**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Design Document**

**for**

**T A Management System**

Version 1.0

Prepared by Team 14:

Based on SRS Version 1.0 prepared by Team 14

| **Thukkani Dinesh Reddy** | **M241022CS** | **dinesh\_m241022cs@nitc.ac.in** |
| --- | --- | --- |
| **Talari Bhanu Prakash** | **M241171CS** | **talari\_m241171cs@nitc.ac.in** |
| **Gayas Uddin Ahmad** | **M241032CS** | **gayas\_m241032cs@nitc.ac.in** |
| **Mohammed Naseef** | **M241272CS** | **mohammed\_m241272cs@nitc.ac.in** |
|  |  |  |

## 

| **Project Manager:** | **Thukkani Dinesh Reddy** |
| --- | --- |
| **Project Client:** | **NITC** |
| **Course:** | **CS6103E - Software Systems lab** |
| **Date:** | **11.11.2024** |

# Glossary

| MERN | MongoDB, Express, React, Node.js |
| --- | --- |
| UML | Unified Modelling Language |
| HTML | Hyper Text Markup Language |
| CSS | Cascading Style Sheets |

## **Table of contents**

[**Glossary**](#_tyjcwt) **2**

[**Table of contents**](#_3dy6vkm) **3**

[**Detailed Design through UML diagrams**](#_1t3h5sf) **4**

[1.1 System model using Class Diagram](#_4d34og8) 4

[1.2 Responsibilities - Use Case Diagram](#_17dp8vu) 5

[1.3 System Interactions through Sequence Diagrams](#_3rdcrjn) 6

[1.3.1](#_26in1rg) Teacher Login Interaction 6

[1.3.2](#_lnxbz9) Student Login Interaction 7

[1.3.3](#_44sinio) Admin Assigns Students to Teachers 8

[1.3.4](#_1y810tw) Student-Teacher Interaction 9

1.3.5 Request for Assignment Change 10

[1.4 Control and Data Flows through Activity Diagrams](#_2xcytpi) 11

[1.4.](#_1ci93xb)1 Student Workflow 11

[1.4.2](#_3whwml4) Teacher Workflow 12

[1.4.3](#_2bn6wsx)  Admin Workflow 13

[**Database Design**](#_qsh70q) **14**

[2.1 ER Diagram](#_3as4poj) 14

[**Implementation Plans**](#_1pxezwc) **15**

[3.1 Technology Stack](#_49x2ik5) 15

[3.2 Work Estimates](#_2p2csry) 15

[**References**](#_3o7alnk) **16**

# 

# 

# 1. Detailed Design through UML diagrams

## 1.1 System model using Class Diagram

Class Diagram in the Unified Modelling Language is a type of static structure diagram that describes the structure of a system by showing the system’s classes, their attributes, operations (or methods) and the relationships among classes.

