

**A PROJECT REPORT ON  
ONLINE ASSIGNMENT MANAGEMENT SYSTEM  
(Online AMS)**

Submitted in partial fulfillment of the requirements for  
the degree of Bachelor of Information and  
Communication Technology in Education  
at Tribhuvan University

**BY  
ANJU GURUNG  
SANDEEPA GURUNG**

**Er. GHAN BAHADUR THAPA  
(Supervisor)**



**Department of Science and Technology  
AADIKAVI BHANUBHAKTA CAMPUS**

Vyas-01, Damauli, Nepal

**(January, 2023)**

## **ACKNOWLEDGEMENT**

The success of this project would not have been possible without the support and guidelines of many individuals, and we are very thankful to those who helped us during our project. We would like to express our project Supervisor **Er. Ghan Bahadur Thapa** who took interest on our project and guide us throughout the project by providing all the necessary ideas, information and knowledge for the project.

We are highly indebted to Aadikavi Bhanubhakta Campus for constant guidance and supervision, as well as for providing all the necessary infrastructures and friendly environment for the successful completion of the project. We would like to thank Mr.Maha Prasad Hadkhale, Campus Chief, Aadikavi Bhanubhakta Campus, for his moral support toward completing our project work. Without his support, the project would not have been completed.

Last, but not the least, we would like to thanks our peers and friends who provided us with valuable suggestions to improve our project.

**Anju Gurung (74288004/074)**

**Sandeepa Gurung (74288020/074)**

## **ABSTRACT**

The manual assignment system is becoming obsolete due to its inconvenience, inefficiency, and low accuracy. After the pandemic time of Covid 19, web-based management systems have been widely implemented due to the development of Web technologies. Therefore in order to effectively manage student assignments in a well-managed way and provide appropriate information to students, a well-designed assignment submission system is needed, hence the need for an online assignment management system to facilitate the collection and marking of the assignments scripts. This project introduces a web-based Assignment Management System which automates all the manual aspects of the conventional way of handling assignments.

The system was created to eliminate the manual process of allocating assignments and collecting feedback for the assignment. The system was implemented as a web-based application through the useful tools of React js, Tailwinds CSS, and JavaScript as the front design while using MySQL for storing and retrieving data as the back-end tool. The main goal of this project is to design and implement online assignment submission and provide an interface used for uploading and downloading assignments by the teacher and the submission by the students within a time, while the teacher also performs the task of grading each student submission automatically. The most obvious advantage offered by this online assignment management system is that it offers an easy and faster transmission of assignments between both the instructor and the students than using the manual process.

# Table of Contents

## PRELIMINARY PAGES

TITLE .....	i
BONAFIDE CERTIFICATE .....	ii
SUPERVISOR’S RECOMMENDATION .....	iii
CERTIFICATE OF APPROVAL .....	iv
ACKNOWLEDGEMENT .....	v
ABSTRACT .....	vi
LIST OF FIGURES.....	x
LIST OF TABLES .....	xii
LIST OF ABBREVIATION .....	xiv

## CHAPTER 1: INTRODUCTION

1.1. Introduction.....	1
1.2. Background of the study.....	2
1.3. Problem Statement and Motivation.....	4
1.4. Objectives .....	4
1.4.1 Primary Objective.....	4
1.4.2 Secondary Objective .....	4
1.5. Project Scope and Direction .....	5
1.6. Limitations .....	5
1.7. Report Organization .....	5

## CHAPTER 2: REQUIREMENT ANALYSIS

2.1. Literature Review.....	6
2.2. Problem Definition .....	8
2.3. Proposed Solution .....	8
2.4. Requirement Analysis .....	8
2.4.1. Functional Requirement.....	9
2.4.2. Non-Functional Requirement.....	9

2.5. Feasibility Study.....	10
2.5.1. Economic Feasibility .....	10
2.5.2. Technical Feasibility .....	11
2.5.3. Operational Feasibility.....	11
2.5.4. Schedule Feasibility .....	11
2.6. Structuring System Requirements .....	14
2.6.1. Use Case Diagram .....	15
2.6.2. ER Diagram.....	19
2.6.3. Data Flow Diagram.....	21
2.6.4. Class Diagram.....	22
2.6.5. Sequence Diagram.....	23
2.6.6. Activity Diagram .....	24
 CHAPTER 3: SYSTEM DESIGN	
3.1. System Architecture and Overview.....	26
3.2. System Design.....	27
3.2.1. Database Schema.....	28
3.3. Data Dictionary .....	29
 CHAPTER 4: SYSTEM IMPLEMENTATION & TESTING	
4.1. Implementation and Overview.....	34
4.2. Tools Used.....	35
4.2.1. Front End Tool.....	35
4.2.2. Back End Tool.....	35
4.3. Testing .....	36
4.3.1. Unit Testing .....	36
4.3.2. Integration Testing .....	39
4.3.3. System Testing .....	40

## CHAPTER 5: CONCLUSION AND RECOMMENDATION

5.1. Conclusion.....	41
5.2. Recommendation .....	41
REFERENCES.....	42
APPENDIX – A: User Interface .....	43
APPENDIX – B: System Coding .....	49

## LIST OF FIGURES

Figure 2.1: Gantt chart for task 1.....	12
Figure 2.2: Gantt chart for task 2 .....	13
Figure 2.3: Gantt chart for task 3 .....	13
Figure 2.4: Gantt chart for schedule.....	14
Figure 2.5: Use Case Diagram.....	15
Figure 2.6: Entity Relationship Diagram.....	19
Figure 2.7: Data Flow Diagram Level-0 .....	20
Figure 2.8: Data Flow Diagram Level-1 .....	21
Figure 2.9: UML class Diagram .....	22
Figure 2.10: Sequence Diagram of Teacher .....	23
Figure 2.11: Sequence Diagram of Student .....	23
Figure 2.12: Activity Diagram of Users .....	24
Figure 2.13: Activity Diagram of Teacher.....	24
Figure 2.14: Activity Diagram of Student .....	25
Figure 3.1: System Architecture .....	26
Figure 3.2: Implementation architecture of Online AMS .....	27
Figure 3.3: Data Schema .....	29
Figure 4.1: Incremental Model .....	34
Figure 1: Interface design of login .....	43
Figure 2: Interface design of Admin Dashboard .....	43
Figure 3: Interface design of Student Dashboard .....	44
Figure 4: Interface design of Teacher Dashboard .....	44
Figure 5: Interface design of Upload Assignment .....	45
Figure 6: Interface design of Allocated Assignment .....	45
Figure 7: Interface design of Add Student .....	46
Figure 8: Interface design of Add Teacher .....	46

Figure 9: Interface design of Class .....	47
Figure 10: Interface design of Subject .....	47
Figure 11: Interface design of Upload Notes .....	48
Figure 12: Interface design of Course .....	48
Figure 13: Database connection .....	49
Figure 14: Create assignment .....	50
Figure 15: Submit assignment .....	52
Figure 16: View assignment .....	55



## LIST OF TABLES

Table 2.1: Details of expenditure .....	10
Table 2.2: Details of Schedule task 1 .....	12
Table 2.3: Details of Schedule task 2 .....	12
Table 2.4: Details of Schedule task 3 .....	13
Table 2.5: Details of Schedule .....	14
Table 2.6: UC Template - 01 .....	15
Table 2.7: UC Template - 02 .....	16
Table 2.8: UC Template - 03 .....	16
Table 2.9: UC Template - 04 .....	16
Table 2.10: UC Template - 05 .....	16
Table 2.11: UC Template - 06 .....	17
Table 2.12: UC Template - 07 .....	17
Table 2.13: UC Template - 08 .....	17
Table 2.14: UC Template - 09 .....	17
Table 2.15: UC Template - 10 .....	18
Table 2.16: UC Template - 11 .....	18
Table 2.17: UC Template - 12 .....	18
Table 3.1: Data dictionary of assignment question .....	29
Table 3.2: Data dictionary of class group .....	29
Table 3.3: Data dictionary of course .....	29
Table 3.4: Data dictionary of note .....	30
Table 3.5: Data dictionary of notification .....	30
Table 3.6: Data dictionary of client details .....	31
Table 3.7: Data dictionary of role .....	31
Table 3.8: Data dictionary of subject .....	31
Table 3.9: Data dictionary of Teacher .....	32

Table 3.10: Data dictionary of user role .....	32
Table 3.11: Data dictionary of user table .....	32
Table 3.12: Data dictionary of assignment allocation .....	33
Table 3.13: Data dictionary of assignment .....	33
Table 4.1: Unit Testing .....	36
Table 4.2: Integration Testing .....	39
Table 4.3: System Testing .....	40

## **LIST OF ABBREVIATION**

AMS    Assignment Management System

CSS    Cascading Style Sheet

DFD    Data Flow Diagram

ER    Entity Relationship

UC    Use Case

UI    User Interface

UML    Unified Modeling Language

# **CHAPTER 1: INTRODUCTION**

## **1.1 Introduction**

Assignment management is one of the fundamental activities in education. College courses include some form of assignment that students complete individually or as a group and submit for marking and feedback. Assignments are a necessary and usual component of learning at all levels of education. They are intended to promote the assimilation and application of principles and procedures presented in class. Assignments can engage students at different cognitive levels of learning, from simple knowledge recognition up to synthesis and evaluation. Along with the development of web technologies web-based systems have been implemented in the field of education. In an Online assignment management system, users can handle their management in an extremely convenient way, user can access the systems from anywhere at any time, get responses and make use of countless online resources and share their own with others. Assignment management involves the allocation of assignments by the teacher to the student, submission of the assignment by students, providing suitable grades to students according to the submission, collection of results, and student information.

Therefore in order to effectively manage the assignment process, there is a need for an automated system, hence the need for an online assignment submission system to facilitate the allocation, collection, and marking of the assignments scripts.

This Online AMS is a web-based application for submitting assignments or projects online without submitting any physical files. In the manual system, students should submit an assignment in form of a hard copy at a fixed due date to the teacher. This process is costly and takes more time. The teacher has to collect all assignments and give grades after checking them. The student's assignment may be lost or damaged before or after submission. This paper will discuss a prototype that will attempt to help in overcoming these issues. This system includes various activities like assignment creation, subject-wise assignments, and collection, reviewing and assigning suitable grades or marks, providing notes, teaching process of a teacher, teacher maintenance, class maintenance, student maintenance returning the assignments to the students with proper feedback. Admin monitor the whole system process. The system has three modules. They are: -

- i) Admin
- ii) Teacher
- iii) Student

## **1.2 Background of the study**

Our project group members performed certain enquiries and research on the online assignment system. Aadikavi Bhanubhakta Campus had used

- i. Google classroom
- ii. Microsoft Office
- iii. Microsoft teams

To provide assignments and take classes online. After using sometime this online assignment system has disappeared for a long time. Students start to join classes physically and start to submit assignments manually.

### **Manual to Email:**

This stage was the initial transition from traditional assignment submission to an optional system allowing distance students to submit assignments as attachments to email messages.

- i. Experience  
The traditional system suffered from a number of problems that increased workload and increased turnaround time.
- ii. Technology  
Internet service providers were not common and most students relied on long-distance phone calls to College modems. The primitive nature of email programs and the huge variety in encoding styles for attachments made submitting computer files are somewhat problematic.
- iii. Individual  
Both students and teachers in this course had significantly greater familiarity and access to computer technology than the general.

### **Email to Google Classroom:**

Like many others the experience gained using online assignment submission by email attachment was at best difficult. Consequently, move to Google Classroom which is based on automated email supplemented with a web page that reported progress.

- i. Experience  
Some of the problems experienced with manual email submission were students using the wrong address, duplication of administration effort, and significant administrative burden for the lecturers who had to confirm receipt by email.

ii. Technology

Internet tools were becoming available and interest in the Web was becoming more prevalent. The combination of increasing numbers and increasing Internet access led to students using the system.

iii. Individual

The course Operating Systems is a prerequisite for System Administration. The course coordinator now had increased experience with the online assignment system.

**Google Classroom to Microsoft Office:**

The use of Google classroom with automation was good. With the increasing presence of the Web, the online assignment management system moved toward entirely onto the Web. The system was not integrated with any organizational system so students used just their student number for assignment submission.

i. Experience

The form of automation still left too much freedom for students to make mistakes like using the wrong subject and message structure and attaching files in the wrong format.

ii. Technology

This is moving into the stage of online assignment where the submission system online became easier where students could upload assignments in the form of pdf, files, and pictures.

iii. Individual

Staff and students are becoming more familiar with the Web and related technology. Using internet explorer submission could be done within a fixed deadline.

**Microsoft Office to Microsoft Teams:**

The student used online submissions as well as joining classes. This integration enabled the system to know which courses a student was enrolled in and also to make use of a global username/password available to all students.

i. Experience

During the pandemic time of covid-19 Microsoft Teams was used to take classes and submit assignments within a deadline. As well, students involved in online exams. By this stage, the basic structure of the system was in place. On-going development has involved the creation of additional services to the existing system.

ii. Technology

The Web had arrived as a major tool for students to submit assignments online as well as participated in online classes.

iii. Individual

The staff makes innovative use of technology to aid their teaching and learning. Both students and teachers share a suitable platform to engage in the class.

### **1.3 Problem Statement and Motivation**

Our members of project group performed certain enquiry and research that how organization can give and take assignments from student systematically in computer basis and manage assignment's record of the students. Our application is able to manage assignments and provides huge convenience to the authority as many processes are automated and work through online. On the other hand, managing accounts and maintaining proper records of all employees' time and attendance to prepare. According to the pervious online assignment system, user faced difficulty to use features in those systems. They have to take some lessons or read instructions properly before using it. The user feels more comfortable to use manually so they are returning to traditional method. The system is not user friendly. They do not have proper responding features after submitting assignment. If we do not solve the problem, they may go back to manual system.

### **1.4 Objectives**

The aim of this project is to develop an online assignment submission and feedback system for college students as well as for lecturers. Below is listed the primary objective and secondary objective of online assignment system:

#### **1.4.1 Primary Objective**

- To develop an online assignment management system for students as well as for lecturers to submit and collect reports.

#### **1.4.2 Secondary Objective**

- To develop an online platform for assessing student's assignment
- To review assignment with interactive reports
- To provide assignment related feedback

## 1.5 Project Scope and Direction

The main intention of this project is to solve the issues encountered in the manual assignment which consumes more time and waste of papers. The system will provide benefits to students and lecturers and will allow students to submit their assignments any time and from anywhere, hence the proposed system is to develop an automated online platform where the student assignment can be managed effectively and efficiently. There is no need of the presences of lecturer as the student is submitting the assignment online.

- It is an online web-based application.
- The project makes easy and comfortable interaction between Teachers and Students.
- The appropriate suggestion and feedbacks can be delivered fast and effectively to students.

## 1.6 Limitations

- Online Learning Requires Additional Training for Instructors
- Online Classes Are Prone to Technical Issues
- Requires you to be an active learner
- It gives you more freedom perhaps more than you handle
- Online courses don't have an instructor to hound you to stay on task

## 1.7 Report Organization

**Chapter 1:** This chapter explains the overview, introduction, problem statement, objectives, scope and limitation of the proposed system.

**Chapter 2:** This chapter covers all the history, methods, requirement specification and feasibility analysis and structured system requirements.

**Chapter 3:** Design of Online Assignment Management System (OAMS) project is explained in detail with all the necessary diagrams and brief functionality.

**Chapter 4:** Process of implementation and testing is described along with all the tools used for the development.

**Chapter 5:** Conclusion and future scope of the application are explained.



## **CHAPTER 2: REQUIREMENT ANALYSIS**

### **2.1 Literature Review**

Traditionally assignment submission involves students physically handing in a paper copy of their teacher which is very time consuming and may not be as secure or efficient as online assignment submission. Additionally, the assignment may be lost or damage before as well as after submitting. The old method that uses paper sheets for submitting assignment is a tedious task. Based on the research, there are many solutions that are available to solve this issue. Submitting assignment through online is the smart way of assignment management system. Online assignment submitting is faster technique than personal submitting in class.

The system we are going to propose is having three sub-systems. The first sub-system is admin where admin gives privileges to the instructor to access a specific class, the second is assignment creation by the instructor and the third is student submission. This system is having three panels. One is from the admin side, the second is from the student side, and another is from the lecturer's side. From the admin's side, the panel will allow adding teacher, student, subject, categorization of classes and students, view submission, assignment, provided notes and grades of a student by an instructor, and feedback. From the student's side, the panel will only allow us to check the feedback, view grades, and submit the assignments. Whereas from the lecturer's side, the system will allow the lecturer to send feedback, upload assignment, and receive the assignment from the students. To be added the system will also allow students to send feedback about the quality of the lecturers who are taking classes which will redirect to administration as well. As the system will be used by all the departments so the storage capacity of the database should be higher than the average one. This will make the system more responsive while sending and receiving data from both sides. Traditionally assignment was taken manually which is very time consuming and often leads to human error. Additionally, the assignment may be lost or damage before as well as after submitting. The old method that uses paper sheets for submitting assignment is a tedious task. Based on the research, there are many solutions that are available to solve this issue. Submitting assignment through online is the smart way of assignment management system. Online assignment submitting is faster technique than personal submitting in class.

