**Work Order System**

Testing Manual

*April 19, 2023*

SLIPPERY ROCK UNIVERSITY

Ellie Wurst [bmw1030@sru.edu](mailto:bmw1030@sru.edu)

Trevor Bell [tlb1024@sru.edu](mailto:tlb1024@sru.edu)

John Friend [jtf1011@sru.edu](mailto:jtf1011@sru.edu)

Hunter Minteer [hgm1005@sru.edu](mailto:hgm1005@sru.edu)

# Contents (hyperlinks lead to the proper page):

**Work Order System ….............................................................................................. 1**

1. [Overview](#Overview).......................................................................................................... 2
2. [Human-Computer Testing](Testing%20Manual.docx#Human-Computer Testing)………...…………………….…………………… 2
3. [Whitebox](Testing%20Manual.docx#Whitebox Testing) Testing……….…………………………………..……………..… 2
4. [Unit](Testing%20Manual.docx#Unit Testing) Testing……………………………………………………..….……..…. 2
5. [Running](#Running Tests) Tests……………………………..…………………..………………3
6. [Unimplemented Tests/Future Tests](#6.Unimplemented Tests/Future Tests)…………………………..……………….4

# 1.Overview

This manual’s main purpose is to highlight the testing created for the Work Order System. It will discuss the tests written in the Work Order System application, how to run them, and other testing that was done during development.

# 2.Human-Computer Testing

Human-Computer Testing was conducted throughout the continued development of the Work Order System Application. This was done by the contributors to the project during all of production. These tests were to ensure things were working as they should. Most of our early testing occurred with direct user input, and trying to do different things in our systems before test-cases were introduced into this project.

# 3.Whitebox Testing

Whitebox Testing was performed by those working on this project. Whitebox testing was performed by testers who had an inside knowledge of the system. Whitebox testing was undergone specifically by attempting to gain access to places where access should have been denied without being able to login. This was done by understanding the file structure of the System as well as our old html. Whitebox testing was implemented to ensure that users could not access pages of the application without being logged in. Whitebox Testing the application caused errors to be discovered causing changes to be made to the WebSecurityConfig Class to ensure that users could only access their appropriate web-content and weren’t able to see content outside of their scope.

# 4.Unit Testing

Since we were creating a fresh project, we had no previous code to build off with our code. Because of this, we had to create new unique tests for all our implementations that we could and to complete a huge chunk of this, we utilized Unit Testing. The majority of our unit tests were to ensure that our functions are running as intended and that our various domains, repositories, and web based tests.

*Tests Written:*

* *BuildingsTest*
* *DepartmentsTest*
* *IncidentTest*
* *RoleTest*
* *RoomsTest*
* *UsersTest*
* *BuildingRepoTest*
* *DepartmentRepoTest*
* *IncidentRepoTest*
* *RoleRepoTest*
* *RoomRepoTest*
* *UsersRepoTest*
* *BuildingsServiceTest*
* *TestingWebAppTest*
* *WebTests*

# 5.Running Tests

This section highlights running tests in the **Work Order System Application**, In order to run a single test class, select a test class in the ***src/test/java*** folder of the project, and find a test class you’d like to run then, right click on the class and select ***Run As***, next you will select ***JUnit Test***.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 1 – Run Menu

If you wish to run all current available tests, right click on the ***src/test/java*** folder of the Work Order System project and refer to (figure 1) for running single tests. If they ran properly you will see the console process and at the bottom of the screen you will see a Junit tab. To review the tests, click the Junit tab and you will be able to see the results.

Graphical user interface, application

Description automatically generated

Figure 2 – JUnit Results

# 6.Unimplemented Tests/Future Tests

Due to some issues, with getting tests to run and applying values to those, not all tests were finished due to time constraints. However, before this we had nothing to start from on this application. We built it from the ground up and did our best to create tests that felt important and worthwhile as we went and refined them towards the end of the project. Sadly, not all the desired tests were completed within our available time. Below are some of the tests not finished or addable in the future currently:

* BuildingsServiceTest (exists but not implemented)
* AdminServiceTest
* DepartmentsServiceTest
* ExcelServiceTest
* IncidentServiceTest
* RoomServicesTest
* UserServicesTest