### 

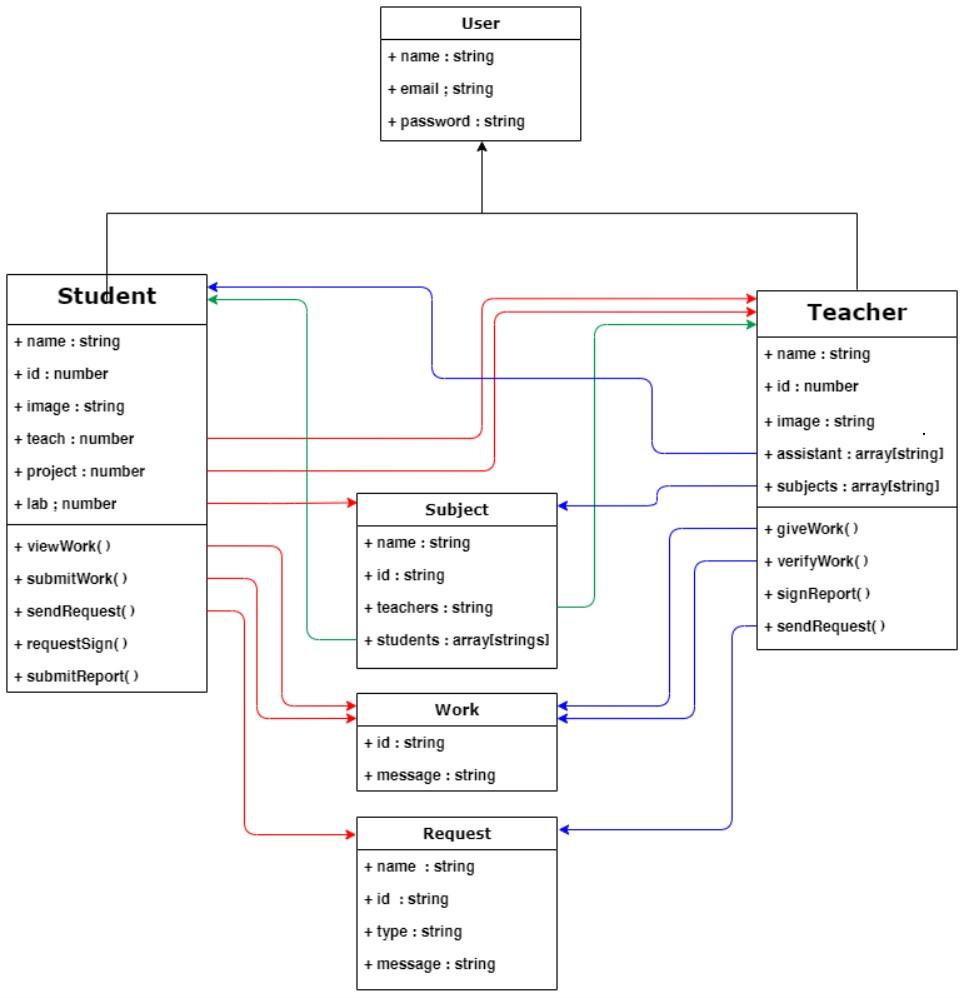
**

Figure 1.1 Class Diagram

## 

## 1.2 Responsibilities - Use Case Diagram

Use case diagram graphically depicts the user's possible interactions with the system. It shows the different types of users (actors) and the use cases that the actors perform when they are using the system to solve the customer’s problem. The actor is shown as a stick person and the use case is shown as an ellipse. Lines indicate which actors perform which use cases.

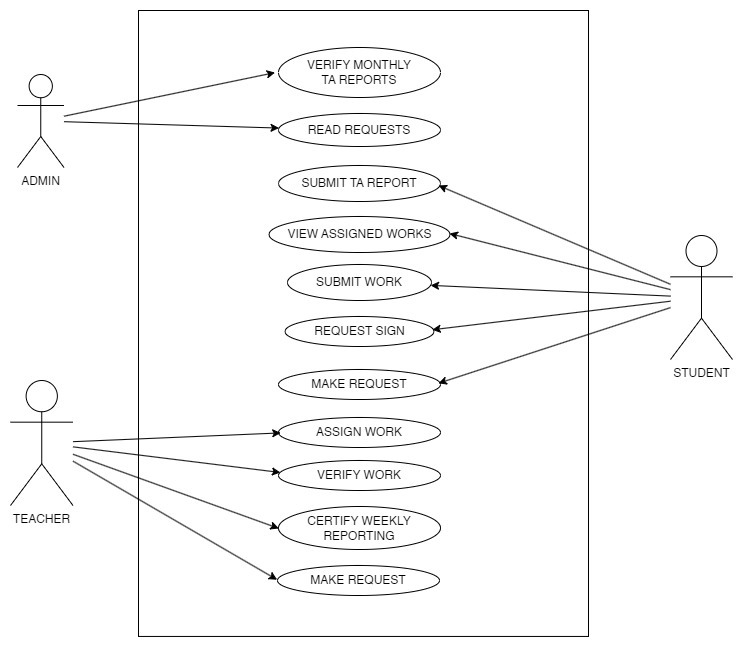
**

Figure 1.2: Use Case Diagram

## 

## 1.3 System Interactions through Sequence Diagrams

Sequence diagrams are interaction diagrams that show the sequence of messages exchanged by the set of objects performing a certain task. A sequence diagram shows, as parallel vertical lines (lifeline), different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur.

