



**Department of Computer Science  
Namal University, Mianwali**

# **Software Requirements Specification (SRS)**

## **Final Year Project Management System**

**Course Title:** Software Engineering

**Team Members:**

Name	Roll Number	Email
Mubashir Hassan	NUM-BSCS-2024-38	bscs24f38@namal.edu.pk
Shumaila Sarfaraz	NUM-BSCS-2024-74	bscs24f74@namal.edu.pk
Sadia	NUM-BSCS-2024-68	bscs24f68@namal.edu.pk

**Submission Date:** December 28, 2025

**Submitted To:**  
Miss Asiya Batool  
Lecturer  
Department of Computer Science

## Contents

<b>1</b>	<b>Introduction</b>	<b>14</b>
1.1	Purpose . . . . .	14
1.1.1	Purpose of the SRS . . . . .	14
1.1.2	Purpose of the System . . . . .	14
1.1.3	Audience of SRS . . . . .	14
1.2	Scope . . . . .	15
1.2.1	Software Product . . . . .	15
1.2.2	Product Features . . . . .	15
1.2.3	Application of the System . . . . .	16
1.2.4	Benefits of the System . . . . .	16
1.3	Definitions, Acronyms, and Abbreviations . . . . .	16
1.4	References . . . . .	16
1.5	Overview . . . . .	17
1.5.1	Content of the SRS . . . . .	17
1.5.2	Organization of the SRS . . . . .	17
<b>2</b>	<b>Overall Description</b>	<b>17</b>
2.1	Product Perspective . . . . .	17
2.1.1	System Context . . . . .	17
2.1.2	Relationship with Other Systems . . . . .	18
2.1.3	Major System Modules . . . . .	18
2.1.4	External Interfaces . . . . .	18
2.1.5	Hardware and Operating Environment . . . . .	19
2.2	Product Functions . . . . .	19
2.3	User Characteristics . . . . .	20
2.3.1	Final Year Students . . . . .	20
2.3.2	Supervisors and Co-supervisors . . . . .	20
2.3.3	Coordinators and Heads of Department . . . . .	20
2.3.4	Industrial Mentors . . . . .	20
2.4	Internal and External Evaluators . . . . .	20
2.4.1	Placement Cell Department . . . . .	21
2.4.2	Super Administrator . . . . .	21
2.5	General Constraints . . . . .	21
2.5.1	Regulatory and Academic Policies . . . . .	21
2.5.2	Hardware and Infrastructure Limitations . . . . .	21
2.5.3	Interfaces with Other Systems . . . . .	21

2.5.4	Parallel Operations . . . . .	21
2.5.5	Programming and Platform Constraints . . . . .	22
2.5.6	Network Communication . . . . .	22
2.5.7	Audit Functions . . . . .	22
2.5.8	Control Functions . . . . .	22
2.5.9	Criticality of the Application . . . . .	22
2.5.10	Safety and Security . . . . .	22
2.6	Assumptions and Dependencies . . . . .	22
2.6.1	User and Device Assumptions . . . . .	23
2.6.2	Network and Connectivity Assumptions . . . . .	23
2.6.3	Hosting and Infrastructure Dependencies . . . . .	23
2.6.4	Access Assumption . . . . .	23
2.6.5	Security and Data Protection Dependencies . . . . .	23
2.6.6	API and External Service Dependencies . . . . .	23
2.6.7	Future Enhancements and Scalability Assumptions . . . . .	23
<b>3</b>	<b>Specific Requirements</b>	<b>23</b>
3.1	Similar Functional Requirements for each user . . . . .	24
3.1.1	Requirement 1: Login . . . . .	24
Introduction	. . . . .	24
Input	. . . . .	24
Processing	. . . . .	24
Outputs	. . . . .	24
Performance Requirements	. . . . .	25
Design Constraints	. . . . .	25
Attributes	. . . . .	25
3.1.2	Requirement 2: Reset or Change Password . . . . .	25
Introduction	. . . . .	25
Input	. . . . .	25
Processing	. . . . .	25
Outputs	. . . . .	26
Performance Requirements	. . . . .	26
Design Constraints	. . . . .	26
Attributes	. . . . .	26
3.1.3	Requirement 3: View and Update Personal Profile . . . . .	26
Introduction	. . . . .	26
Processing	. . . . .	27
Outputs	. . . . .	27
Performance Requirements	. . . . .	27

Design Constraints . . . . .	27
Attributes . . . . .	27
3.1.4 Requirement 4: Receive System Notifications and Announcements	28
Introduction . . . . .	28
Processing . . . . .	28
Outputs . . . . .	28
Performance Requirements . . . . .	28
Design Constraints . . . . .	29
Attributes . . . . .	29
3.2 Student Module . . . . .	29
3.2.1 Requirement 1: Form a New Project Group or Join an Existing Group	29
Introduction . . . . .	29
Input . . . . .	29
Processing . . . . .	29
Outputs . . . . .	30
Performance Requirements . . . . .	30
Design Constraints . . . . .	30
Attributes . . . . .	30
3.2.2 Requirement 2: Send,Accept or Reject Group Invitations . . . . .	30
Introduction . . . . .	30
Input . . . . .	30
Processing . . . . .	31
Outputs . . . . .	31
Performance Requirements . . . . .	31
Design Constraints . . . . .	31
Attributes . . . . .	31
3.2.3 Requirement 3:Browse and Search Available Projects . . . . .	32
Introduction . . . . .	32
Input . . . . .	32
Processing . . . . .	32
Outputs . . . . .	32
Performance Requirements . . . . .	32
Design Constraints . . . . .	33
Attributes . . . . .	33
3.3 Requirement 4: View past Project Description . . . . .	33
Introduction . . . . .	33
Input . . . . .	33
Processing . . . . .	33

Outputs . . . . .	33
Performance Requirements . . . . .	34
Design Constraints . . . . .	34
Attributes . . . . .	34
3.3.1 Requirement 5: Propose New Project Ideas to Supervisors . . . . .	34
Introduction . . . . .	34
Input . . . . .	34
Processing . . . . .	34
Outputs . . . . .	35
Performance Requirements . . . . .	35
Design Constraints . . . . .	35
Attributes . . . . .	35
3.3.2 Requirement 6: Submit Project Proposals and Revisions to Supervisors . . . . .	35
Introduction . . . . .	35
Input . . . . .	36
Processing . . . . .	36
Outputs . . . . .	36
Performance Requirements . . . . .	36
Design Constraints . . . . .	37
Attributes . . . . .	37
3.4 Requirement 7: Check Proposal Approval Status . . . . .	37
Introduction . . . . .	37
Input . . . . .	37
Processing . . . . .	37
Outputs . . . . .	38
Performance Requirements . . . . .	38
Design Constraints . . . . .	38
Attributes . . . . .	38
3.4.1 Requirement 8: Upload Project Documents . . . . .	38
Introduction . . . . .	38
Input . . . . .	38
Processing . . . . .	39
Outputs . . . . .	39
Performance Requirements . . . . .	39
Design Constraints . . . . .	39
Attributes . . . . .	39
3.4.2 Requirement 9: Replace Documents Within Submission Deadlines	40
Introduction . . . . .	40

Input . . . . .	40
Processing . . . . .	40
Outputs . . . . .	41
Performance Requirements . . . . .	41
Design Constraints . . . . .	41
Attributes . . . . .	41
3.5 Requirement 10: View Project Milestone Deadlines . . . . .	41
Introduction . . . . .	41
Input . . . . .	41
Processing . . . . .	42
Outputs . . . . .	42
Performance Requirements . . . . .	42
Design Constraints . . . . .	42
Attributes . . . . .	42
3.5.1 Requirement 11: View Comments from Supervisors, Co-Supervisors and Industrial Mentors . . . . .	43
Introduction . . . . .	43
Input . . . . .	43
Processing . . . . .	43
Outputs . . . . .	43
Performance Requirements . . . . .	44
Design Constraints . . . . .	44
Attributes . . . . .	44
3.6 Requirement 12: View Evaluation Marks . . . . .	44
Introduction . . . . .	44
Input . . . . .	44
Processing . . . . .	44
Outputs . . . . .	45
Performance Requirements . . . . .	45
Design Constraints . . . . .	45
Attributes . . . . .	45
Requirement 13: Receive System Notifications and Announcements	45
3.6.1 Introduction . . . . .	46
Processing . . . . .	46
Outputs . . . . .	46
Performance Requirements . . . . .	46
Design Constraints . . . . .	47
Attributes . . . . .	47

3.6.2	Requirement 14: Schedule Meetings with Supervisors, Co-Supervisors, and Industrial Mentors . . . . .	47
	Introduction . . . . .	47
	Input . . . . .	47
	Processing . . . . .	47
	Outputs . . . . .	48
	Performance Requirements . . . . .	48
	Design Constraints . . . . .	48
	Attributes . . . . .	48
3.7	Supervisor Module . . . . .	48
3.7.1	Requirement 1: View Assigned Project Groups . . . . .	48
	Introduction . . . . .	48
	Input . . . . .	48
	Processing . . . . .	49
	Outputs . . . . .	49
	Performance Requirements . . . . .	49
	Design Constraints . . . . .	49
	Attributes . . . . .	50
3.8	Requirement 2: Review and Approve, Reject, or Request Revisions for Project Documents . . . . .	50
	Introduction . . . . .	50
	Processing . . . . .	50
	Outputs . . . . .	51
	Performance Requirements . . . . .	51
	Design Constraints . . . . .	51
	Attributes . . . . .	51
3.8.1	Requirement 3: Provide Comments on Submissions . . . . .	51
	Introduction . . . . .	51
	Input . . . . .	52
	Processing . . . . .	52
	Outputs . . . . .	52
	Performance Requirements . . . . .	52
	Design Constraints . . . . .	53
	Attributes . . . . .	53
3.8.2	Requirement 4: Schedule Meetings with Students . . . . .	53
	Introduction . . . . .	53
	Input . . . . .	53
	Processing . . . . .	53
	Outputs . . . . .	54

Performance Requirements . . . . .	54
Design Constraints . . . . .	54
Attributes . . . . .	54
3.9 Co-Supervisor Module . . . . .	55
3.9.1 Requirement 1: View Assigned Projects and Student Groups . . . . .	55
Introduction . . . . .	55
Input . . . . .	55
Processing . . . . .	55
Outputs . . . . .	56
Performance Requirements . . . . .	56
Design Constraints . . . . .	56
Attributes . . . . .	56
3.9.2 Requirement 2: Access Student Submissions . . . . .	56
Introduction . . . . .	56
Input . . . . .	56
Processing . . . . .	57
Outputs . . . . .	57
Performance Requirements . . . . .	57
Design Constraints . . . . .	58
Attributes . . . . .	58
3.9.3 Requirement 3: Provide Feedback and Suggestions on Submissions	58
Introduction . . . . .	58
Input . . . . .	58
Processing . . . . .	58
Outputs . . . . .	59
Performance Requirements . . . . .	59
Design Constraints . . . . .	59
Attributes . . . . .	60
3.9.4 Requirement 4: Schedule Meetings with Project Groups . . . . .	60
Introduction . . . . .	60
Input . . . . .	60
Processing . . . . .	60
Outputs . . . . .	61
Performance Requirements . . . . .	61
Design Constraints . . . . .	61
Attributes . . . . .	61
3.10 Head of Department Module . . . . .	62
3.10.1 Requirement 1: View All FYP Projects of the Department . . . . .	62
Introduction . . . . .	62

Input . . . . .	62
Processing . . . . .	62
Outputs . . . . .	63
Performance Requirements . . . . .	63
Design Constraints . . . . .	63
Attributes . . . . .	63
3.10.2 Requirement 2: View and Approve Supervisor and Co-Supervisor Assignments . . . . .	63
Introduction . . . . .	63
Input . . . . .	64
Processing . . . . .	64
Outputs . . . . .	64
Performance Requirements . . . . .	65
Design Constraints . . . . .	65
Attributes . . . . .	65
3.10.3 Requirement 3:Approve or Reject Projects . . . . .	65
Introduction . . . . .	65
Input . . . . .	65
Processing . . . . .	66
Outputs . . . . .	66
Performance Requirements . . . . .	66
Design Constraints . . . . .	66
Attributes . . . . .	67
3.11 Industrial Mentors Module . . . . .	67
3.11.1 Requirement 1: View Assigned Student Projects . . . . .	67
Introduction . . . . .	67
Inputs . . . . .	67
Processing . . . . .	67
Outputs . . . . .	68
Performance Requirements . . . . .	68
Design Constraints . . . . .	68
Attributes . . . . .	68
3.11.2 Requirement 2: Review Project Documents . . . . .	68
Introduction . . . . .	68
Inputs . . . . .	69
Processing . . . . .	69
Outputs . . . . .	69
Performance Requirements . . . . .	70
Design Constraints . . . . .	70

Attributes . . . . .	70
3.11.3 Requirement 3: Provide Comments on Projects . . . . .	70
Introduction . . . . .	70
Inputs . . . . .	70
Processing . . . . .	70
Outputs . . . . .	71
Performance Requirements . . . . .	71
Attributes . . . . .	71
3.11.4 Requirement 4: Schedule Meeting with Project Group . . . . .	72
Introduction . . . . .	72
Inputs . . . . .	72
Processing . . . . .	72
Outputs . . . . .	73
Performance Requirements . . . . .	73
Attributes . . . . .	73
3.12 Placement Cell Department Module . . . . .	73
3.12.1 Requirement 1: View Industry-Based Projects . . . . .	73
Introduction . . . . .	73
Inputs . . . . .	73
Processing . . . . .	73
Outputs . . . . .	74
Performance Requirements . . . . .	74
Design Constraints . . . . .	75
Attributes . . . . .	75
3.12.2 Requirement 2: View Engaged Industrial Mentor Records . . . . .	75
Introduction . . . . .	75
Inputs . . . . .	75
Processing . . . . .	75
Outputs . . . . .	76
Performance Requirements . . . . .	76
Design Constraints . . . . .	77
Attributes . . . . .	77
3.13 Requirement 3: Track Groups Who Work on Industrial Projects . . . . .	77
Introduction . . . . .	77
Inputs . . . . .	77
Processing . . . . .	77
Outputs . . . . .	78
Performance Requirements . . . . .	79
Design Constraints . . . . .	79

Attributes . . . . .	79
3.14 External and Internal (faculty) Evaluators Module . . . . .	79
3.14.1 Requirement 1: View Assigned Projects for Evaluation . . . . .	79
Introduction . . . . .	79
Inputs . . . . .	79
Processing . . . . .	80
Outputs . . . . .	80
Performance Requirements . . . . .	80
Design Constraints . . . . .	80
Attributes . . . . .	81
3.14.2 Requirement 2: Access Project's Documents . . . . .	81
Introduction . . . . .	81
Inputs . . . . .	81
Processing . . . . .	81
Outputs . . . . .	82
Performance Requirements . . . . .	82
Design Constraints . . . . .	82
Attributes . . . . .	82
3.14.3 Requirement 3: View Evaluation Deadlines and Schedules . . . . .	82
Introduction . . . . .	82
Inputs . . . . .	82
Processing . . . . .	83
Outputs . . . . .	83
Performance Requirements . . . . .	83
Design Constraints . . . . .	84
Attributes . . . . .	84
3.14.4 Requirement 4: Submit Evaluation Marks and Remarks . . . . .	84
Introduction . . . . .	84
Inputs . . . . .	84
Processing . . . . .	84
Outputs . . . . .	85
Performance Requirements . . . . .	85
Design Constraints . . . . .	85
Attributes . . . . .	85
3.14.5 Requirement 5: View Evaluation Criteria and Instructions . . . . .	86
Introduction . . . . .	86
Inputs . . . . .	86
Processing . . . . .	86
Outputs . . . . .	87

