Sample Projects

	Project	Scenario / Description	Example Key Function (Development Phase)
1.	**Online Library System**	Manage books, student registration, borrowing/returns, fines.	Fine calculation (days late × fine rate).
2.	**Online Food Ordering App**	Students order food from cafeteria via mobile app.	Total bill calculation with discounts.
3.	**Hospital Patient Management**	Register patients, schedule appointments, billing, medical history.	Appointment scheduling (avoid double booking).
4.	**E-Learning Platform**	Course catalog, video lectures, quizzes, certification.	Quiz grading (auto score calculation).
5.	**Smart Parking System**	IoT + mobile app for booking/reserving parking slots.	Slot allocation (nearest available slot).
6.	**Resume Analyzer (AI-based)**	AI tool to match resumes against job descriptions.	Skill matching algorithm (count matched keywords).
7.	**Online Shopping Cart**	E-commerce website for product browsing and purchase.	Cart total with tax & discount calculation.
8.	**Student Attendance System**	Biometric or app-based attendance tracking.	Attendance percentage calculation.

SDLC Case Study Worksheet

Project Title: E-Learning Platform

Team Name: Devinity

Team Members and Roles: Muhammad Arib (PM / Developer)

Haris (Tester / Developer)

Haider (Business Analyst / Requirement Gathering)

Riyan (System Analyst)

Moin (Tester)

1. Requirements Phase

Write 5 functional and 2 non-functional requirements for your project.

Functional Requirements:

1. The Platform shall allow users (students and instructors) to register and log in using their email and password.
2. The Platform shall allow students to browse and enroll in courses from the course catalog.
3. The Platform shall allow students to attempt quizzes and shall automatically grade them based on predefined answers.
4. The Platform shall allow instructors to upload content, create quizzes, and view student performances
5. The Platform shall allow students to download their course completion certificates.

Non-Functional Requirements:

1. The platform shall be accessible via modern web browsers (Chrome, Firefox, Edge, Safari) on both desktop and mobile devices.
2. The platform should be easy enough that any new user can understand and use it within 30 minutes without needing much help.

2. Design Phase

Draw a simple WBS (3 levels) and one UML diagram (use case/class diagram).

Work Breakdown Structure (WBS): (Write as list or sketch tree diagram)

Level 1: E-Learning Platform Development
Level 2: Major Phases

1. Requirement Gathering
2. Design & Development
3. Testing & Deployment
4. Maintenance & Support

Level 3: Major Task Breakdown

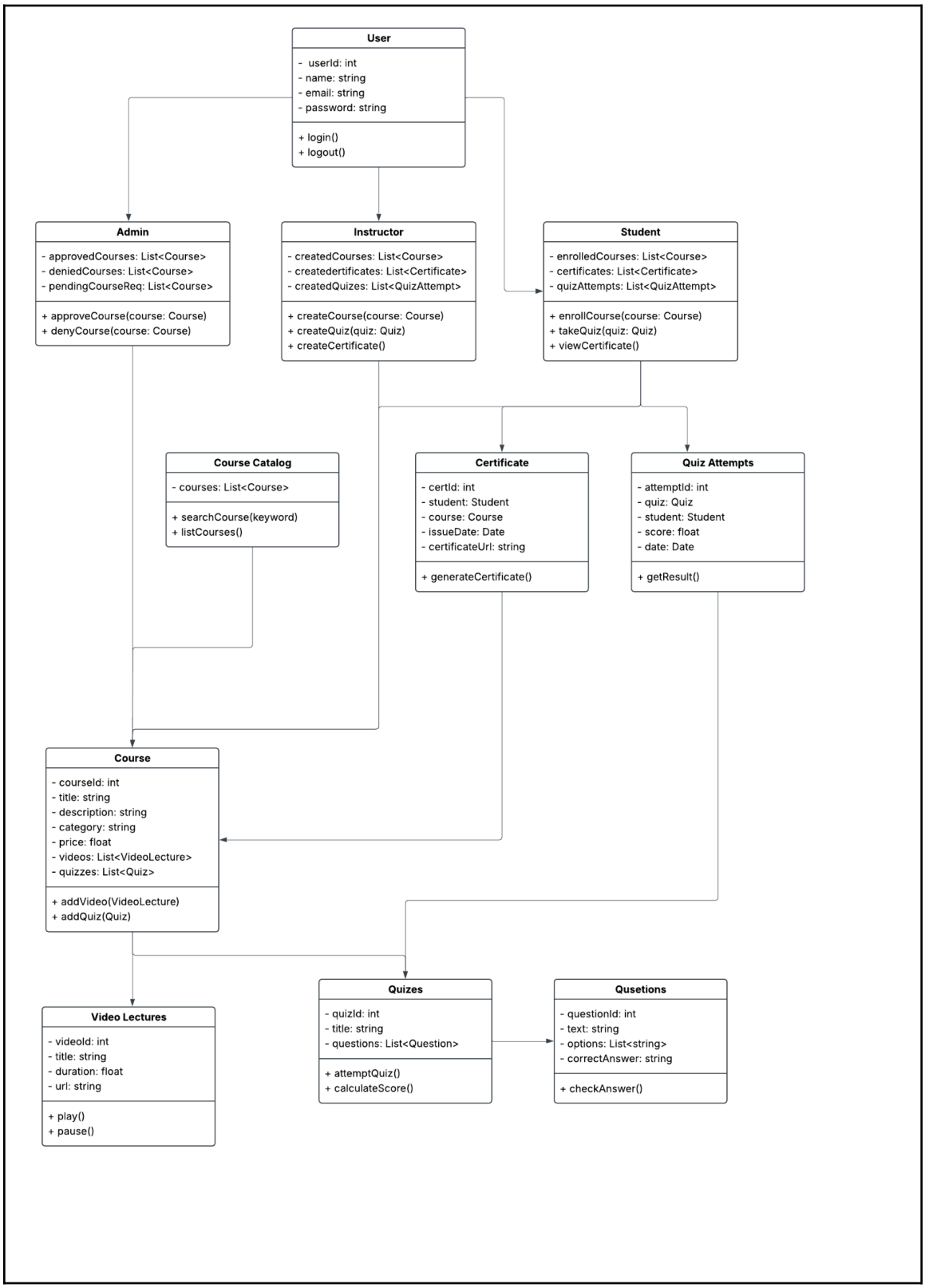
1. Requirement Gathering
   - 1.1 : Stakeholder Meetings
   - 1.2: Define User role (student, instructor, admin)
   - 1.3: Document functional and non-functional requirements

