DALHOUSIE UNIVERSITY

Faculty of Computer Science

CSCI 5308 Advanced Topics in Software Development

AURA (E-Learning Suite)

Project Proposal Document

GROUP 15

Rishi Vimal Vasa (B00902815)

Utsav Vinod Singh (B00923487)

Abhijith Karuthedath Abraham (B00932476)

Anuj Dawar (B00927005)

Table of Contents

AURA (E-Learning Suite)

PROBLEM STATEMENT

The Covid-19 pandemic changed the way classes are conducted in universities. Professors recorded their lectures from home, and students viewed them on their own time.

The concept of flipped classrooms picked up as universities reverted to in-person lessons after the pandemic. In a flipped classroom, students gain first exposure to new material outside of class via reading or recorded lectures, and then use class time to assimilate that knowledge, through live problem-solving and discussions. [1]

Thus, the demand for feature-extensive, easy-to-use E-learning applications has increased significantly. Students and professors require one single interface to keep track of all the activities, assessments, deadlines, and content of the classroom. AURA aims to be one-stop solution for all such needs.

PROJECT OVERVIEW

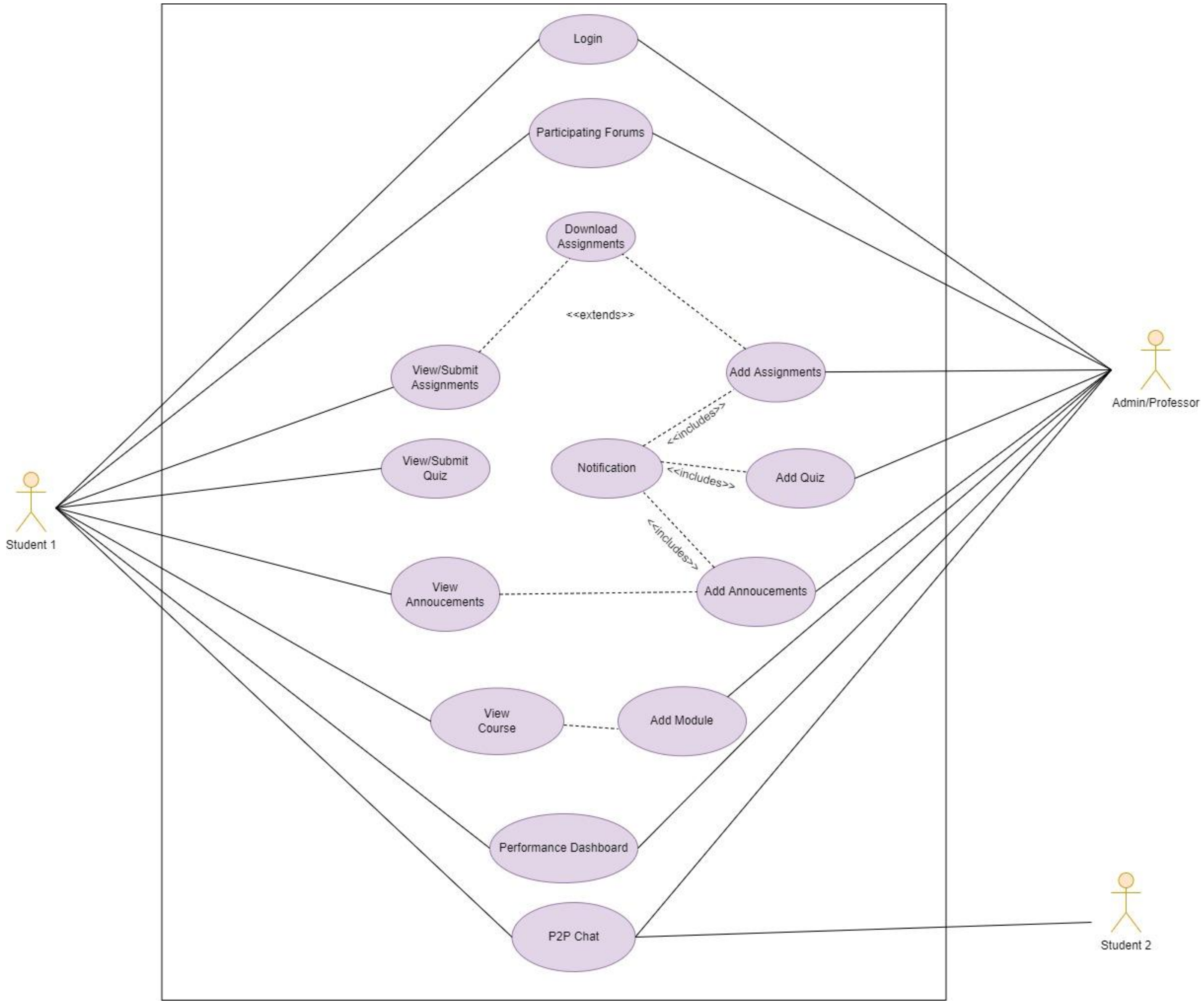
Aura is a one-stop solution to the increasing demand for E-Learning Systems. Aura provides an amalgamation of multiple features which are currently being provided by separate service providers. Universities usually must buy licenses for separate products, and students must monitor many applications to be updated with all information.

The project implements various features in one webapp to form a comprehensive solution such as a Learning Management System, Person to person chat system, Quizzing and assignment module, Announcement tracking, Timelines and Scheduling, etc. Aura simplifies online learning by providing modernized tools to significantly improve the student experience. Students must monitor just one webapp to be updated of all activities going on in the classroom. University professors also get a platform to digitize their course content to optimise learning outcomes.

