**Point of Sales Transaction System for Domino’s Pizza**

**Software Test Plan**

**Version 1.0**

**November 11, 2014**

**Number One Best Professional Development Practices**

**Authors: Abu Kebbie-Anthony, Chris McEligot, Nicolas Mouriski, Richard Tyaba**

1 Introduction 1

1.1 Purpose 1

1.2 Scope 1

1.3 Definitions, Acronyms, and Abbreviations 1

1.4 References 1

1.5 System Overview 1

2 Requirements to be Tested 2

3 Testing Approach 3

4 Test Process 3

4.1 Unit Testing 3

4.2 Integration Testing 3

4.3 Validation and System Testing 3

5 Reporting and Corrective Action 4

6 Test Environment 4

7 Test Procedures 4

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Authors** |
| 11/11/14 | 1.0 | Initial version outlining the testing process to be utilized | Abu Kebbie-Anthony, Chris McEligot, Nicolas Mouriski, and Richard Tyaba |

# Introduction

The Software Test Plan (STP) outlines the purpose and scope of testing the software that is being produced by the Number One Best Professional Development Practices (#1BPDP) team. This document will layout the testing process, from conception to practice.

## Purpose

This document will contain the testing philosophy of #1BPDP. It is a guideline to ensure the highest quality product is produced from the team during and after software development.

## Scope

The scope of this document is #1BPDP’s process of testing from the beginning of software development to the distribution of the final product.

## Definitions, Acronyms, and Abbreviations

#1BPDP – Number One Best Professional Development Practices

GUI – Graphical user interface

POS – Point of sales

SRS – Software Requirements Specification

STP – Software Test Plan

STV – Software Test Volume

## References

* Customer requirements specified on September 4th, 2014
* Software Requirements Specification (SRS) version 2.1; October 28, 2014; #1BPDP
* Traceability Matrix version 2.0; October 28, 2014; #1BPDP
* Software Test Volume (STV) version 1.0; November 11, 2014; #1BPDP

## System Overview

The objective of this project is to develop a point of sales (POS) transaction system for Domino’s Pizza. The system will have the ability to log on with different user accounts, contain a graphical user interface (GUI) for placing orders consisting of common items for sale by Domino’s Pizza, and display an onscreen receipt for a completed order.

# Requirements to be Tested

According to the Software Requirements Specification (SRS), this software project has thirty-three functional requirements that need to be tested. In addition to the functional requirements, the software project will have system, interface, and performance requirements that will also be tested.

1. Functional requirements
   1. Login Screen Components
   2. Employee Code Field
   3. Login Button
   4. Invalid Login Information
   5. Validation Message
   6. Order Placement
   7. Item Quantity
   8. Adding a Pizza
   9. Specialty Pizzas
   10. Toppings
   11. Toppings Display
   12. Price Display
   13. Order Cancellation
   14. Order Completion
   15. Onscreen Receipt Components
   16. Order Number Construction
   17. Order Number Persistency
   18. Empty Order Notification
   19. Add New Employee
   20. Existing Four-Digit Code
   21. Edit Employee
   22. Delete Employee
   23. Attempt to Delete Initial User Account
   24. Account Persistency
   25. Configurable Items
   26. Configuration Ranges
   27. Configuration Persistency
   28. Current User Display
   29. User Logout
   30. Order in Progress Prompt
   31. Initial User Account
   32. Initial Account Permissions
   33. Navigation links
2. System Requirements
   1. Program Works on Minimally Specified System
   2. Program Works on Windows 7
3. Interface Requirements
   1. Program Receives Keyboard Input
   2. Program Receives Mouse Input
4. Performance Requirements
   1. System Load Time Does Not Exceed 5 Seconds
   2. Order Manipulation Response Time Does Not Exceed 1 Second
   3. Order Completion Response Time Does Not Exceed 1 Second
   4. System Navigation Response Time Does Not Exceed 1.5 Seconds

# Testing Approach

As it is necessary, the software developers will perform white and clear box testing as the software is built. This means the developer will test the functionality of their individual portion of code as they check for syntax and logical errors. As the smaller portions of code are integrated into the larger program, the ability to perform black box testing becomes available. The requirements listed above are all of the properties of the program that need to be black box tested.

# Test Process

This program will be built in Java using an object-oriented approach; therefore this project will be tested according to an object-oriented test process.

## Unit Testing

This program’s unit testing will be associated with white box testing. This will be correlated to class testing because of operation encapsulation and the state behavior of the class.

## Integration Testing

Integration testing will also be part of the process of white box testing because of the level of development involved. The integration testing will primarily emulate use-based testing. The tests will begin with independent classes and then proceed to dependent testing. The dependent layers will be tested until the entire system is finished.

## Validation and System Testing

This is the level of testing where black box testing occurs. After the system has been integrated and completed, user-defined actions can be used to test user-defined outputs. The majority of the user-defined actions and outputs are what have been listed in the requirements to be tested. Scenarios will be constructed and performed in order to correctly assess the use-cases based in the requirements.

# Reporting and Corrective Action

Test reporting and corrective action will be done through the Software Test Volume (STV). This set of test procedures will be performed and will evaluate if the requirements have been fulfilled or if they fail to meet what is expected. If a test fails, the developer will be handed the failed test and will correct the issue. When the issue is resolved, the test will then be run again on the new software. This cycle will repeat until the test passes. When this or any other test passes, that test will be filed in the STV as a reference. There should be no need to retest passed tests unless a major change happens to the software. If a major software change were to happen, it will be documented and new testing will need to be performed on all test procedures.

# Test Environment

The formal test environment used will be a fresh version of the software on a computer system that is indicative of the client’s computer system. A fresh version of the software can be defined as the latest edit of the software project. Specifically, the computers in the lab at Schaefer’s 165 will be the testing environment.

# Test Procedures

Test procedures are located in the STV.