2. Design & Development
   - 2.1: UI/UX Design of the platform
   - 2.2: Database Design
   - 2.3: Backend Development
   - 2.4: Frontend Development
   - 2.5: Integration of quiz and video feature
   - 2.6: Implement User Authentication

3. Testing & Deployment
   - 3.1: Write test cases
   - 3.2: Perform Unit Testing
   - 3.3: Perform integration testing
   - 3.4: Fix bugs and retest
   - 3.5: Deploy project to server
   - 3.6: Final Review and client approval

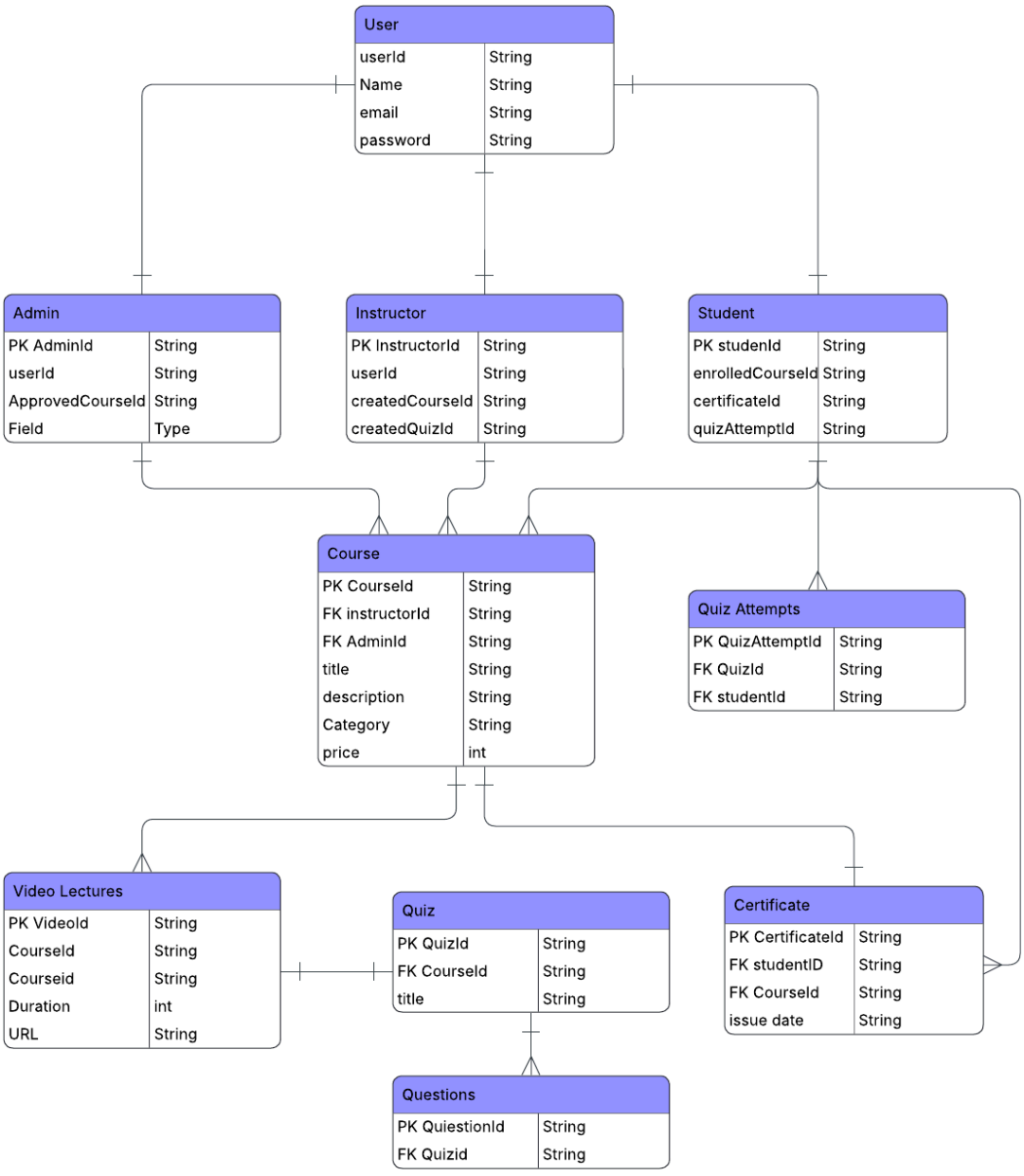
4. Maintenance & Support
   - 4.1: Monitor systems Performance
   - 4.2: Fix post-deployment bugs
   - 4.3: User feedback Collection

