Software Requirements Specifications Document
Automated Live Classroom Performance Evaluation
Group 2
Project 6

180101033- Kartikay Goel
180101023- Falak Chhikara
180101005- Alay Chirag Shah
180101060- Rahul Choudhary

February 10, 2021

Contents

1 Introduction

1.1 Overview

The first chapter contains the introduction part which gives a basic idea about what the project is going to be and provides the definitions which will be used throughout the document and also the basic definitions given by the IEEE Computer society.
The second chapter will have the system requirements which include functional requirements.
The third chapter will include non-functional requirements.

1.2 Purpose

The purpose of this document is to present a detailed description about the "Automated Virtual Classroom Performance and Evaluation". This document will explain the purpose and features of the system, Interfaces of the system, Various constraints under which the system has to operate and requirement analysis which includes both the functional and nonfunctional requirements.

1.3 Definitions

Term	Definition
Users	Student,faculty,Management,Admin.
Admin	Application administrator responsible for application management.
Management	Registered users which manage the entire working of Virtual Classroom.
Faculty	Registered teachers of VCS to teach the students studying in VCS.
Students	Registered users of VCS as the students of the classroom. students of the classroom.
Examination	Test conducted to evaluate the performance of a student in a particular subject/course.
Performance Report	Report showing the progress of a student after the examination is being conducted. It will be a cumulative course report.
Feedback Report	Report showing the feedback of a teacher. It will be a cumulative class feedback.

1.4 Scope of the Project

The Desktop Website will be a platform especially for students who are pursuing their under graduation. Nowadays, automated evaluation is an essential part of any educational organization. The teacher requires the live classroom performance evaluation application to quickly evaluate students' performance and get feedback about their teaching quality. The evaluation process may include:
1. Evaluation of one-word answers and objective type questions.
2. View the examination performance, both historical and real-time, on the teacher/student device.
3. View the top scorer and other related details (average score, lowest score, etc.) if needed.
Students can also see their performance reports instantly (i.e., after the exam over). The automated evaluation helps the teacher save time, which can be better used by solving doubts of the students. A student gets periodic performance reports faster, which helps students to improve their learning.

1.5 Intended Audience

The audience on which we are aiming this project are the students and instructors of schools and colleges.

2 Functional Requirements

R1 Student Sign up

Description: A email-verification link is generated using the credentials of the user. Then the validation link is sent to the email id of the user where the user confirms his/her email and after successful confirmation, the user is added to the system.

R1.1 Generate Validation Link

Input: User Credentials
Output: Validation Link
Description: Validation link is generated using the credentials provided.

R1.2 Send Validation Link

Input: Validation Link
Output: Success/Failure message
Description: Validation link which is generated is sent to the user.

R1.3 Add student

Input: Validation Link
Output: Success/Error message
Description: The student is added after he verifies email with the link sent to him/her.

R2 Instructor Sign Up

Description: A email-verification link is generated using the credentials of the instructor. Then the validation link is sent to the email id of the instructor where the instructor confirms his/her email and after successful confirmation, the instructor is added to the system with escalated privileges.

R2.1 Generate Validation Link

Input: Instructor's Credentials
Output: Validation Link
Description: Validation link is generated using the credentials provided.

R2.2 Send Validation Link

Input: Validation Link
Output: Success/Failure message
Description: Validation link which is generated is sent to the instructor.

R2.3 Add Instructor

Input: Validation Link

Output: Success/Error message
Description: The instructor is added after he verifies email with the link sent to him/her.

R3 TA Sign up

Description: A email-verification link is generated using the credentials of the user. Then the validation link is sent to the email id of the user where the user confirms his/her email and after successful confirmation, the user is added to the system.

R3.1 Generate Validation Link

Input: User Credentials
Output: Validation Link
Description: Validation link is generated using the credentials provided.

R3.2 Send Validation Link

Input: Validation Link
Output: Success/Failure message
Description: Validation link which is generated is sent to the user.

R3.3 Add TA

Input: Validation Link
Output: Success/Error message
Description: The TA is added after he verifies email with the link sent to him/her.

R4 Student Login

Input: Student Credentials
Output: Success/Error message.
Description: Authenticate the user.

