

**Fusion ERP**

# **Software Requirements Specification**

**for**

**GAD 3 – Complaint management system**

**Faculty Mentor:** Dr. Avinash Chandra Pandey

**Prepared by:**

Ayushi Mourya (21BCS050)

Bachuwar Esha (21BCS052)

Kirankumar G (21BCS116)

Nidhi Barapatre (21BCS146)

Sadanand Jaiswal (21BCS180)

**Mentor:**

Pratik Kumar (21BCS164)

# **1. Introduction**

## **1.1 Introduction about the Fusion – A brief Description**

FusionIIIT stands as a testament to the seamless integration and automation of diverse functions within PDPM Indian Institute of Information Technology, Design and Manufacturing, Jabalpur. Crafted with precision using Python 3.8 and powered by the Django Web framework, this initiative is a student-driven endeavor designed to elevate the institute's operational landscape. Encompassing everything from efficient administration management to academic prowess and miscellaneous departmental tasks, FusionIIIT is a holistic solution that harmonizes the intricacies of campus life.

Imagine it as a digital wizard that takes care of everything, from organizing the administrative stuff to making academics smoother. It's not just limited to the usual tasks; FusionIIIT jumps into various departments and sections, making sure every corner of campus life runs smoothly.

In the admin side, it handles the complicated paperwork and processes. For academics, it brings a digital touch, making learning and managing courses easier. But it doesn't stop there; FusionIIIT is like a friendly companion for all the different parts of the campus, making sure everything works well.

In simpler terms, FusionIIIT is not just a tool – it's a helpful friend, making life at PDPM IIITDM Jabalpur more organized and enjoyable for everyone.

## **1.2 Purpose of the module:**

The Centralized complaint system combines the issues related to Computer Center, Hostels and Cleanliness under one roof. This software aims at solving problems of students as well as staff as soon as possible by providing a platform to connect them directly to the Warden/Caretaker/Supervisor of the respective departments. It will allow immediate addressal of complaints

## **1.3 Scope of the module:**

This module aims to provide a comprehensive solution for efficiently managing and documenting user complaints within our institute. This system is designed to replace traditional paper-based complaint tracking methods, ensuring a centralized and organized approach to handling user complaints and their feedback.

## **2. User / Actor characteristics**

### **1. User (Student, Faculty, Staff):**

Represents individuals who intend to lodge a complaint regarding various issues of hostels.

#### **Specific Functionalities:**

- Users can lodge complaints.
- Users can track status of their complaints.
- Users can give feedback.

### **2. Admin /Section Incharge:**

Represents individuals who intend to respond to complaint registered by the Users only when the complaint has been forwarded by the caretaker or not been handled by the caretaker within a specified time period.

#### **Specific Functionalities:**

- Admin/ Section Incharge can respond to the complaints.
- Admin/ Section Incharge can change the status of complaint

### **3. Caretaker:**

Represents individuals who will be receiving all the complaints and taking actions accordingly. If the complaint remains unresolved or can not be handled at this level, it will be forwarded to the higher concerned authorities (Admin/ Section Incharge).

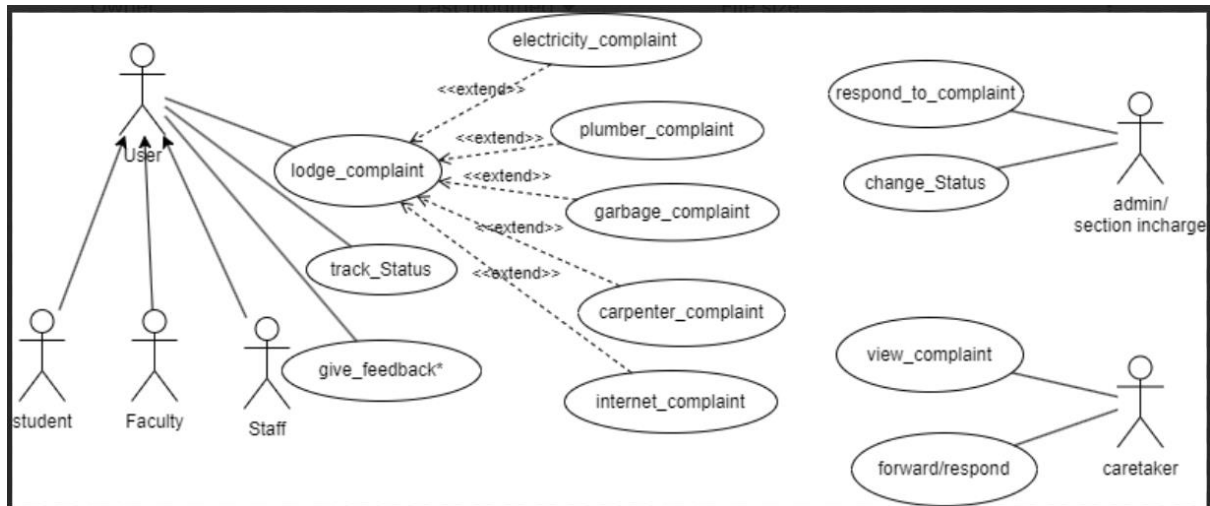
#### **Specific Functionalities:**

- Caretaker can view the complaints registered by the users.
- Caretaker can respond to the complaints.
- Caretaker can forward the complaints to the respective higher authorities.

