

**Project Report**

**Course: Object Oriented Programming**

**Course Instructor: Adil Majeed**

**Presenter: Muhammad Ali Hassan**

**Roll #: 22i-0541**

**Department: Artificial intelligence**

**Section: C**

**Contents:**

* Introduction
* User Roles
* Data Flow
* Class Diagram
* System Implementation
* Limitations
* Conclusion

**Introduction**

The purpose of this project is to design and implement an examination system that enables the instructor to set, conduct, mark, and result preparation of course evaluations. The system has two user roles, teacher and student. The teacher creates a question bank, sets a quiz/assignment, and sets the date/time for the quiz, while the system automatically marks the quiz, generates a marks report, and other analytics reports that provide useful insights about the evaluation.

**Project Overview**

The Examination System is designed and implemented to provide a platform for instructors to create, conduct, and evaluate quizzes/assignments. The system allows two types of users, teachers and students, to interact with the system. Teachers can create a question bank, set quiz timings, generate quizzes, and view result analytics. On the other hand, students can register for courses, attempt quizzes, and view their scores.

The project includes several steps such as User Creation, Course Offering, Course Registration, Question Bank Preparation, Quiz Setting/Generation, Quiz Conduct, Answer Key Generation, Result Preparation, Quiz Analytics, and Attendance Sheet Generation. The system stores all the data permanently on the disk through file reading and writing.

**User Roles**

The three user roles in the system. The teacher creates the question bank, sets the quiz/assignment, and generates the marks report and analytics. The student attempts the quiz and receives their marks and analytics report.

**Data Flow**

The system generates logins for instructors and students by reading their names from text files. It then offers courses based on the details provided in the course offering list and registers students accordingly. The teacher creates the question bank, generates quizzes, and sets the date and time for the quiz. The system allows registered students to attempt the quiz at the set time, randomizing question sequence and answer options to increase security. After the quiz, the system generates an answer key and prepares the result, including generating an attendance sheet and quiz analytics.

**Class Description**

**First Class**

This class is an abstract class that has a pure virtual function log(). It serves as the base class for other classes in the program.

**Auth Class**

This class is used to implement authentication functionality. It has two functions: log() and log1(). The log() function asks the user to enter their username and password, while the log1() function asks the user to enter their user id and password. The entered values are stored in member variables for further use.

**Check Class**

This class provides functionality to check if the entered student, teacher, or class information exists in the respective text files or not. It has several functions that take input parameters and return a boolean value. These functions are:

check\_class(string abc) - It checks if the entered class exists in the "student.txt" file or not.

check\_student(int roll, int pass) - It checks if the entered student's roll number and password exists in the "student.txt" file or not.

check\_teacher(int roll, int pass) - It checks if the entered teacher's id and password exists in the "teacher.txt" file or not.

check\_subject(int code) - It checks if the entered subject code exists in the "teacher.txt" file or not.

**Paper Class**

This class provides functionality to create a question paper for a particular subject. The make\_paper() function asks the admin to select the subject and enter the total number of questions and time allowed for the exam. The admin can then enter the questions one by one, and the program saves the entered questions to a file named after the selected subject.

**Admin Class**

This class is derived from the first, check, auth, and date\_sheet classes. It serves as the main class for the program, and all the functionalities are implemented in this class. The class has several private member variables and functions, which are discussed below.().

**Private member variables**

admin\_username - A constant string that stores the admin's username.

admin\_password - A constant string that stores the admin's password.

**Private member functions**

add\_student() - A function that allows the admin to add a new student to the "student.txt" file.

add\_teacher() - A function that allows the admin to add a new teacher to the "teacher.txt" file.

make\_datesheet() - A function that allows the admin to create a date sheet for different subjects.

make\_paper() - A function that calls the make\_paper() function of the paper class.

view\_datesheet() - A function that calls the view\_datesheet() function of the date\_sheet class.

menu() - A function that displays the main menu of the program and allows the user to select a task.

**Public member functions**

log() - An overridden function that calls the log() function of the auth class.

log1() - An overridden function that

**System Implementation**

The system was developed using object-oriented programming (OOP) concepts, with C++ as the primary programming language. The system stores data permanently on the disk through file reading/writing. We utilized the PlantUML tool to generate the diagram. The project followed an agile software development methodology, with a focus on iterative development and continuous improvement..

**Limitations**

The system has a few limitations, such as potential security risks, scalability issues, and the need for further refinement in terms of analytics and machine learning techniques for better evaluation and assessment.

**Conclusion**

In conclusion, we have designed and implemented an examination system that enables the instructor to set, conduct, mark and result preparation of the course evaluations. The system allows for two kinds of users, i.e. Teacher and Student, and includes features such as Users creation, Course offering, Course registration, Question bank preparation, Quiz setting/generation, Quiz conduct, Answer Key generation, Result preparation, Quiz analytics, and Attendance sheet generation. The system is efficient and easy to use and can be further improved by adding features such as support for different types of questions, online proctoring, and integration with Learning Management Systems. Overall, the examination system we designed will help the instructor in conducting exams and analyzing the performance of the students.

**Future Work**

There are several potential areas for future work and improvements, such as incorporating more advanced analytics and machine learning techniques for better evaluation and assessment, improving security measures, and enhancing the system's scalability to handle a larger number of users and courses.