## Test Plan: Job Table Component

### Introduction

This test plan outlines the testing approach, resources, schedule, and deliverables for the Job Table component of the Odaseva platform.

### Purpose

The purpose of this test plan is to ensure that the Job Table Search and Reorder components functions correctly and displays accurate information regarding the operations executed on the platform.

### Scope

The testing will cover the following aspects of the Job Table component:

- Display of job information including ID, Name, JobID, Status, Operation, Environment, and Create Date.

- Sorting functionalities.

- Responsiveness and usability across different browsers.

### Assumptions

- The data provided in the Job Table is accurate and reflects real-world scenarios.

- The Job Table component interacts correctly with other components of the Odaseva platform.

### Risks

- Variability in data formats or types may lead to display issues.

- Compatibility issues across different browsers and devices.

- Performance degradation with large datasets.

### Test Objectives

1. To verify that Search function displayed accurate Job results.

2. To ensure sorting and filtering functionalities work as expected.

4. To identify and report any defects or inconsistencies.

### Test Environment

- Software:

Web browsers (Chrome, Firefox, Electron).

Automation Tools: npm, Cypress.io, Cucumber BDD, and other cypress open source libraries

Editors: VSCode

Programming Language: JavaScript, TypeScript, Gherkin

- Operating Systems: Windows, macOS

### Test Deliverables

- Test Cases

- Test Reports

- Defect Reports

### Testing Approach

- Functional Testing: Validate the correctness of displayed job information and functionalities such as sorting and filtering.

- Compatibility Testing: Ensure the Job Table component works across different browsers and devices.

- Usability Testing: Evaluate the user experience and interface responsiveness.

- Regression Testing: Re-test previously validated functionalities after any changes or updates.

-Automation Testing: Automate UI tests using Cypress, Javascript, Typescript, Cucumber, Gherkin and run during regression phase of a release cycle.

### Test Cases

Test feature can be found in project folder: ODASEVA\_CYPRESS\_ASSIGNMENT/cypress/e2e/ job\_table\_search.feature

The following are the test scenarios for verifying the Job list table for Search and Reorder functionality:

Search in Jobs functionality - with keywords for any of the columns

1. Verify that Jobs be searched using ID
2. Verify that Jobs can be searched using JobID
3. Verify that Jobs can be searched using JobName
4. Verify that Jobs can be searched using Status
5. Verify that Jobs can be searched using Operation
6. Verify that Jobs can be searched using Environment
7. Verify that Jobs can be searched using Created Date

Search in Jobs and verify ReOrder columns

* 1. Verify that the Jobs list is Sorted when clicked on column names

### Defect Management

- Defects will be logged using a defect tracking tool.

- Each defect will be assigned a severity and priority.

- Defects will be fixed and retested according to their priority.

### Exit Criteria

- All critical and high-priority defects are fixed and retested.

- Test coverage of all specified requirements is achieved.

- Performance meets acceptable criteria.

- Stakeholder approval is obtained.