Performance Requirements . . . . .	87
Design Constraints . . . . .	87
Attributes . . . . .	87
3.15 Coordinator Module . . . . .	87
3.15.1 Requirement 1: Manage FYP Timelines of Milestones . . . . .	87
Introduction . . . . .	87
Inputs . . . . .	87
Processing . . . . .	88
Outputs . . . . .	88
Performance Requirements . . . . .	88
Design Constraints . . . . .	88
Attributes . . . . .	88
3.15.2 Requirement 2: Assign evaluators and evaluation schedules . . . . .	89
Introduction . . . . .	89
Inputs . . . . .	89
Processing . . . . .	89
Outputs . . . . .	89
Performance Requirements . . . . .	90
Design Constraints . . . . .	90
Attributes . . . . .	90
3.15.3 Requirement 3: Assign supervisors and co-supervisors . . . . .	90
Introduction . . . . .	90
Inputs . . . . .	90
Processing . . . . .	91
Outputs . . . . .	91
Performance Requirements . . . . .	91
Design Constraints . . . . .	91
Attributes . . . . .	92
3.15.4 Requirement 4: Handle Issues of All Users related to FYP . . . . .	92
Introduction . . . . .	92
Inputs . . . . .	92
Processing . . . . .	92
Outputs . . . . .	93
Performance Requirements . . . . .	93
Design Constraints . . . . .	93
Attributes . . . . .	93
3.16 Super Administrator Module . . . . .	93
3.16.1 Requirement 1: Full Login System Access . . . . .	93
Introduction . . . . .	93

Inputs . . . . .	94
Processing . . . . .	94
Outputs . . . . .	94
Performance Requirements . . . . .	94
Design Constraints . . . . .	95
Attributes . . . . .	95
3.16.2 Requirement 2: Manage All user Roles and Permission . . . . .	95
Introduction . . . . .	95
Inputs . . . . .	95
Processing . . . . .	95
Outputs . . . . .	96
Performance Requirements . . . . .	96
Design Constraints . . . . .	96
Attributes . . . . .	96
3.16.3 Requirement 3: Create, Update or Delete users . . . . .	96
Introduction . . . . .	96
Inputs . . . . .	97
Processing . . . . .	97
Outputs . . . . .	97
Performance Requirements . . . . .	97
Design Constraints . . . . .	98
Attributes . . . . .	98
3.17 Non-Functional Requirements . . . . .	98
3.17.1 Usability Requirements . . . . .	98
User-Friendly . . . . .	98
Accessibility . . . . .	98
3.17.2 Reliability Requirements . . . . .	99
Availability . . . . .	99
Data Integrity . . . . .	99
Backup and Recovery . . . . .	99
3.17.3 Performance Requirements . . . . .	99
Response Time . . . . .	99
Concurrent User Support . . . . .	99
Data Throughput . . . . .	100
Peak Load Handling . . . . .	100
3.18 Scalability Requirements . . . . .	100
Enhancement of System . . . . .	100
Modular Design . . . . .	100
3.18.1 Security Requirements . . . . .	100

Authentication and Authorization . . . . .	100
Data Protection . . . . .	100
Privacy and Compliance . . . . .	100
<b>4 External Interface Requirements</b>	<b>101</b>
4.1 User Interfaces . . . . .	101
4.1.1 Performance Requirements . . . . .	101
4.1.2 Design Constraints . . . . .	101
Standards . . . . .	101
Hardware Compliance . . . . .	101
Limitations . . . . .	101
4.1.3 Attributes . . . . .	101
Security . . . . .	101
Maintainability . . . . .	101
4.1.4 Other Requirements . . . . .	102
Database . . . . .	102
Operations . . . . .	102
Site Adaptation . . . . .	102
4.2 Hardware Interfaces . . . . .	102
4.3 Software Interfaces . . . . .	102
4.4 Communications Interfaces . . . . .	102
<b>A Appendix A: Use Case Diagram</b>	<b>103</b>
<b>B Appendix B: Context Diagram</b>	<b>104</b>

# 1 Introduction

## 1.1 Purpose

### 1.1.1 Purpose of the SRS

The purpose of this Software Requirements Specification (SRS) is to provide a complete, clear and unambiguous detail of the requirements for the **Final Year Project Management System**. This document shows the system's functional and non functional behavior, operational characteristics, performance requirements and constraints of design. The SRS act as path between the stakeholders and the development team by making a common knowledge of system to implementation. It serves as reference from beginning of project to maintenance.

### 1.1.2 Purpose of the System

The purpose of the FYP Management System is to make the flow of FYP management activities from manual to automate and centralize activities within the university. This system replace manual and spreadsheet based task to make tasks secure, efficient, minimizes redundancy and scalable software and digital platform. This system aims to complete support of FYP tasks including formation of group, project proposal to supervisors, evaluation schedule and final project submission. By automating these tasks, the system increase the transparency, reduce the administrative work and improves coordination among users.

### 1.1.3 Audience of SRS

This SRS document is used for stakeholders that involved in the Final Year Project processes. Following are include:

1. Students
2. Supervisors
3. Co-supervisors
4. Head of the Departments
5. Industrial Mentors
6. Placement Cell
7. Super Admin

8. Department Faculty
9. External Evaluators
10. Coordinators
11. Development team ( Project Manager, developers, Testers )

## **1.2 Scope**

The FYP Management System is a digital platform and software application that make the process of the FYP to automate form and handle the entire lifecycle of final year project at Namal University. This system replace the current excel sheet and manual based processes with digital automate system.

### **1.2.1 Software Product**

This software product “ FYP Management System” is developed that provide modules for students, supervisors, co supervisors, faculty, industrial mentors, coordinator, external evaluators, head of departments, super administrator. This system handles student group formation, proposals, schedule of evaluation and final project submission.

### **1.2.2 Product Features**

**The system will perform a functionalities:**

1. Allow the user (students) to submit projects documents and receive comment from users(supervisors,co supervisors and industrial mentors).
2. Users (Supervisors,co supervisors and industrial mentors) can view documents, give guidance and comments.
3. User (Coordinators) handles project related users issue and manage project schedules, evaluation timelines and deadlines.
4. User (Super administrator) provides access on role based, manage users and system maintenance.
5. Users (Evaluators) can view project related documents and project , provide the final remarks on system.
6. Maintain the records of all past and present FYP’s details and provide support increasing numbers of students and projects as scalability.

**The system will not perform:**

This system can handle non-FYP academic process and support the student activities outside the FYP process.

### **1.2.3 Application of the System**

Final Year Project Management System is web based platform that will be used in university environment to manage and control all processes related to the FYPs. This system will be used whole year by users (students, supervisors, co supervisors, coordinators, HODs, super admin) and some users (external mentors and Faculty for evaluations) will be used for limited time. It provides consistent access to users any time.

### **1.2.4 Benefits of the System**

This system will provide the following benefits:

1. System removes the manual workload, management and reduce human errors.
2. System provides transparency and efficiency in activities.
3. System ensures each user can access the data related to their roles.
4. System maintains the data in centralized records.
5. System has scalable ability that does not affect the performance.

## **1.3 Definitions, Acronyms, and Abbreviations**

Table 1: Definitions, Acronyms, and Abbreviations

Abbr / Acronym	Type	Definition / Description
FYP	Acronym	Final Year Project
FYPMS	Acronym	Final Year Project Management System
API	Acronym	Application Programming Interface
HTTPS	Acronym	Hypertext Transfer Protocol Secure
TLS	Acronym	Transport Layer Security
ID	Abbreviation	Identifier
Admin	Definition	User has full access of system
Super Admin	Definition	same as admin
Dashboard	Definition	This is interface that each user has own according to use and permissions
User Roles	Definition	Access the assigned users to system (Student, Supervisor, Coordinator, etc.)

## **1.4 References**

These are following documents and resources are referenced in this SRS for FYP Management System and provide the guideline and information during the design of this document and system.

1. IEEE Std 830-1984, IEEE Recommended Practice for Software Requirements Specifications, Date: 1984, Publishing Organization: Institute of Electrical and Electronics Engineers

2. Project Proposal, FYP Management System, Date: 2025, Made by: Mubashir Hassan, Sadia, Shumaila Sarfaraz

## 1.5 Overview

### 1.5.1 Content of the SRS

This Software Requirements Specification document provides a detail of requirements for the FYPMS. It shows the overview of the system, its functionality, assumptions, user characteristics and constraints. Further this document shows the functional requirements, non-functional requirements, external interface requirements that are necessary for design, development, testing, deployment and maintenance of the system.

### 1.5.2 Organization of the SRS

This document is settled and organized according to the standard of IEEE 830, 1984. Document organized as follows:

1. **Section 1: Introduction** — It provides the Introduction with subsections that describes purpose, scope, definitions, references and overview of document.
2. **Section 2: Overall Description** — It provides the Overall Description with subsections that describes product perspective, product functions, user characteristics , general constraints and assumption of system.
3. **Section 3: Specific Requirements** — It provides Specific Requirements with subsections that describes the functional and non functional requirements in details.
4. **Section 4: External Interface Requirements** — It provides External Interface Requirements with subsections that describes user, hardware, software and communication interfaces.
5. **Section 5: Appendices and References** — that provides lists of definition, references and additional resources.

## 2 Overall Description

### 2.1 Product Perspective

#### 2.1.1 System Context

Final Year Project Management System (FYPMS) is a web based digital platform that developed to manage and automate the FYP activities in university. This is the centralized platform that handle the administrative and academic processes which are related to

the FYPs. FYPMS is a platform that has core functions related to the FYPs and not rely on the other academic management system. All data and workflows which are related to projects are handled in system to make consistency, efficiency and transparency among the stakeholders. The system is developed with scalable and flexibility to enhance the system in future.

### **2.1.2 Relationship with Other Systems**

This system is the part of the university project management system that support to FYPs activities, large support to multiple and administrative stakeholders without replacing existing processes of academic and enhance the quality in digital and automate form. System is independent in this current step.

### **2.1.3 Major System Modules**

The system consist of following components:

- **System Module:** This modules holds all modules such as students, supervisors, co supervisors etc. All users modules made a system module.
- **Student Module:** This module is used to submit the project related document and review the remarks and guidance from other users.
- **Supervisor, Co-supervisor, and Industrial Mentor Modules:** This allows to view document of project related and provide remarks on it.
- **Coordinator Module:** It provides scheduling, resolve issues related to project.
- **External Evaluator and Faculty Modules:** This module provides limited time access to these to review the FYP documents of selected student groups.
- **Administration Module:** This module provides role base access, handle users accounts and manage the whole system.

### **2.1.4 External Interfaces**

These are the following principal external interfaces:

- **User Interface:** This system is a web based that accessed through the web browser by all authentic users.
- **Database Interface:** This interface used for storing and getting data of student, project details, remarks and final grading.

- **Notification Interface:** This interface used for storing and getting data of student, project details, remarks and final grading.
- **Dashboard Interface:** This interface is different for each user that shows and provides the data and functionality according to roles.

### **2.1.5 Hardware and Operating Environment**

Final Year Project Management System is operate on computing devices. End users get access the system by browser using mobile, PC, or laptop with internet access. System will handle through centralized server or through cloud technology.

## **2.2 Product Functions**

The Final Year Project Management System (FYPMS) handles a data and workflow in centralized, manage the life cycle of FYPs in automate manner at university. The system supports relationship among students, supervisors, mentors and administrative stakeholders that are the part of the FYP system. The system performs the following main functions:

- **User and Role Management:** System provides access on role based and manages users such as student, supervisors, admin and evaluators.
- **Project and Group Management:** System manages the FYP related information, handles the process of group formation and tracks the project groups of students.
- **Project Proposal Management:** System helps to handle the initial proposal submission to supervisors and get accept or reject, remarks and revision of project proposal by assigned persons.
- **Document Management:** System handles the uploading, storing, updating and accessing documents which are related to FYP.
- **Evaluation and Assessment Management:** System handles the evaluations of project to the assigned evaluators, grading and remarks submission by internal and external evaluators.
- **Notification Support:** System shows the notification and announcement of approvals, deadlines and evaluation timelines.

## **2.3 User Characteristics**

Users of the system are related to the academic and administrative roles related to the FYP in university. These users have different in their usability, operational and accessibility in system. These users should have intermediate computing skills and common knowledge of web application use.

### **2.3.1 Final Year Students**

Students are the primary users of this system. They interact with system to perform the specific activities during documents of project submission, view the remarks of supervisors and mentors, get updates and evaluation periods. Student may not have experience of project management system. Student uses system at time of submissions or wants to view remarks of mentor or supervisors.

### **2.3.2 Supervisors and Co-supervisors**

Supervisors and co supervisors are academic persons that have long academic background and experience in supervision of project. Period of the usage to review the submissions, provide remarks and monitor progress of students.

### **2.3.3 Coordinators and Heads of Department**

Coordinators and Heads of Department are experienced academic and administrative users. These are moderate to advance in their expertise and have well knowledge of academic workflow. They interact with system view projects and ensure the quality according guidelines of university. Coordinator can interact to provide the scheduling and resolve issues related to the FYP of other users.

### **2.3.4 Industrial Mentors**

Industrial mentors are industrial professional who are involved in industrial linked projects. Their idea with system of academics may be limited but they have enough technical knowledge to use web tools. They are only interact with system is occasionally and main focus on reviewing projects and give remarks according to industry who it can be make better or what need this project.

## **2.4 Internal and External Evaluators**

Department faculty and External evaluators are active in assessment phase but these faculties also act as supervisors or co supervisors. They have technical skills and great experience in evaluation of projects.

#### **2.4.1 Placement Cell Department**

Placement cell is the core point which get contact with industries and get projects from them. In this system, they view and get record to monitor which students group with industrial projects and which mentors are linked with these groups. Their use of system is less than other users.

#### **2.4.2 Super Administrator**

Super admin is a main that handles all the system. These are technical advance in skills and great knowledge of system workflow. They responsible for manage, handle, update and maintenance of system.

### **2.5 General Constraints**

There is the development and operational constraints of Final Year Project Management System that are several constraints in these such that organizational and technical. Due to these constraints, limit the possible range of design and implementation of the system.

#### **2.5.1 Regulatory and Academic Policies**

The system must work and run under the polices of university and also follow the institutional guidelines to student assessment, supervision and how system data is managed, accessed and secured.

#### **2.5.2 Hardware and Infrastructure Limitations**

FYPMS is a web based system that each user must require the computing resources available in university or where they use. System contains the server where data stores or retrieve where its performance and accessibility depend network, capacity of server and devices used by end users.

#### **2.5.3 Interfaces with Other Systems**

There is no interaction of this system with other systems in current phase. In future, integrate it with other systems of university to increase capability, security and accuracy such as authentication of users through university provided mail or detail of user except external evaluator and industrial mentor.

#### **2.5.4 Parallel Operations**

During initial implementation of this system, it work with old system or method to check the system accuracy and work in correct way.

### **2.5.5 Programming and Platform Constraints**

System is a web based application that access through with web browser. It uses HTML,CSS and JavaScript with their frameworks such Node.js, react.js, angular.js. Currently, there is no use of java, python or C++ etc and there is no enforcing specific programming language or framework.

### **2.5.6 Network Communication**

System relies on standard web communication and must require reliable network over network.

### **2.5.7 Audit Functions**

System must have audit functions to monitor the user activities and make the transparency of system. In this system, administrative should access to make detailed audit of user behavior and activity. Through this, solve the problems or issues of system misuse.

### **2.5.8 Control Functions**

FYPMS has a control function to maintain the system operations. There is role based access to users to control the unauthorized personnel and perform the sensitive action by authorized personnel.

### **2.5.9 Criticality of the Application**

FYPMS is not a life critical which will use in university for FYP management. If system goes into failures, unavailability or data movement inconsistency can consequence delay in FYP related tasks. These type of disruption may impact the student's progress, faculty and administration workload and this effect the all department timelines.

### **2.5.10 Safety and Security**

System should handles sensitive data of users. System should protect the data privacy, access control and follow the standard security and safety measures. System applies the data encryption like (SSL/TLS) and uses the 2 factor authentication.

## **2.6 Assumptions and Dependencies**

This section tells us the assumption and dependencies that effect the requirements in software requirement s specification for FYP Management system. These factors cause the effect if any change occur in these assumptions or dependencies may need to update to the requirements.

### **2.6.1 User and Device Assumptions**

It assumes that each users has computing device such as mobile,tablet, laptop or desktop with internet access.

### **2.6.2 Network and Connectivity Assumptions**

It is assumed that end users have stable internet connectivity and during fluctuate of internet may cause system not perform in smooth manners .