WORKFLOWS AND USE CASES [2]

There are two main workflows which are included:

1) Professors/Admin
2) Students



FEATURES

Professor:

1) Register and add student Details
2) Create a Course Classroom
3) Create schedule for the course
4) Define a table of contents for each of the course modules
5) Add modules for the course
6) Upload lecture recordings, presentations, documents for students to refer
7) Create quizzes and assignments
8) Post Announcements to the classroom
9) Reply in forums and FAQs
10) View student progress and classroom metrics

Student:

1) View and download the course module content
2) Attempt Quizzes and receive grades
3) Upload Assignment submissions
4) View Grades and performance metrics
5) View class schedule and events

Common:

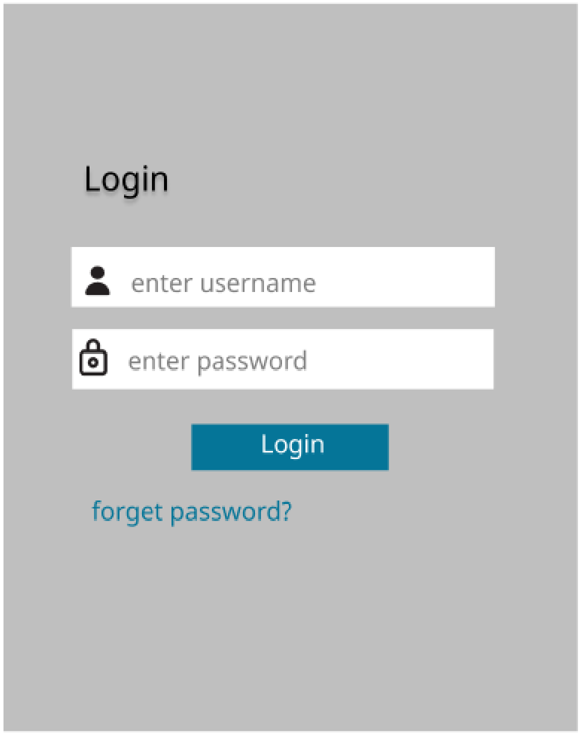
1) Login and Authentication
2) Discussion Forums
3) Receiving events/deadline notifications
4) Shared Calendar
5) Person to Person Direct Chat

EXPECTED USER INTERFACE MOCKUPS [3]

Mockup: Login Module

- Students / Admin can log in using their username and password.
- The Interface to be displayed will be role-based.
- In case the user or admin forgets their password, they can reset it using the forget password interface.

Username and password will be validated from the database. If the details match, the user will be logged in to the system.



Expected Behavior:

When the user clicks on login the system will check for the user profile(admin/student) then validate and grant access based on that.

Error Handling Behavior:

If the username and password combination is incorrect, an error message will be displayed with the error details and the user has a chance to retry.

Mockup: Student Portal

After logging into the system, a student has access to interact with the following features:

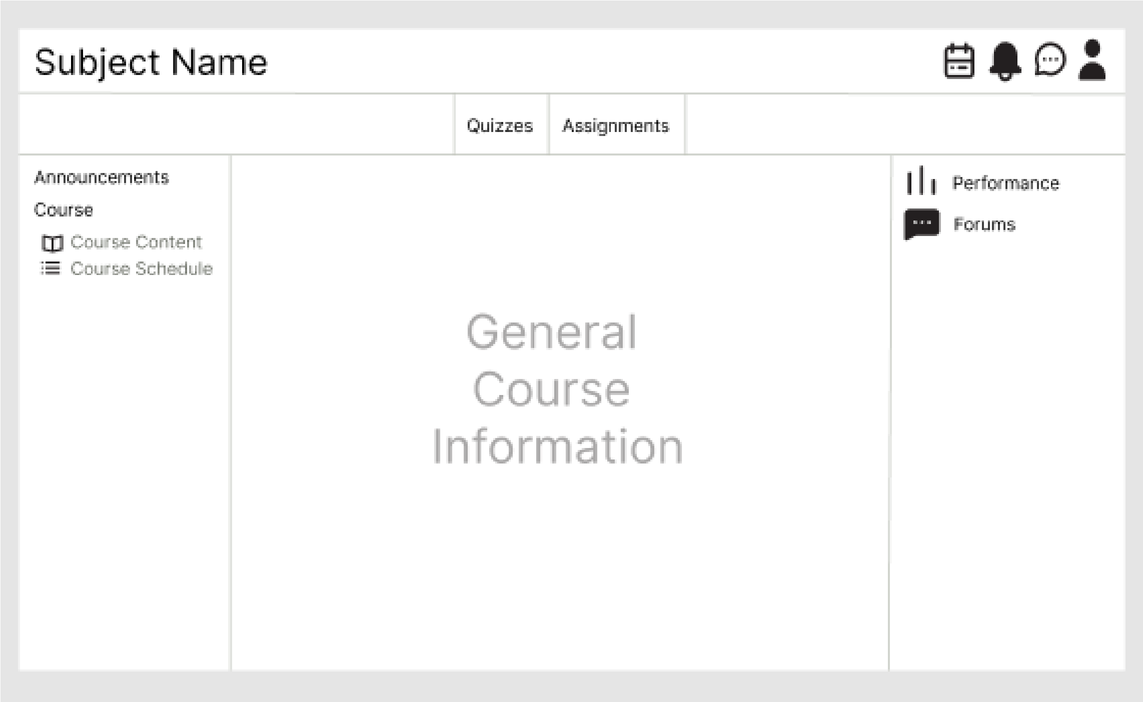
Quizzes and Assignments – New quizzes and assignments added by the Professor are accessible to the student from this section.

Announcements – New announcements, related to class/course by the Professor are available to the students here.

Course – Course details, including the course content, are available here.

Performance – A list of students will be available sorted by their rank (metrics - scores & grades).

Forums – Students can post their queries here. Other students can collaboratively help to resolve the queries.



Expected Behavior:

When students click on the desired option they will be redirected to the corresponding module/sub-module.

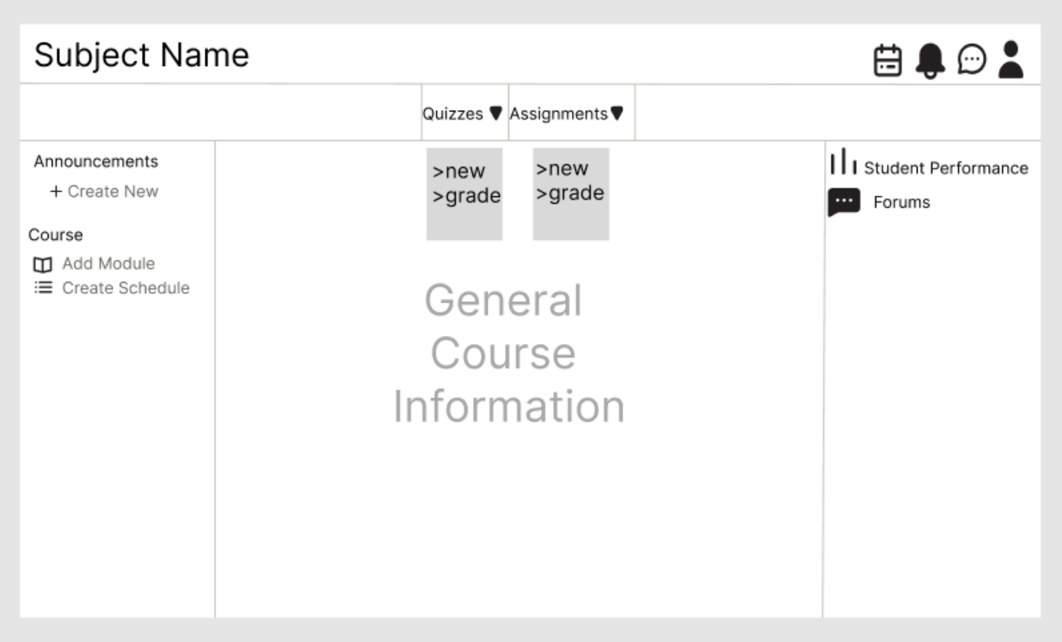
Mockup: Admin Portal

Apart from the modules available in the student portal, the admin has access to other modules which can be used to manage the course.

Announcements – Professors can create new announcements here.

Course – New Module can be added by the professor for the respective subject using add module option.

Quizzes and Assignments – Professor can add and grade new quizzes and assignments.



Expected Behavior:

When the Admin clicks on the desired option they will be redirected to the corresponding module/sub-module.

Mockup: Add Module

Under Course, when professors click on “New Module” the following interface loads.

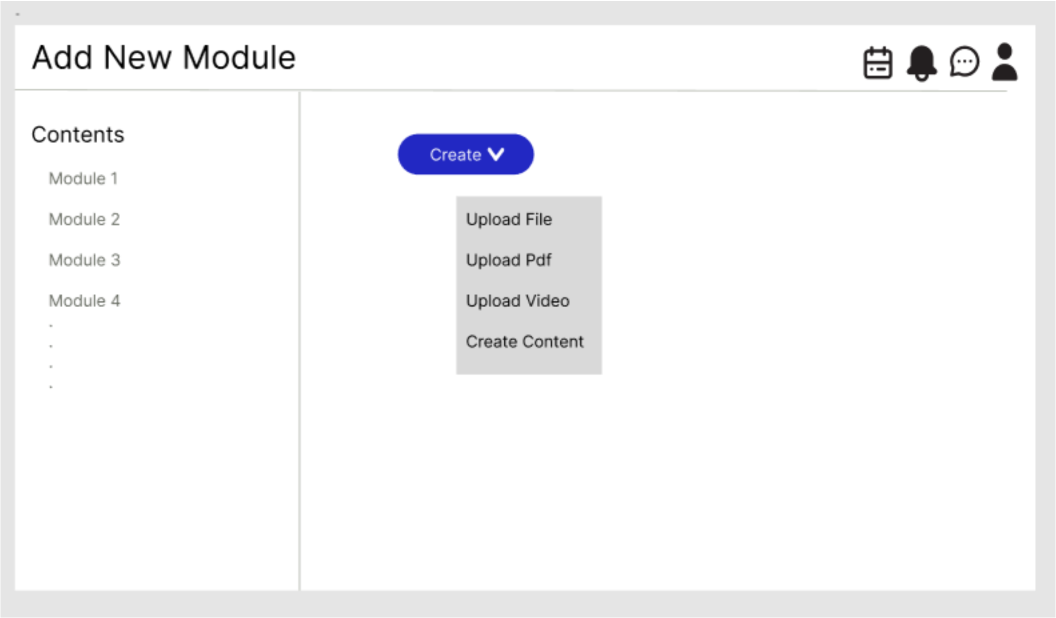
Professors will have the option of creating new content using create button.

The create button has the following options

- Upload Files (Presentations, Documents, ZIPs)
- Upload PDFs
- Upload Video
- Create Content (Text based)

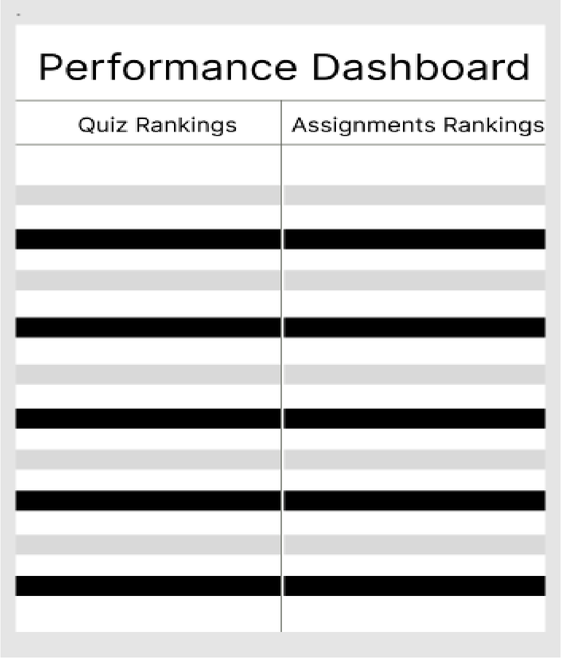
Expected Behavior:
The course will be saved to the database and a status message saying, “Course successfully created” will be visible on the UI.

Error Handling Behavior:
Basic validation including the format of the uploaded content is validated.



Mockup: Performance Dashboards

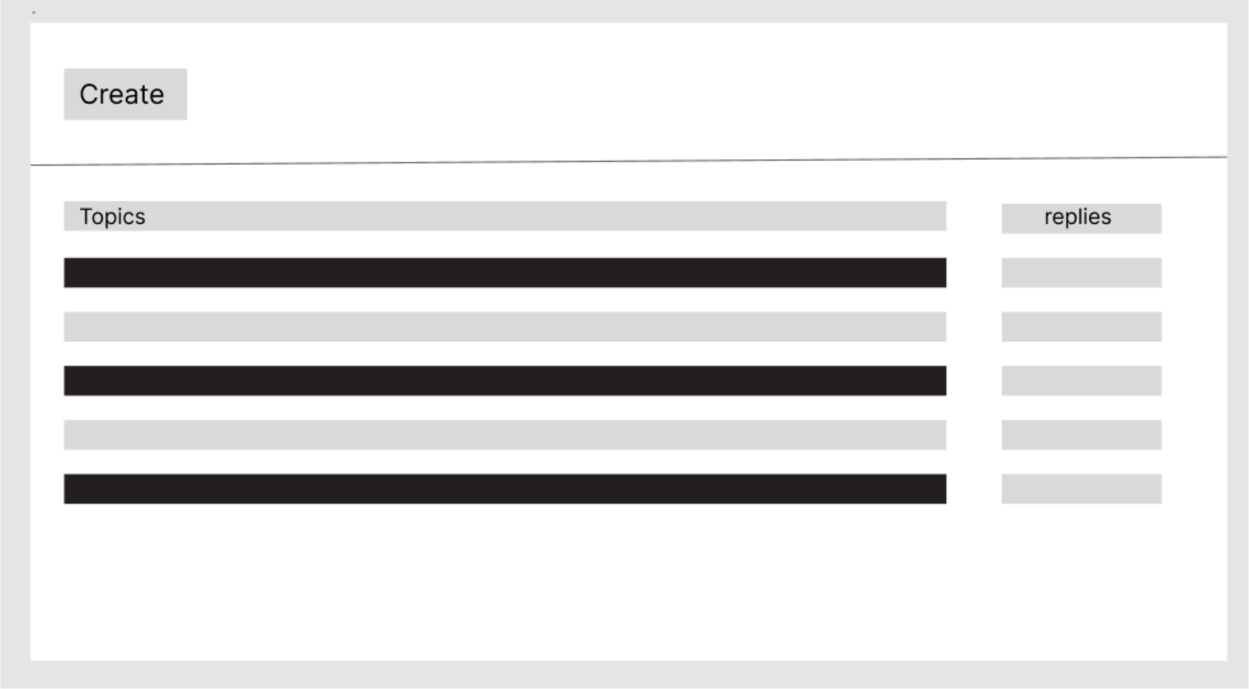
Here the students and professors can view student rankings calculated as per the grade acquired in quizzes and assignments.



Expected Behavior:
The dashboard will display the rankings that is calculated based on the grades obtained.

Mockup: Forums

- Forums consist of the option to create new topics and check replies
- This feature help students to get engage in conversations on important topics.



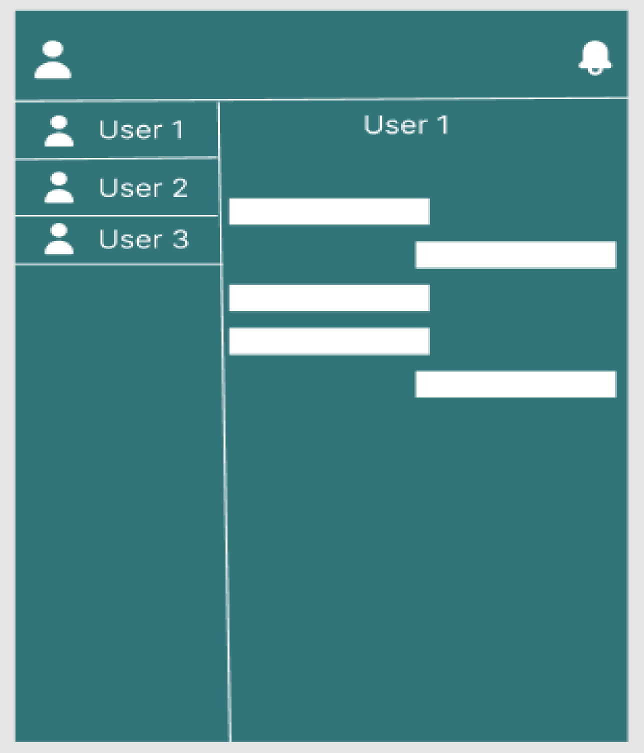
Expected Input:
User could create a new topic for discussion or engage in existing topics.

Expected Behavior:
The new topic that's created will be stored in the database and reflected on the forum.

Error Handling Behavior:
The topic name should not be empty else an error message "please enter the topic name" will be displayed.

Mockup: Chat Application

- The Portal will also consist of a chat module so that students can ask doubts and even communicate with their classmates and professors.



Expected Input:
The user will select a name from the contacts list and a chat video will open for the user to communicate.

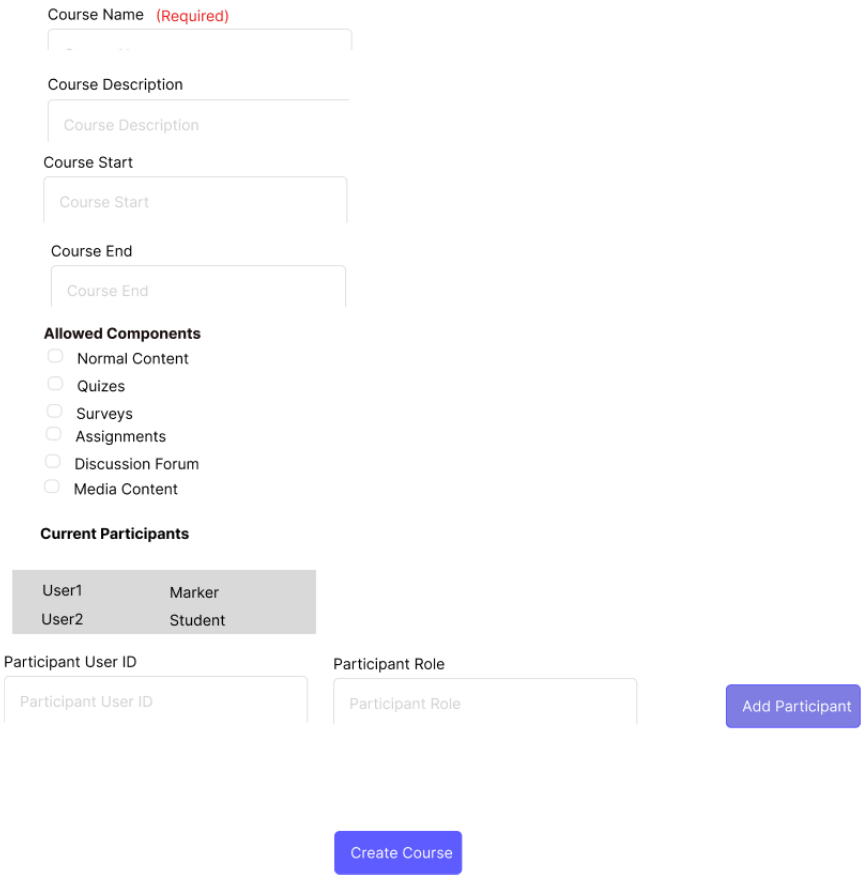
Expected Behavior:
The message will be sent to the selected contact.

Error Handling Behavior:
If there is connection issue(network) an error message “please, try again after some time” will be displayed

Mockup: Create a Course Classroom / Edit Course Classroom

<u>Description:</u>

Admin user will have the ability to create a new course. Admin users will be usually professors. The new course should have a course name and optional description. Admin can choose to plug in features like Normal Content, Quizzes, Surveys, Assignments, Media Content to the course. Admin can then create a course schedule. Admin needs to specify the start date and end date for the whole course, and can specify the start date, due date, and end date for each course module later too. The properties of a course can be edited at any moment and saved.



Course Name (Required)

Course Description

Course Description

Course Start

Course Start

Course End

Course End

Allowed Components

○ Normal Content
○ Quizes
○ Surveys
○ Assignments
○ Discussion Forum
○ Media Content

Current Participants

User1	Marker
User2	Student

Participant User ID

Participant User ID

Participant Role

Participant Role

(Participant Role can be Student, TA, or Marker).

<u>Constraints on Fields:</u>

Start Date – Start date should be a date after the current date.
End Date – End date should be greater than start date.
Allowed Components Checkbox list – At least one of the components must be selected.
Add Participants – A Participant cannot be added once the start date is reached. All participants should have an account created in the app.

Creating a Course:

Expected Input:

- Admin Must Enter a course name. It should not be blank.
- All other fields are optional.
- Admin Clicks on Create course button and course is created.

Expected Behavior:
The course will be saved to database and a status message saying, "Course successfully created" will be visible in UI.

Error Handling Behavior:
Checks if the course name is unique and not blank. Checks if other fields are entered correctly and if constraints are satisfied.

Editing A Course:

Expected Input: Admin can edit any of the fields in the course and click save Button.

Expected Behavior:
The changes will be saved to database and "Course details successfully changed" will be visible in UI.

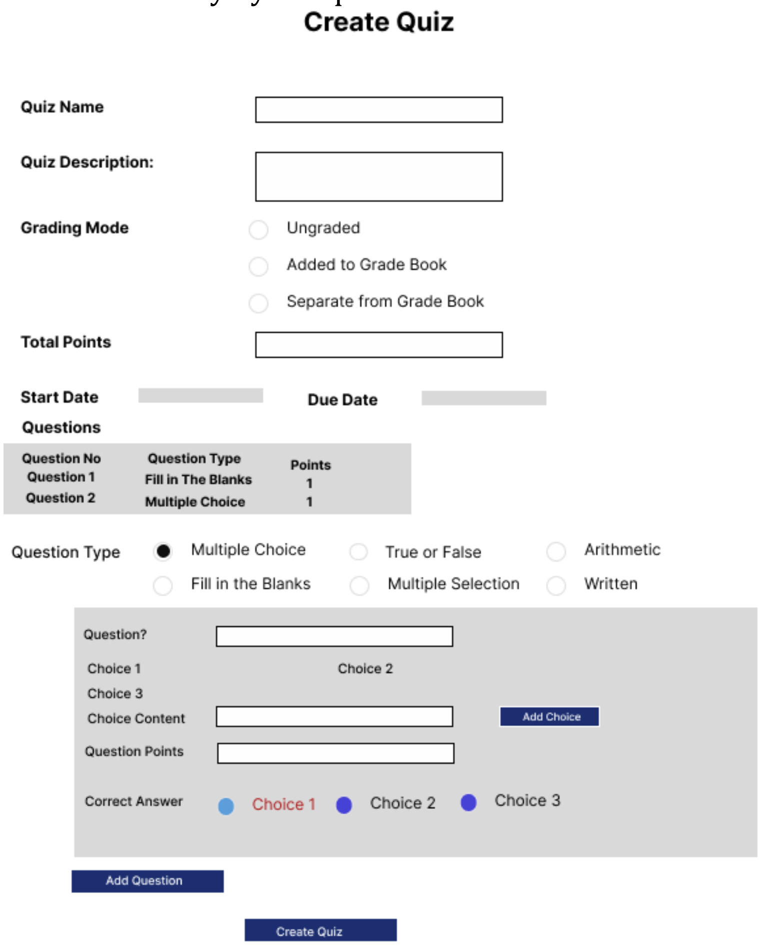
Error Handling Behavior:
Checks if the course name is unique and not blank. Checks if other fields are entered correctly and if constraints are satisfied.

Mockup: Create Quiz

Description:

Admin users can create a quiz as part of a module in a course. The Quiz can be ungraded quiz, graded separately or quiz grade can be part of total grade for the course. Quiz start date refers to the date after which normal users can attend the quiz and due date refers to the date before which quiz should be attended. Admin has the option to add multiple types of questions to the quiz:

1. Multiple Choice – One of the choices is correct
2. True or False - The statement in the question can be true or false. One of these is correct.
3. Arithmetic – The answer to question is a numeric value.
4. Fill in the blanks – There can be one or more blanks in the question and quiz taker must fill them correctly to get complete points for the question or partial points for the question.
5. Multiple Selection – There can be more than one correct choice. Quiz creators can set up grading to have partial points for correct selections and negative points for incorrect selections.
6. Written – The answer to the question is a written statement. Points will be awarded manually by the quiz creator.



Create Quiz

Quiz Name

Quiz Description:

Grading Mode ○ Ungraded ○ Added to Grade Book ○ Separate from Grade Book

Total Points

Start Date **Due Date**

Questions

Question No	Question Type	Points
Question 1	Fill in The Blanks	1
Question 2	Multiple Choice	1

Question Type ● Multiple Choice ○ True or False ○ Arithmetic
○ Fill in the Blanks ○ Multiple Selection ○ Written

Question?

Choice 1 Choice 2

Choice 3

Choice Content

Question Points

Correct Answer ● Choice 1 ● Choice 2 ● Choice 3

Constraints on Fields:

Start Date – Start date should be a date after the current date.
Due Date – End date should be greater than start date.
Quiz Name – Cannot be blank.
Total Points – If grading mode is not ungraded total points should be a value greater than zero.
Question Points – Question Point should be a fraction of total points. And the sum of question points should be equal to total points

Creating a Quiz:

Expected Input:

- Admin Must Enter a Quiz name. It should not be blank.
- All other fields are optional and can be filled in later via edit quiz.
- Constraints should be satisfied for all other fields.
- Admin Clicks on Create Quiz button and quiz is created.

Expected Behavior:

The quiz will be saved to database and a status message saying, "Quiz successfully created" will be visible in UI. And the user can now edit the quiz to add questions or do something else.

Error Handling Behavior:

Checks if the quiz name is unique and not blank. Checks if other fields are entered correctly and if constraints are satisfied. If anything is incorrect that area is highlighted with an error message.

Editing A Quiz:

Expected Input: Admin can edit any of the fields in the quiz and click save Button.

Expected Behavior:

The changes will be saved to database and "Quiz successfully changed" will be visible in UI.

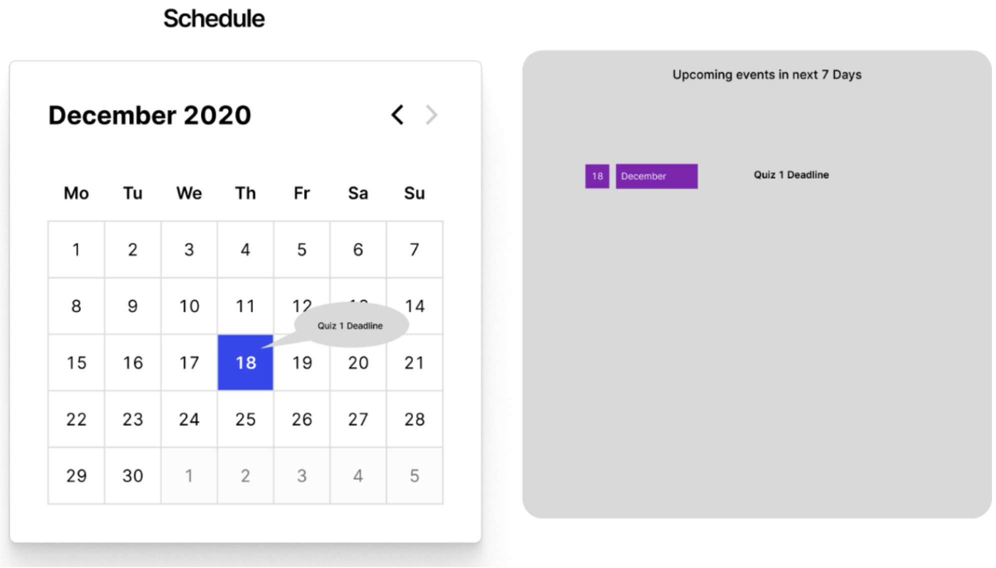
Error Handling Behavior:

Checks if the quiz name is unique and not blank. Checks if other fields are entered correctly and if constraints are satisfied. If anything is incorrect that area is highlighted with an error message.

Mockup: View and Modify Schedule

Description:

All the modules and module components like quizzes and assignments will be added to the schedule based on their start date, end date and due date given while creating them. If the start date is not yet given for a module or other component, they will not be visible in the schedule. Normal users can only view the schedule. Admin users can modify schedules by changing dates of events and deleting events.



Schedule

December 2020

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Upcoming events in next 7 Days

18 December Quiz 1 Deadline

User Permissions:

Normal User: View only

Admin User: View, change date time of events and delete events.

Activity:

Activity can refer to anything like a normal lecture or quiz or assignment or survey or discussion form.

It roughly refers to something the normal user should complete. It will have a start date, end date and due date for graded activities.

Event:

An event can be the start of an activity or the end of the same. Event can also be due date of an activity.

Changing date time of an event (Admin user only):

Constraints.

- If we are changing the start date of an activity, it should not be before the course's start date.
- If we are changing the end date or due date of an activity it should not be before the course end date.
- The activity end date should come after the activity start date and due date should be between start date and end date.

Expected Input:

Case 1: By dragging (Another datetime in the same month)

- User clicks on an event in calendar.
- Drags it to a new date.
- The event will be shifted to the new date
- A pop up will appear to set time. Default value will be same time as for the previous date.
- User can change it to a new time if needed.
- User presses Okay to confirm the change of datetime for event.

Case 2: By clicking and Copying (Another datetime in same month or different month/year)

- User right clicks on an event in calendar.
- Selects move to another date.
- User changes calendar to a new month and year if necessary.
- User left clicks on the day to select it.
- User right clicks and selects reschedule event to this date.
- A pop up will appear to set time. Default value will be same time as for the previous date.
- User can change it to a new time if needed.
- User presses Okay to confirm the change of datetime for event.

Expected Behavior:

The event will be rescheduled to a different date and time, and this will be saved on the database. A status message rescheduled successfully will be shown in UI.

Error Handling Behavior:

Checks if the constraints are satisfied when changing the date of an event. If anything is not satisfied the user cannot change the date and an error message to the corresponding broken constraint will pop up.

<u>Deleting an Event:</u>

Expected Input:

- Admin clicks on an event on calendar.
- Admin right clicks and selects delete.
- A confirmation pop up will be generated.
- If user clicks yes, then the event and the activity associated with it will be deleted.

Expected Behavior:

If the user clicked yes, then the event and activity will be removed from UI. Activity will be removed from the database. Other events associated with the activity will also be deleted from UI.

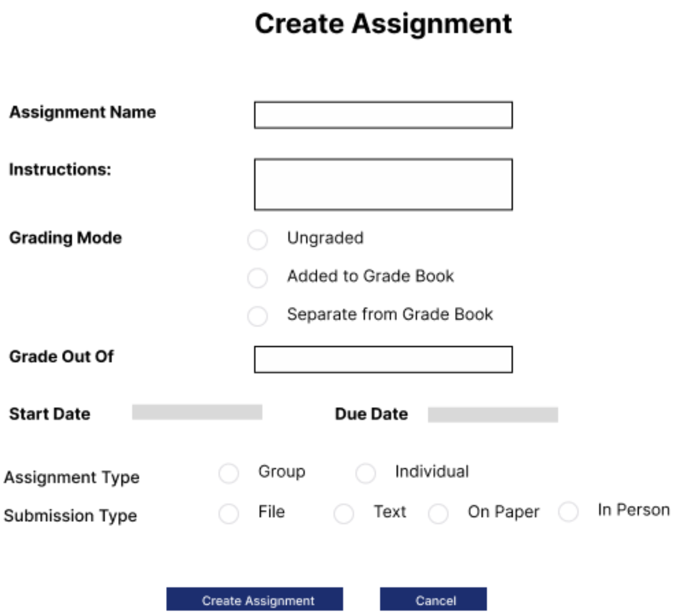
Error Handling Behavior:

Since this is a simple flow, no error handling is required.

Mockup: Create Assignment

Description:

Admin users can create assignments which normal users can submit, and it can be later graded by admin user. Assignments can be group or individual ones. In group assignments multiple participants can be added and evaluated together. There are multiple submission modes available like file upload, adding text, Submitting it on paper and an in-person observation. This gives admin user flexibility in creating assignments.



Create Assignment

Assignment Name

Instructions:

Grading Mode ○ Ungraded ○ Added to Grade Book ○ Separate from Grade Book

Grade Out Of

Start Date **Due Date**

Assignment Type ○ Group ○ Individual

Submission Type ○ File ○ Text ○ On Paper ○ In Person

Create Assignment Cancel

Constraints:

- Assignment name cannot be empty.
- The due date should be always after the start date.

Expected Input:

- User fills out assignment name.
- User fills out other fields in the form.
- User clicks on Create Assignment.

Expected Behavior:

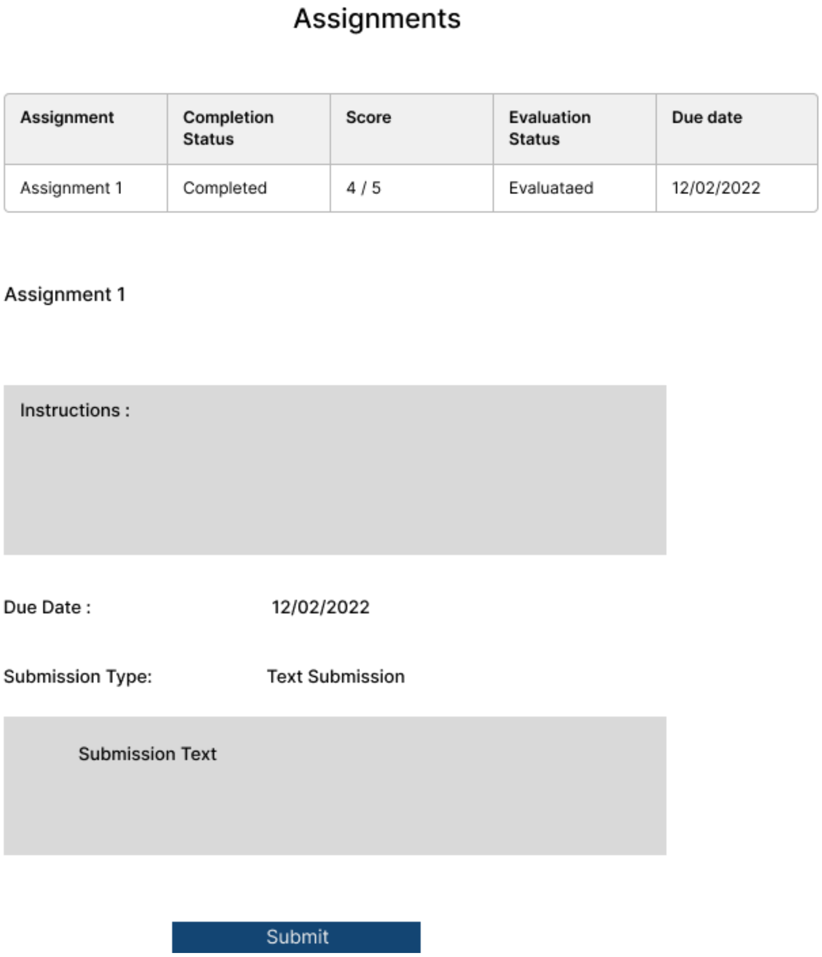
An assignment will be created and saved to database. A status message "Assignment created successfully will appear.

Error Handling Behavior:

Checks if the constraints are satisfied, If not corresponding constraint and error message will be shown in a status message.

Mockup: Submit Assignment

Description:
Normal users can submit an assignment created by the admin user for them before the deadline. For submission options like text and file it can be done through the portal. For submission modes like on paper and in-person submission must be done manually, but normal user can check submission status in the portal.



Assignments

Assignment	Completion Status	Score	Evaluation Status	Due date
Assignment 1	Completed	4 / 5	Evaluataed	12/02/2022

Assignment 1

Instructions :

Due Date : 12/02/2022

Submission Type: Text Submission

Constraints.

- Submission can be done only before the due date.
- Submission Field should not be empty. For text-based submission some text should be there and for file-based submission at least one file should be updated.

Expected Input:

- Normal User selects an assignment from the list.
- A new page opens with the details of the corresponding assignment.
- User uploads a file if submission mode is file or users enters a text if submission mode is text.
- User clicks submit.

Expected Behavior:
The corresponding submission will be added to the database. If the submission has files, it will be also uploaded to the server and handled separately. In the admin user's portal this user's assignment will be added to the submissions list.

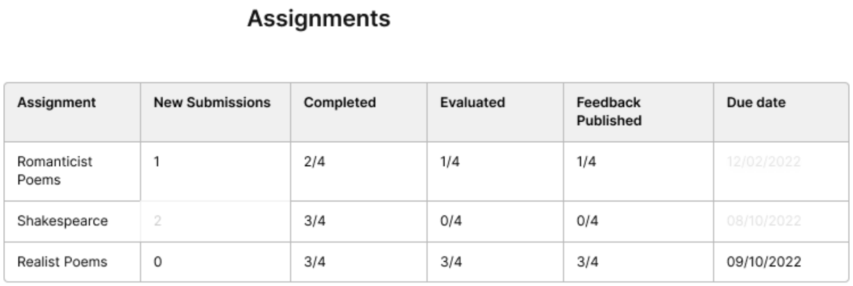
Error Handling Behavior:

- If the submission text is empty an error will be thrown requesting user to add some text.
- If file size exceeds limits an error will be thrown requesting user to upload a smaller file under the limit.
- If the file format is not supported or if the file is found to be malicious, an error is thrown, and the user is informed.

Mockup: Evaluate Assignment

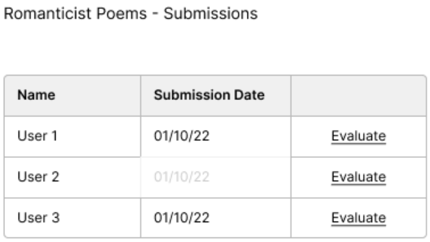
Description:

Admin users can see the list of assignments created. Get statistics like how many students have completed them and how many are left to evaluate. When Admin clicks on an assignment name, they will be redirected to a new page where it lists people who have submitted that assignment. Here admin can click on Evaluate button on one of them. Then a new window will be opened where admin can assign a grade and feedback.



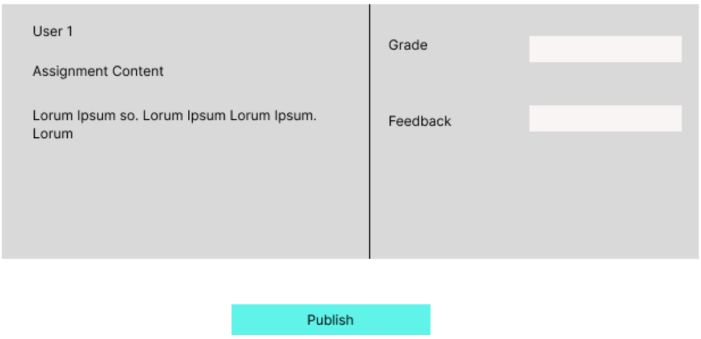
Assignments

Assignment	New Submissions	Completed	Evaluated	Feedback Published	Due date
Romanticist Poems	1	2/4	1/4	1/4	12/02/2022
Shakespearce	2	3/4	0/4	0/4	08/10/2022
Realist Poems	0	3/4	3/4	3/4	09/10/2022



Romanticist Poems - Submissions

Name	Submission Date	
User 1	01/10/22	Evaluate
User 2	01/10/22	Evaluate
User 3	01/10/22	Evaluate



User 1

Assignment Content

Lorum Ipsum so. Lorum Ipsum Lorum Ipsum. Lorum

Grade

Feedback

Constraints.

- Grading is mandatory before publishing.

Expected Input:

- User selects an assignment from the list.
- For assignment user clicks evaluate button on a submission by some other user.
- User enters grade.
- User enters feedback.
- User clicks Publish.

Expected Behavior:

The corresponding submission will be graded and saved to database. In admin user’s submission list for that assignment the corresponding submission will have status “Evaluation Done”. For the normal user who submitted this assignment Grade and feedback will be visible. Grade will be added to grade report.

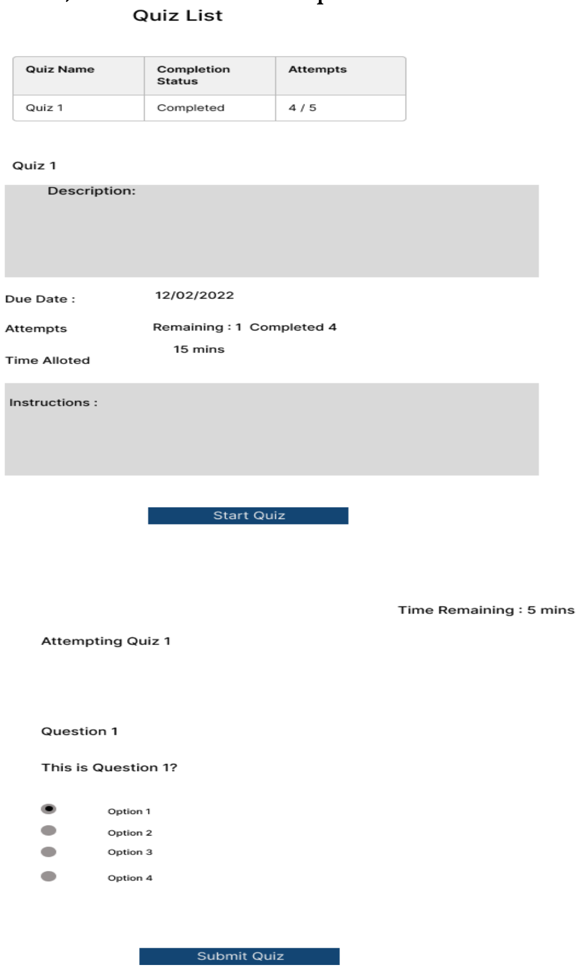
Error Handling Behavior:

If publishing is done while leaving Grade field blank an error is thrown saying “Please enter a Grade”.

Mockup: Attend Quiz

Description:

Normal users can attend quizzes created by admin user. They can get a list of quizzes they need to attend, select one of the quizzes and attend it.



Quiz List

Quiz Name	Completion Status	Attempts
Quiz 1	Completed	4 / 5

Quiz 1

Description:

Due Date : 12/02/2022

Attempts Remaining : 1 Completed 4

Time Alloted 15 mins

Instructions :

Time Remaining : 5 mins

Attempting Quiz 1

Question 1

This is Question 1?

- ● Option 1
- ○ Option 2
- ○ Option 3
- ○ Option 4

Constraints.

- Submission can be done only before the due date.
- Quiz should be completed before the allotted time if the quiz has an allotted time option.

Expected Input:

- Normal User selects a quiz from the list.
- A new page opens with the details of the corresponding quiz.
- User clicks on start quiz; A new window will open with all the quiz questions.
- The User will answer questions one by one.
- User clicks on submit.

Expected Behavior:

The corresponding quiz submission will be added to the database. If the quiz can be auto-graded the grade is computed and will be visible to the user. The grade will also be stored in the database.

If the user runs out of time while attending the quiz, the quiz is auto submitted, and answers user has done will be evaluated for grading.

Error Handling Behavior:

- If user opens the quiz window, and if due date is exceeded before clicking the start quiz button an error is thrown saying user cannot attend quiz as deadline is over.
- For quizzes with unlimited time if deadline is breached while attending the quiz, the quiz cannot be submitted after clicking the submit button. An error will be thrown saying “deadline breached cannot submit”.

SPRINTS AND MILESTONES

We plan to follow Agile methodology throughout the development of the project.
For project planning, we'll be using Atlassian Jira tool.
We shall be having regular teams and in-person SCRUM meetings

The sprints for the development process are divided as follows:

Sprint 1 – (1st Oct - 9th Oct)

1. Jira board setup
2. Creation of CI/CD pipeline
3. Project & Environment Setup
4. Testing CI/CD pipeline integration

Sprint 2 – (10th Oct – 16th Oct)

1. Preliminary UML design
2. Database Design
3. Preliminary test case design
4. 10% feature completion (auth)

Sprint 3 – (17th Oct – 29th Oct)

1. 30% feature completion (dashboards)

Sprint 4 – (30th Oct – 19th Nov)

1. 50% feature completion (student and admin features)
2. Mid project review

Sprint 5 – (20th Nov – 7th Dec)

1. 100% feature completion (student and admin features and p2p chat)
2. Project Demo video creation
3. Thorough testing

REFERENCES AND TOOLS USED

[1] C. Brame, “Flipping the classroom,” *Vanderbilt University*, 31-Jan-1970. [Online]. Available: https://cft.vanderbilt.edu/guides-sub-pages/flipping-the-classroom/. [Accessed: 02-Oct-2022].

[2] “Diagrams.net - free flowchart maker and diagrams online,” *Flowchart Maker & Online Diagram Software*. [Online]. Available: https://app.diagrams.net/. [Accessed: 02-Oct-2022].

[3] “The Collaborative Interface Design Tool.,” *Figma*. [Online]. Available: https://www.figma.com/. [Accessed: 02-Oct-2022].