

# **RMS**

## **(REVIEW MANAGEMENT SYSTEM)**



**BY**  
**CAPT TALHA SAEED**  
**CAPT SALMAN KHALID**  
**CAPT ISHAQ**  
**CAPT DILBER ZAIDI**

**SUPERVISOR**  
**ASST PROF MOBEENA SHEHZAD**

Submitted to the Faculty of Software Department  
National University of Sciences and Technology, Islamabad  
in partial fulfillment for the requirements of a B.E Degree in  
Computer Software Engineering

**June 2021**

## **CERTIFICATE OF CORRECTION & APPROVAL**

*This is to officially state that the thesis work contained in this report titled “RMS” (Review Management System) Is carried out by: **Muhammad Talha Saeed, Salman Khalid, Muhammad Ishaq, Dilber Zaidi** under my supervision and that in my judgment, it is fully ample, in scope and excellence, for the degree of Bachelor of Computer Software Engineering from National University of Sciences and Technology (NUST).*

**Approved By:**

**Signature:** \_\_\_\_\_

**Supervisor: Asst Prof Mobeena Shehzad**  
**MCS, NUST Rawalpindi**

## **DECLARATION OF ORIGINALITY**

We hereby declare that the work contained in this report and the intellectual content of this report are the product of our work. This thesis report has not been formally published in any structure nor does it include any verbatim of the published resources which could be treated as violation of the international copyright decree. We also affirm that we do recognize the terms ‘plagiarism’ and ‘copyright’ and that in case of any copyright infringement or plagiarism established in this thesis, we will be held fully accountable of the consequences of any such violation.

## **Plagiarism Certificate (Turnitin Report)**

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached at the end of the document.

**Signature:** \_\_\_\_\_

**Supervisor:** *Asst Prof Mobeena Shehzad*

## ACKNOWLEDGEMENTS

In the name of Allah, the most merciful and the most Beneficent. Who led us to this extent. May all glory, honor and Adoration be unto Thy Name.

Our special thanks go to our supervisor **Asst Prof Mobeena Shehzad** for guiding us throughout the process that resulted in the successful completion of our project. We would also like to thanks to the faculty of Software Department specially **Maj Zeeshan Zulkifl, Sir Ather Zaidi** for their guidance in the building of the project.

A deep gratitude towards **Dr. Adnan Ahmed Khan** (Head of Computer Software Department) for his guidance and facilitation for the Project.

At last, we are most obliged to our Parents, their support contributed immensely to the success of this project.

*Dedicated to my exceptional parents and adored siblings whose  
tremendous support and cooperation led me to this wonderful  
accomplishment*

## **ABSTRACT**

RMS review management system enable students to submit FYP and supervisors to evaluate FYP. Previously on paper evaluation was done on paper and there were no checks to break conflict of interest. In this system Roles and logins will be given to faculty members. One faculty members can have more than one role at a time and can switch role. Students have their separate console where they can see consultation hours of supervisors, previous projects details, viva schedules, download files and submit FYP proposal. Proposal can only be submitted by the student if he/she has consent of supervisor. After submission FYP proposal appears in supervisor login, he reviews proposal and submit. FYP Coordinator makes committees for Internal Evaluation and External Evaluation of projects. All committee members are notified via emails and these committee members can evaluate project from their logins by using the role of Evaluator. During this whole cycle supervisor keeps records of meetings with students of each project and at the end only those projects will appear for final external viva who have complete recurring report signed by their supervisors and HoD. Reviews by every committee is sent to each member of project including supervisor via emails. After final evaluation lists are generated by the system for result preparation. The purpose of this document is to provide comprehensive requirements and goals for the RMS. This document helps the reader to visualize the intent of the developers and how they expect the program to run and react to input from the users. This document is the documentation and report of the project that was made during the final year of the degree. It will also include a set of use cases that describe interactions the users will have with the software. Finally, this document will also be referenced during implementation and testing of the final system, to be performed at a later iteration.

## Table of Contents

Plagiarism Certificate (Turnitin Report)	iv
Chapter 1	1
1 Introduction	2
1.1 Intended Audience and Reading Suggestions	2
1.2 Motivation	3
1.3 Project Vision	3
1.4 Project Objective	3
1.4.1 Primary Objectives:	3
1.4.2 Academic objectives:	4
1.4.3 Application / End – goal objectives:	4
1.5 Deliverables	4
Chapter 2	5
2 Literature Review	6
2.1 Introduction	6
2.2 Problem Domain	6
2.3 Related Work	7
2.4 Shortcomings/issues	7
2.5 Proposed Project	8



2.6	Deliverables	8
2.6.1	Software Requirement Specification (SRS)	8
2.6.2	Software Architecture Document	8
2.6.3	Software Design document	9
2.6.4	Implementation code Document	9
2.6.5	Software Testing Document	9
2.6.6	Final Project Report	9
2.6.7	User Manual	10
2.7	Technological Requirements	10
2.7.1	Software Interfaces	10
2.7.2	Hardware Interfaces	10
2.7.3	Communications Interfaces	11
2.7.4	Programming Interface	
2.8	User Classes and Characteristics	
2.9	Operating Environment	
2.9.1	Technology Platform	
2.9.1.1	Web Based Front End	
2.9.1.2	Programming Languages	
2.9.1.3	Programming Environment	
2.9.1.4	Database	

2.10	Design and Implementation Constraints	
2.11	User Documentation	
2.12	Assumptions and Dependencies	
Chapter 3		12
3	Overall Description	12
3.1	Product Perspective	12
3.2	Product Functions	13
3..2.1	Supervisor Role	13
3..2.2	Coordinator Role	13
3..2.3	Internal Evaluator 7 <sup>th</sup> Role	13
3.2.4	Proposal Evaluator	13
Chapter 4		
4.1	System Features	19
4.1.1	Use Case UC1: Sign In	20
4.1.2	Use Case UC2: Set Limits	20
4.1.3	Use Case UC3: Set Calender	20
4.1.4	Use Case UC4: Setup Committee	20
4.1.5	Use Case UC5: Allocate Project	20
4.1.6	Use Case UC6: Add/View/Edit Project	20
4.1.7	Use Case UC7: Upload Student/Faculty List	20

4.1.8	Use Case UC8: Conduct Meetings	20
4.1.9	Use Case UC10: Upload SRS	21
4.1.10	Use Case UC10: Evaluate Project	21
4.1.11	Use Case UC11: Accept/Reject Proposal	21
4.1.12	Use Case UC12: View Notice Board	21
4.1.13	Use Case UC13: Search Previous FYPs	21
4.1.14	Use Case UC14 : Download Files	21
4.1.15	Use Case UC15: Logout	21
4.2	Other Non-functional Requirements	22
4.2.1	Performance Requirements	22
4.2.2	Safety Requirements	22
4.2.3	Security Requirements	22
4.2.4	Software Quality Attributes	23
4.2.5	Business rules:	23
4.3	Software Quality Attributes	23
4.3.1	Runtime System Qualities	
4.3.2	Functionality	
4.3.4	Usability	
4.3.5	Non Run-Time Qualities	
4.3.6	Modifiability	

4.3.7	Portability	
4.3.8	Testability	23
Chapter 5		25
5	System Design Specifications	26
5.1	System Architectural Design	
5.1.1	Presentation Layer	
5.1.2	Web Service Layer	
5.1.3	Data Access Layer	26
5.2	Class Diagram	28
5.3	Use Case Diagram	31
5.3.1	Use Case UC1: Sign Up	32
5.3.2	Use Case UC2: Supervisor Consent	33
5.3.3	Use Case UC3: Proposal Submission	35
5.3.4	Use Case UC4: Committee Setup	36
5.3.5	Use Case UC5: Project Evaluation	37
5.3.6	Use Case UC6: Meetings	38
5.3.7	Use Case UC7: Upload Faculty List	39
5.3.8	Use Case UC8: Upload Students List	40
5.3.9	Use Case UC10: Supervision List	42

5.4	Sequence Diagrams	51
5.4.1	User Sign-Up	51
5.4.2	Set Limit	52
5.4.3	Set Calender	53
5.4.4	Upload student Faculty list	54
5.4.5	Setup Committee	55
5.4.6	Upload Proposal	56
5.4.7	Conduct Meetings	57
5.5	Activity Diagrams	58
5.5.1	Coordinator	57
5.5.2	Student	58
5.5.3	Supervisor	59
5.5.4	Project Evaluator	60
5.6	Design Rationale	64
5.7	Component Design	65
Chapter 6		74
6	System Implementation	75
6.1	Technology Used	75
6.1.1	Programming Language Used	75

6.1.2	Development Tools	75
6.1.3	Database	75
6.1.4	Operating System	75
6.2	Complete System Implementation	75
6.2.1	Student Console	76
6.2.2	Faculty Login Module	77
6.2.3	Evaluators Dashboard	78
Chapter 7		83
7	System Implementation	84
7.1	Overview	84
7.2	Unit Testing	84
7.3	Login Feature Testing	84
7.4	System Reliability Testing	85
7.5	Notify Members	87
7.6	Supervisor Consent	89
Chapter 8		91
8	Conclusion and Future Work	92
8.1	Conclusion	92
8.2	Future work	92

Glossary	93
Bibliography	95





# Chapter 1

# 1 Introduction

RMS is web based system, which gives the users facility to submit and evaluate final year projects online. The audience of this system will be students and teachers. The current FYP evaluation cycle is based on paperwork in which users have to maintain all information about FYP and same projects have been created by multiple students without any check.

In this project we develop a comprehensive solution that shall not only maximize the speed of the process but also shall provide fair means of evaluation.

RMS will automate the whole process of proposal submission to project evaluation. Student can submit proposal from any remote area by providing OTP given by supervisor. At the end all projects will be assigned to committees after breaking conflict of interest and committee will evaluate each project online through RMS.

## 1.1 Intended Audience and Reading Suggestions

This document is meant for all the stake holders.

- **Project Supervisor:** It will assist to supervise the project and guide the group available by better manner.
- **Development Team:** It will assist the developer to develop the application and to discover back the functional requirements.
- **Testing Team:** It yearns for assistance to testers to comprehend the constraints.
- **Complainers (Users):** The likely participants of the product, Students and Supervisors who use this system for FYP evaluation and submission through their laptops or systems.
- **UG Project Evaluation Team:** Assessment board which will evaluate the assessment and evolution of UG Projects.
- **Staff:** The intended audience is both higher management and Project coordinator of the CS departments, who will assign roles and projects to supervisors and committee.

## 1.2 Motivation

Since mobile phones have become an intrinsic part of our daily lives, why not use them to save our time and effort in a new way. The traditional way of emergency handling service is based on

calling an ambulance through calling the respective ambulance service. Users needs to wait for the helpline and then provide there location.

To trace the location provided by the user and reach the pin point location is the most difficult job for the ambulance drivers, additionally there may arise navigation issues while navigating to the incident location.

As Careem and Uber are operating intended to the intent of offering cab essential services. We need to design a system that will provide emergency services in the same fashion with accuracy and speed.

### **1.3 Project Vision**

We propose a system that will provide core functionalities i.e. Project Evaluation. This application provides access to previous FYPs of students as well. Following are the Major functionalities of system.

- Login
- Proposal submission
- Setup evaluation committee
- Upload meetings
- Internal evaluation
- External evaluation
- Download Fyp related documents
- Proposal revisions
- Upload Lists

This way RMS will provide online evaluation hat will be accurate and on time.

### **1.4 Project Objective**

#### **1.4.1 Primary Objectives:**

- 1 Desktop platform for proposal submission
- 2 Minimize the time of finding supervisor

- 3 Minimized time & effort to evaluate projects.
- 4 To provide a fair mechanism for evaluation.

#### **1.4.2 Academic objectives:**

- 1 Development of a web based application for end users.
- 2 To have a good hand on using the web technologies i.e.Laravel PhP
- 3 To automate the FYP cycle in the university

#### **1.4.3 Application / End – goal objectives:**

- 1 To facilitate the end users of the application in terms of submission and evaluation.
- 2 To enable students to reach supervisors of project selection on time.
- 3 To save time by generating recurring reports through RMS instead of making recurring reports on paper .
- 4 To enable committee members to mark projects through the system.

### **1.5 Deliverables**

- 1 Complete working project
- 2 Web application
- 3 Documentation
- 4 Video of working of Project

# Chapter 2

## **2 Literature Review**

### **2.1 Introduction**

The Fyp management system was running manually which was a headache for supervisors, students and evaluation committee members. But now due to technological advancements we automate this system and students don't have to be physically there to submit revisions. There is now a full track record of students and supervisor meetings.

The previous work that has been done manually on forms needs to be kept safe. But in our proposed solution, we have designed something that gives facility to the user to complete the final year project online.

### **2.2 Problem Domain**

Keeping in view the current evaluation system, a student faces difficulty to find a supervisor who has a vacant seat and is ready to accept a new project. Students don't know about the availability of supervisors for meetings about projects. Users may incur many problems in this whole process.

This process incurs many problems, like:

- 1** Inability of the user to keep track of all meetings held between students and supervisor.
- 2** Inability of the user to make sure the project has all new features and not been built previously by any group.
- 3** Inability of coordinator to make sure projects of one committee are assigned to others.

It urges the use of modern technology to minimize the effects of this existing disparity by making a web-based application to enable the users to fulfill their responsibilities through the usage of an automated system.

## **2.3 Related Work**

There are some systems which are developed for this purpose of project management but we are bound with strict requirements of our department. This system has been developed according to instructions given by our supervisor.

Project Objectives that distinguishes our solution from the others:

- 1 Our application can be deployed across different universities.
- 2 It will provide a user-friendly environment.
- 3 Will minimize the gap of communication between the student and the supervisor.
- 4 Will minimize the time required to find teacher for project submission and saves a lot of time during evaluations.

## **2.4 Shortcomings/issues**

- 1 Internet connection is essential for the working of RMS on web.
- 2 The application is for web based systems only.
- 3 Currently PhD/MS students can not use this system.
- 4 The user must be a faculty member to login for this system.
- 5 The user must have atleast one role to login for this system.

