COLLEGE OF ENGINEERING PUNE

DEPARTMENT OF COMPUTER ENGINEERING

SOFTWARE ENGINEERING MINI PROJECT – II

Classroom Activity Manager

Submitted by

Group no – T4_23

Division-1

Group Members –

111903064 - PRITI KAMBLE

111903075 - RUTUJA MANGLE

Under the guidance of

Dr.Tanuja R.Pattanshetti,

Asst.Prof,Computer-IT Dept.

Classroom Activity Manager

Problem statement – We aim to develop a Classroom Activity Manager a student-teacher portal with various classroom features. The main focus is to develop a system that will improve Data management in classroom experience for both students and teachers. The whole system will run on internet.

Objectives–

- This Classroom Activity Manager is portal that provides various classroom features like uploading assignments, submissions, class notice and many more.
- Student and teacher can login the portal and experience interactive virtual classroom.
- Teacher can easily manage their students and keep them updated about all of the activities in class.

Motivation –

In the pandemic situation online learning has become the important part of our life. It adds flexibility to the traditional education system. In this tough times we came across various online learning platforms, the curiosity to get to know the internals of such platform made us select the topic of classroom activity manager.

Considering the impact of online learning in our life we tried to develop a Classroom Activity Manager that uses the multi-user concept of Django where student and teacher are different types of users and have different functionalities.

Summary of SRS –

1. Introduction –

The following subsections gives an overview of entire Software Requirements Specification (SRS).

1.1. Purpose –

This document details the Software Requirements Specifications for “Classroom Activity Manager". In the pandemic situation online learning has become the important part of our life. It adds flexibility to the traditional education system. Considering the impact of online learning in our life we tried to develop a Classroom Activity Manager that uses the multi-user concept of Django where student and teacher are different types of user and have different functionalities.

1.2. Scope –

As our web portal aims to provide online platform for classroom activities each student and teacher should have their account. Teacher should have access to all students accounts and their records. Student should have access of contact details of teachers.

1.3. References –

1. https://ieeexplore.ieee.org/document/6749474
2. https://ieeexplore.ieee.org/document/6601952

2. Overall Description –

This section describes the general factors that affect software and its requirements.In order to be easily understandable, this part of SRS provides a background for the requirements.

2.1. Product Perspective –

"Classroom Activity Manager" is a portal based on multiuser concept. Stakeholders for the system are teachers and students. In this portal teacher and student will first create account if not having account or simply login. The assignments and notice uploaded by teacher can be viewed by students. Likewise student's submissions can be viewed by teacher.

2.2. Product functions –

- Users firstly have to register either as student or teacher.
- Teachers have to register for the first so that they can share notices and assignments with students. Teachers can track submissions made by students. Once submission is made by student it can't be modified. The availability of list of all students will make it easy for teacher to review and grade assignments.

- For student user they can submit assignment, view their grades, view notices.

2.3. User Characteristics –

Teachers:

Signup as teacher - Then according to need they can upload assignments for students. Also they can view and grade those assignments.

Students:

Signup as student -. Then they can view assignments given by teacher. They can submit assignments but once submitted they can’t modify it.

2.4. Assumptions –

Every user will be having the appropriate hardware and software configurations as per the necessary requirement.

3. Specific Requirements –

3.1. Software interface –

Frontend – Django framework

Language – python, HTML, CSS

Backend – Sqlite3 database

3.3. Functional Requirements –

It will be requiring two interfaces:

A] For teachers:

Teacher can upload assignments to all students in their class and grade that assignments. Teacher can also edit the marks of students if necessary. Teacher can write notice for class. Teacher can see the list of all students in the class.

B] For students:

Students can download the assignments uploaded by teacher and submit the assignment but once submitted students cannot change it. Student can see the list of all teachers in the portal and can get contact details of any of them.Students can see class notice written by teacher.

3.4. Performance Requirements –

The performance of the software will be as smooth as possible with special consideration on the following parameters-

- Planned approach towards working
- Accuracy
- Reliability
- No Redundancy
- Easy to Operate

3.5. Software quality attributes –

Scalability: The application is scalable for large user data management.

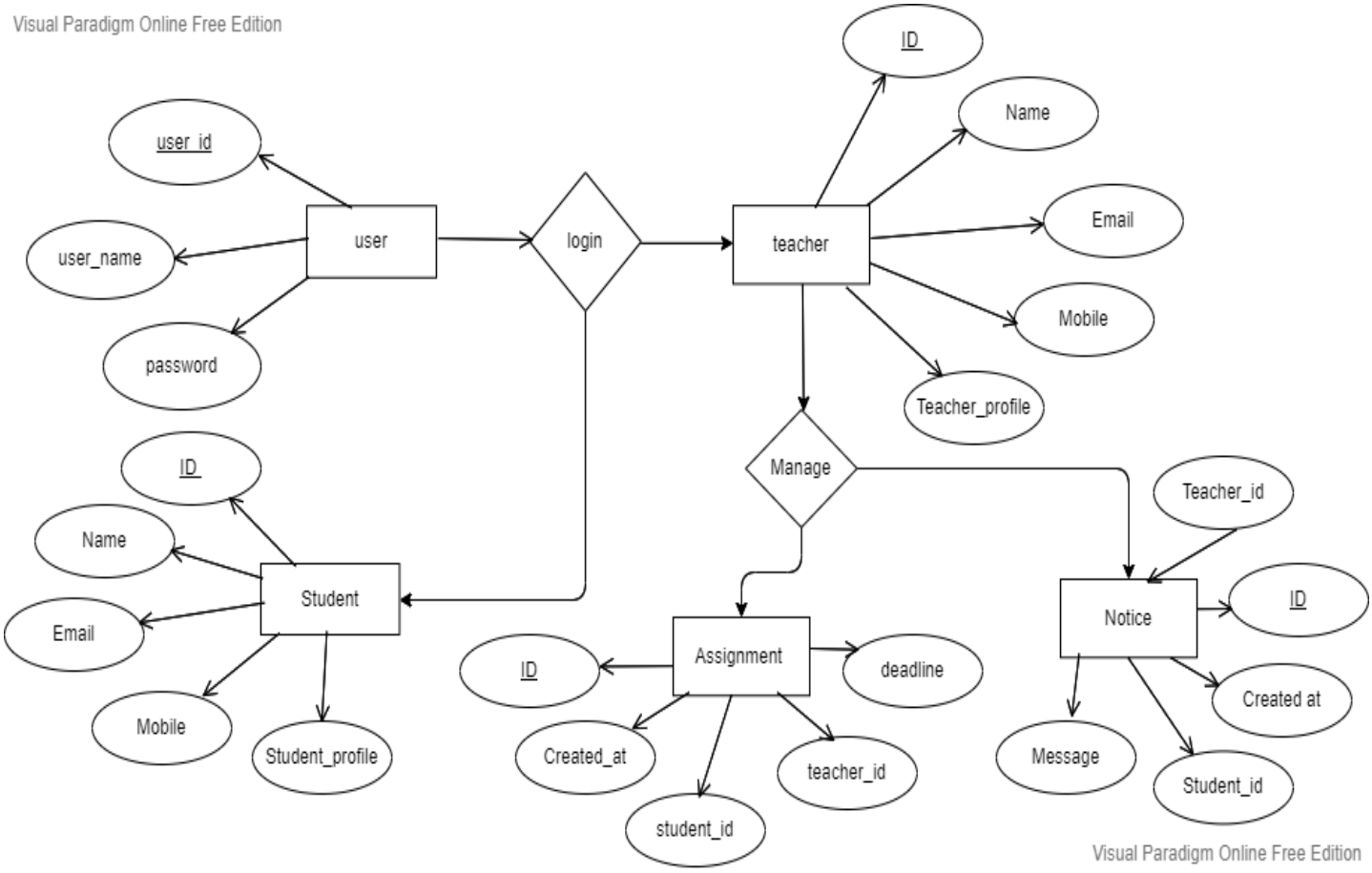
Reliability: The application is reliable for both teachers and students.

Usability: The teacher can grade students for their submissions of assignments. Student can submit solutions for assignments given to them.

ER diagram –

Entity relationship diagrams are used in software engineering during the planning stages of the software project. They help to identify different system elements and their relationships with each other.

The entities in the ER schema become tables, attributes and converted the database schema. Since they can be used to visualize database tables and their relationships it's commonly used for database troubleshooting as well.



UML diagrams –

A UML diagram is a partial graphical representation (view) of a model of a system under design, implementation, or already in existence.

