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 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Overview 4

1.1. Purpose 4

1.2. Scope 4

2. Testing Summary 5

2.1. Scope of Testing 5

3. Analysis of Scope and Test Focus Areas 6

3.1. Release Content 6

3.2. Regression Testing 6

3.3. Platform Testing 6

4. Progression Test Objectives 7

5. Progression Test Objectives 8

6. Other Testing 9

6.1. Security 9

6.2. Stress & Volume Testing (S&V) 9

6.3. Connectivity Testing (CT) 9

6.4. Disaster Recovery/Back Up 9

6.5. Unit Testing 9

6.6. Integration Testing 9

7. Test Strategy 10

7.1. Test level responsibility 10

7.2. Test Type & Approach 10

7.3. Build strategy 10

7.4. Test Execution Schedule 10

7.5. Facility, data, and resource provision plan 11

7.6. Testing Tools 11

7.7. Testing Handover Procedure 12

7.8. Testing Metrics 12

8. Test Environment Plan 13

8.1. Test Environment Man 13

8.2. Test Environment Details 13

8.3. Establishing Environment 13

8.4. Environment Control 14

8.5. Environment Roles and Responsibilities 14

9. Assumptions and Dependencies 15

9.1. Assumptions 15

9.2. Dependencies 15

10. Entry and Exit Criteria 16

11. Administrative Plan 17

11.1. Approvals 17

11.2. Test Milestones and Schedule 17

11.3. Training 17

11.4. Defect Management 17

12. Definitions 18

13. References 19

14. Points of Contact 20

# 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*
* *MenuItem – MenuItem*
* *ToppingTest - Topping*
* *PaymentTest – Payment*
* *CashierTest – Cashier*
* *ChefTest – Chef*

## Regression Testing

*Out of scope.*

## Platform Testing

*Out of scope.*

## Unit Testing

## Unit Testing

# 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 |
| --- | --- | --- | --- | --- | --- |
| PaymentTest | | | | | |
| 1 | getAmountTest | The objective of the test is to test if the Customer or Cashier is able to get the right amount during payment. | Expected: If the amount is $10.0 , function must return $10.0  Result: As expected. | Payment | High |
| 2 | processCouponTest | The objective of the test is to test if the coupon provided is authentic or not. | Expected: If given coupon code exists in the coupon.txt calculate discount and return true.  Result: As expected | Payment  Coupon.txt | High |
| 3 | discountedPriceTest | The objective of the test is to test if the discount has been deducted from total price and returned. | Expected: Discount mentioned in coupon.txt is subtracted from the actual price.  Result: As expected. | Payment  Coupon.txt | High |

| Ref | Function | Test Objective | Evaluation Criteria | X-Ref | P |
| --- | --- | --- | --- | --- | --- |
| PaymentByCashTest | | | | | |
| 1 | getAmountTest | The objective of the test is to test if the Customer or Cashier is able to get the right amount during payment. | Expected: If the amount is $10.0 , function must return $10.0  Result: As expected. | Payment | High |
| 2 | processCouponTest | The objective of the test is to test if the coupon provided is authentic or not. | Expected: If given coupon code exists in the coupon.txt calculate discount and return true.  Result: As expected | Payment  Coupon.txt | High |
| 3 | discountedPriceTest | The objective of the test is to test if the discount has been deducted from total price and returned. | Expected: Discount mentioned in coupon.txt is subtracted from the actual price.  Result: As expected. | Payment  Coupon.txt | High |

| Ref | Function | Test Objective | Evaluation Criteria | X-Ref | P |
| --- | --- | --- | --- | --- | --- |
| PaymentByCardTest | | | | | |
| 1 | authenticateCardTest | The objective of the test is to test if the Customer or Cashier is able authenticate card correctly | Expected: If all fields are properly filled Done button appears.  CardNo is 16 digits and CVV is 3 digits.  Result: As expected. | PaymentByCard | High |
| 2 | getCardNoTest | The objective of the test is to test if last 4 digits of cardno are returned or not. | Expected: If card is authenticated then result is last 4 digits.  Result: As expected | PaymentByCard | 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.*

*<<To be Added in how to run txtx >>*

*<<incomplete>>*

# 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 Develpoper and System Tester | Sandeep Kasavaraju | Responsible for UI design ,implementation logic for handling the payments .Assistance with Menu actions in the initial UI.  Responsible for testing the system as a whole. |
|  |  |  |

# Definitions

The following acronyms and terms have been used through out this document

|  |  |
| --- | --- |
| **Term/Acronym** | **Definition** |
| OrderItem | It is a conceptual class which represents a MenuItem that is added in the Order. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 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/ |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |