Project Outline Report: Student Management System Using Python

1. Introduction

The Student Management System is a simple yet effective application designed to manage student data. It helps in storing, updating, deleting, and viewing student information efficiently. This system is useful in educational institutions to maintain student records digitally instead of paper-based records.

2. Objectives

To create a console-based student management system using Python.

To perform CRUD operations: Create, Read, Update, and Delete student records.

To store student details such as ID, Name, Age, and Grade.

To provide a user-friendly menu-driven interface.

3. Tools and Technologies Used

Programming Language: Python 3.x

Development Environment: Any Python IDE or text editor (e.g., VS Code, PyCharm)

Data Storage: In-memory using Python lists and dictionaries (for simplicity)

4. System Design

The system uses a list to store student dictionaries.

Each student record contains:

Student ID (unique)

Name

Age

Grade

The system allows the following operations:

1. Add a student

2. View all students

3. Search for a student by ID

4. Update a student’s details

5. Delete a student record

6. Exit the system

5. Python Code for Student Management System

```
# Student Management System in Python

students = []  # List to store student records

def add_student():
    student_id = input("Enter Student ID: ")
    # Check for unique ID
    for student in students:
        if student['id'] == student_id:
            print("Student ID already exists. Try again.")
            return
    name = input("Enter Student Name: ")
    age = input("Enter Student Age: ")
    grade = input("Enter Student Grade: ")

    student = {
        'id': student_id,
        'name': name,
        'age': age,
        'grade': grade
    }
    students.append(student)
    print("Student added successfully!\n")

def view_students():
    if not students:
        print("No student records found.\n")
        return
    print("List of Students:")
    print("{:<10} {:<20} {:<5} {:<10}".format('ID', 'Name', 'Age', 'Grade'))
    for student in students:
        print("{:<10} {:<20} {:<5} {:<10}".format(student['id'], student['name'], student['age'],
student['grade']))
    print()

def search_student():
    student_id = input("Enter Student ID to search: ")
    for student in students:
        if student['id'] == student_id:
            print("Student Found:")
            print(f"ID: {student['id']}")
            print(f"Name: {student['name']}")
            print(f"Age: {student['age']}")
            print(f"Grade: {student['grade']}\n")
            return
    print("Student not found.\n")

def update_student():
    student_id = input("Enter Student ID to update: ")
    for student in students:
        if student['id'] == student_id:
            print("Enter new details (leave blank to keep current value):")
            name = input(f"Name [{student['name']}]: ") or student['name']
            age = input(f"Age [{student['age']}]: ") or student['age']
            grade = input(f"Grade [{student['grade']}]: ") or student['grade']

            student['name'] = name
            student['age'] = age
            student['grade'] = grade
            print("Student record updated successfully!\n")
            return
    print("Student not found.\n")

def delete_student():
    student_id = input("Enter Student ID to delete: ")
    for i, student in enumerate(students):
        if student['id'] == student_id:
            students.pop(i)
            print("Student record deleted successfully!\n")
            return
    print("Student not found.\n")

def main():
    while True:
        print("----- Student Management System -----")
        print("1. Add Student")
        print("2. View All Students")
        print("3. Search Student")
        print("4. Update Student")
        print("5. Delete Student")
        print("6. Exit")
        choice = input("Enter your choice (1-6): ")

        if choice == '1':
            add_student()
        elif choice == '2':
            view_students()
        elif choice == '3':
            search_student()
        elif choice == '4':
            update_student()
        elif choice == '5':
            delete_student()
        elif choice == '6':
            print("Exiting Student Management System. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.\n")

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

6. Sample Output

```
----- Student Management System -----
1. Add Student
2. View All Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice (1-6): 1
Enter Student ID: 101
Enter Student Name: Alice
Enter Student Age: 20
Enter Student Grade: A
Student added successfully!

----- Student Management System -----
1. Add Student
2. View All Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice (1-6): 2
```

```
List of Students:
ID        Name                 Age   Grade
101        Alice                20    A

----- Student Management System -----
1. Add Student
2. View All Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice (1-6): 3
Enter Student ID to search: 101
Student Found:
ID: 101
Name: Alice
Age: 20
Grade: A

----- Student Management System -----
1. Add Student
2. View All Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice (1-6): 4
Enter Student ID to update: 101
Enter new details (leave blank to keep current value):
Name [Alice]: Alicia
Age [20]: 21
Grade [A]: A+
Student record updated successfully!

----- Student Management System -----
1. Add Student
2. View All Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice (1-6): 2
List of Students:
ID        Name                 Age   Grade
101        Alicia               21    A+

----- Student Management System -----
1. Add Student
```

```
2. View All Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice (1-6): 5
Enter Student ID to delete: 101
Student record deleted successfully!

----- Student Management System -----
1. Add Student
2. View All Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice (1-6): 2
No student records found.

----- Student Management System -----
1. Add Student
2. View All Students
3. Search Student
4. Update Student
5. Delete Student
6. Exit
Enter your choice (1-6): 6
Exiting Student Management System. Goodbye!
```

7. Conclusion

This project demonstrates a basic Student Management System implemented in Python. It covers all basic operations needed to manage student data effectively. This can be enhanced further by adding file/database support, GUI, and advanced validation