## **2.5 Proposed Project**

RMS review management system is , which will enable students to submit FYP and supervisors to evaluate FYP. Roles and logins will be given to faculty members. One faculty members can have more than one role at a time and can switch role. Students have their separate console where they can see consultation hours of supervisors, previous projects details, viva schedules, download files and submit FYP proposal. Proposal can only be submitted by the student if he/she has

consent of supervisor. After submission FYP proposal appears in supervisor login, he reviews proposal and submit. FYP Coordinator makes committees for Internal Evaluation and External Evaluation of projects. All committee members are notified via emails and these committee members can evaluate project from their logins by using the role of Evaluator. During this whole cycle supervisor keeps records of meetings with students of each project and at the end only those projects will appear for final external viva who have complete recurring report signed by their supervisors and HoD. Reviews by every committee is sent to each member of project including supervisor via emails. After final evaluation lists are generated by the system for result preparation.

## **2.6 Deliverable**

### **2.6.1 Software Requirement Specification (SRS)**

The purpose of this document is to provide comprehensive requirements and goals for the RMS review management system. This document helps the reader to visualize the intent of the developers and how they expect the program to run and react to input from the users. This document is the documentation and report of the project that was made during the final year of the degree. It will also include a set of use cases that describe interactions the users will have with the software. Finally, this document will also be referenced during implementation and testing of the final system, to be performed at a later iteration.

### **2.6.2 Software Architecture Document**

In this document the overall architecture of the system is discoursed, also including the introduction of various components and subsystems. It is chiefly braced by system Architecture diagram which depicts an insider's viewpoint of the system by unfolding the high-level software components that execute the major functions to make the system operational.



### **2.6.3 Software Design document**

The Software Design Document is a record to give documentation that shall be utilized to help in programming advancement by giving the subtleties to how the product ought to be fabricated. Inside the Software Design Document are narratives and graphical documentation of the product plan for the task. It covers every single practical prerequisite and shows how they communicate with one another adroitly. The low-level design additionally appears with respect to how really we have been executing how we are going to actualize these requirements.

### **2.6.4 Implementation code Document**

The implementation code document provides details about the pseudo code for the application and project prototype.

### **2.6.5 Software Testing Document**

This document has testing modules in which there are certain test cases which depicts the correctness and accuracy of the project.

### **2.6.6 Final Project Report**

This is the thesis report which compiles all the previous and current working for the project. Thesis report provides the whole summary for the project and also gives details about each and every aspect of the project starting from introduction of the project, literature review, requirements leading to design discussions then testing and lastly future work and conclusion.

### **2.6.7 User Manual**

User Manual gives details about the use of the product. It contains details as to how to use the product. Its functionalities and details of every aspect as how that works and how to use it. User Manual is for users to get to know the product.

## 2.7 Technological Requirements

RMS entails subsequent software and hardware requirements specifications.

### 2.7.1 Software Interfaces

- 1 This system is web application which requires PC/Laptop having 64 or 32bit operating system.
- 2 RMS shall work on MySQL database management system.

### 2.7.2 Hardware Interfaces

There is no hardware specified interface for this application except PC/Laptops and Cell phones.

### 2.7.3 Communications Interfaces

- 1 The communication interface in our system uses Wi-Fi or cellular data in order to communicate with another user.

### 2.7.4 Programming Interface

Programming interfaces for project are:

- 1 PHPStorm
- 2 LocalHost

## 2.8 User Classes and Characteristics

The software has two types of users i.e. Students and Faculty. These two categories of users shall be provided with different access level to the system and its data and can perform functions assigned to their respective roles.

- **Faculty:** - Supervisor, coordinator and committee member can use this system according to their roles. A faculty member can have more than one role at a time. Each supervisor have login.

**Student:** - The major functionality of this system is we don't provide logins to each student we made a separate console which contains features as follows.

- Schedule of all supervisors
- Limits of supervisors
- Viva lists
- Previous FYPs
- Sample documents

## **2.9 Operating Environment**

OE-1: RMS back-end utility i.e. online server that can be bought and database can be maintained there. All the data will be accessed on the server and data manipulation can be done on the same server.

OE-2: RMS will be managed with MySQL database management system.

OE-3: RMS will run on any web systems with a working internet connection.

OE-4: The hardware, software and technology used should have following specifications:

- This system is web application which requires PC/Laptop having 64 or 32bit operating system.
- RMS shall work on MySQL database management system.
- There is no hardware specified interface for this application except PC/Laptops and Cell phones.

### **2.9.1 Technology Platform:**

#### **2.9.1.1 Web-Based Front End:**

RMS's front-end would be developed for web based systems, providing the users with the interface to login on the server, and manage tasks related to Final Year project. Web development tools (PHP Storm) would be used as the development environment.

#### **2.9.1.2 Programming languages:**

1      Laravel

#### **2.9.1.3 Programming Environment**

.1      PHP Storm

.2      Visual Studio

.3      LocalHost

#### **2.9.1.4 Database**

1.1    MySQL Database

## 2.10 Design and Implementation Constraints

- C-1: Website compatible platform is needed for the end-user.
  - C-2: Deficiency of user-skill in using the product on desktop.
  - C-3: Internet connectivity required.
  - C-4: Use of English language as the main methods for correspondence in the system.
  - C-5: User must be a faculty member for login in.
- Network connectivity issue:
- C-7: in case of inability to exchange data using the network, trigger error message Connection not available
  - C-8: if there should arise an occurrence of not capable to access services of system hardware
  - C-9: If e.g. supervisor project's count reaches the limit and he tries to add another FYP it shows the error message
  - CO-11: Database: mySQL database system

## 2.11 User Documentation

- UD- version 1.0 :Ultimate publication shall be complemented with a user manual to enlighten users how to use RMS. User documentation that would be delivered along with the final product
  - User manual

## 2.12 Assumptions and Dependencies

- AS-1: Basic assumption for development of RMS is that the system should be available 24/7 since users can submit or evaluate project at any time.
- AS-2: The users shall not misuse the application by submitting false information about project that will affect the operation of the application.
- AS-3: Supervisors accept all proposals of any technology within the supervision limit.
- AS-4: The server shall be able to take care of large numbers of requests especially when initially launched, as there will be number of proposals submission requests.
- AS-6: it is assumed that users of RMS have an PC/Laptop accompanied by internet access.
- D-1: RMS shall be permanently dependent on the internet access without internet the application cannot be launched and used for proposal submission.
- D-2: RMS shall be entirely dependent upon the MySQL server for 24/7 access as our database system resides on the online server.

# Chapter 3

## **3 Overall Description**

### **3.1 Product Perspective**

The current FYP evaluation in University is based on the cumbersome process in which users have to keep all forms safe until final evaluation from an external examiner.

In this project we develop a comprehensive solution that shall not only maximize the speed of the process but also shall provide fair means of evaluation and marking.

### **3.2 Product Functions**

Following are the key functions of the Review management system (RMS):

- Login/Sign
- Supervisor role
- Coordinator role
- Proposal Evaluator
- Internal evaluator 7<sup>th</sup> role
- External evaluator 8<sup>th</sup> role
- Internal evaluator 8<sup>th</sup> role

User does not register as their logins are created automatically by the system. They just have to login to the system and through “My Roles” button they can access their roles.

#### **3.2.1 Supervisor Role**

Functionalities of supervisor login are as follows:

- Add/ edit/ view Fyp proposal
- Send consent
- Submit proposal to evaluation committee for review
- Submit revisions of submitted proposals
- Add meetings (recurring report)
- Upload SRS
- Add internal viva 7<sup>th</sup> marks

- Download Fyp related documents
- Change/Update password

### **3.2.2 Coordinator Role**

Functionalities of coordinator login are as follows:

- Set student limit for Fyp supervision automatically/ manually
- Set Fyp calender
- Upload student list
- Add/ view/update/delete students
- Add/ view/update/delete supervisors
- Upload faculty list
- Setup evaluation committees
- view project list
- upload files
- Change/Update password

### **3.2.3 Internal evaluator 7th Role**

Functionalities of this evaluator are as follows:

- view assigned project list
- Enter marks
- Enter comments
- Download/view SRS

### **3.2.4 Proposal Evaluator**

Functionalities of this role are given below:

- view assigned project list
- view revised proposals
- Review projects
- add comments

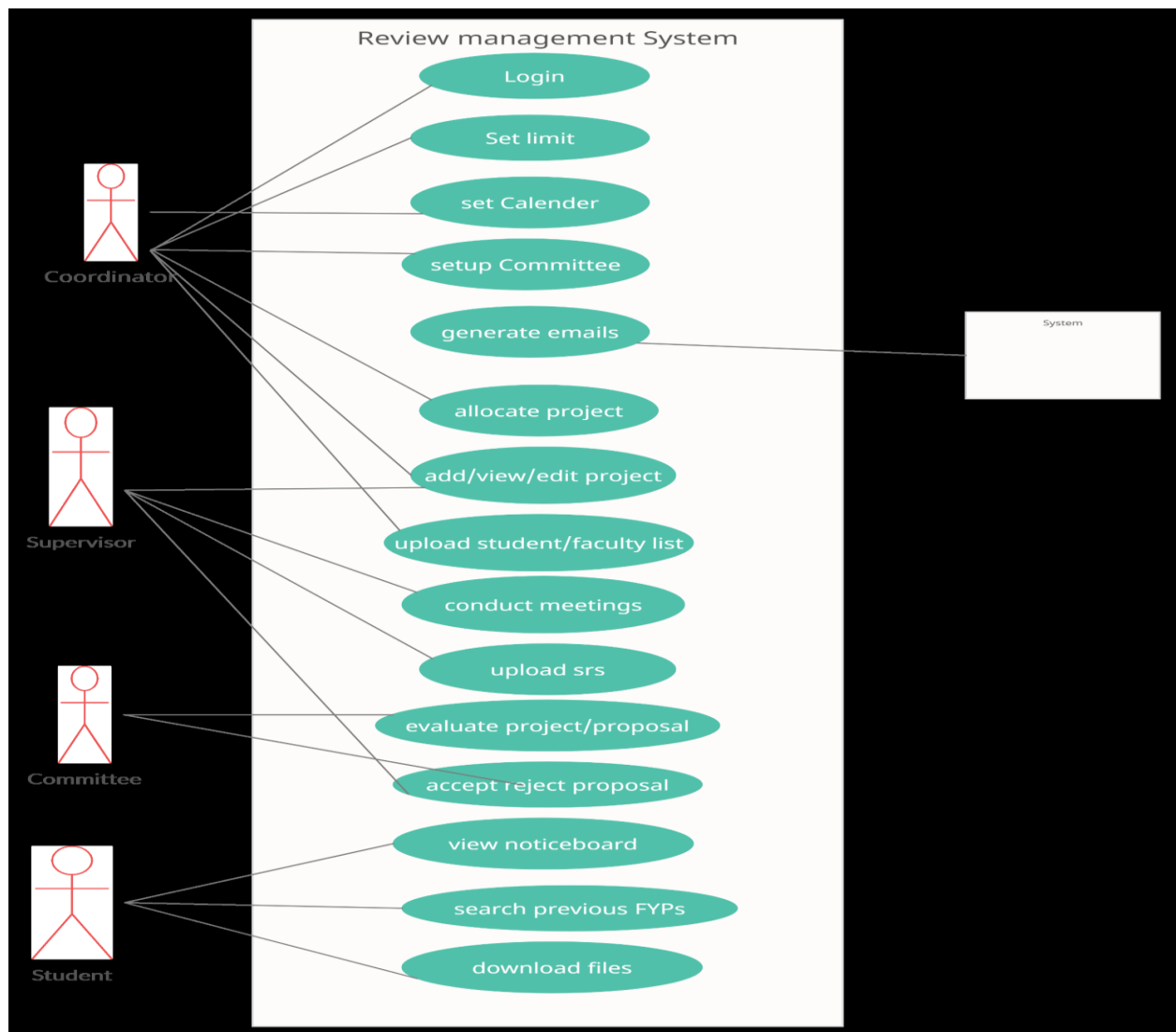




# Chapter 4

# 4 Software Requirements Specification

## 4.1 System Features



## Figure 01-Use Case Diagram

### **4.1.1 Use Case UC1: Sign In**

Users of RMS can Login to the system only if they are already registered that way its provides the authentication of credentials of legitimate users.

### **4.1.2 Use Case UC2: Set Limits**

Coordinator can set supervision limits once before assignment of proposals/projects to committee.

### **4.1.3 Use Case UC3: Set Calendar**

Coordinator can set a Calendar to set dates for proposal submission, proposal evaluation, internal viva 7th, internal viva 8th and external viva 8th.

### **4.1.4 Use Case UC4: Setup Committee**

Coordinator can only set up a committee if there are submitted proposals in the database and the calendar and supervision limit has been set already.

### **4.1.5 Use Case UC5: Allocate Project**

Projects can be allocated after breaking conflict of interest to each committee by coordinator after the setup of committees.

### **4.1.6 Use Case UC6: Add/View/Edit Project**

Students can add and view projects but supervisors can add and edit projects.

### **4.1.7 Use Case UC7: Upload student/faculty List**

Coordinator can upload students and faculty lists and he can also individually add new students and faculty members .

#### **4.1.8 Use Case UC8: Conduct Meetings.**

Supervisor can conduct meetings by entering details of specific projects like: task completion , new task assignment, responsibilities. New meeting can only be conducted if the previous meeting's task is completed.

#### **4.1.9 Use Case UC10: Upload SRS**

Supervisor can upload SRS into the system completed by students.

#### **4.1.10 Use Case UC10: Evaluate Project**

Committee will evaluate the proposal or project and submit revisions after allocation.

#### **4.1.11 Use Case UC11: Accept/reject Proposal**

Proposals submitted by students can be accepted or rejected by the supervisors after reviewing the proposal.

#### **4.1.12 Use Case UC12: View Notice Board**

Students can view notice boards for deadlines regarding proposal submissions, evaluation vivas, supervision limi and downloadable samples.

#### **4.1.13 Use Case UC13: Search Previous FYPs**

Students can view previous FYPs through the student console.

#### **4.1.14 Use Case UC14 : Download Files**

Students can download samples of SRS, Report and templates of various deliverables uploaded by coordinator on the system.

