
SOFTWARE REQUIREMENT SPECIFICATION FOR DIGITAL BOARD MARKER

VERSION 3.0

PREPARED BY

MUHAMMAD HARIS KHAN(2016-CS-105)
HAMZA FAROOQ(2016-CS-122)
AYESHA ATIF(2016-CS-152)
KOMAL SHEHZADI(2016-CS-178)

*University of Engineering and Technology
Lahore*

Copyright © 2019 all rights reserved

Contents

1	Introduction	9
1.1	Background	9
1.2	Motivation	9
1.3	Scope	10
1.4	Problem Statement	10
1.5	Application Areas	11
1.5.1	Educational Institutes:	11
1.5.2	Online Tutors	11
1.5.3	Sketch Artists	11
2	Existing Work	11
2.1	Details of Existing Work	11
2.2	Comparison of Existing Work	12
3	Software Development Life Cycle	12
3.1	SDLC Used	12
3.2	Justification for SDLC	13
4	Requirement Analysis	14
4.1	Functional Requirements	14
4.2	Non-functional Requirements	14
4.2.1	Performance Requirements	14
4.2.2	Security Requirements	15
4.2.3	Software Quality Attributes	15
4.3	User Interfaces (Web App)	16
4.3.1	Main page	16
4.3.2	Registration page	18

4.3.3	Login page	19
4.3.4	All Courses	20
4.3.5	Add Courses page	21
4.3.6	Courses Details	22
4.3.7	Teachers Registration Requests	24
4.3.8	Students Registration Requests	25
4.3.9	Enrollment Requests	26
4.3.10	Lecture Details	27
4.3.11	Notes	28
4.3.12	Upload Notes	29
4.3.13	Course Assignment	30
4.3.14	Submit Assignment	31
4.3.15	Submitted Assignment	32
4.3.16	Upload Assignment	33
4.3.17	Lecture Player	34
4.3.18	Student Details	35
4.3.19	Teacher Details	36
4.3.20	Permission Management	37
4.3.21	Add Groups for Permission Management	38
4.4	User Interfaces (Offline Player)	39
4.4.1	Lecture Player	39
4.4.2	Token based Authentication	41
4.4.3	Rules and Assumption	43
4.4.4	Module Workflow Description	43
4.5	Use Case Diagram	46
4.6	Use Cases(Web App)	50
4.6.1	Use Case UC-1: User Registration	50

4.6.2	Use Case UC-2: User Login	52
4.6.3	Use Case UC-3: Forget Password	54
4.6.4	Use Case UC-4: Teacher's Requests Approval	56
4.6.5	Use Case UC-5: Student's Requests Approval	58
4.6.6	Use Case UC-6: Add New Course	61
4.6.7	Use Case UC-7: Edit Course	63
4.6.8	Use Case UC-8: Delete Course	66
4.6.9	Use Case UC-9: View Courses	67
4.6.10	Use Case UC-10: Upload Course Assignment	69
4.6.11	Use Case UC-11: View Course Assignments	72
4.6.12	Use Case UC-12: Download Course Assignment	74
4.6.13	Use Case UC-13: Delete Course Assignment	77
4.6.14	Use Case UC-14: Add Course Announcement	79
4.6.15	Use Case UC-15 :Edit Course Announcement	82
4.6.16	Use Case UC-16: Delete Course Announcement	85
4.6.17	Use Case UC-17: Student Assignment Submission	88
4.6.18	Use Case UC-18: Upload Course Notes	91
4.6.19	Use Case UC-19: View Course Notes	95
4.6.20	Use Case UC-20: Download Course notes	97
4.6.21	Use Case UC-21: Delete Course Notes	100
4.6.22	Use Case UC-22: Play Course Lectures	102
4.6.23	Use Case UC-23: Delete Course Notes	105
4.6.24	Use Case UC-24: Student's Course Enrollment	108
4.6.25	Use Case UC-25: Course Enrollment Requests Handling	110
4.6.26	Use Case UC-26: Assign Role's Permissions	113
4.6.27	Use Case UC-27: Add New Role	115
4.6.28	Use Case UC-28: Delete Role	117

4.6.29	Use Case UC-29: Add New Class	119
4.6.30	Use Case UC-30: Edit Class	121
4.6.31	Use Case UC-31: Delete Class	123
4.6.32	Use Case UC-32: View Classes	125
4.6.33	Use Case UC-33: Start Lecture	126
4.6.34	Use Case UC-34: End Lecture	128
4.6.35	Use Case UC-35: Assign Courses	129
4.6.36	Use Case UC-36: View Students Assignments	132
4.6.37	Use Case UC-37: Download Student's Assignments . . .	133
4.7	Use Cases (Offline Player)	135
4.7.1	Use Case UC1: User Authentication	135
4.7.2	Use Case UC2: View Lecture Playlist(Online)	137
4.7.3	Use Case UC3: View Lecture Playlist(Offline)	138
4.7.4	Use Case UC4: Play Lecture(Online)	138
4.7.5	Use Case UC5: Play Lecture(Offline)	140
4.7.6	Use Case UC6: View About page	141
4.7.7	Use Case UC7: View Contact us page	141
4.8	Test Cases	142
4.8.1	Test Case TC-01: User/Student Registration-Web Application	142
4.8.2	Test Case TC-02: User Authentication-Web Application	143
4.8.3	Test Case TC-03: View Course dashboard-Web Application	144
4.8.4	Test Case TC-04: View Course Content-Web Application	144
4.8.5	Test Case TC-05: View Course Lectures-Web Application	145
4.8.6	Test Case TC-06: View Lecture Notes-Web Application	146
4.8.7	Test Case TC-07: View Lecture Video-Web Application	148

4.8.8	Test Case TC-08: View Assignments-Web Application	149
4.8.9	Test Case TC-09: View Announcements-Web Application	150
4.8.10	Test Case TC-10: Add Assignments(Teacher)-Web Application	151
4.8.11	Test Case TC-11: Add Announcement(Teacher)-Web Application	153
4.8.12	Test Case TC-12: Add Institute(Super Admin)-Web Application	155
4.8.13	Test Case TC-13: Add Admin(Super Admin)-Web Application	156
5	Planning	158
5.1	Milestones	158
5.2	Work Breakdown Structure (WBS)	158
5.3	Network Dependency Diagram	159
5.4	Gantt Chart	160
6	Design	162
6.1	Data Flow Diagrams (Web App)	162
6.1.1	Student	162
6.1.2	Teacher	163
6.2	Data Flow Diagrams (Offline Player)	163
6.2.1	User General Flow	163
6.2.2	General Flow	165
6.3	Activity Diagrams	166
6.4	Data Modeling	167
6.4.1	ER Diagram	167
6.4.2	Database Diagram	168
6.5	Class Diagram	169

List of Figures

1	Existing Systems	12
2	Main Page	16
3	Main Page	17
4	Registration	18
5	Login	19
6	Course page	20
7	Add Course Page	21
8	Course Details	22
9	Teachers Registration Requests Page	24
10	Students Registration Requests Page	25
11	Enrollment Requests	26
12	Lecture Details(Teacher)	27
13	Notes	28
14	Upload Notes	29
15	Course Assignments	30
16	Submit Assignments	31
17	Submitted Assignments	32
18	Upload Assignments	33
19	Play Lecture	34
20	Student Details	35
21	Teacher Details	36
22	Permission Management	37
23	Add Groups	38
24	lecture player details	39
25	token authentication details	42

26	General User Workflow	44
27	General Application Workflow	45
28	Use Case Diagram Web App(1)	46
29	Use Case Diagram Web App(2)	47
30	Player Application	48
31	Controller Application	49
32	WBS	158
33	Network Dependency Diagram	159
34	Gantt Chart(1)	160
35	Gantt Chart(2)	161
36	Gantt Chart(3)	161
37	Working flow for Student	162
38	Working flow for teacher	163
39	General User Workflow	164
40	General Application Workflow	165
41	General Application Workflow	166
42	Entity Diagram	167
43	Database Diagram	168
44	Class Diagram	169

List of Tables

1	Main Page	17
2	Registration	19
3	Login	20
4	All Courses	21
5	Add Courses	22

6	Courses Details	23
7	Teachers Registration Requests	25
8	Students Registration Requests	26
9	Enrollment Requests	27
10	Lecture Detials	28
11	Notes	29
12	Add Notes	30
13	Assignments	31
14	Submit Assignments	32
15	All Submitted Assignments	33
16	Add Assignments	34
17	Lecture Player	35
18	Student Details	35
19	Teacher details	36
20	Manage Permissions	37
21	Add Groups	38
37	Milestones	158

1 Introduction

1.1 Background

Digital Board Marker is a size efficient, bandwidth saving lecture recording system. It can record lecture, providing automated google search of handwritten words. It provides on the spot wiki. Lecture text notes can be generated automatically. Lecture can be named and divided into topics and subtopics automatically. According to a survey, 94% students go for online help of recently attended lectures because they can't fully grab the concepts. Recorded lectures as video format require so much internet bandwidth to play. In most cases, large sized videos are difficult to handle or download. Because students mostly don't have huge amount of extra space available especially for the CSE students, as they already use bulky software and also students don't have large amount of bandwidth of internet available. Digital Board Marker is an innovation and need of the hour because it is solving the basic problem of all students in society because it saves their time to note the lecture and they can concentrate on the topic completely. As most of the student lost their concentration while writing the detailed lecture. Moreover, sometimes we skip important part of our lecture in the effort of writing whole details on our notebook. So, Digitized Lecture System is leading-edge to solve all these Issues. It has some potential for having positive effect on student learning.

1.2 Motivation

The motivation and purpose to do this project is to minimize the use of resources that are used in lecture systems now a days working in all over the world i.e. video lecture recording and streaming through internet. The first motivation is to deal with the large amount of storage that normally video lectures take. This system is not based on video recording but on recording the writing on the board with marker. It will record the position of the marker as the coordinates of board where marker touches and store it in the text file (which will later be converted and played like a video). This will take minimum amount of database storage to store this kind of data on a website. The second motivation to do this project is to use less internet resources for accessing the lectures. Normally the video lectures

of different institutes worldwide are very large and to download those on the system through internet requires large amount of resources which are normally difficult for students to get and to download it in high quality even more resources are required. The lectures for recording are very low in memory as compared to normal video recording and will require very minimum resources to download on the system. The third motivation is for example a power failure occurred during the lecture and you cannot clearly see the board but teacher is still writing and erases the board after some time, this may result in not getting proper notes or missing the important point of lecture. Moreover students can get benefit by seeing the lecture again and again if they missed any concept or if they were absent minded or not attending lecture. These few are the reasons which motivated us to do this project.

1.3 Scope

Digital board marker mainly cover academic area the main purpose is to provide each and every student all the lectures with better quality and less bandwidth because in Pakistan we students face this issue the most, as we know it cannot be resolved in near future we have to work something out for this issue and that's where this system will work it will provide an interface to all the students which have all the lectures of their respective subjects from their respective teachers which can be streamed online and downloaded for offline to play later on at very low bandwidth. It will provide all the assignment related material and lectures at same platform to students. It is the new revolution in the academic field. Although it covers industry and researches as well.

1.4 Problem Statement

To make a storage and bandwidth efficient system with a lecture player and learning management system for the students and the educational institutes.

1.5 Application Areas

1.5.1 Educational Institutes:

Educational institutes can use digital board marker for online lectures. They can have different users:

- Admin
- Teacher
- Student

1.5.2 Online Tutors

Online tutors can use the system to deliver lectures online with high quality and low storage. Tutor can login and start recording lecture, upload and delete them. Students/Listeners can play and download the lectures.

1.5.3 Sketch Artists

Sketch artists can give online tutorials of sketching with high quality video and less storage use for users. User can play video again and again without using much data.

2 Existing Work

2.1 Details of Existing Work

Online video lecture systems that exist now a days are:

- Edx
- Coursera
- Udemy

- Udacity

These are the system which are very helpful to users and can be accessed by everyone. They provide video lectures of class recording the video and audio. Almost all the systems use this same procedure of recording with camera and microphone.

2.2 Comparison of Existing Work

Initiative	Storage Efficiency	Free to Access	Quality	Storage Space
Edx	×	✓	360p	1.9MB/min
Coursera	×	✓	360p	1.8MB/min
Udemy	×	×	360p	2.0MB/min
Udacity	×	✓	360p	1.8MB/min
DBM	✓	✓	Max	18KB/min

Figure 1: Existing Systems

3 Software Development Life Cycle

3.1 SDLC Used

The SDLC model used is Agile model. Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations.

Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like:

- Planning
- Requirements Analysis
- Design
- Coding
- Unit Testing and
- Acceptance Testing

At the end of the iteration, a working product is displayed to the customer and important stakeholders.

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In Agile, the tasks are divided to time boxes (small time frames) to deliver specific features for a release. Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

3.2 Justification for SDLC

In our project at first requirements were not so clear so we opted for agile model where while working on current requirements we plan next phase easily. At first we worked on waterfall model then analysing the system we have decided to work with agile model because it is most suitable. There are three separate modules working right now. Each module is developing in iteration which is considered to be one week. Each week one person is scrum master who arranges daily group meetings in which we discuss the problems in previous assigned tasks and how to solve them, approach to be taken in coming week tasks.

4 Requirement Analysis

4.1 Functional Requirements

- User should be able to register and login.
- Student should be able to view a course and enrol in the course.
- Admin should be able to add new course.
- Teacher should be able to see student's enrolment requests.
- Admin should be able to see teacher's and students' registration requests and approve them.
- All user should be able to view and download the lecture.
- Teacher should be able to delete a lecture.
- Users should be able to download course notes.
- Teacher should be able to add new course notes.
- Students should be able to download and submit assignments.
- Teacher should be able to add new assignment and download all students' assignments.
- User should be able to play lectures.
- Admin should be able to add new permissions, user groups and assign permissions to users.

4.2 Non-functional Requirements

4.2.1 Performance Requirements

Some performance requirements are listed below.

- The software should support the use of multiple users at a time.

- The database should be able to record thousands of records.
- Live streaming of lecture should also be possible within low bandwidth.
- Software should support the live streaming of multiple lectures at a time.
- No loss of data should occur while uploading the lecture.
- No loss of data should occur while live streaming of lecture.

4.2.2 Security Requirements

Some of the factors that are identified to protect the software from accidental and malicious attacks are discussed below.

- Assign certain function to different modules.
- Check data integrity for critical variables.

4.2.3 Software Quality Attributes

- **Availability:** The system should be available on any time for use of multiple users.
- **Correctness:** The system should provide correct content and lectures of a specific subject.
- **Usability:** The system should satisfy a maximum number of user needs.
- **Confidentiality:** Software should allow only authorized users to access the System.
- **Portability:** Software should be portable to avoid any problem that may occur from moving one OS to another OS.
- **Supportability:** Software should use proper naming conventions and coding standards.

4.3 User Interfaces (Web App)

4.3.1 Main page

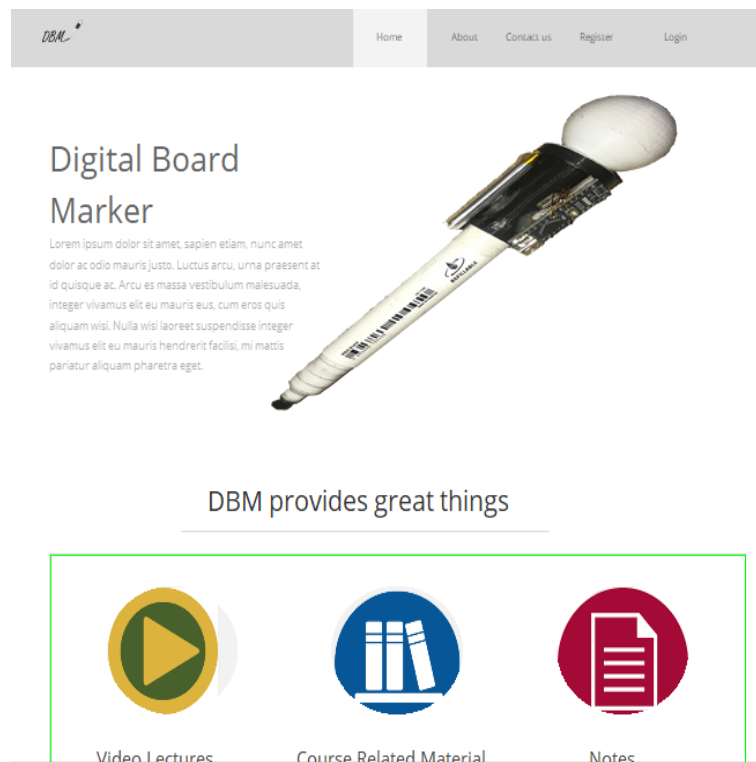


Figure 2: Main Page

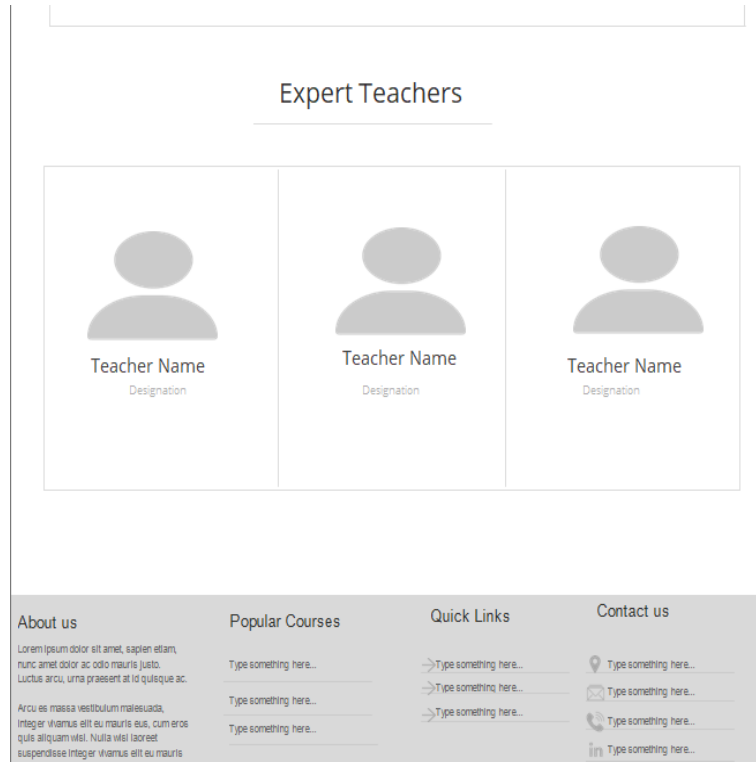


Figure 3: Main Page

Tasks	As an Admin	As a Teacher	As a Student
T1	I shall be able to see register page	I shall be able to see register page	I shall be able to see register page
T2	I shall be able to see login page	I shall be able to see login page	I shall be able to see login page
T3	I shall be able to manage permissions page		
T4	I shall be able to manage courses page		
T5	I shall be able to see all institutes page		
T6	I shall be able to manage all teachers page		

Table 1: Main Page

4.3.2 Registration page

The image shows a registration page for a system named 'DBM'. The page has a navigation bar with links: Home, About, Contact us, Register, and Login. The main heading is 'Registration'. Below the heading, there are several input fields for user registration, each with a red circle and a number indicating its position:

- 1: FIRST NAME (text input)
- 2: LAST NAME (text input)
- 3: CNIC (text input)
- 4: Degree (text input)
- 5: Email (text input)
- 6: COUNTRY (dropdown menu)
- 7: Password (text input)
- 8: University (dropdown menu)
- 9: Confirm Password (text input)

At the bottom right of the form is a 'Register' button. Below the form, there is a footer section with four columns: 'About us', 'Popular Courses', 'Quick Links', and 'Contact us'. Each column contains placeholder text and icons.

Figure 4: Registration

Component	Description	Mandatory	Validation	Type
1	First name of user.	Yes	Only letters	string
2	Last name of user.	Yes	Only letters	string
3	CNIC number of user.	Yes	16 digit number	string
4	Designation of user.	Yes	Only letters	string
5	Email of user.	Yes	Email format	string

6	Country of user.	Yes	Only letters	string
7	Password of user.	Yes	Only letters	string
8	University name of user.	Yes	Only letters	string
9	Re-Enter password of user.	Yes	Only letters	string

Table 2: Registration

4.3.3 Login page

The image shows a web application's login page. At the top, there is a header with a logo and navigation links: Home, About, Contact us, Register, and Login. The main content area is titled "Sign In" and contains two input fields: "Email" and "Password". Below these fields is a "SIGN IN" button. Red circles with numbers 1, 2, and 3 are placed next to the Email field, Password field, and SIGN IN button respectively, with red lines pointing to them. Below the button are two links: "Forgot password?" and "Not a member yet?". The footer contains four columns: "About us", "Popular Courses", "Quick Links", and "Contact us", each with placeholder text and icons.

Figure 5: Login

Component	Description	Mandatory	Validation	Type
1	User registered email.	Yes	Email format.	string
2	User password corresponding to email.	Yes	None	string
3	Sign in button to login to account and perform actions.	Yes	None	submit

Table 3: Login

4.3.4 All Courses



Figure 6: Course page

Component	Description	Mandatory	Validation	Type
1	To view details of courses.	None	None	None
2	To delete a course from list.	None	None	None
3	Search box to search course from the list.	None	None	None
4	Button to add new courses.	None	None	None

Table 4: All Courses

4.3.5 Add Courses page

The screenshot shows the 'Add Course' page of a web application. At the top is a navigation bar with the DBM logo, a user profile icon, and links for Home, About, Contact us, and Logout. Below the navigation bar is a header section titled 'Add Course'. The main content area contains a form with two input fields: 'Course name' (annotated with a red circle 1) and 'Details' (annotated with a red circle 2). Below the 'Details' field is a button labeled 'select title picture from stream' (annotated with a red circle 3). At the bottom of the form is a black 'Add Course' button (annotated with a red circle 4). The footer section is divided into four columns: 'About us' with placeholder text, 'Popular Courses' with three input fields, 'Quick Links' with three arrows and input fields, and 'Contact us' with three icons and input fields. The footer also includes a copyright notice 'Copyrights © | DBM UET Lahore' and social media icons for Twitter, Facebook, and Instagram.

Figure 7: Add Course Page

Component	Description	Mandatory	Validation	Type
1	Name of the course.	Yes	None	string
2	Description related to course.	Yes	None	string
3	Button to add title image for course.	No	None	Submit
4	Save button to save course in database.	None	None	None

Table 5: Add Courses

4.3.6 Courses Details

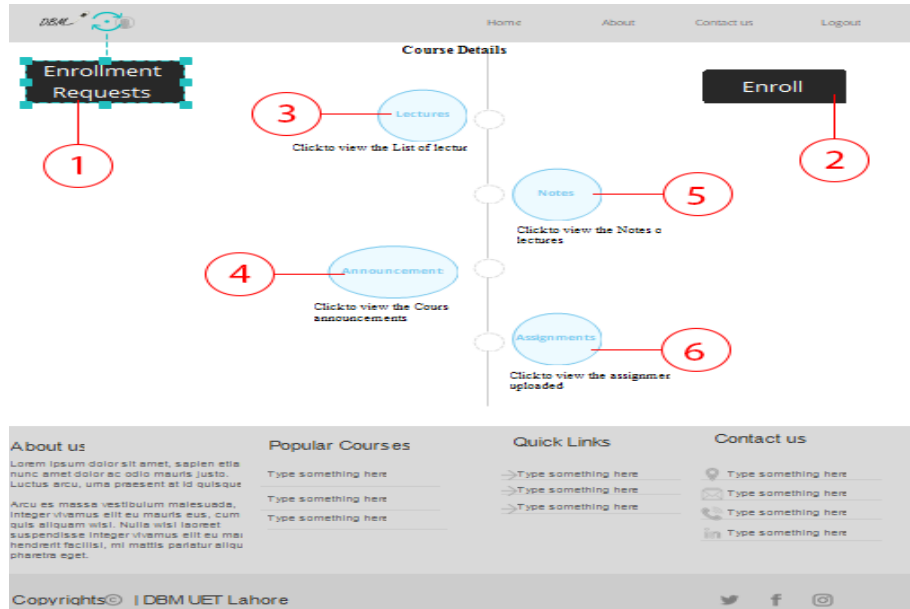


Figure 8: Course Details

Component	Description	Mandatory	Validation	Type
1	To see enrolment request of students in course. It will be only visible to teacher.	None	None	None
2	To enrol in course. It will be only visible to students.	None	None	None
3	To see list of all video lectures.	None	None	None
4	To see list of announcements.	None	None	None
5	To see list of notes.	None	None	None
6	To see list of assignments.	None	None	None

Table 6: Courses Details

4.3.7 Teachers Registration Requests

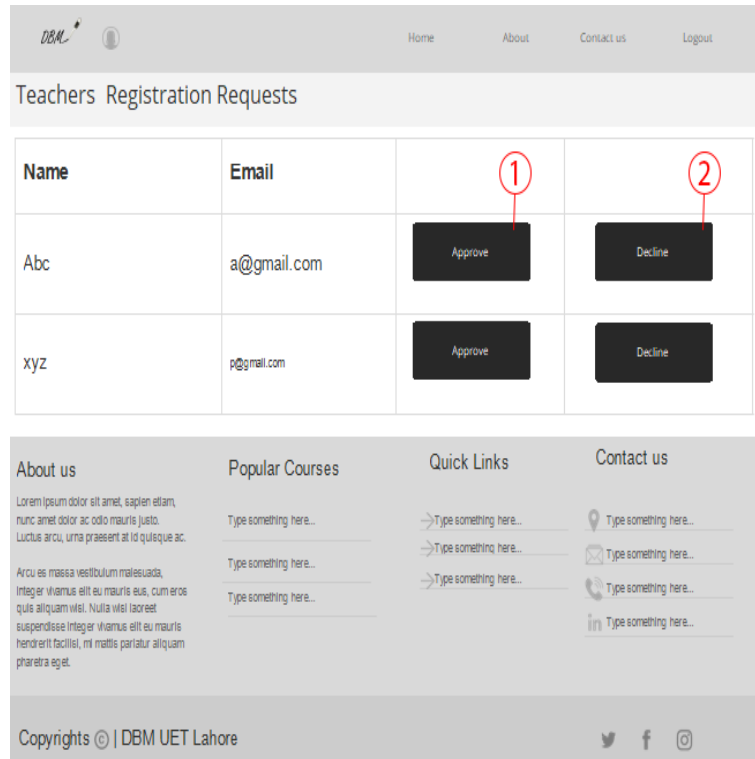


Figure 9: Teachers Registration Requests Page

Component	Description	Mandatory	Validation	Type
1	To approve teachers registration requests. This will only be visible to admin.	None	None	None

2	To delete teachers registration requests. This will only be visible to admin.	None	None	None
---	---	------	------	------

Table 7: Teachers Registration Requests

4.3.8 Students Registration Requests

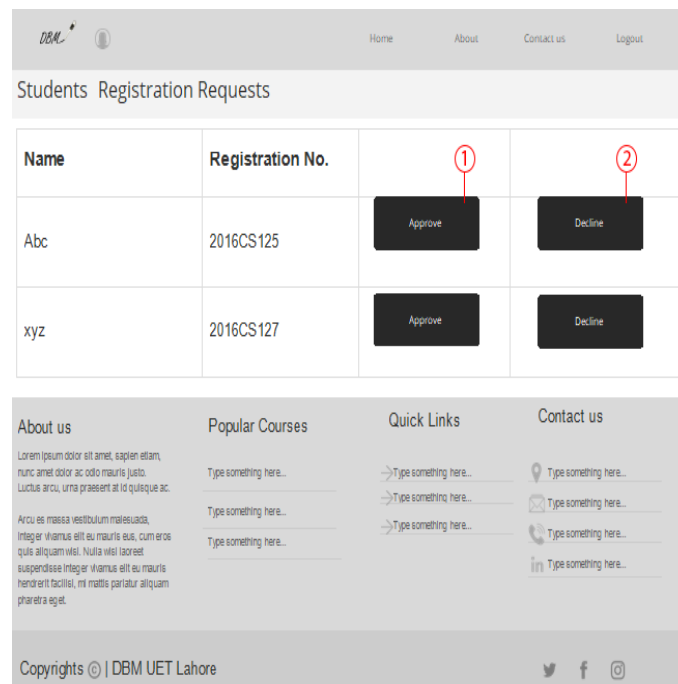


Figure 10: Students Registration Requests Page

Component	Description	Mandatory	Validation	Type
1	To approve students registration requests. This will only be visible to admin.	None	None	None
2	To delete students registration requests. This will only be visible to admin.	None	None	None

Table 8: Students Registration Requests

4.3.9 Enrollment Requests

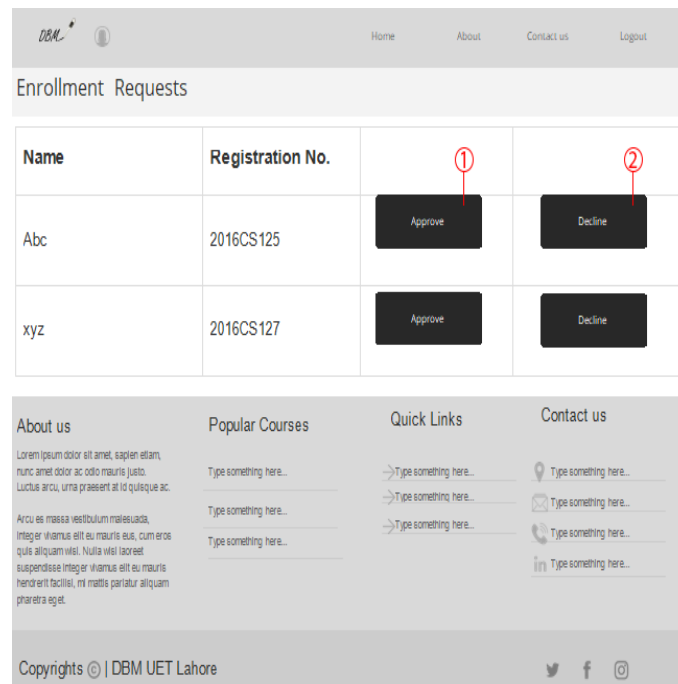


Figure 11: Enrollment Requests

Component	Description	Mandatory	Validation	Type
1	To approve students enrolment requests. This will only be visible to admin.	None	None	None
2	To delete students enrolment requests. This will only be visible to admin.	None	None	None

Table 9: Enrollment Requests

4.3.10 Lecture Details

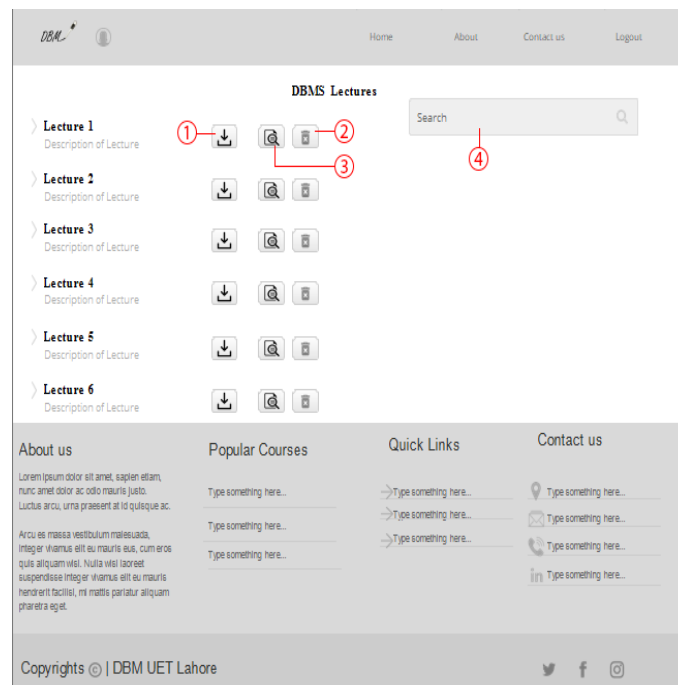


Figure 12: Lecture Details(Teacher)

Component	Description	Mandatory	Validation	Type
1	To download the lecture file.	None	None	None
2	To delete the lecture. This will be only visible to teacher.	None	None	None
3	To play lecture online.	None	None	None
4	A search box to search for lectures from list.	None	None	None

Table 10: Lecture Detials

4.3.11 Notes

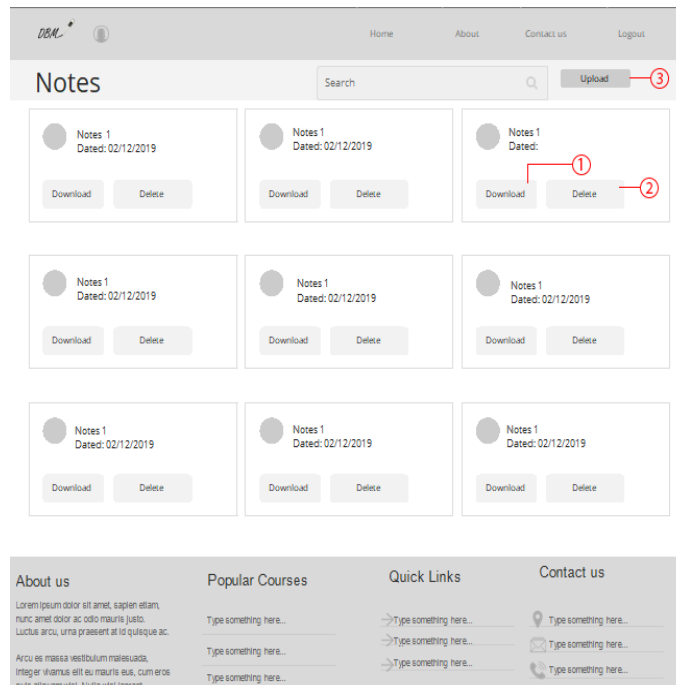


Figure 13: Notes

Component	Description	Mandatory	Validation	Type
1	To download the notes file.	None	None	None
2	To delete the lecture. This will be only visible to teacher.	None	None	None
3	To upload new notes file. This will be only visible to teacher.	None	None	None

Table 11: Notes

4.3.12 Upload Notes

The screenshot displays the 'Upload Notes' interface. At the top, there is a navigation bar with the DBM logo and links for Home, About, Contact us, and Logout. Below this, the page title 'Upload Notes' is centered. The main content area features a 'Lecture Name' input field, a 'Select Notes from stream' button, and an 'Upload' button. Red annotations (circles with numbers 1, 2, and 3) highlight these elements. The footer is divided into four sections: 'About us' with placeholder text, 'Popular Courses' with three input fields, 'Quick Links' with three links, and 'Contact us' with three contact options. The bottom of the page includes a copyright notice for DBM UET Lahore and social media icons.

Figure 14: Upload Notes

Component	Description	Mandatory	Validation	Type
1	Title for notes file name.	Yes	None	string
2	To upload notes file.	Yes	None	None
3	Upload button to save the data.	None	None	None

Table 12: Add Notes

4.3.13 Course Assignment

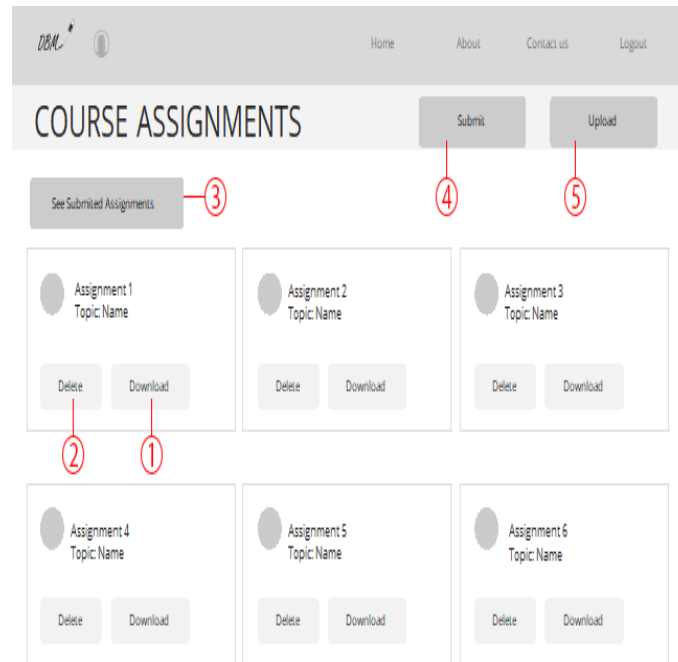


Figure 15: Course Assignments

Component	Description	Mandatory	Validation	Type
1	To download assignment file.	None	None	None

2	To delete assignment file. This will be only visible to teacher.	None	None	None
3	To see all the submitted assignments.	None	None	None
4	To submit any assignment. This will be only visible to students.	None	None	None
5	To upload new assignment. This will be only visible to teacher.	None	None	None

Table 13: Assignments

4.3.14 Submit Assignment

The screenshot shows a web application interface for submitting an assignment. At the top, there is a header with a logo and navigation links: Home, About, Contact us, and Logout. Below the header, the main heading is "Submit Assignment". The form consists of three main parts: a dropdown menu for "Assignment name" (labeled 1), a file upload section with a "Choose" button (labeled 2), and a "Submit" button (labeled 3). The footer contains sections for "About us", "Popular Courses", "Quick Links", and "Contact us", each with placeholder text. At the very bottom, there is a copyright notice "Copyrights © | DBM UET Lahore" and social media icons for Twitter, Facebook, and Instagram.

Figure 16: Submit Assignments

Component	Description	Mandatory	Validation	Type
1	Select assignment name student want to submit.	Yes	None	string
2	Choose assignment file to be submitted.	Yes	None	None
3	To submit assignment and save in database.	None	None	None

Table 14: Submit Assignments

4.3.15 Submitted Assignment

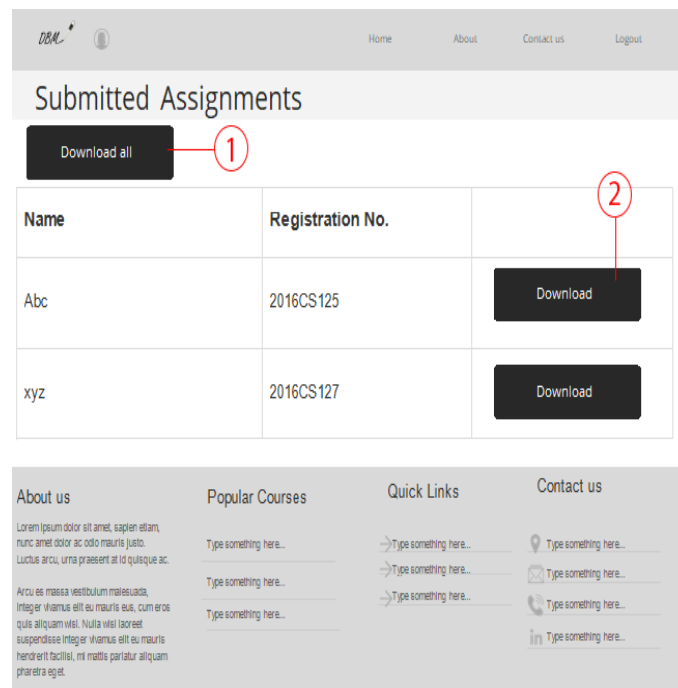


Figure 17: Submitted Assignments

Component	Description	Mandatory	Validation	Type
1	To download all students assignments.	None	None	None
2	To download individual assignments.	None	None	None

Table 15: All Submitted Assignments

4.3.16 Upload Assignment

DBM

Home About Contact us Logout

Upload Assignment

Assignment name

Due Date

Select Notes from steam

Upload

About us

Popular Courses

Quick Links

Contact us

Copyrights © | DBM UET Lahore

Figure 18: Upload Assignments

Component	Description	Mandatory	Validation	Type
1	Title of assignment file.	Yes	None	None
2	Due date to submit assignment.	Yes	Date format	Date
3	To assignment file.	None	None	None
4	To save data in database.	None	None	None

Table 16: Add Assignments

4.3.17 Lecture Player

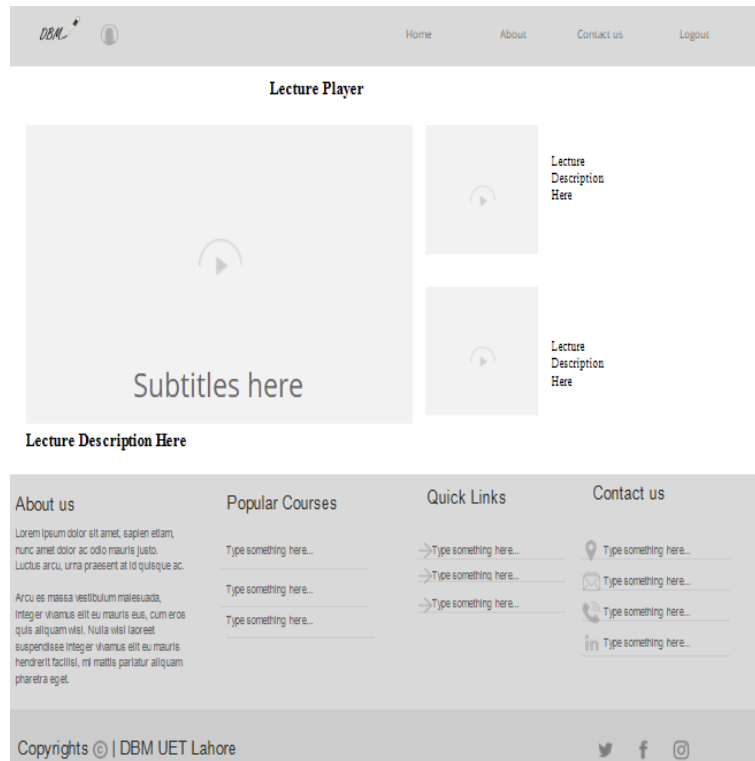


Figure 19: Play Lecture

Tasks	As an Admin	As a Teacher	As a Student
T1	I shall be able to play lecture on video player.	I shall be able to play lecture on video player.	I shall be able to play lecture on video player.

