Software Test Plan

**UMGC CITY APPLICATION**

**Version 2.0**

**04/23/2020**

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

Christy Gilliland

Krystina Poling

Jack Amnuaysirikul

Tarig Abasit

Ziad Elharaoui

Daniel Abresch

Patience Okereke

Melanie Meek

Version History

|  |  |  |
| --- | --- | --- |
| **VERSION NUMBER** | **DATE** | **TITLE OR BRIEF DESCRIPTION** |
| 1.0 | 04/05/2020 | Initial Version |
| 2.0 | 04/23/2020 | First Test Cycle; Section 6 Updated |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

TABLE OF CONTENTS

[1 Introduction 1](#_Toc38837738)

[1.1 Purpose 1](#_Toc38837739)

[1.2 Scope 2](#_Toc38837740)

[1.3 Reference Material 3](#_Toc38837742)

[1.4 Definitions, Acronyms, and Abbreviations 4](#_Toc38837743)

[2 Environmental Requirements 5](#_Toc38837744)

[2.1 Hardware Requirements 5](#_Toc38837745)

[2.2 Software Requirements 5](#_Toc38837746)

[3 Requirements Traceability Matrix 6](#_Toc38837747)

[4 Risks, Assumptions, and Constraints 12](#_Toc38837748)

[4.1 Risks 12](#_Toc38837749)

[4.2 Assumptions 13](#_Toc38837750)

[4.3 Constraints 14](#_Toc38837751)

[5 Software Testing Approach 14](#_Toc38837752)

[5.1 Unit Testing 15](#_Toc38837753)

[5.2 Integration Test Plan 15](#_Toc38837754)

[5.3 Functional Test Plan 18](#_Toc38837758)

[5.4 Test Plan Deliverables 21](#_Toc38837762)

[6 Software Test Cases 21](#_Toc38837763)

[6.1 Test Case 1 23](#_Toc38837764)

[6.2 Test Case 2 24](#_Toc38837765)

[6.3 Test Case 3 25](#_Toc38837766)

[6.4 Test Case 4 26](#_Toc38837767)

[6.5 Test Case 5 27](#_Toc38837768)

[6.6 Test Case 6 28](#_Toc38837769)

[6.7 Test Case 7 29](#_Toc38837770)

[6.8 Test Case 8 30](#_Toc38837771)

[6.9 Test Case 9 31](#_Toc38837772)

[6.10 Test Case 10 32](#_Toc38837773)

[6.11 Test Case 11 33](#_Toc38837774)

[6.12 Test Case 12 34](#_Toc38837775)

[6.13 Test Case 13 35](#_Toc38837776)

[6.14 Test Case 14 36](#_Toc38837777)

[6.15 Test Case 15 37](#_Toc38837778)

[6.16 Test Case 16 38](#_Toc38837779)

[6.17 Test Case 17 39](#_Toc38837780)

[6.18 Test Case 18 40](#_Toc38837781)

[6.19 Test Case 19 41](#_Toc38837782)

[6.20 Test Case 20 42](#_Toc38837783)

[6.21 Test Case 21 43](#_Toc38837784)

[6.22 Test Case 22 44](#_Toc38837785)

[6.23 Test Case 23 45](#_Toc38837786)

[6.24 Test Case 24 46](#_Toc38837787)

[6.25 Test Case 25 47](#_Toc38837788)

[6.26 Test Case 26 48](#_Toc38837789)

[6.27 Test Case 27 49](#_Toc38837790)

[6.28 Test Case 28 50](#_Toc38837791)

[6.29 Test Case 29 51](#_Toc38837792)

[6.30 Test Case 30 52](#_Toc38837793)

[6.31 Test Case 31 53](#_Toc38837794)

[6.32 Test Case 32 54](#_Toc38837795)

[6.33 Test Case 33 55](#_Toc38837796)

[6.34 Test Case 34 56](#_Toc38837797)

[6.35 Test Case 35 57](#_Toc38837798)

[6.36 Test Case 36 58](#_Toc38837799)

[6.37 Test Case 37 59](#_Toc38837800)

[6.38 Test Case 38 60](#_Toc38837801)

[6.39 Test Case 39 61](#_Toc38837802)

[6.40 Test Case 40 62](#_Toc38837803)

[6.41 Test Case 41 63](#_Toc38837804)

[6.42 Test Case 42 64](#_Toc38837805)

[6.43 Test Case 43 65](#_Toc38837806)

TABLE OF FIGURES

[Figure 1 - Risk Management Process 13](#_Toc38667275)

TABLE OF TABLES

[Table 1 - Definitions, Acronyms, and Abbreviations 4](#_Toc38753083)

[Table 2 - Hardware Requirements 5](#_Toc38753084)

[Table 3 - Requirements Matrix 7](#_Toc38753085)

# Introduction

This Software Test Plan (STP) describes the purpose, scope, approach, and resources of all testing activities that will be carried out for the UMGC City Application Version 1.0. The test plan will identify all the items and features to be tested, as well as the types of testing to be performed.

## Purpose

The purpose of this STP is to outline the overall approach and strategy for testing the software requirements as identified in the Software Requirements Specification (SRS) document that was approved by the stakeholders on March 13, 2020. The testing plan and procedures shall be focused on the map/web-based modules of UMGC City Application created by UMGC City Team 1. The STP shall have the following objectives:

* Identify system information and the software/hardware that will be tested
* Describe the testing methods to be utilized
* Detail system requirements for the testing environment
* Identify test cases associated with the requirements in the SRS
* Discuss risks, assumptions, and constraints that may come up while testing

The main goal of testing as specified in this STP is to validate and verify that the map/web-based modules of the UMGC City Application successfully meets the customer’s stated requirements as detailed in the SRS. Various testing methods shall be utilized throughout the testing process to test each component of the UMGC City Application system in accordance with its target production environment specifications. These include unit testing, integration testing, and functional testing. Heavy emphasis shall be placed on functional testing in order to demonstrate the functional capabilities of the map/web-based modules of UMGC City Application, ensuring that excellent working software is delivered at the end. In addition, to help testing stay on track, the traceability of software requirements to test cases shall be provided using the Requirements Traceability Matrix or RTM (see Section 2.3).

## Scope

The scope of this STP 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. In particular, the STP serves as a technical blueprint to be shared with the various stakeholders of the project, particularly the project manager, business analysts, and development, operations, and testing team members for whom will being carrying out the actual tests detailed in the following sections of this document. The goal of this STP is to generate project document artifacts like a testing strategy and test plan, test cases, a traceability matrix, and a software test report to be shared with stakeholders. From the test report, the project team will be able to communicate 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.

This STP is organized into the four main sections as listed below. They constitute the overall testing approach for the map/web-based modules of UMGC City Application:

* Section 1. Introduction – describes and defines the overall purpose, scope, and approach of the Software Testing Plan.
* Section 2. Environment Requirements – specifies the test environment as it pertains to the test plan and procedures. Key hardware and software requirements as well as Requirements Traceability Matrix (RTM) for conducting software testing shall be defined here along with relevant risks, assumptions, and constraints.
* Section 3. Requirements Traceability Matrix - links software test cases to the requirements provided in the Software Requirements Specification (SRS) for easy reference.
* Section 4. Risks, Assumptions, and Constraints – elaborate on the risks, assumptions, and constraints that may appear during software testing.
* Section 5. Software Test Approach – explores various software testing methods, including unit testing, integration testing, and functional testing.

## Section 6. Software Test Cases – identifies all major test cases needed to ensure that all the project requirements as identified in the SRS have been met.

## Reference Material

A complete list of all documents and sources referenced in this Software Test Plan is detailed below.

* Adanza, F. (2017,  January 05). How to manage and identify software testing risks. Retrieved from  [https://www.getzephyr.com/insights/how-identify-and-manage-software-testing-risks](https://www.google.com/url?q=https://www.getzephyr.com/insights/how-identify-and-manage-software-testing-risks&sa=D&source=hangouts&ust=1586017292599000&usg=AFQjCNHTDHlZwN6PvG0yt5a1O1bqR83SWQ)
* Mattp123. (n.d.). Web application requirements. Retrieved from <https://docs.microsoft.com/en-us/dynamics365/customerengagement/on-premises/admin/web-application-requirements>
* IEEE Standard for Software Project Management Plans. IEEE Std 1058.1-1987, 31 Aug. 1988. Accessed via <http://www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf>
* Requirements Traceability Matrix – RTM. Accessed via https://project-management.com/requirements-traceability-matrix-rtm/
* SWEN 670 Capstone Project Guide. Accessed via <https://learn.umuc.edu/d2l/le/content/444089/viewContent/16973490/View>
* Test Plan, 1999, Accessed via <https://sceweb.uhcl.edu/helm/RationalUnifiedProcess/webtmpl/templates/test/rup_tstpln.htm>

## Definitions, Acronyms, and Abbreviations

Table 1 below gives the definitions for the acronyms and abbreviations related to this Software Test Plan of UMGC City Application.

Table 1 - Definitions, Acronyms, and Abbreviations

| **Acronym/**  **Abbreviation** | **Definition** |
| --- | --- |
| ChatBot | A computer program designed to simulate conversation with human users |
| CRUD | Create, Read, Update, and Delete |
| DevOps | Development Operations |
| GHz | Gigahertz |
| GUI | Graphical User Interface |
| IDE | Integrated Development Environment |
| IEEE | Institute of Electrical and Electronics Engineers |
| JDK | Java Development Kit |
| MB | Megabyte |
| OS | Operating System |
| SRS | Software Requirements Specification |
| STP | Software Test Plan |
| SWEN | Software Engineering |
| UMGC | University of Maryland Global Campus |
| RTM | Requirements Traceability Matrix |

# Environmental Requirements

This section specifies the test environment including the physical characteristics, hardware, software, and testing supplies required for conducting software testing.

## Hardware Requirements

Table 2 - Hardware Requirements

| **Component** | **Minimum** | **Recommended** |
| --- | --- | --- |
| Interface | Chrome Browser Version 80.0.3987.149 (Official Build) (64-bit) or later | Chrome Browser Version 80.0.3987.149 (Official Build) (64-bit) or later |
| Network | Ethernet connection (LAN) OR a wireless adapter (Wi-Fi) | Bandwidth greater than 50 KBps (400 kbps) Latency under 150 ms |

## Software Requirements

**Recommended Operating Systems**

* Windows7 or newer

**Recommended Browsers**

The web application can run on any of the following web browsers:

* Google Chrome (preferred) (latest publicly-released version)
* Internet Explorer (latest publicly-released version)
* Mozilla Firefox (latest publicly-released version)

Testing will be completed on the preferred browser, Google Chrome.

# Requirements Traceability Matrix

The Requirements Traceability Matrix (RTM) provided here links the software requirements described in the SRS of UMGC City Team 1 project to appropriate test cases listed in Section 4 of this STP. The goal is to help verify that all the customer’s requirements have been met. For easy reference, the RTM displays all project requirements and their associated test cases for the map/web-based modules of UMGC City Application seen in Table 2 below.



Table 3 - Requirements Matrix

| **ID** | **Requirement ID** | **Requirement Description** | **Test Case ID** |
| --- | --- | --- | --- |
| 1 | OE-1.1 | For increased accessibility, the application shall operate on a web server compatible with a common, modern browser. | TEST-DBA-001 |
| 2 | OE-2.1 | The application shall authenticate users using a secure validation process and Hypertext Transfer Protocol Secure (HTTPS) requests. | TEST-DBA-001, TEST-DBA-007 |
| 3 | REQ-1.2 | There shall be a consistent main navigation bar along the top of web pages to allow the user to quickly navigate within the application. | TEST-DBA-002, TEST-DBA-021 |
| 4 | REQ-1.3 | The UMGC City Admin Portal system shall provide a new user with the ability to sign up for an account by clicking on the “Sign-Up” button. | TEST-DBA-003 |
| 5 | REQ-1.4 | The UMGC City Admin Portal system shall allow a user to sign into his/her existing account by clicking on the “Sign-In” button. | TEST-DBA-004, TEST-DBA-005 |
| 6 | REQ-1.5 | The UMGC City Admin Portal system shall provide a user with a web page dedicated to providing help and support by clicking on the “Help” button. | TEST-DBA-006 |
| 7 | REQ-1.6 | The user shall be brought to the landing page once successfully logged into the system.  The system features of creating a new use case, editing existing use cases, and uploading a use case are now available. | TEST-DBA-007 |
| 8 | REQ-1.7 | The UMGC City Admin Portal system works by accepting data inputted by the user.  The user creates a use case for his/her city. The application shall map the use case to the appropriate city data. | TEST-DBA-017, TEST-DBA-020 |
| 9 | REQ-1.8 | The UMGC City Admin Portal system shall allow a user to manually add a new use case, edit existing use cases, and delete a use case from a list within the account.  In addition, the system shall allow the user to upload use cases from an external file using the “Upload Use Case” button. | TEST-DBA-008, TEST-DBA-009, TEST-DBA-010, TEST-DBA-011, TEST-DBA-016, TEST-DBA-018 |
| 10 | REQ-1.9 | The UMGC City Admin Portal system shall provide a user with a certain amount of memory storage space that is linked to his/her account.  The user can save the use cases and come back to them later. | TEST-DBA-020 |
| 11 | REQ-1.10 | The UMGC City Admin Portal system shall process the project data inputted by the user and generate proper output of static HTML page with appropriate DDL scripts using the “Generate Output” button. | TEST-DBA-012, TEST-DBA-013, TEST-DBA-014 |
| 12 | REQ-2.1 | When the user drags either scrollbar, the map image is moved within the containing window.  Scrolling is set to default Chrome settings. | TEST-MAP-001 |
| 13 | REQ-2.2 | 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. The modal is designed using in-line JavaScript and custom CSS. | TEST-MAP-002 |
| 14 | REQ-2.3 | When the user clicks on the “Zone Information” link, the browser navigates to an external page on the Municode website that contains information that is relevant to the selected zone. | TEST-MAP-003 |
| 15 | REQ-2.4 | When the user clicks on any “Application” link, the browser navigates to an external page on the Municode website that contains the applications that are relevant to the selected zone. | TEST-MAP-004 |
| 16 | REQ-2.5 | When the user clicks on the “Quick Reference Page” link, the browser navigates to an internal page that contains information designed by the city officials.  The information on this page is retrieved from the database. | TEST-MAP-005 |
| 17 | REQ-2.6 | When the user clicks on the “Need Help?” icon, the ChatBot application opens on top of the map interface.  The ChatBot application is designed by UMGC City Team 2. | TEST-MAP- 006 |
| 18 | REQ-3.1 | The system shall direct the user to the “Quick Reference Page” | TEST-MAP-005 |
| 19 | REQ-4.1 | The system shall direct the user to the “Home Occupation” regulations page after the user clicks the regulation link. | TEST-SQL-001 |
| 20 | REQ-4.2 | The system shall direct the user to the “Home Occupation” application page after the user clicks the application link. | TEST-SQL-002 |
| 21 | REQ-5.1 | The system shall direct the user to the “Accessory Dwelling Unit” regulations page after the user clicks the regulations link. | TEST-SQL-003 |
| 22 | REQ-5.2 | The system shall direct the user to the “Accessory Dwelling Unit” application page after the user clicks the application link. | TEST-SQL-004 |
| 23 | REQ-6.1 | The system shall direct the user to the “Short Term Rental” regulations page after the user clicks the regulations link. | TEST-SQL-005 |
| 24 | REQ-6.2 | The system shall direct the user to the “Short Term Rental” application page after the user clicks the application link. | TEST-SQL-006 |
| 25 | REQ-7.1 | The system shall direct the user to the “Accessory Structure” regulations page after the user clicks the regulations link. | TEST-SQL-007 |
| 26 | REQ-7.2 | The system shall direct the user to the “Accessory Structure” application page after the user clicks the application link. | TEST-SQL-008 |
| 27 | REQ-8.1 | The system shall direct the user to the “Exterior Modifications in Landmark District” regulations page after the user clicks the regulations link. | TEST-SQL-009 |
| 28 | REQ-8.2 | The system shall direct the user to the “Exterior Modifications in Landmark District” application page after the user clicks the application link. | TEST-SQL-010 |
| 29 | REQ-9.1 | The system shall direct the user to the “Fence” regulations page after the user clicks the regulations link. | TEST-SQL-011 |
| 30 | REQ-9.2 | The system shall direct the user to the “Fence” application page after the user clicks the application link. | TEST-SQL-012 |
| 31 | REQ-10.1 | The system shall direct the user to the “Tree Removal (Residential)” regulations page after the user clicks the regulations link. | TEST-SQL-013 |
| 32 | REQ-10.2 | The system shall direct the user to the “Tree Removal (Residential)” application page after the user clicks the application link. | TEST-SQL-014 |
| 33 | REQ-11.1 | The system shall direct the user to the “Day-Care” regulations page after the user clicks the regulations link. | TEST-SQL-015 |
| 34 | REQ-11.2 | The system shall direct the user to the “Day-Care” application page after the user clicks the application link. | TEST-SQL-016 |
| 35 | REQ-12.1 | The system shall direct the user to the “Sober Living Facility” regulations page after the user clicks the regulations link. | TEST-SQL-017 |
| 36 | REQ-12.2 | The system shall direct the user to the “Sober Living Facility” application page after the user clicks the application link. | TEST-SQL-018 |
| 37 | UI-1.1 | The UMGC City Admin Portal UI shall display a Home Page Screen. | TEST-DBA-001 |
| 38 | UI-1.2 | The UMGC City Admin Portal UI shall display a Sign-in Screen. | TEST-DBA-004, TEST-DBA-017 |
| 39 | UI-1.3 | The UMGC City Admin Portal UI shall display a Sign-Up for new users. | TEST-DBA-003, TEST-DBA-017 |
| 40 | UI-1.4 | The UMGC City Admin Portal UI shall display a Help Screen. | TEST-DBA-006, TEST-DBA-017 |
| 41 | UI-1.5 | The UMGC City Admin Portal UI shall display a Landing Page Screen to the user when the user inputs valid login credentials into Sign-in Screen. | TEST-DBA-007 |
| 42 | UI-1.6 | The UMGC City Admin Portal UI shall display the New Use Case Screen to the user when the user selects the “New Use Case” option. | TEST-DBA-008, TEST-DBA-020 |
| 43 | UI-1.7 | The UMGC City Admin Portal UI shall display the File Upload Screen to the user when the user selects the “File Upload” option. | TEST-DBA-009, TEST-DBA-018, TEST-DBA-019 |
| 44 | UI-1.8 | The UMGC City Admin Portal UI shall display the Manage Existing Use Case Screen to the user when the user selects the “Manage Existing Use Case” option. | TEST-DBA-010, TEST-DBA-016, TEST-DBA-020 |
| 45 | UI-1.9 | The UMGC City Admin Portal UI shall display the Delete Use Case Screen to the user when the user selects a use case and selects the “Delete Use Case” option. | TEST-DBA-011 |
| 46 | UI-1.10 | The UMGC City Admin Portal UI shall display the Export Project Screen to the user when the “Export Project” option is selected on the Manage Existing Use Case Screen. | TEST-DBA-012 |
| 47 | UI-1.11 | The UMGC City Admin Portal UI shall display the Generate Output Screen to the user when the “Generate output” option is selected on the Export Project Screen. | TEST-DBA-013 |
| 48 | UI-1.12 | The UMGC City Admin Portal UI shall display the Display Preview Screen to the user when the “Display Preview” option is selected on the Export Project Screen. | TEST-DBA-014 |
| 49 | UI-1.13 | The UMGC City Admin Portal UI shall display the Download Project Screen to the user when the “Download” option is selected on the Export Project Screen. | TEST-DBA-015 |

# Risks, Assumptions, and Constraints

The following sections explain the risks, assumptions, and constraints that have been identified for the testing phase of this project development. These factors help define realistic expectations and assist in forming a risk mitigation plan.

## Risks

A risk is an expectation of loss resulting from potential problems that may or may not occur in the future. To protect business interests and the quality of software applications, the testers from UMGC City Team 1 shall quickly and accurately identify and manage software testing risks.

Some of the identified factors that may affect testing effort and increase the risk associated with the success of the testing includes:

* Delay in the development of the application can cause testing to be delayed beyond the end of the course.
* Cloud computing systems are internet-based and therefore service outages, also known as downtime, can occur for any reason and cause testing to be stalled until the system can be brought back up.
* Any changes introduced to the project scope at this point may require rewriting at least some of the test plan, while also causing some previously defined test cases to become obsolete.
* Any updates applied to Google Chrome, Postgres, or Heroku may change the behavior of previously written code. The project is not built to guarantee 100% backward capability and this risk will have to be addressed as future development continues.

Software risk planning is crucial to the success of QA testing and the resulting deployment. To combat risks, there is a need to set up a testing plan that highlights workflow procedures that contribute towards risk mitigation. Below is a risk management process that will be used through the testing phase of the software.



Figure 1 - Risk Management Process

(Retrieved from https://www.getzephyr.com/insights/how-identify-and-manage-software-testing-risks)

Success in mitigating software risks stems directly from upfront assessment of project challenges which includes:

* Defining measures effective in lowering or preventing risks.
* Defining measures to reduce the impact of unalleviated risks.
* Through automated testing, continuously monitoring software and system performance to quickly identify risks.

## Assumptions

* It is assumed that all current project requirements are accurate, and no additional changes will be made between the release of this test plan and the actual test cycle, which will occur between April 6th, 2020 and April 24th, 2020.
* It is further assumed that a live presentation of any test case(s) will not be required. The team will provide the stakeholders with an updated Software Test Plan as well as a Test Report at the completion of this course.

## Constraints

The testing constraints are listed below:

|  |  |
| --- | --- |
| Time | With unforeseen requirement changes occurring during Sprint 2, the start of Sprint 3 was delayed. The schedule change caused a loss of approximately a week of development and testing time. |
| Resources (lack of, insufficient, not trained). | Team 1 does not have any professional software testers, and more than half of the team has no professional developing experience. |
| Test tools (lack of, not suited, difficult to maintain). | The test tool suite was chosen for the suitability it provides for this project’s scope. The selection was based on common, updated methods of testing and schedule constraints were not given a priority in the test design. |

# Software Testing Approach

Software testing for the UMGC City Application incorporates several types of tests to verify that the software works correctly and satisfies requirements. The types of tests included in this plan are unit, integration, and functional testing.

Testing will be performed at several points in the UMGC City Application life cycle as the product is implemented to protect the overall quality of the application. In addition, the test artifacts will never be static and shall be modified and shared with all stakeholders from time to time for the purpose of keeping all stakeholders abreast to the latest testing activities and bug tracking updates. Document Overview

## 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). Unit tests will be designed by the developers and will use white box testing techniques.

5.1.1 Unit Test Plan

Unit testing will be conducted manually by developers prior to commits being pushed to the git repository on GitHub. Unit testing will be executed before any functional and integration tests.

In this project, JUnit will be the testing framework used for unit testing. It is a Java-based testing framework that supports writing and executing unit tests for Java applications. UMGC City Team 1 will be using JUnit version 5.5.1 released on July of 2019.

## Integration Test Plan

The purpose of integration testing is to ensure that software components work properly when used together. When making changes to the project source code, developers will execute integration tests after ensuring that all unit tests pass. Developers must ensure that integration tests pass prior to pushing commits to the GitHub master branch to prevent breaking the build. Integration tests will be designed by the testers.

5.2.1 Purpose

This integration test plan supports the following objectives:

* + Describe the scope of the testing that will occur during integration testing
  + Describe the testing strategies to be employed
  + Present the activities included in the system integration test.

5.2.2 Scope of Testing

The system integration test of the UMGC City Application will include all internal components designed to accomplish business and design requirements. The UMGC City Application is a standalone application and therefore, no additional external integration tests are required.

5.2.3 Testing Strategy

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.

Unit Testing will be completed before integration tests occur. Functional testing will supersede integration testing.



# Integration Test Plan Approach

* Understand architecture of the application.
* Identify the modules.
* Understand what each module does.
* Understand how the data is transferred from one module to another.
* Understand how the data is entered and received into the system (entry point and exit point of the application).
* Segregate the application to suit testing needs.
* Identify and create the test conditions.

# Integration Test Suite Activities

Integration testing will verify the following two protocols in addition to the standard component integration tests:

* Security testing, by testing each end-user’s security access levels. Once all components are combined, no component should be able to access functions the user would otherwise be prohibited from accessing.
* Regression testing to ensure that a change to the system does not introduce new defects. Integration tests should verify the system operates with no errors after each code update and/or addition.

# Roles

This table shows the team members completing integration testing on the application.

| **Name** | **Primary Role** | **Secondary Role(s)** |
| --- | --- | --- |
| Jack Amnuaysirikul | Tester | Requirements Analyst, DBA |
| Melanie Meek | Requirements Analyst | Tester |
| Patience Okereke | Requirements Analyst | Tester |
| Krystina Poling | Lead Developer | Tester/DBA |
| Daniel Abresch | Developer | Tester |
| Ziad Elharaoui | Developer | Requirements Analyst |
| Tarig Abasit | Developer | Requirements Analyst |

## Functional Test Plan

Functionality testing focuses on direct features of the application. The purpose of functionality testing is to ensure that each feature of the application does what it is intended to do. For 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.

Selenium, a free functional testing tool, will be used to test the features of the UMGC city application. The Selenium IDE is available as an add-on for Google Chrome to write and execute tests. The IDE offers recording and playback features that learn the tester’s steps and automate them for on-demand test execution.

5.3.1 Purpose

This functional test plan supports the following objectives:

* + Describe the scope of the testing that will occur during functional testing
  + Describe the testing strategies to be employed
  + Present the activities included in the system functional test.

5.3.2 Scope of Testing

The functional test suite will be conducted on the application following the successful integration of the components identified in the Integration Plan.

It is assumed that unit testing already provided thorough white box testing, extensive coverage of source code, and testing of all module interfaces.

It is critical that all system and subsystem interfaces be tested as well as system performance at this early stage. Testing of the ChatBot functionality and features will not be conducted as it is not a part of UMGC City Team 1’s UMGC City Application.

The interfaces between the following subsystems will be tested:

* 1. City Official (Verify that the user can create an output page and save data to the database.)
  2. City Resident (Verify that the user can access the output page with the appropriate data as entered by the city official user.)

5.3.3 Testing Strategy

The Test Strategy presents the recommended approach to the testing of the application. The previous section described what will be tested; this describes how it will be tested.

The primary considerations for the test strategy are the techniques to be used and the criteria for knowing when the testing is completed. Testing of the application should focus on any target requirements that can be traced directly to use cases. The goals of these tests are to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules.



# 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. |

# 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. |

# Roles

This table shows the team members completing functional testing on the application.

| **Name** | **Primary Role** | **Secondary Role(s)** |
| --- | --- | --- |
| Jack Amnuaysirikul | Tester | Requirements Analyst, DBA |
| Melanie Meek | Requirements Analyst | Tester |
| Patience Okereke | Requirements Analyst | Tester |
| Krystina Poling | Lead Developer | Tester/DBA |
| Daniel Abresch | Developer | Tester |
| Ziad Elharaoui | Developer | Requirements Analyst |
| Tarig Abasit | Developer | Requirements Analyst |
| Christy Gilliland | Project Manager | Developer |

## Test Plan Deliverables

This test plan will be used to track the status of each test case. The test results will be summarized as untested, passed, conditional pass, or failed. Each test case in Section 6 - Software Test Cases will be updated to include:

* + Test status
  + Tested By
  + Test Notes

At the conclusion of milestone 4 and the test plan cycle, UMGC City Team 1 will provide a test report to accompany this completed test plan. While the test plan will include all test results for the cases found in Section 6, the test report will include detailed information for all tests that failed, including the number of failures and how the failure was corrected.

# Software Test Cases

This section documents each test case to be used in software testing. In all test cases, it is assumed that all environmental requirements identified in Section 2 have been met. Therefore, the prerequisites section of the test cases outlined below shall only identify previous test cases that must be run and passed before the current test case is executed.

Initial test cycle update: The test cases are color coded. If a test ever failed, even if it subsequently passed after a fix, the table header shows as red. If a test case was not completed, the table header shows as yellow. If the test case passes, the table header is displayed as green.

## Test Case 1

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-001 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | The home page of UMGC City Application is accessed via the web URL: https://appdev-umgc-cityweb.herokuapp.com | | | | **Test Priority** | | High | | |
| **Requirement ID** | | OE-1.1, OE-2.1, REQ-1.1, UI-1.1 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | N/A | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Launch web application. | | https://appdev-umgc-cityweb.herokuapp.com | Home page of UMGC City Application is loaded. | Application Admin Interface is loaded | | Chrome | | Pass | Pending |

## Test Case 2

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-002 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a persistent navigation bar along the top of web pages within UMGC City Application. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.2 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | N/A | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate the web pages using the main navigation bar. | | Clicks on navigation buttons in the main navigation bar. | System brings the user to a new web page based on the selected navigation button. The main navigation bar stays persistent. | There is no navigation bar. | | Chrome | | Fail | Pending |

## Test Case 3

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-003 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a “Sign-Up” button that brings the user to a web page which allows new users to sign up for an account. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.3, UI-1.3 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | N/A | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the sign-up page. | | Click on the “Sign-Up” button. | System presents the user with the sign-up screen after the “Sign-Up” button is selected. | Pending | | Chrome | | Pass |  |
| 2 | Enter user information | | Enter the username, first name, last name, email, and password | System displays username, first name, last name, email, and password. | Pending | | Chrome | | Pass |  |
| 3 | Verify the email confirmation | | N/A | User receives a sign-up confirmation email | Pending | | Chrome | | Fail | Did not receive confirmation email |

## Test Case 4

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-004 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a “Sign-In” button that brings the user to a web page which prompts the user to sign into the account. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.4, UI-1.2 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | N/A | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the sign-in page. | | Click on the “Sign-In” button. | System presents the user with the sign-in screen after the “Sign-In” button is selected. | Pending | | Chrome | | Pass | Presented with the Sign-In button and was successfully able to select it |

## Test Case 5

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-005 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | The “Sign-In” button is toggled to the “Sign-Out” button to let the user safely and easily log out of the system. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.4 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-DBA-007 (User must be logged in) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Successfully sign into the user account. | | Valid credentials (email address and password). | System toggles the “Sign-In” button to the “Sign-Out” button. | Pending | | Chrome | | Pass | Successfully entered in credentials |
| 2 | Sign out of the system. | | Click on the “Sign-Out” button. | System safely logs the user out after the “Sign-Out” button is selected. | Pending | | Chrome | | Pass | I was able to log out by selecting “Sign-Out” button |

## Test Case 6

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-006 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a “Help” button that brings the user to a web page which provides the user with help options. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.5, UI-1.4 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | N/A | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the help page. | | Click on the “Help” button. | System presents the user with the help screen after the “Help” button is selected. | Pending | | Chrome | | Pass | Presented with help screen after selecting “Help” button |

