# ONLINE EXAMINATION SYSTEM

## Project Proposal Group F

**Group Members :**

Abdullah Tariq  BSCE-24002

Muhammad Umais Ahsan  BSCE-24034

Taha Mughal  BSCE-24043

Aneeka Muzammil  BSCE-24050

Syed Muhammad Moiz Mehdi  BSCE-24052

### Project Title

**Online Examination System using OOP in C++**

### Project Team

- **Team Members:**
  - **Umais:** User Management Module
  - **Taha:** Exam Creation & Management Module
  - **Moiz:** Exam Conducting & Student Attempt Module
  - **Aneeka:** Result Evaluation & Grading Module
  - **Abdullah:** File Handling & Data Persistence Module

## Project Overview

The **Online Examination System** is a **C++-based software** that allows students to take online exams, teachers to create and manage exams, and admins to oversee the entire system. The project incorporates **all Object-Oriented Programming (OOP) concepts**, ensuring modularity, reusability, and efficiency.

---

## Objectives

- Develop a **secure and scalable** online examination system.
- Implement **OOP principles** such as **encapsulation, inheritance, polymorphism, templates, and operator overloading**.
- Ensure **data persistence** using file handling.
- Automate **grading and evaluation** to enhance efficiency.
- Provide an **intuitive user interface** for different user roles (**Admin, Teacher, Student**).

---

## Project Modules & Responsibilities

### 1. User Management Module (Handled by Umais)

- **Manages user registration, authentication, and profile updates**
- **User roles:** Admin, Teacher, and Student

**Classes & Functions:**

- User, Admin, Teacher, Student, UserManager
- void registerUser(string name, string role, string username, string password);
- void loginUser(string username, string password);
- void logoutUser();
- **Template Class:** template <typename T> class ListManager { void addItem(T item); void removeItem(int id); };
- **Operator Overloading:** ostream& operator<<(ostream &out, const User &user);

---

### 2. Exam Creation & Management Module (Handled by Taha)

- **Teachers can create, edit, and delete exams**
- **Supports different types of questions** (MCQs & Descriptive)

**Classes & Functions:**

- Exam, Question, MCQ, Descriptive, ExamManager
- void createExam(int teacherID, string subject, int duration);
- void addQuestion(int examID, string questionText, string type, string answer);

- **Template Class:** template <typename T> class ExamContainer { void addExam(T exam); void removeExam(int id); };
- **Operator Overloading:** Exam operator+(const Exam &e1, const Exam &e2);

---

### 3. Exam Conducting & Student Attempt Module (Handled by Moiz)

- **Handles exam attempts and answer submission**
- **Implements time-based exam control**

**Classes & Functions:**

- ExamSession, AnswerSheet, Timer
- void startExam(int studentID, int examID);
- void submitAnswer(int studentID, int questionID, string answer);
- **Template Class:** template <typename T> class SessionManager { void startSession(T session); void endSession(T session); };
- **Operator Overloading:** ostream& operator<<(ostream &out, const ExamSession &session);

---

### 4. Result Evaluation & Grading Module (Handled by Aneeka)

- **Automates grading and generates student reports**
- **Allows teachers to review student performance**

**Classes & Functions:**

- Grader, Result, ReportCard
- void evaluateExam(int examID);
- void generateReport(int studentID);
- **Template Class:** template <typename T> class GradingSystem { void gradeExam(T result); void displayGrades(); };
- **Operator Overloading:** ostream& operator<<(ostream &out, const Result &result);

---

### 5. File Handling & Data Persistence Module (Handled by Abdullah)

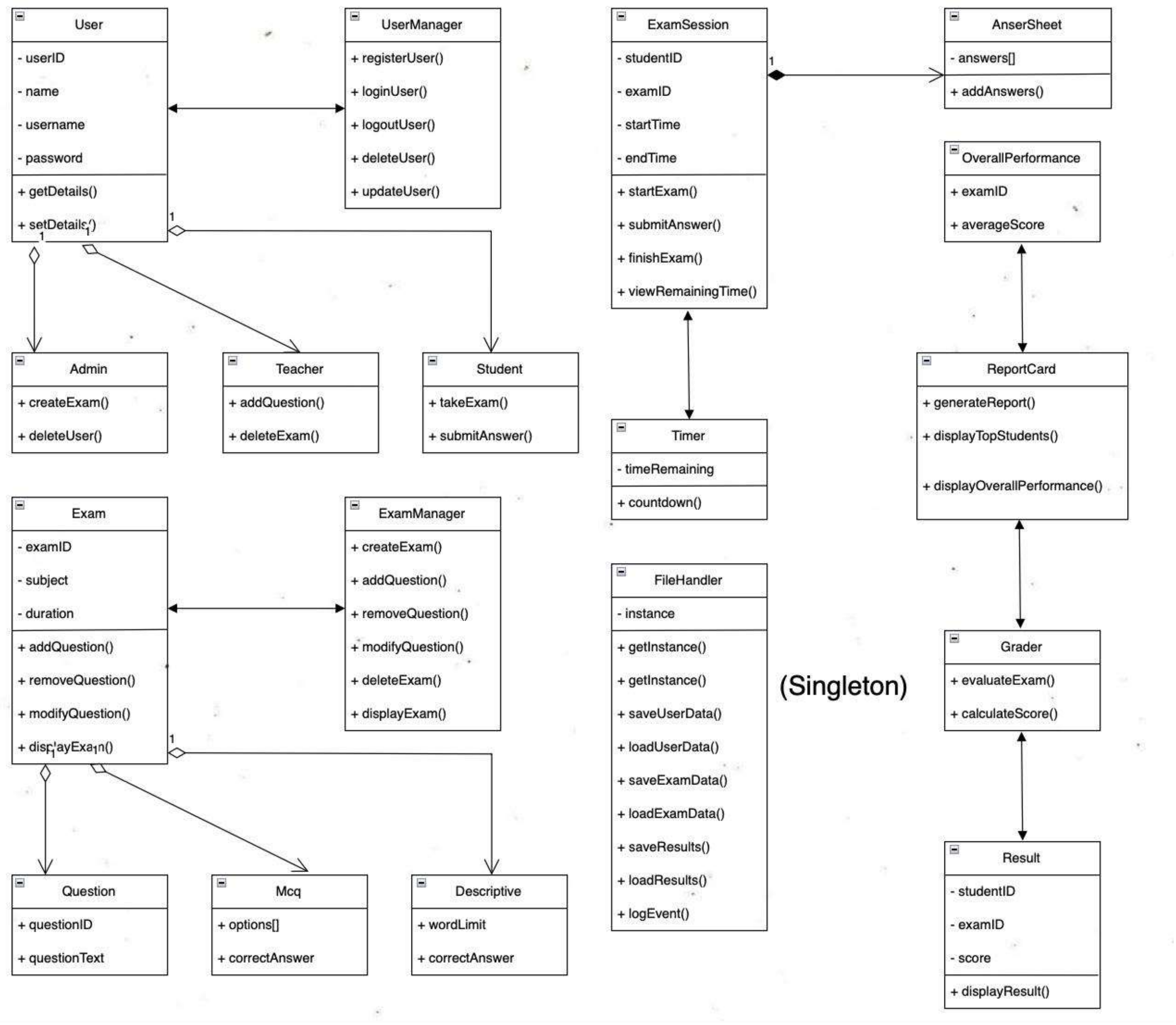
- **Ensures data is saved and retrieved securely**
- **Implements Singleton Pattern for file management**

**Classes & Functions:**

- FileHandler, DataManager
- void saveUserData(string filename);
- void loadUserData(string filename);

- **Template Class:** template <typename T> class FileManager { void saveToFile(T data, string filename); void loadFromFile(string filename, T &data); };
- **Operator Overloading:** ostream& operator<<(ostream &out, const FileHandler &handler);

# UML diagram:


User
- userID
- name
- username
- password
+ getDetails()
+ setDetails()
1
1
1
UserManager
+ registerUser()
+ loginUser()
+ logoutUser()
+ deleteUser()
+ updateUser()
Admin
+ createExam()
+ deleteUser()
Teacher
+ addQuestion()
+ deleteExam()
Student
+ takeExam()
+ submitAnswer()
Exam
- examID
- subject
- duration
+ addQuestion()
+ removeQuestion()
+ modifyQuestion()
+ displayExam()
1
1
1
ExamManager
+ createExam()
+ addQuestion()
+ removeQuestion()
+ modifyQuestion()
+ deleteExam()
+ displayExam()
Question
+ questionID
+ questionText
Mcq
+ options[]
+ correctAnswer
Descriptive
+ wordLimit
+ correctAnswer
ExamSession
- studentID
- examID
- startTime
- endTime
+ startExam()
+ submitAnswer()
+ finishExam()
+ viewRemainingTime()
1
AnserSheet
- answers[]
+ addAnswers()
Timer
- timeRemaining
+ countdown()
FileHandler
- instance
+ getInstance()
+ getInstance()
+ saveUserData()
+ loadUserData()
+ saveExamData()
+ loadExamData()
+ saveResults()
+ loadResults()
+ logEvent()
(Singleton)
OverallPerformance
+ examID
+ averageScore
ReportCard
+ generateReport()
+ displayTopStudents()
+ displayOverallPerformance()
Grader
+ evaluateExam()
+ calculateScore()
Result
- studentID
- examID
- score
+ displayResult()