### **4.1.15 Use Case UC15: Logout**

System intend to permit user to logout of the application successfully.

## **4.2 Other Non-functional Requirements**

### **4.2.1 Performance Requirements**

- Application ought to be light weight and need to send pick up requests and booking requests instantly.
- The front-page load time must be no more than 2 seconds for users that access the application using an internet connection.
- 32bit or 64 bit Operating systems shall be able to run the application
- A secure evaluation process gives friendly advantage for user to believe on service.
- Application can handle many users and requests at a time

### **4.2.2 Safety Requirements**

- Application shall handle any user's information safely.
- Telescope package is used to keep log of users activities that if any mishap occurs service shall provide him as much support as possible.
- User credentials and private info shall not be shared with the rest of the users.

### **4.2.3 Security Requirements**

- Only authorized users can modify their profile and Project information.
- The system shall not be accessed by any unauthorized person.

#### 4.2.4 Software Quality Attributes

- **Availability:** System shall be operating on any time during working hours of the service and can manage more than one request at a time.
- **Reusability:** The components of the system shall be written in a way that they are easy to reuse.
- **Reliability:** The system defect rate shall be as less as possible.

#### 4.2.5 Business rules:

- Application provides services and can be deployed in different campuses using a shared server.
- Users must have one role for login to the application to use the services.

### 4.3 Software Quality Attributes

Quality attributes of RMS are portrayed underneath. In the wake of these characteristics, the quality of RMS shall be enhanced.

#### 4.3.1 Runtime System Qualities

At execution RMS ought to offer its users with features that they can publish and search for the desired services. Some of the qualities that need to be counted in the development of RMS are portrayed here.

#### 4.3.2 Functionality

RMS must provide functions to publish and search the different services. RMS ought to offer the feature of authentication of users.

#### 4.3.3 Availability

RMS should be available 24/7 since the proposal can be submitted at any time.

#### 4.3.4 Usability

Usability is an important criterion in the development of RMS. The system should present all functionalities in such a way that nothing is missed by the user. The graphical user interface of the app is to be designed with usability as the priority. The app will be presented and organized in a manner that is both visually appealing and easy for the user to navigate and insanely user friendly design.

#### **4.3.5 Non-Runtime System Qualities**

These are characteristics of RMS which are required to make this software useful for further enhancements. It will also be helpful in future development as well as extending the system to different environments.

#### **4.3.6 Modifiability**

RMS ought to support modifiability in case any further enhancements or features are effortless to incorporate.

#### **4.3.7 Portability**

The system should work on WIFI as well as 3G network.

RMS should be capable of operating on various gadgets that are operating via 64 or 32 bit operating systems.

#### **4.3.8 Testability**

Various quality assessments ought to be executed so that RMS is exempt of flaws and operates agreeing to requirements.

# Chapter 5



# 5 System Design Specifications

## 5.1 System Architectural Design

Layard architecture will be used with each layer providing a set of functionalities. These layers will be composed of multiple services which will communicate with each other via message passing. Abstractly, the services can be observed as components of the complete solution. Though, on the inside, each facility is made up of software components, exactly as any other application, additionally these components can make use of each other without being aware of the internal implementation

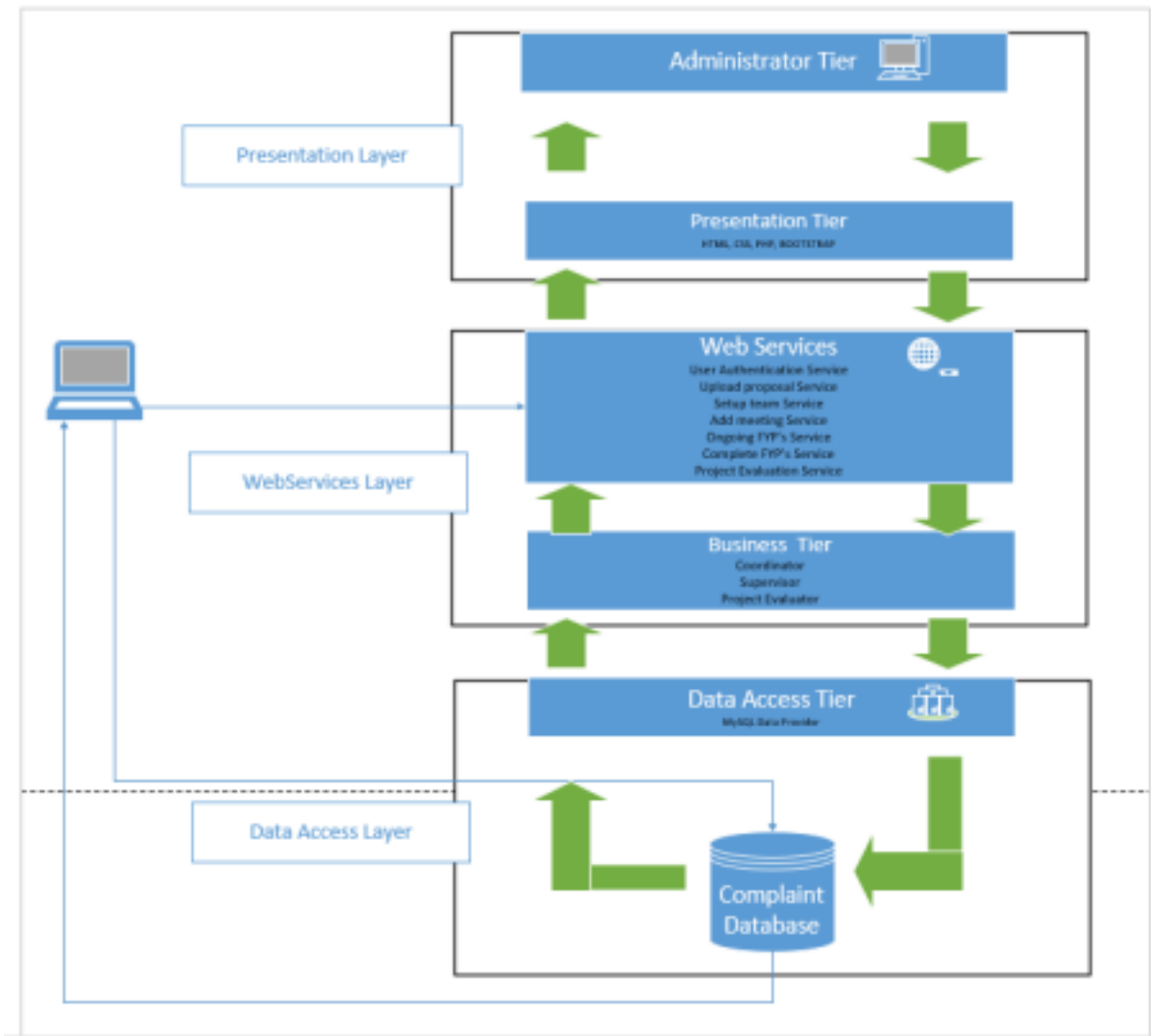


Figure 2 System Architecture

When the user of the RMS login, according to the role's user can access the functionalities. For

example, project evaluator can evaluate the fyp, supervisor can check the check the ongoing fyp's, coordinator can setup the team and all the record of the RMS are store in the database

Following is a detail of the tiers and their components which are as follow.

### **5.1.1 Presentation Layer**

This layer act as a manifesto for the communication of the user and system. This layer presents data to the user and recognizes input from the user.

### **5.1.2 Web service Layer**

It provides the core functionality to applications which a user is required to have from the application.

### **5.1.3 Data Access Layer**

This layer entertains the request from the Service Layer and sends report back data after querying from the database server.

## 5.2 Class Diagram

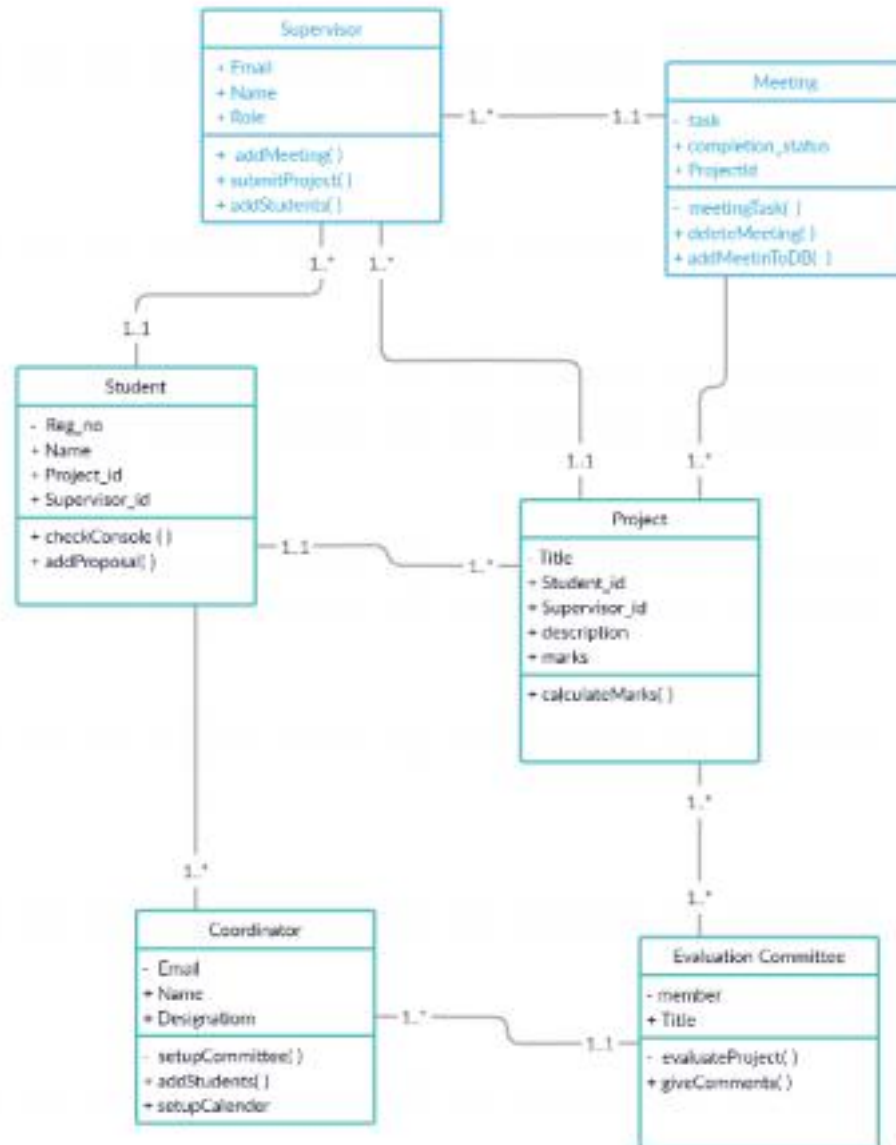


Figure 3 Class Diagram

The description for class diagram is shown on the next page:

Class name	Description
	App class contains all the information that EMTS has to perform. It is the main class which will be acting as a gateway to all the other classes
Supervisor	Supervisor class contains add, edit, view FYP proposal. This class also contains send consent, submit proposal to evaluation committee for review, submit revisions, upload SRS and Download FYP related document, add viva semester marks and can change password.
Meeting	Meeting class contains in which supervisor create meeting, student notified meeting time, and supervisor assign task of students and when student complete their assigned task. Supervisor marks this complete. Supervisor can set minimum 6 meeting according to student project. Student submit their recurring report in which all meeting details are written. Final Evaluators see all meeting details when he evaluates the students.
Student	Student class contains student console in which they can view previous FYP, download different documents, they can see their schedule when their viva conduct and student can supervisor limit and also upload their FYP proposal.
Project	Project class contains all student FYP projects, supervisor name with unique ids.
Evaluation Committee	Evaluation Committee class contains to evaluate students project and upload students marks.
Coordinator	Coordinator class contains in which they can set student limit for FYP, set FYP calendar, upload student list. It can perform all crud operation for students and supervisors and upload faculty list, setup evaluation committees and view project list, upload files

	and change or update password.

