**Fusion ERP**

**Software Requirements Specification**

**for**

**AC-2 Course Registration**

**Faculty Mentor:**

**Dr. Vinod Kumar Jain**

**Prepared by:**

**Aryan Sharma - 21BCS038  
 Bhavik Agarwal - 21BCS056**

**Deepanshu Singh - 21BCS073**

**Samyak Bhargava - 21BEC097**

**Vinayak Mittal - 21BEC124**

**Mentor :**

**Raman Chaudhary - 21BCS170**

**1. Introduction**

* 1. **Introduction about the Fusion – A brief Description**

FusionIIIT stands as a remarkable example of seamlessly integrating and automating diverse functions within PDPM Indian Institute of Information Technology, Design, and Manufacturing, Jabalpur. Developed meticulously using Python3 and driven by the Django Web framework, this student-led initiative aims to enhance the operational landscape of the institute. Covering a spectrum from efficient administration management to academic excellence and various departmental tasks, FusionIIIT stands out as a comprehensive solution that deals with all the complexities of campus life, from optimizing administrative processes to ensuring smoother academic journeys. FusionIIIT actively engages with various departments and sections, ensuring a seamless flow in every nook of campus life.

On the administrative side, FusionIIIT adeptly manages intricate paperwork and processes. In the academic sphere, it introduces a digital touch, simplifying learning and course management. Yet, FusionIIIT transcends its roles, it's more like a congenial companion, consistently contributing to the efficiency of life at PDPM IIITDM Jabalpur

To put it simply, FusionIIIT goes beyond being just a tool, it embodies a supportive companion, significantly contributing to a more organized and enjoyable campus life experience for everyone at PDPM IIITDM Jabalpur.

* 1. **Purpose of the module:**

The purpose of the Course Registration module is to streamline and enhance the overall user experience of the students allowing them to register courses seamlessly. The module seeks to automate all the course registration activities for Academic Section, students and faculties.

* 1. **Scope of the module**

Scope of this module includes complete course registration flow for students allowing them to add, drop, replace and view courses. The module also allows Acad admin to manage courses, collect fee payment and generates course list. The Course registration module is designed for scalability, optimisation and seamless integrations with other modules of the application.

**2. User/Actor characteristics**

**2.1 Student**

Represents individuals who intend to use this module for course registration, and pre-registration, add/drop/replace courses, view registration, backlog form etc. Students are divided on the basis of programme, discipline, year of joining, etc.

**Role:** Initiates the course registration process including replace/drop courses

**Specific Functionalities:**

* Submits personal information for pre-registration required for college administration
* Ability to view the offered course list for each semester along with the course details including credits, number of lectures, labs, etc.
* Generate course list for each semester based on the curriculum
* Select the courses/electives for the current semester assigning priority to each course available in course list
* Can see the details of the registration and the information entered by them
* Can apply register for backlog courses

**2.2 Academic Administration**

Supervises the overall operations of the pre-registration and course registration for each semester.

**Role:** Manages the overall operations of the course registration process, scheduling registration timings, generation course lists, updating academic curriculum, etc.

**Specific Functionalities:**

* Can add or remove courses from the list of available courses for the next semester.
* Update the faculty details who will be taking the course
* Schedule pre-registration and course registration for each semester
* Generate roll list after successful course registration process
* Configure pre-registration and main registration process

**2.3 Faculty**

Can view the courses assigned and and check the students enrolled in his/her course.

**Specific Functionalities:**

* Generate the roll list of the students enrolled in his/her course.
* View the courses assigned to him/her.

