

ACCADEMIA

Learning Management System

Functional Requirements Document

v.0.3

Prepared by:
Satvik Mishra
(Business Analyst)



1. Introduction

This document is created to define the functional requirements of the Accademia Learning Management System (LMS). It shall cover detailed information about the complete system architecture built, along with detailed specifications. This document shall be strictly followed throughout the development cycle.

1.1. Purpose

The purpose of this FRD is to describe all the components, features, and functionalities that shall be provided on the Accademia LMS system.

1.2. Product Scope

For the purpose of making the requirements easier to understand, the platform shall be scalable, flexible, and diverse in architecture for delivering E-Learning services as well as associated microservices, readily throughout the flow of the system.

The Accademia LMS aims to provide E-Learning and Management solutions to Schools and other Educational Institutions. The system shall be built over a web application that shall operate at different layers, which shall be described further.

2. Overall Description

Accademia shall be an interactive E-Learning and Management solution platform where the following services shall be provided:

Micro-service 1:

- a. Super Admin
- b. School Admin
- c. Teacher Admin
- d. Student Admin
- e. Parent Admin
- f. Class Admin

Micro-service 2:

- a. Student Assessment
- b. Student Assignment

- c. Study Material
- d. Attendance System
- e. Notification & Broadcasting
- f. Online Classroom Integration

Micro-service 3:

- a. Student Reporting
- b. Student Analytics

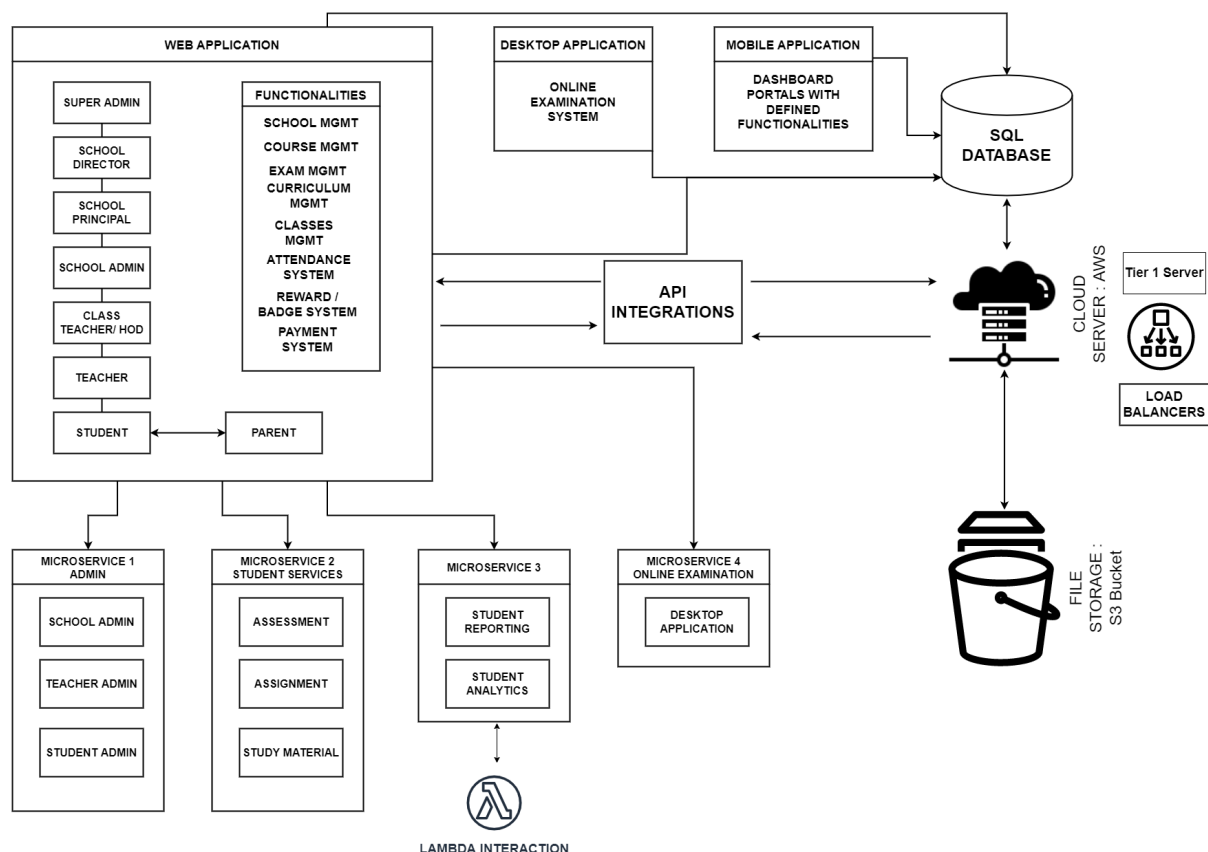
Micro-service 4:

- a. Online Examination - Desktop-based (To be developed later)

The system shall contain a Web-based application, with different login types for the respective layers which shall be defined further.