Table 17: Lecture Player

4.3.18 Student Details

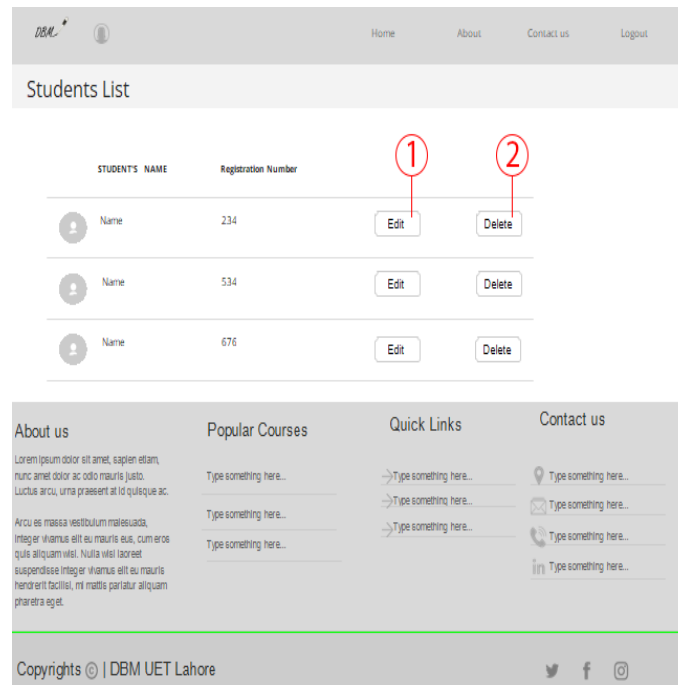


Figure 20: Student Details

Component	Description	Mandatory	Validation	Type
1	To edit the student details.	None	None	None
2	To remove the student from record.	None	None	None

Table 18: Student Details

4.3.19 Teacher Details

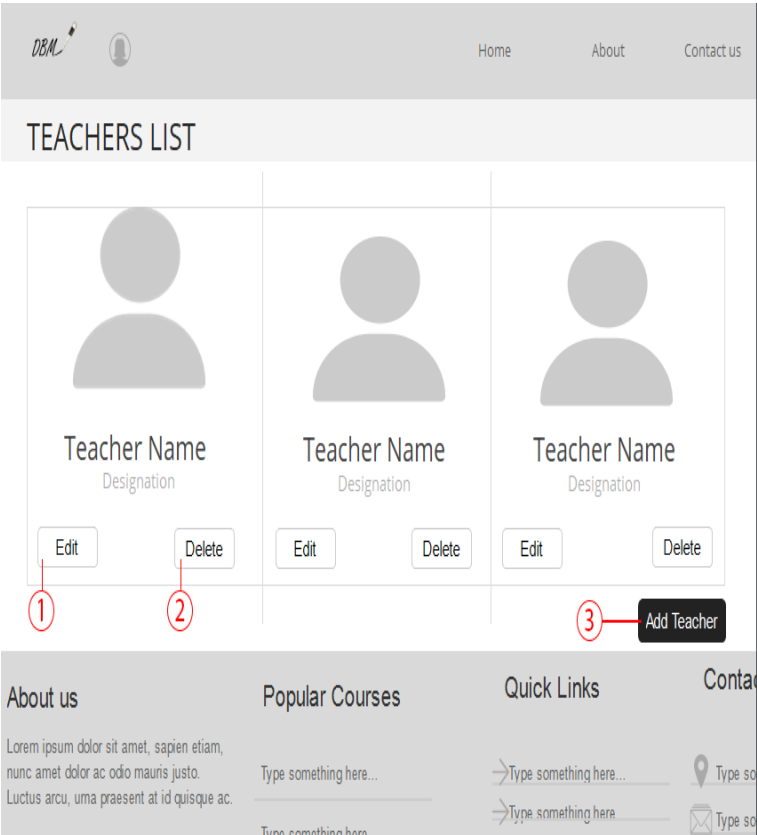


Figure 21: Teacher Details

Component	Description	Mandatory	Validation	Type
1	To edit teachers details.	None	None	None
2	Deletes teacher from the record.	None	None	None
3	Add new teacher in record.	None	None	None

Table 19: Teacher details

4.3.20 Permission Management

The screenshot displays the 'Permission Management' interface. At the top, there is a navigation bar with the 'DBM' logo, a user profile icon, and links for 'Home', 'About', 'Contact us', and 'Logout'. Below this, the 'Permission Management' section is visible. It features a table with four columns: 'GROUP', 'USERS', 'SELECT USER', and 'PERMISSIONS'. The table lists three groups: 'ADMIN', 'TEACHERS', and 'STUDENTS'. Each group row includes a group icon, a name, an 'ALL' checkbox, a 'SELECT USER' dropdown, and a list of three permissions (Permission1, Permission2, Permission3) with checkboxes. Red circles with numbers 1, 2, and 3 highlight specific elements: circle 1 points to the 'Add' button for the 'ADMIN' group's permissions; circle 2 points to the 'Add Group' button at the bottom of the table; and circle 3 points to the 'SELECT USER' dropdown for the 'STUDENTS' group. Below the table, there is a footer section containing 'About us', 'Popular Courses', 'Quick Links', and 'Contact us' blocks, followed by a copyright notice 'Copyrights © | DBM UET Lahore' and social media icons for Twitter, Facebook, and Instagram.

Figure 22: Permission Management

Component	Description	Mandatory	Validation	Type
1	To add new permissions.	None	None	None
2	Select user to allow for permissions.	None	None	None
3	Add new group for permission.	None	None	None

Table 20: Manage Permissions

4.3.21 Add Groups for Permission Management

Figure 23: Add Groups

Component	Description	Mandatory	Validation	Type
1	Group name.	Yes	None	String
2	Save record in database.	None	None	None

Table 21: Add Groups

4.4 User Interfaces (Offline Player)

Offline player has three major sub-modules lecture playlist, token authentication, lecture player.

4.4.1 Lecture Player

4.4.1.1 Short Description

The objective of offline player sub-module Lecture player is to download lectures play offline in the application. The challenge is to move lecture video to a specific position. This application is developed in visual studio as windows form application.

4.4.1.2 Component Diagram

Diagram of lecture player page is shown:

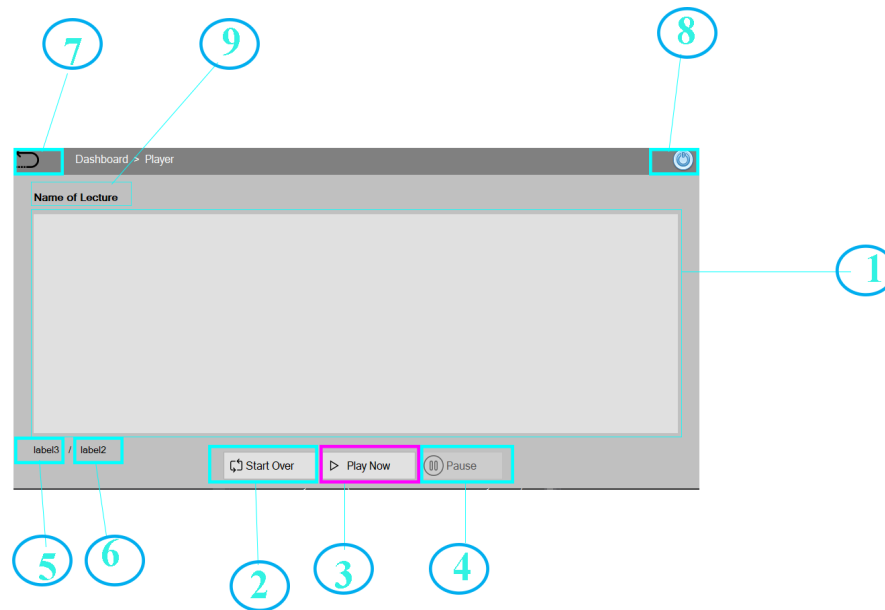
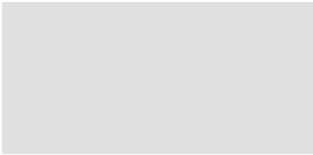
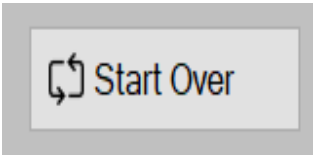
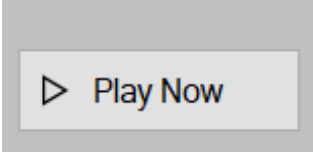

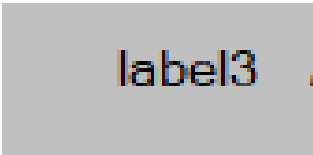
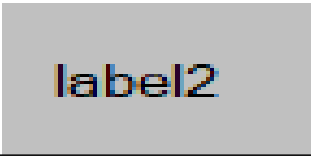


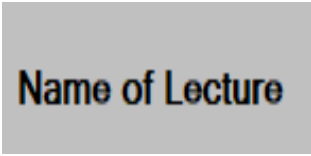


Figure 24: lecture player details

Component number	Description	Component Diagram
1	<p>Name: Player Screen</p> <p>Details: Player screen is a place where animated video drawn by data in json file will be played.</p>	
2	<p>Name: Start Over Button</p> <p>Details: This button is responsible for suspending all the threads except main thread and play the video, audio and timer from the start.</p>	
3	<p>Name: Play Now Button</p> <p>Details: This play now button starts three new threads of audio, video and timer start it is responsible to play video at start.</p>	
4	<p>Name: Pause Button</p> <p>Details: It pauses the audio, video and timer at a specific time when they have been played. It suspends all threads which are responsible to play video for a time being.</p>	
5	<p>Name: Elapsed Time Label</p> <p>Details: This label is responsible to show the elapsed time to the user so that he can keep track of video.</p>	
6	<p>Name: Total Time Label</p>	

	Details: This label shows total time of video at which it will be finished.	
7	Name: Go-Back Button Details: This button takes the user to the previous screen of lecture playlist where he or she can select any other lecture to download and play.	
8	Name: Exit Button Details: As it is quite obvious by the name it exits the application.	
9	Name: Name of Lecture Label Details: This label shows the name of lecture currently playing on the player selected by the user on previous screen.	

4.4.2 Token based Authentication

4.4.2.1 Short Description



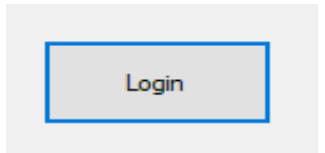
The objective of this sub module is to secure application. Token based authentication with web API will be done and token will be stored in local database. The challenge is we have to keep track of the token expiry on which we are providing privileges to the user. This sub module includes visual studio, and SQLite tools and windows form application technology with C# and SQL Languages.

4.4.2.2 Diagram

Diagram of authentication page is shown:

The diagram shows a mobile application interface for authentication. At the top, there is a dark grey header bar. On the left of the header is a back arrow icon, and on the right is a blue circular icon with a white keyhole. The title 'Authentication' is centered in the header. Below the header, the main area is light grey. It contains two input fields: 'Uername' (note the typo) and 'Password'. Each label is to the left of its respective input field. Both input fields are highlighted with a thick blue border and are connected by a horizontal line to a green circle containing the number '1'. Below these fields is a 'Login' button, which is a grey rectangle with a blue border. A line connects the button to a green circle containing the number '3'. A green circle containing the number '2' is positioned to the right of the 'Password' input field, but it is not connected to any element.

Figure 25: token authentication details

Component number	Description	Component Diagram
1	Name: Username Input Details: Username input takes valid username from user at the time of authentication	
2	Name: Password Input Details: password input takes a password corresponding to given username.	
3	Name: Login Button Details: Login button contacts with WEB API for token info for given username and password.	

4.4.3 Rules and Assumption

Following are rules and cases of assumptions that are assumed to be true while normal working.

- Suitable Operating System must be installed on System such windows XP and higher.
- Suitable Version of .net framework should be installed on system otherwise we have to install app with manifest files.

4.4.4 Module Workflow Description

4.4.4.1 User General Flow

- User starts the application.

- For first time application needs authentication from user to save token information in local database.
- Once user is authenticated, he can view lecture playlist via internet connectivity.
- User can download and play lectures.

4.4.4.2 User General Flow Diagram

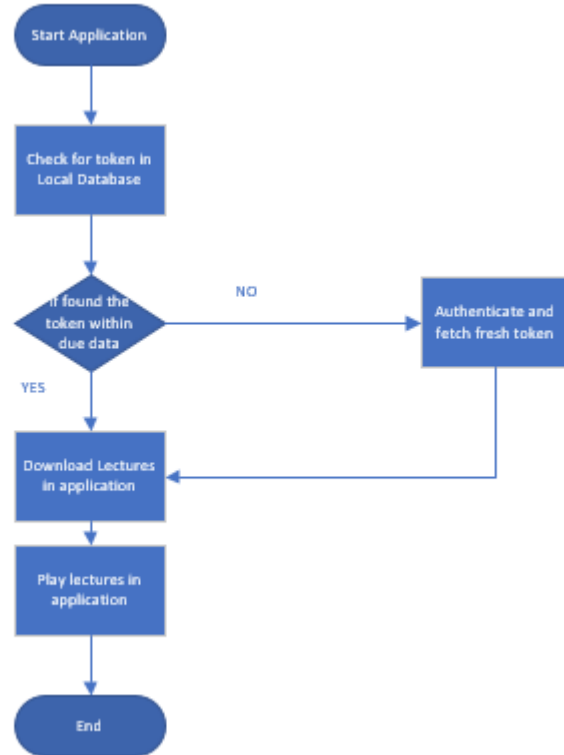


Figure 26: General User Workflow

4.4.4.3 Application General Flow

- Application at start checks in local database for user token and its expiry date if it is expired app asks for authentication again if not it proceeds.
- Application checks for internet connectivity if found it fetches new lecture records from website if not it shows previously fetched and downloaded lectures. Here downloaded lectures can be played.

4.4.4.4 Application General Flow Diagram

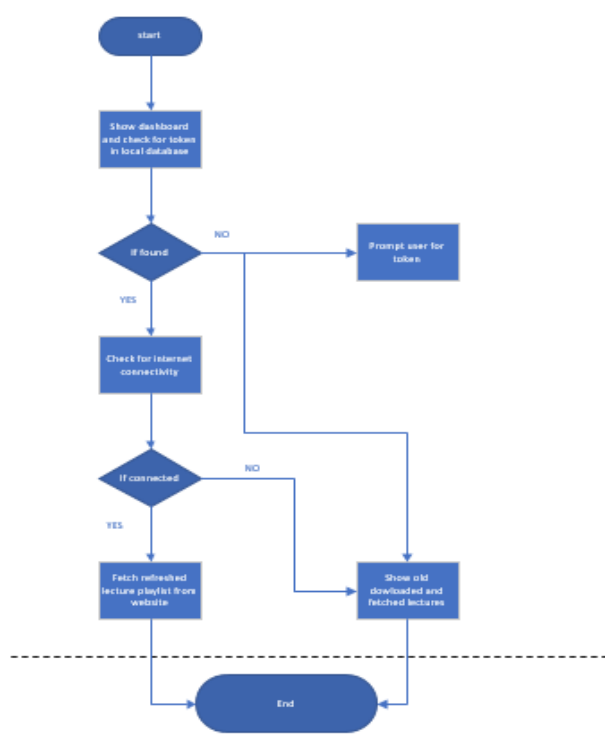


Figure 27: General Application Workflow

4.5 Use Case Diagram

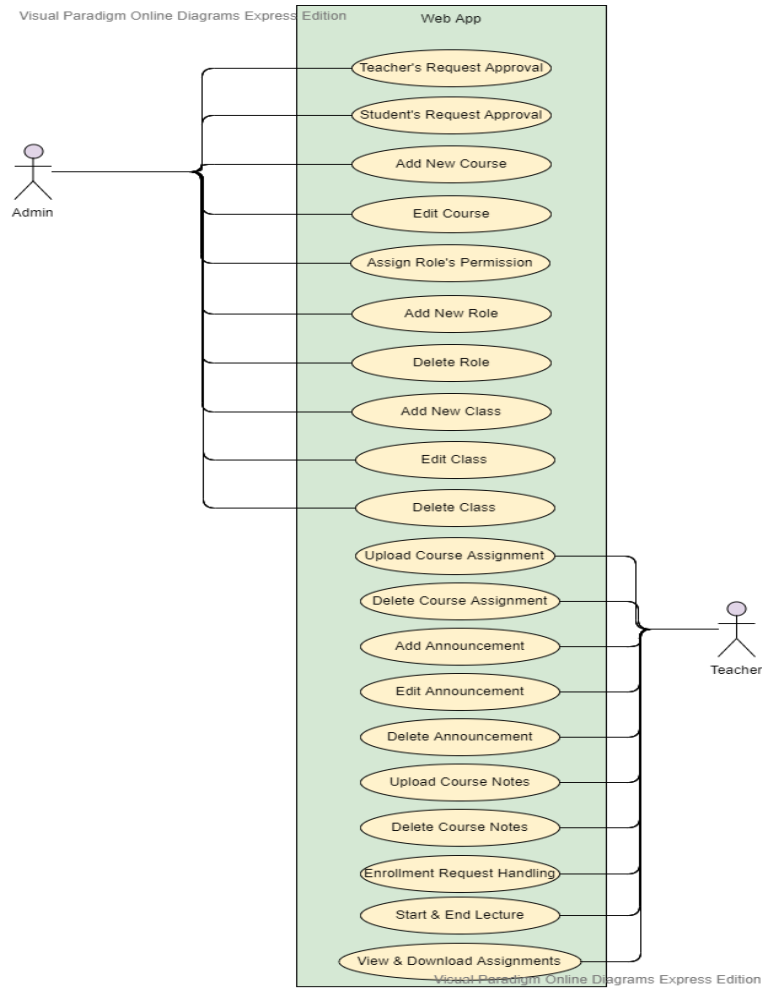


Figure 28: Use Case Diagram Web App(1)

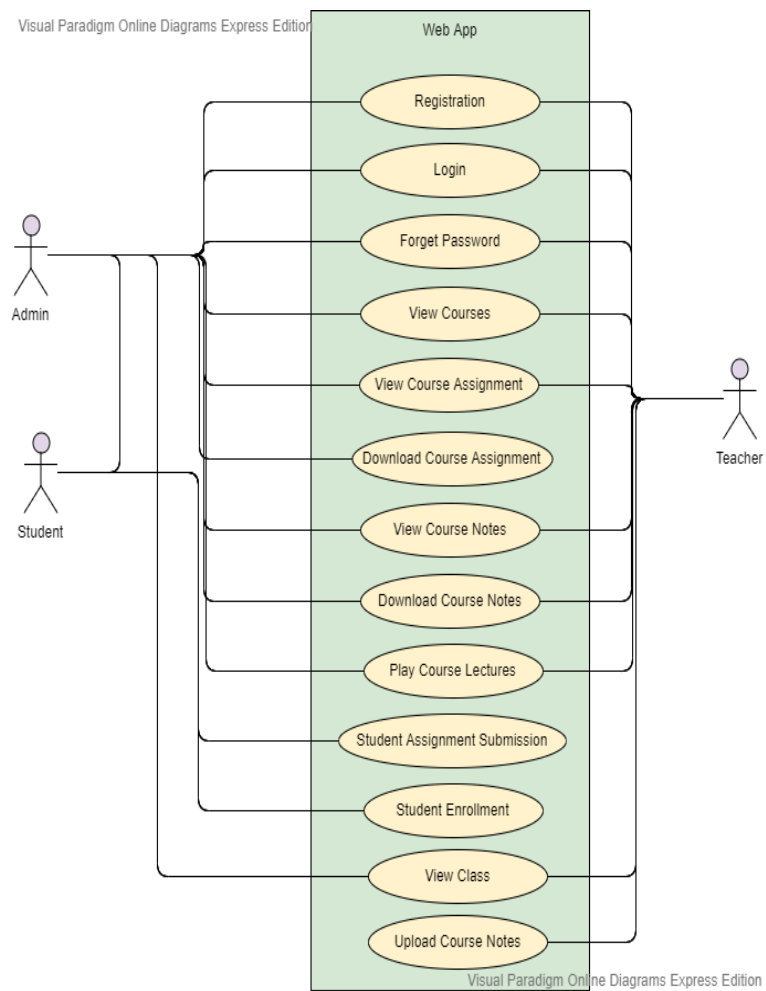


Figure 29: Use Case Diagram Web App(2)