Table 1 System Features

### 5.3 Use Case Diagram

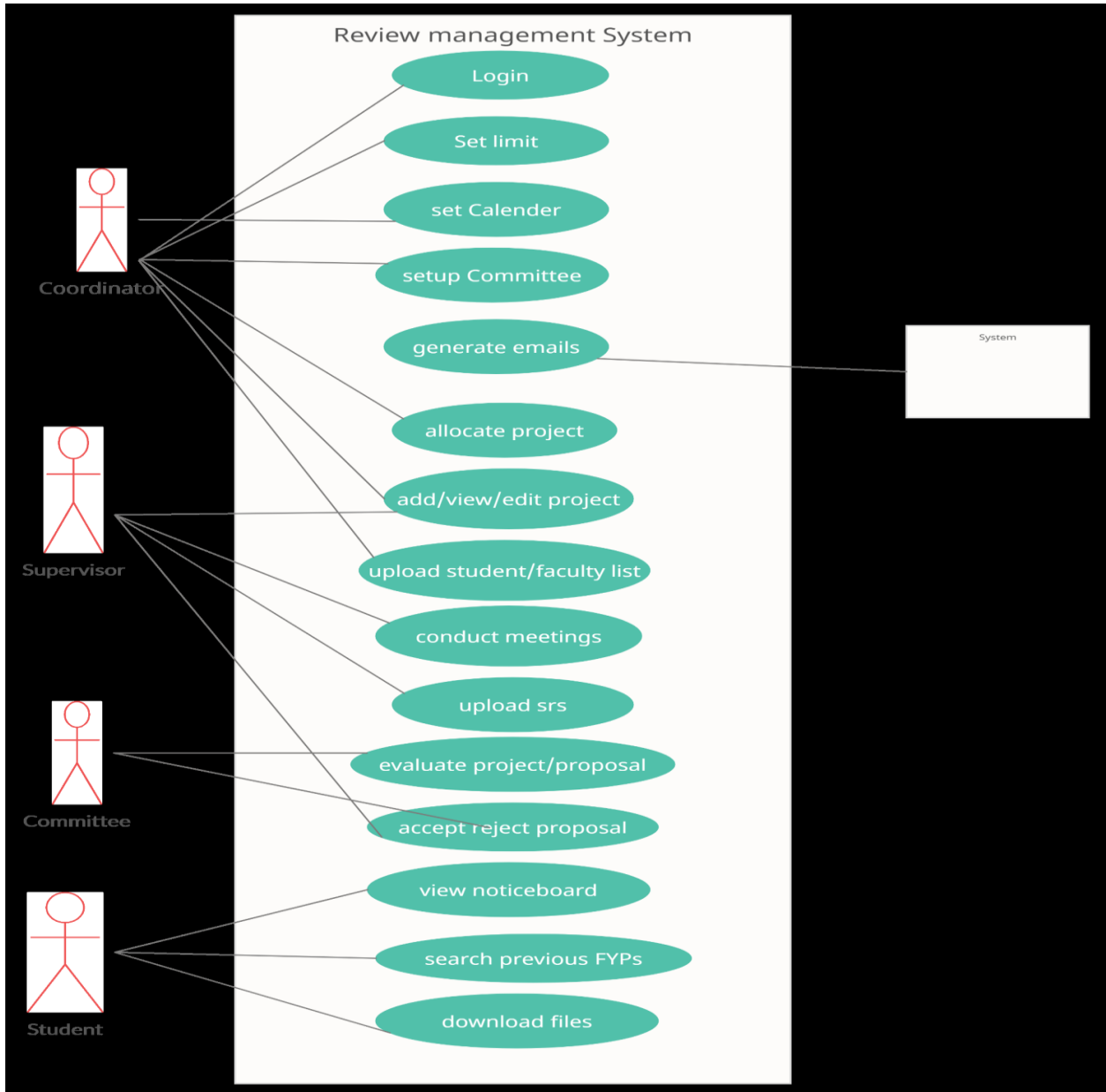


Figure 4 Use Case Diagram

### 5.3.1 Use Case UC1: Sign Up

**Name:** Sign in

**Scope:** RMS

**Primary Actor:** Unregister User

**Description:** First, Login will be done by providing the required data such as email, username and password.

**Stakeholders and Interests:** Registered user: User must login to their account.

**Preconditions:** User must have their account.

**Success Guarantee (Post conditions):**User can access that application and user can avail services now.

**Main Success Scenario:** Following Table shows main success scenario.

<i>User:</i>	<i>System:</i>
<i>Step 1: User requests to login to their account.</i>	<i>Step 2: The system asks for details and provides login form.</i>
<i>Step 3: User enters the detail e.g. username and password and press the login button.</i>	<i>Step 4: The system shall confirm the username and password from existing database.</i>
	<i>Step 5: If the username and password matches, then system login to the user account otherwise show message of invalid username/password and try again.</i>

**Table 1 Login**

## Extension

In extension we explain the alternative scenarios of use case.

### Alternative Flow 1: Username/email doesn't exist

- System shows the message that user name you entered does not exist.
- System asks the user to enter the valid username.

### Alternative Flow 2: Wrong Password

- System shows the message the password you entered is wrong/incorrect.
- System asks the user to enter the correct password.

### Alternative Flow 4: User already login

- User exits in login list so use case end.

### Alternative Flow 5: GPS is off

- If user doesn't on the GPS, the location do not show to the users. User must on GPS.

## 5.3.2 Use Case UC2: Supervisor consent

**Name:** Supervisor consent

**Scope:** RMS

**Primary Actor:** Logged In user

**Description:** Consent will be sent to students by supervisor via email.

**Stakeholders and Interests:** Logged In user: User must login to their account.

**Preconditions:** Users must have their account.

**Success Guarantee (Post conditions):** After receiving supervisor consent through mail, students can submit project proposals.

**Main Success Scenario:** Following Table shows main success scenario.

<i>User:</i>	<i>System:</i>
--------------	----------------

<i>Step 1: Supervisor enters email address of students.</i>	<i>Step 2: The system asks for students' valid email addresses .</i>
<i>Step 3: Supervisor sent constant mail to students.</i>	<i>Step 4: The system shall validate email .</i>
	<i>Step 5: If the email format is correct, the system generates email for students and sends it, otherwise it will ask for the correct email address.</i>

**Table 3 Login**

**Extensions:**

In extension we explain the alternative scenarios of use case.

**Alternative Flow 1: Email is invalid**

- System shows the message that user email you entered is invalid.
- System asks the user to enter the valid email.

**Alternative Flow 2: Empty email field**

- System shows the message the this field is required.
- System asks the user to enter the email address.

**5.3.3 Use Case UC3: Proposal Submission**

**Name:** Proposal submission

**Scope:** RMS

**Primary Actor:** Student



**Description:** Proposal will be submitted by student through their console.

**Stakeholders and Interests:** Student: Student can fill proposal submission form with supervisor consent.

**Preconditions:** User must access the application through the student console.

**Success Guarantee (Post conditions):**Students can submit proposals.

**Main Success Scenario:** Following Table 3.3 shows main success scenario.

<i>User:</i>	<i>System:</i>
<i>Step 1: The Student will request a supervisor for consent email.</i>	<i>Step 2: System will generate email.</i>
<i>Step 3: Students will enter project details in proposal submission form.</i>	<i>Step 4: If a student enters a valid details system accept the proposal form.</i>

**Table 4 Proposal Submission**

### **Extensions:**

In extension we explain the alternative scenario of use case

#### **Alternative Flow 1: Wrong input.**

- If a user enters an invalid input system will not submit a proposal.
- Enter invalid consent.

#### **Alternative Flow 2: Data loading error occur**

- If data loading error occur, then check your internet connection.
- Show message to user 'Must have internet'.

### **5.3.4 Use Case UC4: Committee Setup**

**Name:** Committee setup

**Scope:** RMS

**Primary Actor:** Loggedin user

**Description:** Coordinator setup committees for evaluation of project.

**Stakeholders and Interests:** Logged User: user can check the previously created committees.

**Preconditions:** User must login to the application, and must have coordinator role

**Success Guarantee (Post conditions):**User can access that application and create committee.

**Main Success Scenario:** Following Table 3.5 shows main success scenario.

<i>User:</i>	<i>System:</i>
<i>Step 1: The coordinator uploads student and faculty list</i>	<i>Step 2: System will check all preconditions</i>
<i>Step 3: Coordinator selects teachers and click create committee button.</i>	<i>Step 4: If teachers list is exist system make groups.</i>

**Table 5 View history**

**Extensions:**

In extension we explain the alternative scenario of use case

**Alternative Flow 1: Faculty list not exist**

If the list is not uploaded, the user will not be able to create a committee.

**Alternative Flow 3: Data loading error occur**

- If errors occur in data loading, then check your internet connection.
- Show message to user ‘Must have internet’.

### 5.3.5 Use Case UC5: Project Evaluation

**Name:** Project Evaluation

**Scope:** RMS

**Primary Actor:** LoggedIn User

**Description:** Evaluators can upload marks and comments of project during vivas and can give reviews on proposal.

**Stakeholders and Interests:** LoggedIn User: user can evaluate proposal.

**Preconditions:** User must login to the application, identified and authenticated.

**Success Guarantee (Post conditions):**User can access that application and evaluate project.

**Main Success Scenario:** Following shows main success scenario.

<i>User:</i>	<i>System:</i>
<i>Step 1: The user click project to evaluate.</i>	<i>Step 2: The system shall display form for marking.</i>
<i>Step 3: User click on 'submit'</i>	<i>Step 4: System shall accept the request.</i>

**Table 6 Project Evaluation**

**Extensions:**

**Alternative Flow 1: User doesn't have correct information**

If user enters invalid characters in marks field system shall not proceed submission and ask for valid input.

### 5.3.6 Use Case UC6: Meetings

**Name:** Meetings

**Scope:** RMS

**Primary Actor:** Students and Supervisor

**Description:** Supervisor can enter details of meetings like task given, previous task completion and time.

**Stakeholders and Interests:** LoggedIn User: user can upload meeting details.

**Preconditions:** Without fulfilling this requirement the project can not enter in final evaluation.

**Success Guarantee (Post conditions):** User can access that application and enter meeting details.

**Main Success Scenario:** Following Table shows main success scenario.

<i>User:</i>	<i>System:</i>
<i>Step 1: The user shall mark previous tasks as completed.</i>	<i>Step 2: The system shall accept meeting details</i>
<i>Step 3: The user shall enter details of new meetings.</i>	<i>Step 4: System store the meetings details.</i>

**Table 7 Meetings**

#### **Extensions:**

In extension we explain the alternative scenarios of use case

#### **Alternative Flow 1: Does not mark previous tasks as ‘Completed’.**

- If the user doesn't complete the previous task, a new meeting can not be conducted.

#### **Alternative Flow 2: Data loading error occur**

- If data loading errors occur, then check your internet connection.

### 5.3.7 Use Case UC7: Upload Faculty List

**Name:** Upload Faculty list

**Scope:** RMS

**Primary Actor:** Logged In user

**Description:** Coordinator uploads list in order to assign logins and create committees.

**Stakeholders and Interests:** Logged In user: Coordinator will be able to assign projects and students can submit proposal.

**Preconditions:** Without fulfilling this requirement coordinator can't create committees and evaluation can't be done.

**Success Guarantee (Post conditions):**User can access that application and upload list.

**Main Success Scenario:** Following shows main success scenario

**Table 3.1 Upload List**

<i>User:</i>	<i>System:</i>
<i>Step 1:Coordinator clicks the 'upload' button to submit list</i>	<i>Step 2:The system shall accept and store a list.</i>

**Table 8 Upload List**

**Extensions:**

In extension we explain the alternative scenarios of use case

**Alternative Flow 1: Insufficient internet connectivity**

- Users cannot upload list network services.
- Users need to turn on the internet.

**Alternative Flow 2: User does not have correct format of list**

- If the user does not upload the correct format, the system generates an error message.
- Users must upload a valid list.

### 5.3.8 Use Case UC8: Upload students list

**Name:** Upload students list

**Scope:** RMS

**Primary Actor:** Coordinator

**Description:** Coordinator uploads list in order to calculate supervision limit.

**Stakeholders and Interests:** Coordinator: To upload student list.

**Preconditions:** Students can't upload proposals and the limit of supervision can't be calculated without fulfilling this requirement.

**Success Guarantee (Post conditions):** User can access that application and upload list.

**Main Success Scenario:** Following Table 3.9 shows main success scenario.

<i>User:</i>	<i>System:</i>
<i>Step 1:User presses the button upload list.</i>	<i>Step 2:The system shall validate the list.</i>
<i>Step 3:User clicks submit button to upload.</i>	<i>Step 4:System stores student list.</i>

**Table 9 Alert Nearby driver**

#### **Extensions:**

In extension we explain the alternative scenario of use case

#### **Alternative Flow 1: Insufficient internet connectivity**

- Users cannot upload list network services.

- Users need to turn on the internet.

**Alternative Flow 2: User does not have correct format of list**

- If the user does not upload the correct format, the system generates an error message.
- Users must upload a valid list.

**5.3.9 Use Case UC10: Supervision Limit**

**Name:** Supervision Limit

**Scope:** RMS

**Primary Actor:** Coordinator

**Description:** Coordinator setup limit so that supervisor can supervise projects according to that limit

**Stakeholders and Interests:** Coordinator: Coordinator can set supervision limit.

**Preconditions:** Coordinator must login to the application, identified and authenticated.

**Success Guarantee (Post conditions):** Coordinator can access that application and set supervision limit.

**Main Success Scenario:** Following Table shows main success scenario.

<i>User:</i>	<i>System:</i>
<i>Step 1: The coordinator enter limit.</i>	<i>Step 2: The system shall validate input.</i>
<i>Step 3: The coordinator shall clicks 'set limit' button</i>	<i>Step 4: System shall store limit against each supervisor .</i>

**Table 10 Set limit**

## 5.4 Sequence Diagrams

### 5.4.1 User Sign-In

The below diagram defines the sequence of actions that happens when user login. A user tries to login to the system. The system requests that the actor enter his/her email and password. The actor enters his/her email and password. The system validates the entered email and password and logs the actor into the system.

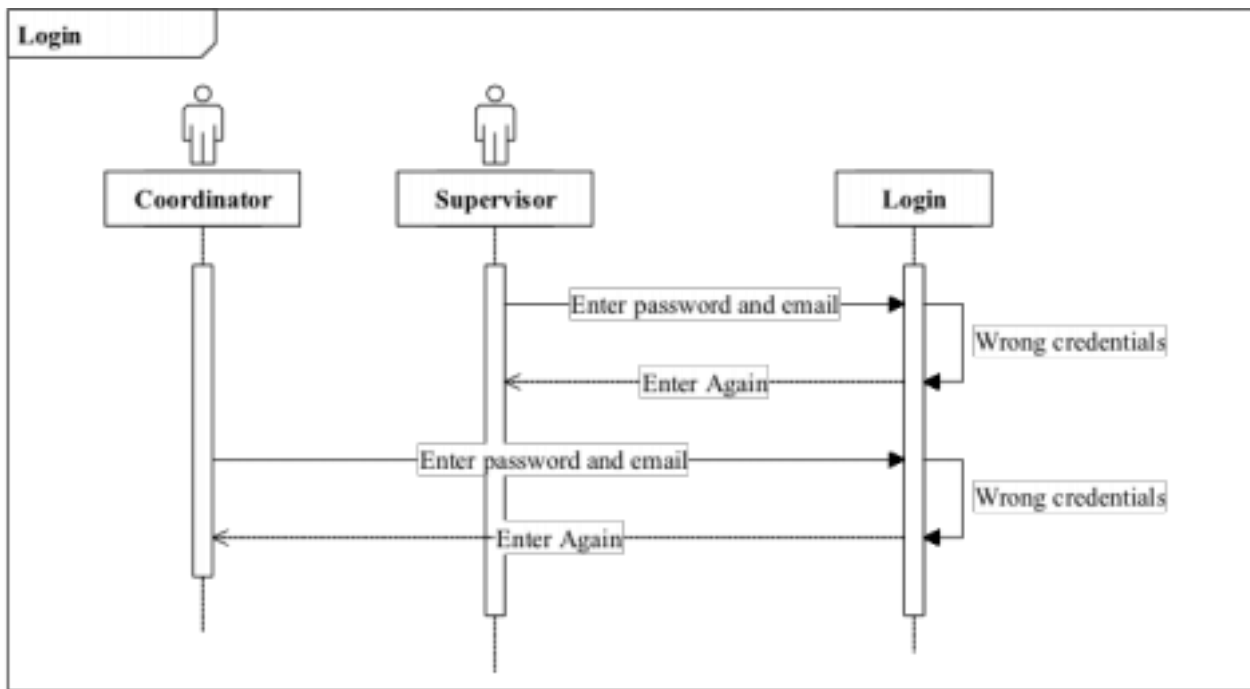


Figure 5 Sequence Diagram Sign-In



## 5.4.2 Set Limit

The below diagram defines the sequence of actions that happens when user set limit. Coordinator by pressing the button set the limit of total student taken under the supervision by each supervisor. The System must set the limit. Add FYP proposal button will be enabled for student and supervisor.

