

## 1. Purpose of This Document

This document explains the various activities which will be performed as a part of testing a new feature employee management and leave management on orangeHRM.

## 2. Application Overview

OrangeHRM is a comprehensive human resource management (HRM) system designed to help businesses streamline HR processes like recruitment, attendance, and performance evaluations. It offers robust modules for personnel information management, leave management, and time tracking, integrated into an intuitive, user-friendly interface. The platform supports customization and scalability, making it suitable for both small businesses and large enterprises looking to optimize their HR operations.

For more information, visit

<https://opensource-demo.orangehrmlive.com/web/index.php/pim/viewEmployeeList>

## 3. Feature Overview

OrangeHRM features a versatile employee management system that includes detailed personnel profiles, organizational structure visualizations, and document management capabilities. Its leave management module allows for easy tracking and approval of employee absences, while custom reports and analytics provide deep insights into HR metrics, enhancing decision-making. Additionally, the system supports recruitment processes with tools for job posting, candidate tracking, and interview scheduling, streamlining the hiring lifecycle.

## 4. Test Objective

The test objective is to verify the functionality of the employee and leave management on orangeHRM, the test will be executed with the help of the test scripts, identifying, fixing, and retesting all high and medium-severity defects.

## 5. Testing Scope

- Adding, editing, and deleting employee details.
- Validation of data entry forms (mandatory fields, data format).
- Leave entitlement.
- Leave application and approval.

## 6. Test Scenarios

### 6.1 User Management

Positive	Negative
1. Test adding a new employee with all required fields filled out correctly.	1. Verify adding new employee with missing required fields.

2. Verify that all details entered while adding an employee are correctly displayed in the employee profile.	2. Verify invalid data entry such as putting numerical data in text field ,invalid email format ,future date of birth etc
3. Test editing various fields in an existing employee's profile and saving changes.	3. Check whether the system can handle duplicate employee record correctly.
4. Test the search functionality using employee name, ID, or other relevant criteria.	4. Check by modifying employee information with a user account.
5. Test deactivating an employee and ensure they are no longer active in the system.	5. Verify exceeding maximum inputs length.
6. Verify that employee listings can be sorted by name, ID, Sub Unit etc.	6. Verify improper file size and file formats in uploads.
7. Ensure only authorized users can add, edit, or view employee information.	7. Verify search functionality with data that doesn't exist in the system.
8. Check input validation for different fields (e.g., phone numbers, email addresses) to ensure the system accepts only valid data.	8. Verify user try to edit same employee information at the same time.
<b>6.2 Leave Management</b>	
9. Test submitting leave request with all field filled correctly.	9. Attempt to apply for leave when employee's leave balance is insufficient.
10. Test the leave approval process by authorized person.	10. Verify submitting leave request of past date.
11. Check cancellation or rejection of leave request giving proper reason.	11. Verify submit leave request without filling mandatory fields.
12. Modifying the existing leave request without approval	12. Verify invalid date formats while applying for leave.
13. Viewing employee's leave history to ensure past leave data is accurately recorded and accessible.	13. Verify uploading files in unsupported formats.
14. Verify generation of report related to leave.	14. Verify attempt to approve/reject a leave request with user account.

## 7. Test Approach By

<b>QA Team</b>	<b>Rupesh,Sonu,Anil</b>
<b>QA Lead</b>	<b>Amrit</b>

## 7. Test Environment and Tools

<b>Version</b>	
<b>Tested Builds</b>	
<b>Environments</b>	<b>Test Server, Stage Server, Production Server</b>
<b>Browser Compatibility</b>	<b>Chrome, Safari, Edge, Firefox</b>

## 8. Exit Criteria

1. Ensuring all critical Test Cases are passed.
2. Min 95% of test cases are passing.
3. Achieving complete functional Coverage
4. Identifying and fixing all the high-priority defects
5. Fixing all the 'Showstopper defects' or 'Blockers' and ensuring that none of the identified Critical defects are in Open Status
6. Re-testing and closing all the high-priority defects to execute corresponding Regression scenarios successfully
7. All changes and requirements are documented