UML specification defines two major kinds of UML diagram: structure diagrams and behavior diagrams.

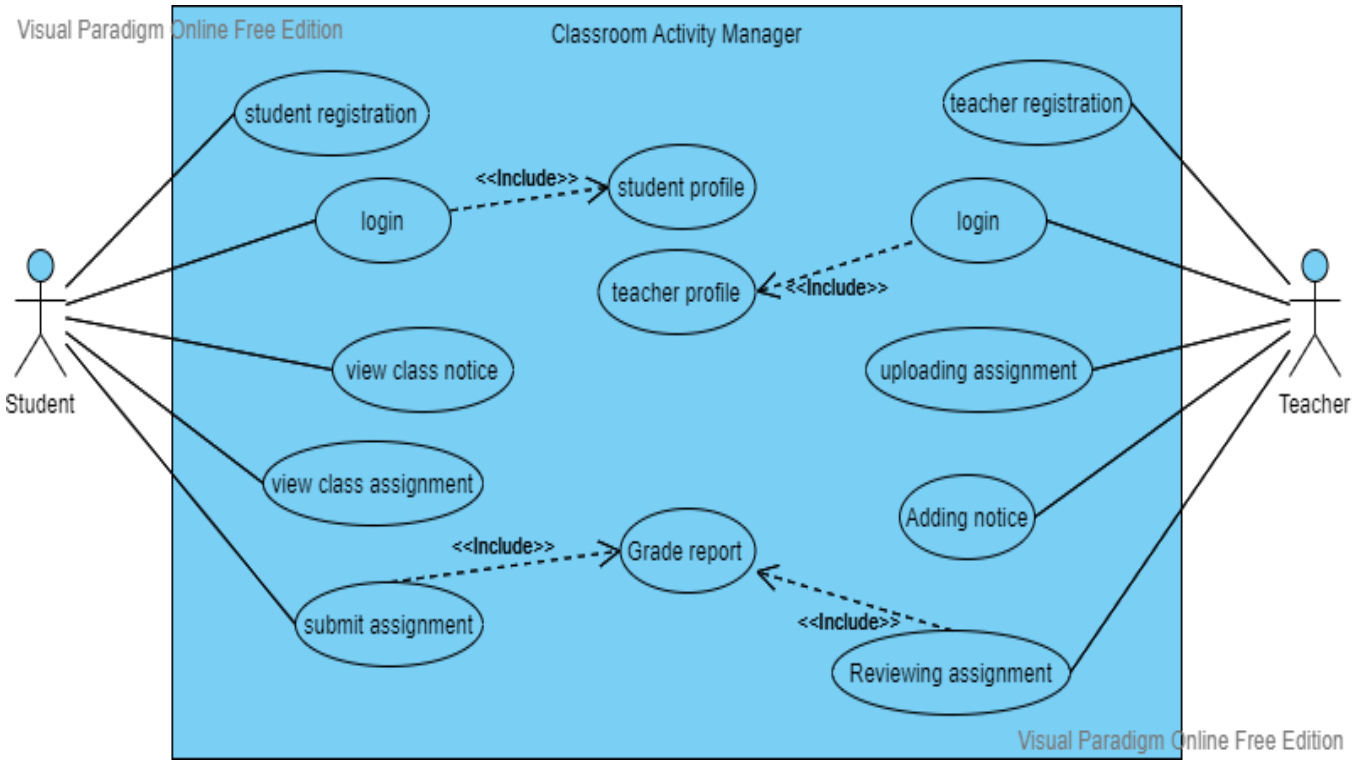
Structure diagrams show the static structure of the system and its parts on different abstraction and implementation levels and how they are related to each other. The elements in a structure diagram represent the meaningful concepts of a system, and may include abstract, real world and implementation concepts.

Behavior diagrams show the dynamic behavior of the objects in a system, which can be described as a series of changes to the system over time.

1. **Use case diagram –**

   In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

   - Scenarios in which your system or application interacts with people, organizations, or external systems.
   - Goals that your system or application helps those entities (known as actors) achieve.
   - The scope of your system.

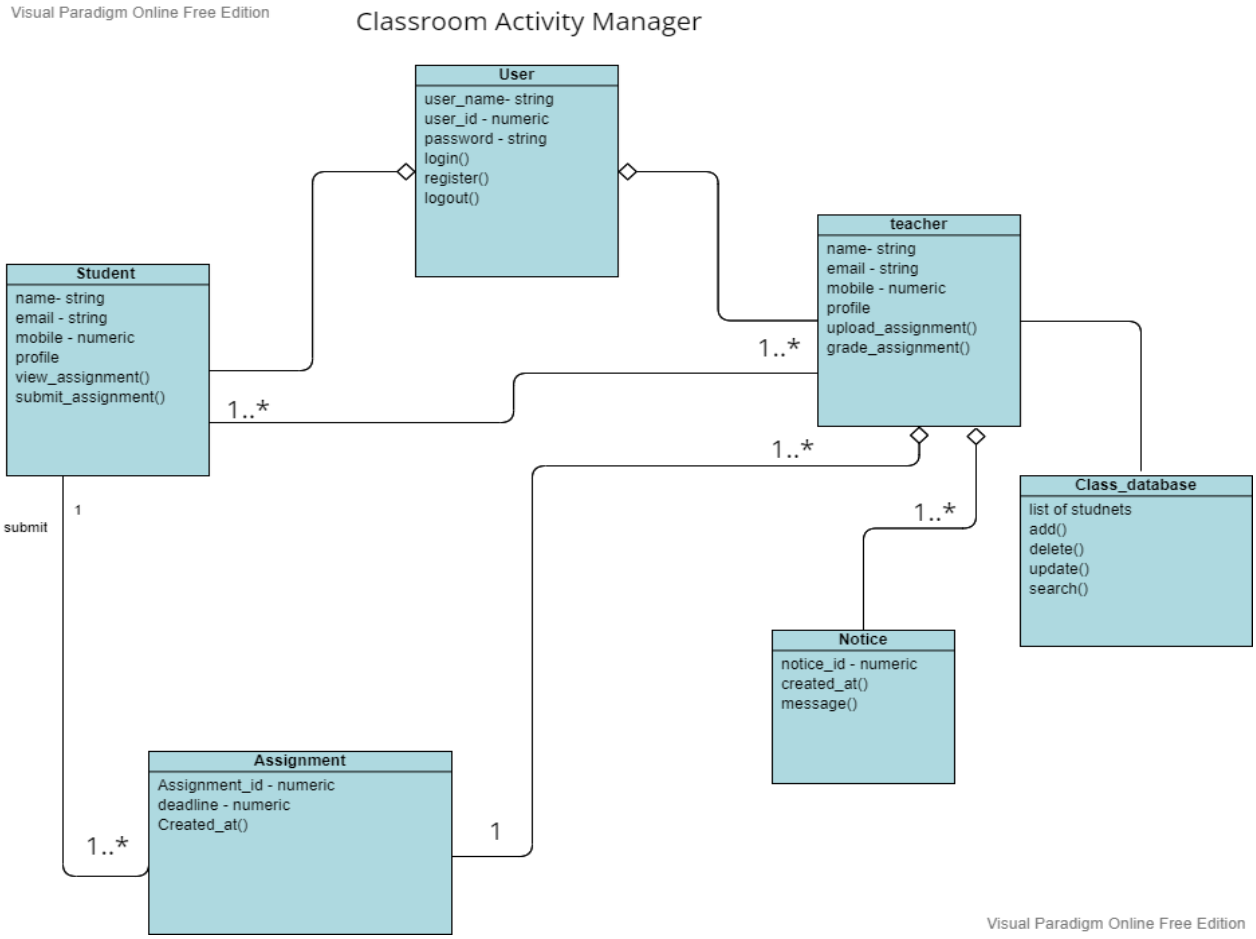


2. Class diagram —

The class diagram is the main building block of object oriented modeling. It is used for general conceptual modeling of the structure of the application and for detailed modeling,translating the models into programming code.

Classes in class diagrams are represented by boxes that are partitioned into three: The top partition contains the name of the class. The middle part contains the class's attributes.The bottom partition shows the possible operations that are associated with the class.