**3. Functional Requirements**

**3.1 Use Case Diagram**



**3.2 Use case Description  
i.**

| **UC ID** | UC#1 | | |
| --- | --- | --- | --- |
| **Use case Name** | Add/Drop Course | | |
| **Description** | The "Add/Drop " focuses on the addition/removal of courses in/from the course list of a UG/PG for the upcoming/ongoing semester. Students select the option of adding/dropping courses from a list of options and can add/drop the courses in/from his/her course list. | | |
| **Actor** | Student | | |
| **Precondition** | The student must be logged in into the system. | | |
| **Main Flow** | 1 | User selects the option of add/drop courses from a list of options. | |
| 2 | User is displayed with the list of courses he/she is enrolled in and the list of courses the user can add. | |
| 3 | User selects the course from the list he/she wants to add or remove. | |
| 4 | The user clicks on the submit button to submit the form. | |
| **Post conditions** | The course list of the user is updated and the user is removed/added from the attendance sheet of the added/removed course. | | |
| **Alternate Flow** | A1 | Users can not have less than 12 credits in a semester. | |
|  | A2 | No vacant seats available for the selected course . | |  |
|  | A3 | User can not have more than 26 credits in a semester | |  |
| **Sub Flow** | NIL | | |  |

**ii.**

| **UC ID** | UC#2 |
| --- | --- |
| **Use case Name** | View Offered Courses |
| **Description** | The “View Offered Courses” use case allows the student whether UG or PG to view the offered courses from the course list offered in the respective semester |
| **Actor** | Student |
| **Precondition** | The student is logged in into the system. |
| **Main Flow** | The Student navigates to the "View Courses" section. |
| The system displays the list of courses available for the respective semester. |
| The Student views the available courses in the list. |
| **Post conditions** | The student navigates to pre-registration to select the viewed courses. |
| **Alternate Flow** | NIL |
| **Sub Flow** | NIL |  |
| **Global Alternate Flow** | 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. |  |

**iii.**

| **UC ID** | UC#3 | | |
| --- | --- | --- | --- |
| **Use case Name** | Replace A Course | | |
| **Description** | The “Replace a course” use case allows the student to view the offered courses and replace the course in the respective semester with extra credits. | | |
| **Actor** | Student | | |
| **Precondition** | P1 | The student is logged in into the system. | |
| P2 | The acad admin has started the registration process. | |
| **Main Flow** | 1 | | The Student navigates to the "Replace Courses" section. |
| 2 | | The system displays the list of courses. |
| 3 | | The Student selects the courses from the list. |
| 4 | | The student submits the list and exits. |
| **Post conditions** | The updated booking information is reflected in the database. | | |
| **Alternate Flow** | A1 | | The course selected by the user does not have vacant seats. |
| **Sub Flow** | NIL | | |  |
| **Global Alternate Flow** | GA1 | | The acad admin can ‘cancel’ the replace course procedure at any time by exercising such an option and will be directed to the dashboard. |  |
| GA2 | | 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. |  |

**iv.**

| **UC ID** | UC#4 | |
| --- | --- | --- |
| **Use case Name** | Pre-Registration | |
| **Description** | The "Pre-Registration” use case allows students to select courses to enroll from the course list and submit the form to register for the next semester. | |
| **Actor** | Student | |
| **Precondition** | The student is logged in into the system and the acad admin has started the pre-registration process. | |
| **Main Flow** | 1 | The Student navigates to the "Pre-registration" section. |
| 2 | The system displays the form to select courses. |
| 3 | The Student selects the courses as per his/her interest and submits the form. A[1] |
| **Post conditions** | The acquired courses are reflected in the database and are reflected on the screen. | |
| **Alternate Flow** | A1 | The total credits required are not fulfilled and the system rejects the submission, requesting to select more courses. |
| **Sub Flow** | NIL | |  |
| **Global Alternate Flow** | GA1 | The Student can ‘cancel’ the procedure at any time by exercising such an option and will be directed to the dashboard. |  |
| GA2 | 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. |  |

**v.**

| **UC ID** | UC#5 | | |
| --- | --- | --- | --- |
| **Use case Name** | View-Registration | | |
| **Description** | The "View Registration" use case allows the student to review his/her selected courses through the Fusion portal. | | |
| **Actor** | Student | | |
| **Precondition** | The student is logged in into the system and registered courses. | | |
| **Main Flow** | 1 | The Caretaker navigates to the "Profile" section. | |
| 2 | The system displays the options to view courses of a semester. | |
| 3 | The Student selects a semester and clicks submit | |
| 4 | The Student’s registered courses will display on the page.[A1] | |
| **Post conditions** | The Student is able to view his registered courses. | | |
| **Alternate Flow** | A1 | 1 | The Student has not selected any courses for the selected semester. |
| A2 | 1 | The Course Registration window is still not open. |
| **Sub Flow** | NIL | | |  |
| **Global Alternate Flow** | GA1 | Courses List is not fetched from the database due to API error | |  |