### 1.3.1 Teacher Login Interaction

The process where teachers log in and access the system to view and manage

assignments.

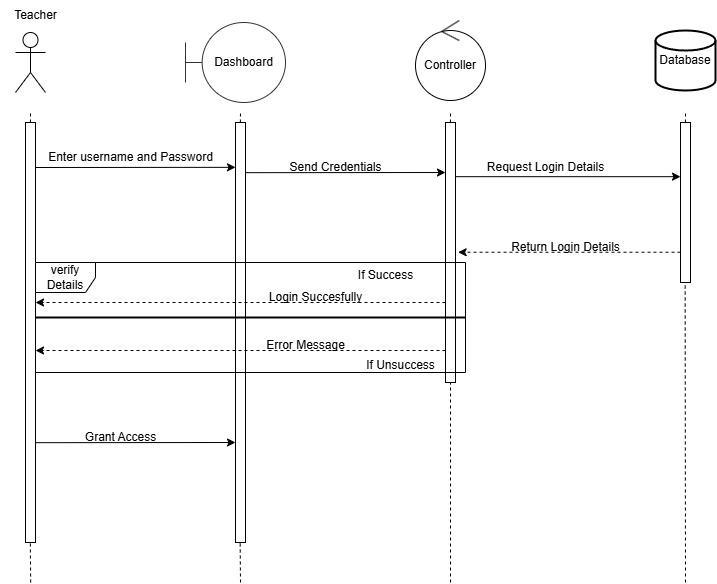
**

Figure 1.3.: Sequence diagram for Teacher Login

### 

### 1.3.2 Student Login Interaction

The process for students to log in and access assignments, resources, and track progress.

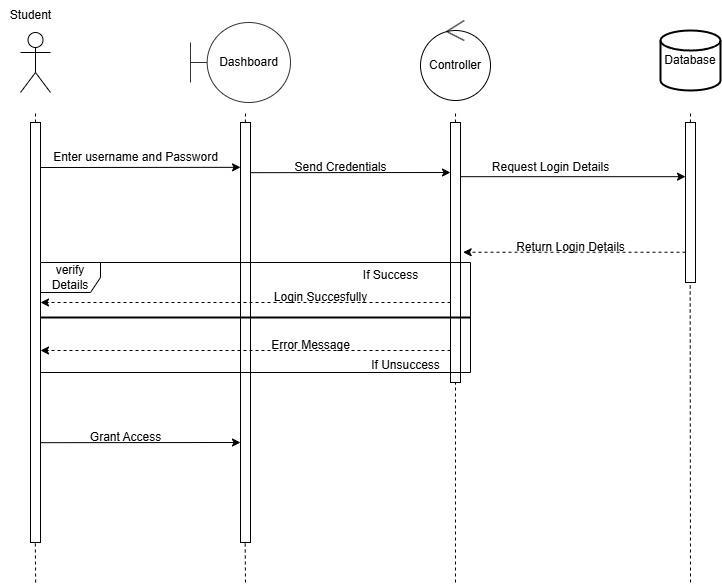
**

Figure 1.4: Sequence diagram for Student Login

### 

### 

### 1.3.3 Admin Assigns Students to Teachers

Admin allocates students to teachers through the system's database.

