Software Requirement Specification

for

Online Proctoring System

Version 1.0

Prepared by
Anmol Waghmare
Aditya Jadhav
Aditya Wagh
Sameesh Yadav
Vageshwar Yadav

Under the Guidance of
Prof. Sameer B. Patil

KIT's College of Engineering

21-12-2020

Table of Content

1. Introduction

The Introduction contains the purpose of the Software Requirements Specification (SRS) in Section 1.1 and its connection with the entire software production process, as well as how this document will change throughout this process as part of the evolution of this document, personnel involved with this project are contained in Section 1.2.

1.1 Purpose

This document is intended to present the requirements of the system to be produced, both functional and non-functional. The specifications contained in this document will be used to support the production of the system in later stages, in an attempt to reduce the development effort involved.

The SRS explains the current system being used by the customer, the reason for the creation of the new system and a broad overview of the proposed system to be developed.

The audience of this document (Software Requirements Specification) primarily includes the Project Team and Project Supervisor.

1.2 Personnel

The personnel involved with this project are outlined in this section. This includes the Project Team and the Project Supervisor.

1.2.1 Our Team

Our development Team consists of:

- Anmol Waghmare
- Aditya Jadhav
- Aditya Wagh
- Sameesh Yadav
- Vageshwar Yadav

1.2.2 Project Supervisor

The supervisor of our team is:

- **Prof. Sameer B. Patil**

  Department of Computer Science and Engineering
  Kolhapur Institute of Technology’s

1.3 Acronyms & Abbreviations

- **AI:** Artificial Intelligence
- **SRS:** System Requirement Specification
- **KIT:** Kolhapur Institute of Technology
- **OE:** Oral Examination
- **POE:** Practical Oral Examination
- **I/O:** Input
- **O/P:** Output

1.4 Definitions

- **Module:** Each separate component developed within the System
- **User:** Student and Teacher
- **Platform:** The project website
- **System:** The project Website
- **M.E.R.N:** Modern Tools used to develop Full stack website (MongoDB, Express JS, React JS and Node JS)

1.5 References

- IEEE-SA Standards Board, IEEE Recommended Practice for Software Requirements Specifications

1.6 Overview

The remainder of this document is structured in the following way:

- **Section 2:** Project Overview - Contains details of the current system, why the system is to be modified, and the proposed system to be developed.
- **Section 3:** Functional Requirements - All of the functional requirements of the proposed system are detailed here.
- **Section 4:** Development Stack- Details the development.
- **Section 5:** Non-functional Requirements - Details the non-functional targets of our system.
- **Section 6:** Other - Other Details

2. Project Overview

2.1 Existing System(s)

The educational institutions use different softwares for conducting various educational related activities.

For Example:

In KIT two different systems are used to manage student's data

- Contineo:

Course and course credit mapping are maintained here, Exam results and attendance of students is also stored and maintained.

- Moodle:

Course details like assignments, tutorials and quizzes for assessments are managed on Moodle.

By using login credentials students can view course materials, submit assignments and take quizzes

Faculty with their credentials can create course materials, schedule assignments and quizzes.

Both these systems are web based.

Cons:

1. Two different systems make data management tasks redundant.
2. Quizzes are unproctored.

3. No proper platform for coding.

2.2 Proposed System

Online Proctoring System as the name suggests is an online platform to conduct proctored exams along with course and attendance management. Major function of this project is to conduct online proctored examination mainly for Multiple Choice Questions, Programming and Oral Examination and storing results of examinations for better grading.

