\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Software Requirement Specification**

**for**

# < AC-2-Admission, Registration, add-drop, fee deposit, course-list generation,Result >

**<Feb,2023>**

Team Members :-

Anunaya - 20BCS001

Abhijit Sarkar - 20BCS006

Aditya Yadav - 20BCS019

Sriram Gone - 20BCS090

Hemant Kumar - 20BCS100

Rohit Perala - 20BCS153

Shivam Patidar - 20BCS207

**1. Problem Description**

## 1.1 Purpose

Academic Procedure Management System (APMS) is a software designed to manage different activities related to the Academic Procedure of PDPM IIITDM Jabalpur. The software is designed to provide automated features to the Academic staff, students and faculties to handle different academic activities.

## 1.2 Intended Audience and Reading Suggestions

The audience for the document reading are users of the system such as academic persons, students and faculties. Developers of the software will use it as a basis for making design and coding.

## 1.3 Product Scope

Although academic affairs consist of many activities but this software product will have a limited scope as some of activities are taken care by other software. This software will take care of the following activities:

Admission, Registration, add-drop, fee deposit, course-list generation, result generation.

**2. User (Actor) Characteristics**

## 2.1 Student

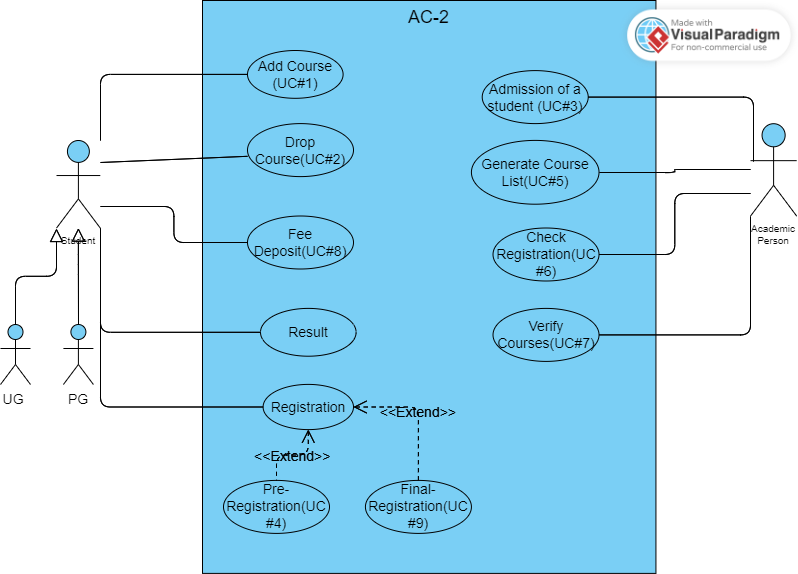
The user will use this module for academic registration, pre-registration, add/drop course and other use cases. Students can be further divided into UG and PG students. PhD and M.Tech students falls under Pg Students category.

## 2.2 Academic Person

The user who will use this module for student related data processing for example –

Student information, student registration information, sending application, etc.

**3.Use-case Diagram**



Link -

https://online.visual-paradigm.com/w/rpcgjfqo/diagrams/#diagram:workspace=rpcgjfqo&proj=0&id=2

**4.Use-case Documentation**

## 4.1 Use Case #1

| **Use Case ID** | **UC#1** |
| --- | --- |
| **Use Case Name** | **Add a course** |
| **Description** | The use case describes how to add courses to the course list of a UG student. |
| **Actor(s)** | UG Student |
| **Precondition** | UG Students must be logged in the system. |
| **Main Flow** | 1.UG Student selects the option of adding courses from a list of options. |
| 2. UG Student is displayed a list of courses he/she can add in his course list. |
| 3. UG Student chooses a course from a list of courses which he/she wants to add. |
| 4. UG student submits the form. |
| **Post-condition** | 1.Successful completion |
|  | 2. Course is added to the course list of UG students. |
| 3. UG Student is added to the attendance sheet of added courses. |
| **Alternate flow** | 1. Constraints not met |
| 2. UG Student is prompted a message about constraints not met due to either of following conditions: |
| 3. UG Students with CPI less than 8.0 are not eligible. |
| 4. UG Students cannot add more than two courses in a semester. |
| 5. UG Students cannot have more than 26 credits in a semester. |
| 6. UG Student is redirected to the form. |
| **Sub Flow** | NIL |

## 4.2 Use Case #2

| **Use Case ID** | **UC#2** |
| --- | --- |
| **Use Case Name** | **Drop a course** |
| **Description** | The use case describes how to drop courses from the course list of a UG student. |
| **Actor(s)** | UG Student |
| **Precondition** | UG Students must be logged in the system. |
| **Main Flow** | 1 .UG Student selects the option of dropping course from a list of options. |
| 2. UG Student is displayed on his/her course list. |
| 3.UG Student chooses the course from the list which he/she wants to drop. |
| 4. UG student submits the form. |
| **Post-condition** | 1.Successful completion |
| 2. Course/s is/are removed from the course list of UG students. |
| 3. UG Student is removed from the attendance sheet of removed courses. |
| **Alternate flow** | 1. Constraints not met |
| 2. UG Students cannot drop more than two courses in a semester. |
| 3. UG students cannot have less than 12 credits in a semester. |
| **Sub Flow** | NIL |

