Software Requirements Specification

For

Department Module (AC6-Web)

Faculty Mentor: Dr. Pritee Khanna

Mentor: Prajjwal Kapoor(21bcs158)

Team Members:

Aryan Shrivastava(21bcs040)

Rushikesh Shinde(21bcs190)

Manoj Panjwani(21bec068)

Sudarshan Patidar(21bcs208)

Jesvia Susan Varghese(21bcsd01)

1. Introduction

1.1 Introduction about the Fusion – A Brief Description

Fusion IIIT serves as a testament to the seamless integration and automation of diverse functions within the precincts of PDPM Indian Institute of Information Technology, Design, and Manufacturing, Jabalpur. Developed with precision through the utilization of Python 3.8 and bolstered by the Django Web framework, this initiative stands as a testament to a student-driven pursuit geared towards the enhancement of the operational fabric of the institute. Its purpose is to provide a comprehensive solution, addressing myriad facets ranging from adept administration management to the facilitation of academic excellence and the efficient handling of departmental tasks, thereby bringing cohesion to the intricate tapestry of campus life.

On the administrative front, Fusion IIIT adeptly manages intricate paperwork and processes. In the academic sphere, it introduces a digital paradigm, simplifying the realms of learning and course management. Its impact, however, transcends these specific domains; Fusion IIIT assumes the role of a genial companion for every facet of the campus, meticulously ensuring the optimal functioning of all processes.

In summary, Fusion IIIT transcends its status as a mere tool; it emerges as a discerning companion, contributing significantly to a more organized and enjoyable milieu for the entire community at PDPM IIITDM Jabalpur.

1.2 Purpose of the module

The purpose of the module is to showcase all available departments, faculty and resources available in this institute, along with the ability to make requests to the faculty and view various announcements made by the faculty and staff members.

The department view will display information regarding the department and facilities offered in the concerned department student details, faculty details, alumni details, and published announcements if any.

1.3 Scope of the module

The Department module will allow students to browse through relevant announcements and view details of faculty or students. This module also allows faculty or admin to make announcements related to either their course or announcements related to their department.

2. User/Actor characteristics

2.1 Student :

Students can do the following tasks:

- Browse announcements which will show announcements which will be filtered based on courses which the student is attending.

- View department details which will show details about the department and its resources available like labs and equipment.

- View faculty details which will display a list of faculties which will contain a link to their information.

- View alumni details which will display details related to alumni filtered based on their year of passing out.

- View student details which will contain a list of students filtered based on year and program.

2.2 Faculty :

The faculty can do the following tasks:

- Browse announcements which will show the announcements made by a faculty regarding a particular course or a particular department.

- View department details which will show details about the department and its resources available like labs and equipment.

- View faculty details which will display a list of faculties which will contain a link to their information.

- View alumni details which will display details related to alumni filtered based on the department and their year of passing out.

- View student details which will contain a list of students filtered based on year and program.

Specific Functionalities:

- Faculty can announce as per their needs. For Eg. If a faculty needs to make an announcement related to their courses this semester they can select the particular course or if they want to make an announcement for all students of the department they can do so.

2.3 Admin :

The faculty can do the following tasks:

- Browse announcements which will show the announcements made by a faculty regarding a particular course or a particular department.

- Make announcements which will allow the faculty to make announcements as per their requirements.

- View department details which will show details about the department and its resources available like labs and equipment.

- View faculty details which will display a list of faculties which will contain a link to their information.

- View alumni details which will display details related to alumni filtered based on the department and their year of passing out.