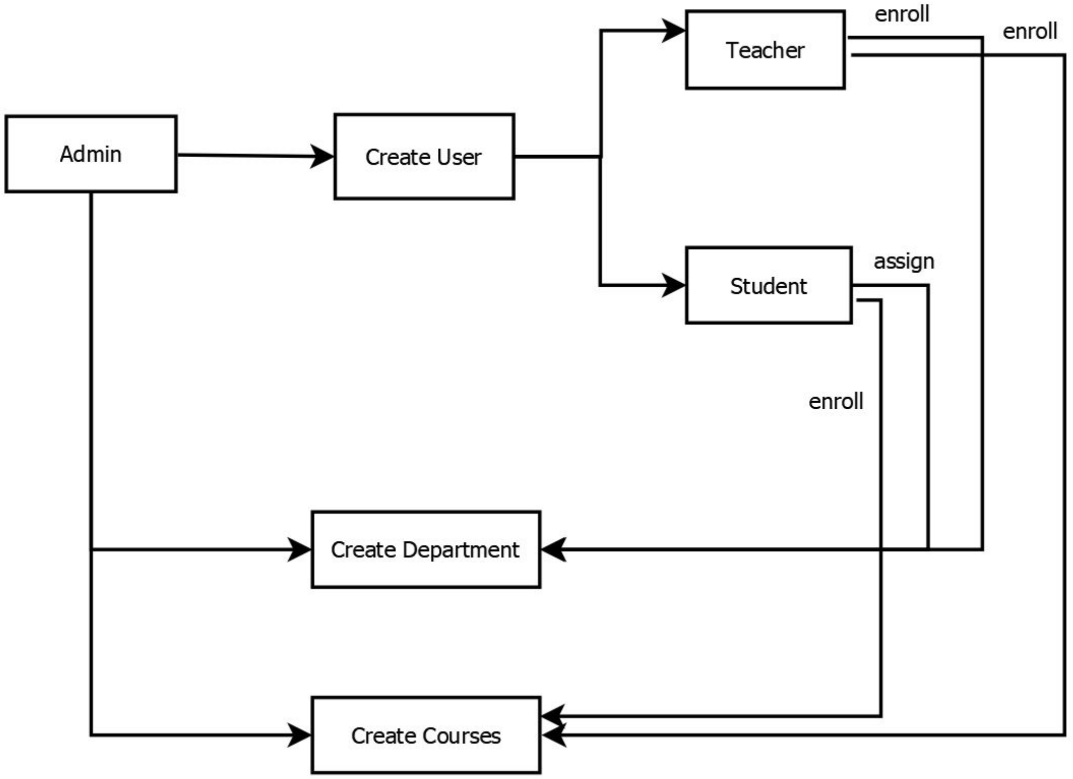

Figure 2.2.1: Proposed System (admin)

Initially an Administrator from the university/Institute will be responsible for creating Departments , classes and courses within the departments assigning course faculty and enroll students to required courses. He is also responsible for creating batches for each class.

Faculty can see the list of all the courses assigned to him. Faculty can then add course contents like assignments, quizzes and schedules.

Students can see course contents of the enrolled courses.

There are ‘5’ main modules within this project:

1. Proctored quiz
2. Proctored OE-POE
3. Assignments
    a. Coding Assignments
4. TimeTable
5. Attendance

Each module has its own interface and functionality. The functionality (refer to Section 3) will also depend on the user type (faculty or Student).

