Test Plan for Vacation Requesting System

1. Introduction

The Vacation Requesting System allows employees to request leave, track their available balance, and route requests for approval. This test plan outlines the approach and coverage to ensure the system meets its functional and non-functional requirements. Testing will include different leave types, workflows, and error handling.

2. Scope

This test plan covers the functional and non-functional aspects of the vacation requesting system. Functional testing includes submitting, approving, rejecting, and canceling leave requests. Non-functional testing includes usability, performance, and data validation checks. The scope does not include backend database testing or integration with external systems unless explicitly mentioned.

3. Objectives

- Ensure the vacation request system allows users to submit, modify, and cancel leave requests accurately.
- Validate leave balances and prevent over-submission of leave.
- Ensure approval workflows function according to user roles.
- Verify error handling for invalid date ranges, user roles, or system failures.

4. Test Items

- Leave Request Module: Testing various leave types (Annual, Sick, Unpaid) and ensuring users can submit leave requests.
- Leave Balance Calculation: Testing leave balance updates after submission, approval, or cancellation of requests.
- Approval Workflow: Ensuring different roles (Employee, Manager, Admin) can approve or reject leave requests.
- UI Components: Checking date pickers, form validation, and error messages.
- Error Handling: Validating appropriate error messages for invalid actions (e.g., leave overlap, negative leave balance).

5. Test Approach

The following types of testing will be performed:

- Functional Testing: Verify that each function of the system performs as expected, including submitting and processing leave requests.
- Positive Testing: Ensure valid inputs (e.g., correct leave dates and balances) behave as expected.
- Negative Testing: Verify that invalid inputs (e.g., future start date after end date) generate appropriate error messages.
- Boundary Testing: Test edge cases, such as a leave request that spans two months or covers the maximum number of days allowed.
- Non-Functional Testing: Focus on usability, ensuring that the system is easy to use and performs well under normal loads.

6. Entry and Exit Criteria

Entry Criteria:

- System design is complete, and the development team has provided a stable build for testing.
- All necessary test cases are written, reviewed, and approved.

Exit Criteria:

- All high-priority test cases are executed with no critical or major defects.
- All identified defects are either resolved or documented for a future release.
- The system is stable, and the performance requirements are met.

7. Test Deliverables

- Test cases document covering all the functional areas of the system.
- Test execution report showing the status of executed test cases.
- Defect report (using the bug report template) outlining any bugs encountered during testing.
- Final test summary report summarizing the overall testing effort and results.

8. Schedule

- Test Planning: 2 days
- Test Case Design: 4 days
- Test Execution: 7 days
- Final Report Submission: 1 day after test completion.

9. Tools and Environment

- Testing Tools: Excel for test case management and bug tracking.
- Environment: Necessary permissions and roles will be configured for different user roles (Employee, Manager, Admin) to ensure complete workflow testing.