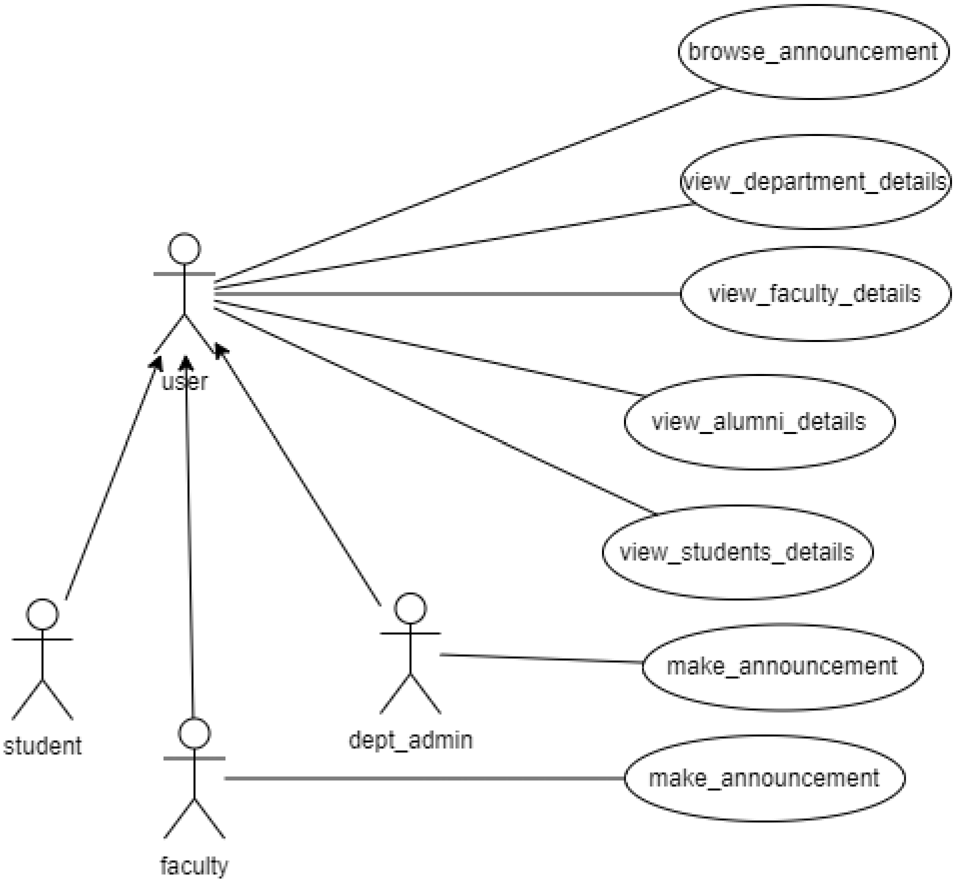
- View student details which will contain a list of students filtered on the basis of year and program.

Specific Functionalities:

- Admin can make an announcement as per their needs .

3 Functional Requirements

3.1 Use Case Diagram



3.2 Use Case Description

Use Case #1

UC ID	UC#1	
Use case name	Browse announcement	
Description	Users of the system can browse the available notifications/announcements as per the need of the user. Announcements will be fetched on the basis of the course in which the student is enrolled and only those announcements will be visible to a particular student.for a user which is student.	
Actor	Admin,Student,Faculty	
Precondition	The user must be logged in and the system is having the updated information of all the departments and announcements made.	
Main Flow	1.	Actor logs into the system and opens the Department Page.
	2.	Actor selects the Browse announcement tab.
Post condition	The necessary details of the department are displayed.	
Alternate Flow	nil	
Sub flow	nil	
Global Alternate flow	GA1	The actor can return to the dashboard at any time by clicking on the dashboard button.

Use Case #2

UC ID	UC#2	
Use case name	Make New announcement for the admin.	
Description	Admin can make new announcement related to department for student and Faculty related to that specific department	
Actor	Admin	
Precondition	Admin logged in ,opened Department module	
Main Flow	1.	Admin selects the “Make Announcement “ tab
	2.	Admin types the announcement and clicks publish.
Post condition	The announcement made by the admin is visible to the faculty and students under the “browse announcement “ option.	
Alternate Flow	nil	
Sub flow	nil	
Global Alternate flow	GA1	The actor can return to the dashboard at any time by clicking on the dashboard button.
		Post condition : the system returns to the actor’s dashboard.

Use case#3

UC ID	UC#3	
Use case name	make new announcement	
Description	Faculty can make new announcement related to department for student or other users related to that specific department	
Actor	Faculty	
Precondition	Faculty logged in ,opened Department module	
Main Flow	1.	Faculty selects the “Make Announcement “ tab
	2.	Faculty selects who he/she wants to make an announcement for i.e either M.Tech/B.Tech students or all students.
	3.	Faculty selects a year. Then the Faculty selects the course which the faculty wants to make an announcement for.
	4.	Faculty publishes the announcement.
Post condition	The announcement made by the faculty is visible to the admin and particular students under the “browse announcement “ option.	
Alternate Flow	nil	
Sub flow	nil	
Global Alternate flow	GA1	The actor can return to the dashboard at any time by clicking on the dashboard button.
		Post condition : the system returns to the actor’s dashboard.

