Vacation Module – Intranet

[case study]

# Introduction

All employees are able to submit/edit/cancel a vacation request.

Each request should be submitted to the approver before the vacation start date with “Vacation Days\*2”

# New Request

### Mandatory fields for submission

* Start Date
* End Date
* Replacement (A replacement must exist in the system and be available within the selected duration.)
* Selected vacation duration has a max limit =< 15 working days.
* Request submission date shall be before the vacation start date with “Vacation Days\*2”

### Optional fields:

* Comments field
* Send notification email to the replacement- checkbox.

### CTA:

* Submit
* Cancel

# Edit / Cancel Request

### Edit/Cancel actions must be done before the submitted request is accepted/declined from the approver. Otherwise, the submitted request appears as read-only.

# Approval WF

### Accept - notification is sent to both requester & replacement to inform them that the request is approved.

### Request Update + Mandatory comment to mention the update - notification is sent to the requester only to update his request with the needed details.

### Reject + Optional reason field - notification is sent to the requester.

Certainly! Let's outline the test scenarios for the Approval WF feature:

1. \*\*Accept Request:\*\*

Understood! Let's create detailed test cases for the "Accept Request" scenario following the same format:

\*\*Test Case 1: Notification to Requester\*\*

- \*\*Title:\*\* Notification to Requester

- \*\*Description:\*\* Verify that when a request is accepted by the approver, a notification is sent to the requester informing them of the approval.

- \*\*Steps:\*\*

1. Log in to the system as the approver.

2. Navigate to the list of pending requests.

3. Select the request awaiting approval.

4. Accept the request.

5. Confirm the action.

6. Verify that a notification is sent to the requester.

- \*\*Test Data:\*\*

- Request awaiting approval.

- \*\*Pre-Conditions:\*\*

- A request must be submitted and awaiting approval.

- \*\*Expected Result:\*\*

- The system successfully sends a notification to the requester, informing them of the approval.

- \*\*Actual Result:\*\* [To be filled after execution]

- \*\*Priority:\*\* Medium

- \*\*Status:\*\* Not executed

\*\*Test Case 2: Notification to Replacement\*\*

- \*\*Title:\*\* Notification to Replacement

- \*\*Description:\*\* Verify that when a request is accepted by the approver, a notification is sent to the replacement informing them of the approval.

- \*\*Steps:\*\*

1. Log in to the system as the approver.

2. Navigate to the list of pending requests.

3. Select the request awaiting approval.

4. Accept the request.

5. Confirm the action.

6. Verify that a notification is sent to the replacement.

- \*\*Test Data:\*\*

- Request awaiting approval.

- \*\*Pre-Conditions:\*\*

- A request must be submitted and awaiting approval.

- \*\*Expected Result:\*\*

- The system successfully sends a notification to the replacement, informing them of the approval.

- \*\*Actual Result:\*\* [To be filled after execution]

- \*\*Priority:\*\* Medium

- \*\*Status:\*\* Not executed

These test cases outline the specific steps to verify that notifications are sent to both the requester and the replacement upon approval of the request. Let me know if you need further adjustments or if you're ready to proceed with the next steps!

2. \*\*Request Update with Mandatory Comment:\*\*

Here's the detailed test case for the "Request Update with Mandatory Comment" scenario:

\*\*Title:\*\* Request Update with Mandatory Comment

\*\*Description:\*\* Verify that when the approver updates the request and adds a mandatory comment, a notification is sent to the requester, prompting them to update their request with the necessary details.

\*\*Steps:\*\*

1. Log in to the system as the approver.

2. Access the request that needs an update.

3. Make the necessary updates to the request.

4. Add a mandatory comment mentioning the update.

5. Save the changes.

6. Verify that a notification is sent to the requester prompting them to update their request with the needed details.

\*\*Test Data:\*\*

- Existing request data.

- Mandatory comment mentioning the update.

\*\*Pre-Conditions:\*\*

- A request must be submitted and awaiting approval.

\*\*Expected Result:\*\*

- A notification is successfully sent to the requester, prompting them to update their request with the necessary details.

\*\*Actual Result:\*\* [To be filled after execution]

\*\*Priority:\*\* Medium

\*\*Status:\*\* Not executed

3. \*\*Reject Request with Optional Reason:\*\*

Here's the detailed test case for the "Reject Request with Optional Reason" scenario:

\*\*Title:\*\* Reject Request with Optional Reason

\*\*Description:\*\* Verify that when a request is rejected by the approver with an optional reason provided, a notification is sent to the requester, informing them of the rejection and optionally including the reason provided by the approver.

\*\*Steps:\*\*

1. Log in to the system as the approver.

2. Access the request that needs to be rejected.

3. Choose the option to reject the request.

4. Optionally provide a reason for the rejection.

5. Confirm the rejection action.

6. Verify that a notification is sent to the requester, informing them of the rejection and optionally including the reason provided by the approver.

\*\*Test Data:\*\*

- Existing request data.

- Optional reason for rejection.

\*\*Pre-Conditions:\*\*

- A request must be submitted and awaiting approval.

\*\*Expected Result:\*\*

- A notification is successfully sent to the requester, informing them of the rejection and optionally including the reason provided by the approver.

\*\*Actual Result:\*\* [To be filled after execution]

\*\*Priority:\*\* Medium

\*\*Status:\*\* Not executed