**Test Plan: Vacation Requesting System**

**1. Introduction**

The Vacation Requesting System aims to streamline the process of employees requesting vacation time and managers approving or rejecting those requests. This test plan will ensure the functionality, usability, and security of the system.

* **Objective**: Ensure the functionality and reliability of the vacation requesting system.
* **Scope**: This test plan covers functional and non-functional requirements, including positive, negative, and edge scenarios

**2. Scope**

This test plan covers the following functionalities of the VRS:

* Submitting Vacation Requests (specifying dates, duration, reason)
* Leave Balance Tracking and Calculations
* Notification System for Request Status Updates
* Display available leave balance.
* Select start and end dates for leave.
* User Management and Permissions

**3. Functional Requirements**

* Users can view available leave balance.
* Users can submit leave requests for different leave types (e.g., Annual Leave, Sick Leave).
* The system validates leave dates and balance.
* Managers can approve or reject leave requests.
* Notifications are sent upon submission, approval, or rejection.
* Users can cancel or modify leave requests.

**4. Non-Functional Requirements**

* System performance and response time
* Usability and user interface consistency
* Security and data privacy
* Error handling and recovery.

**5. Test Environment**

* Operating System: Windows 10
* Browser: Google Chrome (latest version)

**6. Test Cases**

Each test case should include:

* **Test Case ID**: Unique identifier for the test case.
* **Description**: Brief description of the test case.
* **Preconditions**: Any setup or conditions that must be met before executing the test.
* **Steps to Execute**: Detailed steps to perform the test.
* **Expected Results**: The expected outcome of the test.
* **Actual Results**: The actual outcome after executing the test.
* **Status**: Pass/Fail based on whether the actual result matches the expected result.

**7. User Roles and Scenarios**

* **User Roles**:
  + Employee: Can request leave.
  + Manager: Can approve/reject leave requests.
  + HR: Can view all leave requests and balances.
* **Scenarios**:
  + Employee requests leave.
  + Manager approves/rejects leave.
  + HR views leave balances and requests.

**8. Error Handling**

* Verify that appropriate error messages are displayed for invalid inputs.
* Ensure the system handles network interruptions gracefully.
* Check that the system recovers from unexpected errors.

**9. Test Data**

A variety of test data will be used to simulate real-world scenarios, including:

* Sample vacation requests with different durations and reasons
* Sample faculty and staff user accounts with varying leave balances and approval hierarchies
* Scenarios with overlapping vacation requests within a team
* Test cases simulating high user load to assess system performance

**10. Test Tools**

* Manual testing will be the primary method for functional testing.
* Automated testing tools may be used for regression testing and performance testing.

**11. Roles and Responsibilities**

* Test Lead: Responsible for overall test planning, execution, and reporting.
* Testers: Execute test cases, document results, and report bugs.
* Developers: Fix bugs identified during testing.

**12. Schedule**

* A testing schedule will be developed outlining the timeframe for completing each testing phase.

**13. Reporting**

* A comprehensive test report will be created after completion of testing. The report will include:
  1. Detailed results of each test case
  2. Identified bugs and their severity
  3. Recommendations for improvement

**14. Risk Management**

* Potential risks associated with testing will be identified and mitigated. This may include:
  1. Incomplete or inaccurate test data
  2. Unforeseen system behavior

**15. Approval**

This test plan will be reviewed and approved by relevant stakeholders before testing commences.

**16. Maintenance**

This test plan will be updated as needed to reflect changes in the VRS functionality or requirements.