Use case #4

UC ID	UC#4	
Use case name	view student detail	
Description	The actor can view the student registered in respective departments in the institute according to the batches.	
Actor	Student, Faculty, Admin	
Precondition	Actor logged in, opens the Department module and the system has updated information about the students.	
Main Flow	1.	Users select the respective department on the main page of the module.
	2.	Users clicks the students tab and selects the batches
	3.	Users clicks on the respective batch to view the details
Post condition	List of the student for the selected option are displayed	
Alternate Flow	nil	
Sub flow	nil	
Global Alternate flow	GA1	The actor can return to the dashboard at any time by clicking on the dashboard button.
		Post condition: the system returns to the actor's dashboard.

Use case #5

UC ID	UC#5	
Use case name	view faculty detail	
Description	The actor can view the faculty registered in respective departments in the institute according to the department.	
Actor	Student, Faculty, Admin	
Precondition	Actor logged in, opens the Department module and the system has updated information of the faculties.	
Main Flow	1.	Users select the respective department on the main page of the module.
	2.	Users select the “Faculty tab”.
	3.	Users clicks on view faculty to view the details
Post condition	List of the faculties for the selected option are displayed	
Alternate Flow	nil	
Sub flow	nil	
Global Alternate flow	GA1	The actor can return to the dashboard at any time by clicking on the dashboard button.
		Post condition: the system returns to actor’s dashboard.

Use case #6

UC ID	UC#6	
Use case name	view department details	
Description	The actor can view the faculty registered in respective departments in the institute according to the department.	
Actor	Student, Faculty, Admin	
Precondition	User logged in, opens the Department module and the system has updated information of the department.	
Main Flow	1.	User selects the respective department on the main page of the module.
Post condition	List of the department for the selected option are displayed	
Alternate Flow	nil	
Sub flow	nil	
Global Alternate flow	GA1	The actor can return to the dashboard at any time by clicking on the dashboard button.
		Post condition : the system returns to the actor’s dashboard.

Use case #7

UC ID	UC#7	
Use case name	View Alumni Details.	
Description	The actor can view the faculty registered in respective departments in the institute according to the department.	
Actor	Student, Faculty, Admin	
Precondition	User logged in, opens the Department module and the system has updated information of the department.	
Main Flow	1.	User selects the view Alumni tab on the main page of the module.
Post condition	List of the Alumni for the selected option are displayed	
Alternate Flow	nil	
Sub flow	nil	
Global Alternate flow	GA1	The actor can return to the dashboard at any time by clicking on the dashboard button.
		Post condition : the system returns to the actor's dashboard.

3.3 Other constraints

3.3.1 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 another. 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

3.3.2 Tech Stack Used

- **Django**: Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It follows the Model-View-Controller (MVC) architectural pattern.

- **Visual Studio Cod**e: A lightweight and versatile code editor commonly used for developing Flutter applications, which offers extensive Flutter and Dart extensions for enhanced development capabilities.

4. Non- Functional Requirements

4.1 Performance:

- The department module should provide quick response times for user interactions, such as department registration, student data modification, and notification retrieval.

- Response time for actions like department registration, student data updates, and notification retrieval should be within acceptable limits, with a maximum page load time of 3 seconds.

4.2 Scalability:

- The system should scale efficiently to accommodate a growing number of students and departments.

- Performance evaluations should be conducted under increasing load conditions to ensure the system remains responsive as the user base expands.

4.3 Availability:

- The department module should aim for a high availability rate, with a target of 99.9% uptime.

- Scheduled maintenance activities should be planned during non-peak hours to minimize impact on users.

4.4 Security:

- Data confidentiality and integrity should be maintained, with secure storage and transmission of sensitive information.

- Role-based authorization should be implemented to ensure that only authorized users, such as faculty members, can view and modify student data.

- Passwords should be securely hashed and stored to prevent unauthorized access.
- The department module should comply with data protection laws and ensure the privacy of student and faculty information.

4.5 Usability:

- The user interface of the department module should be designed to be intuitive and user-friendly, ensuring a positive user experience for both students and faculty.

4.6 Reliability:

- The system should have a high level of reliability, with minimal downtime for maintenance or unexpected issues.

5. Module dependencies with other fusion modules

5.1. UI Level

User logs into Fusion portion and lands on the main dashboard.The user (student,faculty,department admin) can click on the department button and go to the department page .From there students are able to view announcements regarding their department,courses and view student,faculty, and alumni details.
The faculty is able to make announcements and browse announcements from the department page.

5.2 DB Level :

S.No	Table Name	Primary Key	Foreign Key	owner	Reference Table
1.	department_announcements	id	maker_id_id	fusion_admin	globals_extrainfo
2.	department_specialrequest	id	request_maker_id	fusion_admin	globals_extrainfo

5.3 Module Level :

The module will use the notification module for notifying information about announcements.