Objective: Develop a system to manage students' grades.

You are the headmaster of a school for magical creatures. Each year, you need to track the student's progress, calculate averages, and keep parents updated. Create a grade management system to store and manage students' grades efficiently, allowing easy input and review. Remember, even a dragon deserves fair grades!

Task:

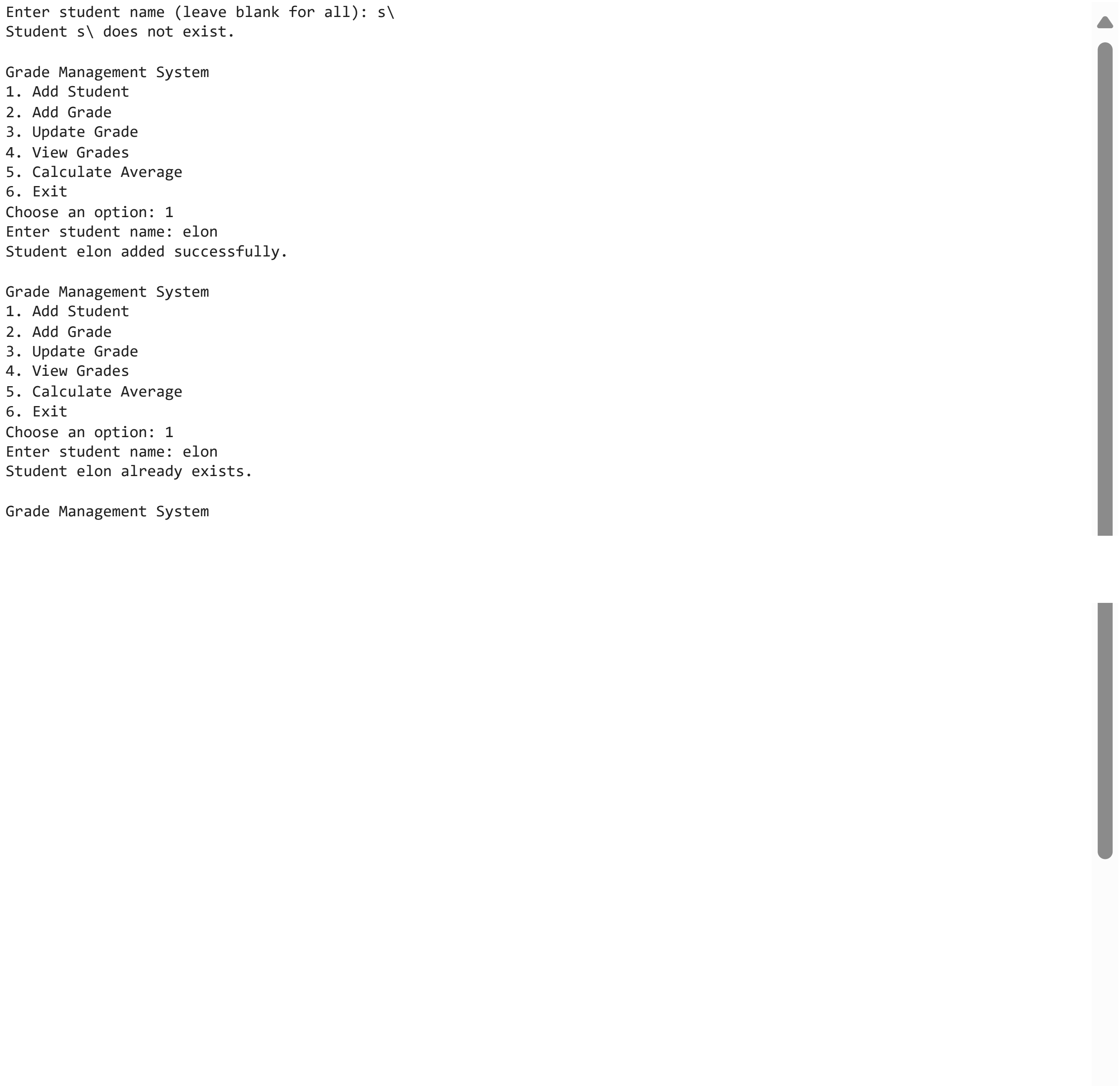
Use a dictionary to store student names and grades. Allow input of new grades, updating existing ones, and viewing a summary. Calculate average grades for individual students and the class.