2.1. System Architecture

The **Accademia LMS** Architecture shall contain the following components:



Broad System Architecture

2.2. Product Components

The **Accademia LMS** system shall contain the following components:

Web Application

The LMS shall have an integrated Web Application with multiple login types at different levels

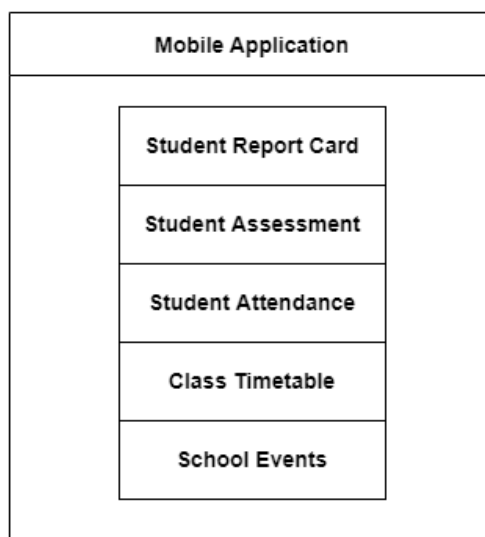
Login Types
<input type="radio"/> Super Admin
<input type="radio"/> School Director
<input type="radio"/> School Principal
<input type="radio"/> School Director
<input type="radio"/> Class Teacher / HOD
<input type="radio"/> Teacher
<input type="radio"/> Student
<input type="radio"/> Parent

Desktop Application

The LMS desktop application shall be built for Online Examination System. It shall be a webcam-based live examination portal.

Mobile Application

The LMS mobile application shall be developed to provide dashboard portal access and certain other functions.



Third-Party Integrations

The LMS shall use API Integrations to enhance user experience at such places wherever required.

2.3. Product Technical Specifications

Provided below are the basic summary specifications that the application systems shall perform. The functions are listed as follows:

- a. Web applications shall be built over the Django Framework powered by Python.
- b. Shall be integrated with a secure database for seamless querying and syncing to ensure information reliability.
- c. A Primary and Secondary Database shall be maintained. The Secondary DB shall be used for modules like reporting, the DB shall be updated every night.
- d. AWS S3 Service shall be used for file storage management.
- e. Cloud Server: AWS shall be used where Tier 1 Service shall be used at the initial stage. Load Balancers shall be used to manage the incoming load.
- f. AWS Lambda shall be used for serverless computing.
- g. Lambda shall be used to 'Dump' Study Material.
- h. MongoDB shall be used to make skeleton structures for the microservices.

- i. Alerts shall be triggered on the set events:
 - i. Email Alerts: Sandgrid shall be used.
 - ii. SMS Alerts: Twilio shall be used
- j. Encryption and data packet security shall be in force.
 - i. AES - 256 Encryption shall be used.
- k. Logging and Exception handling shall be properly managed.
 - i. For Logging Python In-built logging shall be used, also at certain modules AWS Lambda shall be used.
- l. Platform compatibility among modern browsers for the web application.
- m. SSL encryptions shall be used for securing the web application.
- n. Mobile Applications shall be built for Android OS.

3. System Modules, Functionalities, and Features

All the features and functionalities that shall be developed and implemented are described here in detail, and divided into major modules.

3.1. General System Features

The System shall contain the following features:

- a. The application shall be developed on Django Python Framework.
- b. The application shall be built on an integrated model where different functions shall be done at one application itself.
- c. Some functionalities shall use Javascript to optimize web page responses.
- d. UI/UX shall be developed of the highest standard for the E-Learning Industry.
- e. API Integrations shall be used for functionalities like Live Classes, SMS Integrations, Fee Payments (If any), etc.