Many different researches have focused on the subject of an online assignment management system these work can be represented as following:

Sam Hsu (2012) proposed a web based Automated Homework Submission System with three modules namely a User Interface Module, Submission Acceptance Module and Grader interface module. Based on this proposed architecture and recommendations given, the online submission system was modeled. The new system will allow the tutors to upload files, export marks to an excel sheet, students to download files, participate in online assessment and system to send reminders about deadlines to students.[1]

In-house (2007) developed web-based Online Assignment Submission (OAS) system for students and tutors at Wawasan Open University. It was created especially to facilitate students' online assignment submission anywhere, anytime and make it for possible tutors to download students' Tutor-Marked Assignment and upload marked assignments together with online feedback to individual students.[2]

Prof. Mukhtar Ansari et al. (2020) developed Online Assignment System which allows student and faculty to communicate about assignment and queries. After the faculties correct the assignment and return it back, some student does not get their assignment back as anyone can take it or misplace it. To avoid this, they created an online assignment system. The main motive of this project is to create a paperless environment by developing online assignment system. This project also reduces the unnecessary workload of lecturers and student.[3]

Angel System was created for Pennsylvania State University. This system focuses on the assignment management part. All useful features about assignment management of other system are included in this system and new functions that are practical in the management are added. The objective is to provide instructors and students with a convenient, efficient and secure way to handle their assignment activities. [4]

Baha' Aldeen Amayrch (2010) developed online assignment submission for downloading and uploading online assignments for students; with each assignment having information about the instructions, description and deadline and submission details. It offers faster transmission of assignments than using online system. The interface use to invoke different testing program by teachers, so save the time and cost for teachers by enabling them to put up a fast response for students as well as increasing the quality of the feedback provided to students.[5]

Michael Chasen et al. (1997) developed Blackboard to help educators create and manage online course content, assignments and assessments and to communicate with students. Blackboard is a widely used Learning Management System that allows teachers to create and manage assignments, provide feedback and grade student work.[6]

Google Classroom (2014) was developed and is maintained by Google. It is a free, user-friendly and versatile online assignment submission system that allows teachers to create and manage assignments, provide feedback and grade student work. It is an excellent option for educators looking for a simple and effective way to manage online assignments and communication with students.[7]

Overall, an online assignment submission system can help to improve the efficiency and effectiveness of the assignment submission process for both teachers and students.

## **2.2 Problem Definition**

Manual Assignment Management System is a process where a student submit a hard copy of assignment in a fixed due date either in class or at the office. Students have to come to class for just submitting assignment. This process is more time and effort consuming work. In other hand, teacher must preserve a space to arrange all assignments which take large space to store according to every semester and need extra time for organizing assignment according to its group. To solve these issues, we aim to design and implement an online assignment management system where assignments are arranged according to its group and save necessary information into database system. We can get assignment's status through online and submit it from anywhere with internet access.

## **2.3 Proposed Solution**

This method proposed in this paper is to submit assignment through online. In such system, users can handle their management in an extremely convenient way: they can access the systems from anywhere at any time; get response immediately. More importantly, they do not have to worry about operating system and different application software on either server or client side. All we need to have is a browser and ability to get online. This system is a web based system which will help in universities and college to manage and make report of student's assignment without losing or damage.

## **2.4 Requirement Analysis**

In our project, we have collected list of documents with sufficient and necessary requirements for the project development. To derive the requirements, we have done better understanding of the products under development which we achieved through detailed and continuous communications with the project team throughout the software development process.

### 2.4.1. Functional requirement:

➤ **Admin:**

- Admin can administer (add, delete and update) Teacher, Student, course and subjects and oversee students.
- Admin can post and get feedback.
- Admin can view student information and status

➤ **Student:**

- Student can upload the solution of any of his assignment before the deadline
- Student will receive a message from the system after uploading the solution (accept or reject).
- Student gets feedback from the teacher.
- Student can see the information about the assignment (description and instruction, start-time, end-time, how to download and upload the assignment and type of work: individually or as group).

➤ **Teacher:**

- Teacher can add assignment information (description and instruction, start-time, end-time, how to download and upload the assignment and type of work: individually or as group).
- Teacher can see the assignments submitted by students. They can assess the students by browsing a report of the report of assignments and students.
- Teacher can manually send feedback, marks and notes to students.

### 2.4.2. Non-Functional requirement:

- **Security:** The system will be able to prevent illegal or incorrect operations from teachers and students by using certain tools.
- **Understandability:** The system will be easy for students to use and deal with, user friendly by developing good interface.
- **Accessibility:** The system will be available via the internet and can be accessed any time and any place with internet.
- **Performance:** The system will be fast.
- **Reliability:** The defined facility provided by application should be consistent.

## 2.5 Feasibility Study

### 2.5.1 Economic Feasibility

It involves a cost / benefits analysis of the project, helping organizations determine the cost and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility – helping decision - makers determine the positive economic benefits to the organization that the proposed project will provide.

- **Cost estimation:**

Software cost comprises a small percentage of overall computer based system cost. There are a number of factors, which are considered, that can affect the ultimate cost of the software such as – human, technical, hardware and software availability, etc.

S.N	Particulars	Amount
1.	Paper print	Rs.3000/-
2.	Per person	Rs.5000*2 = Rs.10000/-
3.	Internet cost	Rs.4000/-
4.	Miscellaneous expenses	Rs.5000/-
	<b>Total cost</b>	<b>Rs.22000/-</b>

Table 2.1: Details of expenditure

- **Payback period:**

A payback period calculator is a utility tools, shows the time taken to recover the cost of the project or an investment. We can determine the number of years. It takes to recover the cost of the investment. The payback period calculator consists of a formula box, where you enter the initial investment and the periodic cash flow. The payback period will show you the payback period of the investment. Formula:

$$\begin{aligned}\text{Payback period} &= \text{initial investment} / \text{net annual cash inflow} \\ &= 22000 / 30000 = 7 \text{ months}\end{aligned}$$

- **Return on investment:**

Return on investment or ROI shows the return from investments. It helps to choose the best investment across different investment options. In simple terms, the return on investment is a financial ratio that helps to determine the benefit of investment against the costs. We will calculate the return on investment using the formula:

$$\text{ROI} = \text{Net Profit} / \text{Cost of the investment} * 100 = 15000 / 22000 * 100 = 68\%$$

- **AMC (Annual maintenance cost):**

AMC stands for annual maintenance contract. It is also known as annual maintenance charges. It is offered by all manufacture after sales. Since our system is web-based application. Web hosting cost Rs.6000 per annual and AMC cost Rs.6500. The price may go up if new feature is added, regular updates and upgrades may be necessary.

### 2.5.2 Technical Feasibility

Our system is built for any type of operating system. It can adopt the technological upgrades as it is developed under the considerations of software engineering principles. Moreover, it uses object-oriented approach of programming which can enhance the upgrading with new classes and modules as per requirement.

### 2.5.3 Operational Feasibility

This system has a simple UI. Anyone with the basic knowledge of application can easily go to the home page of the application to perform the task and perform actions easily by clicking bottoms. The developed application will be reliable, maintainable, usable, user friendly, sustainable and affordable. The system is tested under several circumstances with varying inputs in unit approach of testing to integrated approach of testing.

### 2.5.4 Schedule Feasibility Study

Schedule Feasibility is defined as the probability of a project to be completed within its scheduled time limits, by a planned due date. If a project has a high probability to be completed on-time, then its schedule feasibility is appraised as high. This assessment is the most important for project success. In scheduling feasibility, an organization estimates how much time the project will take to complete.

S.N.	Activities	Start Date	End Date	Duration(Days)
1.	System analysis	2022/06/11	2022/06/17	7 days
2.	System Design	2022/06/18	2022/07/02	15 days
3.	Coding	2022/07/03	2022/07/27	25 days
4.	Testing	2022/07/28	2022/07/30	3 days
5.	Debugging	2022/07/31	2022/08/01	2 days

Table 2.2: Details of Schedule task 1

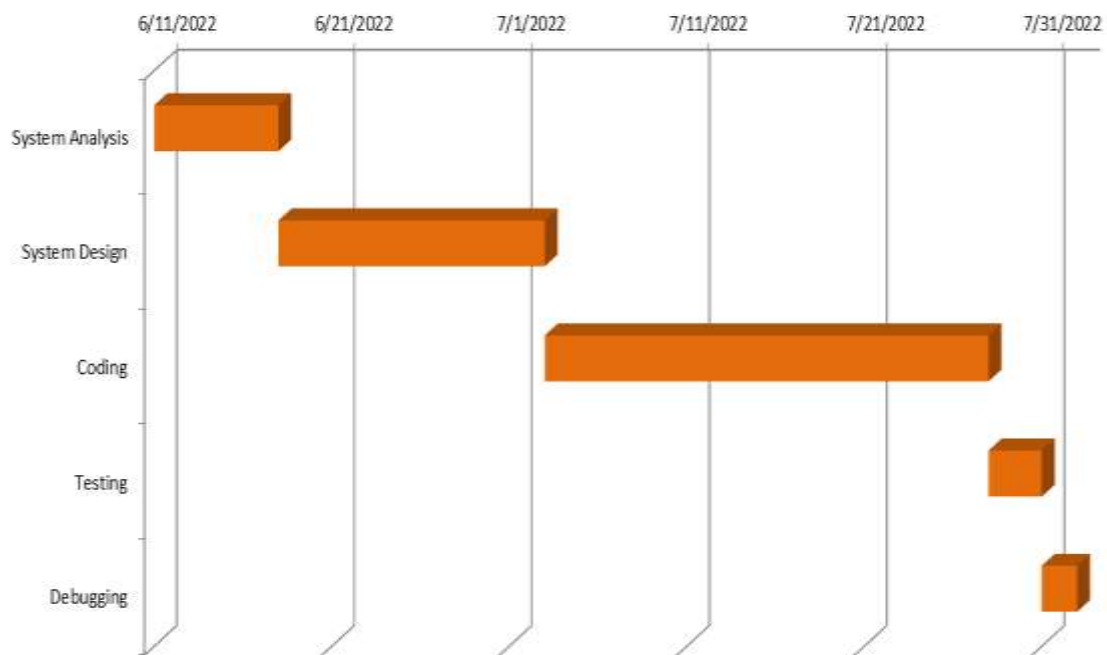


Figure 2.1: Gantt chart for task 1

The diagram explains the schedule of the project where system analysis and design were completed in 7 days and 15 days respectively. Code took 25 days and testing was completed in 3 days.

S.N.	Activities	Start Date	End Date	Duration(Days)
1.	System analysis	2022/08/02	2022/08/08	7 days
2.	System Design	2022/08/09	2022/08/18	10 days
3.	Coding	2022/08/19	2022/09/07	20 days
4.	Testing	2022/09/08	2022/09/10	3 days
5.	Debugging	2022/09/11	2022/09/12	2 days

Table 2.3: Details of Schedule task 2

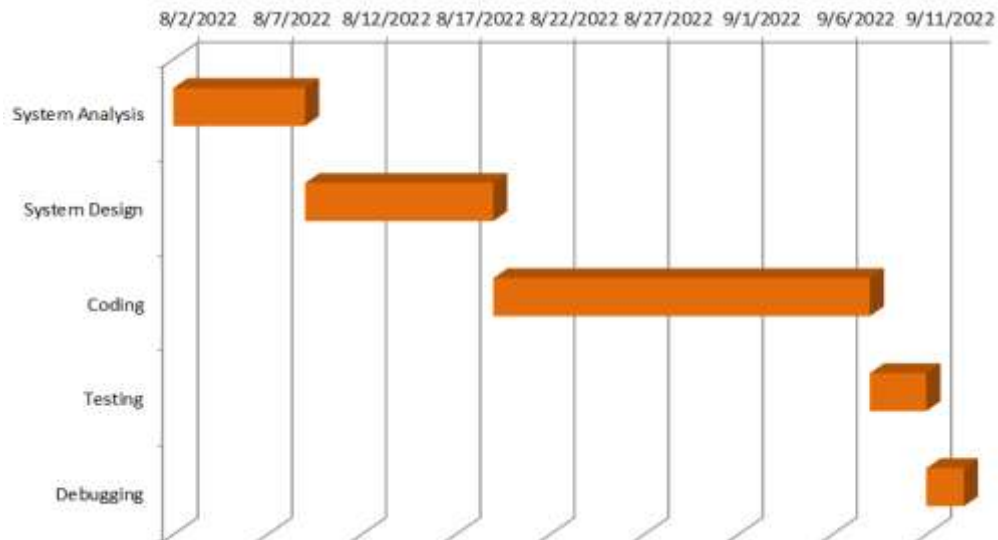


Figure 2.2: Gantt chart task 2

The diagram explains the schedule of the project where system analysis and design were completed in 7 days and 10 days respectively. Code took 20 days and testing was completed in 3 days.

S.N.	Activities	Start Date	End Date	Duration(Days)
1.	System analysis	2022/09/13	2022/09/19	7 days
2.	System Design	2022/09/20	2022/09/26	7 days
3.	Coding	2022/09/27	2022/10/11	15 days
4.	Testing	2022/10/12	2022/10/14	3 days
5.	Debugging	2022/10/15	2022/10/16	2 days

Table 2.4: Details of Schedule task 3

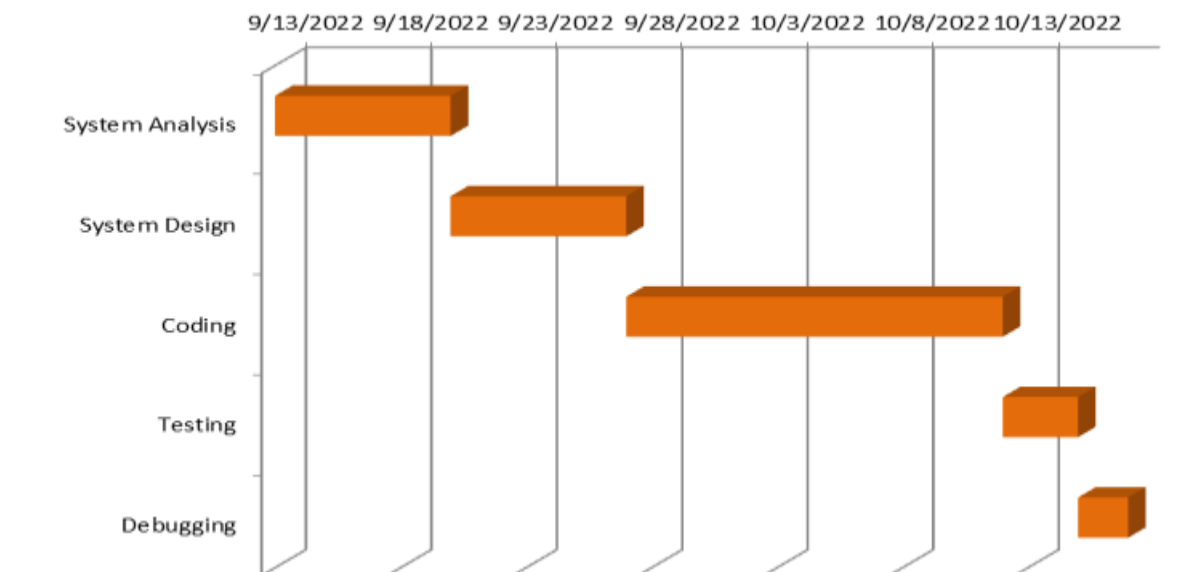


Figure 2.3: Gantt chart task 3



The diagram explains the schedule of the project where system analysis and design were completed in 7 days and 7 days respectively. Code took 15 days and testing was completed in 3 days.