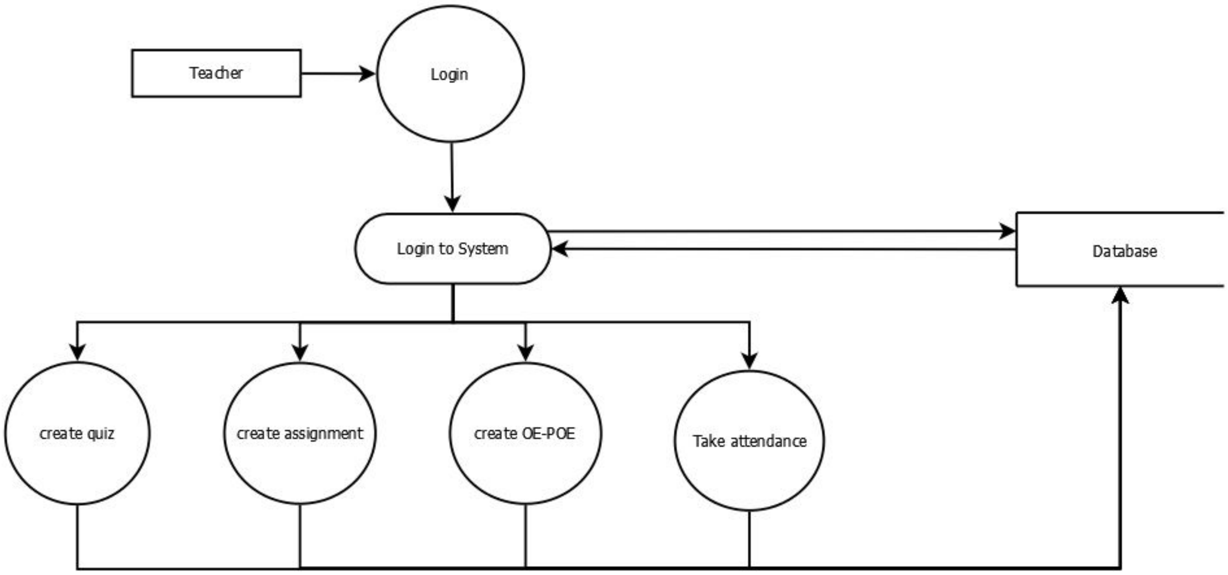

Figure 2.2.2: a) Proposed System Teacher’s View

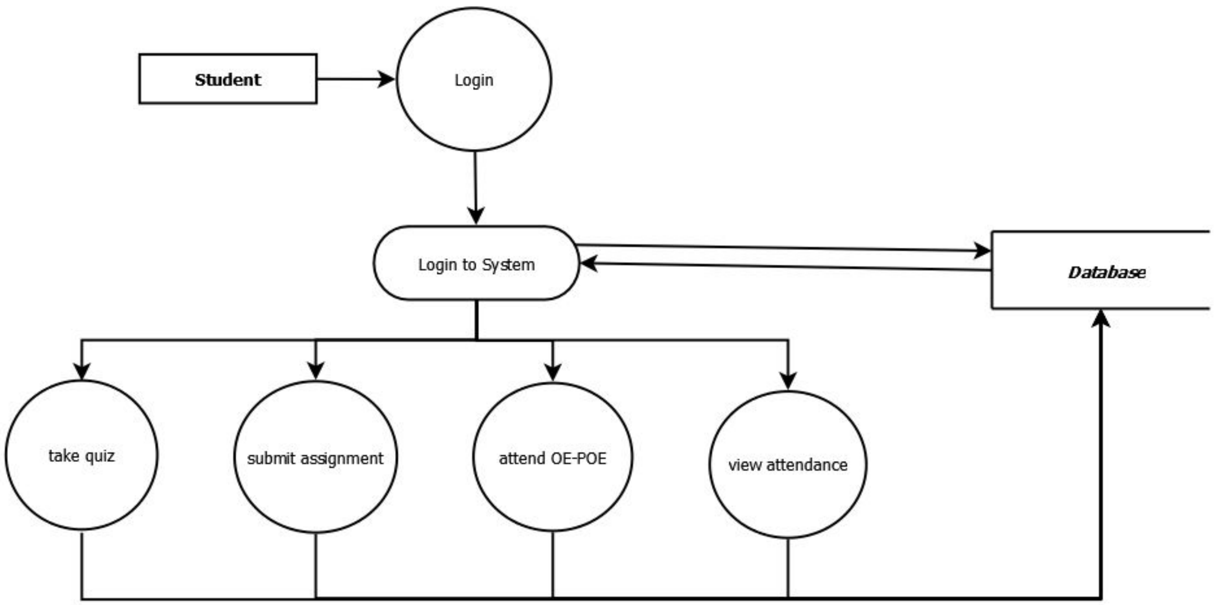

Figure 2.2.2: b) Proposed System Student''s View

2.3 Scope

The objective of this project is to provide a web based platform that can be useful in conducting remote examination and remote course management for students and teachers.

As a part of the project we will be developing 3 main modules: Proctored Quiz , Proctored OE-POE and Coding Assignments. Each module will be described later within the functional requirements in Section 3 of this document.

2.4 User Characteristics

The System being similar to the Course management system, hence the main users will be Teachers and Students.

Teachers will have a higher level of access than students, they can add,delete,edit course contents.Teacher can add attendance details.

The term ‘teacher’ will collectively be used to describe all teaching users, such as lecturers and tutors

Students will have a standard level of access that will not differ between individuals, they can upload assignments, attempt quizzes and OE-POEs, they can also see their performance in exams and attendance reports.

2.5 Constraints

2.5.1 Hardware Constraints

- CPU: Intel or AMD processor with 32-bit or 64-bit.
- RAM: 4GB or more.
- Internet: Internet connection required.
- Webcam must be present for the proctored exam.

2.5.2 Software Constraints

- Software: Web Browser (*Recommended:* Chrome)
- Operating System: Windows 7 or above.
- Web browser must be installed on the system.

3. Functional Requirements

3.1 Proctored Quiz

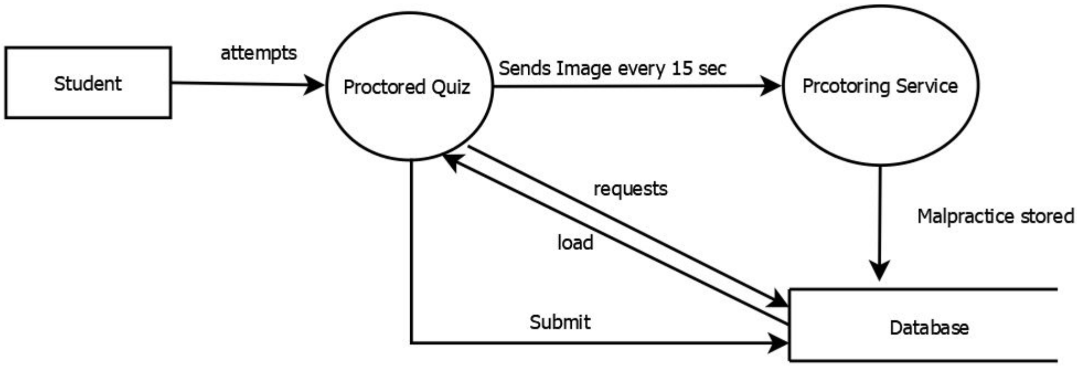

Figure 3.1.1: Proctored Quiz

This function differs depending on the user.

i. MCQ based question and answer exam should be conducted.
ii. Single MCQ should appear on the screen at a time.
iii. MCQ number grid should appear at the right hand side of the screen for navigation.
iv. The exam will be conducted in browser’s full screen mode.
v. No Tab Switching should be allowed during the exam.
vi. Capture and store a clear and visible picture of the attendee at the start of assessment.
vii. Capturing attendee pictures every 15 sec and passing it to AI for identification of any malpractices.
viii. The Attendee should be present in the image at all times if not he should be warned.
ix. If the number of warnings exceeds 5 then a final warning should be issued.
x. Violating the final warning should revoke the user from taking the exam and the user should be directed to “malpractice conducted page”.
xi. The record of the performance of each user in the examination should be stored at the server side.
xii. The record of users reported to be conducting malpractice should be stored.

3.1.1 Creating Quiz

User: Teacher

Description: Teacher will be able to define specifications of the quiz.

The following details must be entered (* indicates required Field):

- Quiz Title*
- Quiz Description
- Quiz start date - time*
- Quiz end date - time*
- Quiz Duration*
- Proctored Option (Yes/No) *
- Review Visibility Option (Yes/No)*

Dependencies/Constraints:

- Quiz start date-time < Quiz end date-time
- Quiz end date-time - Quiz start date-time >= quiz duration

Ranking: Essential

3.1.2 Adding Quiz Question Bank

User: Teacher

Description: Teacher will add question bank in given the format the system should parse the file and render questions if format is correct else show error.

Dependencies/Constraints: Uploaded files must be .txt

Ranking: Essential

3.1.3 Deleting Quiz

User: Teacher

Description: Teacher will be able to delete the quiz.

Dependencies/Constraints: Quiz should be added

Ranking: Essential

3.1.4 Modify Quiz

User: Teacher

Description: Teacher will be able to modify specifications of the quiz.

