**UMGC City Application**

**Test Report**

**Version 1.0**

**April 23, 2020**

**SWEN 670 Spring 2020 – UMGC City Team 1**

Christy Gilliland

Krystina Poling

Jack Amnuaysirikul

Tarig Abasit

Ziad Elharaoui

Daniel Abresch

Patience Okereke

Melanie Meek

**Table of Contents**

[1. Introduction 1](#_Toc38844207)

[1.1 Purpose 1](#_Toc38844208)

[1.2 Scope 1](#_Toc38844209)

[1.3 Application Overview 1](#_Toc38844210)

[2. Testing 2](#_Toc38844211)

[2.1 Test Strategy 2](#_Toc38844212)

[2.1.1. Unit Testing 2](#_Toc38844213)

[2.1.2 White Box Testing/Code Coverage Testing 3](#_Toc38844214)

[2.1.3 Integration Testing 3](#_Toc38844215)

[2.1.4 Functional Testing 3](#_Toc38844216)

[2.1.5 Black Box Testing/ User Acceptance Testing (UAT) 4](#_Toc38844217)

[2.1 Tests Conducted 4](#_Toc38844218)

[3. Test Execution Findings 6](#_Toc38844219)

[3.1 Metrics 6](#_Toc38844220)

[4. Exit Criteria 12](#_Toc38844221)

[5. Conclusion 13](#_Toc38844222)

# 1. Introduction

This Test Report describes the purpose, scope, approach, and results of all testing activities that have been carried out for the UMGC City Application Version 1.0.

## 1.1 Purpose

This document explains the various activities performed as part of the testing of the ‘UMGC City Web/Map-Based’ application. The Test Report shall have the following objectives:

* Describe the test approaches used in testing the UMGC City Application
* Summarize the tests conducted
* Discuss test findings

## 1.2 Scope

The scope of this Test Report is limited to specific software test cases necessary to verify that the map/web-based modules of UMGC City Application have met the software requirements as described in the project SRS. This Test Report provides a summarization to the stakeholders of the project team’s test findings the strategies used to obtain results. In addition, this Test Report also communicates to the stakeholders the current responsibilities of each testing resource, the progress of the testing, and provide detailed status updates on any of the bugs found.

**Features that will not be tested:**

* ChatBot – this feature was not tested as it falls out of scope for Team 1 because it was designed by UMGC City Team 2.

## 1.3 Application Overview

The UMGC City Application is a collection of interfaces designed to assist city officials in creating a database and a customer-facing HTML interface. The customer-facing HTML interface is created from a pre-configured HTML template and data collected from city

management users which is stored in the application’s database. The result of the application’s deliverable is a static HTML page in an easy-to-navigate format for use by city residents. The UMGC City Application is intended to integrate with current websites employed by the cities. However, it does not imply suitability to wholly replace current system.

# 2. Testing

Software testing for the UMGC City Application incorporates several tests, that are used to verify the application’s functionality and ensures the software satisfies all identified requirements.

## 2.1 Test Strategy

The following test strategies were used to assess the UMGC City Application: unit, integration, and functional testing. Testing will be performed iteratively and incrementally throughout the UMGC City Application’s software life cycle to ensure the best overall quality of the application. In addition, tests artifacts follow the same iterative and incremental approach as these documents are living documents and are updated and shared with stakeholders with the intention of keeping all stakeholders abreast to the latest testing activities and bug tracking updates.

## 2.1.1. Unit Testing

Unit testing is a software testing methodology that evaluates the correctness of individual sub-components of the application. Unit testing helps to show that the code works as intended. The threshold level of unit testing for this project is 75% code coverage. Test coverage will be measured by a completed matrix of testable requirements and test cases (see Requirements Traceability Matrix in Software Test Plan). Unit tests will be designed by the developers and will use white box testing techniques.

## 2.1.2 White Box Testing/Code Coverage Testing

This type of testing is based upon white box techniques. The application is tested by utilizing functional test cases written by developers based on actual code.

|  |  |
| --- | --- |
| Test Objective: | Validate internal framework, objects and components |
| Technique: | Execute each test case (JUnit):   * The expected results occur when valid data is used. * The appropriate error / warning messages are displayed when invalid data is used. * Each use case is properly applied. |
| Completion Criteria: | * All planned tests have been executed. * All identified defects have been addressed. |

## 2.1.3 Integration Testing

Integration testing is the process of testing between two or more modules/units to ensure the interfaces are performing correctly in tandem. UMGC City Team 1 will employ the Big Bang Approach for integration testing. In [Big Bang integration testing](http://tryqa.com/what-is-big-bang-integration-testing/), all components or modules are integrated simultaneously, after which everything is tested as a whole. This approach is ideal for smaller projects, such as the UMGC City Application, that do not have any external interfaces.