S.N.	Activities	Start Date	End Date	Duration(Days)
1.	Requirements gathering and feasibility study	2022/05/17	2022/06/10	25 days
2.	Task 1	2022/06/11	2022/08/01	52 days
3.	Task 2	2022/08/02	2022/09/12	42 days
4.	Task 3	2022/09/13	2022/10/16	34 days
5.	Documentation	2022/10/17	2022/10/19	156 days

Table 2.5: Details of Schedule

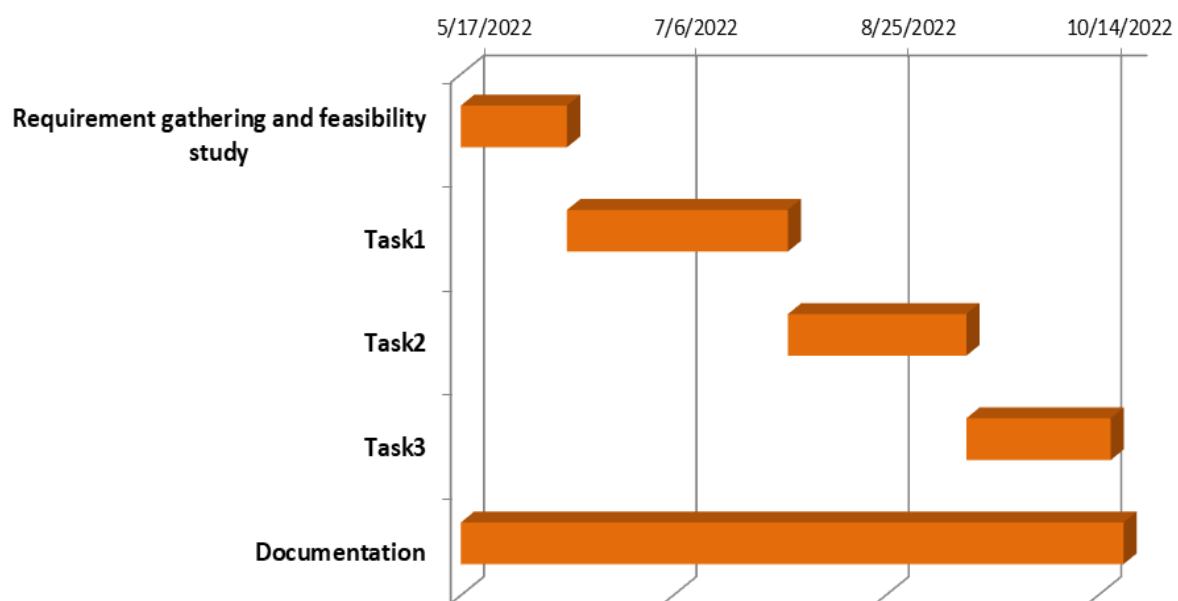


Figure 2.4: Gantt chart for schedule

The diagram explains the schedule of the project where the requirement gathering and feasibility study was completed in 25days. Then, Task 1 was completed in around 52 days collectively. Task 2 was completed in 42 days respectively. Task 3 was completed in 34 days. Side by side, process of documentation was also carried out until the completion of the project.

## 2.6 Structuring System Requirements

Structuring system requirements concentrates on the definition, structure and relationships within data. The characteristics of data captured during data modeling are crucial in the design of databases, programs, computer screens and printed information is essential in ensuring data integrity in an information system.

### 2.6.1 Use Case Diagram

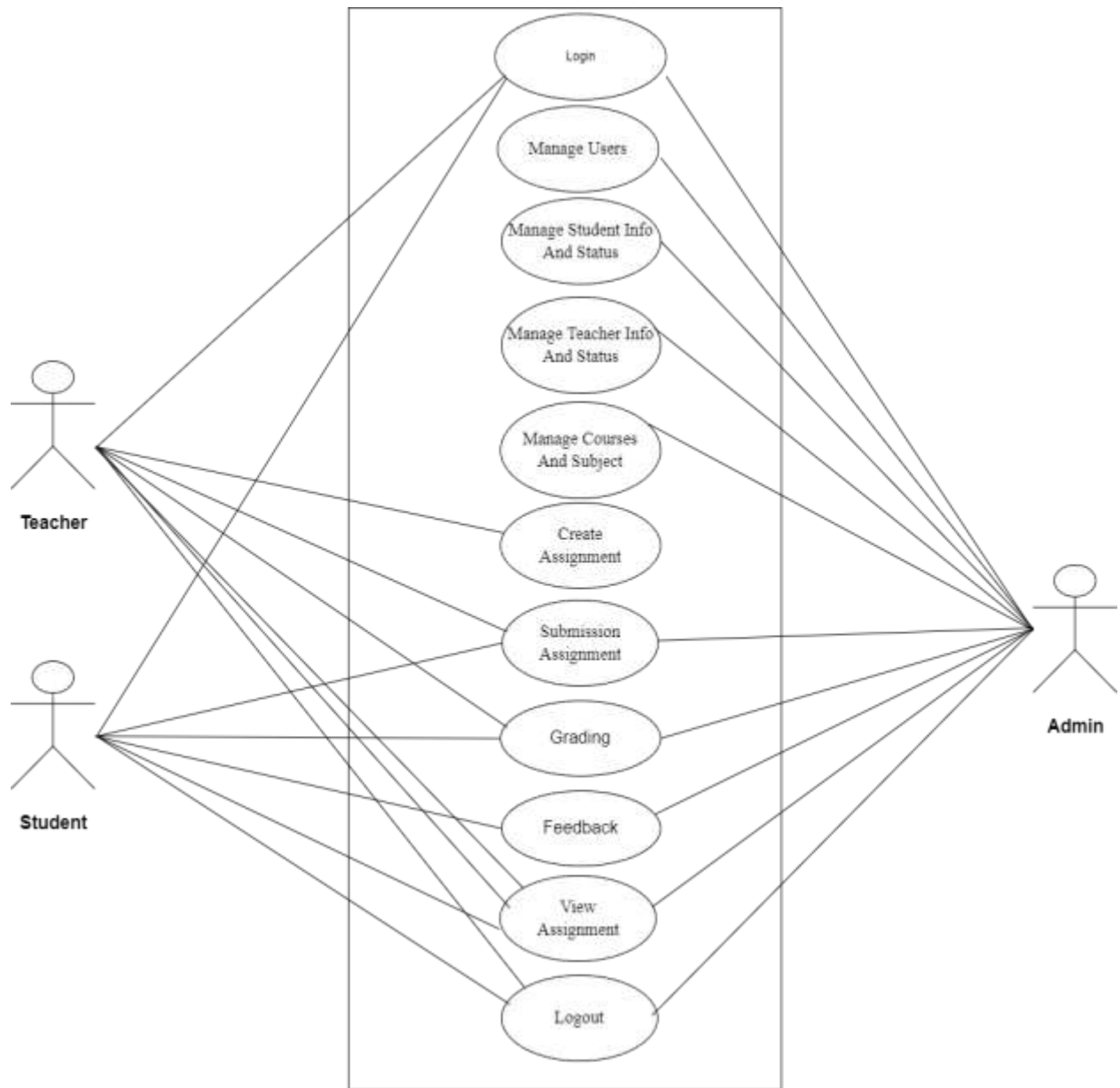


Figure 2.5: Use case diagram

Use Case No.	UC-01
Name	Login
Priority	To login system users
Actors	Admin, Teacher, Student
Pre –condition	Valid username and password must be entered.
Main scenario	User is logged in based on their account type..
Post condition	Redirect to dashboard if username and password is valid.
Related use case	-

Table 2.6: UC Template – 01

Use Case No.	UC-02
Name	Manage Users
Priority	To add, delete and update system users
Actors	Admin
Pre –condition	Admin must be logged in. All fields of user form should be filled and email must be valid.
Main scenario	User is created, deleted and updated in this case by a system administration.
Post condition	Response should be provided to admin whether the user is created, deleted and updated.
Related use case	Add user

Table 2.7:UC Template - 02

Use Case No.	UC-03
Name	Manage Student Info and status
Priority	To add, delete and update student
Actors	Admin
Pre –condition	Admin must be logged in. All fields of student form should be filled and email must be valid.
Main scenario	Student is inserted, deleted and updated in this case by a system administration.
Post condition	Response should be provided to admin whether the student is inserted, deleted and updated.
Related use case	Add student

Table 2.8: UC Template – 03

Use Case No.	UC-04
Name	Manage Teacher Info and status
Priority	To add, delete and update Teacher
Actors	Admin
Pre –condition	Admin must be logged in. All fields of Teacher form should be filled and email must be valid.
Main scenario	Teacher is inserted, deleted and updated in this case by a system administration.
Post condition	Response should be provided to admin whether the Teacher is inserted, deleted and updated.
Related use case	Add Teacher

Table 2.9: UC Template – 04

Use Case No.	UC-05
Name	Manage courses and subjects
Priority	To add, delete and update courses and subjects
Actors	Admin
Pre –condition	Admin must be logged in. All fields of form should be filled.
Main scenario	Course and subjects are inserted, deleted and updated in this case by a system administration.
Post condition	Response should be provided to admin whether the courses and subjects are inserted, deleted and updated.
Related use case	Add Courses and Subjects

Table 2.10: UC Template - 05

Use Case No.	UC-06
Name	Create Assignment
Priority	To add, delete and update assignment
Actors	Teacher
Pre –condition	Teacher must be logged in. All fields of assignment form should be filled.
Main scenario	Assignment is created, deleted and updated in this case by a teacher.
Post condition	Response should be provided to teacher whether the new assignment is created, deleted and updated.
Related use case	Add new assignment

Table 2.11: UC Template – 06

Use Case No.	UC-07
Name	Submission Assignment
Priority	To submit assignment
Actors	Teacher, Student, Admin
Pre –condition	Student must be logged in.
Main scenario	Assignment is submitted by Student and viewed by Teacher and Admin.
Post condition	Response should be provided to Student whether the assignment is submitted or not.
Related use case	Submit assignment

Table 2.12: UC Template – 07

Use Case No.	UC-08
Name	Give Grading
Priority	To give grade in assignment
Actors	Teacher, Admin
Pre –condition	Teacher must be logged in.
Main scenario	Submitted Assignment is graded by Teacher and viewed by Admin
Post condition	Response should be provided to Teacher whether the assignment is evaluated or not.
Related use case	Grading assignment

Table 2.13: UC Template - 08

Use Case No.	UC-09
Name	Feedback
Priority	To give feedback
Actors	Teacher, Admin, Student
Pre –condition	Student must be logged in.
Main scenario	Student gives feedback.
Post condition	Response should be provided to Student whether the feedback is posted or not.
Related use case	Manage Assignment

Table 2.14: UC Template – 09

Use Case No.	UC-10
Name	View Assignment
Priority	To view new Assignment
Actors	Teacher, Admin, Student
Pre –condition	Student must be logged in.
Main scenario	Assignment is created by Teacher and student can view new assignment.
Post condition	Response should be provided to Teacher whether assignment is posted or not.
Related use case	Create new assignment

Table 2.15: UC Template - 10

Use Case No.	UC-11
Name	View Grading
Priority	To view grade of assignment
Actors	Student
Pre –condition	Student must be logged in.
Main scenario	Assignment is evaluated by Teacher and student can view grade of assignment.
Post condition	Response should be provided to Student whether assignment is graded or not.
Related use case	Grading assignment

Table 2.16: UC Template - 11

Use Case No.	UC-12
Name	Logout
Priority	To logout system users
Actors	Admin, Teacher, Student
Pre –condition	-
Main scenario	User is log out on their account type.
Post condition	-
Related use case	-

Table 2.17: UC Template – 12

## 2.6.2 ER Diagram

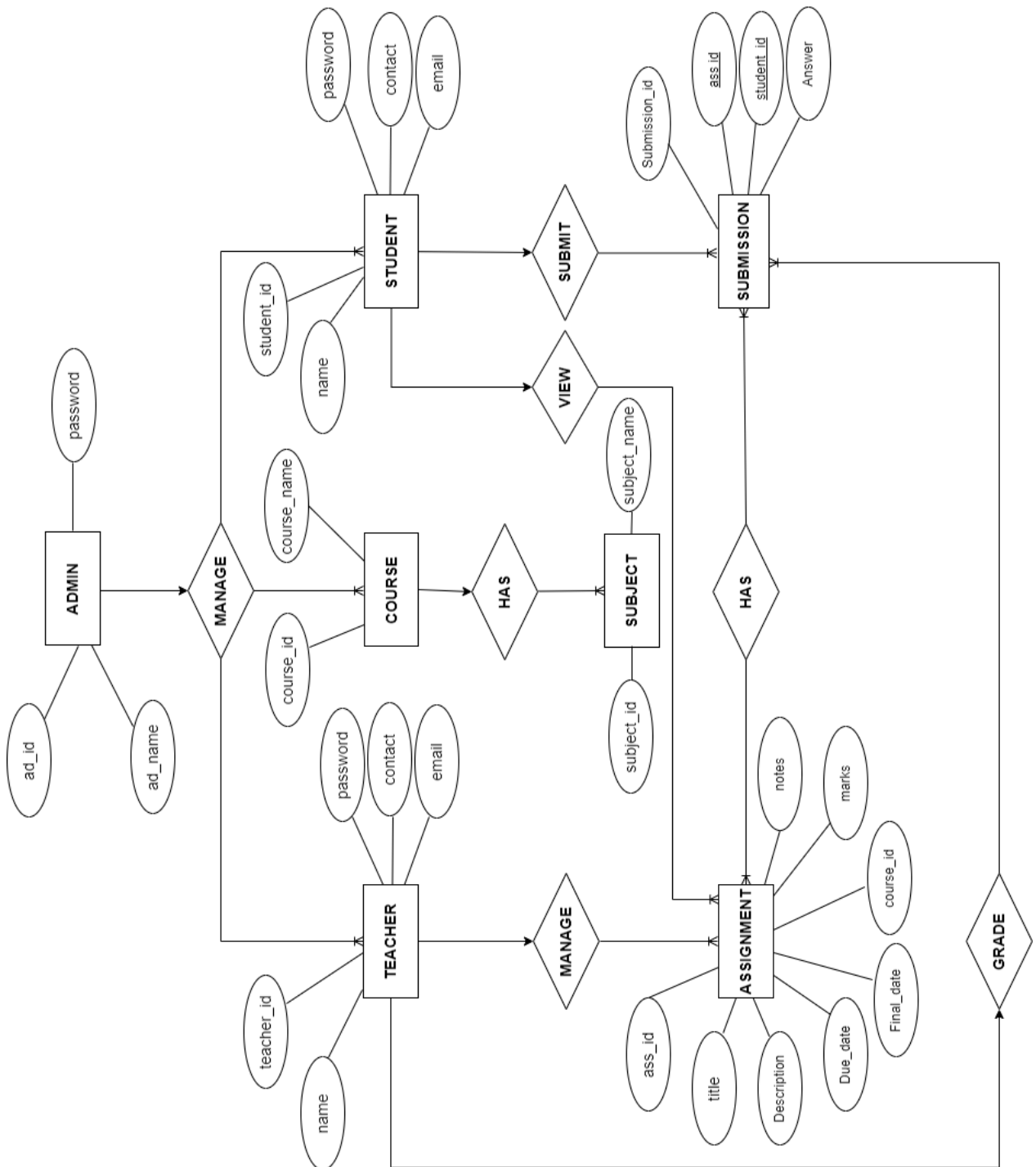


Figure 2.6: Entity Relationship Diagram

### 2.6.3 Data Flow Diagram

Process Modeling (DFD level – 0 and DFD level – 1)