## 3. Functional Requirements

### 3.1 Use Case Diagram

The figure represents the use case diagram for the module features.



## 3.2 Use Case Description:

### 3.2.1: Use Case #1

|                |  |  |
|----------------|--|--|
| UC ID          | UC#1   |  |
| Use Case Name  | <b>lodge_complaint</b>   |  |
| Actor          | Student, Staff, Faculty  |  |
| Pre-Condition  | The user must fill the all the necessary fields in the complaint page.           |  |
| Main Flow      | 1.   | User fills all the necessary columns and clicks on the submit button.    |
|                | 2.   | System displays a message that complaint has been successfully received. |
| Post-Condition | The complaint is successfully received by the system and stored in the database. |  |

### 3.2.2: Use Case #2

|                |  |  |
|----------------|--|--|
| UC ID          | UC#2   |  |
| Use Case Name  | <b>track_Status</b>  |  |
| Description    | After logging into the system, the user can see the complaint history and the status of the respective complaints. |  |
| Actor          | Student, Staff, Faculty  |  |
| Pre-Condition  | User must be logged-in.  |  |
| Main Flow      | 1.   | User choose the Complaint History to check the status of the complaints.   |
|                | 2.   | System displays the page showing the status of all the complaints i.e whether the complaint has been addressed by the caretaker or not, with the time and date when the user had registered the complaint. |
|                | 3.   | User can see which of his complaints are resolved, kept on-hold and complaints that are declined.  |
| Alternate Flow | NIL  |  |

### 3.2.3: Use Case #3

|                       |   |   |
|-----------------------|---|---|
| <b>UC ID</b>          | UC#3  |   |
| <b>Use Case Name</b>  | <b>give_feedback</b>  |   |
| <b>Description</b>    | User has the option to give feedbacks to the service after his/her complaint has been resolved. |   |
| <b>Actors</b>         | Student, Staff, Faculty   |   |
| <b>Pre-Condition</b>  | User must be logged-in.   |   |
| <b>Main Flow</b>      | 1.  | User chooses the feedback option given on the first page.   |
|                       | 2.  | System shows the next page.   |
|                       | 3.  | User choose the type of feedback i.e whether the feedback is about CC (Library+CC), hostel, residential or garbage. |
|                       | 4.  | User gives the feedback.  |
|                       | 5.  | User submits the feedback.  |
|                       | 6.  | System displays the initial page.   |
| <b>Post-Condition</b> | Feedback is successfully received by the system and stored in the database.                     |   |
| <b>Alternate Flow</b> | NILL  |   |

### 3.2.4 Use Case #4

|                       |  |   |
|-----------------------|--|---|
| <b>UC ID</b>          | UC#4   |   |
| <b>Use case Name</b>  | <b>change_status</b>   |   |
| <b>Description</b>    | Admin/Section Incharge can change the status of the complaint.   |   |
| <b>Actor</b>          | Admin/Section Incharge   |   |
| <b>Pre-Condition</b>  | The user must be logged-in. Then he/she must click on the unresolved complaints thereafter on Hold Complaints. |   |
| <b>Main Flow</b>      | 1.   | System displays the page of on hold complaints.             |
|                       | 2.   | User clicks on the change status button at the corner.      |
|                       | 3.   | User changes the state of work from unresolved to resolved. |
| <b>Post-Condition</b> | Student/Staff gets notified with the changed status of the complaint.  |   |
| <b>Alternate Flow</b> | NIL  |   |

### 3.2.5 Use Case #5:

|                       |   |   |
|-----------------------|---|---|
| <b>UC ID</b>          | UC#5  |   |
| <b>Use case Name</b>  | <b>respond_to_complaint</b>   |   |
| <b>Description</b>    | Enables Admin/Section Incharge to efficiently respond to a complaint forwarded by the caretaker.                      |   |
| <b>Actor</b>          | Admin/Section Incharge  |   |
| <b>Pre-Condition</b>  | The user must be logged-in. There should be an existing unresolved complaint.   |   |
| <b>Main Flow</b>      | 1.  | User navigates to the “View Complaint” section. |
|                       | 2.  | Chooses a complaint.                            |
|                       | 3.  | Drafts and submits a response to the complaint. |
| <b>Post-Condition</b> | Successfully respond to the complaint.  |   |
| <b>Alternate Flow</b> | If no complaints are available the system notifies the admin / section incharge that there are no complaints to view. |   |