### **2.6.3 Hosting and Infrastructure Dependencies**

It is assumed that system will hosted on reliable hosts to get availability, speed, scalability and reliability of services.

### **2.6.4 Access Assumption**

It is assumed that user will get access the system through the web browsers such as Chrome etc. if user used the older version may effect the user experience.

### **2.6.5 Security and Data Protection Dependencies**

System follows the international rules of data protection to ensure their features according to the privacy laws.

### **2.6.6 API and External Service Dependencies**

System uses API serves then it follows the international laws of data and it's security, availability and performance are important to smooth running, if the change or disruption in API cause the effect in functionality.

### **2.6.7 Future Enhancements and Scalability Assumptions**

it is assumed that add artificial Intelligent in system to improve the assessment and flow of system and increase the capability of system to apply in different university in future.

## **3 Specific Requirements**

**Note:** Accounts are created by Super Admin and credential are provided to each user by other university process.

### **3.1 Similar Functional Requirements for each user**

#### **3.1.1 Requirement 1: Login**

**Introduction** The purpose of this function is to allow the user to log in to the FYP Management System using the credentials provided by the university. This login functionality ensures that only authorized users can access the system and perform activities related to Final Year Project. Login is required before accessing feature of the system.

#### **Input**

- User ID
- Password

#### **Processing**

**Input Validity Checks** System checks the entered ID present in the database and password matches with database data.

**Sequence of Operations** Users enters login data. System checks the data in database and System provide the access and shows the users dashboard.

**Abnormal Situations** If the User ID or password is incorrect then system shows error message.

**Parameters Affected** User authentication status and login session records are affected.

**Degrade Operation** System performance may reduce during peak hours ,slow internet connectivity or system maintenance.

**Methods Used** System uses role-based authentication and relational database for verification.

**Output Validity Check** System must confirm successful login and ensure correct dashboard visibility.

#### **Outputs**

- User dashboard after successful login.
- Error message in case of invalid data.

## Performance Requirements

**Static Requirement** System shall support login functionality for all users.

**Dynamic Requirement** 95% of login requests shall done in 5 seconds.

## Design Constraints

**Standard Compliance** Login process must follow university security policies.

**Hardware Limitation** Login operation shall work on existing web server and database.

## Attributes

**Availability** Login service shall be available 99% of time except maintenance.

**Security** Only authorized students can log in.

**Maintainability** Login rules and credentials can be updated without system failure.

### 3.1.2 Requirement 2: Reset or Change Password

**Introduction** The purpose of this function is to allow the users to reset or change their password in case they forget it or want to improve account security. This functionality helps users to make secure access to system.

#### Input

- Current Password (only for change password)
- New Password
- Confirm New Password

#### Processing

**Input Validity Checks** System checks current password and new password confirms with confirmation password. It follows security needs.

**Sequence of Operations** Users request password change or reset and system checks inputs and system updates password in database.

**Abnormal Situations** If current password is incorrect or new password is weak and system displays error according to invalid data.

**Parameters Affected** Users authentication data in database is affected.

**Degrade Operation** Password change may delay due to server load or network issues.

**Methods Used** System uses encrypted password storage and authentication mechanisms.

**Output Validity Check** System confirms that password has been updated successfully.

## Outputs

- Password update confirmation message
- Error message for invalid input

## Performance Requirements

**Static Requirement** System shall support password reset for all users.

**Dynamic Requirement** Password update shall complete in 5 seconds.

## Design Constraints

**Standard Compliance** Password handling should follow standard security rules.

**Hardware Limitation** Function shall operate on running web based infrastructure.

## Attributes

**Availability** Password reset feature shall be available during operational time of system.

**Security** Users which are authorized can change their passwords.

**Maintainability** Password policies can be modified easily.

### 3.1.3 Requirement 3: View and Update Personal Profile

**Introduction** The purpose of this function is to allow users to view and update their personal profile information. This includes contact details and academic-related information.

#### Input

- Users Name
- Email Address
- Contact Number
- Profile Picture (optional)

## **Processing**

**Input Validity Checks** System checks email format and contact number.

**Sequence of Operations** User opens profile section and updates information then system saves updated data.

**Abnormal Situations** Invalid data format or missing required fields make a error message to user.

**Parameters Affected** User profile records stored in database is affected.

**Degrade Operation** System may slow down during database maintenance.

**Methods Used** Role-based access control and relational database.

**Output Validity Check** Updated profile must correctly appear on user dashboard.

## **Outputs**

- Updated user profile information
- Error message for invalid input

## **Performance Requirements**

**Static Requirement** System shall support profile viewing and updating for users.

**Dynamic Requirement** Profile updates shall make in 5 seconds.

## **Design Constraints**

**Standard Compliance** Profile information shall follow university record standards.

**Hardware Limitation** Existing database servers shall be used.

## **Attributes**

**Availability** Profile functionality shall be available 99% of time.

**Security** Student can update only their own profile.

**Maintainability** Profile fields can be managed without system downtime.

### **3.1.4 Requirement 4: Receive System Notifications and Announcements**

**Introduction** The purpose of this function is to allow users to receive important notifications and announcements related to FYP activities. This ensures that users get notification according to level and use like student not get the coordinator upload evaluation instruction like that. Only show notifications and announcement according to access.

#### **Input**

- User ID

#### **Processing**

**Input Validity Checks** System checks that user is authorized to receive notifications and that notification is valid.

**Sequence of Operations** System makes notification then checks user access rights. System sends notification to user dashboard.

**Abnormal Situations** System may fail to send notifications if the network or server is down. Delays may occur during maintenance times.

**Parameters Affected** Notification logs, dashboard updates.

**Degrade Operation** Server or network issues may delay notification delivery. System retries automatically when back online.

**Methods Used** Relational database for storing notifications, access control for filtering relevant notifications and announcement.

**Output Validity Check** System ensures that all notifications are visible on user's dashboard.

#### **Outputs**

- Notification received on dashboard

#### **Performance Requirements**

**Static Requirement** System shall support all notifications for users across departments.

**Dynamic Requirement** Notifications shall appear on dashboards within 5 seconds after send.

## Design Constraints

**Standard Compliance** Notifications shall comply with university FYP policies and data confidentiality requirements.

**Hardware Limitation** Notification delivery relies on existing servers and database systems.

## Attributes

**Availability** Notification system shall operate continuously during working hours.

**Security** Logged in authorized users can receive relevant notifications .

**Maintainability** System shall allow addition of new notification types without affecting existing functionality.

## 3.2 Student Module

### 3.2.1 Requirement 1: Form a New Project Group or Join an Existing Group

**Introduction** The purpose of this function is to allow students to form a new Final Year Project group or join an existing group. This ensures that students can organize themselves according to university rules for project group formation.

#### Input

- Group Title (for creating new group)
- Group ID (for joining existing group)
- Group Member Details

#### Processing

**Input Validity Checks** System checks whether group size is within allowed limits and student is not already assigned to another group.

**Sequence of Operations** Student selects create or join option → system validates rules → group is created or student is added.

**Abnormal Situations** Group size exceeded or duplicate group membership.

**Parameters Affected** Group records and student-project association.

**Degrade Operation** System may slow during peak registration periods.

**Methods Used** Rule-based validation and database transactions.

**Output Validity Check** Group details must appear correctly on student dashboard.

## Outputs

- Group creation confirmation
- Successful group joining message
- Error messages for invalid actions

## Performance Requirements

**Static Requirement** System shall support project group formation for all students.

**Dynamic Requirement** Group creation or joining shall process within 5 seconds.

## Design Constraints

**Standard Compliance** Group formation must follow FYP policies.

**Hardware Limitation** Function shall operate on existing web servers.

## Attributes

**Availability** Group formation shall be available during FYP registration phase.

**Security** Only authorized students can create or join groups.

**Maintainability** Group rules can be updated easily.

### 3.2.2 Requirement 2: Send,Accept or Reject Group Invitations

**Introduction** The purpose of this function is to allow students to manage group invitations during the formation of Final Year Project groups. This functionality enables students to send invitations to other students and also respond to invitations received from others. The system ensures that group formation follows university rules, where a project group can have a maximum of two members only.

## Input

- Student Registration Number
- Invitation Action (Send / Accept / Reject)
- Group ID

## **Processing**

**Input Validity Checks** System verifies that the student is not already part of another group and checks that the group size does not exceed the allowed limit.

**Sequence of Operations** Student sends invitation → system validates eligibility → invitation is delivered to invited student. Invited student accepts or rejects → system updates group membership accordingly.

**Abnormal Situations** System shows an error if group size limit is exceeded or student is already assigned.

**Parameters Affected** Group membership records and invitation status.

**Degrade Operation** Network issues or server maintenance may delay the invitation process.

**Methods Used** Rule-based access control and relational database.

**Output Validity Check** System confirms updated group status on student dashboards.

## **Outputs**

- Invitation sent confirmation
- Invitation accepted or rejected notification
- Error messages for invalid actions

## **Performance Requirements**

**Static Requirement** System shall support invitation management for all students.

**Dynamic Requirement** Invitation actions shall be processed within 5 seconds.

## **Design Constraints**

**Standard Compliance** Group formation shall follow university FYP policies.

**Hardware Limitation** Function shall operate on existing web servers.

## **Attributes**

**Availability** Invitation feature shall be available during group formation phase.

**Security** Only authorized students can manage invitations.

**Maintainability** Group size rules can be updated easily.

### **3.2.3 Requirement 3:Browse and Search Available Projects**

**Introduction** The purpose of this function is to allow students to browse and search available Final Year Project ideas provided by supervisors. This helps students explore different project options and select a project that matches their interests and skills.

#### **Input**

- Project Title (optional)
- Project Domain or Category
- Supervisor Name

#### **Processing**

**Input Validity Checks** System checks search keywords and applied filters.

**Sequence of Operations** Student enters search criteria → system retrieves matching projects → results are displayed.

**Abnormal Situations** No matching project found or invalid input.

**Parameters Affected** Project viewing logs.

**Degrade Operation** Heavy system usage may slow down response time.

**Methods Used** Database queries and filtering techniques.

**Output Validity Check** System ensures correct project list is displayed.

#### **Outputs**

- List of available projects
- Filtered project results
- Notification if no project is found

#### **Performance Requirements**

**Static Requirement** System shall allow all students to browse projects.

**Dynamic Requirement** Search results shall be displayed within 4 seconds.

## **Design Constraints**

**Standard Compliance** Projects shall follow approved FYP guidelines.

**Hardware Limitation** Uses existing system infrastructure.

## **Attributes**

**Availability** Project browsing shall be available throughout project selection period.

**Security** Only logged-in students can access projects.

**Maintainability** Project filters and categories can be updated.

### **3.3 Requirement 4: View past Project Description**

**Introduction** The purpose of this function is to allow students to view detailed descriptions of Final Year Projects. This helps students understand project objectives, scope, tools, and expected outcomes before selecting or proposing a project.

#### **Input**

- Project ID

#### **Processing**

**Input Validity Checks** System verifies that the project exists.

**Sequence of Operations** Student selects project → system retrieves project details → displays information.

**Abnormal Situations** Project details unavailable or removed.

**Parameters Affected** Project access records.

**Degrade Operation** Delay due to database or server load.

**Methods Used** Relational database retrieval.

**Output Validity Check** System confirms accuracy of displayed data.

#### **Outputs**

- Complete project description
- Supervisor information
- Project availability status

## **Performance Requirements**

**Static Requirement** System shall allow students to view project details.

**Dynamic Requirement** Project details shall load within 3 seconds.

## **Design Constraints**

**Standard Compliance** Project descriptions shall follow documentation standards.

**Hardware Limitation** Runs on existing web infrastructure.

## **Attributes**

**Availability** Project details shall be accessible during academic sessions.

**Security** Only authorized users can view project information.

**Maintainability** Project details can be updated easily.

### **3.3.1 Requirement 5: Propose New Project Ideas to Supervisors**

**Introduction** The purpose of this function is to allow students to propose new Final Year Project ideas to supervisors when suitable projects are not available. This feature supports innovation and allows students to work on unique ideas under proper supervision.

#### **Input**

- Project Title
- Project Description
- Project Objectives
- Technology Stack

#### **Processing**

**Input Validity Checks** System checks completeness and validity of proposal details.

**Sequence of Operations** Student submits proposal → system forwards proposal to supervisor → supervisor reviews proposal.

**Abnormal Situations** Duplicate proposal or incomplete submission.

**Parameters Affected** Proposal records and supervisor notifications.

**Degrade Operation** Notification delays due to system load.

**Methods Used** Workflow-based approval and relational database.

**Output Validity Check** System confirms successful submission.

## Outputs

- Proposal submission confirmation
- Supervisor response notification
- Error message for invalid proposal

## Performance Requirements

**Static Requirement** System shall support project proposal submission.

**Dynamic Requirement** Proposal shall be submitted within 5 seconds.

## Design Constraints

**Standard Compliance** Proposals shall align with academic policies.

**Hardware Limitation** Uses existing servers and databases.

## Attributes

**Availability** Proposal feature shall be available during project selection phase.

**Security** Only students can submit project proposals.

**Maintainability** Proposal workflow can be updated easily.

### 3.3.2 Requirement 6: Submit Project Proposals and Revisions to Supervisors

**Introduction** The purpose of this function is to allow students to submit their project proposals to the assigned supervisors through the system. This feature also enables students to upload revised versions of the proposal if changes are requested by the supervisor. The system provides a structured and formal way of proposal submission, ensuring transparency and proper documentation throughout the approval process.

## **Input**

- Student Registration Number
- Project Group ID
- Project Proposal Document
- Revision Notes (if applicable)

## **Processing**

**Input Validity Checks** The system verifies that the student belongs to an approved project group and that the supervisor is officially assigned. It also checks whether the uploaded file format and size comply with university guidelines.

**Sequence of Operations** Student uploads proposal → system validates file and group details → proposal is submitted to supervisor → system records submission → confirmation is sent to the student.

**Abnormal Situations** The system displays an error message if the file format is invalid, submission deadline is missed, or the student is not authorized to submit.

**Parameters Affected** Project proposal records, submission timestamps, and revision history.

**Degrade Operation** In case of network failure or server downtime, the submission may be delayed and queued for processing once the system is restored.

**Methods Used** File upload mechanism, database storage, and access control validation.

**Output Validity Check** The system confirms that the proposal has been successfully submitted and stored.

## **Outputs**

- Proposal submission confirmation
- Revision submission acknowledgment
- Error notifications for failed submissions

## **Performance Requirements**

**Static Requirement** The system shall support proposal submissions for all registered students.

**Dynamic Requirement** Proposal uploads shall be processed and stored within 10 seconds.

## Design Constraints

**Standard Compliance** Proposal submission shall follow university FYP documentation standards.

**Hardware Limitation** Function shall operate within existing storage and server infrastructure.

## Attributes

**Availability** Proposal submission shall be available during the official proposal submission period.

**Security** Only authorized students and supervisors can access proposal documents.

**Maintainability** Submission rules and formats can be updated without affecting system stability.

## 3.4 Requirement 7: Check Proposal Approval Status

**Introduction** The purpose of this function is to allow students to check the current approval status of their submitted project proposals. This feature ensures that students are informed about whether their proposal is pending, approved, or requires revision, along with any feedback provided by the supervisor.

### Input

- Student Registration Number
- Project Group ID

### Processing

**Input Validity Checks** The system verifies the student's identity and confirms that a proposal has been submitted for the selected project group.

**Sequence of Operations** Student requests status → system retrieves proposal status → supervisor remarks (if any) are displayed to the student.

**Abnormal Situations** The system displays an error if no proposal exists or access rights are invalid.

**Parameters Affected** Proposal status records and feedback logs.

**Degrade Operation** Temporary delays may occur during high server load or maintenance periods.

**Methods Used** Database querying and role-based access control.

**Output Validity Check** The system ensures that the displayed status is the latest updated by the supervisor.

## Outputs

- Proposal status (Pending / Approved / Revision Required)
- Supervisor feedback or remarks
- Error messages for unavailable records

## Performance Requirements

**Static Requirement** The system shall maintain proposal status records for all student groups.

**Dynamic Requirement** Proposal status shall be displayed within 5 seconds of request.

## Design Constraints

**Standard Compliance** Status labels shall align with university evaluation procedures.

**Hardware Limitation** Function shall work efficiently on existing database servers.

## Attributes

**Availability** Status checking shall be accessible throughout the project lifecycle.

**Security** Only concerned students and supervisors can view proposal status.

**Maintainability** Status categories can be modified if academic policies change.

### 3.4.1 Requirement 8: Upload Project Documents

**Introduction** The purpose of this function is to allow students to upload various project-related documents such as SRS, design documents, progress reports, and final project submissions. This ensures centralized storage and easy access for supervisors and evaluators.