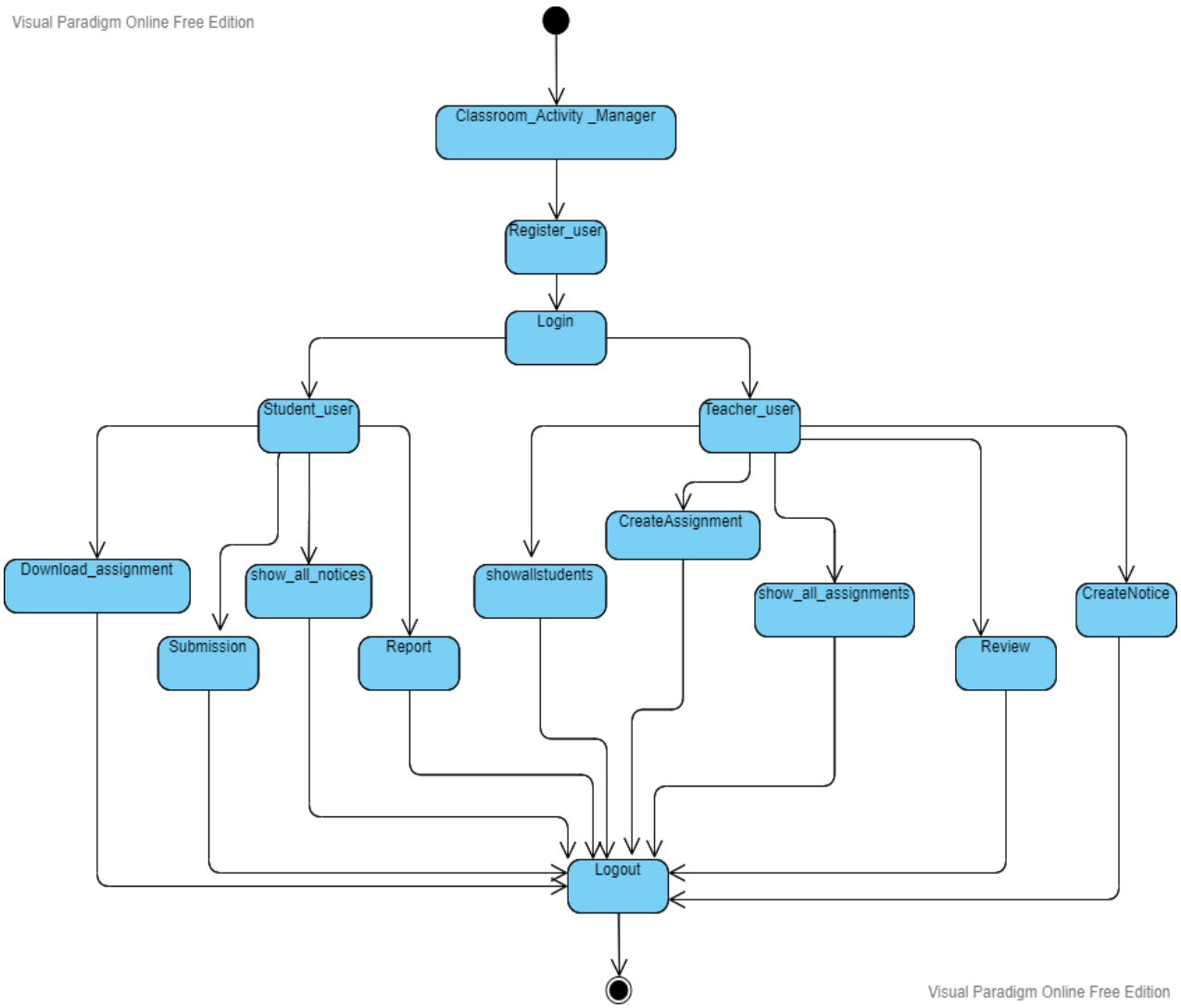
In the following class diagram the types of logical connections used are aggregation and multiplicity.



3. State diagram —

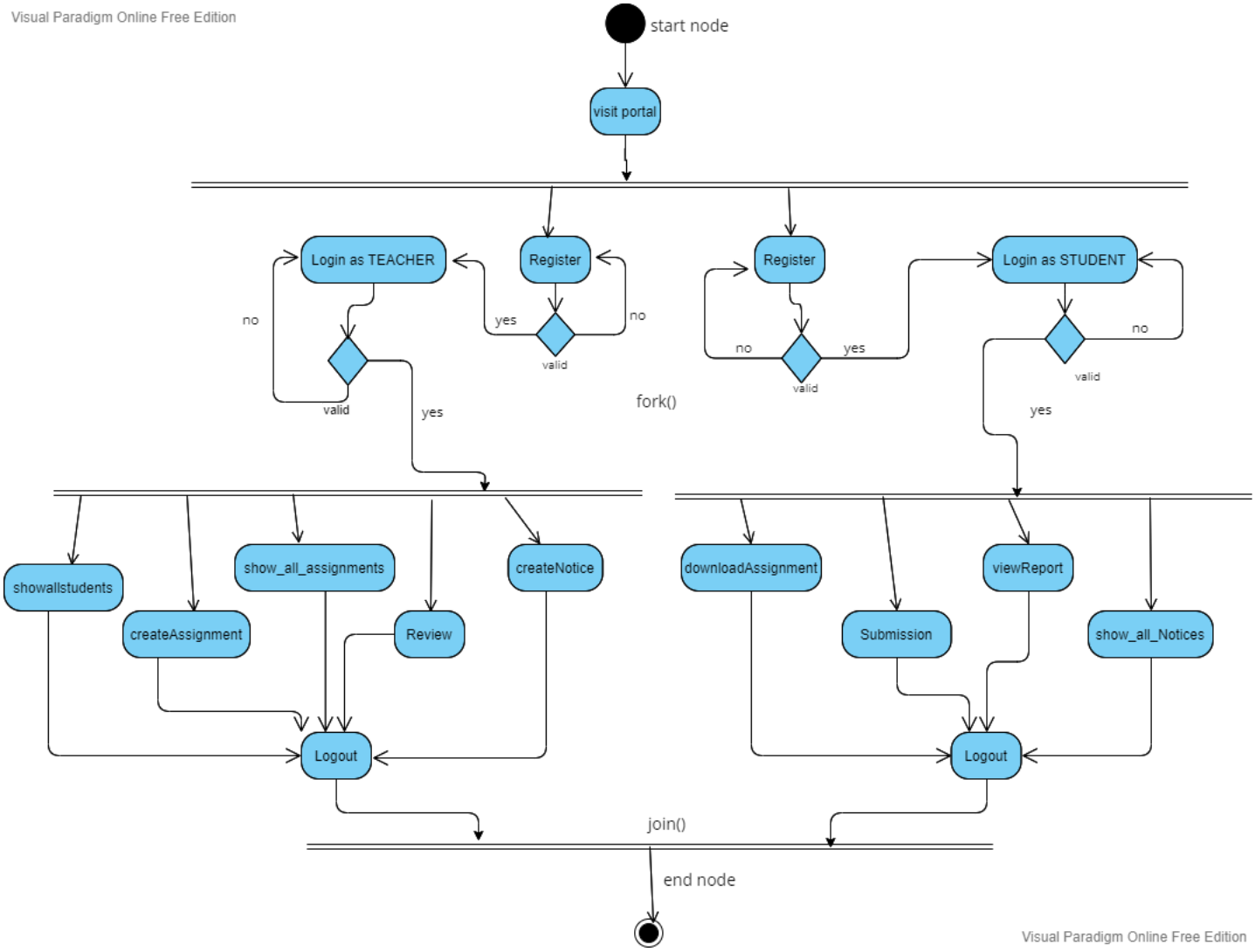
A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems.State diagrams require that the system described is composed of a finite number of states.

A state diagram consists of states, transitions, events, and activities.