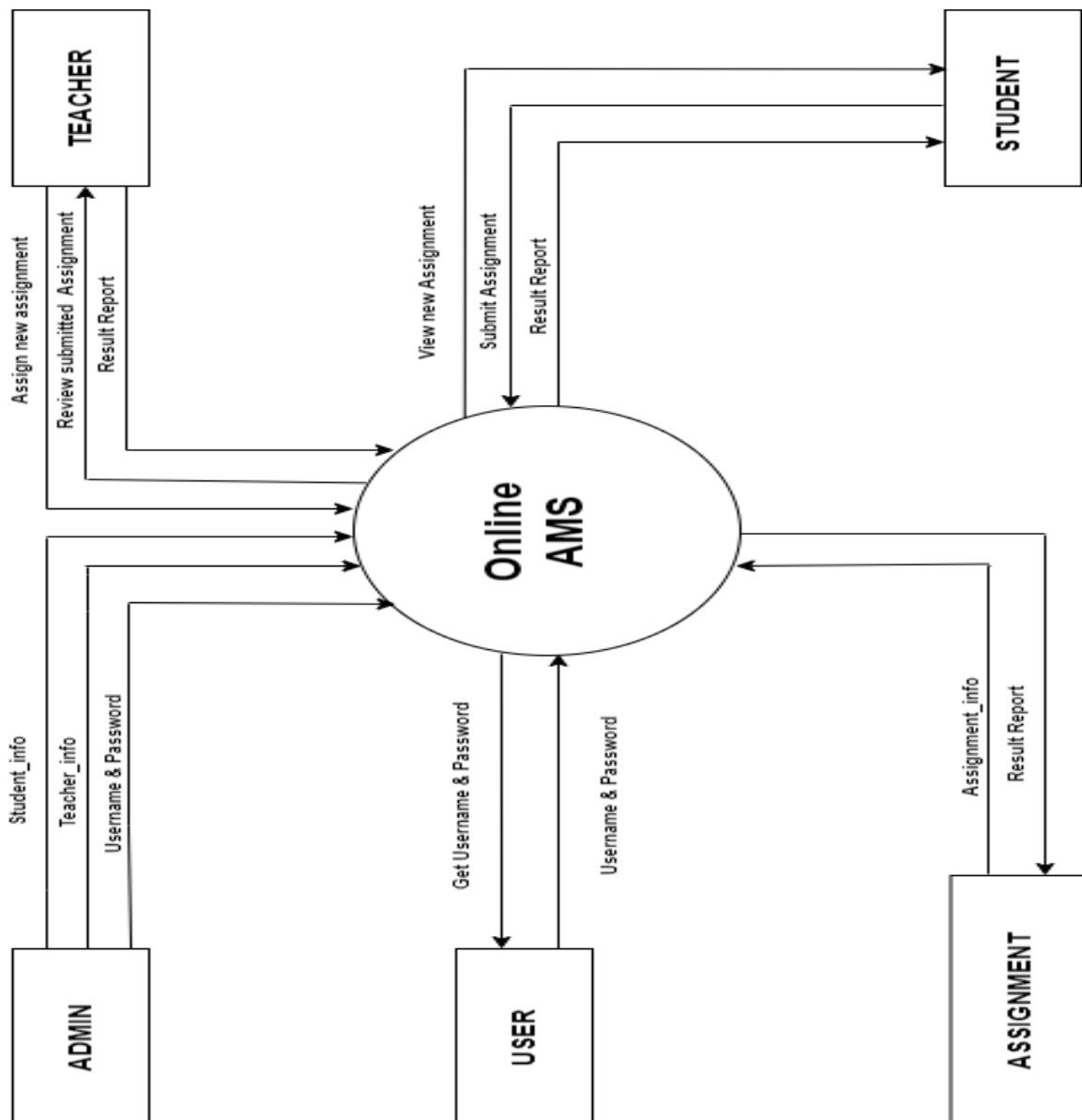


Figure 2.7: Data Flow Diagram Level – 0

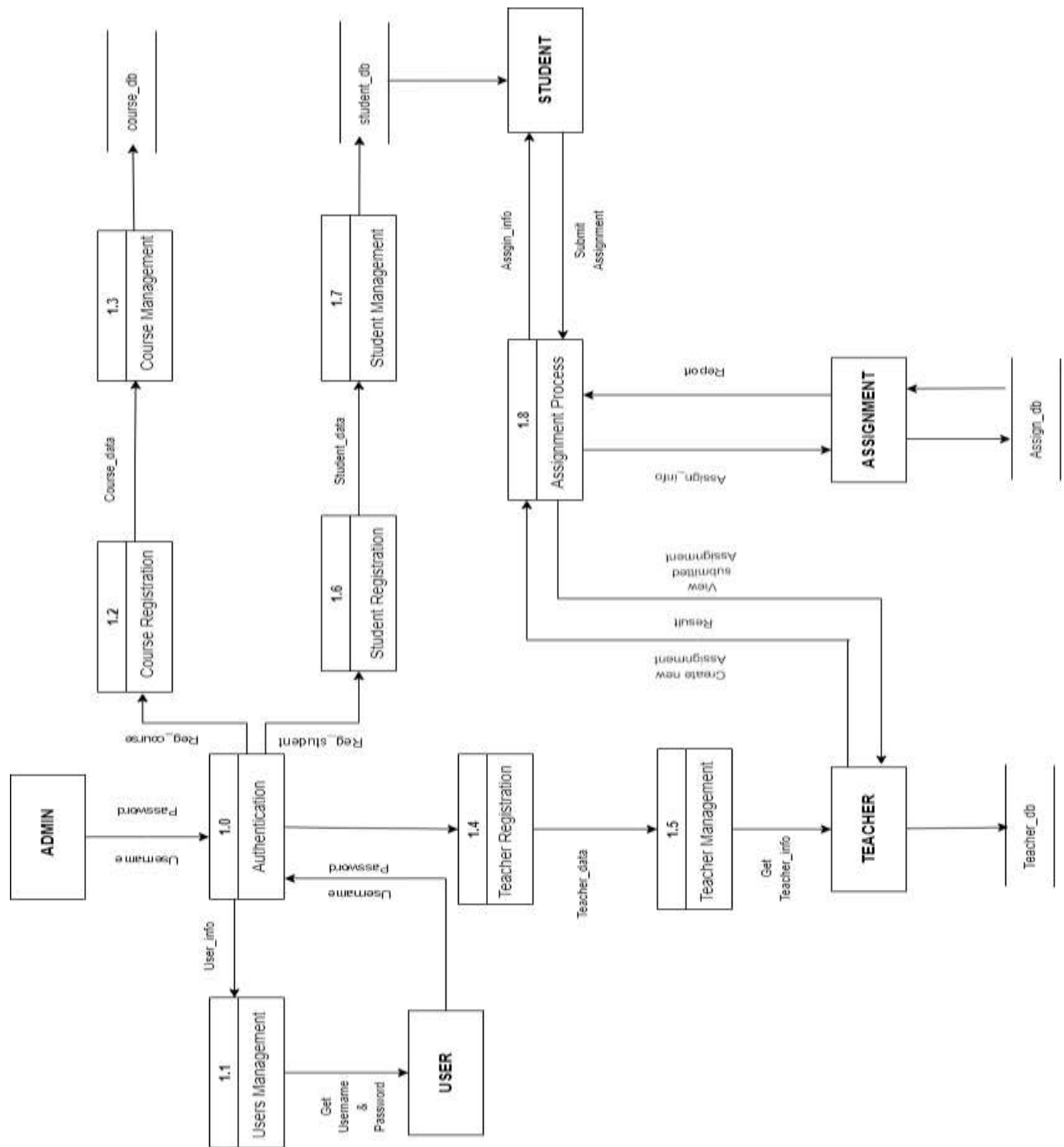


Figure 2.8: Data Flow Diagram Level – 1



## 2.6.4 UML Class Diagram

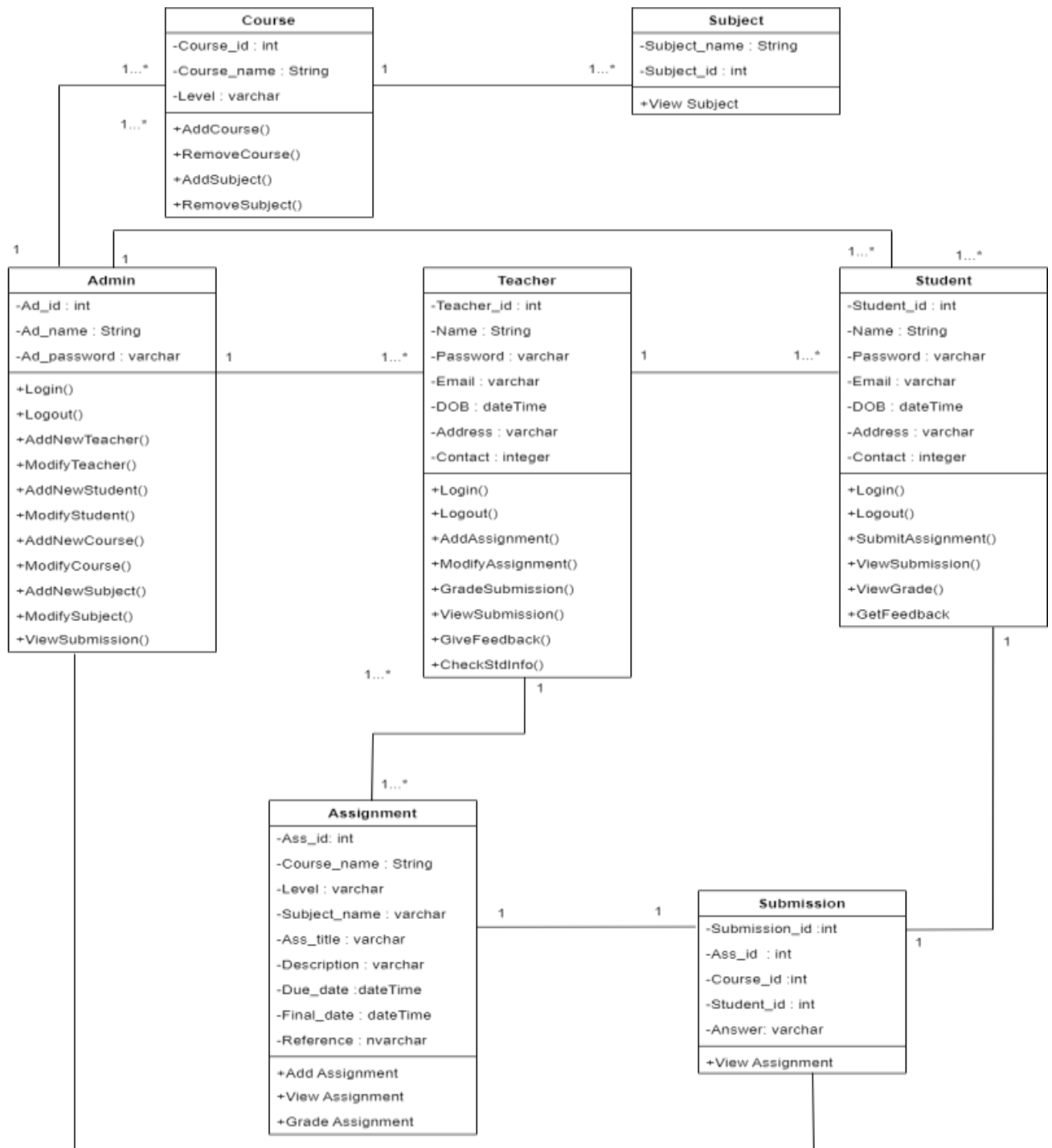


Figure 2.9: UML class Diagram

## 2.6.5 Sequence Diagram

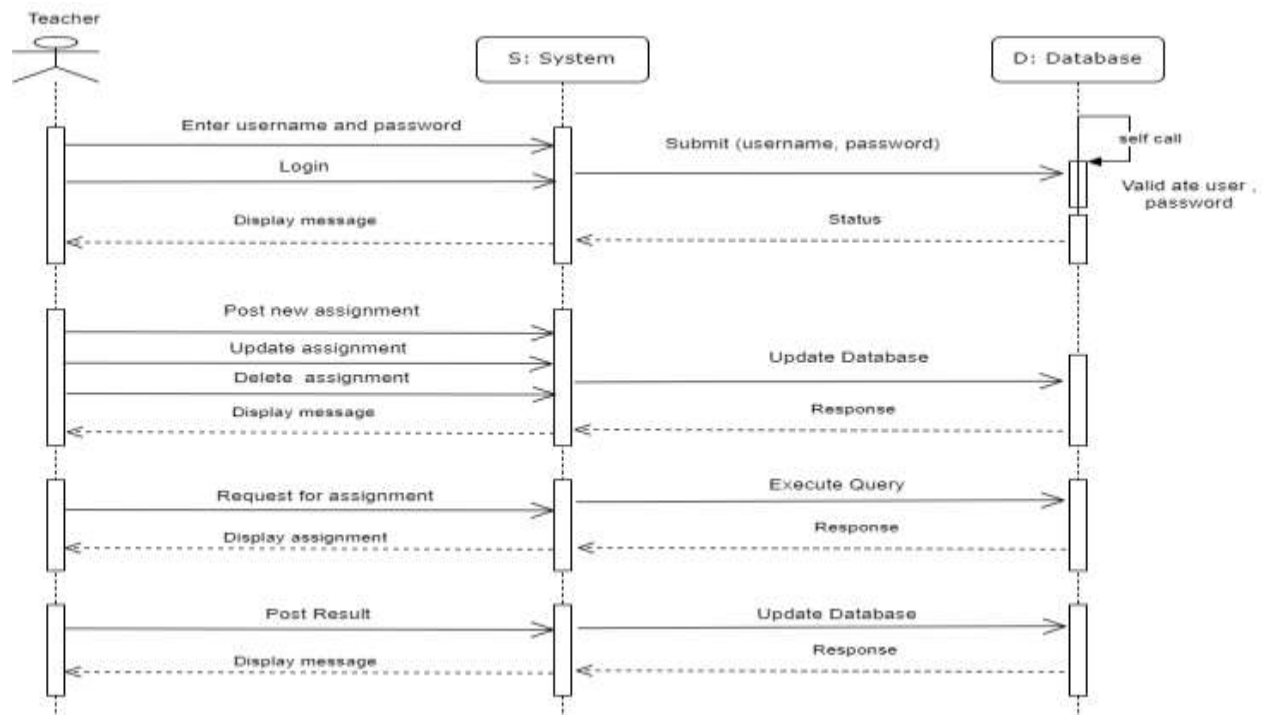


Figure 2.10: Sequence Diagram of Teacher

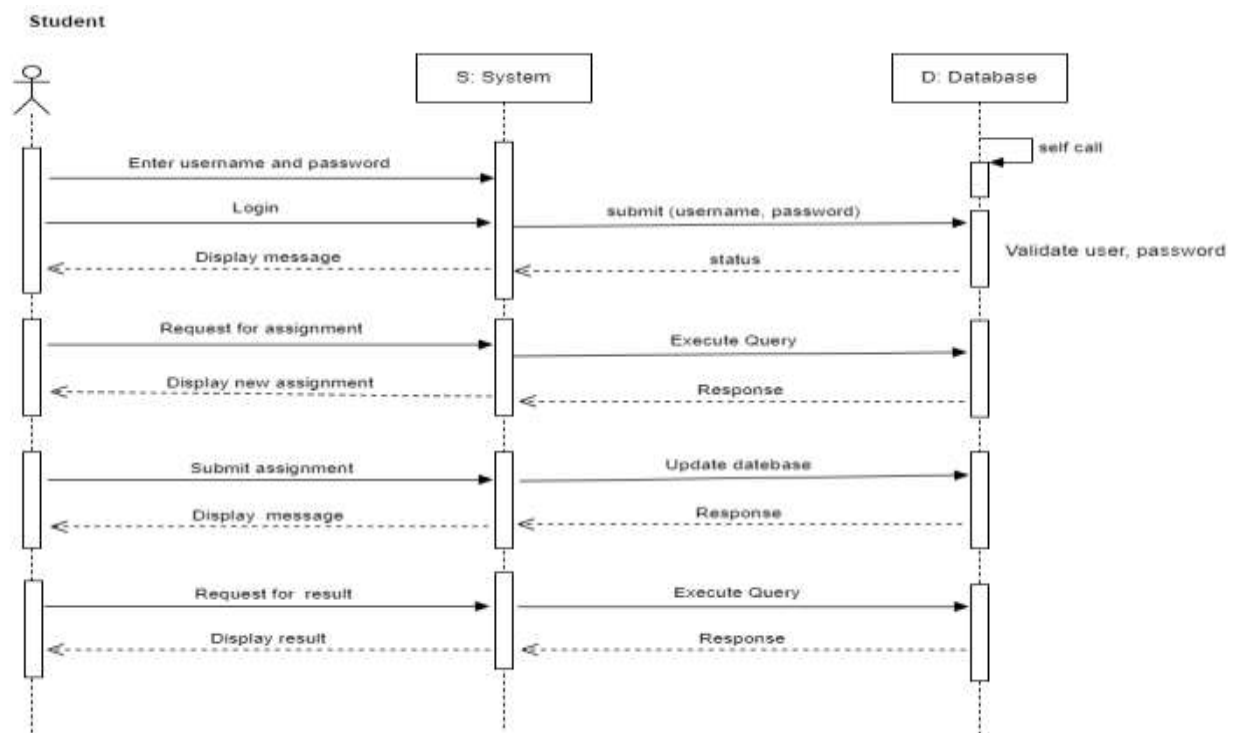


Figure 2.11: Sequence of Student

## 2.6.6 Activity Diagram

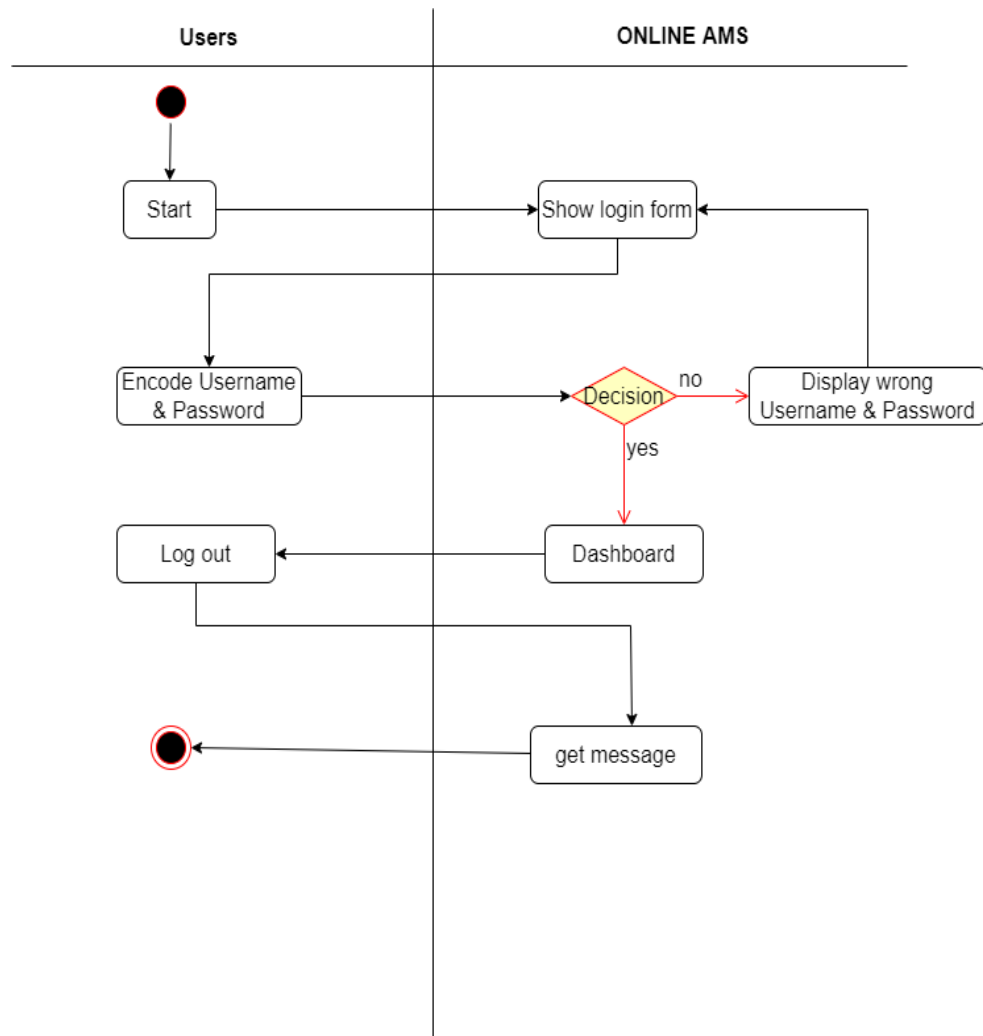


Figure 2.12: Activity Diagram of Users

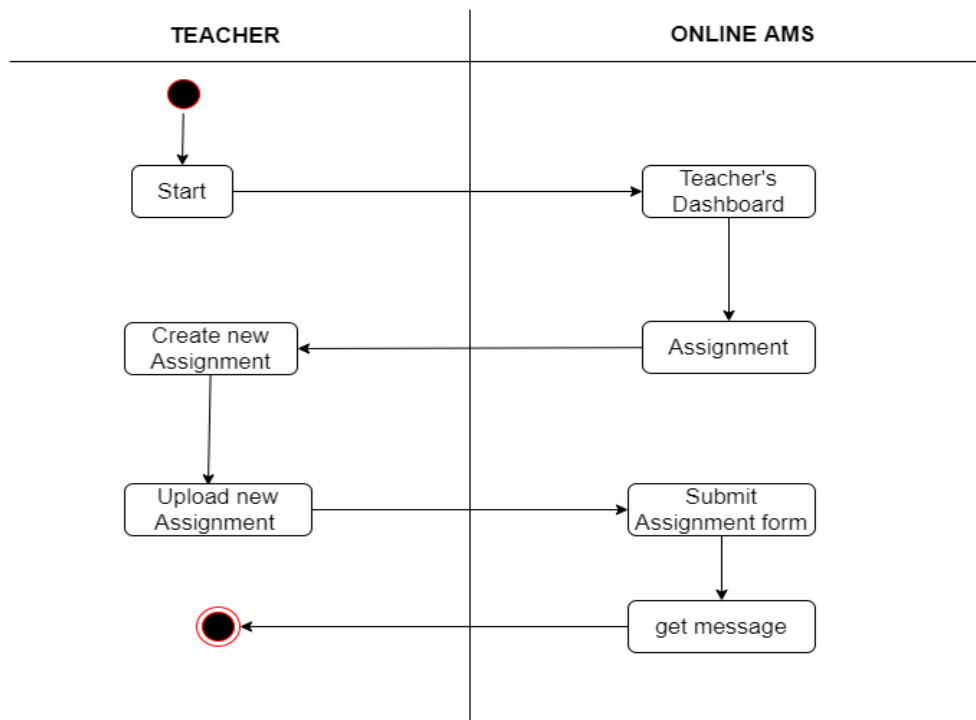


Figure 2.13: Activity Diagram of Teacher

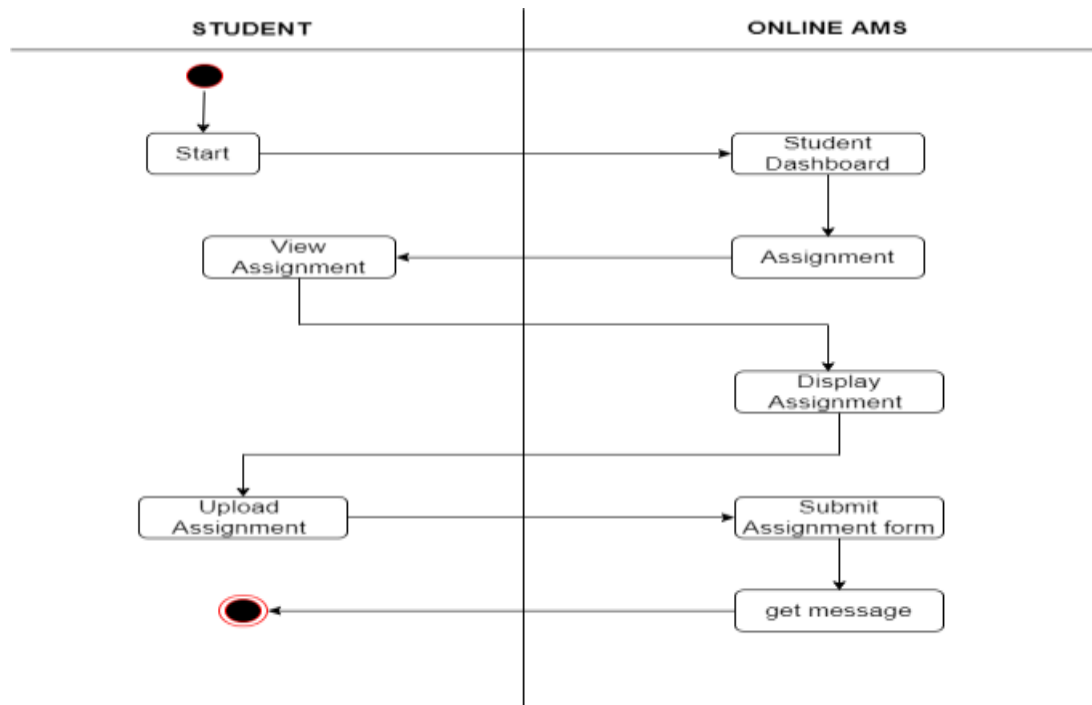


Figure 2.14: Activity Diagram of Student

## CHAPTER 3: SYSTEM DESIGN

### 3.1 System architecture and overview:

System Architecture refers to the overall design and structure of a system, including the hardware, software and network components that make up the system, as well as