3.2. Micro-Service 1

3.2.1 Description

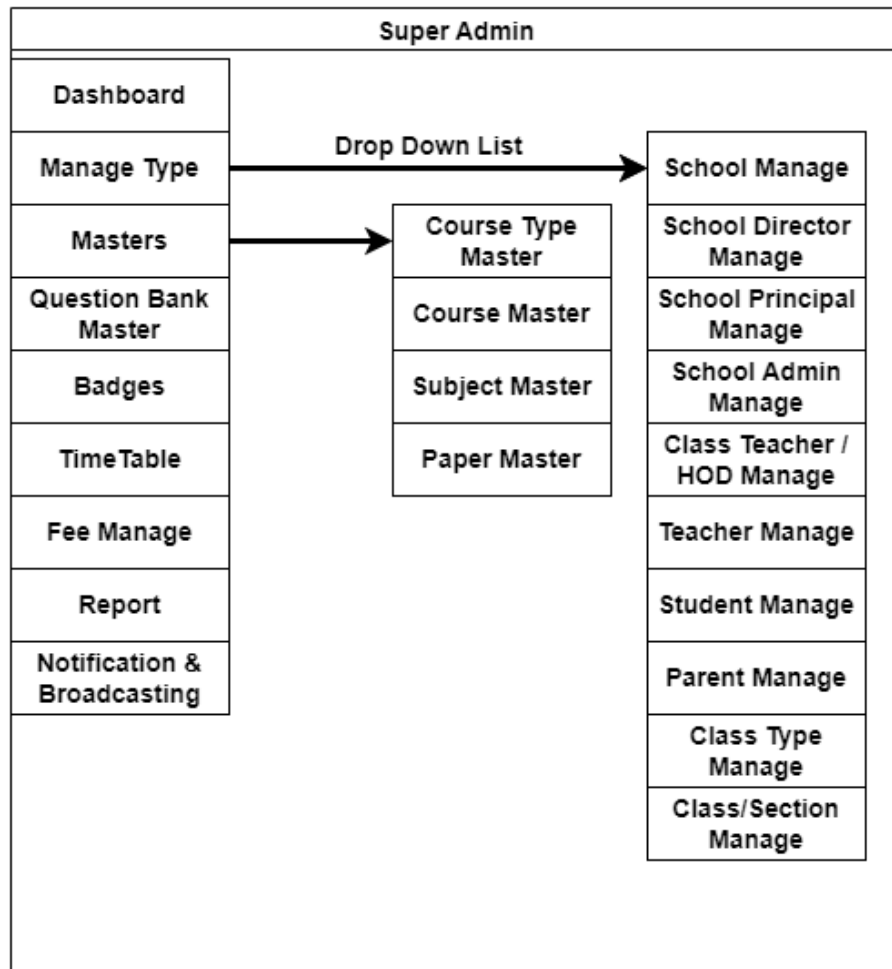
This MS shall deal with the admin part of the system. It is here that the master registries shall be maintained. The MS shall contain the following components:

- a. Super Admin
- b. School Admin
- c. Teacher Admin

- d. Student Admin
- e. Parent Admin
- f. Class Admin

3.2.2 Super Admin Module Functional Structure

The Super Admin Module shall be formed using the following components:



3.2.3 Super Admin Module Functional Requirements

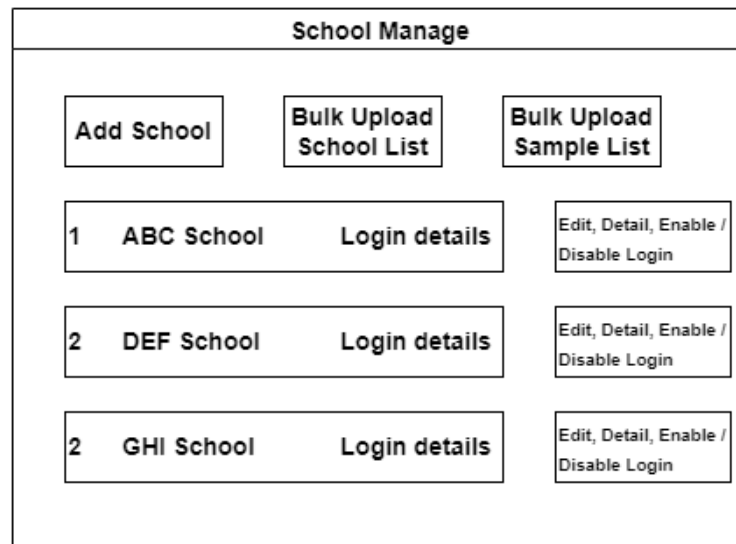
The Super-Admin Login shall be the master admin for the entire system. The FR for the components is described below:

Dashboard: The LMS dashboard shall display the real-time information about the Schools, Directors, Principals, School Admins, Class Teachers / Teachers, Students, Parents, Classes, etc.

Manage Type: This section shall enable the admin to manage the following

School Manage: This section shall enable the admin to manage the associated schools on the LMS system. It shall contain the following functionalities:

- a. Add School
- b. Bulk Upload
- c. Login Details: for every school which shall be generated after registration.
- d. Action Buttons: Edit, Details, Enable / Disable Login.



School Director Manage: This section shall enable the admin to perform the following functions:

- a. Add Director
- b. Action buttons: Edit, Details, Enable / Disable Login.

School Principal Manage: This section shall enable the admin to perform the following functions:

- a. Add Principal
- b. Action buttons: Edit, Details, Enable / Disable Login.

