PIZZA ORDERING SYSTEM

**Test Plan**

*Purpose: To describe the System Test Plan for Pizza Ordering System*

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Description |
| 10/29/2014 | 1.0 | Varsha Chandrashekar  Harshil Shah  Sandeep Kasavaraju |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1. Overview 3](#_Toc402372062)

[1.1. Purpose 3](#_Toc402372063)

[1.2. Scope 3](#_Toc402372064)

[2. Testing Summary 4](#_Toc402372065)

[2.1. Scope of Testing 4](#_Toc402372066)

[3. Analysis of Scope and Test Focus Areas 5](#_Toc402372067)

[3.1. Release Content 5](#_Toc402372068)

[3.2. Regression Testing 5](#_Toc402372069)

[3.3. Platform Testing 5](#_Toc402372070)

[4. Unit Test Objectives 6](#_Toc402372071)

[5. System Testing 8](#_Toc402372072)

[5.1 UI testing 8](#_Toc402372073)

[6. Test Strategy 9](#_Toc402372074)

[6.1. Test level responsibility 9](#_Toc402372075)

[6.2. Test Type & Approach 9](#_Toc402372076)

[6.3. Testing Tools 9](#_Toc402372077)

[6.4. Testing Handover Procedure 9](#_Toc402372078)

[6.5. Testing Metrics 9](#_Toc402372079)

[7. Test Environment Plan 10](#_Toc402372080)

[7.1. Test Environment Details 10](#_Toc402372081)

[7.2. Environment Control 10](#_Toc402372082)

[7.3. Environment Roles and Responsibilities 10](#_Toc402372083)

[8. References 11](#_Toc402372084)

# Overview

## Purpose

The purpose of this document is to define:

* The test scope, focus areas and objectives
* The test responsibilities
* The test strategy for the levels and types of test for this release
* The entry and exit criteria
* Any risks, issues, assumptions and test dependencies
* The test schedule and major milestones
* The test deliverables

## Scope

This document details the testing that will be performed by the project team for the Pizza Ordering System project. It defines the overall testing requirements and provides an integrated view of the project test activities. Its purpose is to document:

* What will be tested;
* How testing will be performed;
* What resources are needed, and when

# Testing Summary

## Scope of Testing

### In scope

*The system lets cashiers serve customers in the store and also take phoned in orders. Customers can place orders through kiosks. Customers can order anything on the menu. Customer can modify the order anytime during ordering. Store managers are able to define a menu, modify the menu, and set the special for the day. Chefs can view orders and mark them as complete when the orders are completed. Unit testing and Integration testing is performed to test the above requirements.*

### Out of scope

*Regression testing and Performance testing is out of scope.*

# Analysis of Scope and Test Focus Areas

## Release Content

*The Unit Test plan for our project is described below.*

*The Junit Test classes that are written are as follows:*

* *OrderTest - Order*
* *MenuTest – Menu*
* *MenuItemTest – MenuItem*
* *MenuItemModificationTest - MenuItemModification*
* *PaymentTest – Payment*
* *PaymentByCardTest - PaymentByCard*
* *PaymentByCashTest - PaymentByCash*

## Regression Testing

*Out of scope.*

## Platform Testing

*Out of scope.*

# Unit Test Objectives

| Ref | Function | Test Objective | Evaluation Criteria | X-Ref | P |
| --- | --- | --- | --- | --- | --- |
| OrderTest | | | | | |
| 1 | testAddItem | The objective of the test is to test if the Customer or Cashier is able to add a Menu item to the order. | Expected: Item of type MenuItem is added to Order.  Result: As expected. | OrderItem  MenuItem | High |
| 2 | testRemoveItem | The objective of the test is to test if the item is removed from the order when Customer decides to remove the item | Expected: Item is removed from the order.  Result: As expected | OrderItem  MenuItem | High |
| 3 | testGetPrice | The objective of the test is to test if the total price of the ordered items is computed. | Expected: Sum of all OrderItems is returned.  Result: As expected. | OrderItem | High |
| 4 | testCancelOrder | The objective of the test is to test if the entire order is cancelled when the Customer decides so. | Expected: The order is cancelled.  Result: As expected. | OrderItem | High |

| Ref | Function | Test Objective | Evaluation Criteria | X-Ref | P |
| --- | --- | --- | --- | --- | --- |
| MenuItemTest | | | | | |
| 1 | testParsing | The objective of the test is to test whether menu file is parsed or not | Expected: MenuItem local variables will be initialized  Result: As expected | MenuItem | High |

| Ref | Function | Test Objective | Evaluation Criteria | X-Ref | P |
| --- | --- | --- | --- | --- | --- |
| MenuItemModificationTest | | | | | |
| 1 | testAddNewItem | The objective of the test is to test if new item is added to menu | Expected: New item will be added to menu.txt.  Result: As expected. | Menu  MenuItem | High |
| 2 | testModifyItem | The objective of the test is to test if we can modify item or set special to menuitem | Expected: Changes will be made according to data modified  Result: As expected | Menu  MenuItem | High |
| 3 | testDeleteItem | The objective of the test is to test if we can delete selected item | Expected: Changes will be made according to data deleted. Item's description will be deleted from menu.txt  Result: As expected | Menu  MenuItem | High |
| 4 | testModifyPizza | The objective of the test is to test if we can modify pizza item details or set special to anytype of pizza | Expected: Changes will be made according to data modified  Result: As expected | Menu  MenuItem | High |