Key Concepts: Dictionaries, loops, input/output handling.

Solution:-

```
class GradeManager:
    def __init__(self):
        self.students = {}

    def add_student(self, name):
        """Add a new student to the system."""
        if name not in self.students:
            self.students[name] = []
            print(f"Student {name} added successfully.")
        else:
            print(f"Student {name} already exists.")

    def add_grade(self, name, grade):
        """Add a new grade for a student."""
        if name in self.students:
            self.students[name].append(grade)
            print(f"Grade {grade} added for {name}.")
        else:
            print(f"Student {name} does not exist. Please add the student first.")

    def update_grade(self, name, old_grade, new_grade):
        """Update an existing grade for a student."""
        if name in self.students:
            try:
                index = self.students[name].index(old_grade)
                self.students[name][index] = new_grade
                print(f"Grade updated from {old_grade} to {new_grade} for {name}.")
            except ValueError:
                print(f"Grade {old_grade} not found for {name}.")
        else:
            print(f"Student {name} does not exist.")

    def view_grades(self, name=None):
        """View grades for a specific student or all students."""
        if name:
            if name in self.students:
                print(f"Grades for {name}: {self.students[name]}")
            else:
                print(f"Student {name} does not exist.")
        else:
            for student, grades in self.students.items():
                print(f"Grades for {student}: {grades}")

    def calculate_average(self, name=None):
        """Calculate the average grade for a student or the class."""
        if name:
            if name in self.students:
                if self.students[name]:
                    average = sum(self.students[name]) / len(self.students[name])
                    print(f"Average grade for {name}: {average:.2f}")
                else:
                    print(f"No grades recorded for {name}.")
            else:
                print(f"Student {name} does not exist.")
        else:
            total_grades = []
            for grades in self.students.values():
                total_grades.extend(grades)
            if total_grades:
                class_average = sum(total_grades) / len(total_grades)
            print(f"Class average: {class_average:.2f}")
        else:
            print("No grades recorded for the class.")

def main():
    manager = GradeManager()

    while True:
        print("\nGrade Management System")
        print("1. Add Student")
        print("2. Add Grade")
        print("3. Update Grade")
        print("4. View Grades")
        print("5. Calculate Average")
        print("6. Exit")

        choice = input("Choose an option: ")

        if choice == "1":
            name = input("Enter student name: ")
            manager.add_student(name)
        elif choice == "2":
            name = input("Enter student name: ")
            grade = float(input("Enter grade: "))
            manager.add_grade(name, grade)
        elif choice == "3":
            name = input("Enter student name: ")
            old_grade = float(input("Enter old grade: "))
            new_grade = float(input("Enter new grade: "))
            manager.update_grade(name, old_grade, new_grade)
        elif choice == "4":
            name = input("Enter student name (leave blank for all): ")
            if name.strip() == "":
                manager.view_grades()
            else:
                manager.view_grades(name)
        elif choice == "5":
            name = input("Enter student name (leave blank for class average): ")
            if name.strip() == "":
                manager.calculate_average()
            else:
                manager.calculate_average(name)
        elif choice == "6":
            break
        else:
            print("Invalid choice. Please choose again.")

if __name__ == "__main__":
    main()
```

```
Grade Management System
1. Add Student
2. Add Grade
3. Update Grade
4. View Grades
5. Calculate Average
6. Exit
Choose an option: 1
Enter student name: Siddharth
Student Siddharth added successfully.

Grade Management System
1. Add Student
2. Add Grade
3. Update Grade
4. View Grades
5. Calculate Average
6. Exit
Choose an option: 2
Enter student name: Siddharth
Enter grade: 10
Student Siddharth  does not exist. Please add the student first.

Grade Management System
1. Add Student
2. Add Grade
3. Update Grade
4. View Grades
5. Calculate Average
6. Exit
Choose an option: 4
```



```
Enter student name (leave blank for all): s\
Student s\ does not exist.

Grade Management System
1. Add Student
2. Add Grade
3. Update Grade
4. View Grades
5. Calculate Average
6. Exit
Choose an option: 1
Enter student name: elon
Student elon added successfully.

Grade Management System
1. Add Student
2. Add Grade
3. Update Grade
4. View Grades
5. Calculate Average
6. Exit
Choose an option: 1
Enter student name: elon
Student elon already exists.

Grade Management System
```