School Admin Manage: This section shall enable the super-admin to perform the following functions:

- a. Add School Admin
- b. Action buttons: Edit, Details, Enable / Disable Login.
- c. Bulk Upload list.

Class Teacher / HOD Manage: This section shall enable the super-admin to perform the following functions:

- a. Add Class Teacher / HOD
- b. Action buttons: Edit, Details, Enable / Disable Login.
- c. Bulk Upload list.

Teacher Manage: This section shall enable the super-admin to perform the following functions:

- a. Add Teacher
- b. Action buttons: Edit, Details, Enable / Disable Login.
- c. Bulk Upload list.

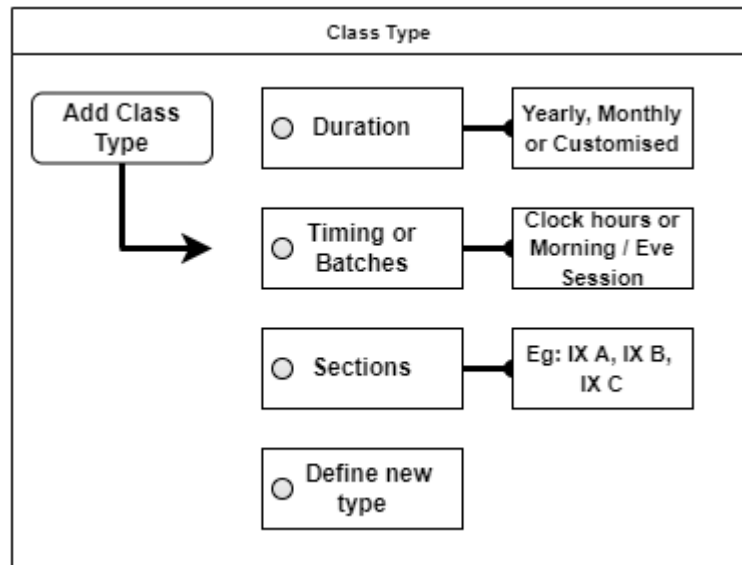
Student Manage: This section shall enable the super-admin to perform the following functions:

- a. Add Student
- b. Action buttons: Edit, Details, Enable / Disable Login.
- c. Bulk Upload list.

Parent Manage: This section shall enable the super-admin to perform the following functions:

- a. Add Student
- b. Action buttons: Edit, Details, Enable / Disable Login.
- c. Bulk Upload list.

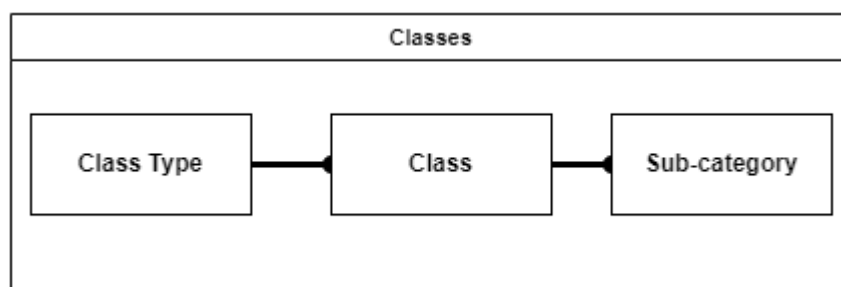
Class Type Manage: This section shall define the class types on the following basis.



It shall also allow the following functionalities:

- It shall allow to add and define a new class type.
- Admin shall customize the contents that shall be kept under each class type list.

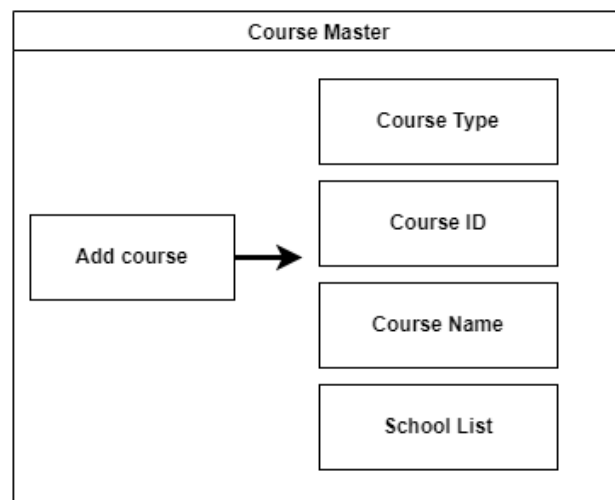
Class / Section Manage: This section shall allow managing the classes and the contents kept under the earlier defined class type for every school. The following action buttons shall also be provided: Edit, Delete.