**vi.**

| **UC ID** | UC#6 | | |
| --- | --- | --- | --- |
| **Use case Name** | Fill Backlog/Improvement Form | | |
| **Description** | The use case allows some specific users to fill the backlog/improvement form for the courses they want to repeat. | | |
| **Actor** | Student | | |
| **Precondition** | The user must be logged in to the system. | | |
| **Main Flow** | 1 | User navigates to the "Backlog/Improvement" section. | |
| 2 | The system displays the list of eligible courses that student can repeat. | |
| 3 | Student fill in the form for the course that he/she wants to repeat. | |
| 4 | Student click on the submit button to submit the form. | |
| **Post conditions** | Student is enrolled in the course he/she wants to repeat. | | |
| **Alternate Flow** | NIL | | |
| **Sub Flow** | NIL | | |  |
| **Global Alternate Flow** | GA1 | 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. | |  |

**vii.**

| **UC ID** | UC#7 | | |
| --- | --- | --- | --- |
| **Use case Name** | Final Registration | | |
| **Description** | The "Final Registration" use case allows the students to make final registrations and make payment. | | |
| **Actor** | Student | | |
| **Precondition** | The student is logged in into the system and the acad admin has configured main registration. | | |
| **Main Flow** | 1 | The student navigates to the registration section and then navigates to the payment section. | |
| 2 | The system displays payment amount and available payment methods for further process. | |
| 3 | The Student selects a payment method to make payment. | |
| 4 | The student makes the transaction from the selected payment gateway/method. [A1][A2] | |
| **Post conditions** | The updated registration and payment information is reflected in the database. | | |
| **Alternate Flow** | A1 | 1 | The payment is successful. |
| 2 | The system displays the success status and the fees receipt of the payment giving option to print it or save it. |
| A2 | 1 | The payment is unsuccessful and the system redirects the student to the dashboard displaying registration pending and payment not successful. |  |
|  |
|  |
| **Sub Flow** | NIL | | |  |
| **Global Alternate Flow** | GA1 | The Student can ‘cancel’ the procedure at any time by exercising such an option and will be directed to the dashboard. | |  |
| GA2 | 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. | |  |

**viii.**

| **UC ID** | UC#8 | | |
| --- | --- | --- | --- |
| **Use case Name** | Generate Roll List | | |
| **Description** | This use case”Generate Roll List” details the procedure for academic admin to generate the list of students eligible for the course registration process. | | |
| **Actor** | Acad Admin | | |
| **Precondition** | The user must be logged in into the system. | | |
| **Main Flow** | 1 | User checks the payment status of students and removes the names of the users from the roll list who left the Institute. | |
| 2 | User then generates the course list for the particular semester. | |
| **Post conditions** | The generated course list is made available to the students where they can view the generated course list. | | |
| **Alternate Flow** | NIL | | |  |
| **Sub Flow** | NIL | | |  |
| **Global Alternate Flow** | GA1 | 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. | |  |

**ix.**

| **UC ID** | UC#9 | | |
| --- | --- | --- | --- |
| **Use case Name** | Configure pre-registration | | |
| **Description** | The "Configure pre-registration" use case highlights the process by which the Acad admin can configure the pre registration process i.e. set the registration window, configure settings etc. | | |
| **Actor** | Acad Admin | | |
| **Precondition** | The Acad Admin must be logged in into the system. | | |
| **Main Flow** | 1 | The User navigates to the "Pre Registration" section. | |
| 2 | Admin choses the time he/she wants to open the registration window. | |
| 3 | Admin divides the pre registration window in slots for every specific batch. | |
| 4 | Admin finalizes the pre registration schedule and provides the details to students. | |
| **Post conditions** | The details of pre registration windows can now be viewed by students. | | |
| **Alternate Flow** | A1 | | The selected time slot for the pre registration window collides with any other allotted schedule. |
| **Sub Flow** | NIL | | |  |
| **Global Alternate Flow** | GA1 | 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. | |  |