R5 Instructor Login

Input: Instructor Credentials
Output: Success/Error message.
Description: Authenticate the Instructor.

R6 TA Login

Input: TA Credentials
Output: Success/Error message.
Description: Authenticate the user.

R7 Course Group

Description: Course group is created and all the events related to the course will be handled here.

R7.1 Add course
Input: Course Name
Output: Success/Error message.
Description: Course is added to the list of courses if successfully created.

R7.2 Course Info
Input: Course Name
Output: Details of the course
Description: We get the details of the requested course.

R7.3 Add Student

R7.3.1 Add student to course
Input: Student details
Output: Success/Error message.
Description: Student is added to list of students in the course.

R7.4 Add TA

R7.4.1 Add TA to course
Input: TA details
Output: Success/Error message.
Description: TA is added to list of TAs in the course.

R7.5 Live discussion rooms

R7.5.1 Live discussion room creation
Input: Discussion room name
Output: Chat room for discussion.
Description: Chat Room will be created and can be accessed by course members for discussions related to the course.

R7.5.2 Send message
Input: Message content
Output: Success/Error message
Description: Message is sent to the discussion room.

R7.5.3 Receive message
Input: Chat room details
Output: Message Content
Description: Message is received and displayed.

R7.5.4 Make group admin
Input: Participant details
Output: Success/Error message
Description: The participant is given admin privileges.

R7.5.5 Dismiss from admin
Input: Participant details
Output: Success/Error message

Description: The participant is exempted from admin privileges.

R7.6 Remove Student
Input: Student details
Output: Success/Error message.
Description: Success if removed by Instructor and the student details are valid. Student is removed from list of students in course.

R7.7 Remove TA
Input: TA details
Output: Success/Error message.
Description: Success if removed by Instructor and the TA details are valid. TA is removed from list of TA's in course.

R8 Examination

R8.1 Create Exam

R8.1.1 Set Timings
Input: Start Time and End Time
Output: Success/Error message.
Description: Start time and the end time of the examination will be set.

R8.1.2 Upload Question List
Input: Question List
Output: Success/Error message.
Description: The questions are uploaded.

R8.1.2.1 Upload MCQ
Input: Question, Options, Answer
Output: Success/Error message.
Description: MCQ question is created with the given options.

R8.1.2.2 Upload Single Word Answer Questions
Input: Question, Answer
Output: Success/Error message.
Description: The questions are uploaded.

R8.1.3 Jumble Questions
Input: Exam details
Output: Success/Error message.
Description: Question order gets jumbled.

R8.1.4 Upload Answer List
Input: Answer List
Output: Success/Error message.
Description: The answers are uploaded.

R8.2 Cancel Exam

Input: Examination Details
Output: Success/Error message.
Description: Examination will be cancelled.

R8.3 Start Exam

Input: Exam details
Output: Exam will be started
Description: The system shall allow the candidate to start a exam at particular duration of time.

R8.4 End Exam

Description: The exam is evaluated and the student gives feedback about the examination.

R8.4.1 Evaluate Exam

Input: Student Responses and Answer List
Output: Performance report of student
Description: The exam is evaluated by matching the student responses with the answer key uploaded by the instructor.

R8.4.2 Feedback by student

Input: Student and paper details.
Output: Feedback is submitted.
Description: Candidate will give feedback to the instructor after the exam is over. The feedback will be based on the quality of questions and concepts used.

R8.5 Examination Statistics

R8.5.1 Student's Score

Input: Examination and Student Details
Output: Examination Score of student
Description: The student gets to know the requested score.

R8.5.2 Average Score

Input: Examination Details
Output: Average Score
Description: The student gets average score of the examination.

R8.5.3 Highest Score

Input: Examination Details
Output: Highest Score
Description: The student gets the highest score and scorer of the examination.

R8.5.4 Marks Distribution Graph

Input: Examination Details
Output: Graph of Marks

Description: The student gets the graph of marks scores in the examination.

R8.5.5 Correct Responses Per Question
Input: Question
Output: Count of Responses
Description: Count of correct responses for each question.

R9 Profile

R9.1 Student Profile

R9.1.1 Student Details
Input: Student Credentials
Output: Student Details
Description: Student details are returned

R9.1.2 Student Calendar
Input: Student and Courses' Details
Output: Calendar
Description: A calendar to display student's exam schedule.