## Test Case 7

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-007 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a landing page after the user successfully logged into the system. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | OE-2.1, REQ-1.6, UI-1.5 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | N/A | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Sign into the user account. | | Valid credentials (email address and password). | System presents the user with the landing page after the log-in credentials are successfully validated. | Pending | | Chrome | | Pass | Successfully entered in credentials and was taken to landing page |

## Test Case 8

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-008 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a “New Use Case” button that brings the user to a web page which allows the user to manually add a new use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.8, UI-1.6 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-DBA-007 (User must be logged in) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the New Use Case screen. | | Click on the “New Use Case” button. | System presents the user with the New Use Case screen after the “New Use Case” button is selected. | Pending | | Chrome | | Pass | Taken to New Use Case screen after selecting button |

## Test Case 9

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-009 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is an “Upload Use Case” button that brings the user to a web page which allows the user to upload use cases from an external file. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.8, UI-1.7 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-DBA-007 (User must be logged in) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the Upload Use Case screen. | | Click on the “Upload Use Case” button. | System presents the user with the Upload Use Case screen after the “Upload Use Case” button is selected. | Pending | | Chrome | | Pending | Not tested because functionality was not implemented during this iteration. |

## Test Case 10

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-010 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is an “Existing Use Case” button that brings the user to a web page which allows the user to manage existing use cases. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.8, UI-1.8 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-DBA-007 (User must be logged in) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the Existing Use Case screen. | | Click on the “Existing Use Case” button. | System presents the user with the Existing Use Case screen after the “Existing Use Case” button is selected. | Pending | | Chrome | | Pass | Taken to Existing Use Case screen after selecting appropriate button |

## Test Case 11

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-011 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a “Delete” button that allows the user to delete a selected use case from a list after a confirmation. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.8, UI-1.9 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-DBA-007 (User must be logged in)  TEST-DBA-010 (User is on Existing Use Case screen) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Select a use case from the list. | | Click on a use case. | System highlights the user selection of the use case. | System highlights the user selection of the use case. | | Chrome | | Pass | Pending |
| 2 | Delete the use case. | | Click on the “Delete” button. | System shall prompt the user to confirm the deletion of the selected use case from the list. | System prompts the user to confirm the deletion of the selected use case from the list. | | Chrome | | Pass | Pending |

## Test Case 12

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-012 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a “Project Export” button that brings the user to a web page which allows the user to export the current project. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.10, UI-1.10 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-DBA-007 (User must be logged in)  TEST-DBA-010 (User is on Existing Use Case screen) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to Project Export screen. | | Click on the “Project Export” button. | System presents the user with the Project Export screen after the “Project Export” button is selected. | System presents the user with the Project Export screen after the “Project Export” button is selected. | | Chrome | | Pass | Pending |

## Test Case 13

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-013 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a “Generate Output” button that the user can select to generate system output to export the project file. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.10, UI-1.11 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-DBA-007 (User must be logged in)  TEST-DBA-012 (User is on Project Export screen) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Generate new system output. | | Click on the “Generate Output” button. | System shall generate output of exporting the project file after the “Generate Output” button is selected. |  | | Chrome | | Pending | Not tested because functionality was not implemented during this iteration. |
| 2 | Observe the output progress. | | None | System shall display real-time progress of generating output. Once 100%, the system shall inform the user that the output is completed. |  | | Chrome | | Pending | Not tested because functionality was not implemented during this iteration. |

## Test Case 14

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-014 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a “Preview” button that the user can select to generate system output to export the project file. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.10, UI-1.12 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-DBA-007 (User must be logged in)  TEST-DBA-012 (User is on Project Export screen)  TEST-DBA-013 (Output is generated) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Preview the system output. | | Click on the “Preview” button. | System shall open up a new window that contains a display preview of project output after the “Preview” button is selected. | A new window displays a preview of project output after the “Preview” button is selected. | | Chrome | | Pass | Pending |

## Test Case 15

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-015 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a “Edit” button that allows the user to edit a selected use case from a list within the account. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.8, UI-1.8 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-DBA-007 (User must be logged in)  TEST-DBA-010 (User is on Existing Use Case screen) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Select a use case from the list. | | Click on a use case. | System highlights the user selection of the use case. | System highlights the user selection of the use case. | | Chrome | | Pass | Pending |
| 2 | Edit the use case. | | Click on the “Edit” button. | System presents the user with the Edit Use Case screen after the “Edit” button is selected. | System presents the user with the Edit Use Case screen after the “Edit” button is selected. | | Chrome | | Pass | Pending |

## Test Case 16

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-017 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a “Submit” button that allows the user data input to be submitted to the system for processing. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.7, UI-1.2, UI-1.3, UI-1.4 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | N/A | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Submit data to the system. | | Click on the “Submit” button. | System accepts the user data input, processes it, and returns appropriate response. | System accepts the user data input, and displays confirmation to the user. | | Chrome | | Pass | Pending |

## Test Case 17

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-018 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a “Browse” button that the user can select to browse the file directory for an external file to be uploaded. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.8, UI-1.7 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-DBA-007 (User must be logged in)  TEST-DBA-009 (User is on Upload Use Case screen) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Browse for an external file to be uploaded. | | Click on the “Browse” button. | System shall open up a file directory window after the “Browse” button is selected. | System opens up a separate file directory window when the “Browse” button is clicked. | | Chrome | | Pass | [Jack 4/20/20]: File directory appears when the “Browse” button is clicked. |
| 2 | Confirm the file location. | | Specify the file directory. | An external file is located within the file directory to be uploaded. | System allows the user to choose an external file from the computer directory for uploading. | | Chrome | | Pass | [Jack 4/20/20]: An external file can be chosen to be uploaded. |

## Test Case 18

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-020 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a “Save” button that the user can select to save the current use case data to the system using relational database. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | OE-3.1, REQ-1.7, REQ-1.9, UI-1.6, UI-1.8 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-DBA-007 (User must be logged in)  [TEST-DBA-008 (User is on New Use Case screen) OR  TEST-DBA-016 (User is on Edit Use Case screen)] | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Save the current progress on a use case. | | Click on the “Save” button. | System shall save the current use case data to the user account after the “Save” button is selected. | Use case is saved to the account database. | | Chrome | | Pass | Pending |

