

Fusion ERP
Software Requirements
Specification
For
AC-6 Department Module
(Mobile App)

Prepared By

Anoop Kumar(21BCS026)

Alok Kumar(21BCS018)

Ankit Singh(21BCS024)

Dharavath Vinod(21BCS076)

Vishal Parihar(21BCS244)

Student Mentor: Varun Sandeep Singh(21BCS237)

Faculty Mentor: Prof. Pritee Khanna

Table of Contents

1. Introduction.....	3
1.1 Introduction about the Fusion – A brief Description.....	3
1.2 Purpose of the module	4
1.3 Scope of the module.....	4
2. User/Actor Characteristics.....	5
2.1. Student.....	5
2.2. Faculty.....	5
2.3. Admin.....	6
3. Functional Requirements.....	7
3.1. Use case diagram.....	7
3.2. Use case description.....	8
3.3. Other constraints.....	15
3.3.1 User Interfaces.....	15
3.3.2 Software (Tech) Stack used.....	15
4. Non-Functional Requirements.....	16
4.1. Performance.....	16
4.2. Scalability.....	16
4.3. Availability.....	16
4.4. Security.....	16
5. Module dependencies with other modules.....	16
5.1. Integration at UI level.....	16
5.2. Database level dependencies.....	17
5.3. Module level dependencies.....	17

1. Introduction

1.1 Introduction about the Fusion – A brief Description

Embark on a transformative journey with the FusionIIIT Mobile App, a cutting-edge solution meticulously crafted using Flutter and Dart to enhance the operational landscape of PDPM Indian Institute of Information Technology, Design and Manufacturing, Jabalpur.

Developed by proactive students, this mobile app serves as a dynamic companion, seamlessly integrating and automating diverse functions across the campus.

Picture the FusionIIIT Mobile App as your digital ally, streamlining administrative tasks, elevating academic experiences, and ensuring the smooth functioning of various departments. Whether you're navigating complex paperwork on the admin side or seeking a digital touch for enhanced learning and course management, FusionIIIT Mobile App goes beyond the ordinary to foster a well-organized and enjoyable campus life.

Much like its web counterpart, the FusionIIIT Mobile App is not just a tool; it's a friendly companion that accompanies you through every facet of campus life. From simplifying administrative processes to enhancing academic efficiency, this app is designed to make your experience at PDPM IIITDM Jabalpur more organized and enjoyable. Embrace the future of campus management with FusionIIIT Mobile App – your key to a seamlessly integrated and automated campus experience.

1.2 Purpose of the module

This document outlines the specific requirements for the development of the "Department Module" within the Institute's Enterprise Resource Planning (ERP) system. The Department Module aims to establish a comprehensive platform, enabling all students at IIITDMJ to access detailed information about various departments and facilitate their registration into specific departments. Furthermore, it will ensure the timely dissemination of notifications regarding meetings, webinars, and essential announcements made by faculty members. The system will empower faculty members to efficiently access and modify student data, offering an automated and streamlined alternative to the current manual application procedures.

1.3 Scope of the module

The Department module is designed to efficiently view all details regarding students , faculty , department and alumni. Seamless interactions with faculty is facilitated through this module . It has three primary actors : Student , Faculty and Admin.

2. User/Actor characteristics

2.1 Student

A student of IIITDM Jabalpur who first logs into the system. He is able to browse through the details of the student, faculty, Alumni and the department

Role: View the details of students , faculty, department and alumni.

Specific Functionalities:

- View Student details
- View Faculty details
- View Department details
- View Alumni details
- Browse announcements made by faculty/administrator

2.2 Faculty

Faculty can view details of students,faculty,department and alumni.

