

FUSION ERP

Software requirement specification

OS-3

NOTIFICATION(APP)

Team Details

T.A Pradhyumn (21BCS219)

Prateek Pratap Singh(21BCS162)

Sahil Gautam(21BCS182)

Arjit Patel(Student Mentor)

Dr. NEELAM DAYAL (FACULTY MENTOR)

1. INTRODUCTION

1.1 INTRODUCTION ABOUT THE FUSION – A BRIEF DESCRIPTION

FusionIIIT exemplifies the seamless integration and automation of diverse functions at PDPM Indian Institute of Information Technology, Design and Manufacturing, Jabalpur. This student-driven initiative, meticulously crafted using Python 3.8 and powered by the Django Web framework, aims to enhance the operational landscape of the institute. From streamlining administration management to fostering academic excellence and handling miscellaneous departmental tasks, FusionIIIT serves as a comprehensive solution that aligns with the intricacies of campus life.

Picture it as a digital wizard that oversees every aspect, from organizing administrative affairs to facilitating smoother academic processes. Its reach extends beyond the conventional tasks, actively engaging with various departments and sections to ensure the seamless functioning of every facet of campus life.

On the administrative front, FusionIIIT efficiently manages complex paperwork and processes, lifting the burden from staff members. In the realm of academics, it introduces a digital touch, simplifying learning and course management. However, its role transcends these boundaries; FusionIIIT emerges as a friendly companion for every corner of the campus, diligently overseeing and optimizing operations.

In essence, FusionIIIT is not merely a tool; it embodies a supportive ally, contributing to a more organized and enjoyable life at PDPM IIITDM Jabalpur for everyone involved.

1.2 PURPOSE OF THE MODULE

The Notification Management module facilitates a user-friendly interface for efficient handling of notifications within the app. Users can seamlessly receive, read, mark as unread, and delete notifications, enhancing control and convenience. The module aims to optimize resource utilization, ensuring operational efficiency and user satisfaction in managing communication within the application.

1.3 SCOPE OF THE MODULE

The Notification Management module facilitates efficient handling of notifications within the app, enabling users to receive, read, mark as unread, and delete notifications. It focuses on optimizing user control, providing administrators with tools for effective system management. The scope includes enhancing user experience, resource optimization, and maintaining operational efficiency in communication.

2. USER/ACTOR CHARACTERISTICS

2.1 STUDENT / FACULTY / STAFF :

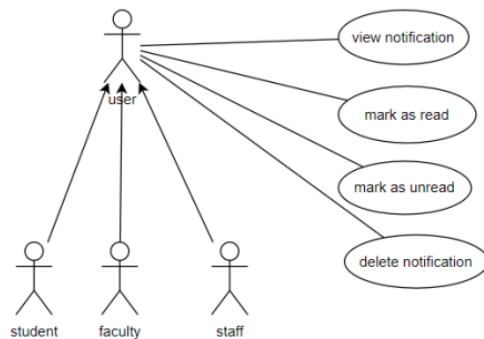
Role: Can generate a notification relevant to the particular module.

SPECIFIC FUNCTIONALITIES :

- 1.The user can open the module by clicking on the module name present in the notification while viewing it.
- 2.The user can click on mark as read on a new notification.
- 3.The user can click on mark as unread on a already marked as read notification.
- 4.The user can delete a notification.

3. FUNCTIONAL REQUIREMENTS

3.1 USE CASE DIAGRAM



Note

Notification panel should be able to retrieve any announcement made or any status change through any of the department and it should be displayed to authorized personnel only

3.2 USE CASE DESCRIPTION

This section describes each use case Description in the use case diagram in all details.

1.

UC ID	UC#1	
Use case Name	view notification	
Description	The " View Notification " use case allows the User to view notifications about their activity from different modules through the Fusion portal.	
Actor	User (Student/Faculty/Staff)	
Precondition	The User is logged in into the portal and has minimum 1 notification.	
Main Flow	1	The User navigates to the “Notifications” section in the dashboard.

	2	Then the user gets displayed the list of their notifications.
	3	The User selects a notification to view.
	4	The User can now read the content of that notification and can see from which module it came from
Post conditions	If that particular notification was new it no longer remains new because it is been read by the user.	
Sub Flow	NIL	
Global Alternate Flow	GA 1	If a technical error occurs during the execution of any action (e.g., database failure, server issues), the system displays an error message and logs the incident.

2.

UC ID	UC#2	
Use case Name	mark as read	
Description	The "Mark as Read" use case allows the User to mark a new notification as read if they wish not	
Actor	User (Student/Faculty/Staff)	
Precondition	The User is logged in into the portal and has minimum 1 notification.	
Main Flow	1	The User navigates to the “Notifications” section in the dashboard.
	2	Then the user gets displayed the list of their notifications.
	3	The User selects a notification to view.

	4	The User can now read the content of that notification and can see from which module it came from
Post conditions	If that particular notification was new it no longer remains new because it is been read by the user.	
Alternate Flow	The user cancels the mark as read action.	
Sub Flow	NIL	
Global Alternate Flow	GA 1	If a technical error occurs during the execution of any action (e.g., database failure, server issues), the system displays an error message and logs the incident.

3.