4. Activity diagram

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed

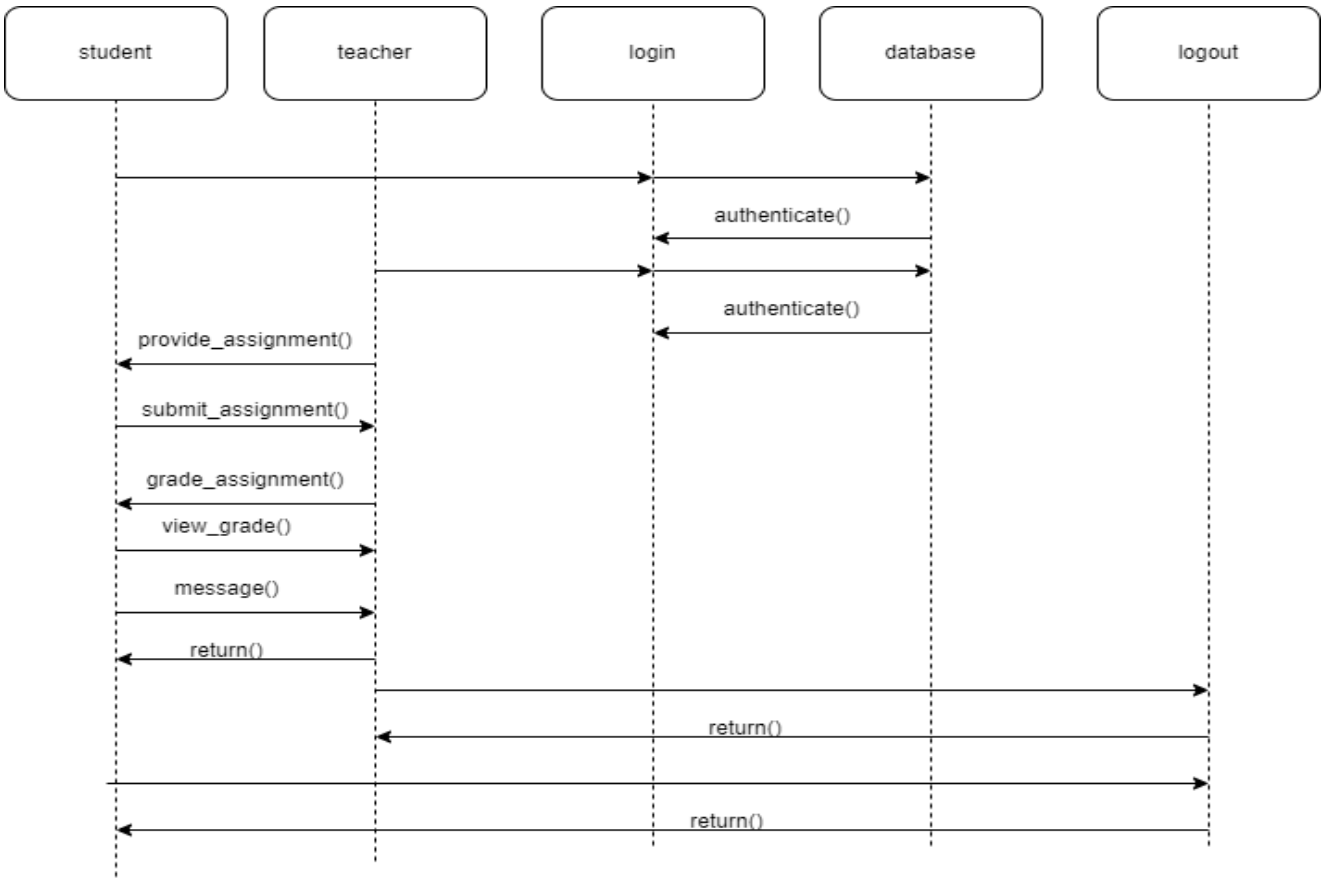


5. Sequence diagram –

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

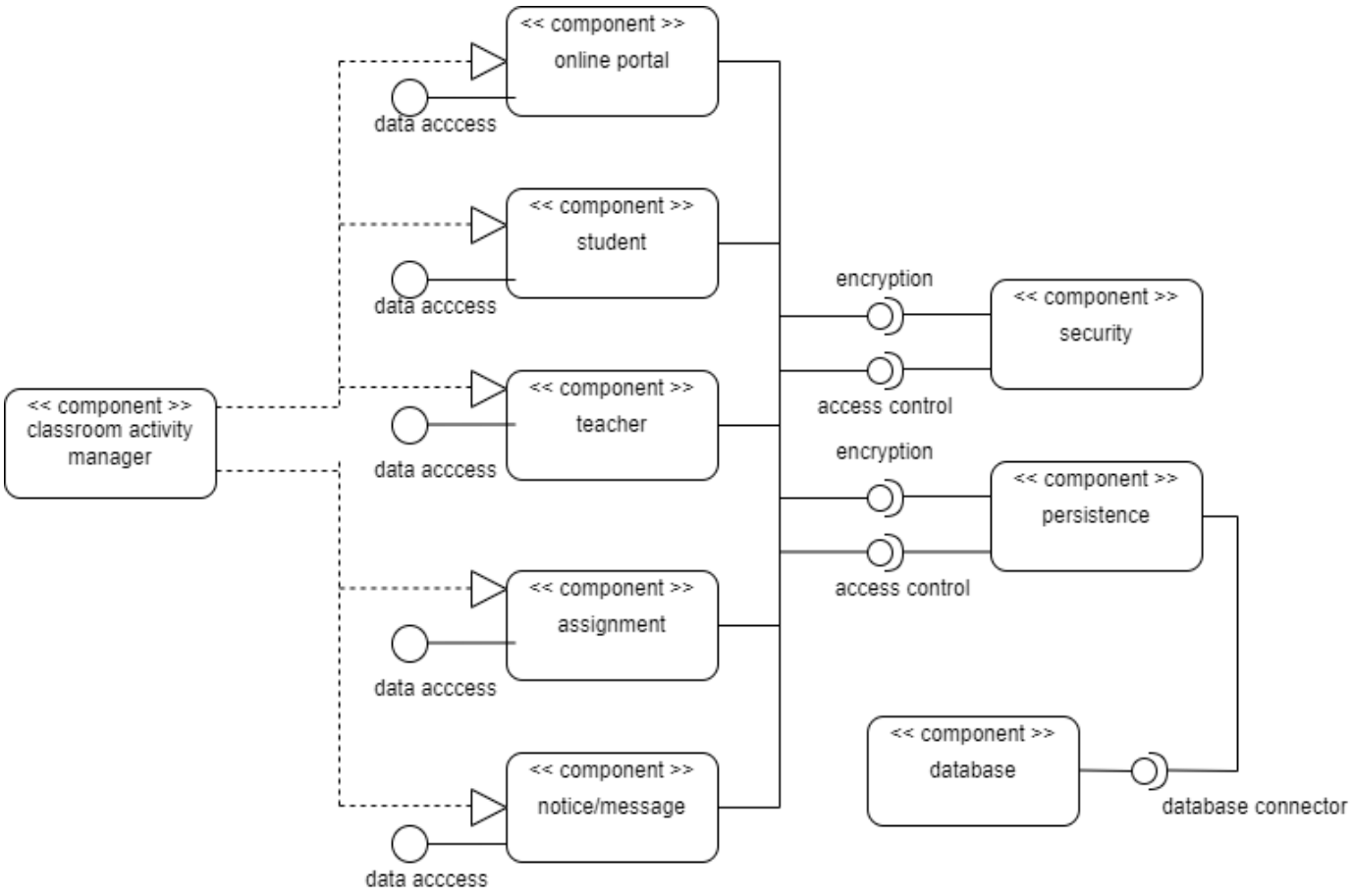
Sequence diagrams can be useful references for businesses and other organizations. Try drawing a sequence diagram to:

- Represent the details of a UML use case.
- Model the logic of a sophisticated procedure, function, or operation.
- See how objects and components interact with each other to complete a process.
- Plan and understand the detailed functionality of an existing or future scenario.