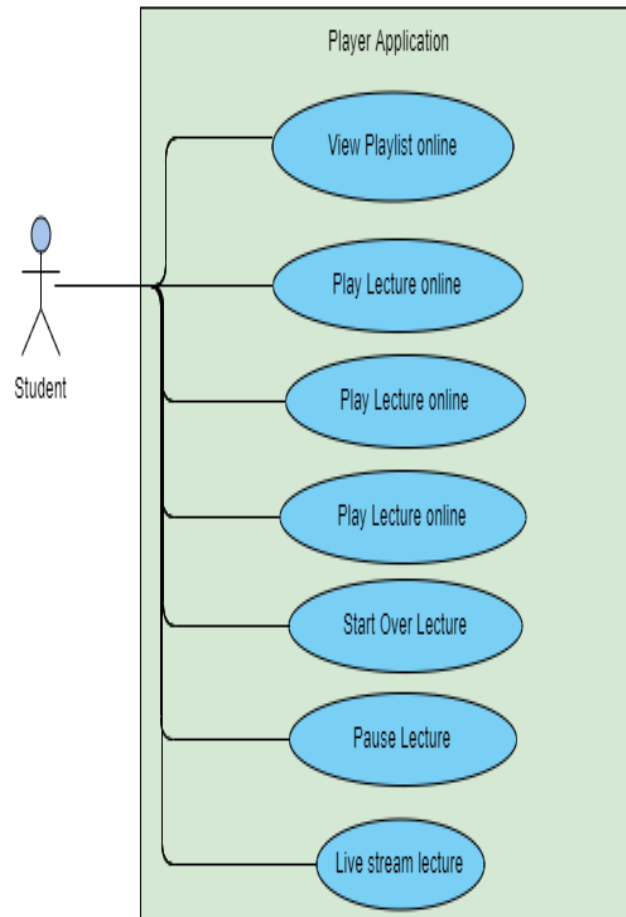


Figure 30: Player Application

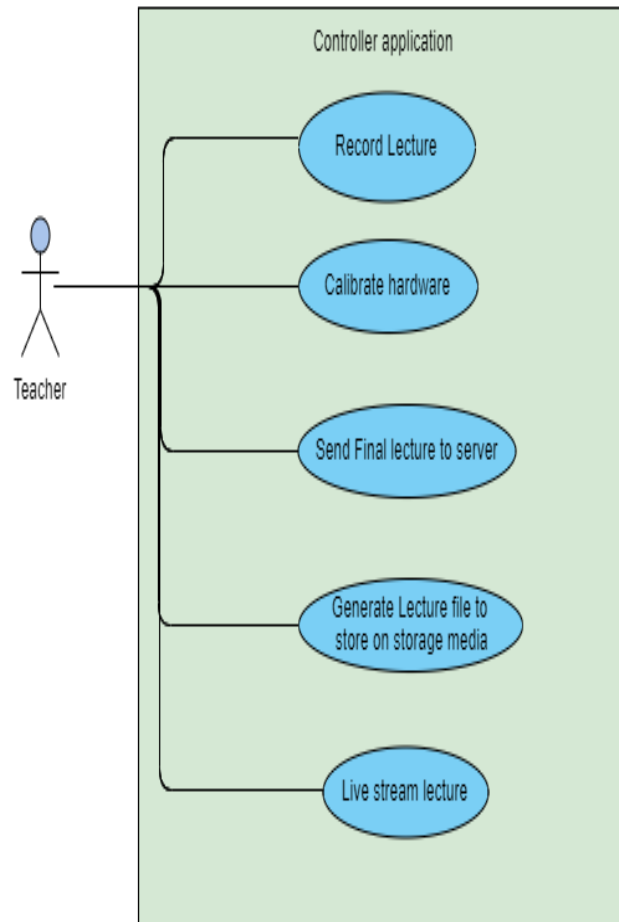


Figure 31: Controller Application

4.6 Use Cases(Web App)

4.6.1 Use Case UC-1: User Registration

Stakeholders: University, Educational Institutes

Primary Actors: Student, Teacher

Post Condition:

User data successfully sent to admin request approval page.

Main Success Scenario:

1. User wants to register.
2. User clicks on the Register link from the main Page header.
3. Registration page will be open by the system.
4. User will enter First Name, Last Name, Email, CNIC, Password and Date of birth.
5. User will select Institute name from the select list.
6. User select whether he is teacher or a student from the select list.
7. If user is student then system will show a input box for student registration No.
8. Student will enter his registration Number.
9. User clicks the **Sign Up** button.
10. System then send data to the Admin Request Approval page for the student request approval.

Alternative Flow:

- a. At any time system fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.

3. Try after sometime might be servers are down.
- 3a. Registration page is not opened.
 1. Refresh the browser.
 2. Reload the page and try again.
- 4a. User entered the invalid First Name, Last Name.
 1. System will show the error message.
 2. System will ask the user to enter the data again.
- 4b. User entered the invalid Email.
 1. System will show the error message.
 2. System will ask the user to enter the data again.
- 4c. User entered the invalid CNIC.
 1. System will show the error message.
 2. System will ask the user to enter the data again.
- 4d. User entered the invalid Registration No.
 1. System will show the error message.
 2. System will ask the user to enter the data again.
- 4e. User entered the invalid Date of birth.
 1. System will show the error message.
 2. System will ask the user to enter the data again.
- 8a. Student entered the registration No. in invalid format.

1. System will show the error message.
 2. System will ask the user to enter the data again.
- 9a. Required fields are empty.
1. System will show the error message.
 2. System will ask the user to complete all the required fields.
- 9b. Sign Up button is not clicked.
1. information will be saved as a draft for later use.

Frequency of Occurrence:

One time per user.

4.6.2 Use Case UC-2: User Login

Stakeholders: University, Educational Institutes

Primary Actors: Student, Teacher, Admin

Preconditions:

- User is approved by the admin.
- User is identified and authenticated.
- User's record is saved.

Post Condition:

- User is logged in into the system.
- User has access to different functionality of the system.

Main Success Scenario:

1. User wants to login.
2. User clicked on the login link to access the login page from the header.
3. System open the login page for the user.
4. User enter email address along with the password.
5. User clicked the **Sign In** button.
6. System validates the user email and password.
7. System will redirect the user to next page.

Alternative Flows:

- a. At any time, system fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 3a. Login page is not opened.
 1. Refresh the browser.
 2. Reload the page and try again.
- 4a. User entered invalid email and password.
 1. System will show the error message.
 2. System will ask the user to enter the data again.
- 5a. Required fields are empty.
 1. System will show the error message.

2. System will ask the user to complete all the required fields.
- 6a. User is not identified or authenticated.
1. System will show error message.
 2. System will ask the user to register yourself first.

Frequency of Occurrence:

Whenever the user wants to login.

4.6.3 Use Case UC-3: Forget Password

Stakeholders: University, Educational Institutes

Primary Actors: Student, Teacher

Precondition:

- User is approved by the admin.
- User is identified and authenticated.
- User's record is saved.
- Login page is opened.

Post Condition:

Password reset successfully by the system.

Main Success Scenario:

1. User forgets his password.
2. User wants to reset his password.
3. User clicks on link **forget password**.
4. System open the **forget password** page for the user.
5. User enter CNIC, new password, confirm password.
6. User click on **Set Password** button.

7. System validate the User.
8. System resets the user password.

Alternative Flows:

- a. At any time, system fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 3a. Forget password page is not opened.
 1. Refresh the browser.
 2. Reload the page and try again.
- 4a. User enter invalid data.
 1. System will show the error message.
 2. System will ask the user to enter the data again.
- 4b. New password and confirm password don't match.
 1. System will show the error message.
 2. System will ask the user to enter the data again.
- 5a. Required fields are empty.
 1. System will show the error message.
 2. System will ask the user to fill all the required fields.
- 6a. User is not identified or authenticated.
 1. System will show error message.
 2. System will ask the user to register yourself first.

Frequency of Occurrence:

Whenever the user wants to reset his password.

4.6.4 Use Case UC-4: Teacher's Requests Approval

Stakeholders: University, Educational Institutes

Primary Actors: Admin

Preconditions:

- Admin is logged in the system..
- Admin is identified and authenticated.

Post Conditions:

- Requests are approved / disapproved.
- On approval / disapproval, Email is sent to the teacher.
- Teachers can login to the system.

Main Success Scenario:

1. Admin wants to handle the requests of teachers.
2. Admin clicks on the Teacher requests approval link from the header.
3. System open the **teachers request approval** page for the admin.
4. Admin can see the list of all the requests of teachers.
5. Admin clicks on the **Approve** button for approval of request.
6. System send the approval email to that teacher.
7. System show the success message to admin that email is sent.
8. System saves the record of the teacher.
9. System removes the request from the page.
10. Admin disapprove the request of a teacher by clicking on **Disapprove** button.
11. System delete that temporary record of the user.

12. System send an disapproval email to the teacher.
13. System show the success message to admin that email is sent.
14. System removes the request from the page.

Alternative Flows:

- a. At any time, system fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 3a. Page is not opened.
 1. Refresh the browser.
 2. Reload the page and try again.
- 5a. Approve button is not working properly.
 1. Admin must have to refresh the page.
 2. Restart the browser and try again.
 3. May be some backend issue, resolve that issue.
- 6-7a. Email is not sent to the teacher.
 1. Admin must again approve the request of teacher.
 2. May be some backend issue, resolve that issue.
- 8-9a. Request is not removed from the page and record is unsaved.
 1. Admin must again approve the request of teacher.

2. May be some backend issue, resolve that issue.
- 10a. Disapprove button is not working properly.
1. Admin must have to refresh the page.
 2. Restart the browser and try again.
 3. May be some backend issue, resolve that issue.
- 12-13a. Email is not sent to the teacher.
1. Admin must again approve the request of teacher.
 2. May be some backend issue, resolve that issue.
- 14a. Request is not removed from the page.
1. Admin must again approve the request of teacher.
 2. May be some backend issue, resolve that issue.

Frequency of Occurrence:

Whenever the requests appear on the page.

Open Issues:

What will happen if admin has mistakenly approve or disapprove the request of teacher?

4.6.5 Use Case UC-5: Student's Requests Approval

Stakeholders: University, Educational Institutes

Primary Actors: Admin

Preconditions:

- Admin is logged in the system..
- Admin is identified and authenticated.

Post Conditions:

- Requests are approved / disapproved.
- On approval / disapproval, Email is sent to the student.
- Students can login to the system.

Main Success Scenario:

1. Admin wants to handle the requests of students.
2. Admin clicks on the student requests approval link from the header.
3. System open the **Students Request Approval** page for the admin.
4. Admin can see the list of all the request of students.
5. Admin clicks on the **Approve** button for approval of request.
6. System send the approval email to that student.
7. System show the success message to admin that email is sent.
8. System saves the record of the student.
9. System removes the request from the page.
10. Admin disapprove the request of a student by clicking on **disapprove** button.
11. System delete that temporary record of the user.
12. System send an disapproval email to the student.
13. System show the success message to admin that email is sent.
14. System removes the request from the page.

Alternative Flows:

- a. At any time, system fails.
 1. User must check the internet connectivity.

2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 3a. Page is not opened.
1. Refresh the browser.
 2. Reload the page and try again.
- 5a. Approve button is not working properly.
1. Admin must have to refresh the page.
 2. Restart the browser and try again.
 3. May be some backend issue, resolve that issue.
- 6-7a. Email is not sent to the student.
1. Admin must again approve the request of student.
 2. May be some backend issue, resolve that issue.
- 8-9a. Request is not removed from the page and record is unsaved.
1. Admin must again approve the request of student.
 2. May be some backend issue, resolve that issue.
- 10a. Disapprove button is not working properly.
1. Admin must have to refresh the page.
 2. Restart the browser and try again.
 3. May be some backend issue, resolve that issue.
- 12-13a. Email is not sent to the teacher.

1. Admin must again approve the request of student.
 2. May be some backend issue, resolve that issue.
- 14a. Request is not removed from the page.

1. Admin must again approve the request of student.
2. May be some backend issue, resolve that issue.

Frequency of Occurrence:

Whenever the admin get requests from students.

Open Issues:

What will happen if admin has mistakenly approve or disapprove the request of student?

4.6.6 Use Case UC-6: Add New Course

Stakeholders: University, Educational Institutes

Primary Actors: Admin

Preconditions:

- Admin is logged in the system.
- Admin is identified and authenticated.

Post Condition:

Course is added successfully.

Main Success Scenario:

1. Admin want to add new course.
2. He clicks on the course link from the header.
3. All courses screen will be open.
4. Admin clicks the **Add Course** button to add new course.
5. System open a new page.

6. Admin add all course related information.

7. Admin clicks the **Submit** button.

Alternative Flows:

a. At any time, System fails.

1. User must check the internet connectivity.

2. Restart the browser and try again.

3. Try after sometime might be servers are down.

2a. Courses link is not shown on the header.

1. Admin must have to refresh the page.

2. Restart the browser and try again.

3a. All courses page is not opened.

1. Admin must check the internet connectivity.

2. Refresh the browser.

3. Reload the page and try again.

4a. Add button not working.

1. Admin must check the internet connectivity.

2. Refresh the browser.

3. Reload the page and try again.

4. May be some backend issue, try to resolve that issue.

5a. New page is not appearing.

1. Admin must check the internet connectivity.

2. Refresh the browser.
 3. Try for sometime, may be servers are down.
- 6a. Admin enter invalid data.
1. System will prompt an error message.
 2. System will ask the user to enter valid data.
- 7a. Required fields are empty.
1. System will prompt an error message.
 2. System will ask the user to enter all the required information.
- 7b. Admin don't click on **Add** button.
1. System will save the record as draft for later use.

Frequency of Occurrence:

Whenever the admin wants to add new course.

4.6.7 Use Case UC-7: Edit Course

Stakeholders: University, Educational Institutes

Primary Actors: Admin

Preconditions:

- Admin is logged in the system.
- Admin is identified and authenticated.
- Course must be added.

Post Condition:

Course is updated successfully.

Main Success Scenario:

1. Admin wants to edit particular course.
2. Admin clicks on the courses link from the header.
3. System open a page of all courses.
4. Admin click on the **Edit** button of course which he wants to update.
5. Course Edit screen will appear.
6. Admin update the required information.
7. Admin clicks the **Update Course** button.
8. System update the course.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
 1. Admin must have to refresh the page.
 2. Restart the browser and try again.
- 3a. All courses page is not opened.
 1. Admin must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.

4a. Edit button not working.

1. Admin must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.
4. May be some backend issue, try to resolve that issue.

5a. Edit Course page is not appearing.

1. Admin must check the internet connectivity.
2. Refresh the browser.
3. Try for sometime, may be servers are down.

6a. Admin enter invalid data.

1. System will prompt an error message.
2. System will ask the user to enter valid data.

7a. Required fields are empty.

1. System will prompt an error message.
2. System will ask the user to enter all the required information.

7b. Admin don't click on **Update Course** button.

1. System will save the record as draft for later use.

Frequency of Occurrence:

Whenever the user wants to update the course information.

4.6.8 Use Case UC-8: Delete Course

Stakeholders: University, Educational Institutes

Primary Actors: Admin

Preconditions:

- Admin is logged in the system.
- Admin is identified and authenticated.
- Course must be added.

Post Condition:

Course is deleted successfully.

Main Success Scenario:

1. Admin wants to delete particular course.
2. Admin clicks on the courses link from the header.
3. System open a page of all courses.
4. Admin click on the **Delete** button of course which he wants to delete.
5. A pop message show "Are you sure you want to delete the course?".
6. Admin clicks the sure button.
7. System delete that course from the record.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.

1. Admin must have to refresh the page.
 2. Restart the browser and try again.
- 3a. All courses page is not opened.
1. Admin must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 4a. Delete button not working.
1. Admin must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
 4. May be some backend issue, try to resolve that issue.
- 5a. Admin clicks the sure button but course is not deleted.
1. May be servers are down. Try again after sometime.
 2. Press again the delete button.

Frequency of Occurrence:

Whenever the admin wants to delete a course. **Open Issues:**
What will happen if admin has mistakenly deleted the course?

4.6.9 Use Case UC-9: View Courses

Stakeholders: University, Educational Institutes

Primary Actors: Admin, Teacher, Student

Preconditions:

- User is logged in the system.

- User is identified and authenticated.
- Courses must be added.

Post Condition:

Courses List is viewed successfully.

Main Success Scenario:

1. Admin wants to view all courses.
2. Admin clicks on the courses link from the header.
3. System open a page of all courses.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
 1. Admin must have to refresh the page.
 2. Restart the browser and try again.
- 3a. All courses page is not opened.
 1. Admin must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.

Frequency of Occurrence:

Whenever the user wants to view all course.

4.6.10 Use Case UC-10: Upload Course Assignment

Stakeholders: University, Educational Institutes

Primary Actors: Teacher

Preconditions:

- Teacher is logged in the system.
- User is identified and authenticated.
- Course is assigned to the teacher.

Post Condition:

Course assignment is uploaded successfully.

Main Success Scenario:

1. Teacher wants to upload course assignment.
2. Teacher clicks on the courses link from the header.
3. System opens a page of all courses.
4. Teacher clicks on the **View** button of course of which he wants to upload assignment.
5. System opens a course dashboard page.
6. Teacher clicks on the **Assignments** tab.
7. System opens the assignments page.
8. Teacher click on **Upload Assignment** button.
9. System opens upload assignment page.
10. Teacher enters all the data.
11. System validates the data.
12. Teacher press the **Upload Assignment** button.
13. System saves the assignment record.

Alternative Flows:

- a. At any time, System fails.
 - 1. User must check the internet connectivity.
 - 2. Restart the browser and try again.
 - 3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
 - 1. User must have to refresh the page.
 - 2. Restart the browser and try again.
- 3a. All courses page is not opened.
 - 1. User must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Reload the page and try again.
- 4a. View button not working.
 - 1. User must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Reload the page and try again.
 - 4. May be some backend issue, try to resolve that issue.
- 5a. Course dashboard page is not opened.
 - 1. User must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Reload the page and try again.

6a. Assignments tab not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

7a. Assignments page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

8a. Upload assignment button not working.

1. Reload the page and try again.
2. Check the internet connectivity.
3. May be some backend issue, try to resolve that issue.

9a. Upload assignment page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

10-11a. User enter invalid data.

1. System will show error message.
2. System will ask the user to enter the data again.

12a. Required fields are empty.

1. System will show error message.
2. System will ask the user to enter the required data.

Frequency of Occurrence:

Whenever user wants to upload assignments.

4.6.11 Use Case UC-11: View Course Assignments

Stakeholders: University, Educational Institutes

Primary Actors: Teacher, Student

Preconditions:

- User is identified and authenticated.
- User is logged in the system.
- Student is enrolled in that course.
- Course is assigned to teacher.

Post Condition:

Course Assignment viewed successfully.

Main Success Scenario:

1. User wants to view course assignment.
2. User clicks on the courses link from the header.
3. System opens a page of all courses.
4. User clicks on the **View** button of course of which he wants to view assignment.
5. System opens a course dashboard page.
6. User clicks on the **Assignments** tab.
7. System opens the assignments page.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.

3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
 1. User must have to refresh the page.
 2. Restart the browser and try again.
- 3a. All courses page is not opened.
 1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 4a. View button not working.
 1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
 4. May be some backend issue, try to resolve that issue.
- 5a. Course dashboard page is not opened.
 1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 6a. Assignments tab not working.
 1. Reload the page and try again.
 2. May be some backend issue, try to resolve that issue.

7a. Assignments page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

Frequency of Occurrence:

Whenever user wants to view assignments.

4.6.12 Use Case UC-12: Download Course Assignment

Stakeholders: University, Educational Institutes

Primary Actors: Teacher, Student

Preconditions:

- User is logged in the system.
- User is identified and authenticated.
- Course is assigned to the teacher.
- Student is enrolled in that course.

Post Condition:

Course assignment is downloaded successfully.

Main Success Scenario:

1. User wants to download course assignment.
2. User clicks on the courses link from the header.
3. System opens a page of all courses.
4. User clicks on the **View** button of course of which he wants to download assignment.
5. System opens a course dashboard page.