**x.**

| **UC ID** | UC#10 | | |
| --- | --- | --- | --- |
| **Use case Name** | Generate Course List | | |
| **Description** | The Generate Course List use case allows the user to generate the course list for each course for each batch after the course registration process. | | |
| **Actor** | Acad Admin | | |
| **Precondition** | The acad admin should be logged into the system. | | |
| New academic year has started. | | |
| **Main Flow** | 1 | The admin selects the option to generate the course list. | |
| 2 | The admin selects the batch and course. | |
| 3 | The academic admin generates the list of enrolled students in the courses. | |
| **Post conditions** | The list of enrolled students should be generated in the excel format. | | |
| **Alternate Flow** | NIL | | |  |
| **Sub Flow** | NIL | | |  |

**xi.**

| **UC ID** | UC#11 | | |
| --- | --- | --- | --- |
| **Use case Name** | Add Courses | | |
| **Description** | The "Add Courses” use case allows acad admin to add courses to the offered course list for the upcoming semester. | | |
| **Actor** | Acad Admin | | |
| **Precondition** | The Acad admin is logged in into the system (as admin). | | |
| **Main Flow** | 1 | The Acad Admin navigates to the add courses section. | |
| 2 | The system displays the list of courses available to add in the course list. | |
| 3 | The Acad Admin selects course(s) to add in the list. [A1][A2] | |
| **Post conditions** | The updated course list is reflected in the database and displayed on the student’s screen. | | |
| **Alternate Flow** | A1 | 1 | The acad admin proceeds with the selected courses(s) and finally adds them to the list by submitting the form. |
| 2 | The system updates the offered course list for the upcoming semester. |
| A2 | 1 | The Acad admin decides to not proceed with the selected course to add and deselects it. |
|  |
|  |
| **Sub Flow** | NIL | | |  |
| **Global Alternate Flow** | GA1 | The Acad Admin can ‘cancel’ the procedure at any time by exercising such an option and will be directed to the add courses section. | |  |
| GA2 | 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. | |  |

**xii.**

| **UC ID** | UC#12 | | |
| --- | --- | --- | --- |
| **Use case Name** | Remove Courses | | |
| **Description** | The "Remove Courses" use case allows the acad admin to remove any course from the courses list. | | |
| **Actor** | Acad Admin | | |
| **Precondition** | The acad admin is logged in into the system. | | |
| **Main Flow** | 1 | The Acad admin navigates to the "Remove Courses" section. | |
| 2 | The system displays the list of all available courses | |
| 3 | The Acad admin selects the course to be deleted. [A1] | |
| 4 | The system shows confirmation to delete course. | |
| 4 | The acad admin confirms that they want to delete the selected course. [A2] | |
| **Post conditions** | The selected course has been deleted from the list of available courses. | | |
| **Alternate Flow** | A1 | 1 | The acad admin presses the back button. |
| A2 | 1 | The acad admin doesn’t confirm to delete the course. |
| **Sub Flow** | NIL | | |  |
| **Global Alternate Flow** | GA1 | The Acad Admin can ‘cancel’ the procedure at any time by exercising such an option and will be directed to the dashboard. | |  |
| GA2 | 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. | |  |

**xiii.**

| **UC ID** | UC#13 | | |
| --- | --- | --- | --- |
| **Use case Name** | Generate Roll List | | |
| **Description** | This use case”Generate Roll List” details the procedure for Faculty to generate the list of students eligible for the course registration process. | | |
| **Actor** | Faculty | | |
| **Precondition** | The user must be logged in into the system. | | |
| **Main Flow** | 1 | User checks the students enrolled in his.her course. | |
| **Post conditions** | The generated course list is made available to the students where they can view the generated course list. | | |
| **Alternate Flow** | NIL | | |  |
| **Sub Flow** | NIL | | |  |
| **Global Alternate Flow** | GA1 | 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. | |  |