| Ref | Function | Test Objective | Evaluation Criteria | X-Ref | P |
| --- | --- | --- | --- | --- | --- |
| MenuTest | | | | | |
| 1 | testFileExist | The objective of the test is to test if file exist in source folder & whether parsing is working | Expected: Toppings & MenuItem variable will be initialized.  Result: As expected. | MenuItem | High |
| 2 | testParsingError | The objective of the test is to test what will happen if file does not match parsing criteria | Expected: Parsing error will be thrown  Result: As expected | MenuItem | High |

# System Testing

## 5.1 UI testing

*The System is started when the PizzaStore.java is run as a Java Application. A Welcome screen is displaying 4 buttons is displayed. The buttons are Employee, Manager, Customer and Exit.*

*When the Employee button is clicked, the user is asked to enter the login credentials, upon success, a screen to place order is displayed.*

*When the Manager button is clicked, the user is asked to enter the login credentials, upon success, a screen providing option to Add menu, Modify Menu and Set Special is displayed.*

*When the Customer button is clicked, the user is directed to Place order page, this is to simulate the concept of Kiosk.*

*When Exit button is clicked, the Welcome screen closes.*

*In the Place order screen, a table of existing Menu is displayed, along with 4 buttons for Place/Modify order, Finish, Add Toppings and Remove item.*

*When the user selects a Menu item from the table, the user can click Add to add item to the order. It is repeated for all the items the User wants to add.*

*When the user clicks of the added item and selects Add topping button, a screen containing all the available toppings is displayed and the user is allowed to select the topping for the Whole pizza or Left or Right side of the Pizza. Radio buttons are provided to perform this operation. The window is closed and the user is navigated back to the Place order screen.*

*When the user clicks on the Remove button, items are removed from the order.*

*When the user clicks on the Finish button, he is redirected to Make Payment window.*

*In Manager window, a table of existing Menu is displayed, along with 3 buttons for Add Item, Modify Item, and Delete item.*

*When user selects add item, new screen asking description of item to be added will pop up. After confirming details, system will add new item to its menu list and displays' latest menu.*

*When user selects modify item, new screen asking description of selected item to be modified will pop up. After confirming details, system will modify details of selected item to its menu list and displays' latest menu.*

*When user selects delete item, new screen asking confirmation about deletion of item to be added will pop up. After confirming details, system will delete selected item from its menu list and displays' latest menu.*

# Test Strategy

## Test level responsibility

|  |  |  |  |
| --- | --- | --- | --- |
| Test Level | External Party | Proj Team | Business |
| Unit Testing |  | P |  |
| Integration Testing |  | P |  |

## Test Type & Approach

|  |  |
| --- | --- |
| Test Type | Objectives |
| Progression Requirements | The objectives are to verify that the application:   * Meets the defined requirements; * Performs and functions accurately; * Correctly handles error conditions; * Interfaces function correctly; * Data load is successful.   Functional testing will occur in an iterative and controlled manner, ensuring the solution matches the defined requirements. |

### Data Requirements

*3 text files are provided. Employees.txt, to store the name of the stakeholders with the login credentials. Menu.txt to store menu items. Config.txt to store the start up configuration.*

### Resources & Skills

* A resource with Java skills;
* A resource with internet technologies understanding.

## Testing Tools

The following tools will be used for testing:

| Process | Tool |
| --- | --- |
| Test case creation | JUnit on Eclipse |

## Testing Handover Procedure

None.

## Testing Metrics

Out of scope

# Test Environment Plan

## Test Environment Details

### Testers

*Define the number of testers who will be involved in testing and their:*

* System access requirements
* Hardware requirements

### Hardware and Firmware

*Nothing specific.*

### Software

*Eclipse must be installed with appropriate jars and plugins to run Junit.*

### Interfaces

*None*

### Other Materials

*None.*

## Environment Control

* Software release control: Github.

## Environment Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| Role | Staff Member | Responsibilities |
| Project Developer | Varsha Chandrashekar | Responsible for overall UI design, implementation of logic for Order (place, modify, cancel), Add toppings, View Order status, Login logic, Order backup, Customer backup, Welcome screen. |
| Project Developer | Harshil Shah | Responsible for Manager User Interface, implementation of logic of Menu Item modification, deletion, addition , & setting special for today. Also responsible for integration of all work done by project developers & for documentation |
|  |  |  |

# References

The following documents have been used to assist in creation of this document.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Document name** | **Version** | **Comments** |
| 1 | http://www.computing.dcu.ie/~davids/courses/CA267/ieee829mtp.pdf |  |  |
| 2 | http://www.softwaretestinghelp.com/test-plan-sample-softwaretesting-and-quality-assurance-templates/ |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |