# Project Design Document Version 1
Team 33

Project Title

Framework for Automated Student Assessment using NLP - Audio

Created By

Devisetti Sai Asrith, Geethika Thota, Sidhi Panda, P Pavan

Client

Arjun Rajasekar, RCTS

## Introduction:

A project design document is a comprehensive plan that outlines the objectives, scope, resources, and strategies for a project. The document serves as a guide for the project team and provides a reference point throughout the project's life cycle, ensuring that the project stays on track and focused on its original objectives.

The project design document typically provides the following clarifications:

- **Project Goals:** Outlines the specific goals and objectives of the project and how they will be achieved.
- **Project Scope:** Defines the boundaries of the project and what is included and excluded.
- **Resources:** Identifies the resources needed to complete the project, including personnel, equipment, and materials.
- **Evaluation and Reporting:** Outlines how the project's success will be measured and reported.

It provides a blueprint for the project team to follow and serves as a reference point to ensure that the project remains focused on its original objectives. Ultimately, a well-designed project design document can increase the likelihood of project success by ensuring that everyone involved in the project is aligned and working towards the same goals.

# System Overview

The Application GradeMate is an automated student assessment app, using the audio as the mode of submission. As the submission mode is audio, this application can be used by non-tech students also. The system aims to provide a more efficient and objective method of evaluating student performance than traditional methods like manual grading and assessment.

The application has a user-friendly interface that guides the user with all the required guidelines. The system contains several components working together, to analyze the student submission. The application records the answer from the student with an inbuilt microphone.

Later we obtain the text version of the audio file submitted and then analyze the correctness of the answer. As a teacher, you have the access to the question-and-answer database, they can post new questions or update the answers for the current questions.

# Design Overview

## Architectural Overview:

The architectural view can be explained using the subsystems or modules involved in the system.

a. The User Authentication:
   - Initial page contains the information regarding the user for login.
   - Users can log in as a student or as a teacher.
   - If a user is using the application for the first time, they can register and then log in.

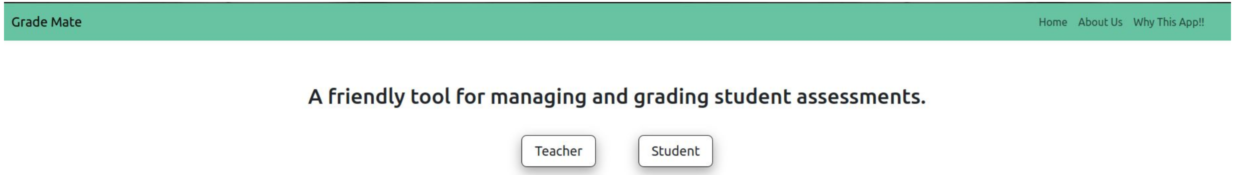
b. The Student Dashboard:
   - The student dashboard has the option to take a test or view the previous questions.
   - If a student takes a test, there will be a dropdown listing all the questions and there will be a option of ending the test.
   - The questions are clickable, and clicking takes the user to the submission page, where the user can record the answer using the inbuilt microphone or return back without submission.

c. The Teacher Dashboard:
   - The teacher dashboard contains an option to update the answers for the existing questions and to post new questions.
   - Even after the teacher selects one of the options, the user can return back without making changes.

# System Interface

## User Interface

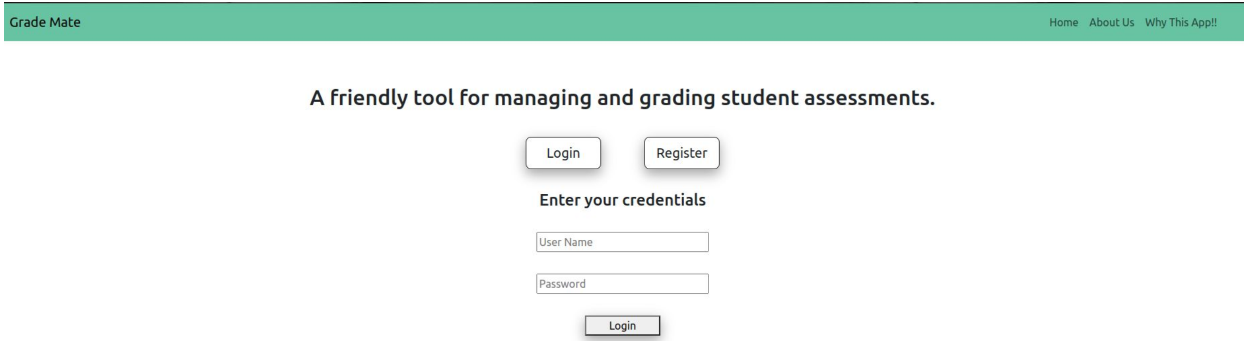
Initial Dashboard:



A friendly tool for managing and grading student assessments.

