CS691 – Computer Science, Fall 2018

Pace University



**SYSTEM TEST PLAN**

**CIRCULAR PIE**

**Table of Contents**

[**1.** **INTRODUCTION** 3](#_Toc133783057)

[**2.** **TESTING SCOPE** 3](#_Toc133783058)

[**3.** **TESTING OBJECTIVES** 3](#_Toc133783059)

[**4.** **TEST PROCESS DEFINITION** 7](#_Toc133783060)

[**5.** **APPROACH TO SYSTEM TESTING** 9](#_Toc133783061)

[**6.** **ENTRY/EXIT CRITERIA** 10](#_Toc133783062)

[**7.** **ENVIRONMENTAL NEEDS** 11](#_Toc133783063)

[**8.** **ROLES AND RESPONSIBILITIES** 12](#_Toc133783064)

[**9.** **TEST CYCLES AND SCHEDULE** 14](#_Toc133783065)

[**10. RISKS AND CONTINGENCIES** 15](#_Toc133783066)

# **INTRODUCTION**

This document describes the System Test Plan that provides a common understanding among the “Circular Pie” project stakeholders on the scope, objectives, and approach to performing the system testing. Also, the document explains the features to be tested, testing entry/exit criteria, resource and responsibilities, and testing schedule.

# **TESTING SCOPE**

The testing scope includes two perspectives - the functional scope and technical scope.

The functional scope includes the following modules of the “Circular Pie” system: This includes testing the user interface, registration and login functionality, menu display and ordering process, payment processing, order confirmation and tracking, delivery notifications, and customer support.

The technical scope includes the following architectural components:

* Web browser
* Application server
* Database server
* Content server

# **TESTING OBJECTIVES**

Test items to be tested should include the following:

**Software Code:** This includes the software programs, scripts, and other code that make up the Circular pie. The version or revision level of each code module should be identified to ensure that the correct version is being tested.

**Data Files:** The data files used by the system, such as customer information, order details, and delivery information, should be identified and their version or revision level specified.

**User Documentation:** The user documentation, including the requirements specification, design specification, user guide, operations guide, and installation guide, should be identified and their version or revision level specified.

The characteristics of the transmittal media used to transfer the test items should also be identified, such as the need for logical or physical transformations before testing can begin.

For example, if the software code is stored on tape, it may need to be transferred to disk before testing can begin.

**Features to be tested.**

Software Items to be tested include the following:

**User Interfaces:** This includes the web interfaces through which the users interact with the system to order and track pizza delivery.

**Database Management:** The database should be tested to ensure that it can handle large volumes of data and is properly secured to protect the customer’s personal and payment information.

**Order Processing:** The order processing module should be tested to ensure that it can handle multiple orders simultaneously, can apply promotions and discounts, and properly apply taxes and fees.

**Payment Processing:** The payment processing module should be tested to ensure that it securely handles payment transactions and refunds.

**Delivery Management:** The delivery management module should be tested to ensure that it can track delivery status, assign delivery person, and communicate delivery status to the customer.

**Reporting and Analytics:** The reporting and analytics module should be tested to ensure that it can provide insights on customer orders, order history, popular items, and other relevant metrics.

Software Features to be tested include the following:

**User Registration and Login:** The registration and login process should be tested to ensure that it is easy to use and properly secured to protect user information.

**Order Placement:** The order placement process should be tested to ensure that it is user-friendly, supports customizations, and properly handles payment and delivery information.

**Order Tracking:** The order tracking feature should be tested to ensure that it provides accurate information on the status of the order and estimated delivery time.

**Customer Support:** The customer support feature should be tested to ensure that it is easily accessible and responsive to customer inquiries and complaints.

Combinations of software features will also be tested to ensure that the system is functioning as intended when multiple features are being used simultaneously. For example, a combination of user registration and order placement might be tested to ensure that user registration information is properly associated with the order being placed. The test design specification associated with each combination of features will be identified and will include tests for each feature being used in combination with other features.

The test design specification for each feature and combination of features will provide a clear and detailed description of the test scenarios and expected results. It will also include the testing environment, test data, and any special testing procedures or tools that are required.

**Features not to be tested.**

**Third-Party Integrations (Advertiser):** Features that require integration with third-party systems or services will not be tested.

**Unsupported Browsers or Operating Systems**: Features that are not supported by certain browsers or operating systems will not be tested. For example, if the system is not supported on Internet Explorer, testing on that browser may not be performed.

# **TEST PROCESS DEFINITION**

**Test Process Phases and Tasks**

Test process definition is the process of defining the activities that will be carried out during testing. The test process typically consists of five phases:

* Test Planning
  + Define scope and objectives of testing.
  + Define roles and responsibilities.
  + Define testing approach.
* Test Design
  + Identify test ideas, define an approach to designing test cases.
  + Develop test case specifications.
  + Measure test coverage
  + Determine requirements for test data.
* Test Preparation
  + Setup a test environment
  + Provision test data
  + Install the software in the test environment.
* Test Execution
  + Execute all test cases.
  + Find and report software defects.
  + Evaluate the system stability.
  + Validate all target features.
* Test Reporting
  + Summarize and report the test execution results.
  + Report defect metrics
  + Evaluate the test exit criteria.
  + Create a test completion report, submit for stakeholder approval.
  + Obtain stakeholder signoff on system testing.

**TEST DELIVERABLES**

The following documents and data will be included as test deliverables:

**Test Plan** - A comprehensive document that outlines the testing approach, objectives, scope, and schedules.

**Test Design Specifications** - Detailed documentation of the test design and the approach used to verify each feature.

**Test Case Specifications** - Detailed documentation of each test case and the input data, output data, and expected results.

**Test Procedure Specifications** - Detailed documentation of the procedures to be followed during test execution, including setup, execution, and teardown steps.

**Test Item Transmittal Reports** - Documentation of the test items that are transmitted to the testing team for testing.

**Test Logs** - Detailed records of the testing activities performed during the testing phase, including test case execution results, defects found, and other relevant information.

**Test Incident Reports** - Documentation of any issues or defects discovered during the testing phase, including their severity and priority.

**Test Summary Reports** - A comprehensive document that summarizes the testing activities, including the results of each test case, any issues discovered, and recommendations for future testing.

# **APPROACH TO SYSTEM TESTING**

**Approach to Functional Testing**

The overall approach to system testing for Circular pie system will involve both functional and non-functional testing.

**Approach to Functional Testing:**

The approach to functional testing for our system will be based on the Black-box method:

• Test cases will be designed using some formal black-box techniques such as boundary-value analysis, equivalent-class partitioning, cause-effect graphing, decision tables, and state-transition testing, where applicable.

• Test execution will be conducted manually, from the user perspective and based on formal test case specifications.

• Functional testing will cover all the features and functionalities of the system such as order placement, payment processing, order tracking, delivery management, etc.

• Test results will be captured and reported in test execution logs.

**Approach to Non-Functional Testing:**

The approach to non-functional testing for our system will include:

• Performance testing to ensure that the system can handle the expected load and response times are within acceptable limits.

• Security testing to ensure that the system is secure against unauthorized access, data breaches, and other security risks.

• Usability testing to ensure that the system is user-friendly and easy to use for customers.

The test results will be captured and reported in test execution logs and summarized in the test summary report.

# **ENTRY/EXIT CRITERIA**

This section defines both Entry and Exit Criteria for test execution and is intended to establish a common understanding about the conditions of when the test execution can start and when it can stop.

**ENTRY CRITERIA**

* The test Entry Criteria include the following items:
* The build is produced and deployed to the test environment.
* The system test plan is produced and approved.
* The test environment is ready for testing.
* Test Designs and test case specifications are completed.

**EXIT CRITERIA**

* The test Fail Criteria include the following items:
* All requirements, in scope of testing, are covered by test cases.
* All test cases have been executed.
* Zero defects of Critical and High-severity remain open.
* Open defects of Medium and Low severity have known workarounds.
* Test Summary report is produced and published.

# **ENVIRONMENTAL NEEDS**

The Test Environment should be available to start test execution. It includes a laptop with a virtual machine running the web server and database. The Website must be secure to protect sensitive information such as credit card details to ensure customer trust.

Application: Circular Pie

Type: Application Architecture

View: Process View

Style: Client-Server Pattern

Diagram

Description automatically generated

# **ROLES AND RESPONSIBILITIES**

The project team has seven members that are assigned various project roles including Project Manager, Product Owner, Lead Business Analyst, Lead Developer, DBA, and Lead QA Analyst. Their responsibilities are defined in the table below.

|  |  |
| --- | --- |
| **PROJECT ROLES** | **ROLE RESPONSIBILITIES** |
| Project Manager | Responsible for planning, executing, and closing projects, managing project timelines, budgets, and resources, and ensuring that project goals and objectives are achieved. |
| Product Owner | The product owner is responsible for defining, prioritizing, and communicating the product vision and requirements to the development team. |
| Lead Developer | Responsible for writing code, designing, developing, and testing software solutions, and ensuring that the software meets quality and performance standards. |
| Lead Business Analyst | Responsible for analyzing business requirements and translating them into functional requirements for the project and working with stakeholders to ensure that the project aligns with business needs. |
| DBA | Responsible for the performance, security, and reliability of an organization's databases. |
| Lead QA Analyst | Responsible for testing the software to ensure that it meets functional requirements, is user-friendly, and is free of bugs or defects. |
| Software Developer | The role of a software developer is to assist in the development process. |

# **TEST CYCLES AND SCHEDULE**

**Cycle 1. User Experience I**

• Test cases will be designed to test the input and filtering functionality of the User Experience Module.

• Test scenarios will be executed manually to ensure that the user can input the order details correctly and the system filters the data as expected.

• Test results will be captured and reported in test execution logs.

**Cycle 2. User Experience II**

• Test cases will be designed to test the pizza generation and viewing functionality of the User Experience Module.

• Test scenarios will be executed manually to ensure that the system generates pizza based on the user's input and the user can modify the pizza as expected.

• Test results will be captured and reported in test execution logs.

**Cycle 3. Payment and User Register/Login**

• Test cases will be designed to test the payment and user registration/login functionality of the Payment and User Register/Login Module.

• Test scenarios will be executed manually to ensure that users can register/login, payment processing works correctly, and user data is stored securely.

• Test results will be captured and reported in test execution logs.

# **10. RISKS AND CONTINGENCIES**

There is a chance of technical difficulties such as website crashes, sluggish loading times, and payment processing failures. - A contingency plan can involve periodic website maintenance and backups, load testing to ensure the website can handle excessive traffic, and collaboration with a reliable payment processing business.