the relationships and interactions between them. It is a high-level view of a system that describes how the components work together to achieve the system's goals. The architecture provides a blueprint for the system, allowing different teams to understand and work on different parts of the system without interfacing with each other.

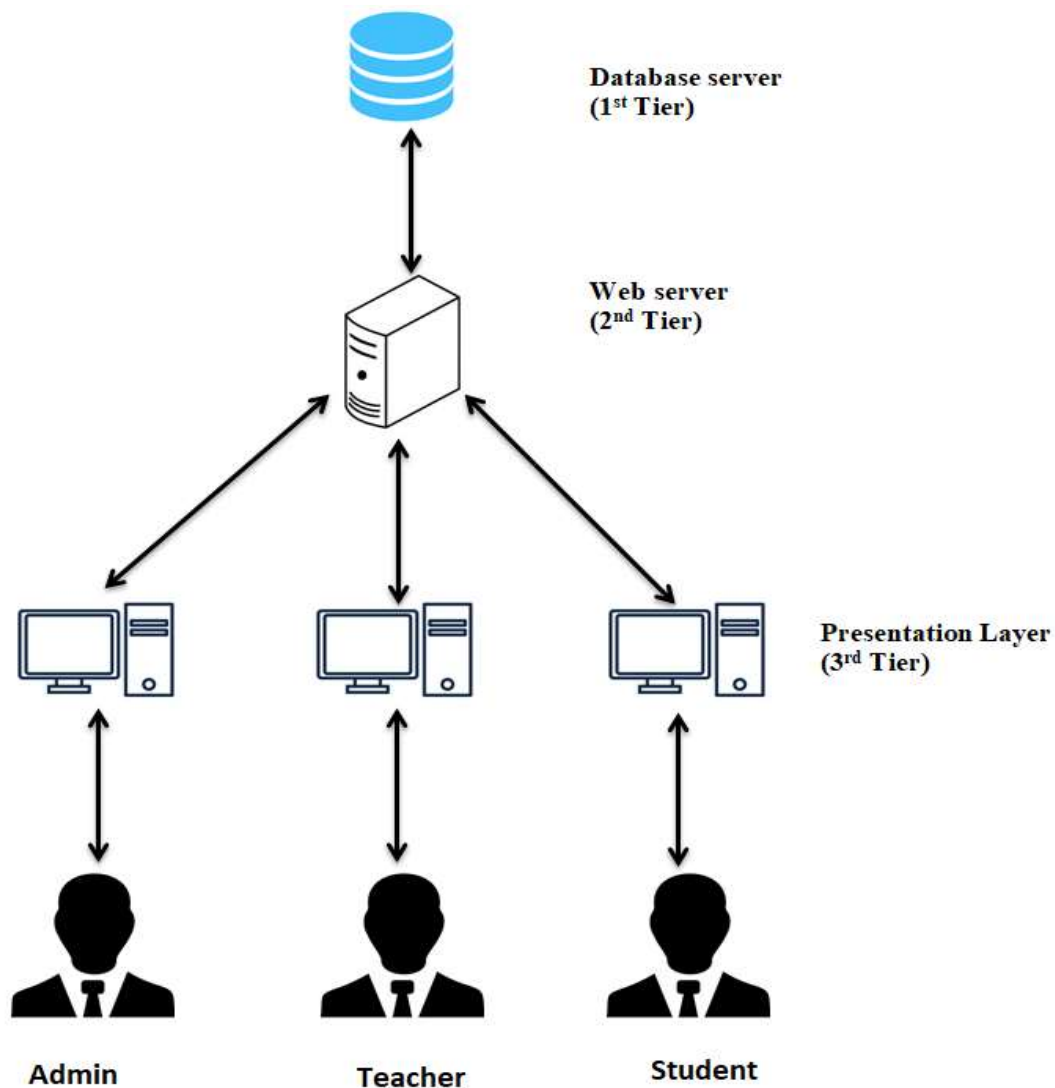


Figure 3.1: System Architecture

### 3.2 System Design

System Design is the process of designing the elements of a system and how they interact to satisfy the requirements of the system. It involves the creation of a detailed model or blueprint of the system, including the physical and logical components and their interfaces. The system design includes the selection of hardware, software and network components, as well as the overall organization and structure of the system. It also includes the definition of interfaces, protocols and standards that are used to connect the system components.