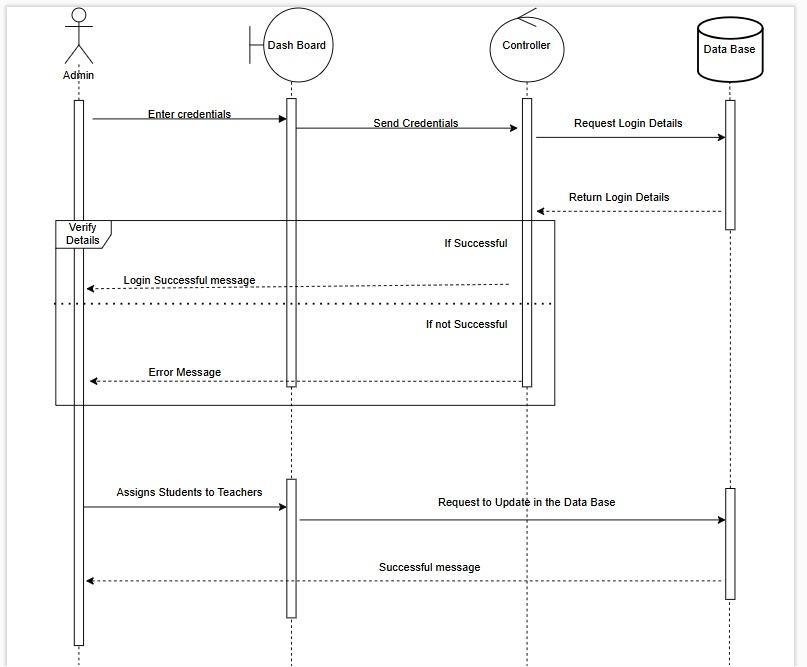
**

Figure 1.5: Sequence diagram for Admin Assigns Students to Teachers

### 

### 

### 

### 1.3.4 Student-Teacher Interaction

Communication and collaboration between students and teachers regarding assignments

and feedback.

### 

Figure 1.6: Sequence diagram for Student-Teacher Interaction

1.3.5 Request for Assignment Change

Students or teachers submit requests to the admin for modifications in

assignments.

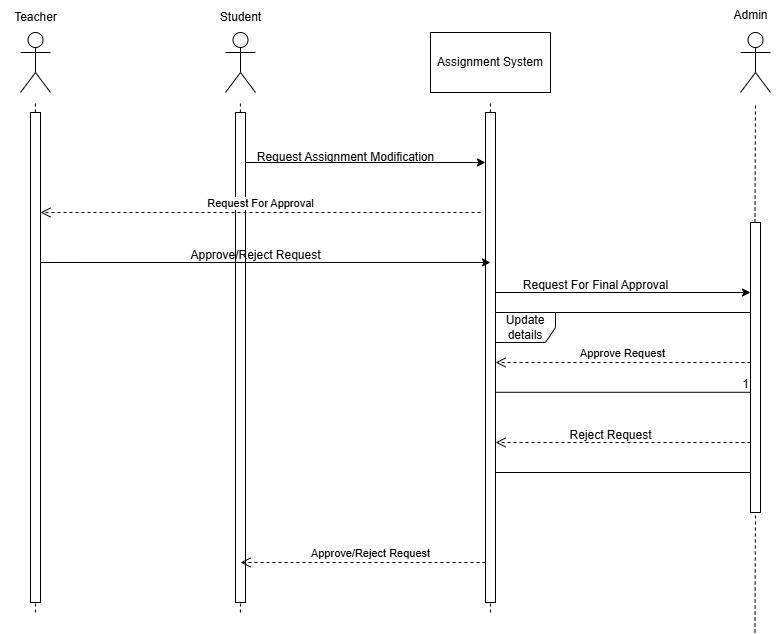


Figure 1.7 Sequence Diagram For Request for Assignment Change

## 

## 1.4 Control and Data Flows through Activity Diagrams

## Activity diagrams graphically represent step-wise activities and actions involved in the workflow within a specific scenario, and helps to understand the flow of work that an object or component performs. Activity diagram uses rounded rectangles to represent a specific system function, arrows to represent flow through the system, decision diamonds to depict a branching decision, and solid horizontal lines to indicate that parallel activities are occurring.