6. User clicks on the **Assignments** tab.
7. System opens the assignments page.
8. User click on **Download Assignment** button.
9. System downloads the assignment file for the user.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
 1. User must have to refresh the page.
 2. Restart the browser and try again.
- 3a. All courses page is not opened.
 1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 4a. View button not working.
 1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
 4. May be some backend issue, try to resolve that issue.

5a. Course dashboard page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

6a. Assignments tab not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

7a. Assignments page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

8a. Download assignment button not working.

1. Reload the page and try again.
2. Check the internet connectivity.
3. May be some backend issue, try to resolve that issue.

Frequency of Occurrence:

Whenever user wants to download assignments.

4.6.13 Use Case UC-13: Delete Course Assignment

Stakeholders: University, Educational Institutes

Primary Actors: Teacher

Preconditions:

- Teacher is logged in the system.
- User is identified and authenticated.
- Course is assigned to the teacher.

Post Condition:

Course assignment is deleted successfully.

Main Success Scenario:

1. Teacher wants to delete course assignment.
2. Teacher clicks on the courses link from the header.
3. System opens a page of all courses.
4. Teacher clicks on the **View** button of course of which he wants to upload assignment.
5. System opens a course dashboard page.
6. Teacher clicks on the **Assignments** tab.
7. System opens the assignments page.
8. Teacher click on **Delete Assignment** button.
9. A pop message show "Are you sure you want to delete the assignment?".
10. Teacher clicks the sure button.
11. System delete that assignment from the record.

Alternative Flows:

- a. At any time, System fails.

1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
1. User must have to refresh the page.
 2. Restart the browser and try again.
- 3a. All courses page is not opened.
1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 4a. View button not working.
1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
 4. May be some backend issue, try to resolve that issue.
- 5a. Course dashboard page is not opened.
1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 6a. Assignments tab not working.
1. Reload the page and try again.

2. May be some backend issue, try to resolve that issue.
- 7a. Assignments page is not opened.
1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 8a. Delete assignment button not working.
1. Reload the page and try again.
 2. May be some backend issue, try to resolve that issue.
- 10-11a. Teacher clicks the sure button but assignment is not deleted.
1. May be servers are down. Try again after sometime.
 2. Press again the delete button.

Frequency of Occurrence:

Whenever user wants to delete assignment.

4.6.14 Use Case UC-14: Add Course Announcement

Stakeholders: University, Educational Institutes

Primary Actors: Teacher

Preconditions:

- Teacher is logged in the system.
- Teacher is identified and authenticated.
- Course is assigned to the teacher.

Post Condition:

Course Announcement is added successfully.

Main Success Scenario:

1. Teacher wants to add course related announcement.
2. Teacher clicks on the courses link from the header.
3. System opens a page of all courses.
4. Teacher clicks on the **View** button of course of which he wants to add announcement.
5. System opens a course dashboard page.
6. Teacher clicks on the **Announcement** tab.
7. System opens the announcement page.
8. Teacher click on **Add Announcement** button.
9. System opens add announcement page.
10. Teacher enters all the data.
11. System validates the data.
12. Teacher press the **Add Announcement** button.
13. System saves the announcement record.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.

2a. Courses link is not shown on the header.

1. User must have to refresh the page.
2. Restart the browser and try again.

3a. All courses page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

4a. View button not working.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.
4. May be some backend issue, try to resolve that issue.

5a. Course dashboard page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

6a. Announcement tab not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

7a. Announcement page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

8a. Add announcement button not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

9a. Add announcement page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

Frequency of Occurrence:

Whenever user wants to add announcement.

4.6.15 Use Case UC-15 :Edit Course Announcement

Stakeholders: University, Educational Institutes

Primary Actors: Teacher

Preconditions:

- Teacher is identified and authenticated.
- Teacher is logged in the system.
- Course is assigned to the teacher.

Post Condition:

Course Announcement is updated successfully.

Main Success Scenario:

1. Teacher wants to edit course related announcement.
2. Teacher clicks on the courses link from the header.
3. System opens a page of all courses.
4. Teacher clicks on the **View** button of course of which he wants to edit announcement.
5. System opens a course dashboard page.
6. Teacher clicks on the **Announcement** tab.
7. System opens the announcement page.
8. Teacher click on **Edit Announcement** button of the announcement.
9. System opens edit announcement page.
10. Teacher edit the required data.
11. System validates the data.
12. Teacher press the **Update Announcement** button.
13. System updates the announcement record.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
 1. User must have to refresh the page.

2. Restart the browser and try again.
- 3a. All courses page is not opened.
1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 4a. View button not working.
1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
 4. May be some backend issue, try to resolve that issue.
- 5a. Course dashboard page is not opened.
1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 6a. Announcement tab not working.
1. Reload the page and try again.
 2. May be some backend issue, try to resolve that issue.
- 7a. Announcement page is not opened.
1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.

8a. Edit announcement button not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

9a. Update announcement page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

10a. Teacher enters the invalid data.

1. System prompts the error message.
2. System will ask the user to enter the data again.

12a. Required fields are empty.

1. System prompts the error message.
2. System will ask the user to enter the data in the empty fields.

Frequency of Occurrence:

Whenever user wants to update announcement.

4.6.16 Use Case UC-16: Delete Course Announcement

Stakeholders: University, Educational Institutes

Primary Actors: Teacher

Preconditions:

- User is identified and authenticated.
- User is logged in the system.

- Course is assigned to the teacher.

Post Condition:

Course announcement is deleted successfully.

Main Success Scenario:

1. Teacher wants to delete course announcement.
2. Teacher clicks on the courses link from the header.
3. System opens a page of all courses.
4. Teacher clicks on the **View** button of course of which he wants to delete announcement.
5. System opens a course dashboard page.
6. Teacher clicks on the **Announcements** tab.
7. System opens the announcements page.
8. Teacher click on **Delete Announcement** button.
9. A pop message show "Are you sure you want to delete the announcement?".
10. Teacher clicks the sure button.
11. System delete that announcement from the record.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.

2a. Courses link is not shown on the header.

1. User must have to refresh the page.
2. Restart the browser and try again.

3a. All courses page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

4a. View button not working.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.
4. May be some backend issue, try to resolve that issue.

5a. Course dashboard page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

6a. Announcement tab not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

7a. Announcement page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

8a. Delete announcement button not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

10-11a. Teacher clicks the sure button but announcement is not deleted.

1. May be servers are down. Try again after sometime.
2. Press again the delete button.

Frequency of Occurrence:

Whenever user wants to delete announcement.

Open Issues:

What will happen if teacher have mistakenly delete the announcement?

4.6.17 Use Case UC-17: Student Assignment Submission

Stakeholders: University, Educational Institutes

Primary Actors: Student

Preconditions:

- User is identified and authenticated.
- User is logged in the system.
- Student is enrolled in that course.

Post Condition:

Assignment is submitted successfully.

Main Success Scenario:

1. Student wants to submit course assignment.
2. Student clicks on the courses link from the header.
3. System opens a page of all courses.
4. Student clicks on the **View** button of course of which he wants to submit assignment.
5. System opens a course dashboard page.
6. Student clicks on the **Assignment** tab.
7. System opens the assignment page.
8. Student click on **Submit Assignment** button.
9. System open a new page.
10. Student upload assignment file which he/she wants to submit.
11. Student clicks on the **Submit** button.
12. System save the assignment of student in the record.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.

2a. Courses link is not shown on the header.

1. User must have to refresh the page.
2. Restart the browser and try again.

3a. All courses page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

4a. View button not working.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.
4. May be some backend issue, try to resolve that issue.

5a. Course dashboard page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

6a. Assignment tab not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

7a. Assignment page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

8a. Submit assignment button not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

9a. Submit assignment page is not opened.

1. User must check the internet connectivity.
2. Reload the page and try again.
3. May be some backend issue, try to resolve that issue.

11a. Assignment file is not attached.

1. System will show error message.
2. System will ask the user to attach the file first.

Frequency of Occurrence:

Whenever user wants to submit assignment.

4.6.18 Use Case UC-18: Upload Course Notes

Stakeholders: University, Educational Institutes

Primary Actors: Teacher

Preconditions:

- Teacher is logged in the system.
- User is identified and authenticated.
- Course is assigned to the teacher.

Post Condition:

Course notes uploaded successfully.

Main Success Scenario:

1. Teacher wants to upload course notes.
2. Teacher clicks on the courses link from the header.
3. System opens a page of all courses.
4. Teacher clicks on the **View** button of course of which he wants to upload assignment.
5. System opens a course dashboard page.
6. Teacher clicks on the **Notes** tab.
7. System opens the Notes page.
8. Teacher click on **Add Notes** button.
9. System opens upload notes page.
10. Teacher enters all the data.
11. System validates the data.
12. Teacher press the **Submit** button.
13. System saves the notes.

Alternative Flows:

- a. At any time, System fails.
 - 1. User must check the internet connectivity.
 - 2. Restart the browser and try again.
 - 3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
 - 1. User must have to refresh the page.
 - 2. Restart the browser and try again.
- 3a. All courses page is not opened.
 - 1. User must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Reload the page and try again.
- 4a. View button not working.
 - 1. User must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Reload the page and try again.
 - 4. May be some backend issue, try to resolve that issue.
- 5a. Course dashboard page is not opened.
 - 1. User must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Reload the page and try again.

6a. Notes tab not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

7a. Notes page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

8a. Upload notes button not working.

1. Reload the page and try again.
2. Check the internet connectivity.
3. May be some backend issue, try to resolve that issue.

9a. Upload notes page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

12a. Required fields are empty.

1. System will show error message.
2. System will ask the user to enter the required data.

Frequency of Occurrence:

Whenever user wants to upload course notes.

4.6.19 Use Case UC-19: View Course Notes

Stakeholders: University, Educational Institutes

Primary Actors: Teacher, Student

Preconditions:

- User is identified and authenticated.
- User is logged in the system.
- Student is enrolled in that course.
- Course is assigned to teacher.
- Course Notes are added.

Post Condition:

Course notes viewed successfully.

Main Success Scenario:

1. User wants to view course notes.
2. User clicks on the courses link from the header.
3. System opens a page of all courses.
4. User clicks on the **View** button of course of which he wants to view notes.
5. System opens a course dashboard page.
6. User clicks on the **Notes** tab.
7. System opens the notes page.

Alternative Flows:

a. At any time, System fails.

1. User must check the internet connectivity.

2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
1. User must have to refresh the page.
 2. Restart the browser and try again.
- 3a. All courses page is not opened.
1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 4a. View button not working.
1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
 4. May be some backend issue, try to resolve that issue.
- 5a. Course dashboard page is not opened.
1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 6a. Notes tab not working.
1. Reload the page and try again.
 2. May be some backend issue, try to resolve that issue.

7a. Notes page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

Frequency of Occurrence:

Whenever user wants to view course notes.

4.6.20 Use Case UC-20: Download Course notes

Stakeholders: University, Educational Institutes

Primary Actors: Teacher, Student

Preconditions:

- User is logged in the system.
- User is identified and authenticated.
- Course is assigned to the teacher.
- Student is enrolled in that course.
- Course notes are added.

Post Condition:

Course notes downloaded successfully.

Main Success Scenario:

1. User wants to download course notes.
2. User clicks on the courses link from the header.
3. System opens a page of all courses.
4. User clicks on the **View** button of course of which he wants to download notes.

5. System opens a course dashboard page.
6. User clicks on the **Notes** tab.
7. System opens the notes page.
8. User click on **Download Notes** button.
9. System downloads the notes file for the user.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
 1. User must have to refresh the page.
 2. Restart the browser and try again.
- 3a. All courses page is not opened.
 1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 4a. View button not working.
 1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.

4. May be some backend issue, try to resolve that issue.
- 5a. Course dashboard page is not opened.
 1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 6a. Notes tab not working.
 1. Reload the page and try again.
 2. May be some backend issue, try to resolve that issue.
- 7a. Notes page is not opened.
 1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 8a. Download notes button not working.
 1. Reload the page and try again.
 2. Check the internet connectivity.
 3. May be some backend issue, try to resolve that issue.

Frequency of Occurrence:

Whenever user wants to download notes.

4.6.21 Use Case UC-21: Delete Course Notes

Stakeholders: University, Educational Institutes

Primary Actors: Teacher

Preconditions:

- Teacher is logged in the system.
- User is identified and authenticated.
- Course is assigned to the teacher.
- Course notes are added.

Post Condition:

Course notes deleted successfully.

Main Success Scenario:

1. Teacher wants to delete course notes.
2. Teacher clicks on the courses link from the header.
3. System opens a page of all courses.
4. Teacher clicks on the **View** button of course of which he wants to delete notes.
5. System opens a course dashboard page.
6. Teacher clicks on the **Notes** tab.
7. System opens the notes page.
8. Teacher click on **Delete Notes** button.
9. A pop message show "Are you sure you want to delete the notes?".
10. Teacher clicks the sure button.
11. System delete that notes from the record.

Alternative Flows:

- a. At any time, System fails.
 - 1. User must check the internet connectivity.
 - 2. Restart the browser and try again.
 - 3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
 - 1. User must have to refresh the page.
 - 2. Restart the browser and try again.
- 3a. All courses page is not opened.
 - 1. User must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Reload the page and try again.
- 4a. View button not working.
 - 1. User must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Reload the page and try again.
 - 4. May be some backend issue, try to resolve that issue.
- 5a. Course dasboard page is not opened.
 - 1. User must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Reload the page and try again.

6a. Notes tab not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

7a. Notes page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

8a. Delete notes button not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

10-11a. Teacher clicks the sure button but notes not deleted.

1. May be servers are down. Try again after sometime.
2. Press again the delete button.

Frequency of Occurrence:

Whenever user wants to delete notes.

Open Issues:

What will happen if teacher has mistakenly deleted the course notes?

4.6.22 Use Case UC-22: Play Course Lectures

Stakeholders: University, Educational Institutes

Primary Actors: Teacher, Student

Preconditions:

- User is identified and authenticated.
- User is logged in the system.
- Student is enrolled in that course.
- Course is assigned to teacher.
- Course Lectures are added.

Post Condition:

Course lectures viewed successfully.

Main Success Scenario:

1. User wants to view course lectures.
2. User clicks on the courses link from the header.
3. System opens a page of all courses.
4. User clicks on the **View** button of course of which he wants to view lectures.
5. System opens a course dashboard page.
6. User clicks on the **Lectures** tab.
7. System opens the lectures page.
8. User clicks on the play button.
9. System play the course lecture for the user.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.

3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
 1. User must have to refresh the page.
 2. Restart the browser and try again.
- 3a. All courses page is not opened.
 1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 4a. View button not working.
 1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
 4. May be some backend issue, try to resolve that issue.
- 5a. Course dashboard page is not opened.
 1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 6a. Lectures tab not working.
 1. Reload the page and try again.
 2. May be some backend issue, try to resolve that issue.

7a. Lecture page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

8a. Play lecture button not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

Frequency of Occurrence:

Whenever user wants to play course lectures.

4.6.23 Use Case UC-23: Delete Course Notes

Stakeholders: University, Educational Institutes

Primary Actors: Teacher

Preconditions:

- Teacher is logged in the system.
- User is identified and authenticated.
- Course is assigned to the teacher.
- Course lectures are added.

Post Condition:

Course lectures deleted successfully.

Main Success Scenario:

1. Teacher wants to delete course lectures.
2. Teacher clicks on the courses link from the header.
3. System opens a page of all courses.

4. Teacher clicks on the **View** button of course of which he wants to delete lectures.
5. System opens a course dashboard page.
6. Teacher clicks on the **Lectures** tab.
7. System opens the lectures page.
8. Teacher click on **Delete Lecture** button.
9. A pop message show "Are you sure you want to delete the lecture?".
10. Teacher clicks the sure button.
11. System delete that lecture from the record.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
 1. User must have to refresh the page.
 2. Restart the browser and try again.
- 3a. All courses page is not opened.
 1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.

4a. View button not working.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.
4. May be some backend issue, try to resolve that issue.

5a. Course dashboard page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

6a. Lectures tab not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

7a. lectures page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

8a. Delete lecture button not working.

1. Reload the page and try again.
2. May be some backend issue, try to resolve that issue.

10-11a. Teacher clicks the sure button but lecture not deleted.

1. May be servers are down. Try again after sometime.
2. Press again the delete button.

Frequency of Occurrence:

Whenever user wants to delete lecture.

Open Issues:

What will happen if teacher has mistakenly deleted the course lectures?

4.6.24 Use Case UC-24: Student's Course Enrollment

Stakeholders: University, Educational Institutes

Primary Actors: Student

Preconditions:

- User is identified and authenticated.
- User is logged in the system.
- Course must be added.

Post Condition:

Course enrollment request sent to the teacher successfully.

Main Success Scenario:

1. Student wants to enroll in a course.
2. Student clicks on the courses link from the header.
3. System open a page of all courses.
4. Students clicks on the **Enroll** button of the course.
5. System change the status of enrollment from Not Enrolled to pending.
6. System send the enrollment request to the course teacher.

Alternative Flows:

- a. At any time, System fails.
 - 1. User must check the internet connectivity.
 - 2. Restart the browser and try again.
 - 3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
 - 1. Student must have to refresh the page.
 - 2. Restart the browser and try again.
- 3a. All courses page is not opened.
 - 1. Student must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Reload the page and try again.
- 4a. Enrollment link is not working.
 - 1. Student must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Reload the page and try again.
- 5a. Enrollment status is not changed.
 - 1. Refresh the browser.
 - 2. Reload the page and try again.

Frequency of Occurrence:

Whenever the user wants to enroll in a course.

4.6.25 Use Case UC-25: Course Enrollment Requests Handling

Stakeholders: University, Educational Institutes

Primary Actors: Teacher

Preconditions:

- Teacher is identified and authenticated.
- Teacher is logged in the system.
- Course is added.
- Teacher is assigned a course.

Post Conditions:

- Course enrollment requests are approved / disapproved.
- On approval / disapproval, Email is sent to the student.
- Students can view course details.

Main Success Scenario:

1. Teacher wants to handle the requests of student's course enrollment.
2. Teacher clicks on courses link from the header.
3. System opens a page of course assigned to teacher.
4. Teacher clicks on the enrollment requests button of the course of which he wants to approve/disapprove requests.
5. System open the **Course Enrollment Requests** page for the teacher.
6. Teacher can see the list of all the request of students.
7. Teacher clicks on the **Approve** button for approval of request.
8. System send the approval email to that student.
9. System show the success message to admin that email is sent.

10. System now allow the student to view all course details.
11. System removes the request from the page.
12. Teacher disapprove the request of a student by clicking on **disapprove** button.
13. System delete that temporary record of the user.
14. System send an disapproval email to the student.
15. System show the success message to admin that email is sent.
16. System removes the request from the page.

Alternative Flows:

- a. At any time, system fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 3a. Assigned Course page is not opened.
 1. Refresh the browser.
 2. Reload the page and try again.
- 4a. Enrollment request link is not working.
 1. Teacher must have to refresh the page.
 2. Restart the browser and try again.
 3. May be some backend issue, resolve that issue.

5a. Course Enrollment Requests page is not opened.

1. Refresh the browser.
2. Reload the page and try again.

7a. Approve button is not working properly.

1. User must have to refresh the page.
2. Restart the browser and try again.
3. May be some backend issue, resolve that issue.

8-9a. Email is not sent to the student.

1. User must again approve the request of student.
2. May be some backend issue, resolve that issue.

11a. Request is not removed from the page.

1. User must again approve the request of student.
2. May be some backend issue, resolve that issue.

12a. Disapprove button is not working properly.

1. User must have to refresh the page.
2. Restart the browser and try again.
3. May be some backend issue, resolve that issue.

14a. Email is not sent to the teacher.

1. User must again approve the request of student.
2. May be some backend issue, resolve that issue.

16a. Request is not removed from the page.

1. Admin must again approve the request of student.
2. May be some backend issue, resolve that issue.

Frequency of Occurrence:

Whenever the teacher get course enrollment requests from students.

Open Issues:

What will happen if teacher has mistakenly approve or disapprove the course enrollment request of student?

4.6.26 Use Case UC-26: Assign Role's Permissions

Stakeholders: University, Educational Institutes

Primary Actors: Admin

Preconditions:

- User is identified and authenticated.
- User is logged in the system.
- Roles must be added.

Post Condition:

Permission are assigned to role successfully.

Main Success Scenario:

1. User wants to assign permissions to different roles.
2. User clicks on the permission link from the header.
3. System open a page of permission management.
4. User assign check the check-box of permissions against the role.
5. User clicks the **Assign** button.
6. System gave the assigned permission to the roles.