UML Sketch (Use Case / Class Diagram/sequence Diagram): (Draw below)



3. Backend Design

Used **LucidChart** for creating UML diagram and ERD.



4. Development Phase

Pseudo-code for Quiz grading (auto score calculation).

```
FUNCTION GradeQuiz
  INPUTS / VARIABLES:
        correctAnswers (array of correct answers)
        studentAnswers (array of student's answers)
        totalQuestions (integer)
        score = 0
        i = 0
        grade
        percentage = 0

  PROCESS:
        FOR i FROM 0 TO totalQuestions {
              IF studentAnswers[i] = correctAnswers[i] THEN
                    score++
        }

        percentage = (score / totalQuestions) * 100

        IF percentage ≥ 80 THEN
              grade = "A"
        ELSE IF percentage ≥ 70 THEN
              grade = "B"
        ELSE IF percentage ≥ 60 THEN
              grade = "C"
        ELSE IF percentage ≥ 50 THEN
              grade = "D"
        ELSE
              grade = "F"

  OUTPUT:

    DISPLAY "Score: ", score

    DISPLAY "Percentage: ", percentage, "%"

    DISPLAY "Grade: ", grade

END FUNCTION
```

5. Testing Phase

Write 3 test cases.

Test Case ID	Description
TC-01	Verify that a registered user can successfully log in with valid credentials.
TC-02	Verify that the system correctly calculates the quiz score and assigns the right grade.
TC-03	Verify that an instructor can create a new course successfully

Test Case ID	Input(s)	Expected Output	Result (Pass/Fail)
TC-01	Email: student@example.com Password: Student@123	System verifies credentials and redirects user to the dashboard/home page.	Pass
TC-02	correctAnswers = [A, B, C, D, A] studentAnswers = [A, B, A, D, C] totalQuestions = 5	Score: 3 Percentage: 60% Grade: C	Pass
TC-03	Course Title: “Introduction to AI” Category: “Computer Science” Instructor: “John” Upload Files: intro.mp4, quiz1.json	System creates a new course record in the database and displays: “Course created successfully.”	Pass

6. Reflection

1. Which SDLC phase was the most challenging? Why?
The Design Phase was most challenging as it required creating clear system architecture, UI/UX layouts, and database structures before development.

2. Which SDLC model (Waterfall, Agile) best fits this project? Why?
The Agile model fits best as it allows flexibility and iterative updates because the platform required continuous feedback from users and iterative improvements

3. How you determine functional and non-functional requirements?
Requirements were identified through stakeholder meetings, analysis of user needs, and reviewing existing e-learning systems to ensure all essential features and performance standards were covered.

7. Attachments

All diagrams (UML and ERD) are directly embedded in this document.