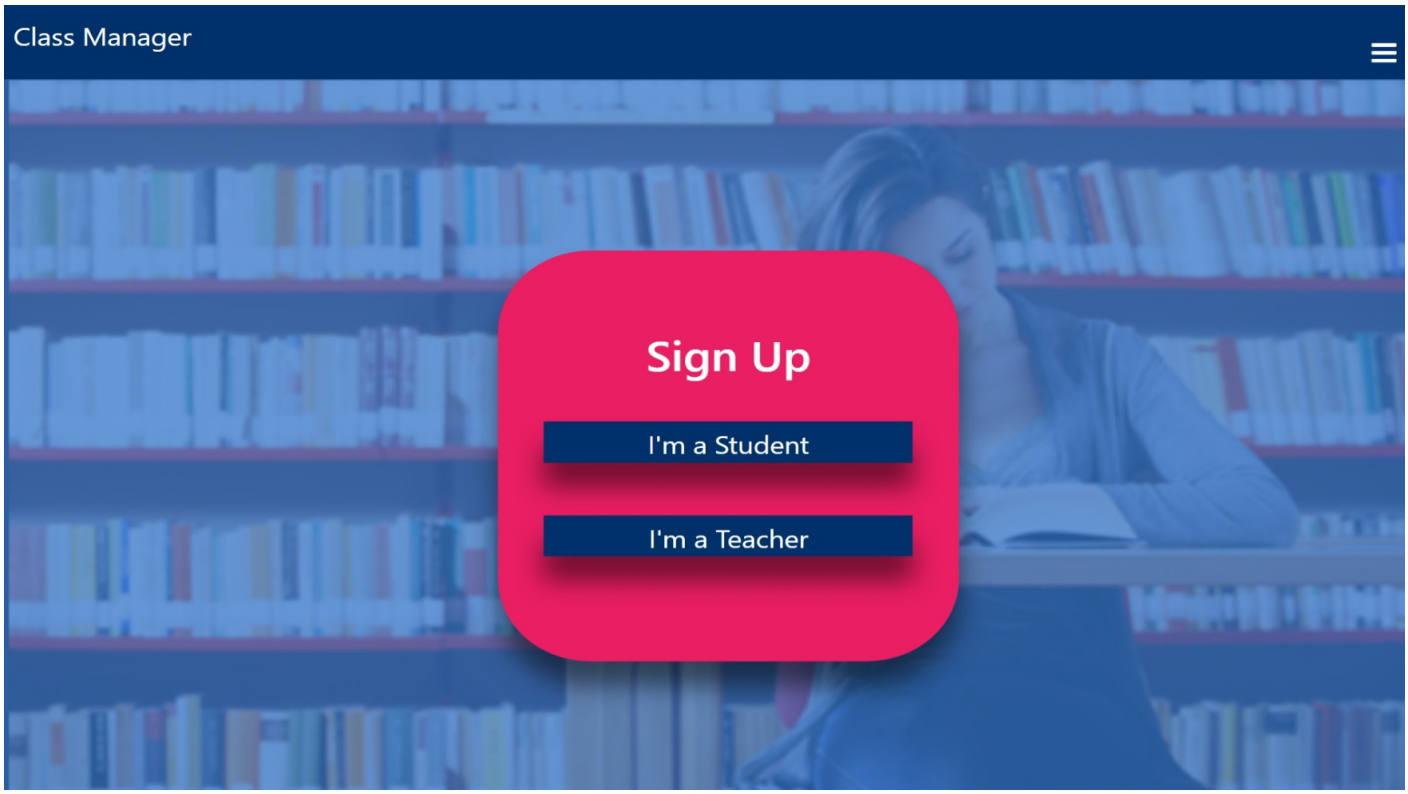
6. Component diagram –

Component diagram is a special kind of diagram in UML. Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

