

Feature Specification: Email Notifications for Key Events

1. Introduction

- **1.1 Feature Name:** Email Notifications for Key Events
- **1.2 Feature ID:** FEAT-004
- **1.3 Version:** 1.0
- **1.4 Date:** June 12, 2025
- **1.5 Author(s):** Maximo Valencia
- **1.6 Stakeholders:** Product Management, Engineering Team, Marketing, Customer Success.
- **1.7 Category:** Backend

2. Overview

This feature will enable the application to send automated email notifications to users for significant events, such as task assignments, comments on their items, or status changes. Users will have granular control over which types of notifications they wish to receive.

3. Goals and Objectives

- **3.1 Primary Goal:** To keep users informed about critical updates and activities related to their work within the application.
- **3.2 Objectives:**
 - Send timely email notifications for predefined event types.
 - Provide users with preferences to enable/disable specific notification types.
 - Ensure emails are well-formatted, clear, and include direct links to the relevant content in the application.
 - Track email delivery status.

4. User Stories

- As a user, I want to receive an email when a new task is assigned to me so I don't miss important work.
- As a user, I want to receive an email when someone comments on an item I own or am following so I can stay engaged.
- As a user, I want to be able to turn off email notifications for certain types of events if I find them too frequent.
- As an administrator, I want to know if notification emails are being sent successfully.

5. Functional Requirements

- **FR-5.1 Notification Event Triggers:** Implement triggers for the following events (initial list, extendable):
 - New task assigned to me.
 - Comment added to my item.
 - Status change on an item I own or follow.
 - (Add more as needed based on application context)
- **FR-5.2 User Notification Preferences:**
 - A new section in User Settings will allow users to toggle each notification type (e.g., "Email me for new assignments: [On/Off]").
 - Changes to preferences must be saved instantly.
- **FR-5.3 Email Content:**
 - Emails must include a clear subject line indicating the event type.
 - Email body should contain relevant details (e.g., item name, initiator, new status, snippet of comment).
 - A direct link to the relevant item/page within the application must be included.
 - A link to "Manage your notification preferences" must be included in the footer of every email.
- **FR-5.4 Email Templating:** Use a templating system for emails to ensure consistency and ease of updates.

- **FR-5.5 Rate Limiting:** Implement rate limiting to prevent overwhelming users with too many emails in a short period.
- **FR-5.6 Delivery Status:** Log the status of email sends (e.g., sent, failed, opened - if supported by email service).

6. Non-Functional Requirements

- **NFR-6.1 Deliverability:** High email deliverability rate (avoiding spam folders).
- **NFR-6.2 Scalability:** The notification system must handle a growing number of users and events.
- **NFR-6.3 Timeliness:** Notifications should be sent within minutes (or near real-time) of the triggering event.
- **NFR-6.4 Security:** Email content should be sanitized to prevent XSS or other vulnerabilities.

7. UI/UX Specifications

- **7.1 Settings Interface:** Clear, organized toggles in user settings.
- **7.2 Email Design:** Professional, branded email templates that are responsive across different email clients.
- **7.3 Email Content Clarity:** Concise and actionable language in emails.

8. Technical Design & Implementation Details (High-Level)

- **8.1 Technology Stack:** Python, Node.js, Java
- **8.2 Email Service Provider (ESP) Integration:** Integrate with a reliable ESP (e.g., SendGrid, Mailgun, AWS SES).
- **8.3 Asynchronous Processing:** Use a message queue (e.g., RabbitMQ, Kafka, Redis Queue) and background workers/jobs to process notification events and send emails asynchronously.
- **8.4 Event-Driven Architecture:** Emit events when specific actions occur (e.g., task.assigned, comment.created, item.status_changed).

- **8.5 Database Schema Changes:** Add a notification_preferences table or extend the users table to store user choices.

9. Test Cases (Examples)

- **TC-9.1:** Create a new task and assign it to another user; verify they receive the assignment email.
- **TC-9.2:** Add a comment to an item owned by another user; verify the owner receives the comment email.
- **TC-9.3:** As a user, disable "New Assignment" emails. Assign a task to that user; verify no email is received.
- **TC-9.4:** Verify clicking the link in the email navigates to the correct item in the application.
- **TC-9.5:** Test email rendering across various email clients (Gmail, Outlook, Apple Mail).

10. Open Questions / Dependencies

- What are the exact wording and content for each email template?
- Which specific ESP will be used?
- Do we need a "digest" option for less frequent email updates? (Out of scope for V1).