## Test Case 19

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-DBA-021 | | | | **Test Module** | | Web Database Application | | |
| **Test Case Description** | | There is a “Back” button that the user can select to go back to the previous screen the user is on. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-1.2 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-DBA-007 (User must be logged in) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate back to the previous screen | | Click on the “Back” button. | Once the “Back” button is selected, the system shall bring the user back to the previous screen. | The system returns to the previous screen. | | Chrome | | Pass | Pending |

## Test Case 20

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-MAP-001 | | | | **Test Module** | | Interactive Map | | |
| **Test Case Description** | | There are window scrollbars that the user can use to scroll and adjust the location of the map image. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-2.1 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | N/A | | | | **Assumptions/**  **Constraints** | | Working Chrome browser | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate the map image using the scrollbars. | | Scrollbar adjustments using mouse or keyboard actions. | When the user drags the scrollbars up, down, left, or right, the map image is moved within the containing window accordingly. | The map image moves according to the user’s input of the scrollbars. | | Chrome | | Pass | [Jack 4/20/20]: Successful map navigation using the scrollbars. |

## Test Case 21

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-MAP-002 | | | | **Test Module** | | Interactive Map | | |
| **Test Case Description** | | There are zones within the map image that the user can click on to open up an informational modal. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-2.2 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | N/A | | | | **Assumptions/**  **Constraints** | | Internet access, responsive map image, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Select a zone within the map image. | | Mouse click on an area within the map. | Once the user clicks on a zone, a modal opens on top of the map to display specific information to the selected zone. | Some of the zones within the map are not responsive to the user’s mouse click | | Chrome | | Fail | [Jack 4/20/20]:  Sent feedback to the developers. Will retest. |
| Re-test |  | |  |  | System opens up the informational model for every zone that is clicked on the map. | | Chrome | | Pass | [Jack 4/22/20]:  All zones are interactive. |

## Test Case 22

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-MAP-003 | | | | **Test Module** | | Interactive Map/Database Query | | |
| **Test Case Description** | | There is a “Zone Information” link that the user can click on to navigate to an external page that contains information about the current selected zone. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-2.3 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-002 (User is on the informational modal) | | | | **Assumptions/**  **Constraints** | | Internet access, responsive map image, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Obtain more information about the selected zone. | | Click on the “Zone Information” link. | Once the user clicks on the “Zone Information” link, the browser navigates to an external page that contains information relevant to the selected zone. | The browser navigates to an external page that contains information relevant to the selected zone. | | Chrome | | Pass | Pending |

## Test Case 23

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-MAP-004 | | | | **Test Module** | | Interactive Map/Database Query | | |
| **Test Case Description** | | There is an “Application” link that the user can click on to navigate to an external page that contains applications relevant to the selected zone. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-2.4 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-002 (User is on the informational modal) | | | | **Assumptions/**  **Constraints** | | Internet access, responsive map image, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the applications relevant to the selected zone. | | Click on the “Application” link. | Once the user clicks on the “Application” link, the browser navigates to an external page that contains applications relevant to the selected zone. | The browser navigates to an external page that contains applications relevant to the selected zone. | | Chrome | | Pass | Pending |

## Test Case 24

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-MAP-005 | | | | **Test Module** | | Interactive Map/Database Query | | |
| **Test Case Description** | | There is a “Quick Reference Page” link that the user can click on to navigate to an internal page that contains information designed by the city officials. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-2.5, REQ-3.1 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-002 (User is on the informational modal) | | | | **Assumptions/**  **Constraints** | | Internet access, responsive map image, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page.” | | Click on the “Quick Reference Page” link. | Once the user clicks on the “Quick Reference Page” link, the browser navigates to an internal page that contains information designed by the city officials. | The browser navigates to an internal page that contains information designed by the city officials. | | Chrome | | Pass | Pending |

## Test Case 25

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-MAP- 006 | | | | **Test Module** | | Interactive Map/ChatBot | | |
| **Test Case Description** | | There is a “Need Help?” icon that the user can select to open up the ChatBot application. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-2.6 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | N/A | | | | **Assumptions/**  **Constraints** | | Internet access, responsive map image, integrated ChatBot | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Launch ChatBot. | | Click on the “Need Help?” icon. | When the user clicks on the “Need Help?” icon, the ChatBot application opens on top of the map interface. | The ChatBot application opens on top of the map interface. | | Chrome | | Pass | The feature was not tested because DevOps and team 2 decided the applications would exist as separate components. |

## Test Case 26

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-001 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is a “Home Occupation” link that the user can click on to navigate to an external page containing relevant regulations for home occupation use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-4.1 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Home Occupation” regulation page. | | Click on the “Home Occupation” regulation link. | The system shall direct the user to the “Home Occupation” regulation page after the user clicks the regulation link. | The system directs the user to the to the Municode page for the “Home Occupation” regulations after the user clicks the regulation link. | | Chrome | | Pass | Pending |

## Test Case 27

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-002 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is a “Home Occupation” link that the user can click on to navigate to an external page containing relevant applications for home occupation use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-4.2 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Home Occupation” application page. | | Click on the “Home Occupation” application link. | The system shall direct the user to the “Home Occupation” application page after the user clicks the application link. | The system directs the user to the to the Municode page for the “Home Occupation” application after the user clicks the application link. | | Chrome | | Pass | Pending |

## Test Case 28

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-003 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is an “Accessory Dwelling Unit” link that the user can click on to navigate to an external page containing relevant regulations for accessory dwelling unit use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-5.1 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Accessory Dwelling Unit” regulation page. | | Click on the “Accessory Dwelling” regulation link. | The system shall direct the user to the “Accessory Dwelling Unit” regulation page after the user clicks the regulation link. | The system directs the user to the to the Municode page for “Accessory Dwelling Unit” regulations after the user clicks the regulation link. | | Chrome | | Pass | Pending |

## Test Case 29

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-004 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is an “Accessory Dwelling Unit” link that the user can click on to navigate to an external page containing relevant applications for accessory dwelling unit use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-5.2 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Accessory Dwelling Unit” application link. | The system shall direct the user to the “Accessory Dwelling Unit” application page after the user clicks the application link. | The system directs the user to the Municode page for Code Compliance Permit Application. | | Chrome | | Pass | The link directs the user to the exact location of the permit description. |

## Test Case 30

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-005 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is a “Short Term Rental” link that the user can click on to navigate to an external page containing relevant regulations for short term rental use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-6.1 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Short Term Rental” regulation link. | The system shall direct the user to the “Short Term Rental” regulation page after the user clicks the regulation link. | The system directs the user to the Municode page for Short Term Rental regulations. | | Chrome | | Pass | The link directs the user to the exact location of the Short term rental regulations. |