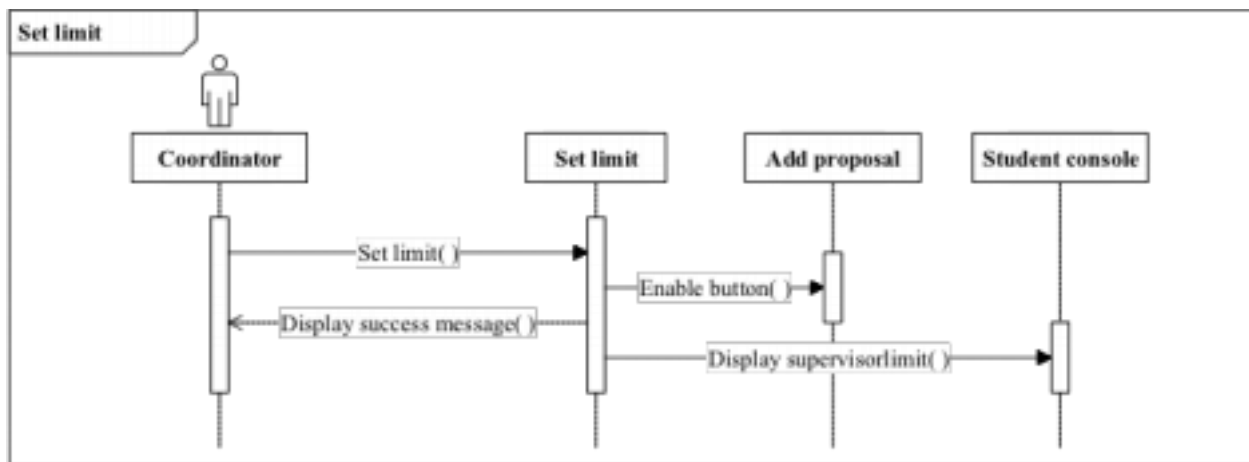


Figure 6 Sequence diagram Set Limit

### 5.4.3 Set Calender

The below diagram defines the sequence of actions that happens when user set calendar. User enter the set calendar button. The system sets calendar. The system displays dates to supervisor and student.

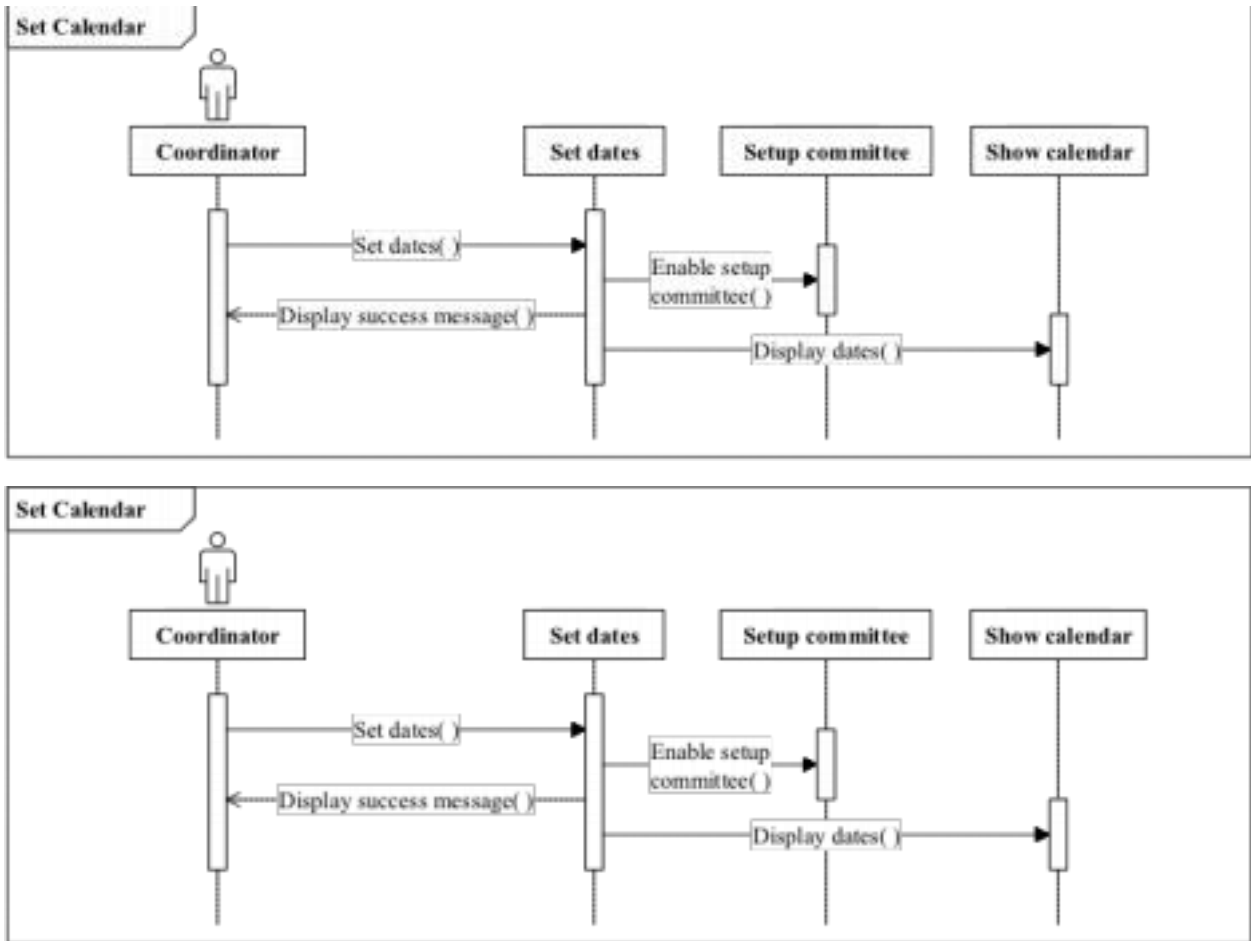


Figure 7 Sequence diagram for set calender

### 5.4.4 Upload student/faculty list

The below diagram defines the sequence of actions that happens when upload student faculty list. The user will click on the add list button. The user will upload the list. The user can set supervision limit and create committee.

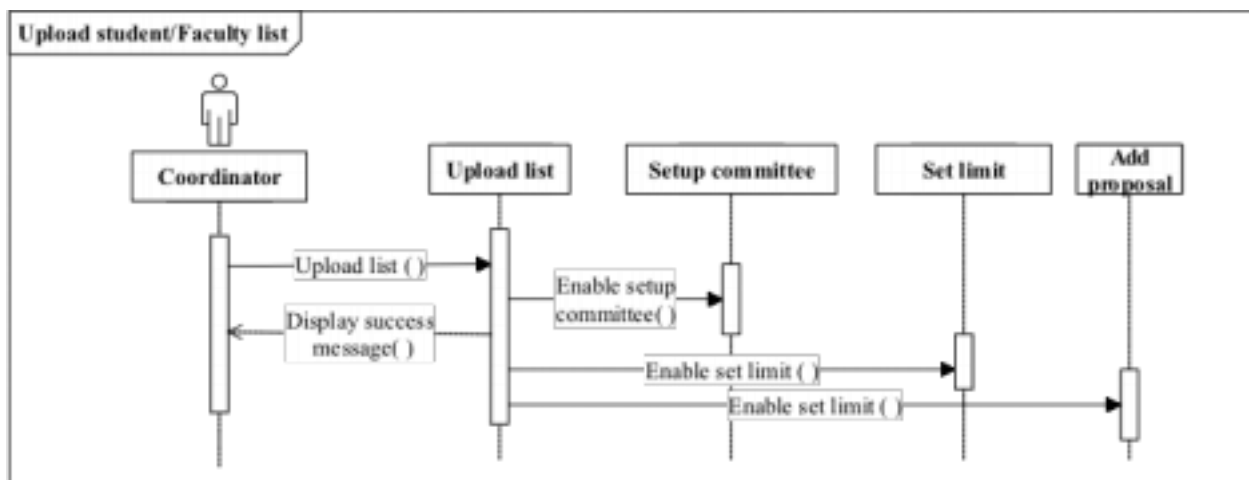


Figure 8 Sequence diagram Upload student/faculty list

### **5.4.5 Setup committee**

The below diagram defines the sequence of actions that happens when user setup committee. The user login credentials and signs in to the system. The user clicks setup committee. The system will create a committee. Then system will create groups under each committee. Emails will be sent to each member. Evaluator roll will be shown on each member login.

**Figure 9 Sequence Diagram Setup Committee**

### **5.4.6 Upload Proposal**

The below diagram defines the sequence of actions that happens when user upload proposal. The user clicks the add proposal button. Uploads necessary details of proposal. The system will update the status of the project to ongoing projects in supervisors' logins and emails will be received to each student associated with that project.

**Figure 10 Sequence Diagram Upload Proposal**

### **5.4.7 Conduct Meetings**

The below diagram defines the sequence of actions that happens when supervisor conduct meetings. User clicks on the meeting button. User will add the description. User will assign task. User will check the status of previous task. User will mark complete or save the meeting as draft. Report of each meeting will be shown to external evaluator at the time of final evaluation.

**Figure 11 Sequence diagram conduct meetings**

## **5.5 Activity Diagrams**

The below diagram defines the stream of the activities that what are the activities which are

perform by the coordinator in the web coordinator interface. It also needs to login and show all the options of the coordinator.

### 5.5.1 Coordinator:

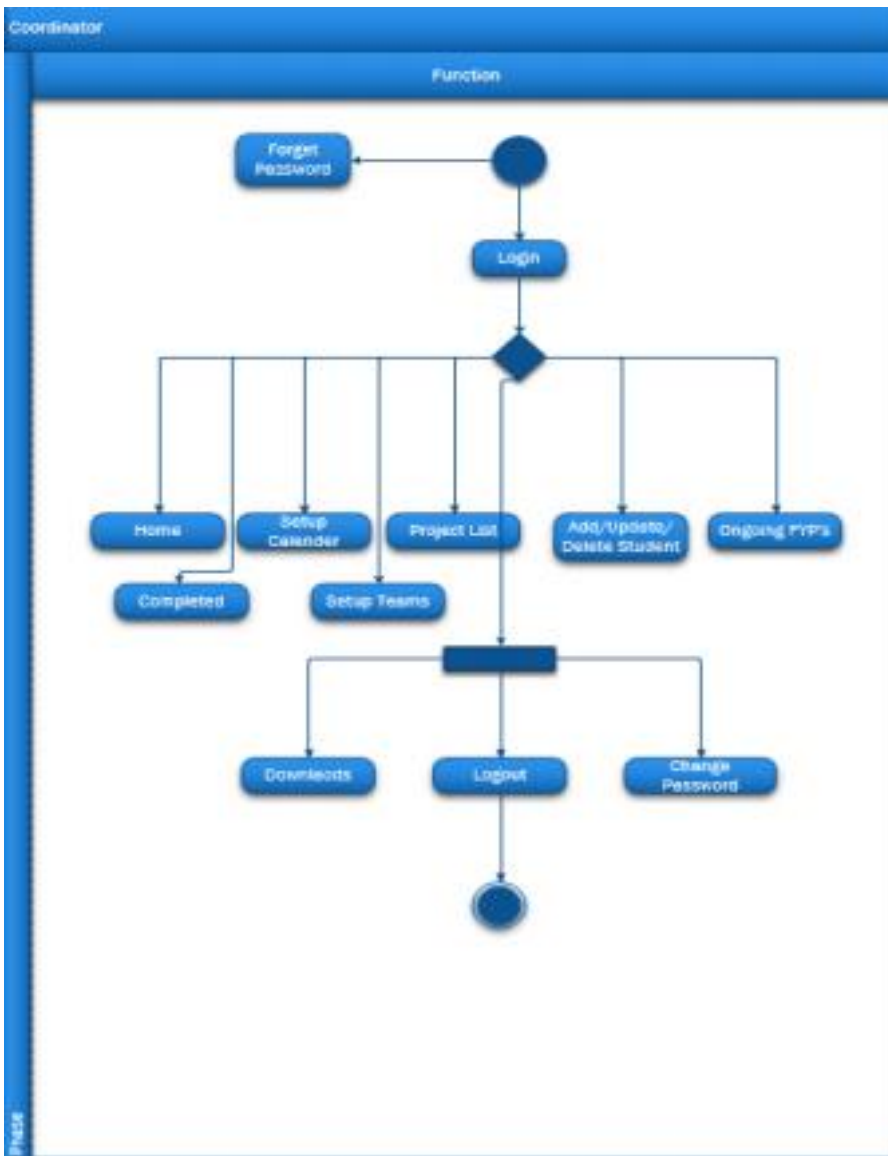
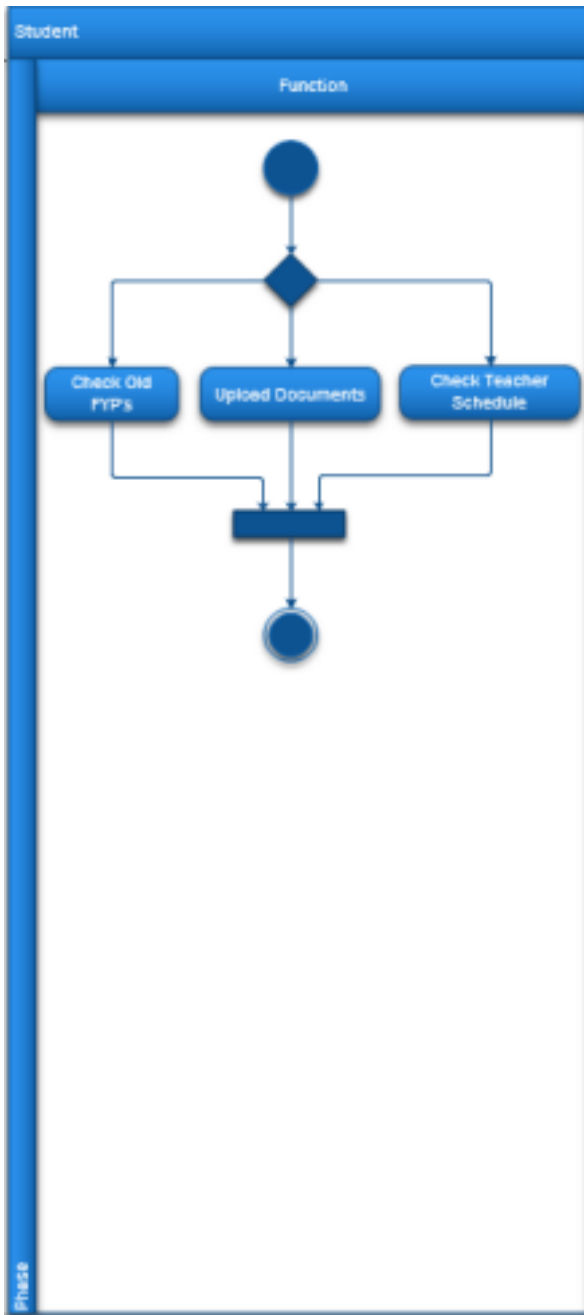