## Input

- Student Registration Number
- Project Group ID
- Project Document File
- Document Type

## **Processing**

**Input Validity Checks** The system checks file format, size, and verifies that the student belongs to the respective project group.

**Sequence of Operations** Student selects document → uploads file → system validates and stores document → confirmation is provided.

**Abnormal Situations** The system generates an error if file size exceeds limits or upload fails.

**Parameters Affected** Project document repository and submission logs.

**Degrade Operation** Uploads may be temporarily unavailable during server downtime.

**Methods Used** Secure file storage system and database indexing.

**Output Validity Check** The system confirms successful upload and correct document association.

## **Outputs**

- Document upload confirmation
- Updated document list
- Error messages for failed uploads

## **Performance Requirements**

**Static Requirement** The system shall support document uploads for all project phases.

**Dynamic Requirement** Documents shall be uploaded and stored within 10 seconds.

## **Design Constraints**

**Standard Compliance** Document formats shall comply with university FYP submission standards.

**Hardware Limitation** Storage capacity shall be managed within existing infrastructure.

## **Attributes**

**Availability** Document upload feature shall be available throughout the academic session.

**Security** Uploaded documents shall be accessible only to authorized users.

**Maintainability** New document types can be added without major system changes.

### **3.4.2 Requirement 9: Replace Documents Within Submission Deadlines**

**Introduction** The purpose of this function is to allow students to replace previously submitted project documents within the allowed submission deadlines. This feature ensures that students can correct mistakes, update content based on supervisor feedback, or submit improved versions of documents while maintaining proper version control and academic integrity.

#### **Input**

- Student Registration Number
- Project Group ID
- Updated Project Document
- Document Type

#### **Processing**

**Input Validity Checks** The system verifies that the replacement request is made before the submission deadline and that the student belongs to the respective project group. It also checks file format and size compliance.

**Sequence of Operations** Student selects document to replace → uploads updated file → system validates deadline and file → old document is archived → new document is saved.

**Abnormal Situations** The system displays an error if the deadline has passed or the file does not meet submission requirements.

**Parameters Affected** Document version records, submission timestamps, and document repository.

**Degrade Operation** Temporary delays may occur during peak submission periods or system maintenance.

**Methods Used** Version control mechanism and secure file storage.

**Output Validity Check** The system confirms that the document has been successfully replaced and updated.

## Outputs

- Document replacement confirmation
- Updated document version display
- Error messages for late or invalid replacements

## Performance Requirements

**Static Requirement** The system shall allow document replacement for all eligible submissions.

**Dynamic Requirement** Document replacement shall be processed within 10 seconds.

## Design Constraints

**Standard Compliance** Document replacement shall comply with university FYP submission policies.

**Hardware Limitation** Function shall operate within existing storage limits.

## Attributes

**Availability** Document replacement shall be available only before deadlines.

**Security** Only authorized group members can replace documents.

**Maintainability** Deadline rules can be updated easily without system redesign.

## 3.5 Requirement 10: View Project Milestone Deadlines

**Introduction** The purpose of this function is to allow students to view all project milestone deadlines defined by the university or project coordinators. This feature helps students manage time effectively by providing clear visibility of important submission dates throughout the Final Year Project lifecycle.

### Input

- Student Registration Number
- Project Group ID

## **Processing**

**Input Validity Checks** The system verifies student credentials and confirms project group association.

**Sequence of Operations** Student requests milestone schedule → system retrieves deadline data → deadlines are displayed in chronological order.

**Abnormal Situations** The system shows an error if milestone data is unavailable.

**Parameters Affected** No system parameters are modified during this operation.

**Degrade Operation** Minor delays may occur during server maintenance.

**Methods Used** Database retrieval and timeline display.

**Output Validity Check** The system ensures that displayed deadlines are current and approved.

## **Outputs**

- List of project milestones
- Associated submission deadlines
- Notifications for upcoming deadlines

## **Performance Requirements**

**Static Requirement** The system shall store milestone deadlines for all FYP sessions.

**Dynamic Requirement** Milestone information shall be displayed within 3 seconds.

## **Design Constraints**

**Standard Compliance** Milestone deadlines shall follow academic calendar guidelines.

**Hardware Limitation** Function shall run on existing web infrastructure.

## **Attributes**

**Availability** Milestone deadlines shall be accessible throughout the project duration.

**Security** Only registered students can view milestone schedules.

**Maintainability** Milestones can be updated without affecting other modules.

### **3.5.1 Requirement 11: View Comments from Supervisors, Co-Supervisors and Industrial Mentors**

**Introduction** The purpose of this function is to allow students to view comments and feedback provided by supervisors, co-supervisors, and industrial mentors on submitted project work. This feature ensures effective communication and helps students improve project quality based on expert guidance.

#### **Input**

- Student Registration Number
- Project Group ID
- Document or Submission Reference

#### **Processing**

**Input Validity Checks** The system verifies that the student belongs to the project group and that comments exist for the selected submission.

**Sequence of Operations** Student selects submission → system retrieves comments → feedback is displayed with author and timestamp.

**Abnormal Situations** The system displays an error if no comments are available or access is unauthorized.

**Parameters Affected** Comment view logs and read status.

**Degrade Operation** Temporary unavailability may occur during database updates.

**Methods Used** Role-based access control and database retrieval.

**Output Validity Check** The system ensures comments are correctly linked to the selected submission.

#### **Outputs**

- Supervisor comments
- Co-supervisor feedback
- Industrial mentor remarks
- Error messages for unavailable feedback

## **Performance Requirements**

**Static Requirement** The system shall store feedback for all project submissions.

**Dynamic Requirement** Comments shall be displayed within 5 seconds.

## **Design Constraints**

**Standard Compliance** Feedback visibility shall follow university communication policies.

**Hardware Limitation** Function shall work efficiently within existing database systems.

## **Attributes**

**Availability** Comment viewing shall be available after feedback is submitted.

**Security** Only authorized students can view project-related comments.

**Maintainability** Feedback structure can be extended to include additional roles.

## **3.6 Requirement 12: View Evaluation Marks**

**Introduction** The purpose of this function is to allow students to view their evaluation marks awarded during different stages of the Final Year Project. This functionality ensures transparency in assessment and helps students understand their performance in proposal evaluation, mid-term evaluation, and final defense.

### **Input**

- Student Registration Number
- Project Group ID
- Evaluation Type (Proposal, Mid, Final)

### **Processing**

**Input Validity Checks** The system verifies that the student belongs to the project group and that the evaluation marks have been officially published by authorized evaluators.

**Sequence of Operations** Student accesses evaluation section → selects evaluation type → system retrieves marks → marks are displayed on dashboard.

**Abnormal Situations** System displays an error if marks are not yet published or access is unauthorized.

**Parameters Affected** Assessment records and evaluation status.

**Degrade Operation** System may respond slowly during peak result publishing time or maintenance.

**Methods Used** Role-based access control and relational database.

**Output Validity Check** System confirms that displayed marks are accurate and approved.

## Outputs

- Displayed evaluation marks
- Evaluation remarks (if provided)
- Error message for unavailable results

## Performance Requirements

**Static Requirement** The system shall support evaluation marks viewing for all students.

**Dynamic Requirement** Marks shall be displayed within 5 seconds.

## Design Constraints

**Standard Compliance** Evaluation marks shall follow university assessment policies.

**Hardware Limitation** Function shall operate on existing servers.

## Attributes

**Availability** Marks shall be available after official publication.

**Security** Only authorized students can view their evaluation results.

**Maintainability** Evaluation formats can be updated without system disruption.

## Requirement 13: Receive System Notifications and Announcements

### **3.6.1 Introduction**

The purpose of this function is to allow students to receive important system notifications and announcements related to the Final Year Project. These notifications include milestone deadlines, evaluation schedules, meeting updates, and general announcements from coordinators, supervisors etc.

#### **Input**

- Students IDs
- Message Content

#### **Processing**

**Input Validity Checks** The system verifies the relevance of notifications based on student role and project group.

**Sequence of Operations** System generates notification → notification is stored → displayed on student dashboard and notification panel.

**Abnormal Situations** Delayed notification delivery due to network or server issues.

**Parameters Affected** Notification logs and read status.

**Degrade Operation** Low network access may delay notification delivery.

**Methods Used** Event-based notification system and database storage.

**Output Validity Check** System ensures notification content is correctly delivered to intended students.

#### **Outputs**

- System notifications
- Announcements alerts
- Unread notification indicators

#### **Performance Requirements**

**Static Requirement** The system shall support notifications for all students.

**Dynamic Requirement** Notifications shall be delivered within 5 seconds of creation.

## Design Constraints

**Standard Compliance** Notifications shall follow university communication guidelines.

**Hardware Limitation** Notification service shall operate on existing infrastructure.

## Attributes

**Availability** Notification service shall be available throughout FYP duration.

**Security** Only authorized announcements shall be visible to students.

**Maintainability** Notification categories can be extended easily.

### 3.6.2 Requirement 14: Schedule Meetings with Supervisors, Co-Supervisors, and Industrial Mentors

**Introduction** The purpose of this function is to allow students to schedule meetings with supervisors, co-supervisors, and industrial mentors for project discussion and guidance.

#### Input

- Student Registration Number
- Project Group ID
- Meeting Date and Time
- Meeting Purpose
- Meeting Participants

#### Processing

**Input Validity Checks** The system checks participant availability and ensures that meeting time does not conflict with existing schedules.

**Sequence of Operations** Student requests meeting → system validates schedule → meeting request is sent → confirmation or rejection is received.

**Abnormal Situations** System shows error if selected time slot is unavailable.

**Parameters Affected** Meeting schedules and availability records.

**Degrade Operation** Scheduling delays may occur during server maintenance.

**Methods Used** Calendar management system and role-based access.

**Output Validity Check** System confirms meeting schedule visibility to all participants.

## Outputs

- Meeting schedule confirmation
- Meeting request notifications
- Error messages for conflicts

## Performance Requirements

**Static Requirement** The system shall support meeting scheduling for all student groups.

**Dynamic Requirement** Meeting scheduling shall be processed within 5 seconds.

## Design Constraints

**Standard Compliance** Meeting scheduling shall comply with academic working hours.

**Hardware Limitation** Function shall operate on existing scheduling services.

## Attributes

**Availability** Meeting scheduling shall be available during active sessions.

**Security** Only authorized users can schedule meetings.

**Maintainability** Scheduling rules can be updated without affecting system flow.

## 3.7 Supervisor Module

### 3.7.1 Requirement 1: View Assigned Project Groups

**Introduction** The purpose of this function is to allow supervisors to view all project groups that have been assigned to them for supervision. This functionality provides a clear overview of all students and groups under the supervisor's responsibility, including project titles, group members, and current progress. It ensures supervisors can track their assigned projects effectively and make informed decisions regarding guidance, feedback, and evaluation.

## Input

- Supervisor ID
- Department information
- Academic session (optional)

## **Processing**

**Input Validity Checks** The system validates the supervisor's ID and ensures the supervisor is authorized to access the list of assigned groups.

**Sequence of Operations** Supervisor logs into the system → navigates to “Assigned Project Groups” section → system retrieves and displays all assigned groups with relevant details including group members, project title, and assigned milestones.

**Abnormal Situations** System displays a notification if the supervisor has no assigned groups or if the data cannot be retrieved due to server or database issues.

**Parameters Affected** Supervisor's dashboard view and project tracking records.

**Degrade Operation** Database maintenance or network latency may delay loading the project group information.

**Methods Used** Relational database queries, role-based access control, and secure data retrieval.

**Output Validity Check** System ensures all assigned project groups are accurately displayed on the supervisor's dashboard.

## **Outputs**

- List of all assigned project groups
- Project group details (members, title, milestones)
- Error messages if retrieval fails

## **Performance Requirements**

**Static Requirement** The system shall support viewing of assigned groups for all supervisors.

**Dynamic Requirement** The assigned groups' data shall load within 5 seconds under normal operating conditions.

## **Design Constraints**

**Standard Compliance** Project group information shall comply with university records and FYP guidelines.

**Hardware Limitation** Dashboard retrieval shall operate on existing servers.

## Attributes

**Availability** Function shall be available during system working hours.

**Security** Only authorized supervisors can access their assigned groups.

**Maintainability** Additional fields for groups or projects can be added without affecting functionality.

## 3.8 Requirement 2: Review and Approve, Reject, or Request Revisions for Project Documents

**Introduction** The purpose of this function is to allow supervisors to review submitted project documents, provide approvals, reject submissions, or request revisions. This ensures that all project documentation meets the required quality standards and adheres to university guidelines before final evaluation.

### Input

- Supervisor ID
- Project Group ID
- Uploaded document(s) for review
- Review action (Approve / Reject / Request Revision)
- Comments or feedback (optional)

### Processing

**Input Validity Checks** System verifies that the document belongs to the assigned project group and is within the submission deadlines. The system also checks if the supervisor is authorized to perform review actions.

**Sequence of Operations** Supervisor selects a project document → reviews content → chooses action (approve, reject, request revision) → optionally adds feedback → system updates the document status and notifies the student group.

**Abnormal Situations** System alerts the supervisor if the document is missing, incorrectly uploaded, or submitted after the deadline.

**Parameters Affected** Project document status, submission records, and notification logs.

**Degrade Operation** Low network speed or server maintenance may delay document retrieval or update.

**Methods Used** Relational database for document storage, role-based access control, and notification services.

**Output Validity Check** System confirms that the document status is updated and visible to the relevant students and supervisors.

## Outputs

- Approval confirmation or rejection notification
- Request revision message sent to students
- Error messages for invalid documents or unauthorized actions

## Performance Requirements

**Static Requirement** System shall support document review for all assigned projects.

**Dynamic Requirement** Review actions and updates shall be processed within 5 seconds.

## Design Constraints

**Standard Compliance** Document review shall follow university FYP policies.

**Hardware Limitation** Document processing shall operate on current servers and storage.

## Attributes

**Availability** Review functionality shall be available during the submission period.

**Security** Only authorized supervisors can approve, reject, or request revisions.

**Maintainability** The system shall allow changes to approval workflows without affecting other modules.

### 3.8.1 Requirement 3: Provide Comments on Submissions

**Introduction** The purpose of this function is to allow supervisors to provide detailed comments and feedback on project submissions. Feedback helps students understand required improvements, learn from mistakes, and adhere to project standards. This functionality ensures proper communication between supervisors and students regarding FYP progress.

## **Input**

- Supervisor ID
- Project Group ID
- Submission document ID
- Comment text (feedback on submission)

## **Processing**

**Input Validity Checks** System verifies that the supervisor is assigned to the project group and that the comment is attached to a valid submission.

**Sequence of Operations** Supervisor opens a submitted document → writes comments → submits feedback → system stores the comments in the database → students can view comments on their dashboard.

**Abnormal Situations** System displays an error if the comment is submitted without valid project or document reference, or if the database fails.

**Parameters Affected** Submission feedback records, project progress logs, and student notification system.

**Degrade Operation** Temporary delays may occur due to server maintenance or network issues.

**Methods Used** Relational database for feedback storage and retrieval, role-based access control for supervisor authorization.

**Output Validity Check** System confirms that comments are correctly linked to the corresponding submission and visible to authorized students.