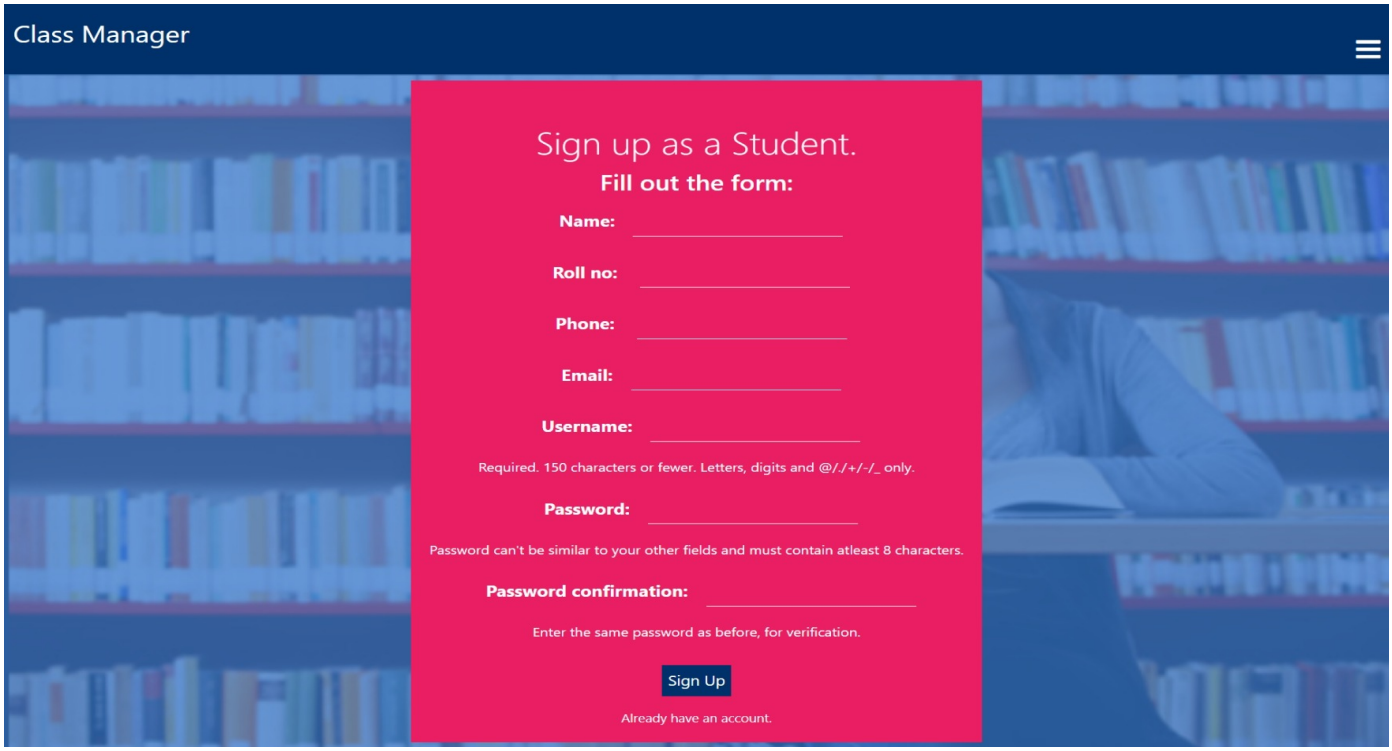


Code and Result –

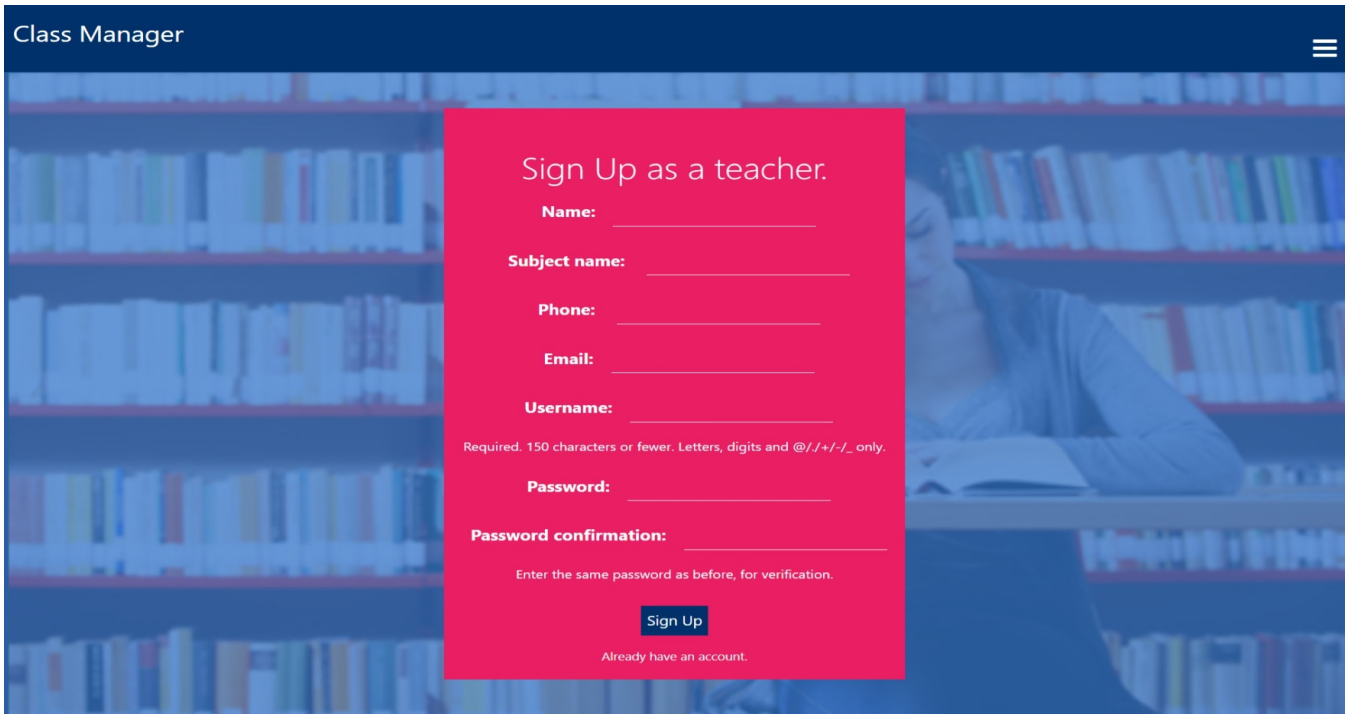
- signup page



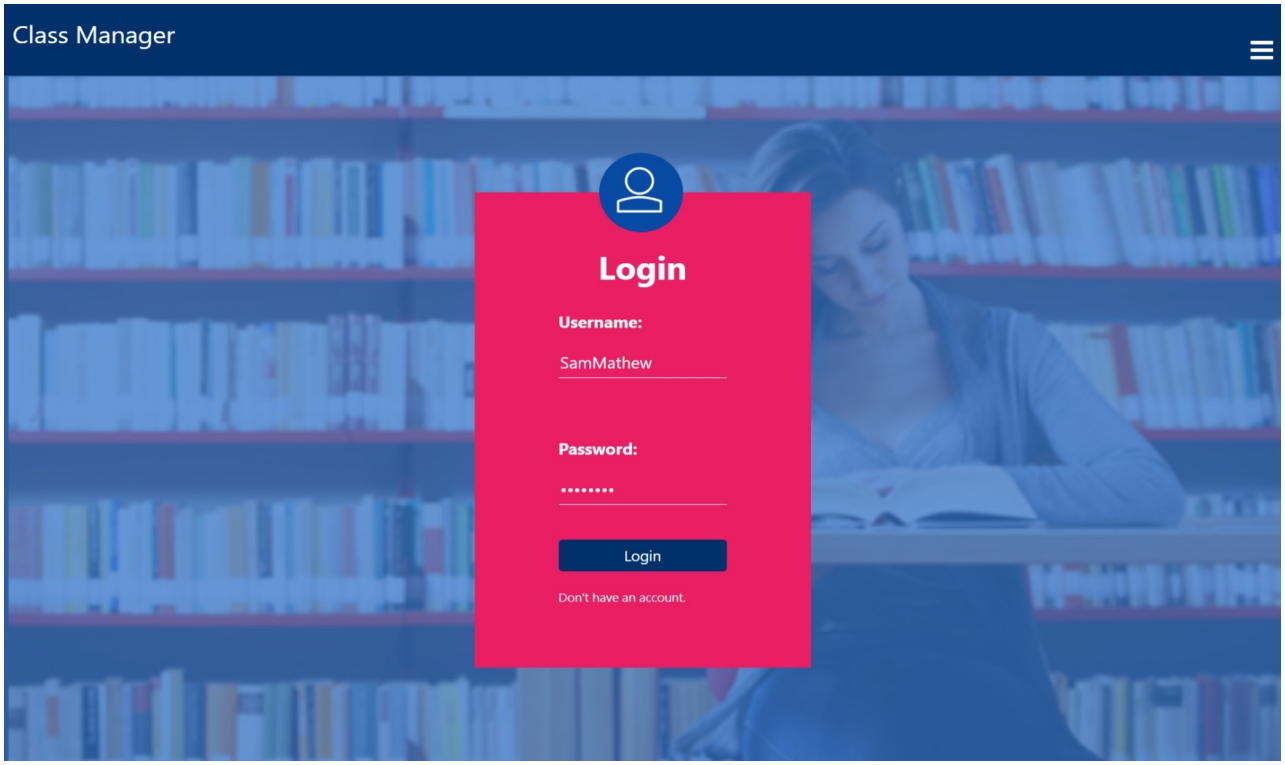
- Student signup



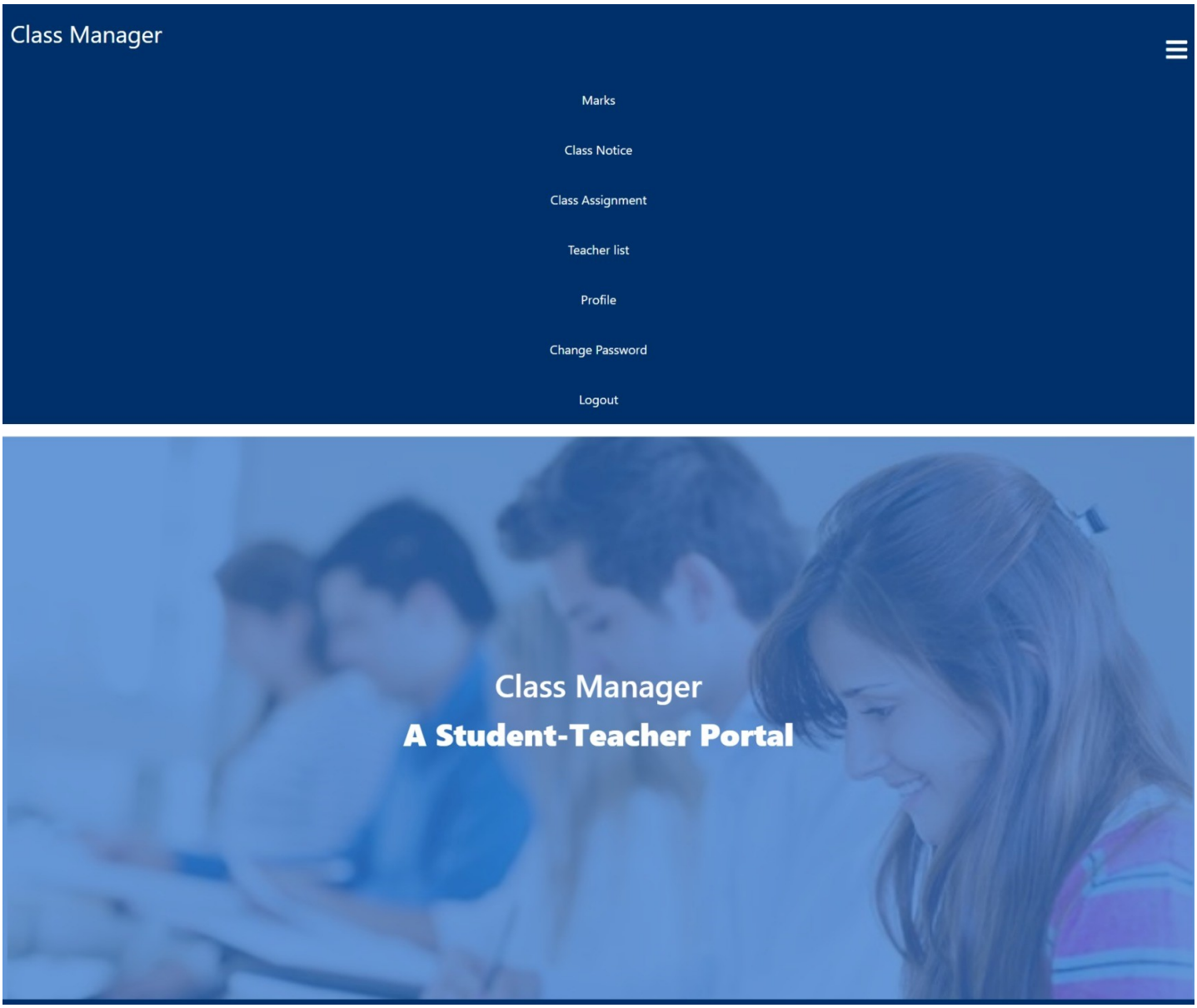
- Teacher signup



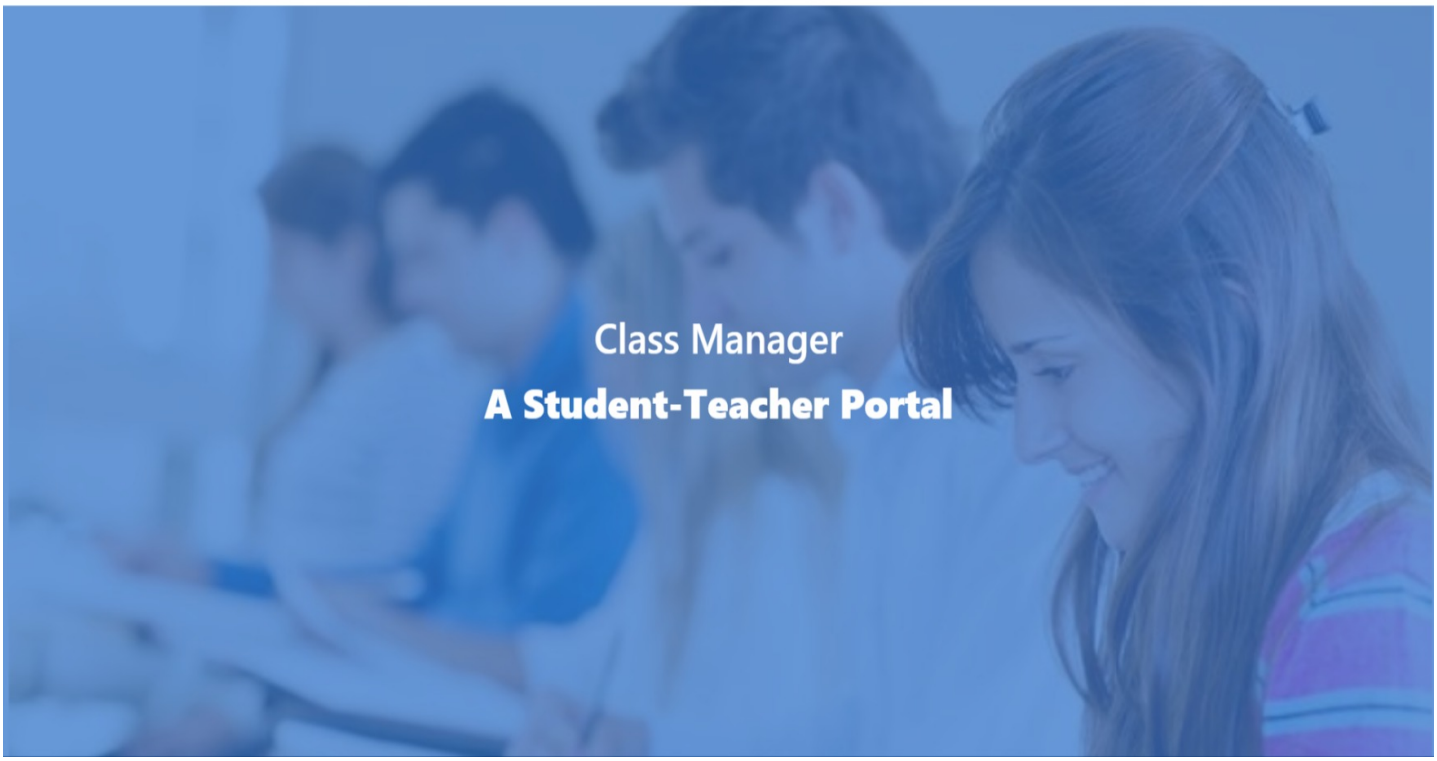
- Login page



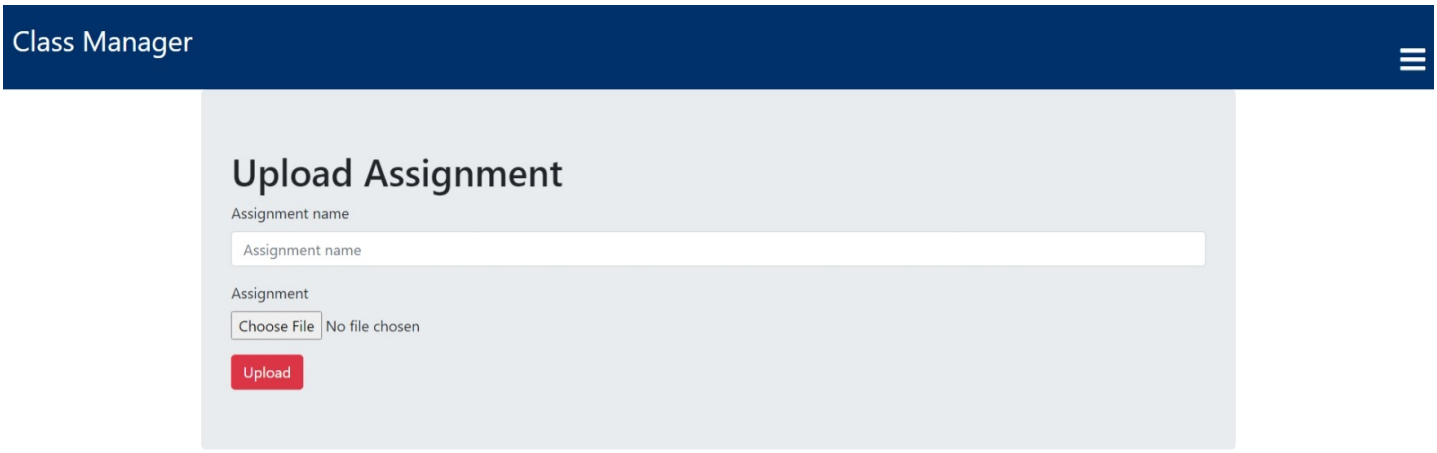
- Student Interface after login



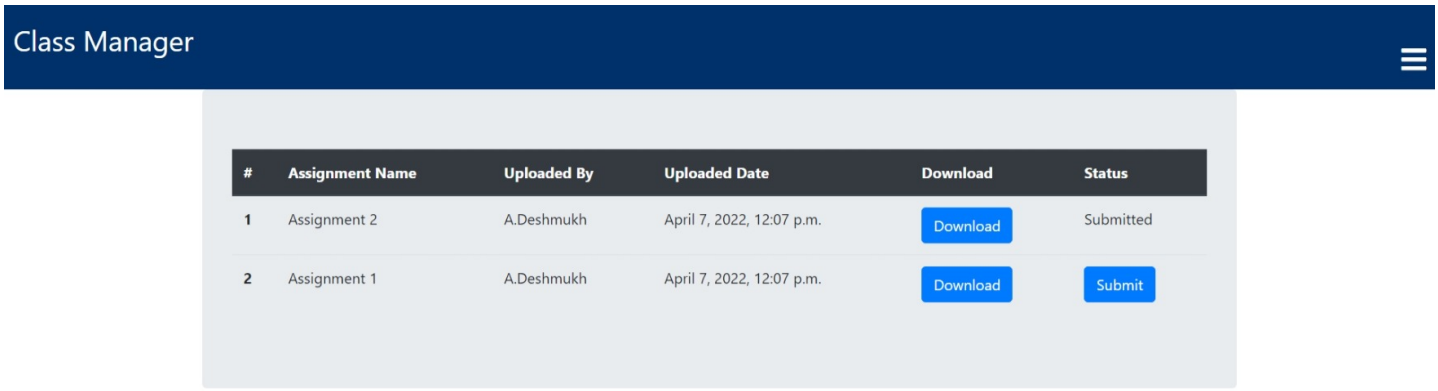
- Teacher interface after login



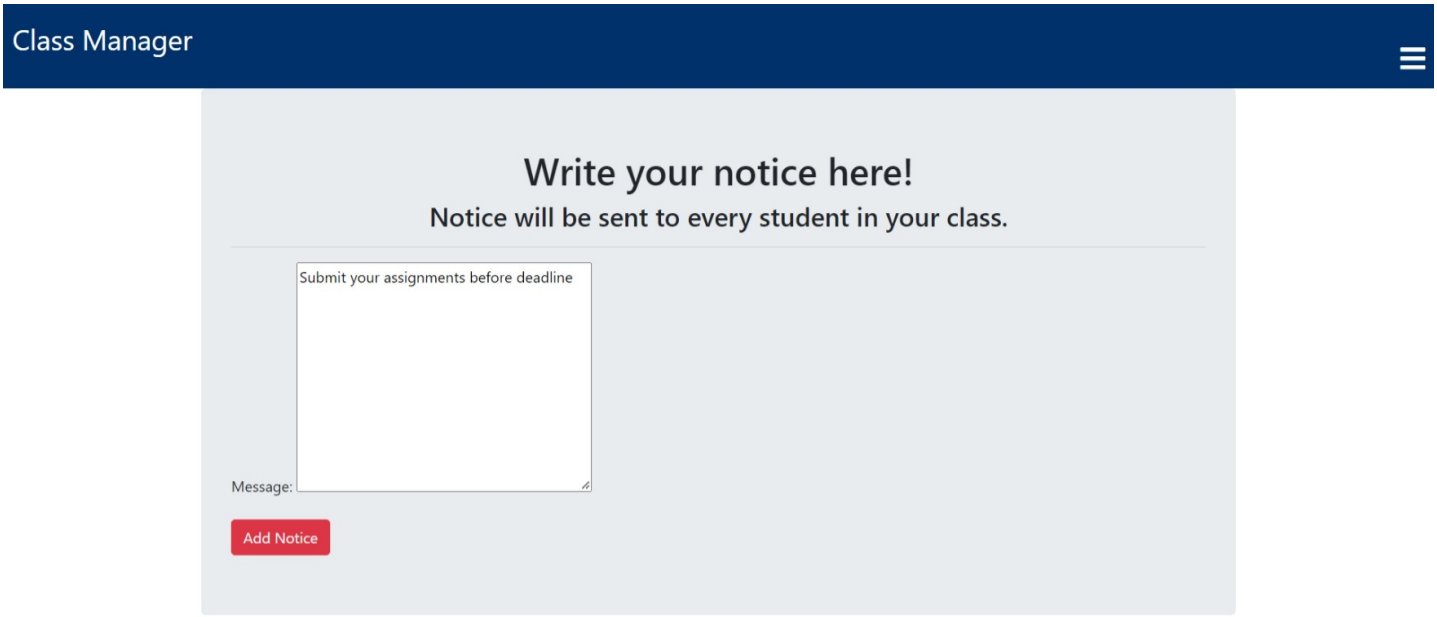
- Upload assignment



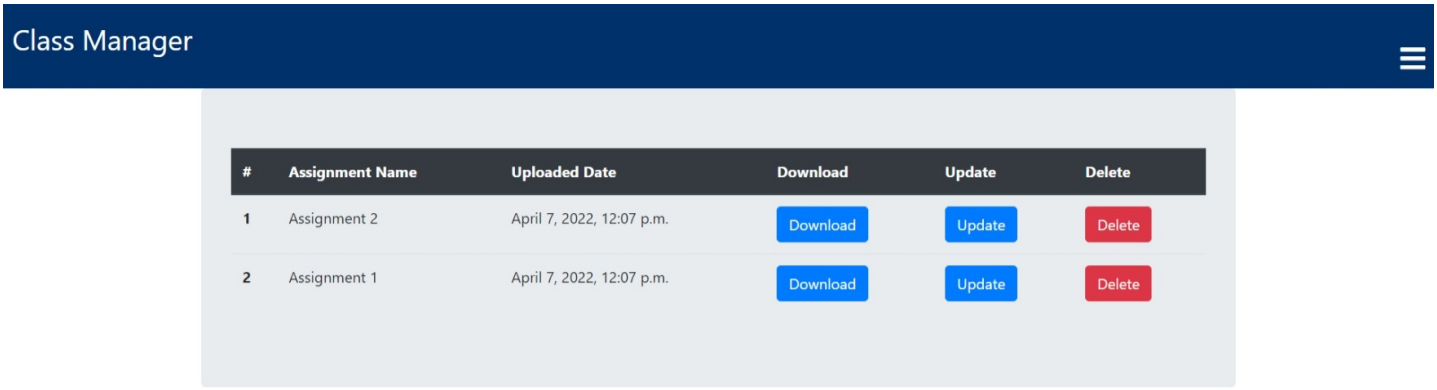
- Download and Submit assignment



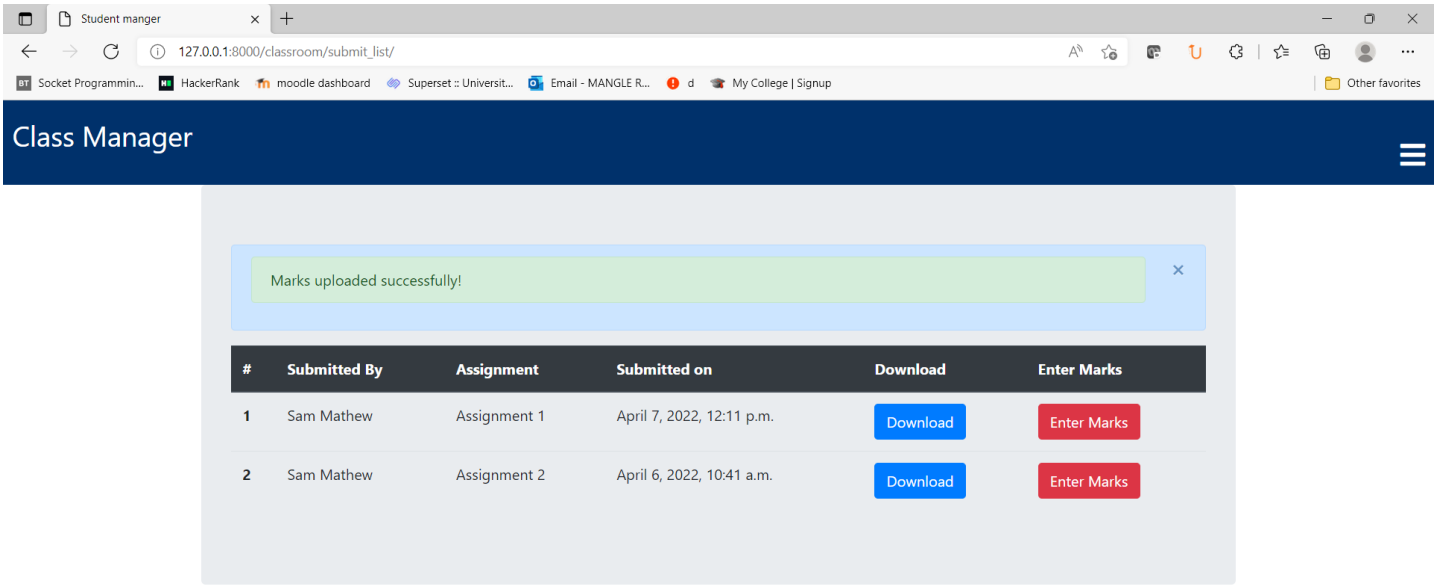
- Notice