Figure 12 Activity Diagram: Coordinator

### **5.5.2 Student:**

The below diagram defines the stream of the activities that what are the activities which are perform by the student in the web student interface.

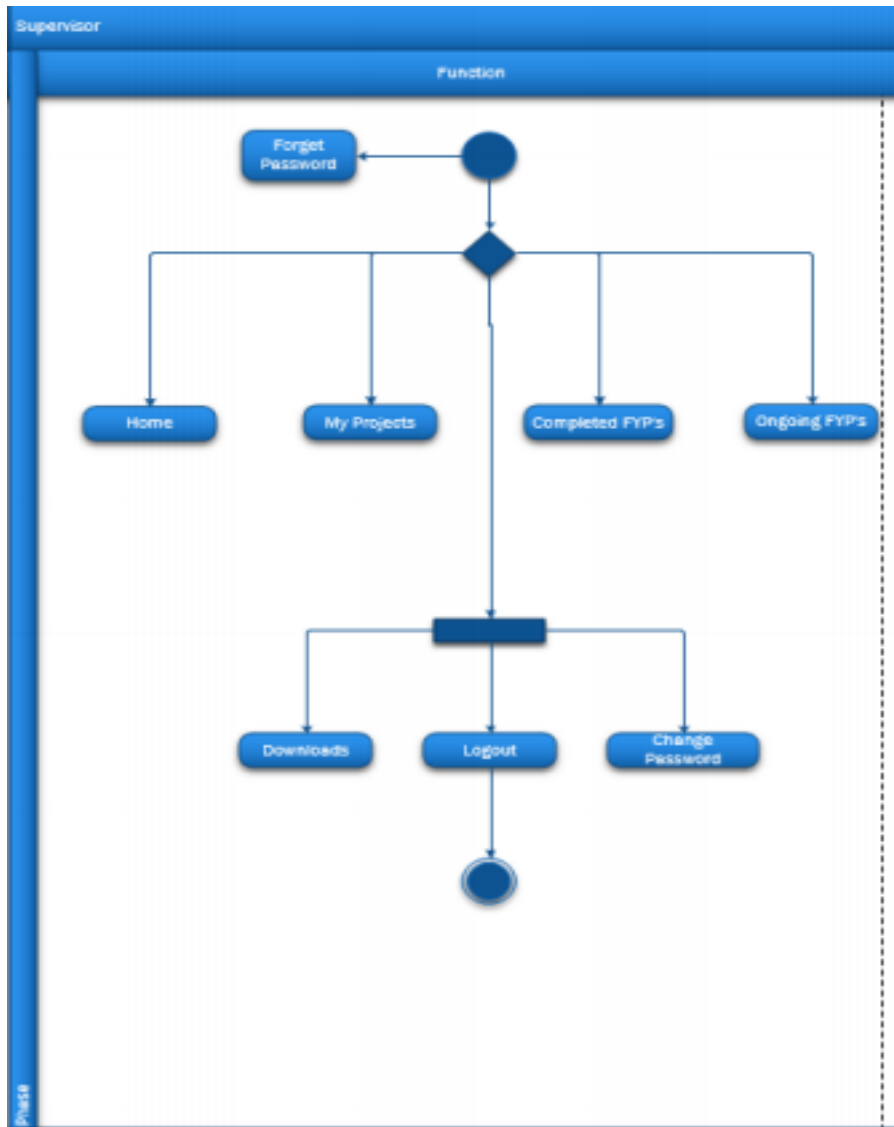




**Figure 13 Activity Diagram: Student**

### 5.5.3 Supervisor

The below diagram defines the stream of the activities that what are the activities which are perform by the supervisor in the web supervisor interface. It also needs to login and show all the options of the supervisor.



**Figure 14 Activity Diagram: Supervisor**

### 5.5.4 Project Evaluator

The below diagram defines the stream of the activities that what are the activities which are perform by the project evaluator in the web project evaluator interface. It also needs to login and show all the options of the project evaluator.



**Figure 15 Activity Diagram: Project Evaluator.**

## 5.6 Design Rationale

The layered architecture design is a strong broadly useful example, making it a decent beginning stage for most applications, especially when you don't know what architecture design is most appropriate for your application. In any case, there are a few interesting points from an architecture design viewpoint while picking the pattern.

The main thing to keep an eye out for is what is known as the architecture sinkhole anti-pattern. This anti-pattern depicts the circumstance where solicitations course through different layers of the design as straightforward go through preparing with practically zero rationale performed inside each layer. For instance, accepting the introduction layer reacts to a solicitation from the client to recover client information. The introduction layer passes the solicitation to the business layer, which essentially passes the solicitation to the constancy layer, which at that point makes

a basic MySQL call to the database layer to recover the client information. The information is then passed right back up the stack with no extra handling or rationale to total, compute, or change the information.

Each layered architecture will have probably a few situations that fall into the design sinkhole hostile to design. The key, be that as it may, is to examine the level of solicitations that fall into this classification. The 80-20 standard is normally a decent practice to follow to decide if you are encountering the design sinkhole against design. It is commonplace to have around 20 percent of the solicitations as straightforward go through preparing and 80 percent of the solicitations having some business rationale related with the solicitation. In any case, in the event that you find that this proportion is switched and a dominant part of your solicitations are basic go through handling, you should consider making a portion of the design layers open, remembering that it will be progressively hard to control change because of the absence of layer confinement.

Another thought with the layered design is that it will in general lean itself toward solid applications, regardless of whether you split the introduction layer and business layers into isolated deployable units. While this may not be a worry for certain applications, it represents some expected issues as far as sending, general strength and dependability, execution, and versatility.

## **5.7 Component Design**

We gave an Object-Oriented description in section 5.7 and summarized each object member function in all the objects created.

### **5.7.1 addProposal ()**

**Begin:**

Select add proposal option

Add all details of Proposal

If Submit button clicked:

Submitted to Committee

Else:

Save as draft

**End**

### **5.7.2 AddMeeting ()**

**Begin:**

Select Add meeting Option

Enter meeting details

If process task mark completed:

Submit meeting.

Else:

Save as draft

**End**

### **5.7.3 EvaluateProposal ()**

**Begin:**

Select Proposal

Enter Comments

If revision == true:

Not accepted

Show revision to supervisor.

Else:

Proposal accepted

**End**

## 5.7.4 SetupCommittee ()

### Begin

Select Setup Committee option

Enter Committee title

Create team or group

If project [] == true

    Assign project

Else

    Project must be uploaded

Email sent to the members

Roles will be provided

### End

# Chapter 6

## **6 System Implementation**

### **6.1 Technology Used**

#### **6.1.2 Programming Language Used**

MySQL database has been used for handling all the data storing, retrieval and fetching that is used to operate the application. The web application for the project was written using Laravel and HTML/CSS and bootstrap.

#### **6.1.3 Development Tools**

Application is developed using PHPStorm.

#### **6.1.4 Database**

The systems Database shall be designed and maintained using MySQL.

#### **6.1.5 Operating System**

Web application developed shall be able to run on all devices having operating system 64 or 32 bit.

### **6.2 Complete System Implementation**

The system comprises two main components. An android application in which separate modules are designed for driver and user i.e. passenger. The major modules of The Android Application will be discussed in detail in the succeeding sections.



## 6.2.1 Student Console

Given that student's are to be notified for upcoming deadlines and downloadable material throughout the FYP lifecycle, a Student Console is added to the application which consists of multiple options for students as follows.

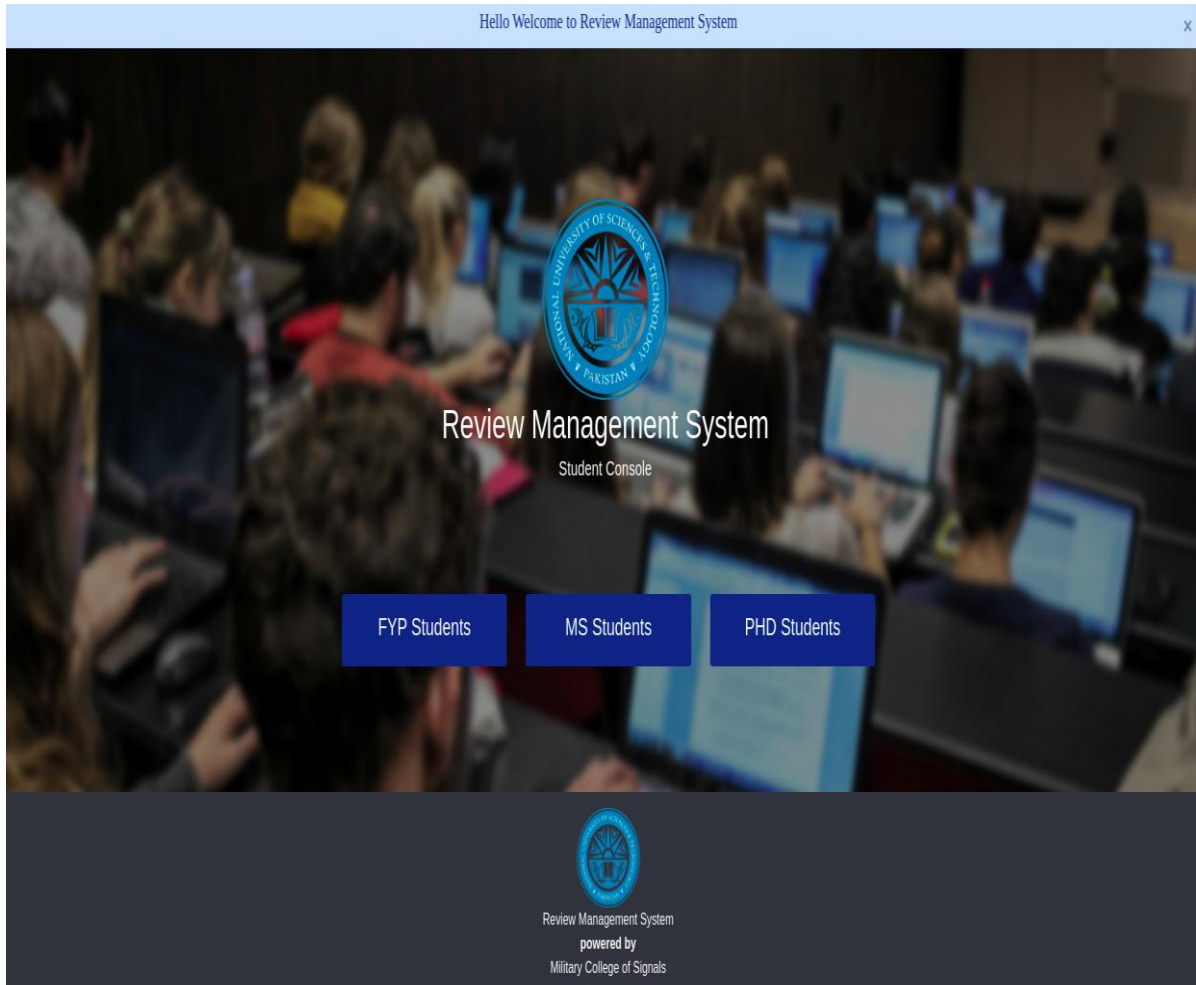


Fig: Student Console Homepage

Add Revision Add New FYP Proposal

### FYP Calendar - Session SP21

**FYP Proposal Evaluations**

Proposal Submission Starts:	Feb 8, 2021
Proposal Submission Deadline ( Final ) :	Apr 5, 2021
Proposal Evaluation Starts:	May 7, 2021
Proposal Evaluation Deadline :	May 24, 2021

**7<sup>th</sup> Semester Evaluations**

SRDS / Demo Submission Starts:	Jun 7, 2021
SRDS / Demo Submission Deadline:	Jun 18, 2021
Supervisor Marking 7 <sup>th</sup> Deadline:	Jun 21, 2021
Internal Evaluation 7 <sup>th</sup> Starts:	Jun 21, 2021 <small>(the process normally completes in 3-5 working days)</small>
Internal Evaluation 7 <sup>th</sup> Deadline:	Jun 25, 2021

