**Test Plan for CityHire**

1. **Introduction**

The purpose of this Software Test Plan is to develop a test plan to assess all functionalities of the CityHire. This Test Plan document documents and tracks the necessary information required to effectively define the approach to be used in the testing of the project’s product. The Test Plan document is created during the Planning Phase of the project.

1. **Scope**

There are various test stages that can be performed during the project life cycle: Unit, Functional, Integration, Interface, Performance, Regression, Acceptance, and Pilot testing. Each testing stage may be performed individually or simultaneously in conjunction with another test stage. Each project will determine the testing stages to be completed, which may or may not include all the stages mentioned.

# Test Strategy and Method

The testing phases to be constructed and controlled in line with the project requirements are defined in the following sub-sections. The testing phases that will be controlled will be determined by numerous aspects such as project size, complexity, cost, and risk, and will be different for each project. Unit, functional, integration, interface, performance, regression, user acceptability, and pilot testing are just a few examples of testing phases. Both implicit and explicit requirements must be tested. The plan for each test stage is devised once the testing phases have been chosen. Finally, the testing effort's coverage will be established to verify that it covers all of the needed regions. Examples of test coverage include: requirements, statements, paths, branches, and usage profiles.

Include an explanation of each level of testing's testing procedures. Simulation, modelling, functional, architectural, top-down, bottom-up, demonstration, inspection, software-in-the-loop, and analysis are examples of testing methodologies. Create test readiness requirements for each level of the testing process. Before entering integration testing, software units must have successfully passed a code peer review and unit testing, and a Go or No Go decision must be made before the software enters user acceptance testing.

* 1. **Unit Testing**

This section must detail how unit testing will be managed on the project, as well as the processes for doing so. This part must include include a description of the unit testing technique, including metrics. If unit testing is not regulated, a recommended strategy is to define the test methodology.

* 1. **Functional Testing**

This section must detail how the project will manage functional testing as well as the processes for doing so. This part should include include a definition of the test methodology, as well as metrics for completing functional testing.

* 1. **Integration Testing**

This section must specify how the project will control integration testing as well as the processes for doing so. This part must include include a definition of the test methodology for integration testing, as well as metrics. The integration testing methodology must address how the integration test requirements will be determined in addition to the testing strategy and coverage approach.

* 1. **Interface Testing**

This section must describe the control procedures for interface testing as well as the strategy and coverage approach, including metrics.

* 1. **Performance Testing**

This section must detail the performance/stress testing, its control processes, and the planned testing methodology, including strategy and coverage, as well as metrics. This section should also explain how performance/stress testing interacts with other types of testing and where the test requirements come from.

* 1. **Regression Testing**

This section must describe the regression testing, its control procedures, and the testing methodology proposed to include strategy and coverage, including metrics.

* 1. **User Acceptance Testing**

This part must include the user acceptability testing control techniques, as well as the strategy and coverage methodology, including metrics. The typical user acceptability testing technique is based on the user's input and must address the system's current business requirements. Because the user defines the test cases, the coverage method for user acceptability testing is generally fragmented. This section must detail how the project will undertake user acceptance testing, as well as how the requirements for user approval will be created.

* 1. **Pilot Testing**

This part must include the pilot testing control techniques, as well as the strategy and coverage methodology, as well as metrics. In some cases, pilot testing may not be necessary. The strategy for pilot testing is usually dictated by the user and must address the system's existing business requirements. Because the user determines the test cases, the coverage strategy for pilot testing is generally fragmented. This section must detail how the project will undergo pilot testing, as well as how the user approval requirements will be produced.