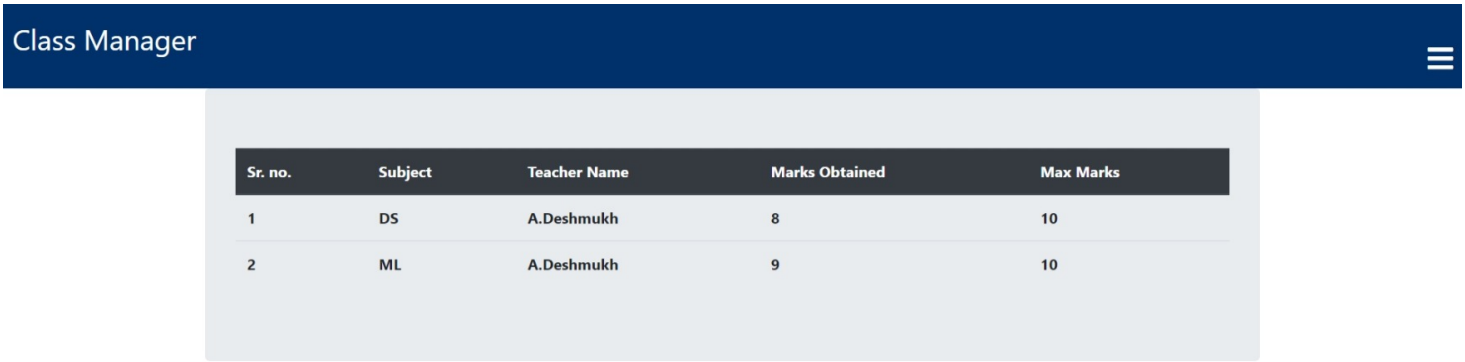
- Assignment list



- Grade assignment



- View grade



Testing –

Any project before exposed to user must be tested to ensure that it behaves as expected. In this project, the application is tested by giving various types of input to check whether they are being validated or not and whether the application is behaving as expected or not.

Conclusion –

Our project is only a venture to satisfy the needs of a classroom. Several user-friendly coding has also adopted. This package shall prove to be a powerful package in satisfying all the requirements of a class. The results obtained from the experiments and testing ensures that the proposed method is efficient and user-friendly. This project which yields centralized system that makes management easier.

Future Scope –

The project has a very vast scope in future. The project can be implemented on intranet in future. Project can be updated in near future as and when requirement for the same arises, as it is very flexible in terms of expansion.

The following are the future scope for the project.

- ➢ Bar code Reader based attendance system
- ➢ Individual Attendance system with photo using Student login

Github URL –

https://github.com/p736/classroom-activity-manager