1. Food textbox does not reset to empty.

* FIXED

1. **UAT Test Result Review**

Andy (Developer): “We have several test procedures to go over today.”

Roy (Commissioner): “Ok before we start please help me to understand what specific test procedures you mean.”

Andy: “Well when we test our programs they must stand up to not only our scrutiny but yours as well.”

Roy: “ Ok proceed.”

Andy: “Alright. First off the application was tested on PS’s with Microsoft Windows 7 and later.”

Roy: “This is acceptable to us, our machines are all upgraded past this point.”

Andy: “Great. Next, given data about your hotels you provided we constructed a database to hold it and use that to test the program. All outside customer information we simulated, normal user info and not.

Roy: “You mean information with mistypes and wrong mismatched data?”

Andy: “Yes exactly.

Roy: “Perfect, and about how many sessions will you spend per specific function of the program?”

Andy: “It could be about a hundred or more. We have actual people trying this out as well as a computer testing many different inputs.”

Roy: “Wow that is great.”

Andy: “When an error is found, all info on the program and the input is recorded.”

Roy: “It seems to us like you have this under control, your scrutiny is high and we agree with these test procedures.”

1. **UAT Test Cases and Test Scripts**

Depend on the number functions in the system to be tested. This section can be a separate document or as an appendix of UTA document.

**6.1 Test Case**

UAT test includes a number of test cases (or test scenarios). In general each test case is designed to test a specific requirement. It includes a test case description, test case initialization (test case setup) and expected result.

**6.2** **Test Script**

For each test case has a number of test scripts that describes the exact actions that a UAT tester should perform in steps during the test for that particular test case. It provides a rigorous format for recording test results.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***TEST SCRIPT: 1.01 (Front Desk Log In)*** | | | | | |
| **Action Step** | **Test Action** | **Expected Results** | **Meets Expected**  **Yes No** | | **Comments** |
| 1 | The Front Desk Employee enters their valid login name and password and clicks the **Login** button. | The system verifies that the ID and password combination matches a valid user, and displays the Front Desk Employee main screen. | ✓ |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***TEST SCRIPT: 1.02 (Kitchen Log In)*** | | | | | |
| **Action Step** | **Test Action** | **Expected Results** | **Meets Expected**  **Yes No** | | **Comments** |
| 1 | Kitchen Staff enters his valid login name and password and clicks the **Login** button. | The system verifies that the id and password combination matches a valid user, and displays the Kitchen Staff main screen. | ✓ |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***TEST SCRIPT: 1.03 (Invalid Log In)*** | | | | | |
| **Action Step** | **Test Action** | **Expected Results** | **Meets Expected**  **Yes No** | | **Comments** |
| 1 | Front Desk Employee or Kitchen Staff enters an invalid login name and password and clicks the **Login** button. | The system verifies that the ID and password combination does not match a valid Front Desk Employee or Kitchen Staff, and displays the “You have entered an invalid ID or password, please try again.” | ✓ |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***TEST SCRIPT: 1.04 (Reset Log In)*** | | | | | |
| **Action Step** | **Test Action** | **Expected Results** | **Meets Expected**  **Yes No** | | **Comments** |
| 1 | Front Desk Employee or Kitchen Staff enters an invalid login name and password and clicks the **Clear** button. | The system resets the *ID* and *password* fields. | ✓ |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***TEST SCRIPT: 1.05 (Exit Log In)*** | | | | | |
| **Action Step** | **Test Action** | **Expected Results** | **Meets Expected**  **Yes No** | | **Comments** |
| 1 | Front Desk Employee or Kitchen Staff clicks the **Exit** button. | The system closes the application | ✓ |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***TEST SCRIPT: 2.01 (Order Food)*** | | | | | |
| **Action Step** | **Test Action** | **Expected Results** | **Meets Expected**  **Yes No** | | **Comments** |
| 1 | Front Desk clicks **Order Food** button. | System will display a form with all available food to be selected from. | ✓ |  |  |
| 2 | Front Desk will enter all the food orders you want and click the **Submit** button. | System will display a confirmation screen asking the teller to confirm the creation of the account, with **YES** and **NO** radio buttons. | ✓ |  |  |
| 3 | Front Desk will edit the order if necessary.. | System will go back to the order and edit as needed. | ✓ |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***TEST SCRIPT: 3.01 Check In*** | | | | | |
| **Action Step** | **Test Action** | **Expected Results** | **Meets Expected**  **Yes No** | | **Comments** |
| 1 | Front Desk clicks **Check In** button. | System will display a form, requesting *first and last name, address, phone number, credit card number, and email address* | ✓ |  |  |
| 2 | Front Desk will enter all the appropriate information and click the **Submit** button. | System will display a confirmation screen asking the Front Desk to confirm the creation of the account, with **YES** and **NO** radio buttons. | ✓ |  |  |
| 3 | Front Desk will click the **NO** button. | System will redisplay the Front Desk screen | ✓ |  |  |