### 1.4.1 Student Work Flow

The steps involved in assigning tasks to students, from creation by the teacher to student

access.

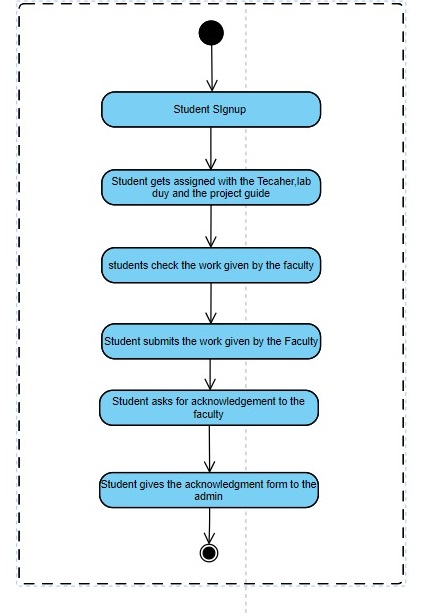
**

Figure 1.8: Activity Diagram For Student Work Flow

### 

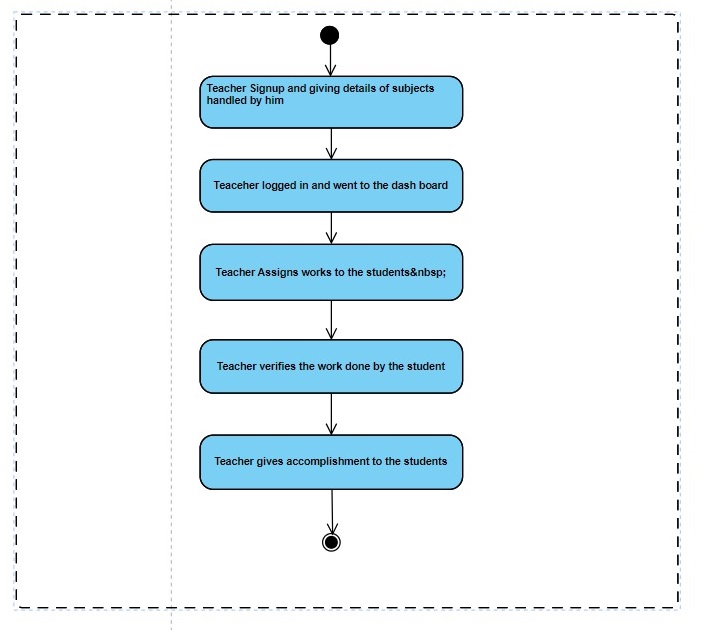
### 

### 1.4.2 Teacher Work Flow

The teacher workflow activity diagram shows steps from lesson planning to student

assessment, highlighting decisions and iterative tasks like adapting materials and

grading. It emphasises streamlined processes for effective teaching outcomes

**

### Figure 1.9: Activity Diagram For Teacher Workflow

### 

### 

### 1.4.3 Admin Workflow

The admin workflow activity diagram outlines tasks from scheduling and resource allocation to managing records and overseeing compliance, emphasising efficient

operations and support for faculty and student needs.

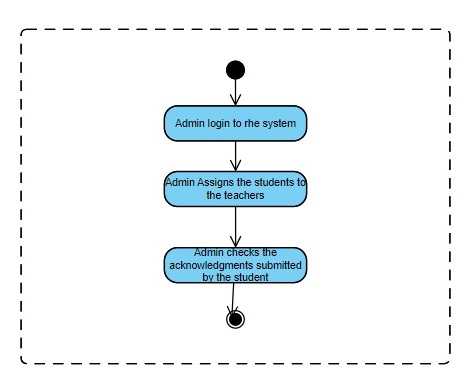
**

Figure 1.10 Activity diagram for Admin Work flow

# 2. Database Design

## 2.1 ER Diagram

ER (Entity-Relationship) model is designed to represent the things that a system needs

to remember in order to perform the system functionalities. It graphically represents the

data model that defines the information structure which should be implemented in the

database. The data objects (entity) are represented by a labelled rectangle and

the relationships are indicated with a labelled line connecting objects.