7. System saves record.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 2a. Permission link is not shown on the header.
 1. Student must have to refresh the page.
 2. Restart the browser and try again.
- 3a. Permission Management page is not opened.
 1. Student must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 4a. Assign button is not working.
 1. Student must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
 4. May be backend issue. Resolve that issue.

Frequency of Occurrence:

Whenever the user wants to assign permission to role.

4.6.27 Use Case UC-27: Add New Role

Stakeholders: University, Educational Institutes

Primary Actors: Admin

Preconditions:

- Admin is identified and authenticated.
- Admin is logged in the system.

Post Condition:

Role is added successfully.

Main Success Scenario:

1. Admin want to add new role/group.
2. He clicks on the permissions link from the header.
3. Permission Management screen will be open.
4. Admin clicks the **Add Group** button to add new role.
5. System open a new page.
6. Admin enter the group name.
7. Admin select the permissions which he want to assign to the role.
8. Admin clicks the **Submit** button.

Alternative Flows:

a. At any time, System fails.

1. User must check the internet connectivity.
2. Restart the browser and try again.
3. Try after sometime might be servers are down.

- 2a. Permissions link is not shown on the header.
 - 1. Admin must have to refresh the page.
 - 2. Restart the browser and try again.
- 3a. Permission Management page is not opened.
 - 1. Admin must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Reload the page and try again.
- 4a. Add Group button is not working.
 - 1. Admin must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Reload the page and try again.
 - 4. May be some backend issue, try to resolve that issue.
- 5a. New page is not appearing.
 - 1. Admin must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Try for sometime, may be servers are down.
- 8a. Required fields are empty.
 - 1. System will prompt an error message.
 - 2. System will ask the user to enter all the required information.
- 8b. Admin don't click on **Submit** button.
 - 1. System will save the record as draft for later use.

Frequency of Occurrence:

Whenever the admin wants to add new role.

4.6.28 Use Case UC-28: Delete Role

Stakeholders: University, Educational Institutes

Primary Actors: Admin

Preconditions:

- Admin is logged in the system.
- Admin is identified and authenticated.
- Role must be added.

Post Condition:

Role is deleted successfully.

Main Success Scenario:

1. Admin wants to delete particular role.
2. Admin clicks on the permissions link from the header.
3. System open a page of Permission Management page.
4. Admin click on the **Delete** button of role which he wants to delete.
5. A pop message show "Are you sure you want to delete the role?".
6. Admin clicks the sure button.
7. System delete that role from the record.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.

2a. Permissions link is not shown on the header.

1. Admin must have to refresh the page.
2. Restart the browser and try again.

3a. Role Management page is not opened.

1. Admin must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

4a. Delete button not working.

1. Admin must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.
4. May be some backend issue, try to resolve that issue.

5a. Admin clicks the sure button but role is not deleted.

1. May be servers are down. Try again after sometime.
2. Press again the delete button.

Frequency of Occurrence:

Whenever the admin wants to delete a role.

Open Issues:

What will happen if admin has mistakenly deleted the role?

4.6.29 Use Case UC-29: Add New Class

Stakeholders: University, Educational Institutes

Primary Actors: Admin

Preconditions:

- Admin is logged in the system.
- Admin is identified and authenticated.

Post Condition:

Class is added successfully.

Main Success Scenario:

1. Admin want to add new class.
2. He clicks on the classes link from the header.
3. All classes screen will be open.
4. Admin clicks the **Add Class** button to add new class.
5. System open a new page.
6. Admin add all class related information.
7. Admin clicks the **Submit** button.

Alternative Flows:

a. At any time, System fails.

1. User must check the internet connectivity.
2. Restart the browser and try again.
3. Try after sometime might be servers are down.

2a. Classes link is not shown on the header.

1. Admin must have to refresh the page.

2. Restart the browser and try again.
- 3a. All classes page is not opened.
1. Admin must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 4a. Add button not working.
1. Admin must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
 4. May be some backend issue, try to resolve that issue.
- 5a. New page is not appearing.
1. Admin must check the internet connectivity.
 2. Refresh the browser.
 3. Try for sometime, may be servers are down.
- 7a. Required fields are empty.
1. System will prompt an error message.
 2. System will ask the user to enter all the required information.
- 7b. Admin don't click on **Add** button.
1. System will save the record as draft for later use.

Frequency of Occurrence:

Whenever the admin wants to add new class.

4.6.30 Use Case UC-30: Edit Class

Stakeholders: University, Educational Institutes

Primary Actors: Admin

Preconditions:

- Admin is logged in the system.
- Admin is identified and authenticated.
- Class must be added.

Post Condition:

Class is updated successfully.

Main Success Scenario:

1. Admin wants to edit particular class.
2. Admin clicks on the classes link from the header.
3. System open a page of all classes.
4. Admin click on the **Edit** button of class which he wants to update.
5. Class Edit screen will appear.
6. Admin update the required information.
7. Admin clicks the **Update Class** button.
8. System update the class.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.

2a. Classes link is not shown on the header.

1. Admin must have to refresh the page.
2. Restart the browser and try again.

3a. All classes page is not opened.

1. Admin must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

4a. Edit button not working.

1. Admin must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.
4. May be some backend issue, try to resolve that issue.

5a. Edit Class page is not appearing.

1. Admin must check the internet connectivity.
2. Refresh the browser.
3. Try for sometime, may be servers are down.

7a. Required fields are empty.

1. System will prompt an error message.
2. System will ask the user to enter all the required information.

7b. Admin don't click on **Update Class** button.

1. System will save the record as draft for later use.

Frequency of Occurrence:

Whenever the user wants to update the class information.

4.6.31 Use Case UC-31: Delete Class

Stakeholders: University, Educational Institutes

Primary Actors: Admin

Preconditions:

- Admin is logged in the system.
- Admin is identified and authenticated.
- Class must be added.

Post Condition:

Class is deleted successfully.

Main Success Scenario:

1. Admin wants to delete particular class.
2. Admin clicks on the classes link from the header.
3. System open a page of all classes.
4. Admin click on the **Delete** button of class which he wants to delete.
5. A pop message show "Are you sure you want to delete the class?".
6. Admin clicks the sure button.
7. System delete that class from the record.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.

2a. Classes link is not shown on the header.

1. Admin must have to refresh the page.
2. Restart the browser and try again.

3a. All classes page is not opened.

1. Admin must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

4a. Delete button not working.

1. Admin must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.
4. May be some backend issue, try to resolve that issue.

5a. Admin clicks the sure button but class is not deleted.

1. May be servers are down. Try again after sometime.
2. Press again the delete button.

Frequency of Occurrence:

Whenever the admin wants to delete a class. **Open Issues:**

What will happen if admin has mistakenly deleted the class?

4.6.32 Use Case UC-32: View Classes

Stakeholders: University, Educational Institutes

Primary Actors: Admin, Teacher

Preconditions:

- User is logged in the system.
- User is identified and authenticated.
- Classes must be added.

Post Condition:

Classes List is viewed successfully.

Main Success Scenario:

1. User wants to view all classes.
2. User clicks on the classes link from the header.
3. System open a page of all classes.

Alternative Flows:

a. At any time, System fails.

1. User must check the internet connectivity.
2. Restart the browser and try again.
3. Try after sometime might be servers are down.

2a. Courses link is not shown on the header.

1. User must have to refresh the page.
2. Restart the browser and try again.

3a. All courses page is not opened.

1. User must check the internet connectivity.
2. Refresh the browser.
3. Reload the page and try again.

Frequency of Occurrence:

Whenever the user wants to view all classes.

4.6.33 Use Case UC-33: Start Lecture

Stakeholders: University, Educational Institutes

Primary Actors: Teacher

Preconditions:

- User is logged in the system.
- User is identified and authenticated.
- Class must be added.

Post Condition:

Lecture recording started successfully.

Main Success Scenario:

1. Teacher wants to start the lecture in the class.
2. User clicks on the **Lecture Commands** link from the header.
3. System open a page of Lecture Commands.
4. User enter the required information.
5. User clicked the start button.

Alternative Flows:

- a. At any time, System fails.
 - 1. User must check the internet connectivity.
 - 2. Restart the browser and try again.
 - 3. Try after sometime might be servers are down.
- 2a. Lecture Commands link is not shown on the header.
 - 1. User must have to refresh the page.
 - 2. Restart the browser and try again.
- 3a. Lecture commands page is not opened.
 - 1. User must check the internet connectivity.
 - 2. Refresh the browser.
 - 3. Reload the page and try again.
- 4a. Required fields are empty.
 - 1. System show error message.
 - 2. System will ask the user to enter the required information.

Frequency of Occurrence:

Whenever the user start the lecture.

4.6.34 Use Case UC-34: End Lecture

Stakeholders: University, Educational Institutes

Primary Actors: Teacher

Preconditions:

- User is logged in the system.
- User is identified and authenticated.
- Class must be added.

Post Condition:

Lecture recording stopped successfully.

Main Success Scenario:

1. Teacher wants to end the lecture in the class.
2. User clicks on the **Lecture Commands** link from the header.
3. System open a page of Lecture Commands.
4. User enter the required information.
5. User clicked the end button.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 2a. Lecture Commands link is not shown on the header.
 1. User must have to refresh the page.

2. Restart the browser and try again.
- 3a. Lecture commands page is not opened.
 1. User must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 4a. Required fields are empty.
 1. System show error message.
 2. System will ask the user to enter the required information.

Frequency of Occurrence:

Whenever the user end the lecture.

Open Issue:

What will happen if teacher mistakenly press the end button but lecture is not ended yet?

4.6.35 Use Case UC-35: Assign Courses

Stakeholders: University, Educational Institutes

Primary Actors: Admin

Preconditions:

- User is logged in the system.
- User is identified and authenticated.
- Courses must be added.

Post Condition:

Course is assigned successfully.

Main Success Scenario:

1. Admin wants to assign courses to teacher.
2. Admin clicks on the courses link from the header.
3. System open a page of all courses.
4. Admin clicks the **View** button.
5. System open the course dashboard.
6. Admin clicks on the **Assign Course** button.
7. System open the assign course page for the user.
8. Admin select the teacher to which he wants to assign course.
9. Admin clicks the **Submit** button.
10. System saves the record i.e assign course to the teacher.

Alternative Flows:

- a. At any time, System fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 2a. Courses link is not shown on the header.
 1. Admin must have to refresh the page.
 2. Restart the browser and try again.
- 3a. All courses page is not opened.
 1. Admin must check the internet connectivity.

2. Refresh the browser.
 3. Reload the page and try again.
- 4a. View button is not working.
1. Reload the page and try again.
 2. May be some backend issue. Admin must resolve it.
- 5a. Course dashboard page is not opened.
1. Admin must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 6a. Assign course button is not working.
1. Reload the page and try again.
 2. May be some backend issue. Admin must resolve it.
- 7a. Assign Course page is not opened.
1. Admin must check the internet connectivity.
 2. Refresh the browser.
 3. Reload the page and try again.
- 9a. User didn't select any teacher.
1. System show error message.
 2. System will ask the user to select atleast one teacher.

Frequency of Occurrence:

Whenever the user wants to assign course.

4.6.36 Use Case UC-36: View Students Assignments

Stakeholders: University, Educational Institutes

Primary Actors: Teacher

Preconditions:

- Teacher is identified and authenticated.
- Teacher is logged in the system.
- Course is added.
- Teacher is assigned a course.
- Course assignments uploaded.

Post Conditions:

Students assignments viewed successfully.

Main Success Scenario:

1. Teacher wants to view students assignments.
2. Teacher clicks on courses link from the header.
3. System opens a page of all courses.
4. Teacher clicks on the course which he is assigned and click on the button of submitted assignments.
5. System open the **Students Assignments** page for the teacher.
6. Teacher can see the list of all the assignments of students.

Alternative Flows:

- a. At any time, system fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.

3. Try after sometime might be servers are down.
- 3a. Assigned Course page is not opened.
 1. Refresh the browser.
 2. Reload the page and try again.
- 4a. Students Assignments button is not working.
 1. Teacher must have to refresh the page.
 2. Restart the browser and try again.
 3. May be some backend issue, resolve that issue.
- 5a. Students Assignments page is not opened.
 1. Refresh the browser.
 2. Reload the page and try again.

Frequency of Occurrence:

Whenever the teacher wants to view the students assignment.

4.6.37 Use Case UC-37: Download Student's Assignments

Stakeholders: University, Educational Institutes

Primary Actors: Teacher

Preconditions:

- Teacher is identified and authenticated.
- Teacher is logged in the system.
- Course is added.
- Teacher is assigned a course.
- Students have submitted the assignments.

Post Conditions:

Teacher downloaded the assignments successfully. **Main Success Scenario:**

1. Teacher wants to download the students assignments.
2. Teacher clicks on assigned courses link from the header.
3. System opens a page of all courses assigned to teacher.
4. Teacher clicks on the **Students Assignments** link.
5. System open the **Students Assignments** page for the teacher.
6. Teacher can see the list of all the students assignments.
7. Teacher clicks on the **Download** button.
8. System download the Assignments files for the user.

Alternative Flows:

- a. At any time, system fails.
 1. User must check the internet connectivity.
 2. Restart the browser and try again.
 3. Try after sometime might be servers are down.
- 3a. Assigned Course page is not opened.
 1. Refresh the browser.
 2. Reload the page and try again.
- 4a. Students Assignments link is not working.
 1. Teacher must have to refresh the page.
 2. Restart the browser and try again.

3. May be some backend issue, resolve that issue.
- 5a. Students Assignments page is not opened.
 1. Refresh the browser.
 2. Reload the page and try again.
- 7a. Download button is not working properly.
 1. User must have to refresh the page.
 2. Restart the browser and try again.
 3. May be some backend issue, resolve that issue.
- 8a. Assignments files not downloaded.
 1. User must have to refresh the page.
 2. Restart the browser and try again.
 3. May be some backend issue, resolve that issue.

Frequency of Occurrence:

Whenever the teacher wants to download the Students Assignments.

4.7 Use Cases (Offline Player)

4.7.1 Use Case UC1: User Authentication

Stakeholders: University, Educational Institutes

Primary Actors: Student, Teacher

Post Condition:

User data successfully sent to server and token successfully saved in local database.

Main Success Scenario:

1. User has to authenticate to use other features.
2. User clicks on the authenticate button on dashboard.
3. Authentication page will be opened.
4. User will enter username and password.
5. User clicks the **Login** button.
6. System then send data to the web api to authenticate user and show proper message based on data.

Alternative Flow:

a*. At any time system fails.

1. User must check the internet connectivity.
 2. Try after sometime might be servers are down.
- 3a. Authentication page is not opened.
1. Make sure the internet connectivity.
 2. try again.
- 4a. User entered the invalid username.
1. System will show the error message.
 2. System will ask the user to enter the data again.
- 4b. User entered the invalid password.
1. System will show the error message.
 2. System will ask the user to enter the data again.

4.7.2 Use Case UC2: View Lecture Playlist(Online)

Stakeholders: University, Educational Institutes

Primary Actors: Student, Teacher

Post Condition:

All the downloaded and recently fetched lectures from website are shown in lecture playlist.

Main Success Scenario:

1. User must have the authentication token in local database which is not expired.
2. User must have the stable internet connection.
3. If token is expired then fresh token can be fetched from website.
4. User will open lecture playlist page from navbar.
5. User clicks the **Fetch Lectures** button on lecture playlist page.
6. System then send request to website with token.
7. All the downloaded lectures will be same and newly fetched lectures will be shown on page.

Alternative Flow:

a*. At any time system fails.

1. Start the application again.
2. User must check the internet connectivity.
3. Try after sometime might be servers are down.

3a. Lecture playlist is not opened.

1. Make sure the internet connectivity and token is present in local database which is not expired.
2. try again.

4.7.3 Use Case UC3: View Lecture Playlist(Offline)

Stakeholders: University, Educational Institutes

Primary Actors: Student, Teacher

Post Condition:

All the downloaded and previously fetched lectures from website are shown in lecture playlist.

Main Success Scenario:

1. User must have the authentication token in local database which is not expired.
2. If token is expired then fresh token can be fetched from website but it requires internet connectivity.
3. User will open lecture playlist page from navbar.
4. All the downloaded and previously fetched lectures will be shown on page.

Alternative Flow:

a*. At any time system fails.

1. Start the application again.
- 3a. Lecture playlist is not opened.
1. Make sure the token is present in local database which is not expired.
 2. try again.

4.7.4 Use Case UC4: Play Lecture(Online)

Stakeholders: University, Educational Institutes

Primary Actors: Student, Teacher

Post Condition:

All the downloaded lectures can be played in lecture player and recently fetched lectures can be downloaded.

Main Success Scenario:

1. User must have the stable internet connection.
2. User must have the authentication token in local database which is not expired.
3. If token is expired then fresh token can be fetched from website.
4. User will open lecture playlist page from navbar.
5. All the downloaded and newly fetched lectures will be shown on page.
6. Then user can download any lecture by clicking on checkbox of that lecture.
7. When checkbox is ticked it means lecture is downloaded user can play that lecture by clicking on **Play** button.
8. It will open the lecture player where user can play, pause and start over that lecture.

Alternative Flow:

a*. At any time system fails.

1. check the internet connectivity.
 2. Start the application again.
- 3a. Lecture player is not opened.
1. Make sure the lecture is downloaded and checkbox is ticked.
 2. Make sure the token is present in local database which is not expired.
 3. try again.

4.7.5 Use Case UC5: Play Lecture(Offline)

Stakeholders: University, Educational Institutes

Primary Actors: Student, Teacher

Post Condition:

All the downloaded lectures can be played in lecture player.

Main Success Scenario:

1. User must have the authentication token in local database which is not expired.
2. If token is expired then fresh token can be fetched from website but it requires internet connectivity.
3. User will open lecture playlist page from navbar.
4. All the downloaded and previously fetched lectures will be shown on page.
5. Then user can play any lecture which is downloaded whose checkbox is ticked.
6. When checkbox is ticked it means lecture is downloaded user can play that lecture by clicking on **Play** button.
7. It will open the lecture player where user can play, pause and start over that lecture.

Alternative Flow:

a*. At any time system fails.

1. Start the application again.
- 3a. Lecture player is not opened.
1. Make sure the lecture you are trying to play is downloaded.
 2. Make sure the token is present in local database which is not expired.
 3. try again.

4.7.6 Use Case UC6: View About page

Stakeholders: University, Educational Institutes

Primary Actors: Student, Teacher

Post Condition:

About page of dbm offline player is shown successfully.

Main Success Scenario:

1. User will open the About page from navbar.
2. User will click **About** button on navbar.
3. About page will be opened.
4. Now user can read about our terms and condition and how to use Offline player.

Alternative Flow:

a*. At any time system fails.

1. Start the application again.
- 3a. About page is not opened.
1. try again.

4.7.7 Use Case UC7: View Contact us page

Stakeholders: University, Educational Institutes

Primary Actors: Student, Teacher

Post Condition:

Contact us page of dbm offline player is shown successfully.

Main Success Scenario:

1. User will open the Contact us page from navbar.
2. User will click **Contact us** button on navbar.
3. Contact us page will be opened.

4. Now user can read contact info in case of any problem user can contact us by that information.

Alternative Flow:

a*. At any time system fails.

1. Start the application again.

3a. Contact us page is not opened.

1. try again.

4.8 Test Cases

4.8.1 Test Case TC-01: User/Student Registration-Web Application

Test Description: Check if user can get register on DBM.

Test Priority: High

Pre-Requisite: Valid email address

Test Steps:

#	Actions	Inputs	Expected Output	Actual Output	Result	Comments
1	User opens registration page.	Enters a url of registration page.	Registration page is opened.	Registration page is opened.	Pass.	Registration page should be opened with all the fields.
2	User enters all the data in valid format and client side validation is done.	Data field Correct and incorrect both.	For correct data no error is displayed and for incorrect data errors are displayed.	For correct data no error is displayed and for incorrect data errors are displayed.	Pass	

3	User clicks the register button.	All the data fields.	For valid data fields data is saved in database via web api and a successfully registered message is displayed.	For valid data fields data is saved in database via web api and a successfully registered message is displayed.	Pass	
---	----------------------------------	----------------------	---	---	------	--

4.8.2 Test Case TC-02: User Authentication-Web Application

Test Description: Check if user can login in web app.

Test Priority: High

Pre-Requisite: user must be registered on web app.

Test Steps:

#	Actions	Inputs	Expected Output	Actual Output	Result	Comments
1	User opens login page.	Enters a url of login page.	Login page is opened.	Login page is opened.	Pass.	Login page should be opened with all the fields.
2	User enters valid email address and password.	Email address and password.	User can see course dashboard.	User can see course dashboard.	Pass	
3	User enters invalid email address or password.	Email address and password.	User can see an error message for invalid email address and password.	User can see an error message for invalid email address and password.	Pass	

4.8.3 Test Case TC-03: View Course dashboard-Web Application

Test Description: Check if user can view course dashboard with all the courses in web app.