## Test Case 31

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-006 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is a “Short Term Rental” link that the user can click on to navigate to an external page containing relevant applications for short term rental use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-6.2 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Short Term Rental” application link. | The system shall direct the user to the “Short Term Rental” application page after the user clicks the application link. | The system directs the user to the Municode page for Code Compliance Permit Application. | | Chrome | | Pass | The link directs the user to the exact location of the permit description. |

## Test Case 32

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-007 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is an “Accessory Structure” link that the user can click on to navigate to an external page containing relevant regulations for accessory structure use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-7.1 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Accessory Structure” regulation link. | The system shall direct the user to the “Accessory Structure” regulation page after the user clicks the regulation link. | The system directs the user to the Municode page for Accessory Structure Regulations. | | Chrome | | Pass | The link directs the user to the exact location the Acc. Structure Regulations. |

## Test Case 33

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-008 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is an “Accessory Structure” link that the user can click on to navigate to an external page containing relevant applications for accessory structure use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-7.2 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Accessory Structure” application link. | The system shall direct the user to the “Accessory Structure” application page after the user clicks the application link. | The system directs the user to the Municode page for Code Compliance Permit Application. | | Chrome | | Pass | The link directs the user to the exact location of the permit description. |

## Test Case 34

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-009 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is an “Exterior Modifications in Landmark District” link that the user can click on to navigate to an external page containing relevant regulations for exterior modifications in landmark district use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-8.1 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Exterior Modifications in Landmark District” regulation link. | The system shall direct the user to the “Exterior Modifications in Landmark District” regulation page after the user clicks the regulation link. | The system directs the user to the Municode page for “Exterior Modifications in Landmark District” regulations after the user clicks the regulation link. | | Chrome | | Pass | Pending |

## Test Case 35

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-010 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is an “Exterior Modifications in Landmark District” link that the user can click on to navigate to an external page containing relevant applications for exterior modifications in landmark district use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-8.2 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Exterior Modifications in Landmark District” application link. | The system shall direct the user to the “Exterior Modifications in Landmark District” application page after the user clicks the application link. | The system directs the user to the Municode page for “Exterior Modifications in Landmark District” applications after the user clicks the application link. | | Chrome | | Pass | The link directs the user to the exact location of the permit description. |

## Test Case 36

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-011 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is a “Fence” link that the user can click on to navigate to an external page containing relevant regulations for fence use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-9.1 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Fence” regulation link. | The system shall direct the user to the “Fence” regulation page after the user clicks the regulation link. | The system directs the user to the Municode “Fence” regulation page after the user clicks the regulation link. | | Chrome | | Pass | Pending |

## Test Case 37

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-012 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is a “Fence” link that the user can click on to navigate to an external page containing relevant applications for fence use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-9.2 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Fence” application link. | The system shall direct the user to the “Fence” application page after the user clicks the application link. | The system directs the user to the Municode page for “Fence” applications after the user clicks the application link. | | Chrome | | Pass | The link directs the user to the exact location of the permit description. |

## Test Case 38

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-013 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is a “Tree Removal (Residential)” link that the user can click on to navigate to an external page containing relevant regulations for tree removal (residential) use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-10.1 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access, working database server | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Tree Removal (Residential)” regulation link. | The system shall direct the user to the “Tree Removal (Residential)” regulation page after the user clicks the regulation link. | The system directs the user to the Municode page for “Tree Removal (Residential)” regulations after the user clicks the regulation link. | | Chrome | | Pass | Pending |

## Test Case 39

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-014 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is a “Tree Removal (Residential)” link that the user can click on to navigate to an external page containing relevant applications for tree removal (residential) use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-10.2 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Tree Removal (Residential)” application link. | The system shall direct the user to the “Tree Removal (Residential)” application page after the user clicks the application link. | The system directs the user to the Municode “Tree Removal (Residential)” application page after the user clicks the application link. | | Chrome | | Pass | The link directs the user to the exact location of the permit description. |

## Test Case 40

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-015 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is a “Day-Care” link that the user can click on to navigate to an external page containing relevant regulations for day-care use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-11.1 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Day-Care” regulation link. | The system shall direct the user to the “Day-Care” regulation page after the user clicks the regulation link. | The system directs the user to the Municode “Day-Care” regulation page after the user clicks the regulation link | | Chrome | | Pass | Pending |

## Test Case 41

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-016 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is a “Day-Care” link that the user can click on to navigate to an external page containing relevant applications for day-care use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-11.2 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Day-Care” application link. | The system shall direct the user to the “Day-Care” application page after the user clicks the application link. | The system directs the user to the Municode “Day-Care” application page after the user clicks the application link | | Chrome | | Pass | The link directs the user to the exact location of the permit description. |

## Test Case 42

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-017 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is a “Sober Living Facility” link that the user can click on to navigate to an external page containing relevant regulations for sober living facility use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-12.1 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Sober Living Facility” regulation link. | The system shall direct the user to the “Sober Living Facility” regulation page after the user clicks the regulation link. | The system directs the user to the Municode “Sober Living Facility” regulation page after the user clicks the regulation link. | | Chrome | | Pass | Pending |

## Test Case 43

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | | TEST-SQL-018 | | | | **Test Module** | | Database Query | | |
| **Test Case Description** | | There is a “Sober Living Facility” link that the user can click on to navigate to an external page containing relevant applications for sober living facility use case. | | | | **Test Priority** | | High | | |
| **Requirement ID** | | REQ-12.2 | | | | **Test Method** | | Functional Testing | | |
| **Prerequisites** | | TEST-MAP-005 (User is on the Quick Reference Page) | | | | **Assumptions/**  **Constraints** | | Internet access | | |
| **Test Execution Steps** | | | | | | | | | | |
| **Step #** | **Action** | | **Input** | **Expected Output** | **Actual Output** | | **Test Browser** | | **Test Result** | **Test Comment** |
| 1 | Navigate to the “Quick Reference Page”. | | Click on the “Sober Living Facility” application link. | The system shall direct the user to the “Sober Living Facility” application page after the user clicks the application link. | The system directs the user to the Municode “Sober Living Facility” application page after the user clicks the application link. | | Chrome | | Pass | The link directs the user to the exact location of the permit description. |