UC ID	UC#3	
Use case Name	mark as unread	
Description	The "Mark as Unread" use case allows the User to mark an already read notification as unread if they wish not to have that notification as read through the Fusion portal.	
Actor	User (Student/Faculty/Staff)	
Precondition	The User is logged in into the portal and has minimum 1 notification.	
Main Flow	1	The User navigates to the “Notifications” section in the dashboard.
	2	Then the user gets displayed the list of their notifications.

	3	The User selects a notification to view.
	4	The User can now read the content of that notification and can see from which module it came from
Post conditions	That particular already marked as read notification now becomes new.	
Alternate Flow	The user cancels the mark as unread action.	
Sub Flow	NIL	
Global Alternate Flow	GA 1	If a technical error occurs during the execution of any action (e.g., database failure, server issues), the system displays an error message and logs the incident.

4.

UC ID	UC#4	
Use case Name	Delete notification	
Description	The "Delete Notification" use case allows the User to delete a notification from the notifications they get displayed through the Fusion portal.	
Actor	User (Student/Faculty/Staff)	
Precondition	The User is logged in into the portal and has minimum 1 notification.	
Main Flow	1	The User navigates to the “Notifications” section in the dashboard.
	2	Then the user gets displayed the list of their notifications.
	3	The User selects a notification to view.

	4	The User can now read the content of that notification and can see from which module it came from
Post conditions	That particular notification now gets deleted and longer displayed to the User.	
Alternate Flow	The user cancels the delete action .	
Sub Flow	NIL	
Global Alternate Flow	GA 1	If a technical error occurs during the execution of any action (e.g., database failure, server issues), the system displays an error message and logs the incident.

3.3 OTHER FUNCTIONAL REQUIREMENTS

- Notification Integration:
 - All modules will utilize the Notification module for issuing notifications and alerts to relevant stakeholders. This includes raising concerns about grades or any other pertinent issues.
- Notification Generation and Visibility:
 - Each module must facilitate the creation and visibility of notifications for respective users.
- Critical Inventory Alerts:
 - The system will generate alerts for critical inventory levels and other crucial updates.

3.4 OTHER CONSTRAINTS

3.4.1 User Interfaces

The user interface should comply with the color scheming and dashboard design of the FUSIONIIT. Users should be able to navigate from one functionality to other. Inter module navigation should be smooth. All the

functionalities should be easy to use and no specific training should be required for the usage of the module.

3.4.2 Tech Stack Used

- 1) Backend: Django(Python Based Web-Framework).
- 2) Frontend: Flutter(Dart Based framework)

3.4.3 Business rules : Acad-Admin will take care of verified results and will look after any screwing patterns in gradings.

4. NON- FUNCTIONAL REQUIREMENTS

4.1 **PERFORMANCE:**

- Swift response to user interactions is imperative. The notification module should exhibit low response times for generating notifications and managing read/unread statuses.

4.2 **SCALABILITY:**

- The notification module should efficiently handle a significant number of concurrent users. Scalability testing should be conducted under varying load conditions to ensure optimal performance.

4.3 **AVAILABILITY:**

- The notification module is expected to maintain a high level of availability, targeting a 99.9% uptime. Users should experience minimal downtime for seamless access.

5. MODULE DEPENDENCIES WITH OTHER FUSION MODULES

5.1. **UI LEVEL**

Integration in Fusion

At the UI level, the Notification Module seamlessly integrates with other modules, maintaining a consistent theme across the Fusion platform. Upon signing into the Fusion portal, users will find a notification bar in the dashboard, offering the following functionalities:

5.2. DB LEVEL DEPENDENCIES:

- Mark as read
- Mark as unread
- Description of the notification and the module name. Clicking the module name redirects to that specific module.
- Every module will share the notifications schema with the notifications module.

5.3. MODULE LEVEL DEPENDENCIES :

1. In the file tracking module, upon sending a notification, the sender will receive an acknowledgment notification confirming that the file has been sent successfully. Simultaneously, the recipient will be notified that the file has been received.

2. The doctor appointment form within the health center module is not working, while the ambulance request function is operational, and ambulance requests are directed to the compounder. After submitting feedback, it is unclear who receives the notifications.

3. Notifications are not functioning in the visitors hostel module due to its incomplete functionality. The form details are not being transmitted to the caretaker responsible for the visitors' hostel.

4. Within the gymkhana module, all notifications are operational, except for the club membership form, where the selection of the club is not functioning, rendering it inactive. The co-convener can send voting poll notifications to specific batches, event notifications to all batches, and session notifications to all batches within the gymkhana module.

5. In the research procedures module, the patent form is not working, so it is currently non-functional in generating notifications. Similarly, within the other academic procedures module, both the leave module and assistant - claimship module forms have notification functions defined, but they are not working. These are currently not active for generating notifications.

6. In the complaint system module, once a student submits a complaint, a notification will be forwarded to the designated caretaker assigned to handle that specific complaint.

7. Within the mess module, notifications will be accessible upon submission of feedback, requests for changes in the mess menu, and leave or vacation requests from the staff. Additionally, when a student is added to the mess committee, they will receive a notification informing them of their new role.

8. In the Department module the publication of an announcement by the department head, notifications will be sent to both the department's students and the head of the department. These notifications will be accessible in the announcements section.