## 2.1.4 Functional Testing

Functionality testing focuses on direct features of the application. The purpose of functionality testing is to ensure that each feature of the application work as intended. This project, functional testing will utilize a black-box approach. The goal of these test cases is to determine if the actual output/action is the expected output/action. UMGC City Team 1 selected Selenium, a free functional testing tool, to test the features of the UMGC city application.

## 2.1.5 Black Box Testing/ User Acceptance Testing (UAT)

This type of testing is based upon black box techniques. The application is tested by interacting with the application and analyzing the output.

|  |  |
| --- | --- |
| Test Objective: | Ensure proper application navigation, data entry, processing, and retrieval. |
| Technique: | Execute each test case:   * The expected results occur when valid data is used. * The appropriate error / warning messages are displayed when invalid data is used. * Each use case is properly applied. |
| Completion Criteria: | * All planned tests have been executed. * All identified defects have been addressed. |

## Tests Conducted

Over 50 tests were conducted to ensure the functionality of the UMGC City Application. The tests conducted were grouped into three test categories based on modules: Web Application, Interactive Map, and Database Query. The table below summarizes the tests conducted.

Table 1 - Test Categories

| **Test Case ID** | **Test Description** |
| --- | --- |
| TEST-DBA-001 - TEST-DBA-021 | This group of test cases tested the functionality of different components of the web application, such as confirming “Sign Up” button, “Sign In” button, and navigation bar are working as they were designed to. The tests also ensure that UI screens are displaying correct contents as designed. |
| TEST-MAP-001 - TEST-MAP-005 | This group of test cases focuses on the Interactive Map’s functionality. Link elements were tested to ensure that they were functioning as well as directing users to the proper information per the client requirements. |
| TEST-SQL-001 - TEST-SQL-018 | This group of test cases tested the database to ensure the correct information was pulled with each query. |

# 3. Test Execution Findings

## 3.1 Metrics

Test metrics help to measure the success of the project and implementation of requirements. Metrics help support the claim that requirements were satisfied and help the team understand where improvements can be made. Four statistics are used to calculate test metrics. These four statistics are the number of test cases planned, the number of test cases that were executed, the number of test cases that passed and the number of test cases that failed. The statistics for the UMGC City Web Project are shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Cases Planned** | **Test Cases Executed** | **Test Cases Passed** | **Test Cases Failed** |
| 43 | 40 | 37 | 3 |

Test Execution Percentage (Number of tests executed/Number of tests planned) = 93%

Passed Test Cases Percentage (Number of Passed Tests/Total number of tests executed) = 92.5%

Failed Test Cases Percentage (Number of Failed Tests/Total number of tests executed = 7.5%

Fixed Defects Percentage (Defects Fixed/Defects Reported) = 33%

Defects Deferred Percentage (Defects deferred for future/Total Defects Reported) = 66.6%

**3.2 Failed Test cases**

The following test cases logged failed results during the first round of testing.

**3.2.1 Finding 1**

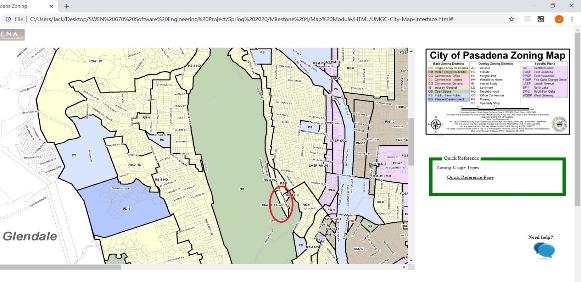
**Test Case:** *TEST-MAP-002*

**Failing Requirement:** REQ-2.2

**Expected Output**: When the user clicks on a zone within the map, a modal opens on top of the map to display specific information to the selected zone.

**Actual Output**: *Non-responsive Map Zone*

**Screenshot of failure:**

****

**Exit Status:** The issue has been corrected. The zone code was added to a modal.

**3.2.2 Finding 2**

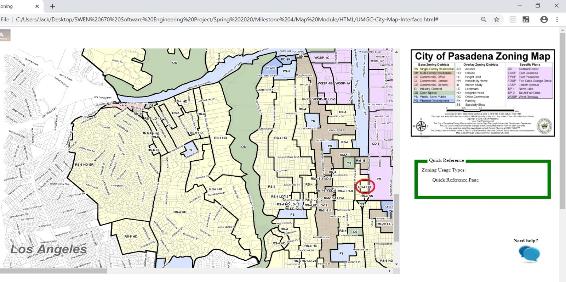
**Test Case:** *TEST-MAP-002*

**Failing Requirement:**REQ-2.2

**Expected Output**: When the user clicks on a zone within the map, a modal opens on top of the map to display specific information to the selected zone.