## **Outputs**

- Comment successfully saved notification
- Comments visible on student dashboards
- Error messages for invalid comment submission

## **Performance Requirements**

**Static Requirement** System shall support supervisor comments for all assigned projects.

**Dynamic Requirement** Comments shall appear in student dashboards within 5 seconds of submission.

## **Design Constraints**

**Standard Compliance** Comments shall follow university FYP communication and feedback standards.

**Hardware Limitation** Function shall operate on existing web servers and database systems.

## **Attributes**

**Availability** Comment functionality shall be available during submission and review periods.

**Security** Only supervisors can provide comments for their assigned groups.

**Maintainability** Feedback mechanisms can be enhanced in future without affecting other system functions.

### **3.8.2 Requirement 4:Schedule Meetings with Students**

**Introduction** The purpose of this function is to allow supervisors to schedule meetings with students regarding project progress, clarifications, or guidance. This functionality ensures effective communication between supervisors and student groups and enables proper tracking of meeting schedules within the Final Year Project (FYP) system.

#### **Input**

- Supervisor ID
- Student Group ID
- Proposed meeting date and time
- Meeting venue or virtual platform link
- Agenda or topic for discussion (optional)

#### **Processing**

**Input Validity Checks** System validates the supervisor ID and confirms that the students belong to an assigned project group. It also checks for scheduling conflicts and ensures that the meeting falls within university working hours.

**Sequence of Operations** Supervisor navigates to the “Schedule Meeting” section → selects project group → enters meeting details → system validates inputs → schedules the meeting → sends notifications to the student group.

**Abnormal Situations** System displays an error if there is a scheduling conflict, invalid date/time, or if students are unavailable. Network delays may prevent timely notification delivery.

**Parameters Affected** Meeting schedules, notifications, and student dashboards.

**Degrade Operation** Network issues, server maintenance, or database downtime may delay scheduling or notifications.

**Methods Used** Relational database to store meeting information, rule-based scheduling checks, and notification system for alerts.

**Output Validity Check** System confirms scheduled meetings are accurately displayed on supervisor and student dashboards.

## Outputs

- Meeting scheduled confirmation
- Notifications sent to students
- Error messages for invalid or conflicting meetings

## Performance Requirements

**Static Requirement** System shall support scheduling of multiple meetings for all assigned project groups.

**Dynamic Requirement** Meeting scheduling and notifications shall be processed within 5 seconds.

## Design Constraints

**Standard Compliance** Meetings shall comply with university working hours and FYP guidelines.

**Hardware Limitation** Function shall operate on existing web servers and scheduling database.

## Attributes

**Availability** Scheduling feature shall be available during the project period.

**Security** Only authorized supervisors can schedule meetings for their assigned groups.

**Maintainability** System shall allow updates to meeting scheduling rules or notification formats without affecting other modules.

### 3.9 Co-Supervisor Module

#### 3.9.1 Requirement 1: View Assigned Projects and Student Groups

**Introduction** The purpose of this function is to allow co-supervisors to access a list of projects and student groups assigned to them. This ensures co-supervisors have visibility of their responsibilities and can monitor student progress effectively. The system provides an organized view of each group's members, assigned project titles, and relevant deadlines.

##### Input

- Co-Supervisor ID
- Project or Group Filter (optional)

##### Processing

**Input Validity Checks** System validates the co-supervisor's authorization and confirms that the requested projects are assigned to them. Filters are checked for valid criteria such as group ID or project category.

**Sequence of Operations** Co-supervisor accesses the “Assigned Projects” section → system retrieves assigned projects from the database → displays student groups along with project details including group members, project title, and submission deadlines.

**Abnormal Situations** System displays an error if the co-supervisor has no assigned projects or if the database fails to fetch the relevant records. Network delays may cause slower loading times.

**Parameters Affected** Database retrieval logs, dashboard view sessions.

**Degrade Operation** If the system experiences high traffic or maintenance, loading of assigned projects may be delayed.

**Methods Used** Relational database queries, role-based access control, and session management.

**Output Validity Check** System confirms that all assigned projects and student group details are correctly displayed to the co-supervisor without omissions.

## Outputs

- List of assigned projects with group members
- Project titles and submission deadlines
- Notifications if no projects are assigned

## Performance Requirements

**Static Requirement** The system shall support retrieval and display of assigned projects for all co-supervisors.

**Dynamic Requirement** Assigned projects and student group lists shall be loaded within 5 seconds.

## Design Constraints

**Standard Compliance** Project and group views shall comply with university FYP management policies.

**Hardware Limitation** Function shall operate on existing web and database servers.

## Attributes

**Availability** Feature shall be available during working hours and accessible remotely.

**Security** Only authorized co-supervisors can view their assigned projects.

**Maintainability** System shall allow modifications to assignment rules without impacting the display functionality.

### 3.9.2 Requirement 2: Access Student Submissions

**Introduction** The purpose of this function is to enable co-supervisors to access and review project submissions made by their assigned student groups. This ensures that co-supervisors can monitor student progress, verify the quality of work, and provide timely feedback to maintain FYP standards.

## Input

- Co-Supervisor ID
- Project ID or Student Group ID
- Submission Type (Proposal, Report, Milestone Documents)

## **Processing**

**Input Validity Checks** System verifies that the co-supervisor is authorized for the selected project or group and that the requested submission type exists. File formats and size limitations are validated for proper access.

**Sequence of Operations** Co-supervisor selects a project or student group → system fetches submissions from the database → files are displayed or downloaded for review.

**Abnormal Situations** System displays errors if submissions are missing, file formats are unsupported, or access is attempted on unassigned projects. Network issues may delay file retrieval.

**Parameters Affected** Student submission logs, access records, and database file references.

**Degrade Operation** Server maintenance or network downtime may delay access to submissions.

**Methods Used** Database retrieval, file management, access control, and validation rules for file integrity.

**Output Validity Check** System ensures that all requested submissions are correctly retrieved and accessible to the co-supervisor.

## **Outputs**

- Access to submitted project documents
- Error messages for missing or unauthorized files
- Downloadable files for offline review

## **Performance Requirements**

**Static Requirement** The system shall support all co-supervisors in accessing submissions for their assigned groups.

**Dynamic Requirement** Submissions shall be retrieved within 5 seconds under normal operating conditions.

## Design Constraints

**Standard Compliance** Submission access shall follow university FYP guidelines and file handling policies.

**Hardware Limitation** Function shall work on existing web servers and database infrastructure.

## Attributes

**Availability** Submission access shall be available during working hours and project periods.

**Security** Only authorized co-supervisors can access submissions of assigned groups.

**Maintainability** System shall allow updates for new submission types or formats without affecting access.

### 3.9.3 Requirement 3: Provide Feedback and Suggestions on Submissions

**Introduction** The purpose of this function is to allow co-supervisors to provide detailed feedback and suggestions on student project submissions. This ensures students receive constructive guidance to improve their work and align with FYP quality standards.

#### Input

- Co-Supervisor ID
- Project ID
- Submission File or Document
- Feedback Comments
- Suggested Changes or Recommendations

#### Processing

**Input Validity Checks** System verifies that the co-supervisor is authorized to provide feedback for the selected project/group. Feedback entries are checked for completeness and format.

**Sequence of Operations** Co-supervisor selects the submission → system displays the document → co-supervisor enters feedback and suggestions → system validates input → feedback is stored in the database and linked to the respective submission.

**Abnormal Situations** System shows an error if the co-supervisor attempts to give feedback on unassigned projects or if the database fails to save the comments. Network disruptions may delay feedback submission.

**Parameters Affected** Feedback records, submission status, and student notification logs.

**Degrade Operation** Server or network issues may temporarily prevent feedback submission.

**Methods Used** Relational database for storing feedback, role-based access, and validation of input formats.

**Output Validity Check** System confirms that feedback is stored correctly and visible to both students and supervisors as per access permissions.

## Outputs

- Confirmation of feedback submission
- Display of comments on student dashboards
- Error messages for failed submissions

## Performance Requirements

**Static Requirement** System shall allow feedback provision for all co-supervisors on assigned projects.

**Dynamic Requirement** Feedback entries shall be processed and visible within 5 seconds.

## Design Constraints

**Standard Compliance** Feedback and suggestions shall follow university FYP evaluation guidelines.

**Hardware Limitation** Function operates on existing servers and relational database infrastructure.

## Attributes

**Availability** Feedback feature shall be available during submission periods and working hours.

**Security** Only authorized co-supervisors can provide feedback for assigned projects.

**Maintainability** Feedback templates and formats can be updated without affecting system operation.

### 3.9.4 Requirement 4: Schedule Meetings with Project Groups

**Introduction** The purpose of this function is to allow co-supervisors to schedule meetings with their assigned student project groups. This ensures proper coordination, progress tracking, and guidance throughout the FYP process. The system provides an interface for setting meeting dates, times, and agendas, while sending automatic notifications to all participants.

#### Input

- Co-Supervisor ID
- Project Group ID
- Meeting Date and Time
- Meeting Venue or Online Link
- Agenda or Purpose of Meeting

#### Processing

**Input Validity Checks** System verifies that the co-supervisor is authorized for the selected group. The chosen date and time are checked against existing meetings to avoid conflicts. Agenda or meeting description is validated for completeness.

**Sequence of Operations** Co-supervisor selects a project group → system checks eligibility → co-supervisor enters meeting details → system validates input → meeting is saved in the database → notifications are sent to students and other stakeholders.

**Abnormal Situations** System shows an error if the selected date/time conflicts with another meeting, the co-supervisor is unauthorized, or mandatory meeting details are missing. Network delays may postpone notification delivery.

**Parameters Affected** Meeting schedules, student and co-supervisor notifications, and project progress logs.

**Degrade Operation** System maintenance or low network connectivity may temporarily delay meeting scheduling or notifications.

**Methods Used** Relational database for storing meetings, role-based access control, and automated notification service.

**Output Validity Check** System confirms that meeting details are correctly saved and notifications are delivered to authorized users.

## **Outputs**

- Confirmation of scheduled meeting
- Notification sent to project group members
- Error messages in case of conflicts or invalid input

## **Performance Requirements**

**Static Requirement** System shall allow all co-supervisors to schedule meetings for their assigned project groups.

**Dynamic Requirement** Meeting scheduling and notification delivery shall be completed within 5 seconds under normal operation.

## **Design Constraints**

**Standard Compliance** Meeting scheduling shall follow university FYP coordination policies.

**Hardware Limitation** Function shall operate on the existing web and database servers.

## **Attributes**

**Availability** Meeting scheduling shall be available during working hours and project periods.

**Security** Only authorized co-supervisors can schedule meetings for their assigned groups.

**Maintainability** Meeting templates and agenda formats can be updated without affecting system operation.

### 3.10 Head of Department Module

#### 3.10.1 Requirement 1: View All FYP Projects of the Department

**Introduction** The purpose of this function is to allow the Head of Department (HoD) to view all Final Year Project (FYP) projects within the department. This enables the HoD to monitor the status of projects, track progress, and ensure that departmental policies and academic standards are being followed.

##### Input

- HoD Username (automatically verified via session)
- Filters (Optional: Project Title, Supervisor, Status, Group Members)

##### Processing

**Input Validity Checks** System ensures that only the authenticated HoD can access departmental project information. Filters are validated for correctness (e.g., dates in correct format, supervisor IDs exist).

**Sequence of Operations** HoD accesses FYP projects section → system retrieves all project records from the department → applies any filters provided → displays project details on the HoD dashboard.

**Abnormal Situations** Invalid filter inputs or database connectivity issues may prevent full retrieval of project records. System displays error messages if no records match the filter criteria.

**Parameters Affected** Departmental project records, project status logs, and reporting metrics.

**Degrade Operation** Database maintenance, server downtime, or network delays may slow retrieval of project information.

**Methods Used** Relational database queries, role-based access control, and filtering algorithms.

**Output Validity Check** System ensures all retrieved project records match the department and any applied filters.

## Outputs

- Display of all departmental FYP projects
- Filtered project lists based on user-selected criteria
- Error messages for invalid filters or system issues

## Performance Requirements

**Static Requirement** System shall support retrieval of all FYP projects for the department at any time.

**Dynamic Requirement** Project lists shall load and display within 5 seconds for up to 500 active projects.

## Design Constraints

**Standard Compliance** Project information retrieval shall comply with university privacy and data protection policies.

**Hardware Limitation** Function shall operate on existing web servers and database infrastructure.

## Attributes

**Availability** Project viewing shall be available 24/7, except during scheduled maintenance.

**Security** Only the HoD can access departmental project information.

**Maintainability** Filters and project display format can be updated without affecting system operations.

### 3.10.2 Requirement 2: View and Approve Supervisor and Co-Supervisor Assignments

**Introduction** The purpose of this function is to allow the HoD to review, approve, or request modifications for supervisor and co-supervisor assignments to student groups. This ensures that assignments align with departmental policies, faculty workload distribution, and project requirements.

## **Input**

- Student Group ID
- Assigned Supervisor ID
- Assigned Co-Supervisor ID
- HoD Decision (Approve / Request Modification)

## **Processing**

**Input Validity Checks** System validates that group IDs exist, supervisors are valid and available, and that workload rules are followed.

**Sequence of Operations** HoD views assigned supervisors and co-supervisors → reviews assignment against workload policies and project requirements → approves or requests modification → system updates assignment records and notifies relevant parties.

**Abnormal Situations** Invalid supervisor or co-supervisor IDs, duplicate assignments, or conflicts with faculty availability trigger error messages.

**Parameters Affected** Supervisor and co-supervisor assignment records, student group records, and notification logs.

**Degrade Operation** Network issues, server downtime, or database maintenance may delay updates or notifications.

**Methods Used** Relational database operations, role-based access control, and notification services.

**Output Validity Check** System confirms that assignment updates are saved accurately and reflected in dashboards for students and faculty.

## **Outputs**

- Confirmation of assignment approval
- Notification for modification requests
- Error messages for invalid assignments or conflicts

## **Performance Requirements**

**Static Requirement** System shall allow the HoD to approve or modify assignments for all supervisors and co-supervisors in the department.

**Dynamic Requirement** Assignments shall be processed and notifications sent within 5 seconds.

## **Design Constraints**

**Standard Compliance** Assignments shall follow departmental workload and university regulations.

**Hardware Limitation** Function shall work on existing web servers and database infrastructure.

## **Attributes**

**Availability** Approval functionality shall be available during working hours and scheduled maintenance windows.

**Security** Only the HoD can approve or modify assignments.

**Maintainability** Assignment rules and faculty availability can be updated without downtime.

### **3.10.3 Requirement 3:Approve or Reject Projects**

**Introduction** The purpose of this function is to allow the HoD to approve or reject FYP projects proposed by students. This ensures that projects meet departmental standards, academic policies, and FYP guidelines before students proceed with implementation.

#### **Input**

- Project ID
- Project Title
- Student Group Members
- Supervisor and Co-Supervisor IDs
- HoD Decision (Approve / Reject)
- Comments or Feedback

## **Processing**

**Input Validity Checks** System verifies project existence, assigned supervisors and co-supervisors, and ensures the project adheres to departmental policies.

**Sequence of Operations** HoD views project details → reviews supervisor assignments and project proposal → approves or rejects project → system updates project status → sends notifications to students and supervisors.

**Abnormal Situations** Invalid project ID, missing documentation, or policy violation triggers error messages.

**Parameters Affected** Project approval status, student records, and supervisor dashboards.

**Degrade Operation** Server downtime or network issues may delay approval or notification.

**Methods Used** Role-based access control, relational database operations, and automated notifications.

**Output Validity Check** System confirms project approval or rejection is correctly reflected in all relevant dashboards.

## **Outputs**

- Project approval confirmation
- Project rejection notification with feedback
- Error messages for invalid project information

## **Performance Requirements**

**Static Requirement** System shall allow the HoD to approve or reject all projects within the department.

**Dynamic Requirement** Project approval or rejection updates shall be processed within 5 seconds.

## **Design Constraints**

**Standard Compliance** Project approvals shall comply with university FYP policies and departmental standards.

**Hardware Limitation** Function shall operate on existing web and database servers without additional resources.

Project approval function shall be available during working hours and maintenance windows.

Only the HoD can approve or reject projects.

Approval rules and feedback templates can be updated without system downtime.

### **3.11 Industrial Mentors Module**

#### **Attributes**

##### **3.11.1 Requirement 1: View Assigned Student Projects**

**Introduction** The main purpose of this function is to allow Industrial Mentors to view all the projects that are specifically assigned to them by the placement cell. This functionality helps mentors keep track of their responsibilities, ensures that they know which student groups they are guiding, and facilitates organized project supervision. The system ensures that only authorized mentors can access their assigned projects, maintaining data confidentiality and integrity.