R9.1.3 Completed Exams
Input: Student Credentials
Output: Details of Completed Exams
Description: The student gets the list of his/her completed Exams with the examination details.

R9.1.4 Upcoming Exams
Input: Student Credentials
Output: Details of Upcoming Exams
Description: The student gets the list of his/her Upcoming Exams with the examination details.

R9.2 Instructor Profile

R9.2.1 Instructor Details
Input: Instructor Credentials
Output: Instructor Details
Description: The instructor Details are returned

R9.2.2 Scheduled Exams
Input: Instructor Credentials
Output: Details of Scheduled Exams
Description: The instructor gets the list of his/her Scheduled Exams with the examination details.

R9.2.3 Completed Exams
Input: Instructor Credentials
Output: Details of Completed Exams

Description: The instructor gets the list of his/her completed Exams with the examination details.

R10 Logout
Input: User details
Output: Success/Error message
Description: The user is logged out from the system.

3 Non-Functional Requirements

1 Usability Requirement Gathering - Contextual Inquiry

1.1 Goals

To understand the requirements of college instructors and students for the purpose of automated evaluation and live classroom performance for better control over the usage of time in evaluation and understanding performance.

1.2 Planning Stage

We aimed to observe the users to collect information about the desirable features that they require in this automated exam evaluation application. We contacted people whom we felt to be best suited for the end-user of our application and scheduled an online meeting via Microsoft Teams with them to observe them doing the required tasks.

1.3 Initiate Stage

We met on an online conference with the intended end-users on the scheduled time.

1.4 Execution Stage

In a video call session with a user, we provided him with a test we asked him to perform a demo on how he would give an actual exam during this we deduced the following observations.

Students

1. The student enters his/her credentials and logs in in the exam by clicking on the link.
2. The student is shown the instructions of the exams where he/she is shown the marking scheme and the time of the examination and other important information.
3. The user is now on the first page of the examination.
4. The remaining time will be shown to the user.
5. The user can attempt any question at any time and and move back and forth according to his/her needs.
6. The user is able to see the attempted and the non-attempted questions.
7. It will be helpful if the result is shown as soon as the exam is over.
8. The answer key will be provided to each student after exam of all the students is over.

Instructors

1. We concluded few difficulties that will be faced by the instructors after our observation, creating a plagiarism free online exam

paper for students is a difficult job for instructors.
2. While creating the questions, the questions order must be jumbled so that the order of one word and MCQ types questions is changed.
3. Also as the question paper contains questions of different answer types, such as short answer, long answer, MCQ, one word answer. So instructor must manually set the specific answer type required for questions.
4. Scheduling the exam at a specific time for a large scale of students is difficult as instructors have to reach to each student with question paper access link.
5. Also there was no way for instructors to get an efficient and valid feedback from students as soon as the exam ends.
6. Instructors desire to have a detailed analysis of students performance in exams as soon as the exam ends. Cumulative exam performance reports are also of great use for instructors.

1.5 Close Stage

We thanked the users for their inputs and participation in the online conference.

1.6 Reflect Stage

We generated some ideas for requirements of our software from the observations we recorded.
1. The application we make should be user friendly.
2. A pop up notification before each exam for the student.
3. There should be a feature in the application to mark questions for review.
4. We should also be able to attempt our previous questions.
5. The student should know the number of questions attempted at the end of exam.
6. A feedback form to let the instructor know how the students think the exam was.
7. There should be a chat room for the students to interact with the instructor if they face difficulties during the exam.
8. The instructor should be given the features to distribute the questions randomly and jumble the options.
9. The performance of the student should be shown after each examination.
10. The top scorer should be displayed to other students so that everyone gets motivation to work well.

11. A students timeline should be shown where his/her progress is shown.
12. A peer performance report should be shown to each student so that a student gets to know where his/her is lagging in.

1.7 Affinity Diagram

We performed the contextual inquiry with our classmates and school teachers through Microsoft Teams and Google Meet. First of all, we had a look on the difficulties faced by teachers and students and then we asked them some clarifying doubts. We then created the affinity diagram, shown below, which depicts the issues/requirements that the user faced/needed.