Role: Make announcement related to academics ,internships or initiatives

Specific Functionalities:

- View Student details
- View Faculty details
- View Department details
- View Alumni details
- Make announcements
- Browse announcements

2.3 Admin

Head of department

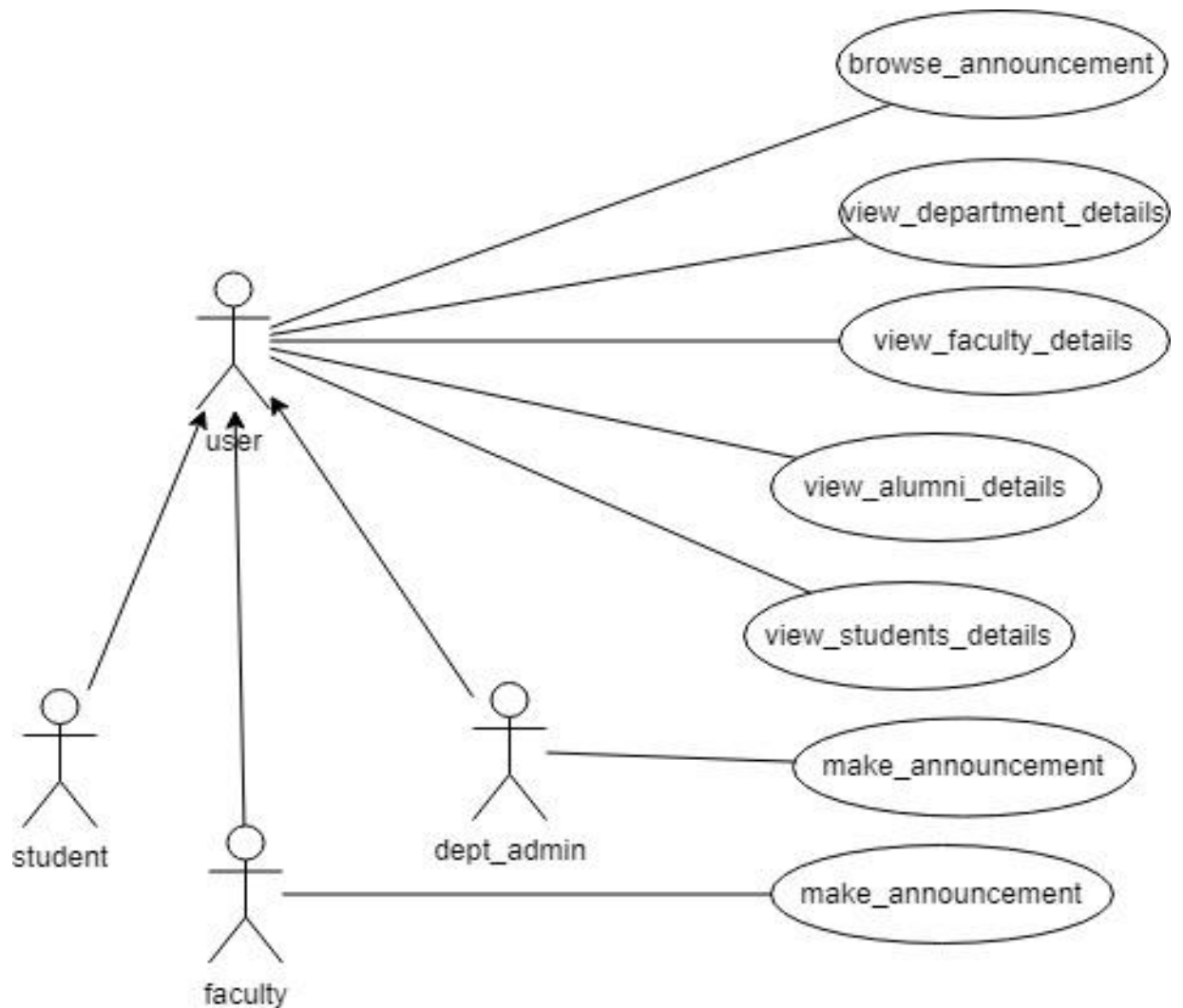
Role: Responsible for circulating notification / making announcement regarding meetings/webinar/conferences. Each Department has different admin and the respective admin can make the announcement for that department .

Specific Functionalities:

- View Student details
- View Faculty details
- View Department details
- View Alumni details
- Make announcements for the faculty and students
- Browse announcements

3. Functional Requirements

3.1 Use case diagram



3.2 Use case description

Use case #1

UC ID	UC#1	
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 Department module and system has updated information of the students.	
Main Flow	1.	Actor selects the respective department on the main page of the module.
	2.	Actor clicks the “Students” tab and selects the batches.
	3.	Actor 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 #2

UC ID	UC#2	
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 Department module and system has updated information of the faculties.	
Main Flow	1.	Actor selects the respective department on the main page of the module.
	2.	Actor selects the "Faculty tab".
	3.	Actor clicks on the 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 the actor's dashboard.