Masters: This section shall manage the course structure for the schools. It shall contain the following components:

- a. Course Type Master: This section shall define the Macro category under which the courses shall be formed. Admin can add as many types. *“Admin shall also link schools to the course type in this section”*
- b. Course Master: This section enables the admin to form courses under different types. A unique course ID shall be required which will keep the course distinct. A list of schools shall appear, but only those which are linked with the particular course type.

Error/Exceptions: If a course is not linked to the institute an error shall be thrown.



- c. Subject Master: This section enables the admin to create subjects under the pre-defined courses. A unique subject ID is required which will keep the subject distinct. A list of schools shall appear, but only those which are linked with the particular course.

(Topic Management - A particular section for topics covered under the subjects shall also be developed).


Error/Exceptions: If a course is not linked to the institute an error shall be thrown.

- d. Paper Master: This section enables the admin to create Papers under the pre-defined subjects. A unique paper ID is required which will keep the paper distinct. A list of schools shall appear, but only those which are linked with the particular subject.

Error/Exceptions: If a subject is not linked to the institute an error shall be thrown.

- e. Topic Master: It shall contain functionalities like add topic, view topic, delete topic and activate / deactivate topic.

Question Bank Master: This section shall manage a repository of the questions to be created by the schools. It shall be structured in the following manner:

Question Bank					
Year	School	Class	Course	Subject	Topic
<div>Filter</div> <div>By Difficulty</div> <div><input type="checkbox"/> Easy</div> <div><input type="checkbox"/> Moderate</div> <div><input type="checkbox"/> Tough</div> <div>By Question Type</div> <div><input type="checkbox"/> Objective</div> <div><input type="checkbox"/> Subjective</div> <div>By Marks</div>		Question List	<div> Search Box</div>		

Add Question		
Select Course	Select Subject	Year
Question Type	Sub Question Type	Topic
Question		
Select No of Options	Difficulty Type	
Correct Marks	Negative Marks	

The Questions shall be retrievable at all ends that are mapped in accordance. The QB management shall be provided at school levels drilling down to teacher level.

Badges: This section shall enable the admin to create new badges and manage the list of badges. These shall be available to use in different schools to reward the students.

Add Badge
Title
Select badge trigger
Subject Completed
Topic Completed
Image
Description

A list of badges shall be formed which shall be triggered on the completion of a specific event. The following are some examples:

Verified Account	
Unstoppable	Subject Completed
Unbeatable	Topic Completed
Achiever	Subject Completed

Time-Table: This section shall enable the admin to create different timetables according to different parameters. It enables to do real-time Timetable planning, Subject allocation, Academic Calendars & Attendance Management. The timetable shall be created according to the following parameters:

Time - Table Management			
Timetable (Description)	Academic Details	Start Date	End Date
Week Date	DD/MM/YY	Search	Reset
Class Timing	Monday DD/MM/YY	Tuesday DD/MM/YY	Wednesday DD/MM/YY

Attendance Management						
Take Attendance						
Time Table Name	Academic Details	Class				
Subject Name	Class Timing	Class Type: Theory / Practical				
#	Attendance ○ P ○ L ○ A	Student ID	Roll No	EnRoll No	Name	Remark

Fee Management: This section shall deal with the fee management and dues of the students. The School Admin shall manage the required functions.

Super Admin Functionalities:

- a. Add Parameters for fee mgmt
- b. Define Due Criteria
- c. View Dues
- d. Deactivate/Activate Student

School Admin Functionalities:

- a. View Student
- b. Edit Student
- c. Search Student
- d. View Dues
- e. Deactivate / Activate Student

Class Teacher & Teacher shall have the following functionalities:

- a. View Student
- b. Search Student
- c. View Dues
- d. Request for Deactivation / Activation.

Students and Parents shall have the following functionalities:

- a. View Student
- b. View Dues

Fee Management							
Student ID	Roll no	Name	Course	Year	Fee	Paid	Due

Report Card Management: This section shall enable the admin to manage the reporting-related functionalities of the students.

Super - Admin functionalities:

- a. Add Parameters for Report Card Generation
- b. Add Passing Criteria
- c. View Student Report
- d. De-activate/Activate student

School Admin functionalities:

- a. View Student Details
- b. Update Details
- c. View Student Report
- d. De-activate/Activate student

Class Teacher/Teacher functionalities:

- a. View Student Details
- b. Update Details
- c. Publish Student Report
- d. View Student Report
- e. View Report Issues

Student functionalities:

- a. View Report
- b. Report Issue

Parent functionalities:

- a. View Report
- b. Report Issue

Notification & Broadcasting: This section shall enable the admin to push notifications to all the users throughout the system and also broadcast circulars.