## 4.3 Use Case #3

| **Use Case ID** | **UC#3** |
| --- | --- |
| **Use Case Name** | **Admission of a student** |
| **Description** | This use describes how the academic person uploads details of a new student for admission. |
| **Actor(s)** | Academic person |
| **Precondition** | Academic person must be logged in the system. |
| **Main Flow** | 1. Academic person selects the manage student profile option. |
| 2. Academic person uploads the verified student details. |
| **Post Condition** | 1.Admission form submitted Successfully. |
|  | 2. Student registered in the database. |
| **Alternate flow** | 1. Failure in form submission. |
| 2. Unverified student details. |
| **Sub Flow** | NIL |

## 4.4 Use Case #4

| **Use Case ID** | **UC#4** |
| --- | --- |
| **Use Case Name** | **Pre Registration** |
| **Description** | The use case handles the student registration system. |
| **Actor(s)** | Student |
| **Precondition** | 1.Student must be logged in the system. |
| 2.Pre-Registration form which includes courses of the next semester is provided to the students. |
| **Main Flow** | 1. Students select the courses to enroll for next semester. |
| 2. Students click the Register button to register. |
| **Post-condition** | Pre-registration done successfully. |
| **Alternate flow** | Return back to the same page with blank form. |
| **Sub Flow** | NIL |

## 4.5 Use Case #5

| **Use Case ID** | **UC#5** |
| --- | --- |
| **Use Case Name** | **Generate course list** |
| **Description** | The use case describes how the Academic person generate the course list. |
| **Actor(s)** | Academic person |
| **Precondition** | 1. Academic person must be logged in the system. |
| 2. Academic year has started. |
| **Main Flow** | 1.Academic person selects the option to generate the course list. |
| 2. Academic person selects the course. |
| 3. Academic person selects the batch. |
| 4.Academic person generate enrolled student excel sheet. |
| **Post-condition** | List generated successfully. |
| **Alternate flow** | NIL |
| **Sub Flow** | NIL |

## 4.6 Use Case #6

| **Use Case ID** | **UC#6** |
| --- | --- |
| **Use Case Name** | **Checks\_registration** |
| **Description** | This use case describes how the Academic Person checks the final registration of students for the semester |
| **Actor(s)** | Academic Person |
| **Precondition** | 1. Academic Person must be logged in the system. |
| **Main Flow** | 1.Academic Person selects the option of Checking Registrations of all students. |
| **Post-condition** | 1.Student can check the updated registered courses through his link. |
| **Alternate flow** | NIL |
| **Sub Flow** | NIL |

## 4.7 Use Case #7

| **Use Case ID** | **UC#7** |
| --- | --- |
| **Use Case Name** | **verify\_added\_courses** |
| **Description** | This use case describes how the Academic Person checks the registered courses of each student after the last date of adding dropping courses. |
| **Actor(s)** | Academic Person |
| **Precondition** | 1. Academic Person must be logged in the system. |
| **Main Flow** | 1.Academic Person selects the option of Checking Registrations of all students. |
| 2.He may drop the courses and add courses according to his will if the courses doesn’t meet the CPI requirements. |
| **Post-condition** | 1.Student can check the updated registered courses through his link. |
| **Alternate flow** | NIL |
| **Sub Flow** | NIL |

## 4.8 Use Case #8

| **Use Case ID** | **UC#8** |
| --- | --- |
| **Use Case Name** | **Fee\_deposit** |
| **Description** | This use case describes how a student deposits a fee for the next semester. |
| **Actor(s)** | Student |
| **Precondition** | 1. Students must be logged in the system. |
| **Main Flow** | 1. Students select the option of fee deposit. |
| 2. Students redirect to the fee payment portal. |
| **Post-condition** | 1. Submission successful. |
| **Alternate flow** | 1. Failure in payment. |
| **Sub Flow** | NIL |

## 4.9 Use Case #9

| **Use Case ID** | **UC#9** |
| --- | --- |
| **Use Case Name** | **Final Registration** |
| **Description** | The use case handles the student registration system. |
| **Actor(s)** | Student |
| **Precondition** | 1.Student must be logged in the system. |
| 2.Final-Registration form which includes courses of the next semester is provided to the students. |
| **Main Flow** | 1. Students confirm the courses to enroll for next semester. |
| 2. Student selects the fee payment mode and enters the transaction number. |
| 3. Students click the Register button to register. |
| **Post-condition** | None |
| **Alternate flow** | Students can change the opted course. |
| **Sub Flow** | NIL |

• **Software Quality Attributes**

## •Usability

* **User Interface**: The system shall allow a user to interface with it through mouse/keyboard events on text-fields, check-boxes, buttons and drop down boxes. Also, the user interface should be such that the system is easy to use.
* **Compatibility**: The user should be able to properly view the web-site in most of the available browsers (Internet Explorer, Firefox and Safari).
* **Validations**: The user should be flagged appropriate warnings/errors upon invalid input or operations, with specific reference to the error.

## •Correctness

•Data content in files sent by one employee to another employee must be correct. Incorrect leads to failure of the system.

## •Robustness

•Interaction between any two employees should not impact on interaction of other employees.