### 3.2.6 Use Case #6

|                       |   |  |
|-----------------------|---|--|
| <b>UC ID</b>          | UC#6  |  |
| <b>Use case Name</b>  | <b>view_complaint</b>   |  |
| <b>Description</b>    | Caretaker can view the complaint detail.  |  |
| <b>Actor</b>          | Caretaker   |  |
| <b>Pre-Condition</b>  | The Caretaker must be logged-in. There should be existing complaint submitted by the User.              |  |
| <b>Main Flow</b>      | 1.  | Caretaker login into the Complain Management System. |
|                       | 2.  | Navigates to the “View Complaint” section.           |
|                       | 3.  | Selects a complaint from the list.                   |
| <b>Post-Condition</b> | NIL   |  |
| <b>Alternate Flow</b> | If no complaints are available the system notifies the care taker that there are no complaints to view. |  |

### 3.2.6 Use Case #7:

|                       |   |  |
|-----------------------|---|--|
| <b>UC ID</b>          | UC#7  |  |
| <b>Use case Name</b>  | <b>forward / respond</b>  |  |
| <b>Description</b>    | Caretaker can forward the complaint to concerned higher authorities or respond to the complaint if it has been resolved or will be resolved without the interference of higher authorities. |  |
| <b>Actor</b>          | Caretaker   |  |
| <b>Pre-Condition</b>  | The Caretaker must be logged-in. There should be an existing complaint that needs to be forwarded.  |  |
| <b>Main Flow</b>      | 1.  | User navigates to the “View Complaint” section.                          |
|                       | 2.  | Choose a complaint and either forward or respond by changing the status. |
| <b>Post-Condition</b> | The Caretaker may update the complaint status.  |  |
| <b>Alternate Flow</b> | If no complaints are available the system notifies the care taker that there are no complaints to view.   |  |



### **3.3 Other Functional Requirements**

- This module will use notification and alert to notify the complaints to respective actors.
- This software should be compatible with different browsers like Google Chrome, Firefox, Edge and Safari. Users should be able to use and navigate on any browser available.

### **3.4 Other Constraints:**

#### **3.4.1 User Interface:**

The color palette should match the design and pattern of the FusionIIIT dashboard. All the buttons should be clickable and there should be smooth navigation from one functionality to another. Inter module navigation should be reachable from within the module. The UI should be easy to use and intuitive.

#### **3.4.2 Tech Stack:**

Web application uses Django framework for frontend and backend functionality. PostgreSQL is used as the RDBMS.

## **4. Non-Functional requirements**

### **4.1 Ensuring security**

The user has to login before he/she could access anything or make a complaint. Admin can only make changes in the categories of complaints and other related issues. Access to any application resource will depend upon user's designation.

### **4.2 Performance Requirements**

The server on which the website and database are hosted must be online 24X7. It must be able to handle multiple requests of different users at a time. Response time of the complaint management system should be less. Response time refers to the waiting time while the system accesses queries and retrieves the information from the database.

### **4.3 Reliability**

It should be robust with high degree of fault tolerance. Any user error should be identified and user should be notified about it. It should have mechanisms to overcome hardware failures, power failures and roll back databases to their last checkpoint. This application should be highly reliable and it should generate all the updated information by student, faculty and administrator in correct order.

### **4.4 Usability**

The interface should be easy to use. The web interface should be intuitive and easy to navigate. Users should be able to understand the menu and options provided by the software. Any notification or error messages should be precise and clear. The software should be able to understand and capable to learn, Use and attractive to the user when used under specified condition.

## 5 Module dependencies with other fusion modules

### 5.1 UI Level :

Users log in to the Fusion application and are presented with the main dashboard. The user (student, staff, faculty) can lodge a complaint by filling the required input fields. They can also view their respective complaint history.

### 5.2 Database Level:

Following are the tables and their respective dependencies.

| S.no | Table Name                        | Foreign key     | Referenced Table           |
|------|-----------------------------------|-----------------|----------------------------|
| 1    | complaint_system_caretaker        | staff_id_id     | globals_extrainfo          |
| 2    | complaint_system_hall             | staff_id_id     | globals_extrainfo          |
| 3    | complaint_system_studentcomplaint | complainer_id   | globals_extrainfo          |
|      |                                   | worker_id_id    | complaint_system_workers   |
| 4    | complaint_system_supervisor       | sup_id_id       | globals_extrainfo          |
| 5    | complaint_system_workers          | caretaker_id_id | complaint_system_caretaker |

### 5.3 Module Level:

- **Notification:**

Dependency Direction: Complaint management system Module depends on Notifications Extension.

**Purpose:** To trigger notification when a complaint is registered or solved.

- **Dashboard:**

Dependency Direction: Complaint management system Module depends on Dashboard Extension.

**Purpose:** To provide a unified and real-time interface for monitoring, tracking, and resolving complaints.

- **FTS:**

Dependency Direction: Complaint management system Module depends on FTS Extension.