Test Priority: High

Pre-Requisite: user must be registered on web app.

Test Steps:

#	Actions	Inputs	Expected Output	Actual Output	Result	Comments
1	User opens login page.	Enters a url of login page.	Login page is opened.	Login page is opened.	Pass.	Login page should be opened with all the fields.
2	User enters valid email address and password.	Email address and password.	User can see course dashboard.	User can see course dashboard.	Pass	
3	User can see all the courses on dashboard.	login is done.	All the courses registered in users institute can be shown now.	All the courses registered in users institute can be shown now.	Pass	

4.8.4 Test Case TC-04: View Course Content-Web Application

Test Description: Check if user can view course content of a course in web app.

Test Priority: High

Pre-Requisite: user must be registered on web app and can login and see course dashboard.

Test Steps:

#	Actions	Inputs	Expected Output	Actual Output	Result	Comments
1	User opens login page.	Enters a url of login page.	Login page is opened.	Login page is opened.	Pass.	Login page should be opened with all the fields.
2	User enters valid email address and password.	Email address and password.	User can see course dashboard.	User can see course dashboard.	Pass	
3	User can see all the courses on dashboard.	login is done.	All the courses registered in users institute can be shown now.	All the courses registered in users institute can be shown now.	Pass	
4	User clicks on a course he is enrolled/assigned in.	course user enrolled/assigned in.	All the course content of the clicked course can be shown now.	All the course content of the clicked course can be shown now.	Pass	

4.8.5 Test Case TC-05: View Course Lectures-Web Application

Test Description: Check if user can view course lectures of a course in web app.

Test Priority: High

Pre-Requisite: user must be registered on web app and can login and see course dashboard.

Test Steps:

#	Actions	Inputs	Expected Output	Actual Output	Result	Comments
1	User opens login page.	Enters a url of login page.	Login page is opened.	Login page is opened.	Pass.	Login page should be opened with all the fields.
2	User enters valid email address and password.	Email address and password.	User can see course dashboard.	User can see course dashboard.	Pass	
3	User can see all the courses on dashboard.	login is done.	All the courses registered in users institute can be shown now.	All the courses registered in users institute can be shown now.	Pass	
4	User clicks on a course he is enrolled/assigned in.	course user enrolled/assigned in.	All the course lectures of the clicked course can be shown now.	All the course lectures of the clicked course can be shown now.	Pass	

4.8.6 Test Case TC-06: View Lecture Notes-Web Application

Test Description: Check if user can view notes of a specific lecture in a course in web app.

Test Priority: High

Pre-Requisite: user must be registered on web app and can login and see course dashboard and can view all the lectures.

Test Steps:

#	Actions	Inputs	Expected Output	Actual Output	Result	Comments
1	User opens login page.	Enters a url of login page.	Login page is opened.	Login page is opened.	Pass.	Login page should be opened with all the fields.
2	User enters valid email address and password.	Email address and password.	User can see course dashboard.	User can see course dashboard.	Pass	
3	User can see all the courses on dashboard.	login is done.	All the courses registered in users institute can be shown now.	All the courses registered in users institute can be shown now.	Pass	
4	User clicks on a course he is enrolled/assigned in.	course user enrolled/assigned in.	All the course lectures of the clicked course can be shown now.	All the course lectures of the clicked course can be shown now.	Pass	
5	User clicks on a lecture of course.	lecture info.	All the notes of that lecture which is clicked can be shown viewed.	All the notes of that lecture which is clicked can be shown viewed.	Pass	

4.8.7 Test Case TC-07: View Lecture Video-Web Application

Test Description: Check if user can view Video of a specific lecture in a course in web app.

Test Priority: High

Pre-Requisite: user must be registered on web app and can login and see course dashboard and can view all the lectures.

Test Steps:

#	Actions	Inputs	Expected Output	Actual Output	Result	Comments
1	User opens login page.	Enters a url of login page.	Login page is opened.	Login page is opened.	Pass.	Login page should be opened with all the fields.
2	User enters valid email address and password.	Email address and password.	User can see course dashboard.	User can see course dashboard.	Pass	
3	User can see all the courses on dashboard.	login is done.	All the courses registered in users institute can be shown now.	All the courses registered in users institute can be shown now.	Pass	
4	User clicks on a course he is enrolled/assigned in.	course user enrolled/assigned in.	All the course lectures of the clicked course can be shown now.	All the course lectures of the clicked course can be shown now.	Pass	

5	User clicks on a lecture of course.	lecture info.	Video of that lecture which is clicked can be shown viewed.	Video of that lecture which is clicked can be shown viewed.		
---	-------------------------------------	---------------	---	---	--	--

4.8.8 Test Case TC-08: View Assignments-Web Application

Test Description: Check if user can view assignments in a course in web app.

Test Priority: High

Pre-Requisite: user must be registered on web app and can login and see course dashboard and can view all the lectures.

Test Steps:

#	Actions	Inputs	Expected Output	Actual Output	Result	Comments
1	User opens login page.	Enters a url of login page.	Login page is opened.	Login page is opened.	Pass.	Login page should be opened with all the fields.
2	User enters valid email address and password.	Email address and password.	User can see course dashboard.	User can see course dashboard.	Pass	
3	User can see all the courses on dashboard.	login is done.	All the courses registered in users institute can be shown now.	All the courses registered in users institute can be shown now.	Pass	

4	User clicks on a course he is enrolled/assigned in.	course user enrolled/assigned in.	All the course lectures and a button of assignment on navbar can be shown now.	All the course lectures and a button of assignment on navbar can be shown now.	Pass	
5	User clicks on a assignment button on navbar.	course info for assignment retrieval.	Table of all the assignment with status can be shown.	Table of all the assignment with status can be shown.	Pass	

4.8.9 Test Case TC-09: View Announcements-Web Application

Test Description: Check if user can view announcements in a course in web app.

Test Priority: High

Pre-Requisite: user must be registered on web app and can login and see course dashboard and can view all the lectures.

Test Steps:

#	Actions	Inputs	Expected Output	Actual Output	Result	Comments
1	User opens login page.	Enters a url of login page.	Login page is opened.	Login page is opened.	Pass.	Login page should be opened with all the fields.
2	User enters valid email address and password.	Email address and password.	User can see course dashboard.	User can see course dashboard.	Pass	

3	User can see all the courses on dashboard.	login is done.	All the courses registered in users institute can be shown now.	All the courses registered in users institute can be shown now.	Pass	
4	User clicks on a course he is enrolled/assigned in.	Course user enrolled/assigned in.	All the course lectures and a button of announcements on navbar can be shown now.	All the course lectures and a button of announcements on navbar can be shown now.	Pass	
5	User clicks on a announcement button on navbar.	Course info for announcement retrieval.	All the announcements with date and time can be shown.	All the announcements with date and time can be shown.	Pass	

4.8.10 Test Case TC-10: Add Assignments(Teacher)-Web Application

Test Description: Check if teacher can add assignment in a course in web app.

Test Priority: High

Pre-Requisite: user must be registered on web app and can login and see course dashboard.

Test Steps:

#	Actions	Inputs	Expected Output	Actual Output	Result	Comments
1	User opens login page.	Enters a url of login page.	Login page is opened.	Login page is opened.	Pass.	Login page should be opened with all the fields.
2	User enters valid email address and password.	Email address and password.	User can see course dashboard.	User can see course dashboard.	Pass	
3	User can see all the courses on dashboard.	Login is done.	All the courses registered in users institute can be shown now.	All the courses registered in users institute can be shown now.	Pass	
4	User clicks on a course he is enrolled/assigned in.	Course user enrolled/assigned in.	All the course lectures and a button of assignment on navbar can be shown now.	All the course lectures and a button of assignment on navbar can be shown now.	Pass	
5	User clicks on assignment button on navbar.	Course info for assignment retrieval.	Table of all the assignment with an add button can be shown.	Table of all the assignment with an add button can be shown.	Pass	

6	User clicks on add assignment button.	Course info for assignment.	Page with all the fields to add assignment can be shown.	Page with all the fields to add assignment can be shown.	Pass	
7	User enters all the fields, file and clicks add button.	fields and file with course info.	A success message should be displayed in case of successfully adding the assignment.	A success message should be displayed in case of successfully adding the assignment.	Pass	

4.8.11 Test Case TC-11: Add Announcement(Teacher)-Web Application

Test Description: Check if teacher can add announcement in a course in web app.

Test Priority: High

Pre-Requisite: user must be registered on web app and can login and see course dashboard.

Test Steps:

#	Actions	Inputs	Expected Output	Actual Output	Result	Comments
1	User opens login page.	Enters a url of login page.	Login page is opened.	Login page is opened.	Pass.	Login page should be opened with all the fields.

2	User enters valid email address and password.	Email address and password.	User can see course dashboard.	User can see course dashboard.	Pass	
3	User can see all the courses on dashboard.	Login is done.	All the courses registered in users institute can be shown now.	All the courses registered in users institute can be shown now.	Pass	
4	User clicks on a course he is enrolled/assigned in.	Course user enrolled/assigned in.	All the course lectures and a button of announcement on navbar can be shown now.	All the course lectures and a button of announcement on navbar can be shown now.	Pass	
5	User clicks on announcement button on navbar.	Course info for announcement retrieval.	Table of all the announcement with an add button can be shown.	Table of all the announcement with an add button can be shown.	Pass	
6	User clicks on add announcement button.	Course info for announcement.	Page with all the fields to add announcement can be shown.	Page with all the fields to add announcement can be shown.	Pass	

7	User enters all the fields, file and clicks add button.	fields and file with course info.	A success message should be displayed in case of successfully adding the announcement.	A success message should be displayed in case of successfully adding the announcement.	Pass	
---	---	-----------------------------------	--	--	------	--

4.8.12 Test Case TC-12: Add Institute(Super Admin)-Web Application

Test Description: Check if Admin can add institute in web app.

Test Priority: High

Pre-Requisite: Admin can login and see navbar with insitute tab.

Test Steps:

#	Actions	Inputs	Expected Output	Actual Output	Result	Comments
1	User opens login page.	Enters a url of login page.	Login page is opened.	Login page is opened.	Pass.	Login page should be opened with all the fields.
2	User enters valid email address and password.	Email address and password.	User can see all admin related options on navbar.	User can see all admin related options on navbar.	Pass	

3	Admin clicks on institute tab.	Admin info.	All the institutes registered in web app and an add button can be shown now.	All the institutes registered in web app and an add button can be shown now.	Pass	
4	Admin clicks on add button.	Admin info.	Page with institute field will be viewed now.	Page with institute field will be viewed now.	Pass	
5	Admin enters all the fields and clicks add button.	institute info.	A success message will be displayed when institute is added in web app.	A success message will be displayed when institute is added in web app.	Pass	

4.8.13 Test Case TC-13: Add Admin(Super Admin)-Web Application

Test Description: Check if Super Admin can add admin for a institute in web app.

Test Priority: High

Pre-Requisite: Admin can login and see navbar with Add Admin tab.

Test Steps:

#	Actions	Inputs	Expected Output	Actual Output	Result	Comments
1	User opens login page.	Enters a url of login page.	Login page is opened.	Login page is opened.	Pass.	Login page should be opened with all the fields.

2	User enters valid email address and password.	Email address and password.	User can see all admin related options on navbar.	User can see all admin related options on navbar.	Pass	
3	Admin clicks on Admin tab.	Admin info.	All the Admins registered with a certain institute in web app and an add button can be shown now.	All the Admins registered with a certain institute in web app and an add button can be shown now.	Pass	
4	Admin clicks on add button.	Admin info.	Page with institute dropdown and admin credentials inputs will be viewed now.	Page with institute dropdown and admin credentials inputs will be viewed now.	Pass	
5	Admin enters all the fields and clicks add button.	Admin info.	A success message will be displayed when admin of an institute is added in web app.	A success message will be displayed when admin of an institute is added in web app.	Pass	

5 Planning

5.1 Milestones

Module Name	Time Required
Marker Hardware	1 Month
Voice Transceiver	1 Month
Controller Application	2 Months
Offline Player Application	2 Months
LMS	4 Months
WebGL Player	1 Month

Table 37: Milestones

5.2 Work Breakdown Structure (WBS)

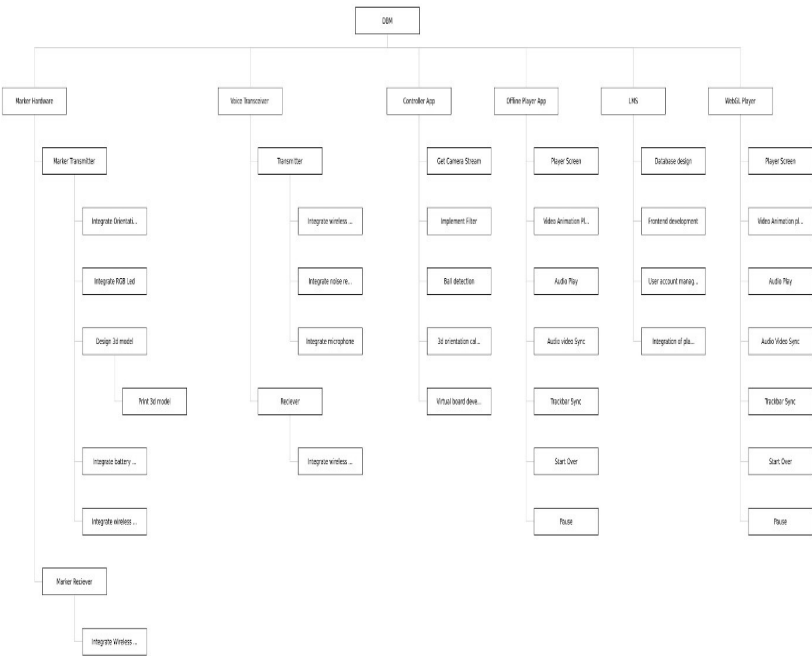


Figure 32: WBS

5.3 Network Dependency Diagram

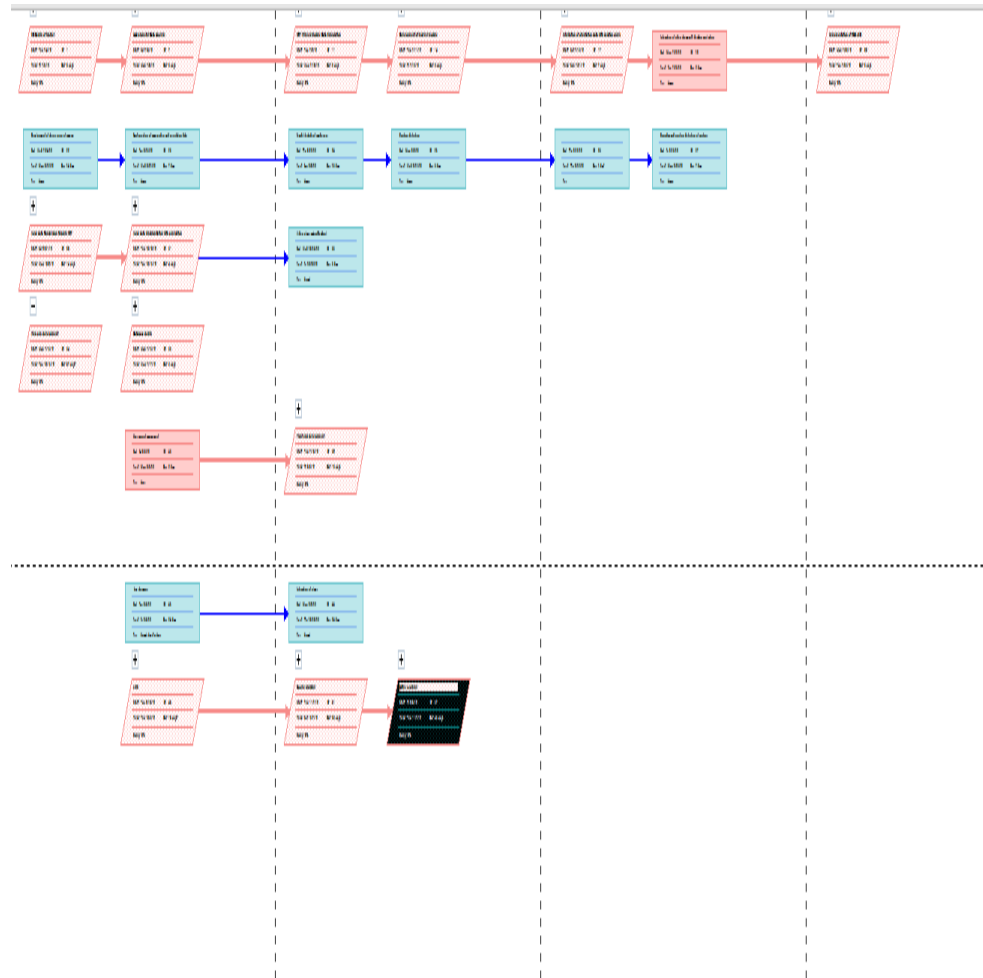


Figure 33: Network Dependency Diagram

5.4 Gantt Chart

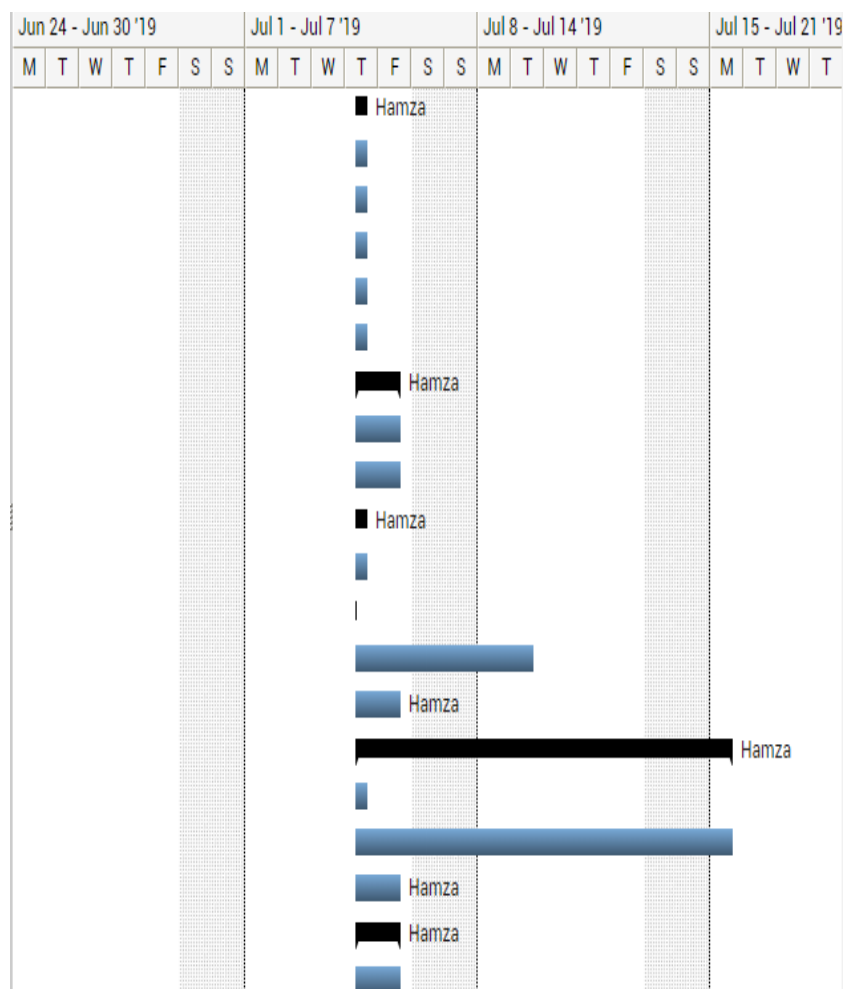


Figure 34: Gantt Chart(1)

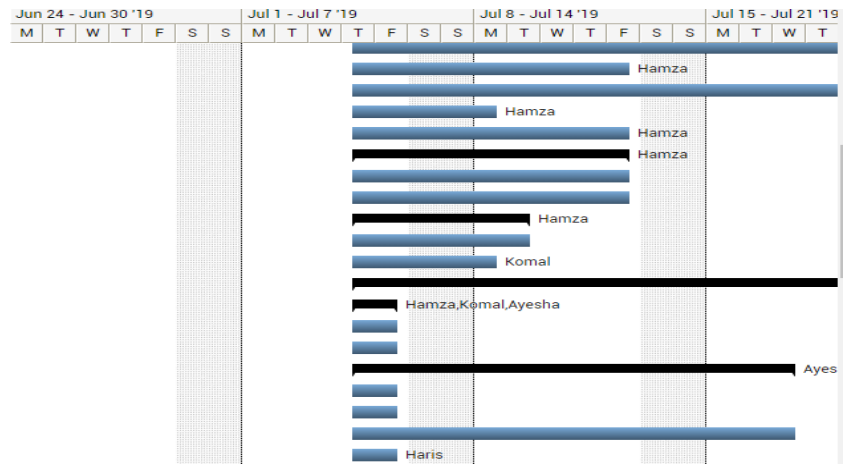


Figure 35: Gantt Chart(2)