#### **Inputs**

- Mentor Credentials: Login details provided by the placement cell.
- Assigned Project IDs: Unique identifiers of projects allocated to the mentor.
- Student Project Data: Titles, proposals, progress reports, and related documents uploaded by students.

#### **Processing**

- Input Validity Checks:
  1. Verify mentor login credentials.
  2. Ensure that the mentor is assigned to the specific projects.
  3. Validate the existence and accessibility of project data.
- Sequence of Operations:
  1. Mentor logs into the system using their credentials.
  2. System displays a dashboard of all projects assigned to the mentor.
  3. Mentor selects a project to view detailed information and documents.
  4. System ensures that project details are current and properly formatted.

### Abnormal Situations

- If a mentor attempts to view projects not assigned to them, access is denied.
- Missing or corrupted project files trigger notifications for corrective action.

Methods Used Role-based access control (RBAC) ensures secure access to only authorized project information.

### Outputs

- Dashboard listing all assigned student projects.
- Access to project details, including uploaded documents and submission timelines.
- System logs showing mentor access for auditing purposes.

### Performance Requirements

- Project lists and details must load within 2–3 seconds for 99% of requests.

### Design Constraints

- Must follow university and departmental data access guidelines.
- Should function on existing institutional servers without additional resources.

### Attributes

- **Availability:** Function is accessible during working hours, excluding scheduled maintenance.
- **Security:** Only assigned mentors can access their project data.
- **Maintainability:** Dashboard logic and project display can be updated without system redesign.

#### **3.11.2 Requirement 2: Review Project Documents**

**Introduction** The main purpose of this function is to allow Industrial Mentors to review the documents submitted by students for their assigned projects. This functionality ensures that mentors can assess the progress, quality, and completeness of student work, provide constructive feedback, and verify compliance with academic and industrial standards.

## **Inputs**

- Mentor Credentials: Login details provided by the placement cell.
- Assigned Project IDs: Unique identifiers of projects allocated to the mentor.
- Student Project Documents: Uploaded proposals, progress reports, final reports, and supplementary files.

## **Processing**

- Input Validity Checks:
  1. Verify mentor login credentials and role authorization.
  2. Confirm that the mentor is assigned to the selected project.
  3. Validate the file format and completeness of uploaded documents (e.g., PDF, DOCX).
- Sequence of Operations:
  1. Mentor logs into the system.
  2. Selects a specific project from their assigned list.
  3. Views and examines uploaded documents for completeness, correctness, and quality.
  4. Provides feedback comments if necessary.
  5. System logs the review activity and timestamps it.

## Abnormal Situations

- Unauthorized access to unassigned projects triggers an error message.
- Corrupted or unsupported file formats generate a notification.
- Missing project files prompt an alert to the coordinator.

Methods Used Role-based access control (RBAC) for secure and authorized access; database transactions to log review actions.

## **Outputs**

- Reviewed project documents accessible on mentor dashboard.
- Notifications confirming document review logged in the system.
- Audit logs of mentor review activities for accountability.

## Performance Requirements

- Documents should open and display within 3–4 seconds for 99% of requests.

## Design Constraints

- Document review must adhere to university and departmental guidelines.
- Functionality must work with existing servers without extra hardware.

## Attributes

- **Availability:** Accessible during working hours except for scheduled maintenance.
- **Security:** Only assigned mentors can access and review documents; all actions are logged.
- **Maintainability:** Document handling logic can be updated without major system redesign.

### **3.11.3 Requirement 3: Provide Comments on Projects**

**Introduction** This function allows Industrial Mentors to provide detailed feedback and comments on student projects. The purpose is to guide students, highlight areas of improvement, and ensure the project aligns with academic and industrial standards.

#### Inputs

- Mentor Credentials: Login information provided by the placement cell.
- Assigned Project IDs: Unique identifiers for the projects assigned to the mentor.
- Student Project Documents: Submitted proposals, progress reports, and other relevant files.
- Feedback/Comments: Mentor's evaluation notes, suggestions, and recommendations.

#### Processing

- Input Validity Checks:
  1. Verify mentor identity and authorization for the project.
  2. Ensure feedback/comments follow character limits and formatting rules.
  3. Check that comments correspond to the correct project ID.

- Sequence of Operations:
  1. Mentor logs into the system.
  2. Selects a project from the assigned list.
  3. Views the uploaded documents.
  4. Adds comments, suggestions, or guidance notes for the project.
  5. Saves and submits the comments, which are logged with timestamps.
  6. System sends notifications to students, supervisors, and coordinators regarding the new feedback.

### Abnormal Situations

- Attempt to comment on unassigned projects triggers an error.
- System rejects comments exceeding allowed length or incorrect format.
- Network issues may temporarily prevent comment submission.

Methods Used Role-based access control (RBAC), secure database transactions, automated notifications via email or system alerts, audit logging for accountability.

### Outputs

- Mentor comments visible to students and supervisors on dashboards.
- Notifications confirming feedback submission.
- Audit logs of mentor feedback actions.

### Performance Requirements

- 99% of comments should be submitted and reflected in the system within 3 seconds.

### Attributes

- **Security:** Only assigned mentors can provide feedback; all comments are encrypted and logged.
- **Availability:** Accessible during working hours except scheduled maintenance.
- **Maintainability:** Comment submission logic can be updated without system redesign.

### **3.11.4 Requirement 4: Schedule Meeting with Project Group**

**Introduction** This function enables Industrial Mentors to schedule meetings with their assigned student project groups. The purpose is to provide guidance, clarify doubts, review progress, and ensure projects align with academic and industrial standards.

#### **Inputs**

- Mentor Credentials: Login information provided by the placement cell.
- Assigned Project IDs: Unique identifiers for projects assigned to the mentor.
- Proposed Meeting Dates and Times: Suggested schedule for discussion with the project group.
- Student Availability Data: Optional input to verify suitable meeting times.

#### **Processing**

- Input Validity Checks:
  1. Verify mentor identity and project assignment.
  2. Check that proposed meeting time is within working hours and not in the past.
  3. Validate that the selected students belong to the assigned project group.
- Sequence of Operations:
  1. Mentor logs into the system.
  2. Selects a project from the assigned list.
  3. Chooses a suitable date and time for the meeting.
  4. Confirms student group availability.
  5. Schedules the meeting and stores it in the system database.
  6. Sends notifications automatically to students, supervisors, and coordinators.

#### **Abnormal Situations**

- Attempt to schedule meetings outside working hours or in the past triggers an error.
- Conflicts with existing schedules result in a warning for rescheduling.
- Network or server issues may delay scheduling notifications.

**Methods Used** Role-based access control (RBAC), secure database transactions, automated notification services, audit logging.

## Outputs

- Scheduled meetings visible to mentors, students, and supervisors.
- Notifications confirming meeting details sent to all participants.
- Audit logs of scheduled meetings for accountability.

## Performance Requirements

- 99% of meeting schedules and notifications should be processed and reflected in the system within 3 seconds.

## Attributes

- **Security:** Only assigned mentors can schedule meetings; meeting data is encrypted and logged.
- **Availability:** Accessible during working hours except for scheduled maintenance.
- **Maintainability:** Scheduling logic can be updated without redesigning the system.

## 3.12 Placement Cell Department Module

### 3.12.1 Requirement 1: View Industry-Based Projects

**Introduction** This function allows Placement Cell staff to view all industry-based projects registered in the FYP Management System. It ensures proper monitoring of student engagement with industrial partners, tracks mentor assignments, and facilitates oversight of project progress.

## Inputs

- User Credentials: Login information for authorized Placement Cell staff.
- Project Filters (Optional): Criteria to search or sort projects, such as project status, student group, or assigned mentor.

## Processing

### • Input Validity Checks:

1. Verify Placement Cell staff login credentials.
2. Ensure staff has access permissions for industry-based projects.
3. Validate filter parameters if applied.

- **Sequence of Operations:**

1. Staff logs into the system.
2. Accesses the list of all industry-based projects.
3. Views project details, including student group, assigned mentor, and project progress.
4. System retrieves real-time data from the database and displays it on the dashboard.

Abnormal Situations

1. Unauthorized access attempts trigger an error message and are logged.
2. Missing project data or mentor assignment records generate a warning notification.
3. System failure or network issues temporarily prevent access, with proper error handling.

Parameters Affected

- Project records accessed by Placement Cell staff.
- Dashboard view logs and audit trails for monitoring access.

Degrade Operation Heavy server load may slow retrieval of project lists temporarily, but access remains functional.

Methods Used

1. Role-Based Access Control (RBAC) to ensure only authorized staff can view projects.
2. Database queries for retrieving up-to-date project data.
3. Audit logging to record all access for accountability.

## Outputs

- List of industry-based projects displayed on the dashboard.
- Project details including student group, mentor, status, and progress.
- Notifications or warnings in case of missing data or access issues.

## Performance Requirements

- **Static Requirement:** System shall allow Placement Cell staff to view all industry-based projects securely.
- **Dynamic Requirement:** 99% of project list retrieval requests should be processed within 3 seconds under normal load.

## Design Constraints

- **Standards Compliance:** Must follow university guidelines for data access and confidentiality.
- **Hardware Limitation:** Function must operate on existing servers without additional hardware.

## Attributes

- **Availability:** Available during working hours; occasional maintenance may cause temporary downtime.
- **Security:** Access restricted to authorized Placement Cell staff; sensitive data is encrypted.
- **Maintainability:** Project viewing logic can be updated without redesigning the system.
- **Transferability / Conversion:** Can integrate with university project management systems in the future.

### 3.12.2 Requirement 2: View Engaged Industrial Mentor Records

**Introduction** This function allows Placement Cell staff to access records of all Industrial Mentors engaged with student projects. It ensures proper monitoring of mentor participation, helps track mentor assignments, and facilitates coordination between students, supervisors, and mentors.

#### Inputs

- User Credentials: Login information for authorized Placement Cell staff.
- Mentor Filters (Optional): Search parameters such as mentor name, assigned project, or availability status.

#### Processing

- **Input Validity Checks:**

1. Verify Placement Cell staff login credentials.
2. Ensure staff has authorization to view mentor records.
3. Validate filter or search criteria if applied.

- **Sequence of Operations:**

1. Staff logs into the system.
2. Accesses the section for Industrial Mentor records.
3. Views the list of engaged mentors along with project assignments and contact details.
4. System retrieves real-time mentor information from the database and displays it accurately.

#### Abnormal Situations

1. Unauthorized access triggers an error message and is logged for security.
2. Missing or incomplete mentor assignment data generates a warning notification.
3. Network interruptions temporarily prevent access, with proper error messages.

#### Parameters Affected

- Mentor assignment records and project engagement logs accessed by staff.
- Audit logs tracking access and retrieval activity.

Degrade Operation High server load may slow retrieval of mentor records but access remains functional.

#### Methods Used

1. Role-Based Access Control (RBAC) ensures only authorized Placement Cell staff can view mentor data.
2. Database queries for retrieving accurate mentor assignment records.
3. Audit logging to track access and maintain accountability.

## Outputs

- Complete list of Industrial Mentors engaged in student projects.
- Details of mentor assignments, contact information, and project involvement.

## Performance Requirements

- **Static Requirement:** The system shall allow Placement Cell staff to view engaged Industrial Mentor records securely.
- **Dynamic Requirement:** 99% of mentor record retrieval requests should be completed within 3 seconds under normal load.

## Design Constraints

- **Standards Compliance:** Must follow university guidelines for mentor data confidentiality and reporting.
- **Hardware Limitation:** Function must operate efficiently on existing servers without additional hardware.

## Attributes

- **Availability:** Available during working hours, with minimal downtime for maintenance.
- **Security:** Access restricted to authorized staff; mentor records are encrypted.
- **Maintainability:** Mentor viewing functionality can be updated without system redesign.
- **Transferability / Conversion:** Can integrate with university project management systems in the future.

### **3.13 Requirement 3: Track Groups Who Work on Industrial Projects**

**Introduction** This function allows Placement Cell staff to monitor and track student groups involved in industry-linked projects. It ensures that student progress is aligned with project milestones, facilitates coordination with Industrial Mentors, and provides oversight for timely completion of projects.

#### Inputs

- User Credentials: Login details for authorized Placement Cell staff.
- Project Identifiers: Unique IDs of industry-linked projects to filter student groups.
- Tracking Filters (Optional): Parameters such as student group name, project stage, or mentor assigned.

#### Processing

- **Input Validity Checks:**
  1. Verify staff login credentials and authorization.
  2. Ensure selected project IDs exist and are linked to industry-based projects.

3. Validate any applied filters or search parameters.

- **Sequence of Operations:**

1. Staff logs into the system.
2. Navigates to the student project tracking section.
3. Selects the projects or groups to track.
4. Views real-time progress, submissions, and milestone completion status.
5. System updates tracking logs and timestamps automatically.

#### Abnormal Situations

1. Unauthorized access attempts trigger an error and are logged.
2. Missing project or group data generates a warning notification.
3. Network interruptions may delay tracking information but system provides proper error messages.

#### Parameters Affected

- Student group progress records and milestone logs.
- Mentor assignment and tracking data.
- Audit logs for tracking activity.

Degrade Operation System may experience slower response under high load but remains accessible for tracking purposes.

#### Methods Used

1. Role-Based Access Control (RBAC) ensures only authorized Placement Cell staff can access student group data.
2. Database transactions to retrieve and update project tracking data reliably.
3. Audit logging to maintain accountability and trace all tracking activities.

## Outputs

- Complete list of student groups working on industry-linked projects.
- Current project status, milestones achieved, and mentor assignments.
- Notifications or warnings if data is incomplete or inconsistent.

## Performance Requirements

- **Static Requirement:** The system shall allow Placement Cell staff to track student groups on industrial projects securely and efficiently.
- **Dynamic Requirement:** 99% of tracking requests should be processed within 3–5 seconds under normal conditions.

## Design Constraints

- **Standards Compliance:** Must follow university policies for student data access and confidentiality.
- **Hardware Limitation:** Function must operate using existing servers without additional hardware requirements.

## Attributes

- **Availability:** Accessible during working hours with minimal downtime.
- **Security:** Only authorized staff can view tracking data; all records are encrypted.
- **Maintainability:** Tracking functionality can be updated without major redesign.
- **Transferability / Conversion:** Can integrate with future project management or evaluation modules.

## 3.14 External and Internal (faculty) Evaluators Module

### 3.14.1 Requirement 1: View Assigned Projects for Evaluation

**Introduction** This function allows External and Internal Evaluators to view the list of Final Year Projects assigned to them for assessment. It ensures that evaluators have access only to projects they are authorized to evaluate, supporting impartial and organized evaluation processes. The system provides secure access to project details and related documents.

#### Inputs

- Evaluator ID: Credentials provided by the university to identify the evaluator.
- Assigned Project List: System-generated list of projects allocated to the evaluator.

### **Processing Input Validity Checks:**

1. System verifies evaluator login credentials and role authorization.
2. Confirms that projects listed are specifically assigned to the logged-in evaluator.

### **Sequence of Operations:**

1. Evaluator logs into the FYPMS using university-provided credentials.
2. System retrieves the assigned project list from the database.
3. Evaluator selects a project to view its details and related documents.

### **Abnormal Situations:**

1. Attempt to access projects not assigned to the evaluator triggers an access denied error.
2. Missing or corrupted project data triggers a warning message.

Parameters Affected: Access logs, timestamps, and project view history.

Degrade Operation: Temporary network issues may delay retrieval of assigned projects.

Methods Used: Role-based access control (RBAC) and secure database queries.

### **Outputs**

- Display of assigned projects with project ID, title, and relevant documents.
- Notifications if project data is unavailable or access is restricted.

