Logbook Task 2: Option 3, Task 10

# Test Plan: Instrument Number Feature of Helpdesk Application

#### Rachel Dupuy

[1. Introduction 3](#_Toc1786264095)

[1.1 Project Background 3](#_Toc1679682523)

[1.2 Testing Scope 3](#_Toc608708917)

[1.3 Test Objectives 3](#_Toc203456699)

[2. Testing Approach 5](#_Toc331443773)

[2.1 Risks/Constraints 5](#_Toc284552069)

[2.2 Test Type 5](#_Toc1082549083)

[2.2.1 Requirements Testing 5](#_Toc980450217)

[2.2.2 User Interface Testing 5](#_Toc802700963)

[2.2.3 System Testing 5](#_Toc1042213724)

[2.2.4 Security Testing 6](#_Toc1783627977)

[3. Test Plan 7](#_Toc1337773905)

[3.1 Test Team 7](#_Toc1391704814)

[3.2 Test Environment 7](#_Toc1078894331)

[3.3 Test Deliverables 7](#_Toc1349249083)

[3.3.1 Use Cases 7](#_Toc530369950)

[3.3.2 Unit Testing 8](#_Toc752167828)

[3.3.3 Test Results 8](#_Toc1505597408)

[3.3.4 Test Report 8](#_Toc1784671457)

[3.4 Test Schedule 8](#_Toc448235195)

[4. Risks 8](#_Toc1295749877)

# Introduction

This document will seek to outline the testing process for the Helpdesk Application Instrument List feature. It will describe the testing approach, test data, and the deliverables for the testing process. This document will also detail the risks involved with the testing process of the application.

## 1.1 Project Background

The Helpdesk application is an application designed to ease the burden of doing daily processes on the Helpdesk team. Any SQL queries that are usually manually implemented, can be implemented automatically by filling in the appropriate fields and clicking a button. The Instrument List feature is a feature that allows Helpdesk staff check whether a specific Instrument number exists, and if not, will create a new instrument number with that number.

A colleague and I were tasked by our manager with designing and developing this application. After meeting with the client and discussing the project requirements, we split the design and development work between the two of us. In other words, my colleague worked on one aspect of the project whilst I worked on the other. This meant that we would also be tasked with designing a test plan for each of our perspective parts. As part of this test plan, my other colleague would be responsible for carrying out some of the testing and designing use cases to be tested. Therefore, I needed to delegate some of the testing to my colleague. This is the first part of the application to be completed and ready for testing. Each consecutive part will receive the same treatment.

## 1.2 Testing Scope

The test plan test scope encompasses all areas of the Instrument Number feature to be tested. These areas have been chosen as they reflect the project requirements and have been deemed by me to be integral to user experience and overall functionality of the feature.

* User Interactions (Type in number, click buttons, etc.)
* Events (Button Clicks, Hover, etc.)
* Database Responses (Exists, Not Exists, Errors, etc.)

## 1.3 Test Objectives

* Test the project requirements for the Instrument Number feature, to ensure that they have been met.
* Test User Interface of Instrument Number feature and application to ensure that it functions correctly and does not interfere with any other application interactions.
* Test Database interactions involved in the Instrument Number feature, to ensure the database is being correctly contacted, and that there are no security risks to the database. Also confirm that the database is not corrupted by any of the requests.
* Test Graphical User Interface to ensure that it is user-friendly and user intuitive.

# Testing Approach

The testing approach encompasses the test types I have elected to carry out on the Instrument Number feature as well as any risks and constraints associated with these test types. These test types have been chosen as they cover not only the project requirements, but also functional and non–functional requirements of an application feature.

## 

## 2.1 Risks/Constraints

* There is only a time limit of two weeks to complete the testing process.
* Other parts of the application may have issues with could prevent the Instrument Number feature from functioning properly.