Use case #3

UC ID	UC#3	
Use case name	View _department_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 Department module and system has updated information of the department.	
Main Flow	1.	Actor 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 #4

UC ID	UC#4	
Use case name	View_alumni_detail	
Description	The actor can view the Alumni registered in respective departments of the institute according to the department.	
Actor	Student, Faculty, Admin	
Precondition	Actor logged in, opens Department module and system has updated information of the Alumni.	
Main Flow	1.	Actor selects the respective department on the main page of the module.
	2.	Actor selects the "Alumni tab".
	3.	Actor clicks on the view Alumni to view the details
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.

Use case #5

UC ID	UC#5	
Use case name	Browse_announcement	
Description	User of the system can browse the available notifications/announcement	
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 Department Page.
	2.	Actor selects the Browse announcement option
Post condition	The necessary details of the department are displayed . The system returns to the actor's dashboard	
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 #6

Use case description for Admin (extending User) as an actor.

UC ID	UC#6	
Use case name	make_announcement	
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
	4.	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 #7

Use case description for Faculty (extending User) as an actor.

UC ID	UC#7	
Use case name	make_announcement	
Description	Faculty can make new announcement related to department for student and admin 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 B.Tech/M.Tech students or all students.
	3.	Faculty selects a year .Then faculty selects the course which the faculty wants to make an announcement for.
	4.	Faculty types the announcement and clicks publish
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.
--	--	--

3.3 Other constraints

3.3.1 User Interface

- The application's interface will be developed using Flutter and Dart and is intended to be hosted on a mobile platform.
- The module will experience differing initial load times based on the strength of the mobile device's internet connection, influenced by the source media from which it is accessed.
- The mobile app's user interface should prioritize simplicity and ease of use in its design.

3.3.2 Software(Tech) Stack Used

- **Flutter SDK:** The official software development kit for building natively compiled applications for mobile, web, and desktop from a single codebase.
- **Dart Programming Language:** The language used to write Flutter applications, providing a reactive and component-based structure.
- **Flutter Framework:** The UI toolkit that facilitates the creation of natively compiled applications for mobile, web, and desktop from a single codebase.
- **Android Studio:** Popular integrated development environments (IDEs) for Flutter development, offering features like code completion, debugging, and more.
- **Visual Studio Code with Flutter Extension:** A lightweight and powerful code editor with extensions for Flutter development.
- **Firebase:** A mobile and web application development platform that provides various services such as authentication, cloud storage, and a real-time database.
- **Git:** A version control system commonly used to manage and track changes in the source code.
- **GitHub:** Platforms for hosting and collaborating on Git repositories.

4. Non-Functional Requirements

4.1 Performance

The system is required to exhibit prompt responsiveness to user interactions. Specifically, the response time for actions related to booking, inventory updates, and notifications is mandated to be minimized.

4.2 Scalability

The system must demonstrate the capability to accommodate a substantial volume of concurrent users. Evaluation of system performance should be conducted under conditions of increasing load.

4.3 Availability

The system is expected to maintain an availability rate of 99.9%, ensuring continuous operational functionality.

4.4 Security

Robust measures must be implemented to ensure the confidentiality and integrity of data. Role-based authorization mechanisms should be in place to restrict users to actions pertinent to their designated roles

5. Module dependencies with other modules

5.1 Integration at the UI level

Users log into the Fusion application and land on the main dashboard. The user (student, faculty, and other community members of IIITDMJ) can navigate to the department module . Navigation should include direct links or buttons in Fusion's menu, ensuring a clear path for each actor (Student, Faculty, Admin) to access their specific functionalities within the Department Module.

5.2 Database level dependencies

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 dependencies

Department registration Module interact with Notification, and Dashboard.

Faculty details are gathered from HR department and Alumni details are gathered from Training & Placement cell.

Student details from course registration .