**Performance Requirements Static:** System shall display all projects assigned to the evaluator securely.

**Dynamic:** 99% of assigned project lists shall load within 2–3 seconds under normal conditions.

**Design Constraints Standards Compliance:** Must comply with university data privacy and evaluation policies.

**Hardware Limitation:** Function operates on existing servers with no additional hardware required.

**Attributes Availability:** Function should be available during evaluation periods.

**Security:** Evaluators can access only their assigned projects; data is protected and access is logged.

**Maintainability:** Project list display logic can be updated without affecting other modules.

**Transferability/Conversion:** Can integrate with future evaluation or grading systems.

### **3.14.2 Requirement 2: Access Project's Documents**

**Introduction** This function allows External Evaluators to securely access all documents related to the Final Year Projects assigned to them. It ensures that evaluators can review proposals, progress reports, final reports, presentations, and supporting materials before completing evaluations. The system guarantees that only authorized evaluators can view project documents, maintaining confidentiality and integrity.

#### **Inputs**

- Evaluator ID: University-provided credentials to authenticate the evaluator.
- Project ID: Identifier of the assigned project.
- Document Type: Specifies the type of document to be accessed (proposal, report, presentation, etc.).

#### **Processing Input Validity Checks:**

1. System verifies evaluator credentials and confirms project assignment.
2. Checks if the requested document exists and is in a supported format (PDF, DOCX, PPTX).

#### **Sequence of Operations:**

1. Evaluator selects the assigned project from their dashboard.
2. System retrieves all associated project documents from the database or file storage.
3. Documents are displayed or downloaded securely for evaluation purposes.

#### **Abnormal Situations:**

1. Attempt to access unassigned project documents triggers access denial.
2. Missing or corrupted documents generate an error message.

Parameters Affected: Document access logs, timestamps, and evaluator activity logs.

Degrade Operation: System may experience slower document retrieval during peak load or network issues.

Methods Used: Secure role-based access control (RBAC), encrypted storage, and secure file transfer protocols.

## Outputs

- Display or download of project documents to the evaluator.
- Error messages for inaccessible or missing documents.

**Performance Requirements Static:** System shall allow evaluators to access all documents of assigned projects securely.

**Dynamic:** 99% of requested documents shall be accessible within 2–5 seconds under normal conditions.

**Design Constraints Standards Compliance:** Must comply with university confidentiality and document management standards.

**Hardware Limitation:** Function operates on existing servers and storage infrastructure.

**Attributes Availability:** Function should be available during evaluation periods.

**Security:** Documents are accessible only by assigned evaluators; all access is logged and encrypted.

**Maintainability:** Document access modules can be updated without affecting other system functionalities.

**Transferability/Conversion:** Can integrate with future document management or grading systems.

### 3.14.3 Requirement 3: View Evaluation Deadlines and Schedules

**Introduction** This function allows External Evaluators to view all important deadlines and schedules for the evaluation of Final Year Projects assigned to them. It ensures timely evaluation and proper planning, allowing evaluators to meet university standards and submit assessments within the specified periods.

## Inputs

- Evaluator ID: University-provided credentials to authenticate the evaluator.
- Project ID: Identifier for the assigned project(s).

- Schedule Type: Type of schedule or deadline (submission, evaluation, feedback).

### **Processing Input Validity Checks:**

1. System verifies evaluator credentials and project assignments.
2. Ensures that schedule data exists for the assigned project.

### **Sequence of Operations:**

1. Evaluator logs into the system.
2. Navigates to the “Evaluation Deadlines & Schedules” section.
3. System retrieves all relevant deadlines and schedules from the database.
4. Displays deadlines and schedules in chronological order, with reminders for upcoming tasks.

### **Abnormal Situations:**

1. If the evaluator is not assigned to any projects, the system displays a notification indicating no schedules available.
2. System errors or database unavailability may delay schedule retrieval.

**Parameters Affected:** Access logs, timestamps, and notification flags for upcoming deadlines.

**Degrade Operation:** During server maintenance or high load, schedule display may be slightly delayed but remains accessible.

**Methods Used:** Role-based access control (RBAC), database queries, automated reminders and alerts.

## **Outputs**

- Display of all evaluation deadlines and schedules relevant to the evaluator.
- Notifications or alerts for upcoming submission or evaluation deadlines.
- Error messages if schedules are unavailable or access is unauthorized.

**Performance Requirements Static:** System shall allow evaluators to view all project-related deadlines and schedules.

**Dynamic:** 99% of schedule requests shall be displayed within 2 seconds.

**Design Constraints Standards Compliance:** Must follow university academic calendar and evaluation policies.

**Hardware Limitation:** Operates on existing servers and database systems.

**Attributes Availability:** Function should be available throughout the evaluation period.

**Security:** Only assigned evaluators can view schedules; all access is logged.

**Maintainability:** Schedule display logic can be updated without affecting other system modules.

**Transferability/Conversion:** Can be integrated with future automated notification or calendar systems.

#### **3.14.4 Requirement 4: Submit Evaluation Marks and Remarks**

**Introduction** This function enables External Evaluators to submit their evaluation scores, comments, and recommendations for assigned Final Year Projects. It ensures fair, accurate, and timely assessment while maintaining academic standards. The system securely records the evaluations and prevents unauthorized changes once submitted.

##### **Inputs**

- Evaluator ID: Credentials provided by the university for authentication.
- Project ID: Identifier for the assigned project.
- Evaluation Form: Predefined form including grading rubrics, criteria, and fields for remarks.
- Marks and Remarks: Scores, comments, and suggestions entered by the evaluator.

##### **Processing Input Validity Checks:**

1. Verify evaluator credentials and project assignment.
2. Ensure that all mandatory fields in the evaluation form are completed.
3. Validate marks against allowed ranges defined by university evaluation policies.

##### **Sequence of Operations:**

1. Evaluator logs in and accesses the assigned project.
2. Completes the evaluation form with marks and remarks.
3. Submits the evaluation; the system encrypts and stores the data securely.

4. System locks the evaluation form to prevent further edits.
5. Notifications are sent to coordinators and HoD about submission.

#### **Abnormal Situations:**

1. Incomplete forms trigger error messages and prevent submission.
2. Unauthorized access attempts are logged and blocked.
3. System or network failure during submission may temporarily prevent data entry; retry options are provided.

**Parameters Affected:** Project evaluation records, submission timestamps, audit logs.

**Degrade Operation:** Submission may be delayed during high server load but is queued for processing.

**Methods Used:** Role-based access control (RBAC), secure database transactions, audit logging.

#### **Outputs**

- Confirmation message for successful submission.
- Updated evaluation record accessible to coordinators and HoD.
- Notifications of submission sent to relevant stakeholders.
- Error messages for incomplete forms or unauthorized attempts.

**Performance Requirements Static:** System shall allow evaluators to submit marks and remarks for assigned projects.

**Dynamic:** 99% of evaluation submissions shall be stored and acknowledged within 3–5 seconds.

**Design Constraints Standards Compliance:** Must follow university evaluation policies and rubrics.

**Hardware Limitation:** Operates using existing servers and infrastructure.

**Attributes Availability:** Available during evaluation periods only.

**Security:** Marks and remarks are encrypted; only assigned evaluators can submit.

**Maintainability:** Evaluation submission logic can be updated without affecting other modules.

**Transferability/Conversion:** Can integrate with university grading systems in future.

### **3.14.5 Requirement 5: View Evaluation Criteria and Instructions**

**Introduction** This function allows External Evaluators to access the detailed evaluation criteria and instructions provided by the university for assessing Final Year Projects. It ensures that evaluations are consistent, standardized, and aligned with academic policies. Evaluators can refer to rubrics, scoring guidelines, and special instructions for each project.

#### **Inputs**

- Evaluator ID: Credentials provided by the university for authentication.
- Project ID: Identifier for the assigned project.
- Evaluation Guidelines Document: Predefined set of rubrics, scoring instructions, and academic policies.

#### **Processing Input Validity Checks:**

1. Verify evaluator credentials and assignment to the project.
2. Ensure that the evaluation criteria document exists for the project.

#### **Sequence of Operations:**

1. Evaluator logs into the system.
2. Selects the assigned project.
3. Opens and reviews evaluation criteria, rubrics, and instructions.
4. System tracks access in audit logs.

#### **Abnormal Situations:**

1. Unauthorized access triggers an error message and logging.
2. Missing or corrupted evaluation guidelines document generates a notification to administrators.

Parameters Affected: Access logs, audit trails for evaluator actions.

Degrade Operation: System may delay document retrieval under high server load; cached copies may be used to maintain access.

Methods Used: Secure role-based access control, document management system, audit logging.

## **Outputs**

- Display of evaluation criteria and instructions for the assigned project.
- Notifications of missing or updated guidelines sent to evaluators.
- Logged access history for accountability.

**Performance Requirements Static:** System shall allow evaluators to view evaluation criteria and instructions for their assigned projects.

**Dynamic:** 99% of document accesses should occur within 2–3 seconds under normal conditions.

**Design Constraints Standards Compliance:** Must follow university evaluation policies and formatting standards.

**Hardware Limitation:** Operates on existing servers and network infrastructure.

**Attributes Availability:** Accessible throughout the evaluation period.

**Security:** Only assigned evaluators may view criteria; content cannot be modified.

**Maintainability:** Evaluation instructions can be updated by administrators without affecting the evaluator interface.

**Transferability/Conversion:** System can integrate with future university evaluation platforms.

## **3.15 Coordinator Module**

### **3.15.1 Requirement 1: Manage FYP Timelines of Milestones**

**Introduction** Purpose of this function is that system makes the coordinator to manage the timeline of FYP milestones. This contains the creating, updating, scheduling and making the deadlines of project milestones.

## **Inputs**

1. **Milestone Title:** make the title of milestone of projects.
2. **Milestone Description:** add the detail of milestone of project.
3. **Document Upload:** if need document to upload then upload.
4. **Start and end date:** make the date of start to end like presentation of proposal date.

## Processing

1. **Input Validity Checks:** System checks the dates of milestone lies in the session time line and also ensure the start and end date of mile stone.
2. **Sequence of Operations:** Coordinator accessed through its dashboard. Coordinator has ability to enter or update the milestone details. System checks all input and save milestone in database.
3. **Abnormal Situations:** System checks invalid date range and duplicate of milestone. System will validation error.
4. **Parameters Affected:** This functionality affects the milestone records and submission availability.
5. **Degrade Operation:** System has low network access, maintenance of system or database and notification service may make delay.
6. **Methods used:** System uses the relational base database to make easy data access
7. **Output Validity Check:** System must confirm the visibility to accessed users.

## Outputs

1. System displays milestone to users dashboard like student, supervisors.
2. Display the error messages in term of invalid data or duplicate milestone.

## Performance Requirements

1. **Static Requirement:** The system shall support all departments.
2. **Dynamic Requirement:** 95 percent creation or modification of milestone operates in 5 second.

## Design Constraints

1. **Standard Compliance:** Milestone shall according to university timeline.
2. **Hardware Limitation:** Procedure of milestone do in running web server and database.

## Attributes

1. **Availability:** The milestone shall available 99 percent of time except maintenance.
2. **Security:** Coordinator can allowed to handle milestone not others.

### 3.15.2 Requirement 2: Assign evaluators and evaluation schedules

**Introduction** Purpose of this function is that system allows the Coordinator to assign the internal and external evaluator to FYP groups and evaluation schedules. This function makes the process evaluation schedule and evaluator according to the university assessment policies.

#### Inputs

1. **Project Group ID:** Coordinator get the groups IDs to assign the evaluators.
2. **Evaluators IDs (Faculty and External):** Assign the evaluators to projects.
3. **Evaluation Type:** Select the type of evaluation such proposal presentation or final defense.
4. **Evaluation date:** Define a schedule evaluation.

#### Processing

1. **Input Validity Checks:** System should check the evaluator availability. Check the date not conflict with academic holidays.
2. **Sequence of Operations:** Coordinator has ability to selects project groups and evaluators. Set the evaluation time and type. Saves these in database and publishes the data to student and evaluators with notifications.
3. **Abnormal Situations:** System displays the schedule conflict and shows the error when the invalid selection.
4. **Parameters Affected:** Affected the assessment records, evaluation status.
5. **Degrade Operation:** System has low network access and maintenance of system or database may make delay.
6. **Methods used:** It provides the role base access and uses relational database.
7. **Output Validity Check:** System must confirm the evaluators views the assigned projects and correctly appears on dashboards.

#### Outputs

1. System displays the schedule of evaluation on student and evaluators.
2. Display the error messages.
3. Sends notification on users who has authorized to see.

## Performance Requirements

1. **Static Requirement:** The system shall support the evaluation schedule and evaluator for each department.
2. **Dynamic Requirement:** Evaluator assignment and schedule updates in 5 second in system and display to authorized users.

## Design Constraints

1. **Standard Compliance:** Milestone shall according to university timeline.
2. **Hardware Limitation:** Procedure of assigning do in running web server and database.

## Attributes

1. **Availability:** Function shall available during working sessions with low downtime.
2. **Security:** Authorized user can set schedule and assign evaluators.
3. **Maintainability:** System shall maintain and configured without downtime of system.
4. **Transfer ability / conversion:** In future, system may be integrated with examination and scheduling system.

### 3.15.3 Requirement 3: Assign supervisors and co-supervisors

**Introduction** Purpose of this function is that system allows the coordinator to assign the supervisors and co supervisors to FYP groups. This function shows the each group get well guidance of project with according to department.

#### Inputs

1. **Project Group ID:** Get IDs of each group of project to assign the supervisor and co supervisors.
2. **Supervisor and Co supervisors IDs:** Coordinator assign the supervisor and co supervisors.
3. **Department:** Select the department to differentiate among the department of supervisors and co supervisors.

## **Processing**

1. **Input Validity Checks:** System check the groups and supervisors and co supervisors have same department.
2. **Sequence of Operations:** Coordinator select the project group and supervisors and co supervisors. Check the both have same department and system saves the data in database. System sends notification to concern users.
3. **Abnormal Situations:** System shows the error when different department group or supervisors are selected or invalid selection.
4. **Parameters Affected:** It affects the project supervision records and project status.
5. **Degrade Operation:** System has low network access and maintenance of system or database may make delay.
6. **Methods used:** It provides the role base access and uses relational database.
7. **Output Validity Check:** Supervisor can access assigned project data and correctly data appear on dashboard of supervisors.

## **Outputs**

1. System displays both details of student and supervisors on dashboard of each other.
2. Display the error messages.
3. Both get notification of assignment of project group and supervisor and co supervisor.

## **Performance Requirements**

1. **Static Requirement:** The system shall show supervisor assignment to each FYP group.
2. **Dynamic Requirement:** System shall handle the operation of assignment and updates in 5 second.

## **Design Constraints**

1. **Standard Compliance:** Assignment of supervisors shall according to university policies.
2. **Hardware Limitation:** Procedure of assigning supervisor or co supervisors in working and existing web infrastructure.

## Attributes

1. **Availability:** Function shall available during working time except the maintenance time.
2. **Security:** Authorized user can assign the supervisor and co supervisors.
3. **Transfer ability / conversion:** In future, system may be integrated with external system which university is using.

### 3.15.4 Requirement 4: Handle Issues of All Users related to FYP

**Introduction** Purpose of this function is that system allows the coordinator to manage and settle the FYP related issue raised by users that are in FYP process. Issues like projects conflict, deadlines problems, scheduling issues etc.

#### Inputs

1. **Status of issues:** Check the status of issue related to the FYP.
2. **Add in issues:** Coordinator add the issue in issue dashboard for solution.
3. **Issue resolved or not status:** shows this status to only issue created by user.
4. **Remarks:** Provides remarks on solving issue.

#### Processing

1. **Input Validity Checks:** Coordinator checks the related issue FYP and enter valid user ID user who created issue.
2. **Sequence of Operations:** User must submit the issue that display on coordinator screen and check the issue related to FYP then if related to FYP issue that can be added in issue section. Coordinator change the status when issue resolved and provide remarks.
3. **Abnormal Situations:** When issue is not related to the FYP or invalid details of issue make notification invalid detail.
4. **Parameters Affected:** It affect the issue statuses in database.
5. **Degrade Operation:** System has low network access and maintenance of system or database may make delay.
6. **Methods used:** System uses the role based access and rule based FYP workflow validation.

7. **Output Validity Check:** System must confirm the data movement in database correctly and show on dashboard.