## 2.2 Test Type

### 2.2.1 Requirements Testing

Testing of the Instrument Number feature of the Helpdesk application will involve ensuring that it functions according to the previously specified project requirements. These include:

Verifying that user input is correctly read and manipulated.

Ensuring that requests to the database are functioning correctly.

Verifying that any output is correct.

### 2.2.2 User Interface Testing

Evaluate the Graphical User Interface and ensure that it functions as it should, and that it is user-friendly and intuitive.

### 2.2.3 System Testing

Verify that the Instrument Number feature of the Helpdesk application functions correctly together with other aspects of the application.

### 2.2.4 Security Testing

Ensure that all sensitive data in the database is not compromised by any of the requests sent by the Instrument Numbers feature.

# Test Plan

The test plan outlines crucial information regarding testing, such as the team involved and the tasks that have been delegated to them, the test environment used, and all deliverables expected at each phase of testing. These deliverables are decided by the project requirements and what has been requested by the project manager.

## 3.1 Test Team

|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | Responsibility |
| Rachel Dupuy | Developer | Create Test Plan, Requirements Testing, User Interface Testing |
| Karl Mulcahy | Independent Tester | Create Use Cases, System Testing, Security Testing |

## 3.2 Test Environment

The Instrument Number feature of the Helpdesk application will be tested in a dedicated development environment. This environment has been specifically established by the department to be able to test applications before they are made live to users as well as any updates, patches and fixes that any already live applications require. It is called ITRISdev and it compiles a fully functioning version of the application which developers then can implement and test new features without effecting the functionality of the live application.

The IDE for editing the application and creating tests will be Microsoft Visual Studio. The unit tests will also be created in Visual Studio.

## 3.3 Test Deliverables

### 3.3.1 Use Cases

There are three main use cases of the Instrument Number feature to be tested. The first is a basic use case, in which the user inputs an already existing value into the input field. The user should expect some sort of indication that their input has been read. This indication should be in the form of a dialog box or an alert. The second use case will also be a basic use case in which the user inputs a new value into the input field. The user should receive some sort of indication that their value is a new value and that a new instrument number has been created. The third use case will be when a user inputs an incorrect value or simply leaves the input field empty. The user should receive some sort of indication that their input is invalid, such as a dialog box or alert.

### 3.3.2 Unit Testing

Unit tests will be created to test each of the JavaScript functions in the Instrument Number feature. Each function sends an HTTP request to the database. The unit tests will compare these request responses with an expected value. If the expected values are the same, the test is passed or else it fails. These tests will be run together.

### 3.3.3 Test Results

The results of every test should be recorded along with test parameters and deliverables. These should then be used to compile a test report.

### 3.3.4 Test Report

A test report should be compiled when all testing is complete. It should document each of the testing conditions as well as the results of these tests.

## 3.4 Test Schedule

|  |  |  |
| --- | --- | --- |
| **Name** | **Task** | **Date** |
| Rachel Dupuy | Requirements Testing | 08/01/2024 |
| Karl Mulcahy | Use Cases | 08/01/2024 |
| Rachel Dupuy | User Interface Testing | 10/01/2024 |
| Karl Mulcahy | Unit Testing & System Testing | 10/01/2024 |
| Karl Mulcahy | Security Testing | 15/01/2024 |
| Rachel Dupuy | Test Report | 18/01/2024 |

# Risks

These are the risks associated with testing, the impact it could have on the application, and what setbacks can occur.

* Testing may uncover serious bugs/issues with the Instrument Number feature which may impact the overall timeline of the Helpdesk Application development project.
* Additional testing may be required which may impact on the resource budget for the Helpdesk Application.

# Conclusion

This testing plan outlines the testing approach, timeframe, test type and deliverables expected at each phase of testing. It also describes the team involved in testing and what tasks they have to complete. This test plan will be submitted to the project manager (my manager) who will then approve it to be implemented. Once it has been approved, my colleague and I will then follow it to ensure that adequate testing and quality assurance has been carried out on the Instrument Numbers feature of the Helpdesk application.