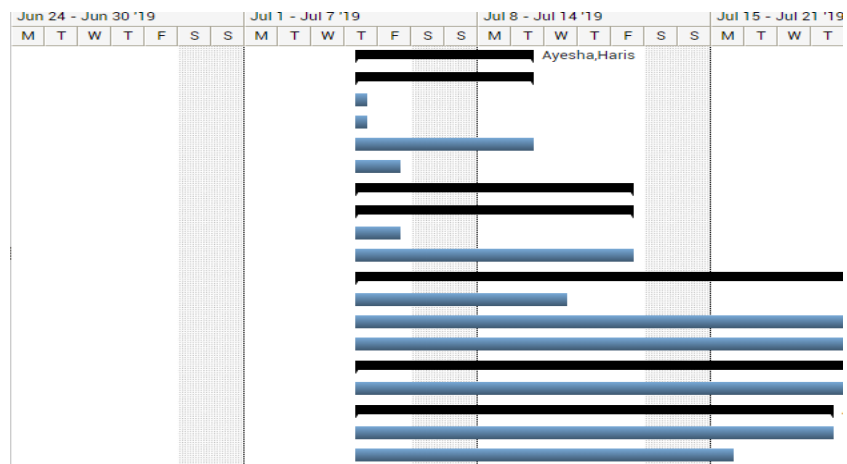


Figure 36: Gantt Chart(3)

6 Design

6.1 Data Flow Diagrams (Web App)

6.1.1 Student

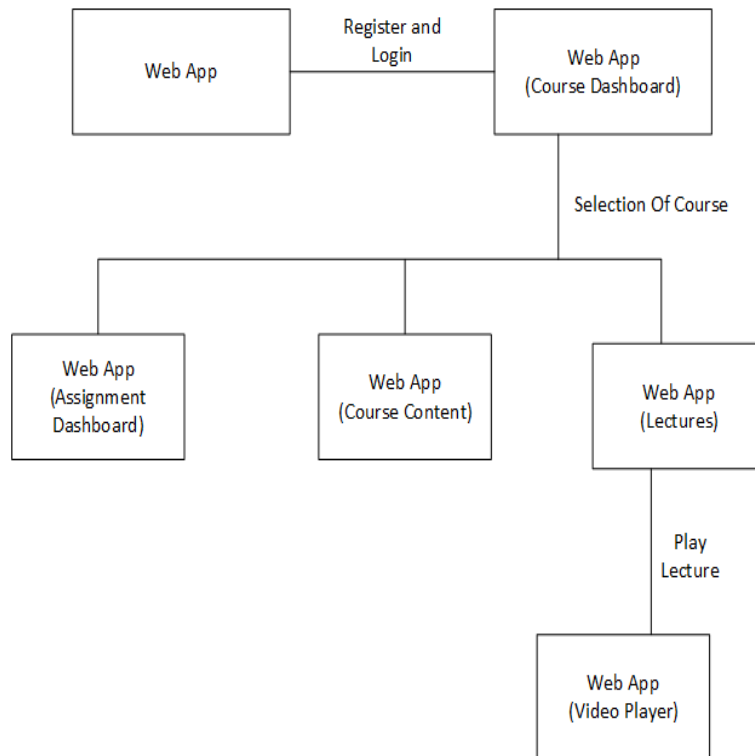


Figure 37: Working flow for Student

6.1.2 Teacher

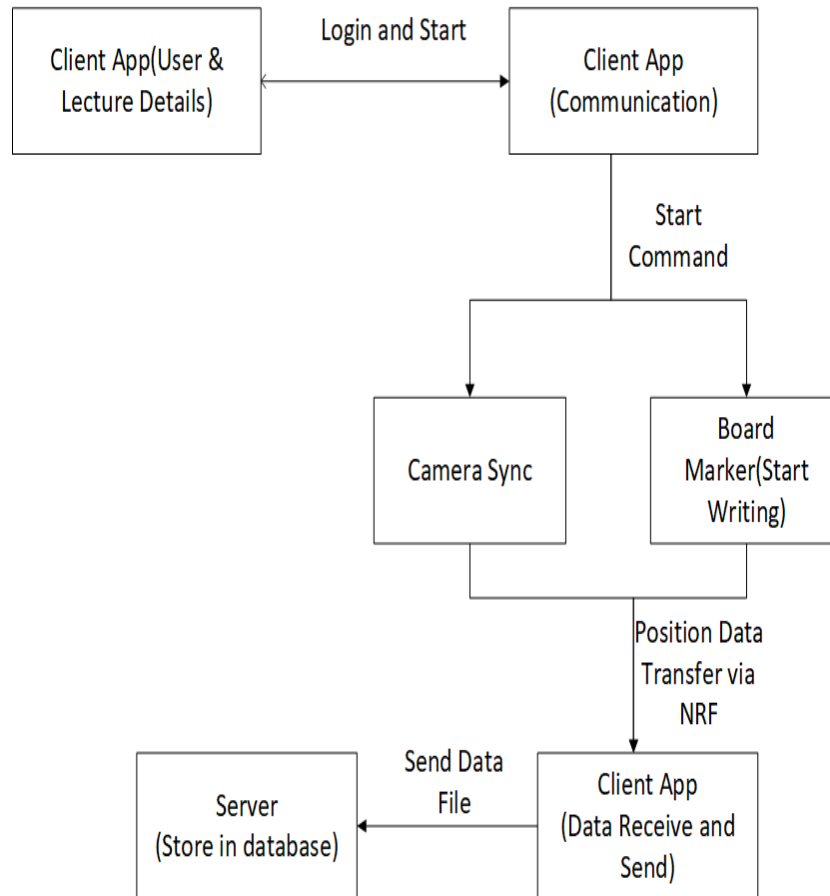


Figure 38: Working flow for teacher

6.2 Data Flow Diagrams (Offline Player)

6.2.1 User General Flow

- User starts the application.
- For first time application needs authentication from user to save token information in local database.

- Once user is authenticated, he can view lecture playlist via internet connectivity.
- User can download and play lectures.

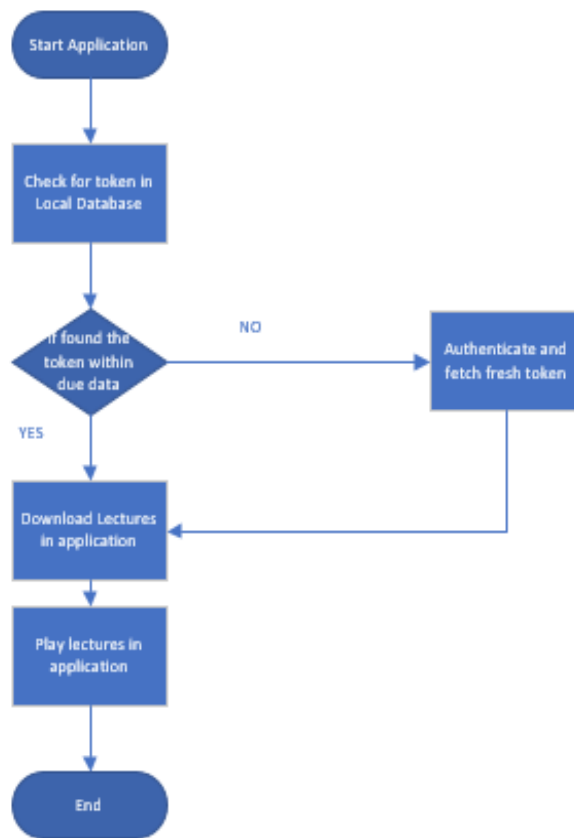


Figure 39: General User Workflow

6.2.2 General Flow

- Application at start checks in local database for user token and its expiry date if it is expired app asks for authentication again if not it proceeds.
- Application checks for internet connectivity if found it fetches new lecture records from website if not it shows previously fetched and downloaded lectures. Here downloaded lectures can be played.

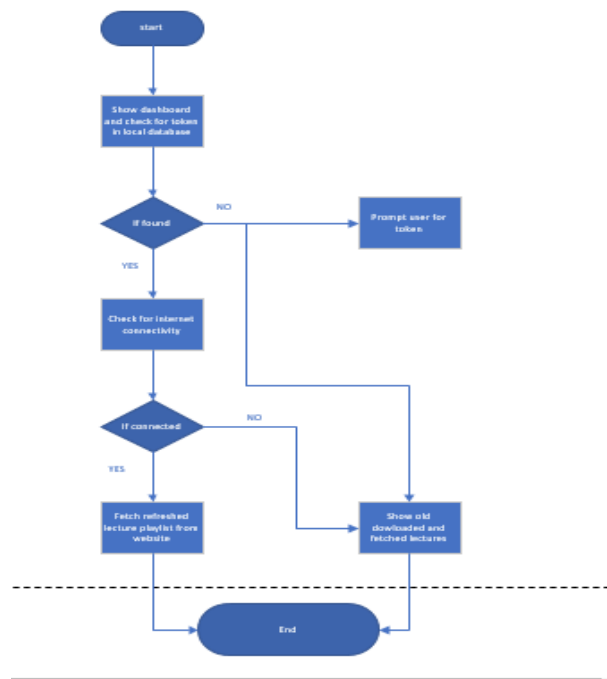


Figure 40: General Application Workflow

6.3 Activity Diagrams

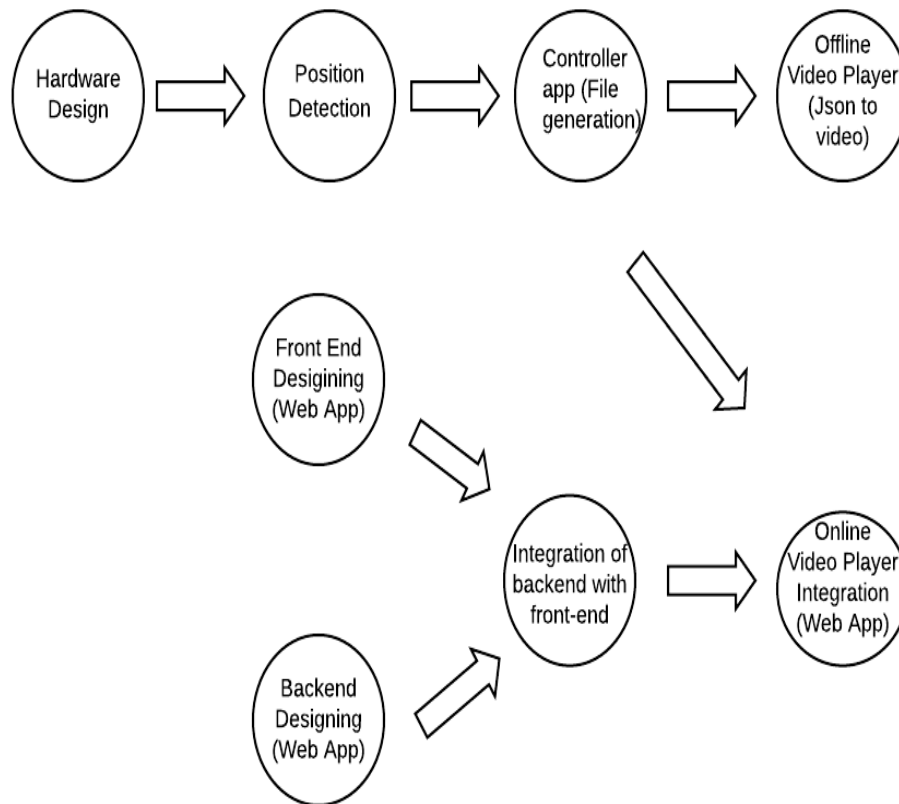


Figure 41: General Application Workflow

6.4 Data Modeling

6.4.1 ER Diagram

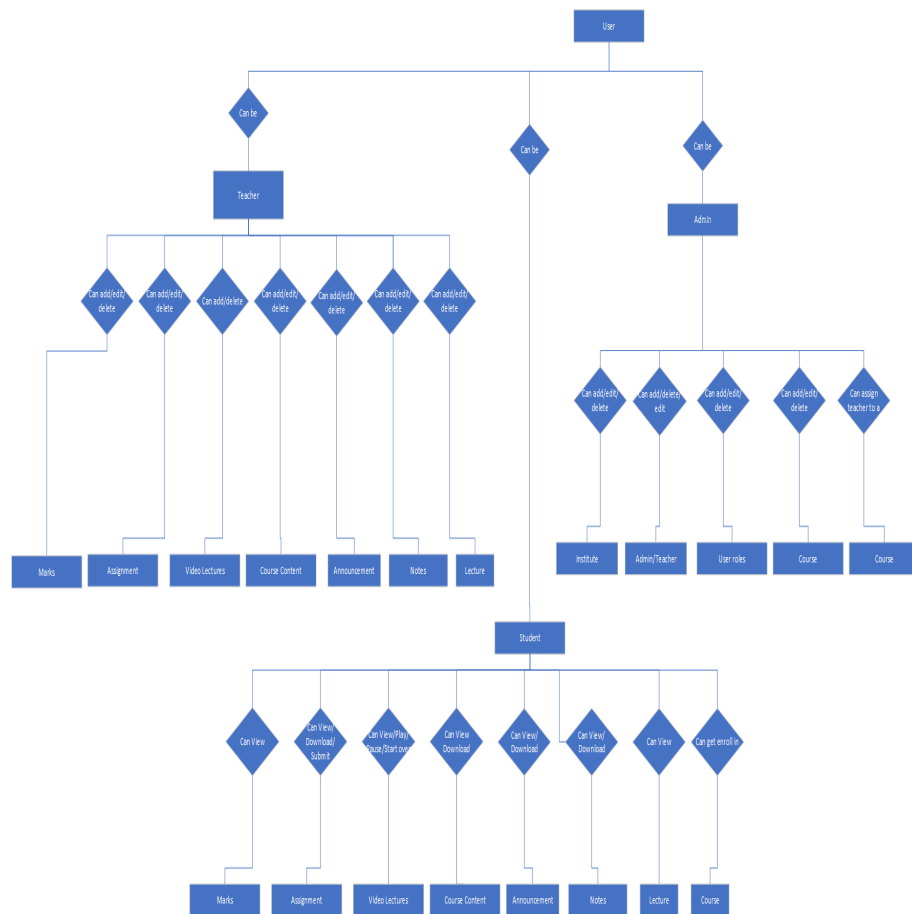


Figure 42: Entity Diagram

6.4.2 Database Diagram

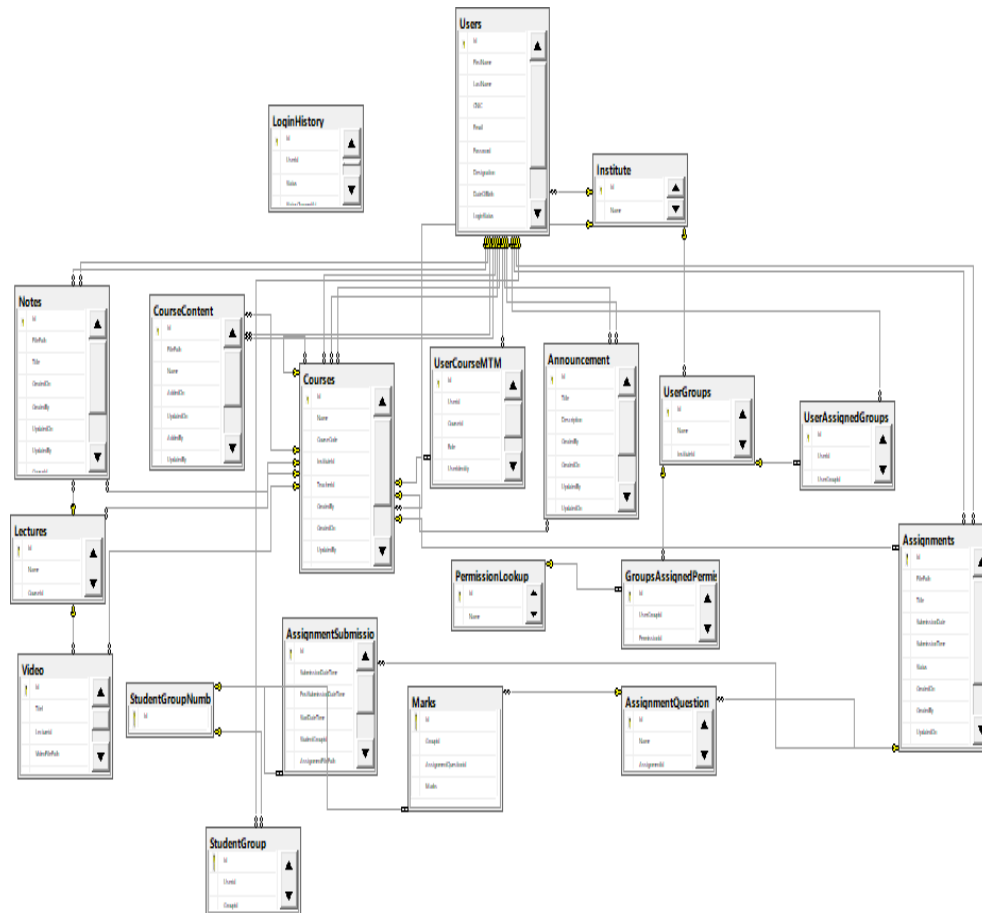


Figure 43: Database Diagram

6.5 Class Diagram

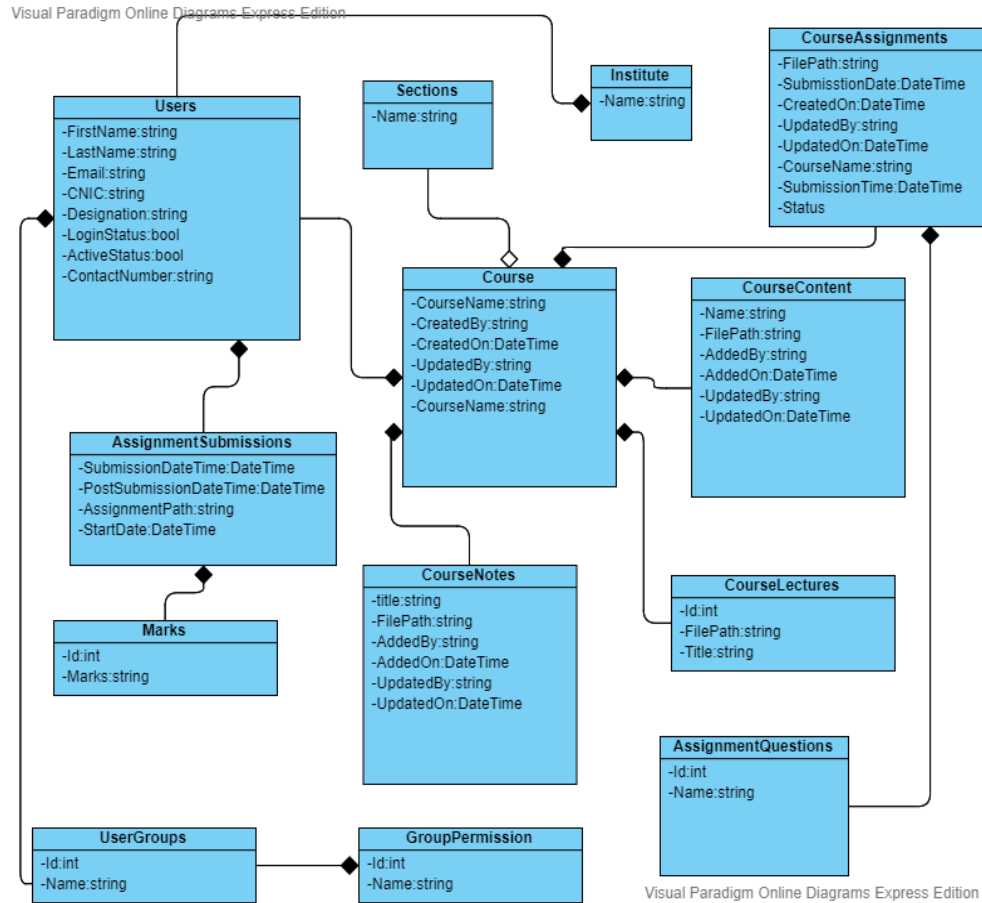


Figure 44: Class Diagram