Super-Admin Functionalities:

- a. Create Parameters for Notification & Broadcasting
- b. Create Notification / Circular
- c. Add Recipients
- d. Schedule Notification / Circular
- e. Delete Notification / Circular

School-Admin Functionalities

- a. Create Notification / Circular
- b. Add Recipients
- c. Schedule Notification / Circular
- d. Delete Notification / Circular

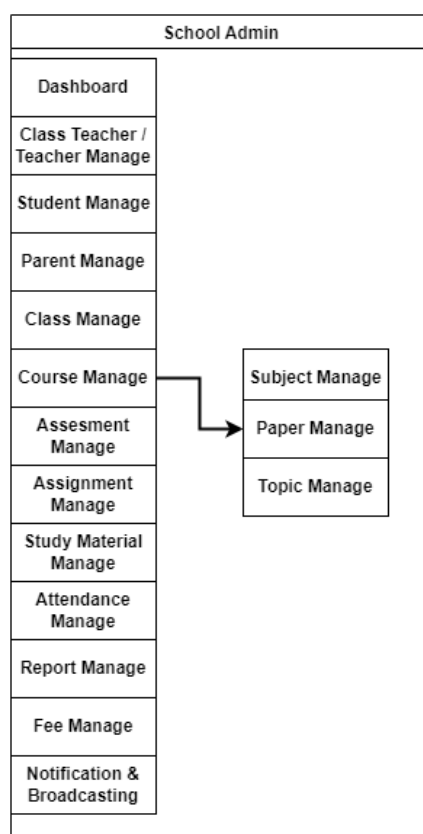
Class Teacher / Teacher Functionalities:

- a. Create Notification

- b. Add Recipients
- c. Schedule Notification
- d. Delete Notification

“Circular shall not be created at this level”

3.2.3 School Admin Functional Structure



3.2.4 School Admin Functional Requirements

The school admin shall be provided with extensive management authorities for the particular school. The Functional Requirements for the components are the following:

Class Teacher / Teacher Manage: School Admin shall have the following functionalities regarding this section.

- a. Add Class Teacher / Teacher

- b. View Class Teacher / Teacher
- c. Edit Class Teacher / Teacher
- d. Delete Class Teacher / Teacher
- e. De-activate / Activate Teacher.

Student Manage: School Admin shall have the following functionalities regarding this section.

- a. Add Student
- b. View Student
- c. Edit Student
- d. Delete Student
- e. View Student Attendance
- f. View Student Report
- g. View Student Dues
- h. Activate / Deactivate Student.

Parent Manage: School Admin shall have the following functionalities regarding this section.

- a. Add Parent
- b. View Parent
- c. Edit Parent
- d. Delete Parent
- e. Activate / Deactivate Parent.

Class Manage: School Admin shall have the following functionalities regarding this section.

- a. Add Class
- b. Edit Class
- c. Delete Class
- d. View Class Performance

Course Manage: School Admin shall have the following functionalities regarding this section.

- a. Add Course
- b. Edit Course
- c. Delete Course
- d. Activate / Deactivate Course.

It shall also contain the following sub-components:

Subject Manage: School Admin shall have the following functionalities regarding this sub-section.

- a. Add Subject
- b. Edit Subject
- c. Delete Subject
- d. Activate / Deactivate Subject.

Paper Manage: School Admin shall have the following functionalities regarding this sub-section.

- a. Add Paper
- b. Edit Paper
- c. Delete Paper
- d. Activate / Deactivate Paper

Topic Manage: School Admin shall have the following functionalities regarding this sub-section.

- a. Add Topic
- b. Edit Topic
- c. Delete Topic
- d. Activate / Deactivate Topic

Assessment Manage: School Admin shall have the following functionalities regarding this section.

- a. View Assessment
- b. Approve Assessment
- c. Reject Assessment
- d. Reasons for Rejection

Assignment Manage: School Admin shall have the following functionalities regarding this section.

- a. View Assignment
- b. Approve Assignment
- c. Reject Assignment
- d. Reasons for Rejection

Study Material Manage: School Admin shall have the following functionalities regarding this section.

- a. View Study Material
- b. Approve Study Material
- c. Reject Study Material

- d. Reasons for Rejection

Attendance Manage: School Admin shall have the following functionalities regarding this section.

- a. View Attendance (Viewable according to different filters eg, Class / Section / Student, etc).
- b. Search Student
- c. Activate / Deactivate Student.

Report Manage: School Admin shall have the following functionalities regarding this section.