The following details must be entered (* indicates required Field):

- Quiz Title*
- Quiz Description
- Quiz start date - time*
- Quiz end date - time*
- Quiz Duration*
- Proctored Option (Yes/No) *
- Review Visibility Option (Yes/No)*

Dependencies/Constraints:

- Quiz start date-time < Quiz end date-time
- Quiz end date-time - Quiz start date-time >= quiz duration
- Quiz should already be added

Ranking: Essential

3.1.5 Attempt Quiz

User: Student

Description:

Student:

- ○ Before the quiz start time: Quiz should be visible to students but the attempt button will not be available before the start time
- ○ After quiz start time: The student should be given a button to attempt the quiz
- ○ After quiz end time: The attempt button should not be available. If a student fails to attempt the quiz then it should display a “Not Attempted” message. If a student attempts a quiz then the score and review should be visible as per the quiz review visibility options set by the teacher.

Dependencies/Constraints: Quiz should be added

Ranking: Essential

3.1.6 Get Report of Quiz

User: Teacher

Description:

Teacher:

- ○ After quiz start and end time: The Number of students attempted the quiz and average of the marks scored should be displayed. Individual student performance could be monitored.
- ○ After Quiz Completion the assessment report could be exported in excel format.

Dependencies/Constraints: Quiz should be added

Ranking: Essential

3.2 Proctored OE-POE

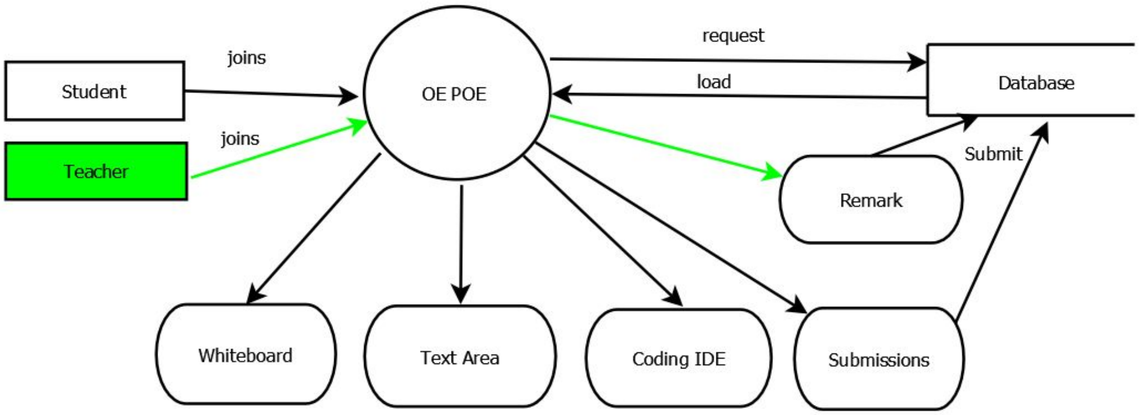

Figure 3.2.1: Proposed OE-POE

OE-POE should be conducted live in form of video conferencing, students will log in and wait for their turn while the teacher and any other person to conduct will remain live on the platform. It involves following sub functionalities.

3.2.1 Creating OE-POE

User: Teacher

Description: Teacher can schedule audio/video conference for Oral Exam/Practical and Oral Exam.

Following details required to schedule OE/POE

- Course Name.
- Type of Assessment(OE/POE).
- Date and Time.
- Duration.
- Specification of participants.
    - Batch, Group
- Audio , Video or Both.

Dependencies/Constraints:

- start time < due time.
- An Assessment Task must have been completed.

Ranking: Essential.

3.2.2 View OE-POE

User: Teacher,student

Description: Students will see the Course of OE/POE Exam Schedule.

Following information will be visible to the students:

- Course Name.
- Type of Assessment(OE/POE).
- Date and Time.
- Duration.
- Specification of participants.
- Audio , Video or Both.
- Button to join the exam.
- Indication of availability.

Dependencies/Constraints:

- Students will only see upcoming OE/POE.
- Students won’t be able to see the exam after scheduled Time.

Ranking: Essential.

3.2.3 Attempt OE/POE

User: Student, Teacher

Description: After student clicks on “attempt”

User gets a window which includes the Title window.

Join Button: Join the scheduled meeting for OE/POE after clicking.

After Joining depending user should be able to see following window:

- Supervisor Window: Containing Info about examiners member
- Whiteboard: to scribble where both participant and examiner can write
- A Text Area for writing where both participant and examiner can write
- A Coding IDE if POE assessment selected
- Teacher has an extra window : Report Window:
    - Teacher can give remarks to particular student and is stored on database

Dependencies/Constraints:

- No attempts allowed after the due date.
- Cannot attempt before start date-time.
- Simultaneous scribbling and writing

Ranking: Essential

3.2.4 Deleting OE/POE

User: Teacher

Description: Teacher will be able to delete the Schedule of Entry.

Delete option present under course contents assigned to the user.

Dependencies/Constraints:

Ranking: Essential

3.2.5 Exporting Report

User: Teacher

Description: Teacher will be able to get an Excel file of the OE POE.

Export option present under course contents assigned to the user.

Dependencies/Constraints: File format xls

Ranking: Essential

3.3 Coding Assignment

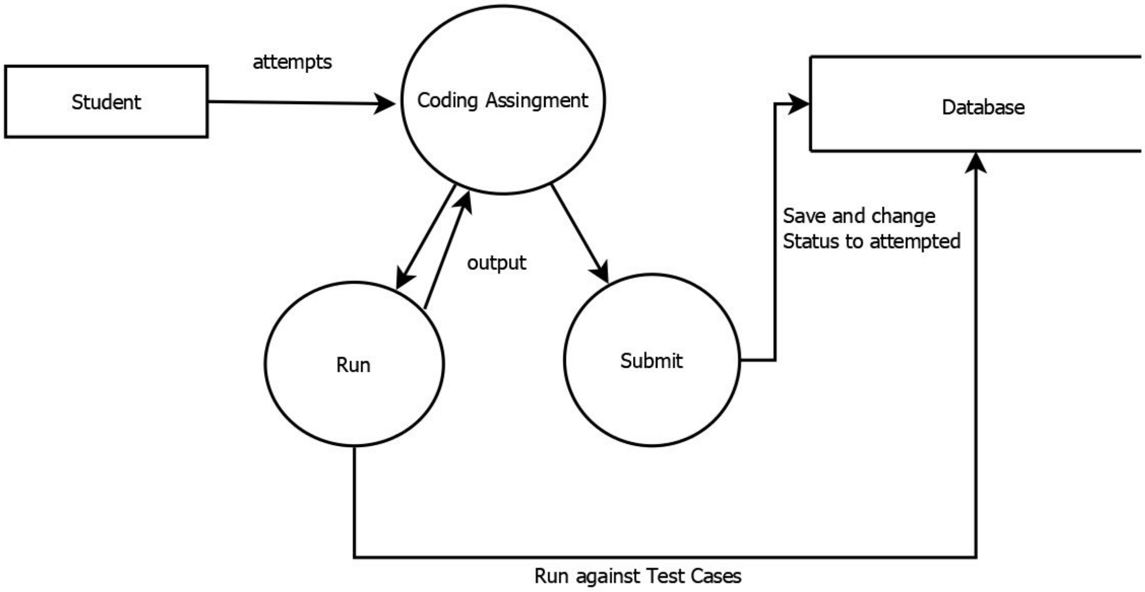

Figure 3.3.1: Coding Assignment

3.3.1 Create Assignment

User: Teacher

Description: Creating coding assignment and specifying all below configs.

The following details must be entered (* indicates required Field):

- Assignment Title*
- Problem Statement*
- Proctored (Yes/No) *
- Programming Language*
- Input format
- Add test cases file* (txt specific format)
- Output format
- Start date - time*
- Due date - time*

Dependencies/Constraints:

- start date-time < due date-time
- Programming Languages Includes: C, C++, Java, Python and C#

- Can select only 1 programming language
- Test Case Format will be txt as per problem statement

Ranking: Essential

3.3.2 Viewing Coding Assignment

Users: Student,Teacher

Description: Student's view of the assignment

User will get a list of Course Contents under the specific course there he can see Coding assignment, after opening the assignment user will be able to Title , Problem statement , i/p o/p format , his attempt status, due date and a conditional button at the bottom

Attempt Status will be: Attempted or Not attempted

And Button Will be: Attempt if Accessed before due time else No Button.