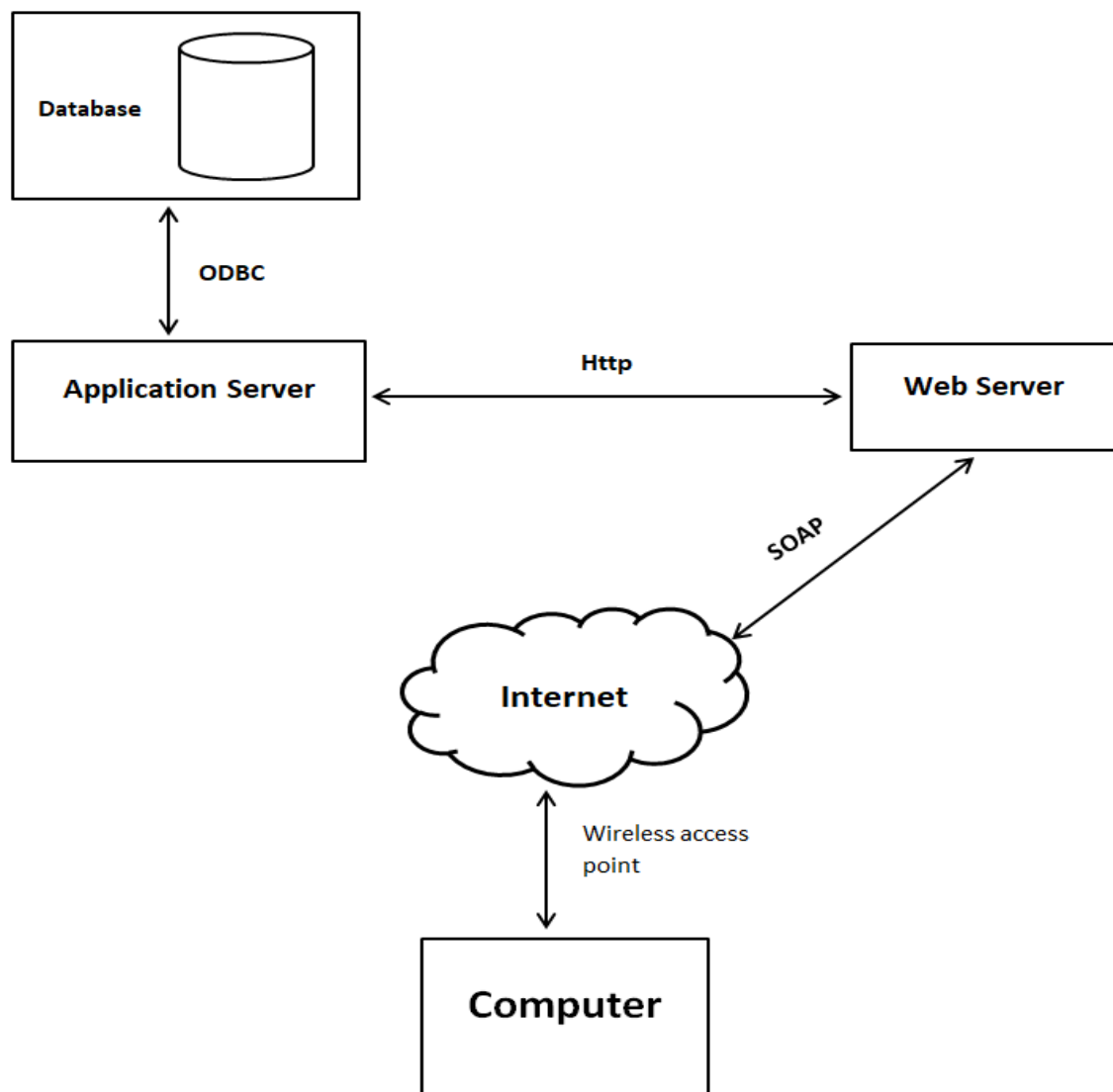


Figure 3.2: Implementation architecture of Online AMS

### 3.2.1. Database Schema

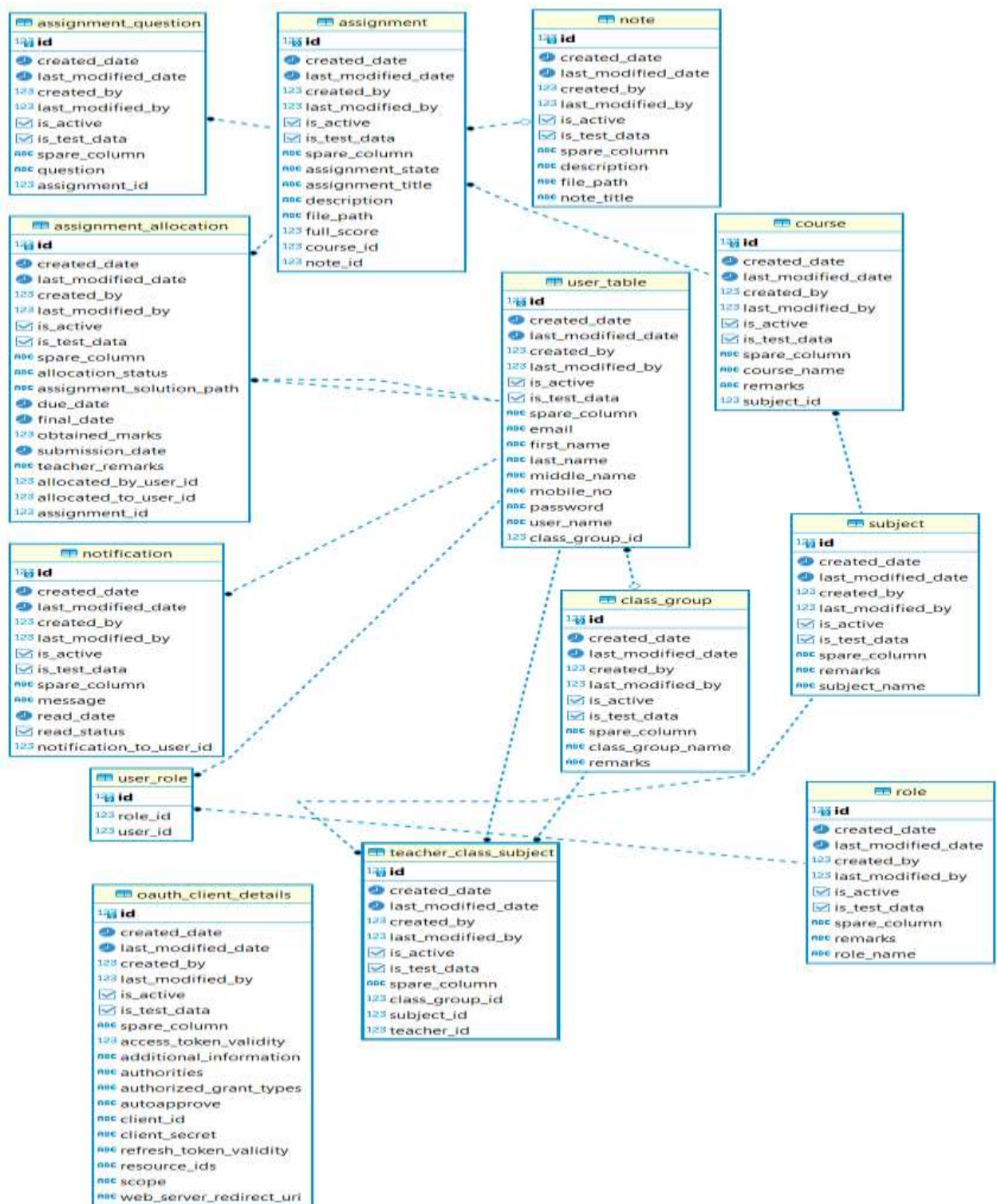


Figure 3.3: Data Schema

### 3.3 Data Dictionary

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
id	1	int4			[v]		
created_date	2	timestamp			[ ]		
last_modified_date	3	timestamp			[ ]		
created_by	4	int4			[ ]		
last_modified_by	5	int4			[ ]		
<input checked="" type="checkbox"/> is_active	6	bool			[ ]		
<input checked="" type="checkbox"/> is_test_data	6	bool			[ ]	false	
spare_column	8	varchar(255)		<a href="#">default</a>	[ ]		
question	9	varchar(255)		<a href="#">default</a>	[v]		
assignment_id	10	int4			[v]		

Table 3.1: Data dictionary of assignment question

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
id	1	int4			[v]		
created_date	2	timestamp			[ ]		
last_modified_date	3	timestamp			[ ]		
created_by	4	int4			[ ]		
last_modified_by	5	int4			[ ]		
<input checked="" type="checkbox"/> is_active	6	bool			[ ]		
<input checked="" type="checkbox"/> is_test_data	7	bool			[ ]	false	
spare_column	8	varchar(255)		<a href="#">default</a>	[ ]		
class_group_name	9	varchar(20)		<a href="#">default</a>	[v]		
remarks	10	varchar(100)		<a href="#">default</a>	[ ]		

Table 3.2: Data dictionary of class group

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
id	1	int4			[v]		
created_date	2	timestamp			[ ]		
last_modified_date	3	timestamp			[ ]		
created_by	4	int4			[ ]		
last_modified_by	5	int4			[ ]		
<input checked="" type="checkbox"/> is_active	6	bool			[ ]		
<input checked="" type="checkbox"/> is_test_data	7	bool			[ ]	false	
spare_column	8	varchar(255)		<a href="#">default</a>	[ ]		
course_name	9	varchar(20)		<a href="#">default</a>	[v]		
remarks	10	varchar(100)		<a href="#">default</a>	[ ]		
subject_id	11	int4			[v]		

Table 3.3: Data dictionary of course












Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
 id	1	int4			[v]		
 created_date	2	timestamp			[ ]		
 last_modified_date	3	timestamp			[ ]		
 created_by	4	int4			[ ]		
 last_modified_by	5	int4			[ ]		
<input checked="" type="checkbox"/> is_active	6	bool			[ ]		
<input checked="" type="checkbox"/> is_test_data	7	bool			[ ]	false	
 spare_column	8	varchar(255)		<a href="#">default</a>	[ ]		
 description	9	text		<a href="#">default</a>	[v]		
 file_path	10	varchar(255)		<a href="#">default</a>	[v]		
 note_title	11	varchar(255)		<a href="#">default</a>	[v]		

Table 3.4: Data dictionary of note










Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
 id	1	int4			[v]		
 created_date	2	timestamp			[ ]		
 last_modified_date	3	timestamp			[ ]		
 created_by	4	int4			[ ]		
 last_modified_by	5	int4			[ ]		
<input checked="" type="checkbox"/> is_active	6	bool			[ ]		
<input checked="" type="checkbox"/> is_test_data	7	bool			[ ]	false	
 spare_column	8	varchar(255)		<a href="#">default</a>	[ ]		
 message	9	text		<a href="#">default</a>	[v]		
 read_date	10	timestamp			[ ]		
<input checked="" type="checkbox"/> read_status	11	bool			[v]		
 notification_to_us...	12	int4			[v]		

Table 3.5: Data dictionary of notification


















Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
 id	1	int4			[v]		
 created_date	2	timestamp			[ ]		
 last_modified_date	3	timestamp			[ ]		
 created_by	4	int4			[ ]		
 last_modified_by	5	int4			[ ]		
<input checked="" type="checkbox"/> is_active	6	bool			[ ]		
<input checked="" type="checkbox"/> is_test_data	7	bool			[ ]	false	
 spare_column	8	varchar(255)		<a href="#">default</a>	[ ]		
 access_token_vali...	9	int4			[ ]		
 additional_infor...	10	text		<a href="#">default</a>	[ ]		
 authorities	11	varchar(255)		<a href="#">default</a>	[ ]		
 authorized_grant...	12	varchar(255)		<a href="#">default</a>	[ ]		
 autoapprove	13	varchar(255)		<a href="#">default</a>	[ ]		
 client_id	14	varchar(255)		<a href="#">default</a>	[v]		
 client_secret	15	varchar(255)		<a href="#">default</a>	[ ]		
 refresh_token_vali...	16	varchar(255)		<a href="#">default</a>	[ ]		
 resource_ids	17	varchar(255)		<a href="#">default</a>	[ ]		
 scope	18	varchar(255)		<a href="#">default</a>	[ ]		
 web_server_redir...	19	varchar(255)		<a href="#">default</a>	[ ]		

Table 3.6: Data dictionary of client details









Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
 id	1	int2			[v]		
 created_date	2	timestamp			[ ]		
 last_modified_date	3	timestamp			[ ]		
 created_by	4	int4			[ ]		
 last_modified_by	5	int4			[ ]		
<input checked="" type="checkbox"/> is_active	6	bool			[ ]		
<input checked="" type="checkbox"/> is_test_data	7	bool			[ ]	false	
 spare_column	8	varchar(255)		<a href="#">default</a>	[ ]		
 remarks	9	varchar(100)		<a href="#">default</a>	[ ]		
 role_name	10	varchar(20)		<a href="#">default</a>	[v]		

Table 3.7: Data dictionary of role









Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
 id	1	int4			[v]		
 created_date	2	timestamp			[ ]		
 last_modified_date	3	timestamp			[ ]		
 created_by	4	int4			[ ]		
 last_modified_by	5	int4			[ ]		
<input checked="" type="checkbox"/> is_active	6	bool			[ ]		
<input checked="" type="checkbox"/> is_test_data	7	bool			[ ]	false	
 spare_column	8	varchar(255)		<a href="#">default</a>	[ ]		
 remarks	9	varchar(100)		<a href="#">default</a>	[ ]		
 subject_name	10	varchar(20)		<a href="#">default</a>	[v]		

Table 3.8: Data dictionary of subject

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
<a href="#">123</a> id	1	int4			[v]		
<a href="#">🕒</a> created_date	2	timestamp			[ ]		
<a href="#">🕒</a> last_modified_date	3	timestamp			[ ]		
<a href="#">123</a> created_by	4	int4			[ ]		
<a href="#">123</a> last_modified_by	5	int4			[ ]		
<input checked="" type="checkbox"/> is_active	6	bool			[ ]		
<input checked="" type="checkbox"/> is_test_data	7	bool			[ ]	false	
<a href="#">ABC</a> spare_column	8	varchar(255)		<a href="#">default</a>	[ ]		
<a href="#">123</a> class_group_id	9	int4			[v]		
<a href="#">123</a> subject_id	10	int4			[v]		
<a href="#">123</a> teacher_id	11	int4			[v]		

Table 3.9: Data dictionary of Teacher

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
<a href="#">123</a> id	1	int4			[v]		
<a href="#">123</a> role_id	2	int2			[v]		
<a href="#">123</a> user_id	3	int4			[v]		

Table 3.10: Data dictionary of user role

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
<a href="#">123</a> id	1	int4			[v]		
<a href="#">🕒</a> created_date	2	timestamp			[ ]		
<a href="#">🕒</a> last_modified_date	3	timestamp			[ ]		
<a href="#">123</a> created_by	4	int4			[ ]		
<a href="#">123</a> last_modified_by	5	int4			[ ]		
<input checked="" type="checkbox"/> is_active	6	bool			[ ]		
<input checked="" type="checkbox"/> is_test_data	7	bool			[ ]	false	
<a href="#">ABC</a> spare_column	8	varchar(255)		<a href="#">default</a>	[ ]		
<a href="#">ABC</a> email	9	varchar(30)		<a href="#">default</a>	[v]		
<a href="#">ABC</a> first_name	10	varchar(15)		<a href="#">default</a>	[v]		
<a href="#">ABC</a> last_name	11	varchar(15)		<a href="#">default</a>	[v]		
<a href="#">ABC</a> middle_name	12	varchar(15)		<a href="#">default</a>	[ ]		
<a href="#">ABC</a> mobile_no	13	varchar(30)		<a href="#">default</a>	[ ]		
<a href="#">ABC</a> password	14	varchar(255)		<a href="#">default</a>	[v]		
<a href="#">ABC</a> user_name	15	varchar(20)		<a href="#">default</a>	[v]		
<a href="#">123</a> class_group_id	16	int4			[ ]		

Table 3.11: Data dictionary of user table

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
id	1	int4			[v]		
created_date	2	timestamp			[ ]		
last_modified_date	3	timestamp			[ ]		
created_by	4	int4			[ ]		
last_modified_by	5	int4			[ ]		
<input checked="" type="checkbox"/> is_active	6	bool			[ ]		
<input checked="" type="checkbox"/> is_test_data	7	bool			[ ]	false	
spare_column	8	varchar(255)		<a href="#">default</a>	[ ]		
allocation_status	9	varchar(255)		<a href="#">default</a>	[v]		
assignment_soluti...	10	varchar(255)		<a href="#">default</a>	[ ]		
due_date	11	timestamp			[v]		
final_date	12	timestamp			[v]		
obtained_marks	13	int4			[ ]		
submission_date	14	timestamp			[ ]		
teacher_remarks	15	text		<a href="#">default</a>	[ ]		
allocated_by_user...	16	int4			[v]		
allocated_to_user...	17	int4			[v]		
assignment_id	18	int4			[v]		

Table 3.12: Data dictionary of assignment allocation

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
id	1	int4			[v]		
created_date	2	timestamp			[ ]		
last_modified_date	3	timestamp			[ ]		
created_by	4	int4			[ ]		
last_modified_by	5	int4			[ ]		
<input checked="" type="checkbox"/> is_active	6	bool			[ ]		
<input checked="" type="checkbox"/> is_test_data	7	bool			[ ]	false	
spare_column	8	varchar(255)		<a href="#">default</a>	[ ]		
assignment_state	9	varchar(255)		<a href="#">default</a>	[ ]		
assignment_title	10	varchar(255)		<a href="#">default</a>	[v]		
description	11	text		<a href="#">default</a>	[v]		
file_path	12	varchar(255)		<a href="#">default</a>	[ ]		
full_score	13	int4			[v]		
course_id	14	int4			[v]		
note_id	15	int4			[ ]		

Table 3.13: Data dictionary of assignment

## CHAPTER 4: SYSTEM IMPLEMENTATION & TESTING

### 4.1. Implementation and overview:

Incremental model is used for our project as, our project starts with a set of requirements and requirements divided into multiple standalone modules. Incremental 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 Increment, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of module adds function to the previous release. The process continues until the complete system achieved.

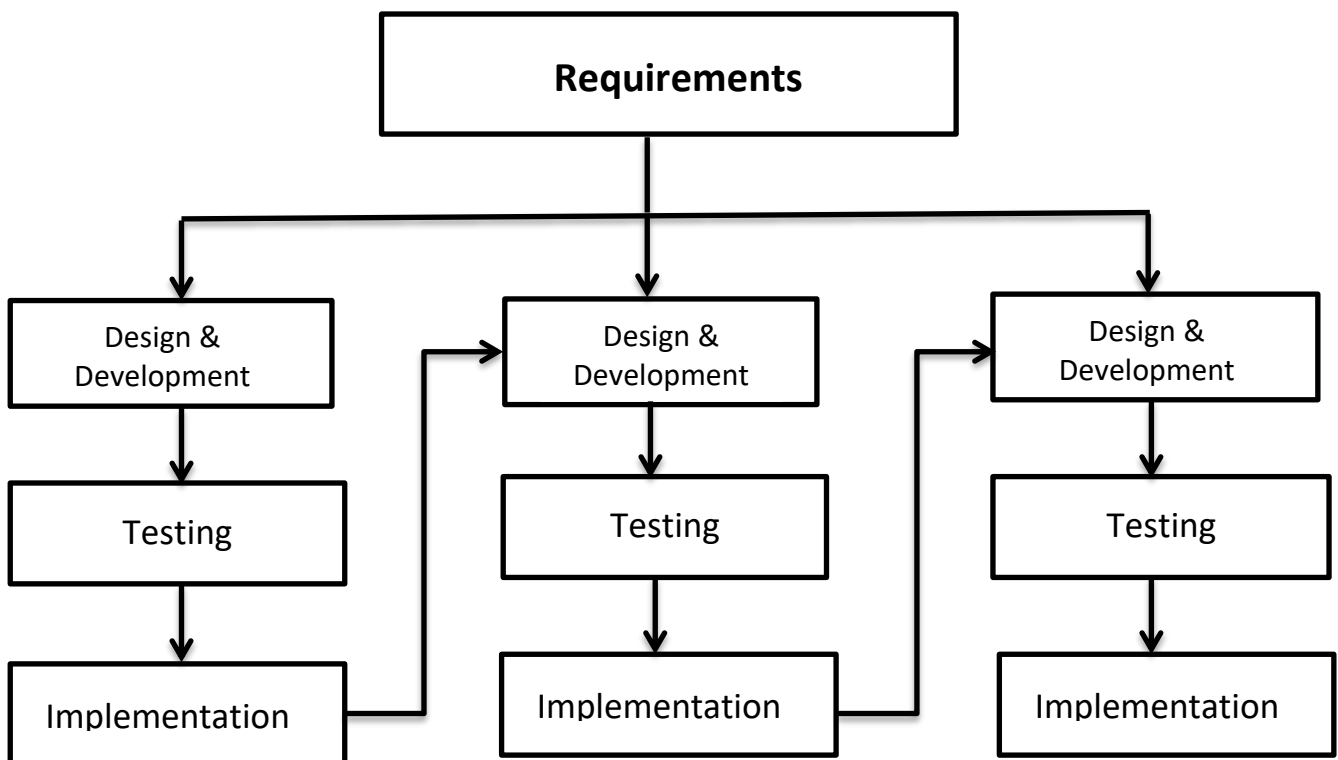


Figure 4.1: Incremental Model

## 4.2 Tools Used

### 4.2.1 Front End Tools

- **VS Code:**

Visual Studio Code (famously known as VS Code) is a free open source text editor by Microsoft. VS Code is available for Windows, Linux and macOS. Although the editor is relatively lightweight, it includes some powerful features that have VS Code one of the most popular development environment tools in recent times.

- **React JS:**

React JS is a front-end library that has gradually become the go-to framework for modern web development within the JavaScript community. The React JS framework is an open-source JavaScript framework and library developed by Facebook. It is used for building interactive user interfaces and web applications quickly and efficiently with significantly less code than you would with vanilla JavaScript.

- **Bootstrap:**

Bootstrap is the most popular CSS Framework for developing responsive and mobile-first websites. It is a free, open source front-end development framework for creation of websites and web apps.

