

# Test Plan

## Test case 1: Initialize database

This test case deals with creating the database, initializing it on the server and getting it ready for data importation.

**Developer's test:** The developer tested this use case while creating it, and 3 more times when changing the server.

**Software risks:** This test case has a third party risk of failure, which has to do with the host of the application (amazon web services).

### Environment:

MySQL Version 14.14 Distribution 5.5.40, for Linux (x86\_64) using readline 5.1

Amazon Linux AMI 2014.09.1 x86\_64 HVM

### Pass/fail criteria:

This test passes 100% of the times, because of the script to create the tables.

## Test case 2: Viewing Map of Vacant Property Records

This test case deals with viewing vacant properties records in map format, after the user has filtered the results chosen to be displayed.

**User's test:** User selects what type of results to display, after displaying the table of results, the user selects the number of results to be displayed in the map.

**Software risks:** this test case has a third party limitation of the number of results to be displayed. The third party service used for this task is google maps geocoder to convert street addresses to latitude and longitude in order to have a pin drop on the map. The free version of the tool allows 12 results to be displayed per request.

### Environment:

PHP 5.3.29

MySQL Version 14.14 Distribution 5.5.40, for Linux (x86\_64) using readline 5.1

JavaScript 1.5

### Pass/fail criteria:

This test passes when the number of results to be mapped is less than 12.

## Test case 3: Viewing Map of Service Request Records

This test case deals with viewing vacant properties records in map format, after the user has filtered the results chosen to be displayed.

**User's test:** User selects what type of results to display, after displaying the table of results, the user selects the number of results to be displayed in the map.

**Software risks:** this test case has a third party limitation of the number of results to be displayed. The third party service used for this task is google maps geocoder to convert street addresses to latitude and longitude in order to have a pin drop on the map. The free version of the tool allows 12 results to be displayed per request.

### Environment:

PHP 5.3.29

MySQL Version 14.14 Distribution 5.5.40, for Linux (x86\_64) using readline 5.1

JavaScript 1.5

### Pass/fail criteria:

This test passes when the number of results to be mapped is less than 12.

## Test case 4: Viewing One Neighborhood's Statistics

This test case deals with viewing information about a single neighborhood.

**User test:** After selecting the desired neighborhood, a table shows up displaying the percentage of each vacant property type and service request type in the selected neighborhood, perimeter, area, acreage, the number of requests and the addresses that have multiple requests.

**Software risks:** database or server failure.

### Environment:

PHP 5.3.29

MySQL Version 14.14 Distribution 5.5.40, for Linux (x86\_64) using readline 5.1

### Pass/fail criteria:

This test passes 100% of the times.

## Test case 5: Viewing all Service Request Records and Vacant Building Records for a Location

This test case deals with viewing service requests and vacant properties for a specific address.

**User test:** the user chooses a street address that they want to view all types of records for that particular address.

**Developer test:** the developer tested this feature for database injections.

**Software risks:** database or server failure.

**Environment:**

PHP 5.3.29

MySQL Version 14.14 Distribution 5.5.40, for Linux (x86\_64) using readline 5.1

**Pass/fail criteria:**

This test passes 100% of the times.

## Test case 6: Visitor Exports Service Request Records or Vacant Building Records

This test case deals with downloading the results of the filtration of the service requests and the vacant buildings. Exports for service requests are done in the service requests page, and exporting of the vacant building is done in the vacant properties page.

After filtering the records, the results can be downloaded as an excel file.

**User test:** this feature was tested by the user and the developer on both sets of records.

**Software risk:** this feature has no risk of failing, assuming the records were fetched properly (which has no link to exporting it to an excel file).

**Environment:**

JavaScript 1.5

**Pass/fail criteria:**

This test passes 100% of the times.

## Test case 7: Logs on and off

This test case deals with administrator logging on and logging off the administrator panel.

**Administrator test:** a designated administrator logs in to the system using their username/email and password. If the log in is successful, he'll be redirected to the administrator panel, if not, a message will tell him that the credentials entered are wrong, and he'll be brought to the login screen again.

**Unit Test:** a unit test is designed to test this feature, and make sure it runs perfectly, and as expected. This test compares a given username and password to those stored in the database, as well as testing a false username and password as it's not in the database.

**Developer test:** the developer tested this feature for database injections.

**Software risk:** the risk arising in this test can be server/database failure

**Environment:**

PHP 5.3.29

PHPUnit 3.4.0

MySQL Version 14.14 Distribution 5.5.40, for Linux (x86\_64) using readline 5.1

**Pass/fail criteria:**

This test passes 100% of the time on both acceptance test and rejection test.

## Test case 8: Administrator Fetches and Imports Data from Open Data Cincy

This test deals with importing data from opendatacincy.com. For this task to be completed, a registered administrator should login, and in the administrator panel, there's a link that imports files automatically to the server.

**Administrator test:** the test is done by deleting the existing files in the downloading directory and click on the link then check the directory if it contains the files.

**Software risk:** the risk for this task arises when dealing with a newly implemented server, where there's a bit of permissions to set up to get this function working properly. Another risk can be a third party risk such as a server failure, or an opendatacincy.com problem.

**Environment:**

PHP 5.3.29

**Pass/fail criteria:**

This test passes 100% of the time

## Test case 9: Administrator Creates a New Administrator User

This test deals adding a new administrator to the list of administrators who can login to the system and perform administrator tasks, such as adding a new administrator, deleting an existing administrator or importing new data.

**Administrator test:** the test is done by accessing the add administrator page and entering the new administrator full name, username, password and email. After that, the new administrator is added, and can be seen on the list of existing administrators.

**Unit Test:** a unit test is designed to test this feature, and make sure it runs perfectly, and as expected. This test compares a given username and password to those stored in the database, as well as testing a false username and password as it's not in the database.

**Software risk:** the risk of this task is database/server failure.

**Environment:**

PHP 5.3.29

**Pass/fail criteria:**

This test passes 100% of the time

## Test case 10: Administrator Removes an Administrator User

This test deals removing an existing administrator. After being deleted, the removed user cannot access the system, since its profile is completely deleted from the database.

**Administrator test:** the test is done by login in to the system, on the administrator panel, an administrators table is displayed and next to each entry there's a delete button that allows to remove that administrator from the system completely, except for the logged in administrator.

**Software risk:** the risk of this task is database/server failure.

**Environment:**

PHP 5.3.29

**Pass/fail criteria:**

This test passes 100% of the time