- a. View Student Report
- b. Activate / Deactivate Student

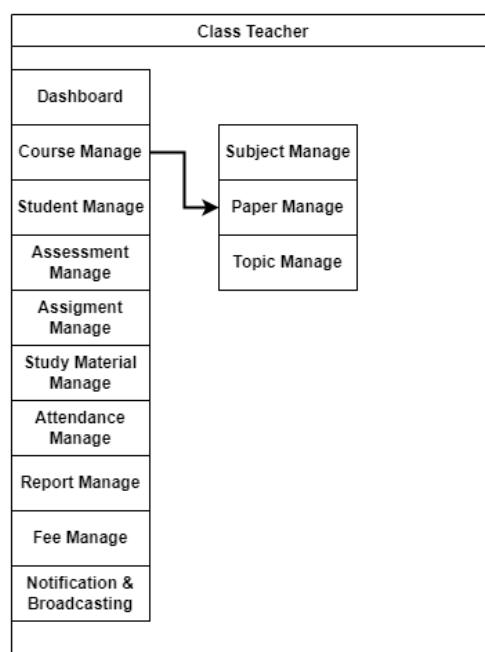
Fee Manage: School Admin shall have the following functionalities regarding this section.

- a. View Student Fees (Viewable according to different filters)
- b. Search Student
- c. View Dues
- d. Activate / Deactivate Student.

Notification & Broadcasting: School Admin shall have the following functions regarding this section.

- a. Push Notification / Circular.
- b. Schedule Notification / Circular.
- c. Add Recipients
- d. Delete Notification / Circular.

3.2.5 Class Teacher / HOD Functional Structure



3.2.6 Class Teacher / HOD Functional Requirements

The Class teacher shall have the following functions in the respective components.

Course Manage: The class teacher shall have the following functionalities regarding this section.

- a. View Course
- b. Edit Course

It shall also contain the following sub-components:

Subject Manage: The class teacher shall have the following functionalities regarding this sub-section.

- a. View Subject
- b. Edit Subject

Paper Manage: The class teacher shall have the following functionalities regarding this sub-section.

- a. Add Paper
- b. Edit Paper

- c. View Paper
- d. Delete Paper

Topic Manage: The class teacher shall have the following functionalities regarding this sub-section.

- a. Add Topic
- b. Edit Topic
- c. Delete Topic

Student Manage: The class teacher shall have the following functionalities regarding this section.

- a. View Student
- b. View Student Attendance
- c. View Student Report
- d. View Student Dues
- e. Request Student Activation / Deactivation

Assessment Manage: The class teacher shall have the following functionalities regarding this section.

- a. Create Assessment
- b. Publish Assessment
- c. View Assessment
- d. Edit Assessment
- e. Delete Assessment

Assignment Manage: The class teacher shall have the following functionalities regarding this section.

- a. Create Assignment
- b. Publish Assignment
- c. View Assignment
- d. Edit Assignment
- e. Delete Assignment

Study Material Manage: The class teacher shall have the following functionalities regarding this section.

- a. Create Study Material
- b. Publish Study Material
- c. View Study Material
- d. Edit Study Material
- e. Delete Study Material

Attendance Manage: The class teacher shall have the following functionalities regarding this section.

- a. View Attendance
- b. Search Student
- c. Verify Attendance
- d. Edit Attendance
- e. Request Student Activation / Deactivation

Report Manage: The class teacher shall have the following functionalities regarding this section.

- a. View Student Report
- b. Edit Student Report
- c. Request Student Activation / Deactivation

Fee Manage: The class teacher shall have the following functionalities regarding this section.

- a. View Student Fees
- b. Search Student
- c. View Dues
- d. Request Student Activation / Deactivation.

Notification: The class teacher shall have the following functionalities regarding this section.

- a. Push Notification
- b. Schedule Notification
- c. Add Recipients
- d. Delete Notification

3.2.7 Teacher Functional Requirements

The Teacher shall have mostly overlapping functions with the Class Teacher. Only the differing functionalities are mentioned below.

Course Manage: The Teacher shall only be permitted to add & edit topics

“All other functionalities shall be available to the Teacher”.

3.2.8 Student Functional Structure

Student	
<div>Dashboard</div> <div>Assessments</div> <div>Assignments</div> <div>Study Material</div> <div>Attendance</div> <div>Report Card</div> <div>Time Table</div> <div>Profile</div>	

3.2.9 Student Functional Requirements

The Student shall require the following functionalities in the respective components.

Dashboard: The Student Dashboard shall be very interactive with Real-time information regarding the Student Academics. The dashboard shall contain the following details:

- a. Student Ongoing Performance Stats.
- b. Student Ongoing Courses
- c. Progress Bar in each course with the time spent and remaining duration
- d. Analytics Regarding Student Performance in all fields.
- e. Fee Status.
- f. Badges & Rewards earned.
- g. Events

Assessments: The Student shall have the following functionalities in this section.