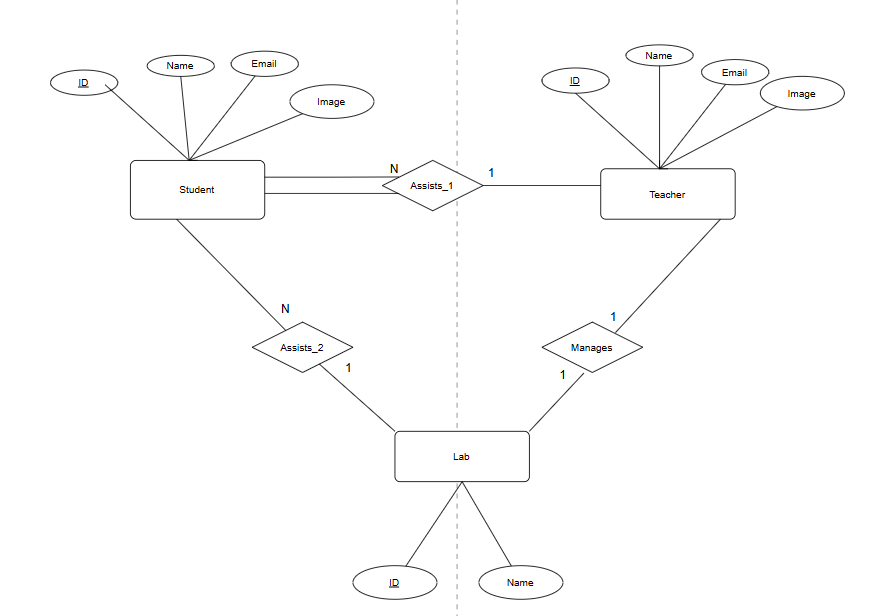
**

Figure 1.11: ER Diagram

# 

# 3. Implementation Plans

## 3.1 Technology Stack

3.1.1 Software Platform

• Visual Studio Code

• Git Bash

• MS Command Prompt

• Chrome

• Postman

3.1.2 Programming Languages

• JavaScript

3.1.3 Web Page Design

• HTML

• CSS

• Bootstrap

3.1.4 Managing Scripts

• Node.js

3.1.5 Frameworks

• Express

• React

3.1.6 Database Design

•MongoDB

## 3.2 Work Estimates

| **Description** | **Time Estimate**  **(Hours)** | **Team Members Involved** | **Date of Completion** |
| --- | --- | --- | --- |
| Backend | 50 | Thukkani Dinesh Reddy | 9-11-2024 |
| Frontend | 64 | Talari Bhanu Prakash  Gayas Uddin Ahmad | 15-11-2024 |
| Database | 10 | Mohammed Naseef | 18-11-2024 |

# 

# References

* Meyer, E. A. (2006). *CSS: The Definitive Guide: The Definitive Guide*. " O'Reilly Media, Inc.".
* Crockford, D. (2008). *JavaScript: The Good Parts: The Good Parts*. " O'Reilly Media, Inc.".
* Fedosejev, A. (2015). *React. js essentials*. Packt Publishing Ltd.
* Banker, K., Garrett, D., Bakkum, P., & Verch, S. (2016). *MongoDB in action: covers MongoDB version 3.0*. Simon and Schuster.
* Hahn, E. (2016). *Express in Action: Writing, building, and testing Node. js applications*. Simon and Schuster.
* Cantelon, M., Harter, M., Holowaychuk, T. J., & Rajlich, N. (2014). *Node. js in Action* (pp. 17-20). Greenwich: Manning.

## 

## 

## 