**8<sup>th</sup> Semester Evaluations**

FOR INTERNAL

Powered By  
MCS Rawalpindi

Fig: FYP Schedule

1	<a href="#">Template-of FYP-proposal-submission-form-FA20.docx.docx</a> <small>Modified at: 2021-03-20 00:57:04</small>	<a href="#">Download</a>
2	<a href="#">Game-Development-Guidelines-FA20-and-onwards.docx</a> <small>Modified at: 2020-11-23 07:58:16</small>	<a href="#">Download</a>
3	<a href="#">Software Requirement and Design Specification (SRDS).doc</a> <small>Modified at: 2020-11-23 07:58:23</small>	<a href="#">Download</a>
4.1	<a href="#">01-Template-FYP-Report-PartA-IntroPages-23Nov20.docx</a> <small>Modified at: 2020-11-23 08:19:46</small>	<a href="#">Download</a>
4.2	<a href="#">02-Template-FYP-Report-PartB-Chapters-23Nov20.docx</a> <small>Modified at: 2020-11-23 08:19:59</small>	<a href="#">Download</a>
5	<a href="#">Guidelines-for-developing-FYP-Demonstration-Video.pdf</a> <small>Modified at: 2020-11-23 07:59:29</small>	<a href="#">Download</a>
6	<a href="#">Recurring-Report-for-meeting-record-Annuxure-C-03Dec18.doc</a> <small>Modified at: 2021-03-05 08:54:29</small>	<a href="#">Download</a>

Powered By  
MCS Rawalpindi

Fig: FYP Downloads

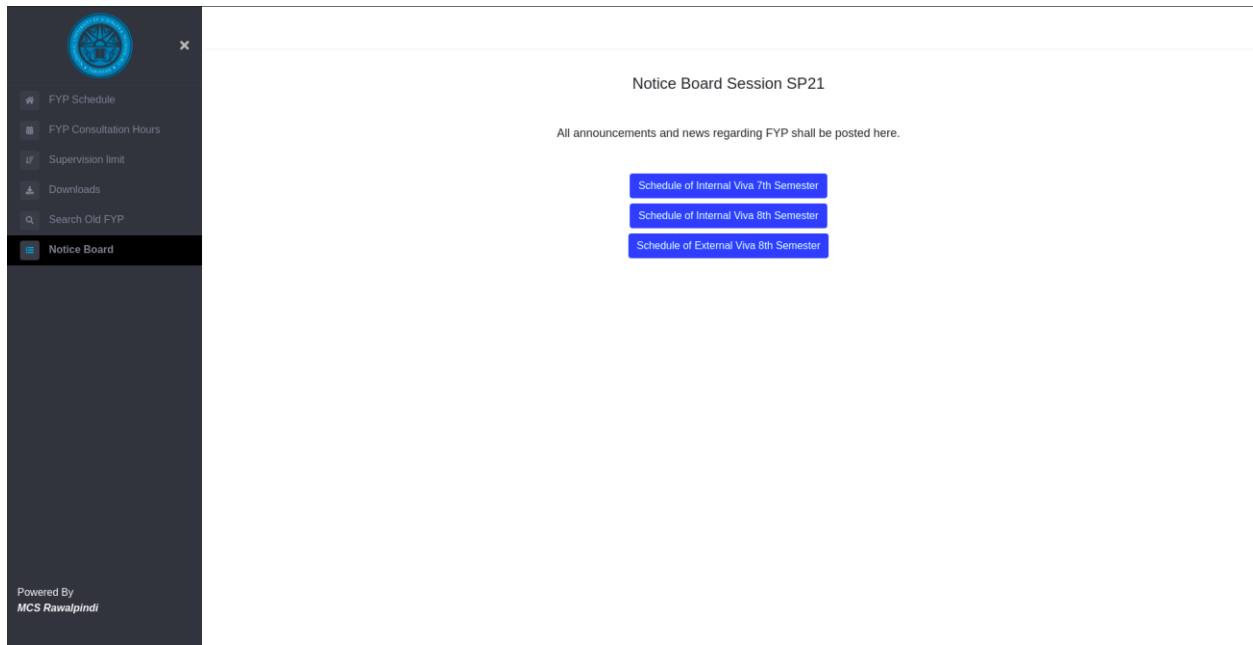


Fig: Notice Board

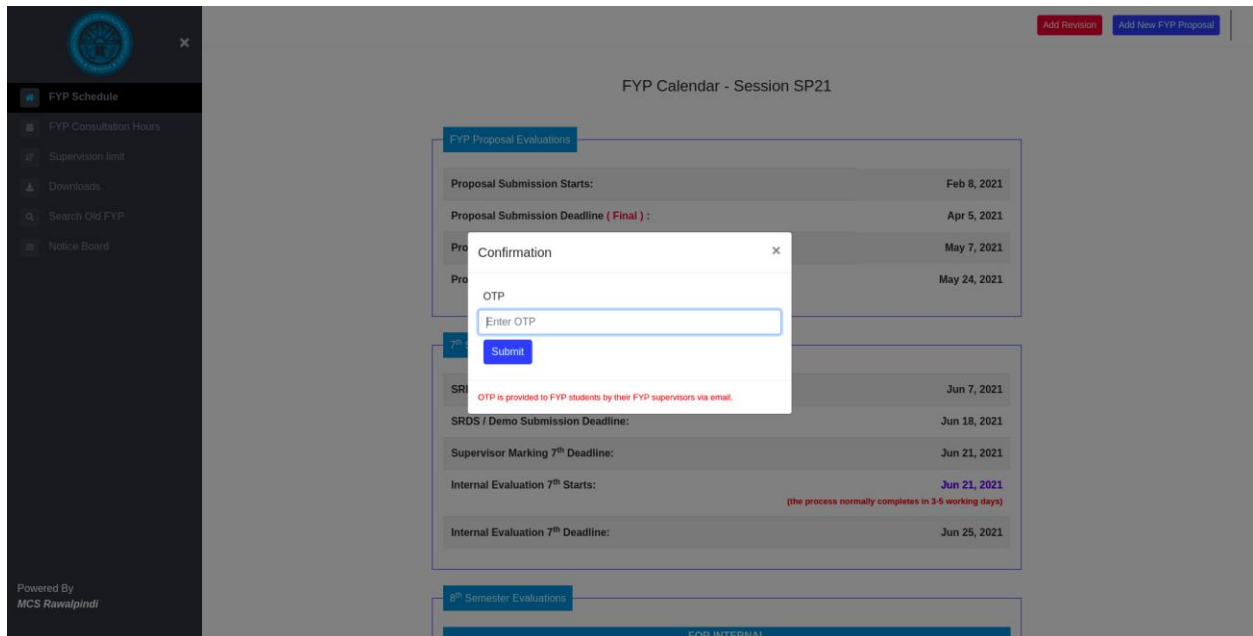


Fig: Student OTP submission for proposal submission.

## 6.2.2 Faculty Login Module

As students have login-less console, faculty has its own console following is the login screen for Faculty.

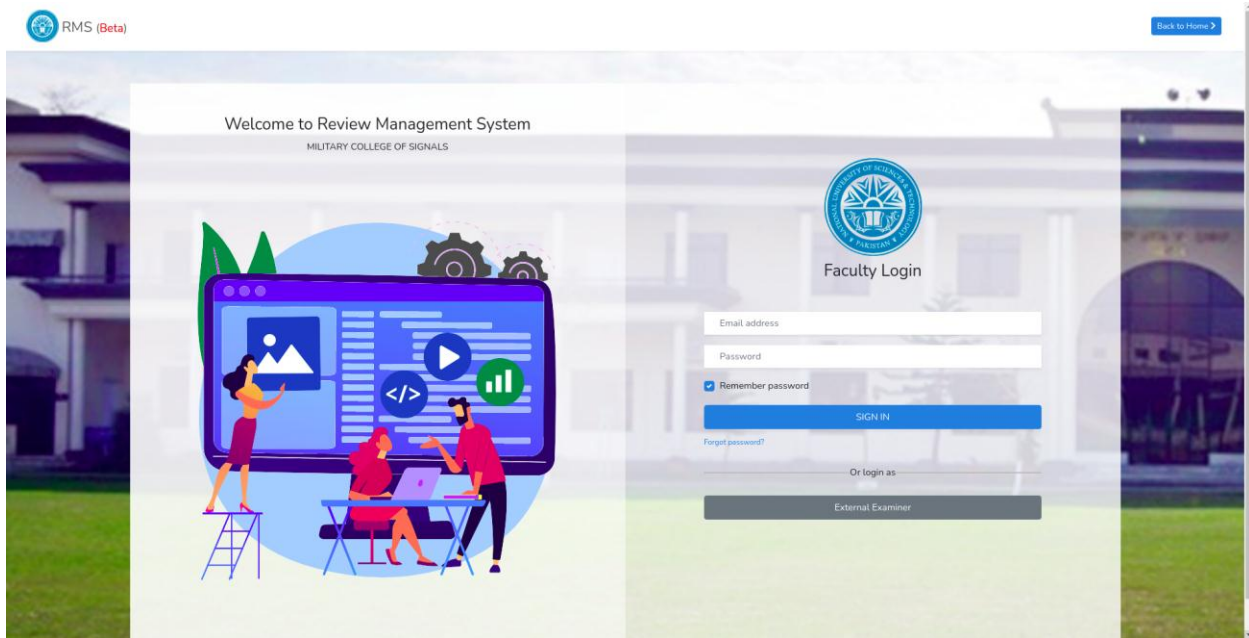


Figure 20 Login Module

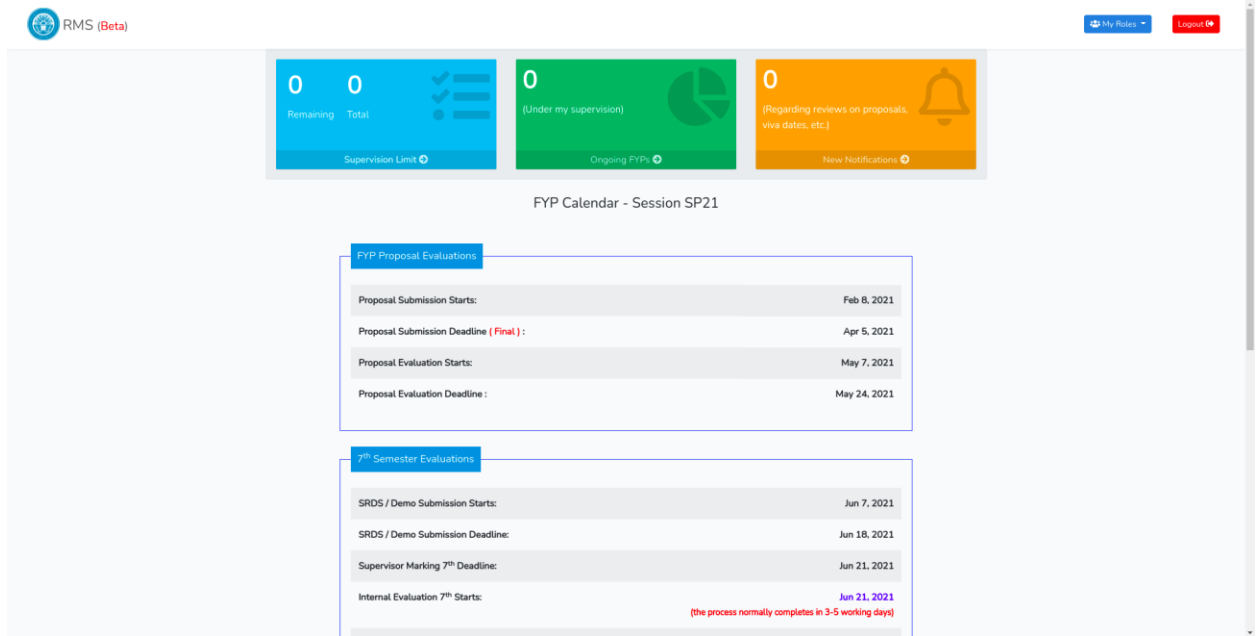


Fig: Faculty Dashboard

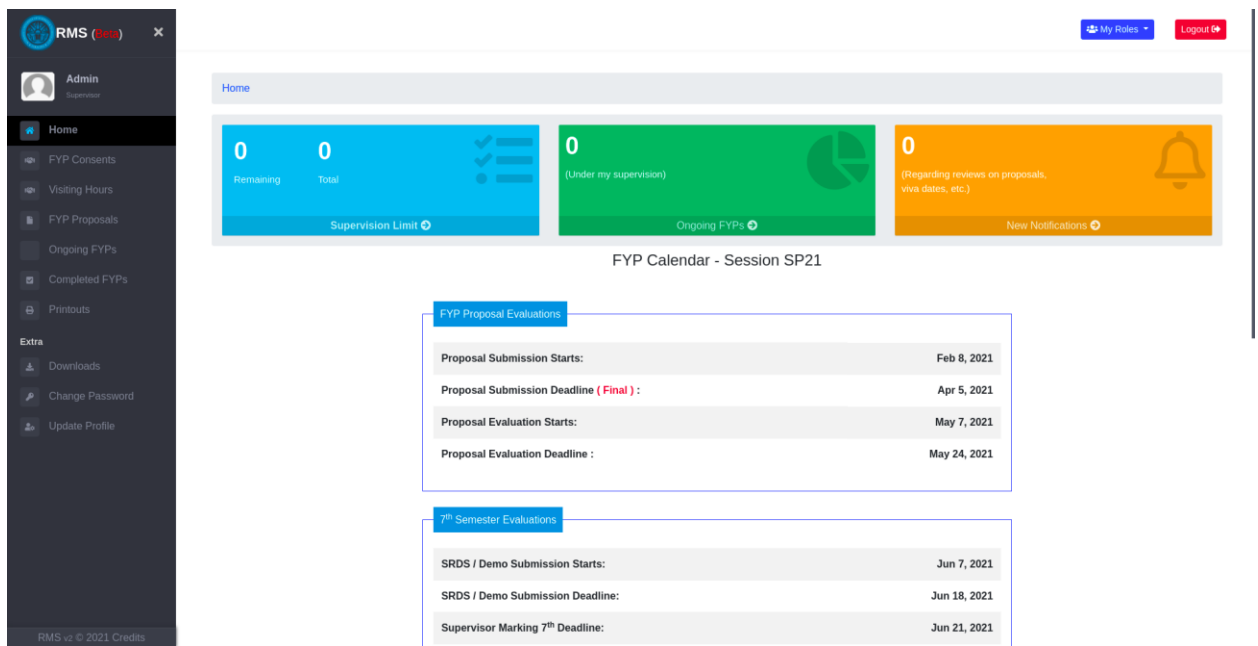


Fig: Supervisor Dashboard

RMS (beta) x

Admin  
Supervisor

- Home
- FYP Consents
- Visiting Hours**
- FYP Proposals
- Ongoing FYPs
- Completed FYPs
- Pinouts