On Clicking the button attempt should start.

Dependencies/Constraints:

- No attempts allowed after the due date.
- Cannot attempt before start date-time.

Ranking: Essential

3.3.3 Attempt Assignment

User: Student

Description: After student clicks on "attempt"

User gets a window which includes Title , problem statements , i/o o/p format and a coding IDE with a run button and a submit button and an output window.

Run Button: Runs the code written on IDE against test cases and shows output on output window

Submit Button: Submit code and status changes to attempted.

Dependencies/Constraints:

- No attempts allowed after the due date.
- Cannot attempt before start date-time.

Ranking: Essential

3.3.4 Deleting Assignment

User: Teacher

Description: Teacher will be able to delete the assignment.

Delete option present under course contents assigned to the user.

Dependencies/Constraints:

Ranking: Essential

3.3.5 Modify Assignment

User: Teacher

Description: Teacher will be able to modify specifications of the quiz, Modify option present under course contents assigned to the user , will be able to edit all the parameters assigned during creation.

Dependencies/Constraints:

Ranking: Essential

3.3.6 Get Report of the Assignment

User: Teacher

Description:

User is able to download report of the assignment performance in the from of excel sheet , the sheet contains following column

- Student ID
- Full Name
- Time Taken
- Score (No. of test cases passed)
- Malpractice Count

Dependencies/Constraints: Document format: excel (.xls)

Ranking: Essential

3.4 Time-Table

Time table is course specific and should be added during course creation by the admin

3.5 Attendance

User: Teacher

Description: This feature will allow the user to maintain day to day class attendance records classified by courses.

Features:

- Creating a attendance record entry with the following details:
    - Course name
    - Course Conductor
    - Day/Date
    - Time period (start and end of class)
- Viewing past records.
- Updating or deleting previous entries.
- Storage and presentation of statistical data of a course or an individual.

Dependencies/Constraints:

- Time period specified should consist of a positive frame of time.
- The timetable should be consistent with the attendance record entries.
- Changes in schedule will be logged.
- No more than one entry should exist having the same recorded instance of time.

Ranking: Essential

3.6 Login Logout

Description: User should be able to login and logout. Login from more than one device should not be permitted.

Ranking: Desirable

4. Development Stack

M.E.R.N Stack (MongoDB Express React Nodejs) , along with MERN Stack Oracle DB can be used as per requirement.

Front End Web Site: React JS

Backend : Express and Node JS

Database: MongoDB and Oracle

5. Non Functional Requirement

5.1 Performance Requirement

5.1.1 Multiple Access

The platform should allow multiple users to access it.

Maximum capacity: depends on the server capacity (required: 100)

Ranking: Desirable

5.2 Other Requirement

The following non-functional requirements are related to some of the software system attributes

5.2.1 Maintainability

Description: The module shall be designed with the view that bugs may need fixing, for future optimisations and for adding extra functionalities to the system.

Constraints: Inexperience of Team Members in design.

5.2.2 Extendability

Description: The module shall be designed with the view that these ”optional” functional requirements will be implemented at a later date. Also the module shall be designed with a view that other tasks may be added to the module.

Constraints: The current set of tasks are not independent of each other and the inexperience of Team Members in design.

5.2.3 Security

Description: The system shall make sure that the correct interfaces and functionalities are available according to the user type (Teacher or Student) that is using it. Cryptographic techniques should be used to protect sensitive information.

Constraints: The user type is recognised by the system.

6. Other

6.1 User Interface

Online Proctoring System will provide an interface between all the individuals involved in the educational organization and its activities. Individual classes involve Teachers/Professors and Students. Teachers will handle all the essential academic activities using the proposed system.

6.2 Software Interface

The proposed system is a web based application and will run on a web browser to provide the later specified utilities. Operating systems that support the latest version of web browsers like Chrome or Firefox are compatible.