## Outputs

1. System displays the issues status updates.
2. Display the error messages.
3. Remarks on issue of coordinators.

## Performance Requirements

1. **Static Requirement:** The system shall support the managing FYP related issues for all users related to projects.
2. **Dynamic Requirement:** 99 percent of issue updates shall process in 5 second.

## Design Constraints

1. **Standard Compliance:** Issue managing shall be according to FYP.
2. **Hardware Limitation:** Function shall operate in web based infrastructure

## Attributes

1. **Availability:** Function shall available 99 percent of time except the maintenance time.
2. **Security:** System shall handle by coordinator issue related to FYP.
3. **Maintainability:** Issues shall maintain issue categories and workflow.
4. **Transfer ability / conversion:** In future, system may be integrated with namhal to make the system accuracy.

## 3.16 Super Administrator Module

### 3.16.1 Requirement 1: Full Login System Access

**Introduction** Purpose of this function is that system allows the super admin to get complete access over all modules, data and administrative access control. This function makes the centralized control, maintenance, update and governance of system. This full access makes to see the activities of user, handle and ensure the system operation with university regulation.

## Inputs

1. **Admin ID:** Enter email which is used for account and quantity of request is one at a single time.
2. **Admin Credential:** Get a dashboard of password change or reset through email ID for limited time. Enter new password.

## Processing

1. **Input Validity Checks:** System should check the authorize and authenticate of super admin and validate the credential to provide overall access of system.
2. **Sequence of Operations:** Super admin login the system. It checks the privileges and provide the access admin module to super admin.
3. **Abnormal Situations:** When user attempt to login authorized to deny access and invalid administration credential to display errors. When system fails to preserve the system state and notify the admin.
4. **Parameters Affected:** Affected the roles and access of system.
5. **Degrade Operation:** System has low network access and maintenance of system or database may make delay and if system in critical mean main services down it should provide read only access.
6. **Methods used:** System used the role based access control and secure authentication.
7. **Output Validity Check:** System must confirm the changes over system accurately.

## Outputs

1. System displays the system access confirmation and
2. Display errors when its access denial or invalidation.

## Performance Requirements

1. **Static Requirement:** The system shall support the one super admin session at time.
2. **Dynamic Requirement:** System proceed the admin operations in 5 seconds.

## Design Constraints

1. **Standard Compliance:** All super admins are traceable when login system.
2. **Hardware Limitation:** Procedure in web based infrastructure.

## Attributes

1. **Availability:** Function shall available at operational hours.
2. **Security:** Admin access protected by strong factors authentication.
3. **Maintainability:** System shall modify able when not in downtime where possible.
4. **Transfer ability / conversion:** Admin module shall support integration university software admin system.

### 3.16.2 Requirement 2: Manage All user Roles and Permission

**Introduction** Purpose of this function is that system allows the super admin to manage all users role and permissions in the system. This Functionality can provide access data relevant users like initial proposer shows to supervisor no to others users and this ensure the proper access control and security of system.

#### Inputs

1. **Module selection:** Add a functionality to provide the access according to role.
2. **Permissions:** Add permissions to user what do.

#### Processing

1. **Input Validity Checks:** System should verify admin authorization and check the status of user. Make the roles and permissions are valid.
2. **Sequence of Operations:** Super admin access the role and permission management module. Select the target user and assigned, modify or revoked the roles and permissions to user.
3. **Abnormal Situations:** When user enter invalid role to show the rejection error and user.
4. **Parameters Affected:** User access rights, system authorization table and role base permission mapping.

5. **Degrade Operation:** If system permission fails then system remain in previous configuration.
6. **Methods used:** System uses the Role base access and permission hierarchy in it.
7. **Output Validity Check:** Provide the confirmation role or permission assign or update.

## Outputs

1. System displays successful role assign or update.
2. System provides the update of permissions.
3. Display the error or failure messages.

## Performance Requirements

1. **Static Requirement:** System shall support role management for users.
2. **Dynamic Requirement:** Role and permission tasks shall done in 5 seconds in normal load.

## Design Constraints

1. **Standard Compliance:** Role and permission management shall according to university IT rules.
2. **Hardware Limitation:** Procedure work in standard server and database infrastructure.

## Attributes

1. **Availability:** Function shall available during operational hours..
2. **Security:** Admin shall have access to manage all roles and permissions.
3. **Transfer ability / conversion:** In future, system may be integrated with system which university is using.

### 3.16.3 Requirement 3: Create, Update or Delete users

**Introduction** Purpose of this function is that system allows the super admin to handle the user accounts in FYPMS. This contains creating new user , update the available user or delete the user account to ensure user records and controlled access.

## Inputs

1. **Create dashboard:** Enter full detail of user and role. Assigned the default credentials to users.
2. **Update dashboard:** Admin can access to manage user by updating the users details and credentials.
3. **Delete Dashboard:** Admin has access to delete the users from system.

## Processing

1. **Input Validity Checks:** System should check the super admin privilege and provide the access. System can access profiles data in valid place.
2. **Sequence of Operations:** System checks the admin privileges and select the user management user modules. Admin can select create, update or delete operations. System checks valid inputs and performs the action. System updates the database.
3. **Abnormal Situations:** When user add duplicate user display duplicate message.
4. **Parameters Affected:** It affect the issue statuses in database. Shows the invalid message when invalid data write.
5. **Degrade Operation:** System has low network access and maintenance of system or database may make delay.
6. **Methods used:** System uses the user account management and data validation rules.
7. **Output Validity Check:** System must confirm the account create , update or deletion.

## Outputs

1. System displays the operation confirmation message and update notification.
2. Display the error messages like invalid data and duplicate user.

## Performance Requirements

1. **Static Requirement:** The system shall support user management for all user.
2. **Dynamic Requirement:** User management tasks done in 5 second in normal conditions.

## Design Constraints

1. **Standard Compliance:** User account management should follow the international predefined laws.
2. **Hardware Limitation:** Procedure shall do in available server and processing capacity.

## Attributes

1. **Availability:** Function shall available 95 percent of time during operational time except the maintenance time.
2. **Security:** Admin can only responsible for user management not others.
3. **Maintainability:** There should be configured with system without modify source code.
4. **Transfer ability / conversion:** User management shall transfer for future up gradation.

## 3.17 Non-Functional Requirements

Non-functional requirements of the Final Year Project Management System (FYPMS) provide attributes that shows how system will ensure scalability, efficiency, security, responsiveness and reliability in academic part. These requirements support a large number of users including students, faculty members, coordinators and administrators in Final Year Project phase.

### 3.17.1 Usability Requirements

Usability requirements makes FYPMS easy to understand and use for technical and non-technical users.

**User-Friendly** The system should provide web-based interface accessible with browsers. All the tasks like submitting proposals, reviewing documents, and assigning evaluators should reachable in less steps. Clear labels, navigation and logical workflows should make less user confusion.

**Accessibility** The system should have accessibility features like fonts which are able to read, use well and proper color contrast, and good navigation. All system messages should present in proper manner and understandable to users with different technical expertise level.

### **3.17.2 Reliability Requirements**

Reliability is essential to maintain trust in academic records and evaluation processes.

**Availability** The FYPMS should maintain an availability of 99 percent out of 100 percent or higher during the academic year. Scheduled maintenance should be planned outside critical academic periods where possible.

**Data Integrity** All system transactions should be processed reliably to prevent data loss or duplication. Project records, evaluation marks, and the approvals should remain consistent across system sessions. The system shall ensure that incomplete or failed operations do not corrupt stored data.

**Backup and Recovery** The system should make backup of data to protect the previous task after failure. In case of system failure, data should be recoverable within an acceptable time frame defined by university policy. Backup data should protect from illegal users.

### **3.17.3 Performance Requirements**

Performance requirements ensure that the FYPMS delivers smooth and timely system responses during normal and peak academic workloads, such as proposal submission deadlines and evaluation periods.

**Response Time** System responsiveness is the main thing that provide system operation time efficiency. Normal response time for user tasks such as login, viewing project details, or accessing evaluation schedules and should not exceed 2 seconds under normal running conditions and when time consuming operations like uploading project documents should be done in 5 seconds, depending on file and network connectivity. During peak time when user submit like proposal submission deadlines or final evaluations, system response time should not degraded by more than 15 percent compared to normal conditions.

**Concurrent User Support** The FYPMS is mainly designed to support concurrent access by the multiple stakeholders. The system should support at least 2,000 active users simultaneously, including students, supervisors, evaluators, and also coordinators. Concurrent access to the shared resources such as the main project documents and evaluation schedules should not cause system crashes or any kind of data inconsistency. Load management mechanisms should ensure stable performance during the high-usage periods.

**Data Throughput** Efficient handling of the sensitive academic data is essential for smooth system operation. The system should handle all kind of multiple concurrent database transactions related to project submissions, approvals, and evaluations without any noticeable delays. The document uploads, including proposal files and final reports, should support file sizes defined by university policy while maintaining acceptable upload performance. Notification processing (announcements, deadline alerts) should be delivered to users without any significant delay.

**Peak Load Handling** System should handle the load of operations when use peak time like submission files during near deadlines. Servers and database has multiple paths to handle the users. This time other users access the their modules activities normally.

### **3.18 Scalability Requirements**

Security requirements are to protect sensitive academic and personal information either it is about the involved members or about the project.

**Enhancement of System** The FYPMS should support increasing numbers of students, projects, and academic sessions without major redesign. The system architecture shall allow future enhancements, such as integration with other university systems.

**Modular Design** The system should be designed in such manner that new features can added in method not effect the functionality. Single modules (student management, evaluation, administration) should work independently to enhance maintainability.

#### **3.18.1 Security Requirements**

Security requirements protect sensitive academic and personal information.

**Authentication and Authorization** Users should authenticate using university-provided credentials. Role-based access control should bound the functions to user permissions or roles such that student, supervisor, coordinator, admin. There should get access of data to authorized user.

**Data Protection** Important data like passwords and evaluation results should store it securely and provide protection. Data movement between users and system shall protected from unauthorized access.

**Privacy and Compliance** The system should comply with university data protection and privacy policies. Access to student academic data shall be limited to authorized academic personnel only.

## 4 External Interface Requirements

This section describes all external interface requirements of the Final Year Project (FYP) Management System. These requirements explain how the system will interact with users, hardware, software, databases, and communication networks. The goal is to ensure smooth operation, secure communication, and proper data exchange between the system and external entities.

### 4.1 User Interfaces

The FYP Management System will be a web-based application accessed through a standard web browser. Dashboard will given to students, supervisors, coordinators and administrators related to their roles and responsibilities .

#### 4.1.1 Performance Requirements

The user interface should load pages within 2–3 seconds under normal internet conditions. User actions such as form submission, project registration, and status updates should be processed efficiently without noticeable delay.

#### 4.1.2 Design Constraints

**Standards** System interface follow the web rules and standard such as HTML5, CSS3, and JavaScript to ensure compatibility across modern browsers. The layout will be simple and easy to understand, following standard usability guidelines.

**Hardware Compliance** The system will work on standard devices including desktop computers, laptops, tablets, and smartphones. No special hardware is required other than a device with internet access.

**Limitations** The interface performance may depend on internet speed and browser compatibility. Very old browsers may not fully support all features.

#### 4.1.3 Attributes

**Security** User authentication will be required to access the system. Passwords will be stored in encrypted form. Session management will be used to prevent unauthorized access.

**Maintainability** The interface design will be modular so that changes or updates can be made easily without affecting the whole system.

#### **4.1.4 Other Requirements**

**Database** The user interface will interact with a centralized database to store and retrieve data such as user accounts, project details, supervisor assignments, and evaluation records.

**Operations** Users will be able to perform operations such as login, project submission, supervisor selection, progress updates, and result viewing through the interface. **Site Adaptation** The interface will be responsive, allowing it to adjust automatically according to different screen sizes and devices.

**Site Adaptation** The interface will be responsive, allowing it to adjust automatically according to different screen sizes and devices

## **4.2 Hardware Interfaces**

The FYP Management System does not require any dedicated hardware interfaces. It supports computing devices such that PC, laptops, mobile, Tablets. Input devices like touch screen, keyboard and mouse, will use for interaction and servers help the system application and database.

## **4.3 Software Interfaces**

The system interact should occur with these software components:

- Interact with web browsers like chrome, firefox etc.
- A backend server which connected and implemented through frameworks of javascript.
- Interact with relational database management system like MySQL.
- Email service software for password recovery.

APIs will be used for communication between the frontend and backend components of the system.

## **4.4 Communications Interfaces**

The system shall use standard communication protocols for data exchange:

- Communication between the client and server will use HTTPS.
- TLS encryption will be applied to ensure secure data transmission.
- Email communication will be used for notifications such as account creation, password reset.

## A Appendix A: Use Case Diagram

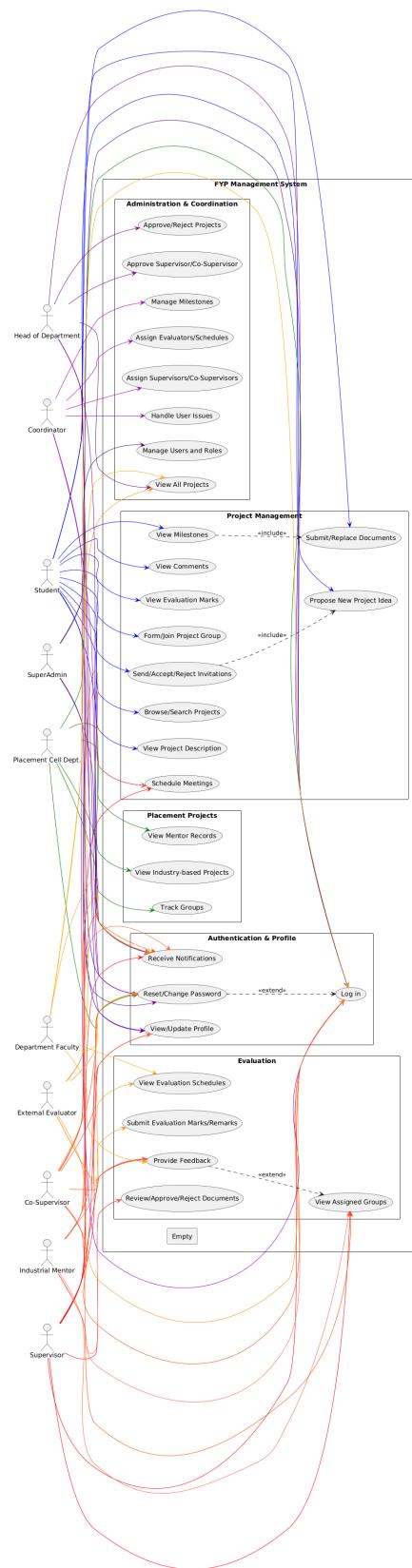


Figure 1: Use Case Diagram

## B Appendix B: Context Diagram

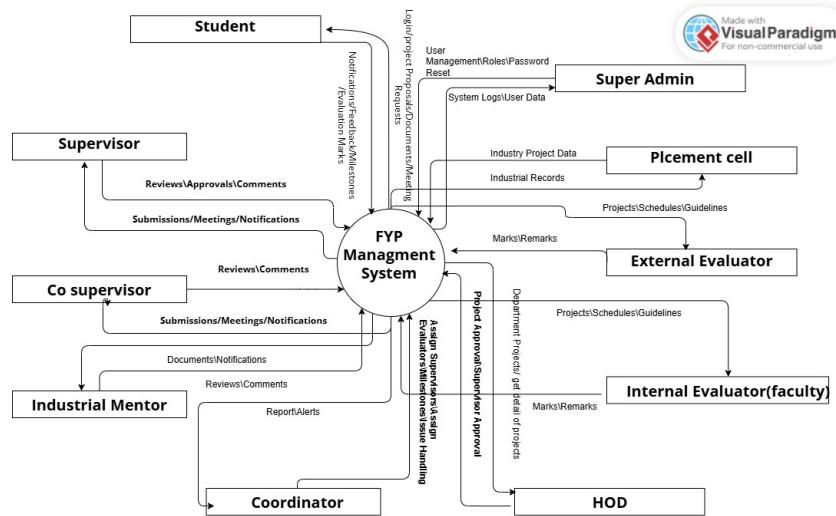


Figure 2: Context Daigram