- a. View Assessment
- b. Attempt Assessment

- c. Upload Assessment
- d. View Assessment Evaluation.
- e. Request Re-evaluation.

Assignments: The Student shall have the following functionalities in this section.

- a. View Assignment
- b. Attempt Assignment
- c. Upload Assignment
- d. View Assignment Evaluation
- e. Request Re-evaluation.

Study Material: The Student shall have the following functionalities in this section.

- a. View Study Material
- b. Download Study Material

Attendance: The Student shall have the following functionalities in this section.

- a. Verify Attendance (Form shall be used)
- b. For Online Classes Time-in / Time- Out shall be recorded and verified by the above form.
- c. Attendance Support (In case of discrepancy).

Report Card: The Student shall have the following functionalities in this section.

- a. View Report Card
- b. List of Report Records
- c. Download Report Card

Time Table: The Student shall have the following functionalities in this section.

- a. View Time Table for a particular date / subject etc
- b. Create Reminders

Profile: The Student shall have the following functionalities in this section.

- a. Update Profile
- b. Request Change

3.3 Micro-Service 2

3.3.1 Description

This MS shall dedicatedly deal with the Services to be provided at the student end. It shall be a crucial component of the system. The MS shall contain the following components.

- a. Student Assessment
- b. Student Assignment
- c. Study Material
- d. Attendance System
- e. Notification & Broadcasting
- f. Online Classroom Integration

3.3.2 Student Assessment Functional Requirements

This section shall have the following functionalities:

- a. Add Assessment
- b. Choose Course
- c. Choose Subject
- d. Choose no of sections
- e. Choose assessment type in each section (subjective / objective)
- f. Choose question type (Single choice / Multi Choice)
- g. Choose assessment duration
- h. Choose start and end time
- i. Input Guidelines.
- j. Add Questions
- k. Assessment Status shall be displayed.
- l. Validations shall be placed to fulfill requirements.
- m. Publish / Unpublish / Send for Approval.

3.3.3 Student Assignment Functional Requirements

This section shall have the following functionalities:

- a. Add Assignment
- b. Choose Course
- c. Choose Subject
- d. Choose no of sections
- e. Choose Assignment duration
- f. Input Guidelines

- g. Add Questions
- h. Student can upload assignments
- i. Publish / Unpublish / Send for Approval.

3.3.4 Study Material Functional Requirements

This section shall have the following functionalities:

- a. Add Study Material
- b. Choose Study Material File Format Type
- c. Upload Material
- d. Student shall read the material in the Application
- e. Tracking to be applied on the material read to monitor progress.

3.3.5 Attendance System

This section shall have the following functionalities:

- a. Fetch online attendance
- b. Verify attendance via form
- c. Time In / Time out to be recorded.
- d. In case of offline classes the same form shall record the attendance.
- e. Attendance shall be mapped with each unique Student ID.

3.3.6 Notification & Broadcasting

This section shall have the following functionalities:

- a. Broadcast to whom
- b. Upload file (If any)
- c. Fill in the message.

3.3.7 Online Classroom Intergration

This section shall have the following functionalities:

- a. Integration shall be done through API of the online classes service provider.
- b. Sessions, Server and Client Model shall be used for the integration with the API. (Subject to change)

3.4 Micro-Service 3

3.4.1 Description

This MS shall deal with the following two components:

- a. Student Reporting
- b. Student Analytics

3.4.2 Student Reporting Functional Requirements

This section shall have the following functionalities:

- a. Data shall be extracted, organised and prepared for report generation.
- b. Filtering, Sorting and grouping tools continually function throughout the lifecycle of reports.
- c. Report shall be delivered in the required format.

3.4.3 Student Analytics Functional Requirements

This section shall have the following functionalities:

- a. Data shall be extracted, organised and prepared for data visualisation.
- b. Analytics shall make the student dashboard very interactive and easy to understand.
- c. Static and dynamic images, tables, charts & graphs, Infographics shall be used for data visualization.

UI Screens

S.No	Module	Sub Module	No of Screens
1	Super Admin	Manage Type	29
2		Masters	18
3		Question Bank	3
4		Badges	3
5		Time Table	3
6		Fee Manage	3
7		Report Card	3
8		Notification & Broad	3
9	School Admin		39
10	School Principal		39
11	School Director		39
12	Class Teacher / HOD		41
13	Teacher		41
14	Student		29
15	Parent		15
	Total		308

S.No	Resource	Person	Days
1	Backend Developer	2	40
2	UI Designer	1	7
3	Frontend Developer	2	12
4	Testing Engineer	1	15
5	Deployment Engineer	1	5

*Project Testing shall run simultaneously throughout the Development Life-cycle.

*Development shall be flexible, allowing minor changes at the required levels.