Extra

- Downloads
- Change Password
- Update Profile

RMS v2 © 2021 Credits

My Roles Logout

Visiting Hours

DAY	TIME		VENUE
Monday	--:--	To --:--	
Tuesday	--:--	To --:--	
Wednesday	--:--	To --:--	
Thursday	--:--	To --:--	
Friday	--:--	To --:--	

Reset Update

Fig: Supervisor Visiting Hrs

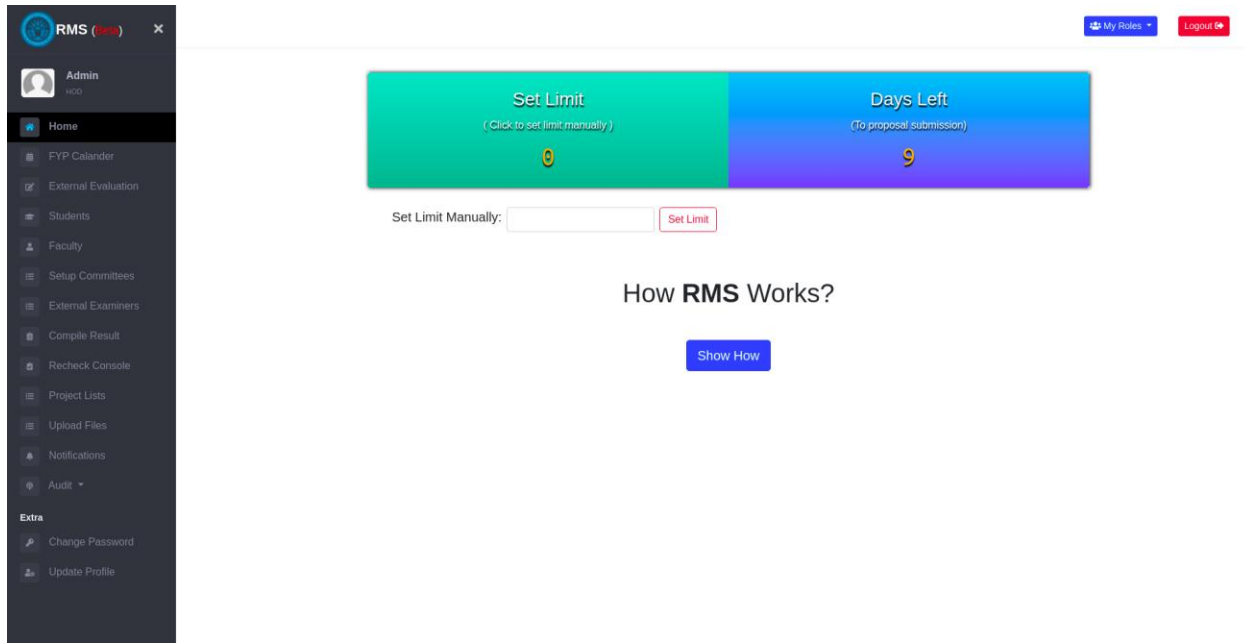


Fig: HOD/Coordinator Dashboard

### 6.2.3 Evaluator's Dashboard

As there are multiple Evaluation teams / committees each of the select member have their own dashboard where they can evaluate projects.

<< Back

### FYP Internal Evaluation Form- 8<sup>th</sup> semester

Project Title: 3D Driving School (Simulation Based)  
Project Stream: Mobile App

Individual Evaluation		Marks Ranking: 5 (Excellent)   4 (Good)   3 (Fair)   2 (Poor)   1 (Fail)		
Criteria	(Evaluate each student based on his/her work)			
Inamullah Shah CITIFA17-BSE-077/WAH	Question 1*	Question 2*	Question 3	
	Front end interfacing	Logical design	Use cases and analysis	
	3	2	2	
Project Evaluation		Marks Ranking: 5 (Excellent)   4 (Good)   3 (Fair)   2 (Poor)   1 (Fail)		
Criteria	(Evaluate this group based on team work)			
	1. FYP Report (e.g. Completeness of chapters, report formatting, Use case diagram, Sequence diagram, Class diagram, ERD/hardware configuration)			<a href="#">FYP Report</a> <a href="#">View Project Demo</a> 3
	2. Frontend / GUI Web/Android: input validation check, navigate through all screens and menus, user friendliness, responsiveness (works for Desktop/Mobile) Hardware: showcase hardware product with respect to various functionalities & commands, Product-like look. Game: game menus, characters, game levels and their difference from each other with respect to environment, difficulty, features.			3

Figure 21 Internal / External 7<sup>th</sup> /8<sup>th</sup> semester evaluation.

[My Roles](#) [Logout](#)

#### General Stats

[Print List](#)

Total Projects	Remaining Projects	Accepted Projects	Under Review
37	3	29	5

Pending Projects

Reviewed Projects

No.	Title	Student Name	Registration No.	Submitted On	Comments	Status	Review
1	Raneej Collection	SYEDA RANIA FATIMA AREEJ RAFIQUE	00000000000 00000000000	Jun 17, 2021		Not Reviewed	<a href="#">Review 1</a>
2	Online Journal Management system	MUHAMMAD ALI ZAIN ALI TAHIR	00000000000 00000000000	Jun 8, 2021		Not Reviewed	<a href="#">Review 1</a>
3	Preserving the Privacy of Banking Transactions using Differential Privacy	ASAD JAVED ABDUL SABOOR	00000000000 00000000000	Jun 13, 2021		Not Reviewed	<a href="#">Review 1</a>

Fig: Proposal Evaluation Dashboard



My Roles
Logout

### FYP Proposal Submission Form

Print Form

<b>Detail of Project Members:</b>	MUHAMMAD ALI , 00000000000 , 03138979059 ZAIN ALI TAHIR , 00000000000 , 03015693180
<b>Title:</b>	Online Journal Management system
<b>Which real world Problem shall be solved by this project?</b>	<p><b>Initial:</b> To overcome the physical problem related to Author, Editor and Publisher. We provide a web based platform which manage and publish the research paper. It'll reduce the time and energy related to clerical, administration and managerial tasks associated with journal editorial process</p> <p><b>Revision 1:</b> To overcome the physical problem related to Author, Editor and Publisher. We provide a web based platform which manage and publish the research paper. Managing the entire submission and flexible and configurable editorial workflow and publishing your articles. It'll reduce the time and energy related to clerical, administration, managerial and publication tasks associated with journal publication</p> <p>To overcome the physical problem related to Author, Editor and Publisher. We provide a web based platform which manage and publish the research paper: <a href="#">Managing the entire submission and flexible and configurable editorial workflow and publishing your articles</a>, It'll reduce the time and energy related to clerical, <a href="#">administration</a> <a href="#">administration managerial</a> and <a href="#">managerial publication</a> tasks associated with journal <a href="#">editorial process publication</a></p>
<b>Project Description (What, How and Objectives):</b>	<p><b>Initial:</b> Journal management system covers submission, review, editing, publication and archive processes deal with the management of research papers. Give access to common reader to facilitate from published paper, Author upload the file, editor review and manage the document and send to the publisher. This system provide an effective, and un-centralized supervision by different levels of the journal staff to allow them to control the submission, reviewing, tracking, evaluating and publishing an article via the web. Use HTML, CSS as a front end to manage the outlook of system interface and PHP as a backend to manage the database related to the file revision, submission, administration, edition and publication process.</p> <p><small>Design system for efficient, flexible and configurable editorial workflow. Online submission and management of all content</small></p>

**Fig: Proposal Evaluation Screen**

# Chapter 7

## **7 System Implementation**

### **7.1 Overview**

Testing of software projects include different levels of testing to ensure that the software which is being developed is error and fault free. The different levels at which testing was performed is argued here.

### **7.2 Unit Testing**

It includes the testing of each module at completion.

### **7.3 Login Feature Testing**

<b>Test Name</b>	<b>Validate login Credentials.</b>
------------------	------------------------------------

<b>Screen to be tested</b>	Login Screen.
<b>Input</b>	Enter email & password.
<b>Output</b>	Wrong credentials.
<b>Expected output</b>	Login to the system.
<b>Actual Result</b>	In this test the user was tested if he/she provides wrong email or wrong password the system didn't allow him/her logged into the profile and to enter in the particular area. But when valid credentials were entered then the system will provide its functionality to be performed.
<b>Test Verified by</b>	

#### 7.4 System Reliability testing

<b>Test Name:</b>	<b>System Reliability.</b>
<b>Screen to be tested</b>	Forget Password on login page.
<b>Input</b>	Enter your email.
<b>Output</b>	Valid email pop up update password section.
<b>Expected output</b>	Some action against pressing forget password.
<b>Actual Result</b>	In this test system will ask user to enter its email, then on pressing forget password button it will redirect user to our forget password web page link on their user will be able to update it password to continue use our system.
<b>Test Verified by</b>	

#### 7.5 Notify Committee Members

<b>Test Name</b>	<b>Notify Committee Members.</b>
<b>Screen to be tested</b>	Committee Setup Screen.
<b>Input</b>	Valid user email address to send email.
<b>Output</b>	Didn't send email to user with entered email.
<b>Expected output</b>	Email will be sent.
<b>Actual Result</b>	In this test system will ask user(coordinator) to press notify button to send email notifications to, but when he/she pressed button email will not send to some user.
<b>Test Verified by</b>	

## 7.6 Supervisor Consent

<b>Test Name</b>	<b>Supervisor Consent.</b>
<b>Screen to be tested</b>	Supervisor Dashboard.
<b>Input</b>	Valid user student registration no. to send email.
<b>Output</b>	Didn't send email to student with entered email and registration no.
<b>Expected output</b>	Email will be sent.
<b>Actual Result</b>	In this test system will ask user(supervisor) to sent consent to send email to students, but when he/she pressed button email will not send to some students.
<b>Test Verified by</b>	

# Chapter 8

## **8 Conclusion and Future Work**

### **8.1 Conclusion**

This project is quite tough as we have been given the opportunity to manage a project from start to finish. It was due to this project we came to know how professional software's are designed.

The objective of this project was mainly resolving all these problems and providing the department the facility to run FYP cycle in a timely manner. Everyone will be able to use it.

So we have worked on most of the things we have proposed in the proposal to make an efficient and better system which will help the CS department to get evaluation of FYP done.

Due to constraints of time and team size, the scope of the project was kept small. Initially we will deploy the system in one department, but its scope can be enhanced with the passage of time.

We firmly believe that our project can genuinely bring about a significant change in the performance.

### **8.2 Future work**

In Future the RMS will be able to integrate MS and PHD students also.

## Glossary

API	Application Programming Interface
App	Application
AS	Assumption
Black box Testing	Testing emphasizes on the external behaviour of the software entity
CO	Constraints
App	Application
OTP	One Time Password
DBMS	Database Management System
FYP	Final Year Project
FRs	Functional Requirements
GUI	Graphical User Interface
IDE	Integrated Development Environment
iOS	Mobile Operating System created and developed by Apple
MCS	Military College of Signals



NFRs	Non Functional Requirements
NUST	National University of Science and Technology
OE	Operating Environment
OS	Operating System
MySQL	Database
REQ	Requirement
SQL	Structured Query Language
SR	Safety Requirements
SRS	Software Requirements Specification
UD	User Documentation
UML	Unified Modelling Language
White Box Testing	Testing emphasizes on the internal behaviour of the software entity

**Table 21 Glossary**

## **Bibliography**

<https://www.coursera.org/>

<https://www.quora.org/>

[MySQL :: MySQL Documentation](#)

<https://www.w3schools.com>

<https://laravel.com/>

**Turnitin Plagiarism Report:**

## FYP THESIS

### ORIGINALITY REPORT

<b>18%</b>	<b>4%</b>	<b>0%</b>	<b>17%</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

### PRIMARY SOURCES

<b>1</b>	Submitted to De Montfort University Student Paper	<b>9%</b>
<b>2</b>	Submitted to Higher Education Commission Pakistan Student Paper	<b>4%</b>
<b>3</b>	Submitted to Indian Institute of Technology Guwahati Student Paper	<b>1%</b>
<b>4</b>	Submitted to Colorado Technical University Online Student Paper	<b>1%</b>
<b>5</b>	Submitted to HELP UNIVERSITY Student Paper	<b>1%</b>
<b>6</b>	Submitted to University of Greenwich Student Paper	<b>&lt;1%</b>
<b>7</b>	www.inettutor.com Internet Source	<b>&lt;1%</b>
<b>8</b>	Submitted to 於2012-06-13提交至Higher Education Commission Pakistan Student Paper	<b>&lt;1%</b>

9	Submitted to Arab Open University Student Paper	<1 %
10	pt.scribd.com Internet Source	<1 %
11	Submitted to CITY College, Affiliated Institute of the University of Sheffield Student Paper	<1 %
12	www.bestpractices.cahwnet.gov Internet Source	<1 %
13	www.slideshare.net Internet Source	<1 %
14	Submitted to Universiti Teknikal Malaysia Melaka Student Paper	<1 %
15	www.tvassignmenthelp.com Internet Source	<1 %
16	Submitted to Mississippi State University Student Paper	<1 %
17	Submitted to University of Technology, Sydney Student Paper	<1 %
18	dl.lib.mrt.ac.lk Internet Source	<1 %
19	spectrum.library.concordia.ca Internet Source	<1 %

“