**xiv.**

| **UC ID** | UC#14 | | |
| --- | --- | --- | --- |
| **Use case Name** | View Assigned Courses | | |
| **Description** | This use case allows the faculty to view the courses assigned to them. | | |
| **Actor** | Faculty | | |
| **Precondition** | The user must be logged in into the system. | | |
| **Main Flow** | 1 | User checks the course assigned to him/her. | |
| **Post conditions** | NIL | | |
| **Alternate Flow** | NIL | | |  |
| **Sub Flow** | NIL | | |  |
| **Global Alternate Flow** | GA1 | 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. | |  |

**3.4 Other constraints**

**3.4.1 User Interfaces**

The user interface should comply with the colour scheming and dashboard design of the FUSION IIIT. 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

**3.4.2 Technology Stack Used**

Django will be used for backend development using Python, and Flutter for the frontend, using Dart.

**3.4.3 Data Migration**

Existing data from the college database including the data of the students, courses etc needs to be migrated to Fusion database.

**3.4.3 Business rules**

1. Only registered students who have paid their fees should be eligible to register for the courses.

2. Each course has a capacity limit and thus students should not be allowed to register for a course once its capacity is free.

3. Course registration is allowed only during a pre-specified registration period and students should be allowed to register only during that window.

4. If there are any compulsory prerequisites for a course, then only those students who have completed those prerequisites should be allowed to register in that course.

**4. Non- Functional Requirements**

**4.1 Performance Requirements:**

The system should respond to user requests very fast and be able to handle 500 concurrent users during the registration. The system should also be scalable in order to handle an increase in the number of students and courses over time.

**4.2 Reliability and Availability:**

The system should be available during the critical registration periods. The system should be reliable and not fail during the registration process thus affecting the procedure.

**4.3 Security Requirements:**

Before using the app, the users must authenticate using a secure login mechanism. Role based access control should be implemented to ensure users can perform only those actions which they are authorized to.

**4.4 Compatibility:**

The system should be compatible with most of the popular operating systems (eg. Android, IOS).

**4.5 Maintainability:**

The source code should follow coding standards, be well documented in order to facilitate future enhancements and maintenance.

**5. Module dependencies with other fusion modules**

**5.1. UI Level**

The Course Registration module inFusion relies on the integration with other modules at the UI level.

UI Integration with -

* **Program and Curriculum (AC1):** Students and academic advisors will use this integration to ensure that registered courses align with the chosen program and curriculum.
* **Course Management (AC3):** Students will use this integration to make informed decisions during course selection.
* **Dashboards (GAD5):** Dashboard will be used to visual representation of the registered student data.

**5.2 DB Level Dependencies**

Course Registration module in Fusion at the DB Level Dependencies with other modules -

* **Program and Curriculum (AC1): T**he Course Registration module likely maintains a foreign key relationship with the Program and Curriculum database tables. This ensures that the selected courses align with the programs and curricula offered by the institution.
* **Course Management (AC3):** Regular synchronization processes between the Course Registration and Course Management databases to update and fetch course details, availability, and prerequisites.
* **Dashboards (GAD5):** The Dashboard module's database-level interactions involve managing and processing data from various sources to provide meaningful visual representations.
* **Department Module (AC6):**  The Department module will be an interaction with Course Registration for managing the roll list of all the students enrolled in the particular department.

**5.3. Module Level Dependencies**

* **Program and Curriculum (AC1):** The Course Registration module depends on the Program and Curriculum module to retrieve accurate information about available programs, associated courses, and curriculum details.
* **Course Management (AC3):** The Course Registration module relies on the Course Management module to integrate course data, ensuring that students have up-to-date information about course availability, prerequisites, and descriptions.
* **Other Academic Procedures (AC4):** The Course Registration module needs to align with academic procedures defined in the Other Academic Procedures module.