**Actual Output**: *Non-responsive Map Zone*

**Screenshot of failure:**

******

**Exit Status:** The issue has been corrected. The zone code was added to a modal.

**3.2.3 Finding 3**

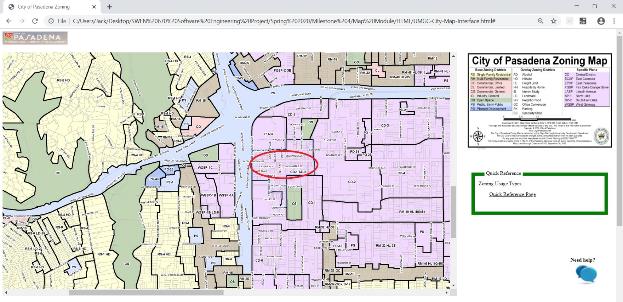
**Test Case:** *TEST-MAP-002*

**Failing Requirement:**REQ-2.2

**Expected Output**: When the user clicks on a zone within the map, a modal opens on top of the map to display specific information to the selected zone.

**Actual Output**: *Non-responsive Map Zone*

**Screenshot of failure:**

**

**Exit Status:** The issue has been corrected. The zone code was added to a modal.

**3.2.4 Finding 4**

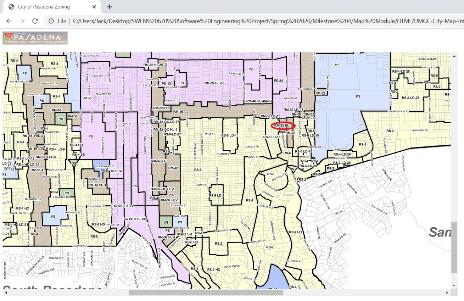
**Test Case:** *TEST-MAP-002*

**Failing Requirement:**REQ-2.2

**Expected Output**: When the user clicks on a zone within the map, a modal opens on top of the map to display specific information to the selected zone.

**Actual Output**: *Non-responsive Map Zone*

**Screenshot of failure:**

**

**Exit Status:** The issue has been corrected. The zone code was added to a modal.

**3.2.5 Finding 5**

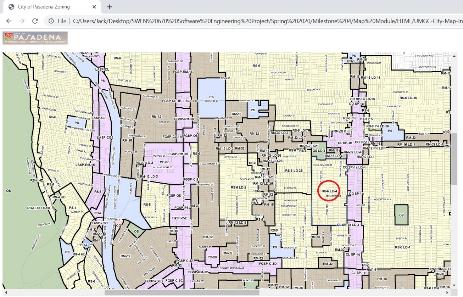
**Test Case:** *TEST-MAP-002*

**Failing Requirement:**REQ-2.2

**Expected Output**: When the user clicks on a zone within the map, a modal opens on top of the map to display specific information to the selected zone.

**Actual Output**: *Non-responsive Map Zone*

**Screenshot of failure:**

**

**Exit Status:** The issue has been corrected. The zone code was added to a modal.

**3.2.6 Finding 6**

**Test Case:** TEST-DBA-003

**Failing Requirement:**REQ-1.3, UI-1.3

**Expected Output**: User receives a sign-up confirmation email

**Actual Output**: Did not receive confirmation email

**Screenshot of failure:** N/A

**Exit Status:** The issue has been corrected. Users should expect to receive a confirmation email.

**3.2.7 Finding 7**

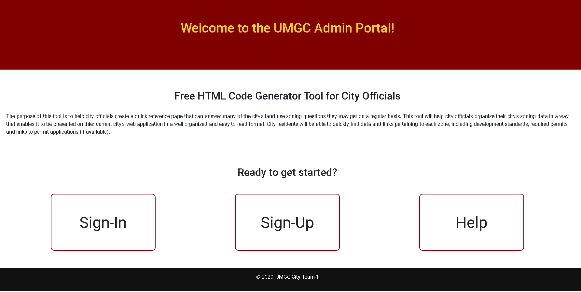
**Test Case:** TEST-DBA-002

**Failing Requirement:**REQ-1.2

**Expected Output**: There is a persistent navigation bar along the top of web pages within UMGC City Application.

**Actual Output**: A navigation bar does not exist.

**Screenshot of failure:**



**Exit Status:** The issue has been deferred.

# 4. Exit Criteria

The exit criteria for the UMGC City Application highly depend on the progression of the software testing phase. The exit criteria for terminating or accomplishing the process of testing are:

* Execution of at least 90% of planned test cases.
* Preferred and enough coverage of the requirements and functionalities under test.
* No high priority or severity or critical bug has been left out.
* All defects in critical or major severity categories has been verified and closed.
* Any open defects in trivial severity shall be addressed in future releases of the UMGC City Application.

# 5. Conclusion

Since most of the exit criteria mentioned in Section 4 have been met and satisfied, the User Interface component of this application for the City of Pasadena is suggested to ‘Go Live’ by the Testing team. This includes any desired database maintenance. Appropriate User/Business acceptance testing should be performed before a ‘Go Live’ deployment.

Additional development contribution would be required to fully implement the Use Case creation and export HTML features.