### 4.2.2 Back End Tool:

- **Language: JAVA**

Java is a simple programming language. Java makes writing, compiling and debugging programming easy. It helps to create reusable code and modular programs. It was originally designed for developing programs for set-top boxes and handheld devices but later became a popular choice for creating web applications.

- **Database: DBeaver**

DBeaver is a free open source database management tool. It provides features to manage the full data lifecycle on your desktop, from the definition of a database table to the update of a column value.

- **IntelliJ IDEA:**

IntelliJ IDEA is a special programming environment or integrated development environment (IDE) largely meant for java. This environment is used especially for the development of programs. The biggest reason it is regarded as one of the best programming tools based on Java is its assistance features, which makes it is easy to use and makes the programs created by it very well designed. It also has advanced error checking features which allows faster and easier error checking.

## 4.3 Testing

### 4.3.1 Unit Testing

S.N	Test case id	Test description	Input test data	Expected result	Actual result	remarks
1	T-01	Login user with email and password	admin 12345(wrong username & password)	Failed to login with alert box	Username or Password Did Not Match ! Contact Your administrator.	Failed to login.
		Login user with email and password	admin Test@123	Login and redirect to dashboard	Success	Login success
2	T-02	Enter teacher first name, middle name, last name, user name, email, password, phone no	Sandeepa, Gurung, sandeepa, sandeepagurung012@gmail.com, grgdeepa, 9844905303	Display success alert	Success	Add teacher success.
3	T-03	Enter student first name, middle name, last name,	Anju, Gurung, sandeepa, anjuagurungicte@gmail.com,	Display success alert	Success alert	Add student success.

		user name, email, password, phone no	anju12345, 9844905203			
4	T-04	Role list	Admin, Teacher, Student	Display	Successful	Successful
5	T-05	Teacher name, Class name, Subject	Sandeepa Gurung, Eight semester, Microprocessor	Display success alert	Success alert!	Successful
6	T-06	Class group, Remarks	Eight Semester, Eight Semester	Display success alert	Success alert!	Successful
7	T-07	Subject name, Remarks	Microprocessor, Microprocessor	Display success alert	Display data has been successfully saved	Successful
8	T-08	Course name, Subject, Remarks	BICTE, Microprocessor, Microprocessor	Display success alert	Display data has been successfully saved	Successful
9	T-09	Teacher name, Review	Sandeepa Gurung, Class is going well !	Submitted successfully	successfull	Successful
10	T-10	New Password, Confirm Password	Test@12345, Test@12345	Display success	successfull!	Password changed successfully
11	T-11	Note title, Description, Upload Note	Microprocessor, Take help, Document	Display success alert	Successful	Submitted successfully



12	T-12	Assignment title, Description, Full score, Subject,Courses, Note	Microprocessor, What is microprocessor, 50, Microprocessor, BICTE, file	Submitted successfully	Successful	Successful
13	T-13	Assignment, Class group, Student, Due date, Final Date	Microprocessor, Eight Semester, Anju Gurung, 01-28-2023, 01-30-2-23	Display success alert	Successful	Successful.
14	T-14	Assignment title, Assignment file, Full marks, Subject name, Course name, Action	Microprocessor, file, 50, Microprocessor, BICTE, Pending submission, Marked Submission, Submitted Assignment, Deadline Exceed Assignment	Display success alert	Successful	Successful.
15	T-15	Review by, Review for, Review Message, Submitted Date	Admin, Sandeepa Gurung, "Class is going well", 01-12-2023	Display Message	Successfully Displaed	Successful
16	T-16	Assignment title, Assignment file, Full marks, Subject name, Course name, Assigned by	Microprocessor, file, 50, Microprocessor, BICTE, Sandeepa Gurung	Display success alert	Successful	Successful

17	T-17	Teacher name, Review	Sandeepa Gurung, "Class is going well"	Display success alert	Successful	Successful
18	T-18	Message	"Good"	Display Message	Successful	Successful

Table 4.1: Unit Testing

### 4.3.2 Integration Testing

Test suite plan	Test case id	Description	Remarks
User	T-01	Login	Test success.
Admin	T-02	Add teacher	Test success.
	T-03	Add student	Test success.
	T-04	Add role list	Test success.
	T-05	Add teacher class mapping	Test success
	T-06	Add class group	Test success.
	T-07	Add subject	Test success.
	T-08	Add courses	Test success.
	T-09	Add Review	Test success
	T-10	Change password	Test success
Teacher	T-11	Add note	Test success
	T-12	Upload assignment	Test success
	T-13	Allocate Assignment	Test success
	T-14	Allocated Assignment	Test success
	T-15	Review	Test success
Student	T-16	View assignment	Test success
	T-17	Upload assignment	Test success
	T-18	Review	Test success
	T-19	Notification	Test success

Table 4.2: Integration Testing

### 4.3.3 System Testing

Test suite plan	Test case id	Description	Remarks
User	T-01	Login	Test success.
Teacher	T-02	Add notes	Test success.
	T-03	Upload assignment	Test success.
	T-04	Allocate assignment	Test success.
	T-05	Allocated assignment	Test success.
	T-06	Add review	Test success.
Student	T-07	View assignment	Test success
	T-08	Upload assignment	Test success
	T-09	View grades	Test success
	T-10	Add review	Test success

Table 4.3: System Testing

## **CHAPTER 5: CONCLUSION AND RECOMMENDATION**

### **5.1 Conclusion**

Our assignment management system ‘Online AMS’ is a web based online assignment submission system which facilitated submission of assignments by students has been presented. System consisted of modules where students could download assignment created by teachers and upload their assignment on due dates. Notification are sent to students when new students when new assignment is added and also when teacher grade the assignment.

Our main goal is to provide teachers, students and administrators with a centralized platform to create, organize and manage assignments, assessments and course materials. This system aims to improve the efficiency, effectiveness and organization of the assignment submission process, while also providing a better overall teaching and learning experience.

### **5.2 Recommendation**

Due to the limitation of time faced during the course of this project work, an important feature that can add to the efficiency of the system functionality was identified with system. Therefore, the identified system feature is recommended for further development on the system. The recommendation is:

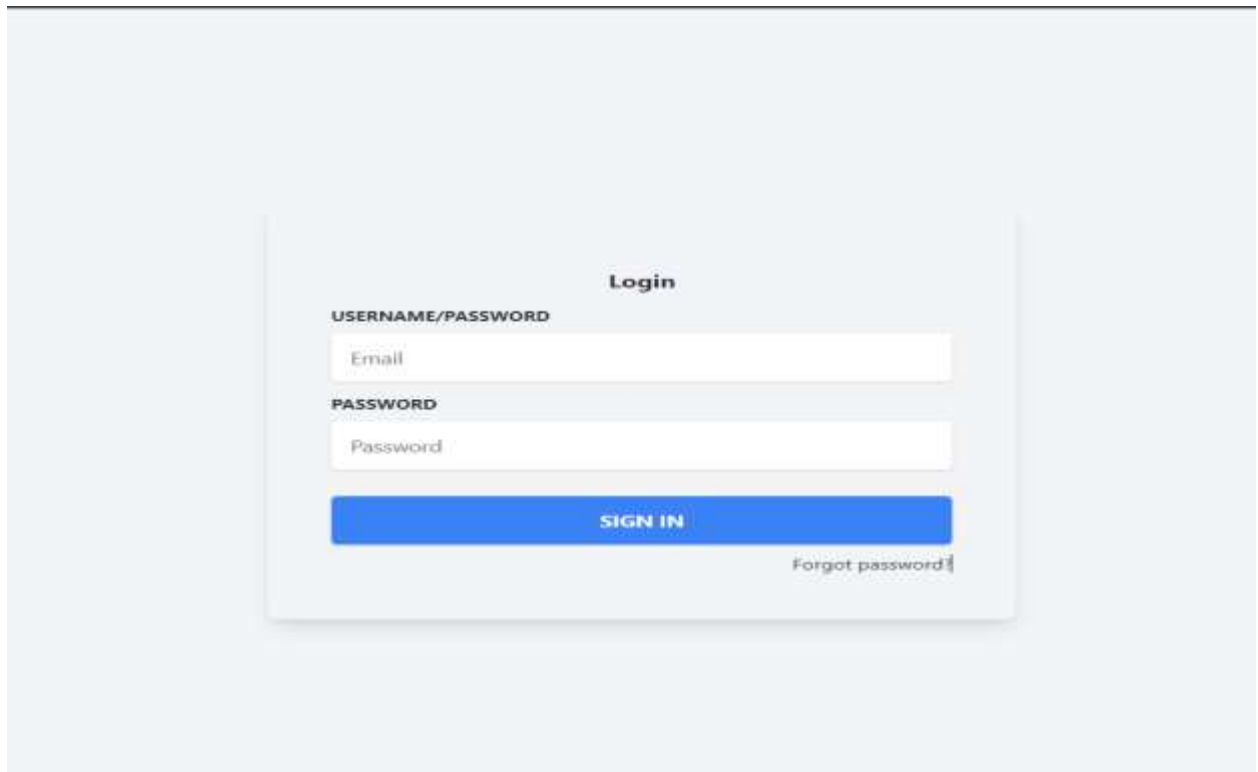
- i. To improve the system posting of student scores directly to their mobile phone after marking the assignments.
- ii. Implementation of the system in a way that can handle the plagiarism between students.
- iii. Furthering research into this subject, given its great potentials.

## References

- Sam Hsu, "HWSAM: A Web-based Automated Homework Submission System", 1998 ASEE/IEEE Frontiers in Education Conference (FIE '98), Nov 4-7,1998, Tempe, Arizona,pp.580-582.
- Wawasan Open University (2011) About Us. Retrieved November 28 from [http://www.wou.edu.my/study\\_with\\_us.html](http://www.wou.edu.my/study_with_us.html)
- Blackboard, 2014. Blackboard website for Asia Pacific [online] Available at: <http://www.blackboard.com> [Accessed 30 May 2014].
- Paul Darbyshire, Distributed Web-Based Assignment Management, Web-Based Learning and Teaching Technologies: Opportunities and Challenges, Anil Aggarwal (ed), pp 198-215.
- Brown, S., Race, P., & Bull, J.(1999). Computer-Assisted Assessment in Higher Education. London: Clays.
- <http://www.edu-link.com>
- A. Tregobov, "*The Web-Based Assignment Submission Systems*", presented at NAWeb '98, University of New Brunswick, Canada, 1998.
- David Jones, Sandy Behrens, "*Online Assignment Management: An Evolutionary Tales*", Hawaii International Conference on System Sciences, vol.5, no. 5,pp.156c, Track 5, 2003.
- Bridge, P. and Appleyard, R.M. (2008). A comparison of electronic and paper-based assignment submission and feedback. British Journal of Educational Technology, 39 (4), 669-271.
- BlackboardLearn, Release 9.1 – Our Next Generation Teaching and learning Platform Available at: [https://online.fiu.edu/files/blacklearn/black\\_learn\\_brochure.pdf](https://online.fiu.edu/files/blacklearn/black_learn_brochure.pdf) [Accessed 1 June 2014].
- Google for Education. Google Classroom. (2014). <https://www.google.com/edu/classroom> . Retrieved January 12, 2023
- Linnaeus University Sweden [online] Available at: <http://Inu.divaportal.org>
- Denton's Electronic Feedback Software <http://www.dtfs/e-assessment.edu.com>

## APPENDIX – A: USER INTERFACE

### 3.2.2 Interface design



The login interface is a centered card on a light blue background. It features a title 'Login' at the top. Below the title, there are two input fields: the first is labeled 'USERNAME/PASSWORD' and contains the placeholder text 'Email'; the second is labeled 'PASSWORD' and contains the placeholder text 'Password'. A blue button with the text 'SIGN IN' is positioned below the password field. To the right of the button, there is a link that says 'Forgot password?'. The entire card has a subtle shadow.