The initial dashboard contains requests the user information whether the user is a student or a teacher.
On selection of one of the options, the use will be redirected to the login/registration page

# Login/Registrations Page:

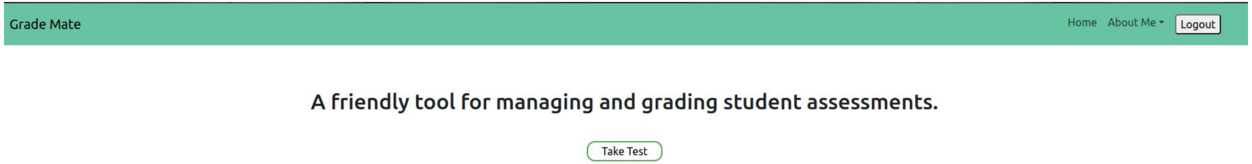


A friendly tool for managing and grading student assessments.

Enter your credentials

On selection of one of the options, the login/registrations page of the corresponding type will be displayed. If the user is new to the application, he can register, and after registration, user will be directed back to login page.

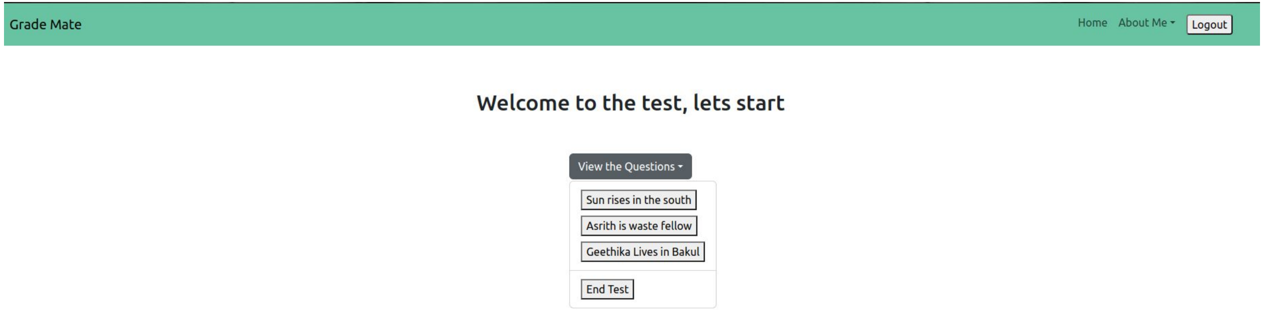
# The Student Dashboard:



A friendly tool for managing and grading student assessments.

If the user is a student, the student dashboard is displayed. There are options to take the test and in the dropdown there are options to view the profile of students and the past questions the student answered. After logging in there is always an option to logout.

## The Test page:



Welcome to the test, lets start

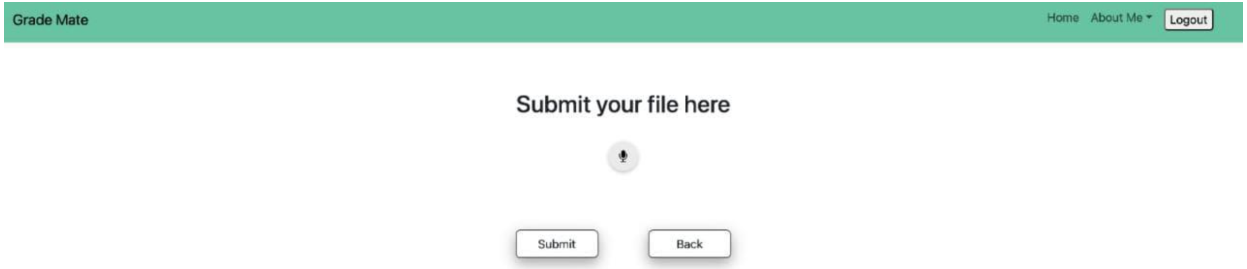
Sun rises in the south

Asrith is waste fellow

Geethika Lives in Bakul

After the student, starts the test, there will be a dropdown for viewing the question. The questions are self clickable which redirects to the submission page corresponding to the question. There is an option of ending the test in the last.

The Submission page:


Grade Mate

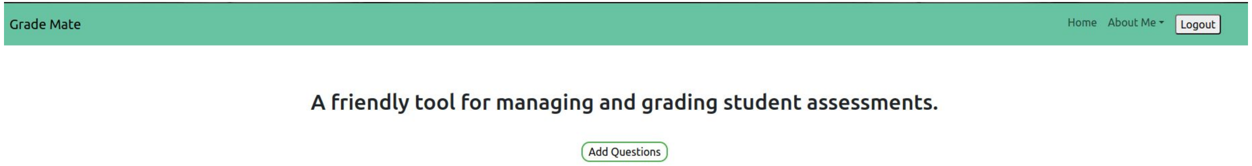
Home About Me Logout

Submit your file here

Submit Back

There is an inbuilt microphone, the student need to press the button and need to record the answer and need to submit. The audio file is then pushed into the backend and later converted into text using external NLP modules(Whisper in this project).

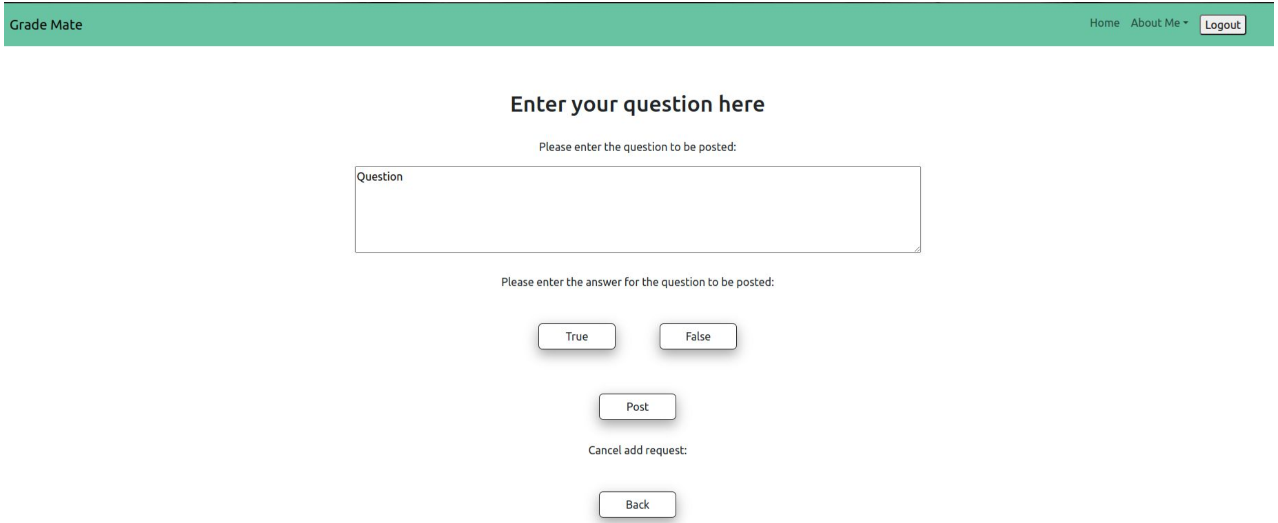
# The Teacher Dashboard:



A friendly tool for managing and grading student assessments.

If the user is a Teacher, the teacher dashboard is displayed. There are options to add a question and in the dropdown there are options to view the profile of teacher and update the answers of the previously posted questions. After logging in there is always an option to logout.

# The Post Questions Page:



Enter your question here

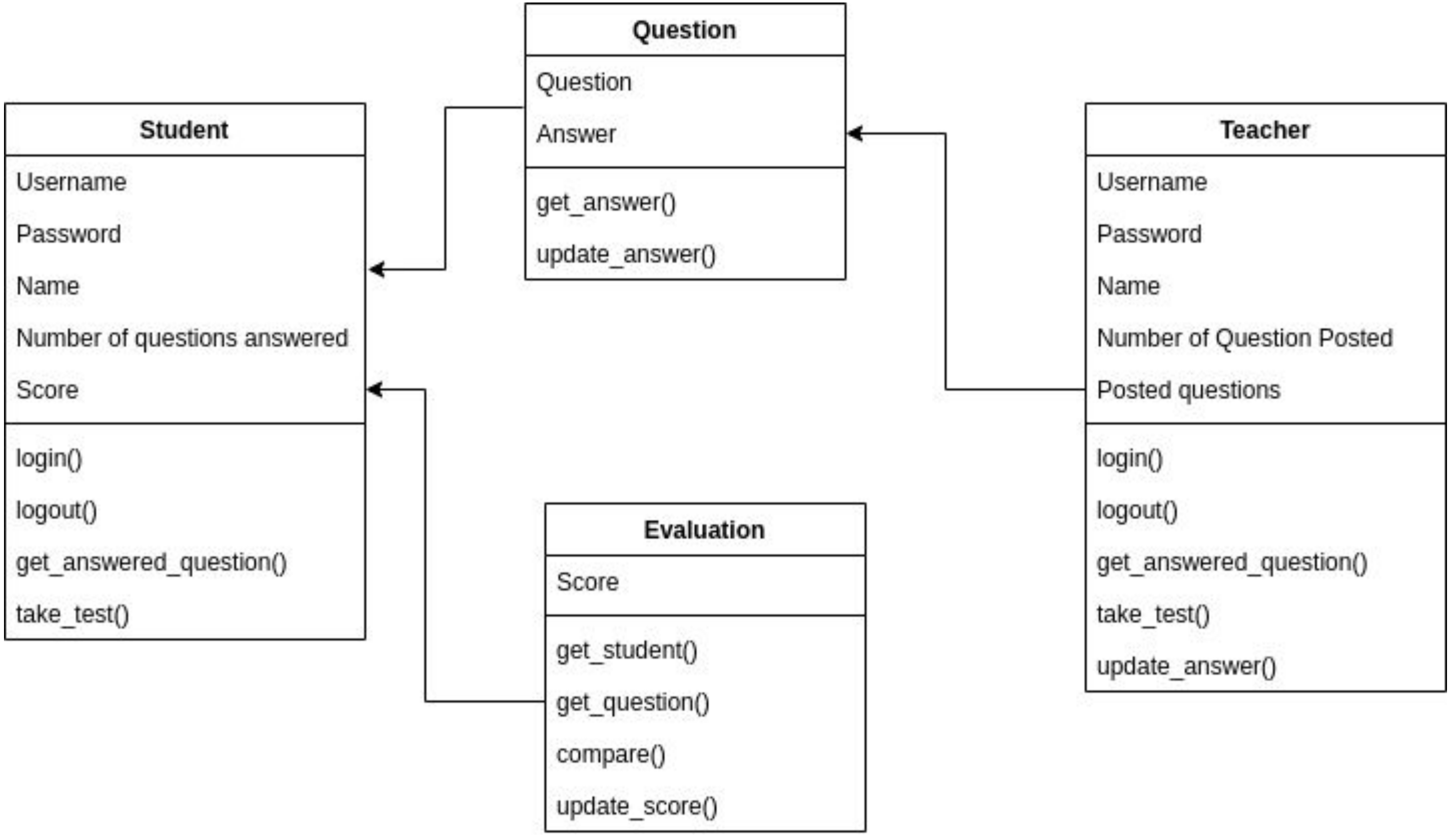
Please enter the question to be posted:

Please enter the answer for the question to be posted:

Cancel add request:

The question can be posted into the database, by entering the question in the place provided. The answer for the questions can be set as true or false and the question can be posted or can return back without posting.

## Model:



**Question**

Question

Answer

get_answer()

update_answer()

**Student**

Username

Password

Name

Number of questions answered

Score

login()

logout()

get_answered_question()

take_test()

**Teacher**

Username

Password

Name

Number of Question Posted

Posted questions

login()

logout()

get_answered_question()

take_test()

update_answer()

**Evaluation**

Score

get_student()

get_question()

compare()

update_score()

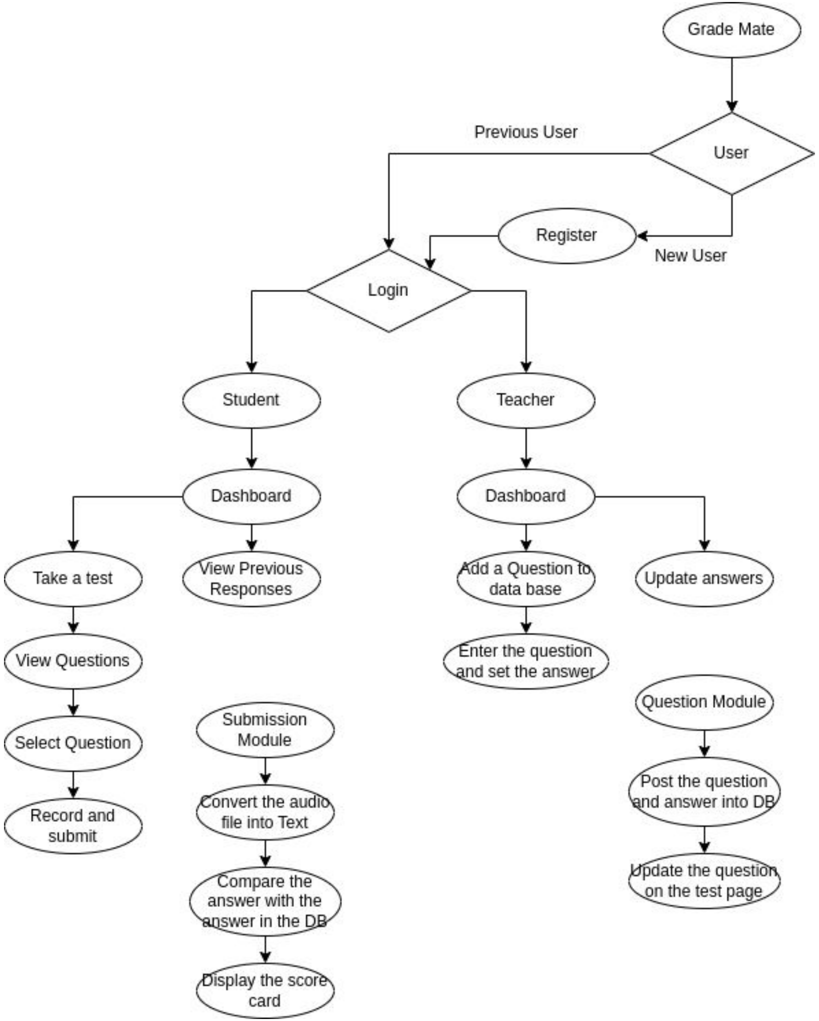
| | |
|---|---|
| Student | Class state:<br>● UserName<br>● Password<br>● Name<br>● Number of questions answered<br>● Score<br>Class behaviour:<br>● login()<br>● logout()<br>● get_answered_questions()<br>● take_test() |
| Teacher | Class state:<br>● UserName<br>● Password<br>● Name<br>● Number of questions posted<br>● Posted Questions<br>Class behaviour:<br>● login()<br>● logout()<br>● get_posted_questions()<br>● update_answer() |

| | |
|---|---|
| Question | Class state:<br>● Question<br>● Answer<br>Class behaviour:<br>● get_answer()<br>● update_answer() |
| Evaluation | Class state:<br>● Score<br>Class behaviour:<br>● get_student()<br>● get_question()<br>● compare()<br>● update_score() |

## APIs:

We are done building the backend for the gradmate application, yet we need to integrate it with the user interface, an API wrapper was defined to link the external ML model to the backend. This should be further explained in the version2 doc as we still need more.

## Sequence Diagram:



## Design Rationale:

Initially we were on search for the optimum NLP model that can be used for the project, later we installed the model successfully and implemented it to work ie, to convert an audio file into text.
We built the UI for the application, the client was not that particular about the interface unless it was clear and self evident for the user.
Now, we need to compare the text file with correct answer, so we need to find a model that does it.
We are done with it.
The further updates that need to be made in the next version:

- The API wrappers must be updated.
- The user interface must be updated with all the extra buttons functioning.