Figure 1: Interface design of login

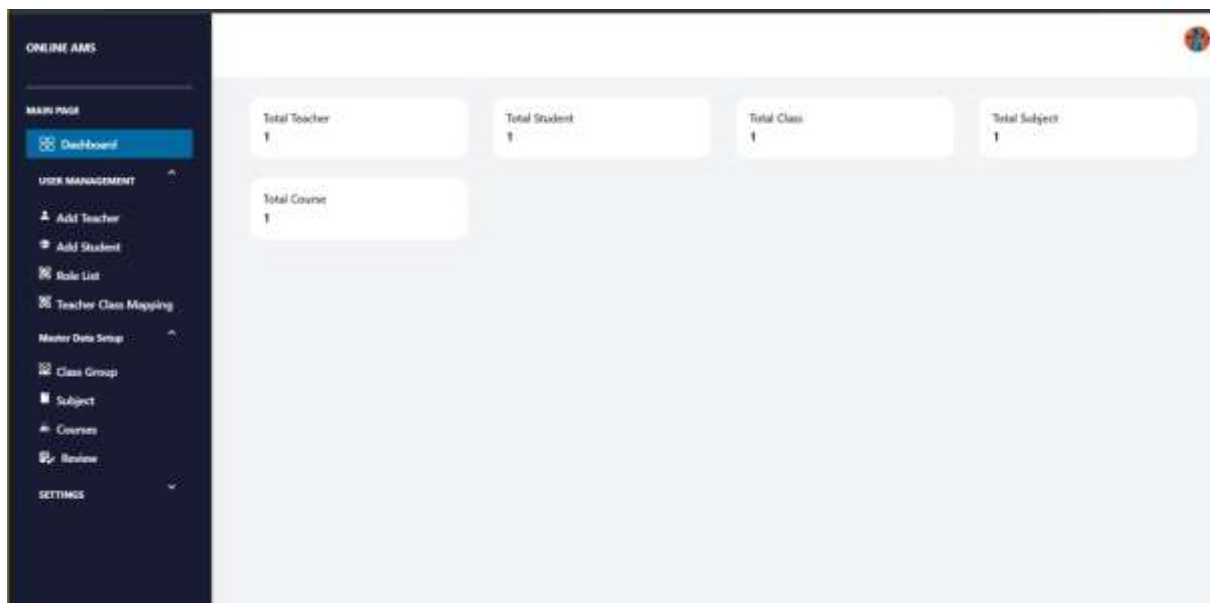


Figure 2: Interface design of Admin Dashboard

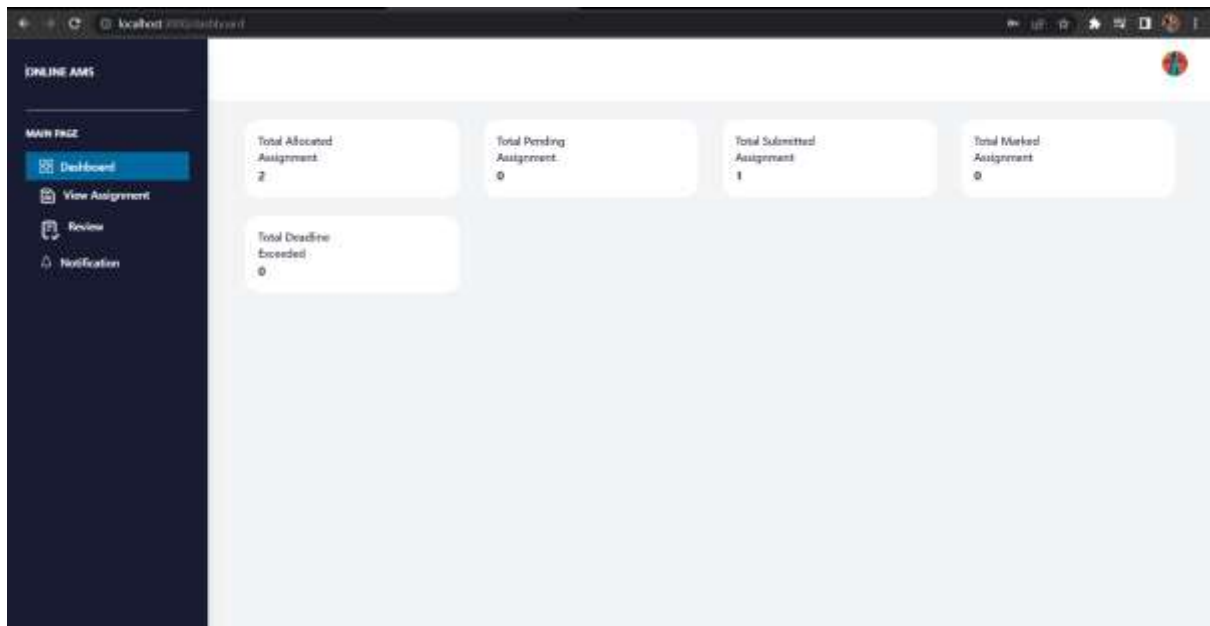


Figure3: Interface design of Student Dashboard

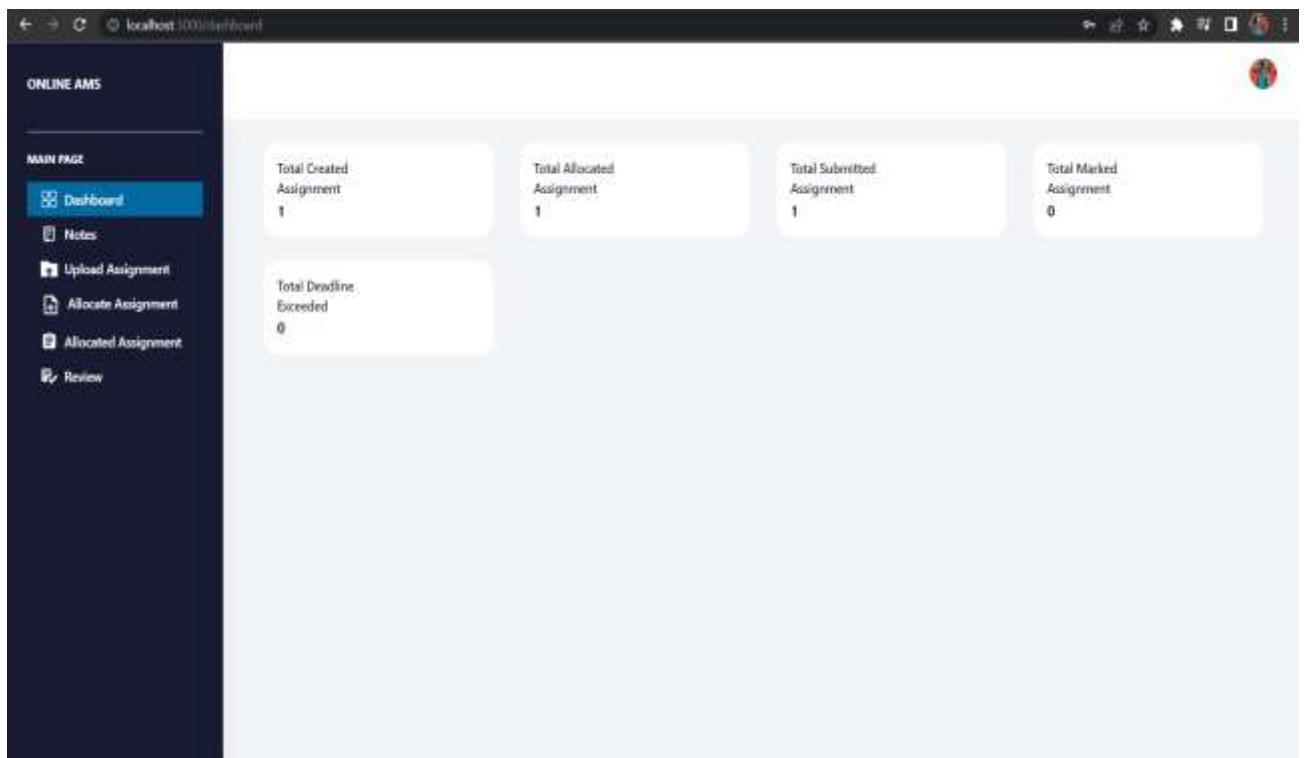


Figure 4: Interface design of Teacher Dashboard

**ONLINE AMS**

**MAIN PAGE**

- Dashboard
- Notes
- Upload Assignment**
- Allocate Assignment
- Allocated Assignment
- Review

### Upload Assignment

Assignment Title \*

Description \*

Full Score \*

Subject \*

Course \*

Note

**UPLOAD ASSIGNMENT FILE** No file found

**SUBMIT**

Assignment Title	Subject Name	Course Name	Assignment File Name	Action
No data available				

Figure 5: Interface design of Upload Assignment

**ONLINE AMS**

**MAIN PAGE**

- Dashboard
- Notes
- Upload Assignment
- Allocate Assignment
- Allocated Assignment**
- Review

### Allocated Assignment

**BACK TO LIST**

Assignment Title:

ID	Student Name	Marks Obtained	Teacher Remarks	Action
1	Geurab Pantha	0		<b>VIEW</b>

Figure 6: Interface design of Allocated Assignment



ONLINE AMS

MAIN PAGE

Dashboard

USER MANAGEMENT

Add Teacher

Add Student

Role List

Teacher Class Mapping

Master Data Setup

SETTINGS

Review

Add Student

First Name \*

Arju

Middle Name

Last Name \*

Gurung

User Name \*

arju

Email \*

arjagurungict033@gmail.com

Password \*

Mobile Number \*

9816198560

Class Group \*

None

SUBMIT

CANCEL

S.N.	Name	Middle Name	Last Name	User Name	Mobile Number	Role	Class	Action
1	Arju	-	Gurung	arju	9816198560	Student	None	EDIT DELETE
2	Sujan	-	Shrestha	sujan	9844905303	Student	None	EDIT DELETE

Figure 7: Interface Design of Add Student

ONLINE AMS

MAIN PAGE

Dashboard

USER MANAGEMENT

Add Teacher

Add Student

Role List

Teacher Class Mapping

Master Data Setup

SETTINGS

Review

Add Teacher

FIRST NAME \*

Sulove

MIDDLE NAME

Kumar

LAST NAME \*

Shrestha

USER NAME \*

sulove

EMAIL \*

sulovekumar123@gmail.com

PASSWORD \*

\*\*\*\*\*

MOBILE NUMBER \*

977-9844905303

SUBMIT

CANCEL

S.N.	Name	Middle Name	Last Name	User Name	Mobile Number	Role	Action
1	Sandhya	-	Gurung	sandhya	984895303	Teacher	EDIT DELETE

Figure 8: Interface design of Add Teacher

ONLINE AMS

MAIN PAGE

Dashboard

USER MANAGEMENT

Add Teacher

Add Student

Role List

Teacher Class Mapping

Master Data Setup

Class Group

Subject

Courses

SETTINGS

Review

Class

Class Group \*

Remarks \*

Nine

Ninem hi

SUBMIT

CANCEL

S.N.	Class Group	Remarks	Action
1	100004	Ajghghy87187u	EDIT <span>DELETE</span>
2	BCTEW75	il	EDIT <span>DELETE</span>
3	Nine	Ninem hi	EDIT <span>DELETE</span>

Figure 9: Interface design of Class

ONLINE AMS

MAIN PAGE

Dashboard

USER MANAGEMENT

Add Teacher

Add Student

Role List

Teacher Class Mapping

Master Data Setup

Class Group

Subject

Courses

SETTINGS

Review

Subject

Subject Name \*

Remarks \*

Microprocessor

Microprocessor

SUBMIT

CANCEL

S.N.	Subject	Remarks	Action
1	Microprocessor	Microprocessor	EDIT <span>DELETE</span>
2	Software Engineering	Hi	EDIT <span>DELETE</span>

Figure 10: Interface design of Subject

ONLINE AMS

MAIN PAGE

Dashboard

Notes

Upload Assignment

Allocate Assignment

Allocated Assignment

Review

Upload Notes

Note Title \*

Description \*

Computer

www.youtube.com

UPLOAD NOTE

virtual info...pdf

SUBMIT

Note Title	Description	File Name	Action
Microprocessor	microprocessor	ASSIGNMENT8842904455756443551.DOCX	EDIT <span>DELETE</span>
Computer	www.youtube.com	ASSIGNMENT2811451719438726461.DOCX	EDIT <span>DELETE</span>

Figure 11: Interface design of Upload Notes

ONLINE AMS

MAIN PAGE

Dashboard

USER MANAGEMENT

Add Teacher

Add Student

Role List

Teacher Class Mapping

Master Data Setup

Class Group

Subject

Courses

SETTINGS

Review

Course

Course Name \*

Subject \*

Remarks \*

BCTED

Microprocessor

Microprocessor

SUBMIT

CANCEL

S.N	Course	Subject	Remarks	Action
1	BCTED	Microprocessor	IS	EDIT <span>DELETE</span>
2	BCTED	Microprocessor	Microprocessor	EDIT <span>DELETE</span>

Figure 12: Interface design of Course

## APPENDIX – B: System Coding

datasource:  
url: jdbc:postgresql://localhost:5432/assignment\_db  
username: postgres  
password: postgres  
driver-class-name: org.postgresql.Driver  
hikari:  
minimum-idle: 5  
maximum-pool-size: 300  
connection-timeout: 600000  
idle-timeout: 60000  
max-lifetime: 1800  
leak-detection-threshold: 600000  
initialization-mode: never #set this to always in first execution  
jpa:  
show-sql: true  
hibernate:  
ddl-auto: update  
database: postgresql  
properties:  
hibernate:  
jdbc:  
lob: non\_contextual\_creation: true  
format\_sql: true  
mvc:  
throw-exception-if-no-handler-found: true  
jackson:  
serialization:  
FAIL\_ON\_EMPTY\_BEANS: false  
INDENT\_OUTPUT: true

Figure 13: Database connection

```

<div className="bg-white shadow-md rounded px-8 pt-6 pb-8 mb-4 flex flex-col my-2">
  <div className="card-body">
    <div className="flex">
      <h6 className="mr-auto text-xs mb-3 font-bold leading-tight tracking-tight text-gray-900 md:text-xl dark:text-white">
        Create Assignment
      </h6>
      <Button variant={'contained'} onClick={() => navigate(-1)}>Back to list</Button>
    </div>
    <hr className="mb-3" />
    <Formik
      enableReinitialize
      initialValues={initialValue}
      validationSchema={questionValidation}
      onSubmit={async (values,action) => {
        action.setSubmitting(false)
        await handleSaveAssignment(values,action.resetForm)
      }}
    >
      {({
        values,
        errors,
        touched,
        handleChange,
        handleBlur,
        setFieldValue,
      }) => (
        <Form>
          <div className="-mx-3 md:flex mb-6">
            <FieldArray name="assignmentQuestionDtoList">
              ({({ insert, remove, push }) => (
                <div >
                  {values.assignmentQuestionDtoList.length > 0 &&
                    values.assignmentQuestionDtoList.map((friend, index) => (
                      <div className="flex px-3 mb-2" key={friend.id}>
                        <div>
                          <label>
                            htmlFor={`assignmentQuestionDtoList.${index}.question`}
                            Question {index + 1}
                          </label>
                          <Field
                            className={"input--primary"}
                            name={`assignmentQuestionDtoList.${index}.question`}
                            type="text"
                          />

```

```

<ErrorMessage
name={`assignmentQuestionDtoList.${index}.question`}
component="div"
className="flex gap-1 mt-2 text-red-500 text-xs"
/>
</div>
<div className="ml-2">
<button
type="button"
className={`bg-red-500 px-3 rounded hover:bg-red-300 text-white mt-8`}
onClick={() => remove(index)}
>
X
</button>
</div>
</div>
)))}
<Button
type="button"
variant={`contained`}
className="secondary"
onClick={() => push({ id: generateUUID(), question: " })}>
Add Question
</Button>
</div>
)}
</FieldArray>
</div>
<div className="flex flex-row-reverse mt-6">
<Button
variant="contained"
color="success"
className="ml-auto"
type="submit">Submit</Button>
</div>
</Form>
)}
</Formik>
</div>
</div>
<DataTable columns={columns} data={assignmentList}/>
</>

```

Figure 14: Create assignment

```

<div>
<div className="-mx-3 md:flex mb-6">
<div className="sm:w-1/2 h-1/2 md:w-4/12 px-3 mb-2">
<label
className="block tracking-wide text-grey-darker text-xs font-bold mb-2"
htmlFor={"assignmentTitle"}>
Assignment Title</label>
<input
className={'input--primary'}
disabled
value={fullAssignmentForm?.assignmentDto?.assignmentTitle}
/>
</div>
<div className="sm:w-1/2 h-1/2 md:w-4/12 px-3 mb-2">
<label
className="block tracking-wide text-grey-darker text-xs font-bold mb-2"
htmlFor={"assignmentTitle"}>
Assignment File</label>
<input
className={'input--primary'}
disabled
value={fullAssignmentForm?.assignmentDto?.assignmentTitle}
/>
</div>
</div>
<div className="h-1/2 mb-2">
<label
className="block tracking-wide text-grey-darker text-xs font-bold mb-2"
htmlFor={"assignmentTitle"}>
Question List</label>
<Datatable columns={column}
data={fullAssignmentForm?.assignmentDto?.assignmentQuestionDtoList || []} />
</div>
<div className="-mx-3 md:flex mt-6">
<div className="sm:w-1/2 h-1/2 md:w-4/12 px-3 mb-2">
<label
className="block tracking-wide text-grey-darker text-xs font-bold mb-2"
htmlFor={"referenceNote"}>
Reference Note</label>
<Button
title="Download"
className="group flex gap-1 mr-3"
style={{ cursor: 'pointer' }}
onClick={() =>
handleDownloadFile(fullAssignmentForm?.assignmentDto?.noteId,"NOTE",`
${fullAssignmentForm?.assignmentDto?.noteFileName}`)}

```

```

>
{fullAssignmentForm?.assignmentDto?.noteFileName}
</Button>
</div>
<div className="sm:w-1/2 h-1/2 md:w-4/12 px-3 mb-2">
<label
className="block tracking-wide text-grey-darker text-xs font-bold mb-2"
htmlFor={ "fullMaksrs" }>
Full Marks</label>
<input
className={ 'input--primary' }
disabled
value={ fullAssignmentForm?.assignmentDto?.fullScore }
/>
</div>
</div>
<div className="-mx-3 md:flex mt-6 ">
<div className="sm:w-1/2 h-1/2 md:w-4/12 px-3 mb-2">
<label
className="block tracking-wide text-grey-darker text-xs font-bold mb-2"
htmlFor={ "dueDate" }>
Due Date</label>
<input
className={ 'input--primary' }
disabled
value={ fullAssignmentForm?.dueDate }
/>
</div>
<div className="sm:w-1/2 h-1/2 md:w-4/12 px-3 mb-2">
<label
className="block tracking-wide text-grey-darker text-xs font-bold mb-2"
htmlFor={ "finalDate" }>
Final Date</label>
<input
className={ 'input--primary' }
disabled
value={ fullAssignmentForm?.finalDate }
/>
</div>
</div>
{
  (fullAssignmentForm?.allocationStatus === "SUBMITTED" ||
fullAssignmentForm?.allocationStatus === "MARKED") && (
    <div className={ 'bg-gray-100 shadow-md rounded px-8 pt-6 pb-8 mb-4 flex flex-col my-2' }>

```



```

<div className="-mx-3 md:flex mt-6 ">
<div className="sm:w-1/2 h-1/2 md:w-4/12 px-3 mb-2">
<label
className="block tracking-wide text-grey-darker text-xs font-bold mb-2"
htmlFor={"submittedSolution"}>
Submitted Solution</label>
<Button
title="Download"
className="group flex gap-1 mr-3"
style={{ cursor: 'pointer' }}
onClick={() =>
handleDownloadFile(fullAssignmentForm?.id,"ASSIGNMENT_SOLUTION",`${fullAssign
mentForm?.assignmentSolutionFileName} `)}
>
{fullAssignmentForm?.assignmentSolutionFileName}
</Button>
</div>
<div className="sm:w-1/2 h-1/2 md:w-4/12 px-3 mb-2">
<label
className="block tracking-wide text-grey-darker text-xs font-bold mb-2"
htmlFor={"submittedDate"}>
Submitted Date</label>
<input
className={'input--primary'}
disabled
value={fullAssignmentForm?.submittedDate ? fullAssignmentForm?.submittedDate : ""}
/>
</div>

```

Figure 15: Submit assignment

```

const assignedAction = ({row}) => {
  return (
    <>
    <Button onClick={() => navigate(`/viewassignment/${encodeURIComponent(row.original)}`)}`>
    >View</Button>
    </>
  )
}

const handleDownloadFile = async (id) => {
  const requestBody = {
    id,
    fileType: "ASSIGNMENT",
  }
  const res = await props.downloadFileAction(requestBody)
  if (res.data instanceof Blob) {
    downloadFile(res.data, 'ASSIGNMENT.pdf', 'application/pdf')
  }
}

const downloadAssignment = ({row}) => {
  const {id,assignmentFileName} = row.original
  return (
    <Button onClick={() => handleDownloadFile(id)}>{assignmentFileName}</Button>
  )
}

const column = [
  {Header: "SN",Cell:addSerialNumber},
  {Header:"Assignment Title",accessor: "assignmentTitle"},
  {Header:"Assignment File",Cell:downloadAssignment},
  {Header:"Full Marks",accessor: "fullScore"},
  {Header: "Subject Name",accessor: "subjectName"},
  {Header: "Course Name",accessor: "courseName"},
  {Header:"Assigned By",accessor: "allocatedBy"},
  {Header: "Action",Cell:assignedAction}
]

```

Figure 16: View assignment