Documentation

Short description of the problem:

The University Management System is a program designed to manage students, administrators, and courses in a university setting. The system allows administrators to add, edit, and view information about students, administrators, and courses. Students can view their course enrollments and enroll in new courses. The program aims to simplify the management of university-related data and provide an easy-to-use interface for both students and administrators.

Class structure and description:

1. Person: An abstract base class representing a person in the university. It has two attributes: id_number and name. It has two subclasses: Student and Admin.

2. Student: A class representing a student, with an additional attribute standing to store the student's academic standing (freshman, sophomore, etc.).

3. Admin: A class representing an administrator, with an additional attribute email to store the admin's email address.

4. Course: A class representing a course, with attributes course_code and course_name.

5. Saver: A class responsible for saving updated data to CSV files. It includes methods like save_students, save_courses, save_admins, and save_students_courses.

6. Loader: A class responsible for loading data from CSV files. It includes methods like load_students, load_courses, load_admins, and load_students_courses.

7. Display: A class responsible for displaying data to the user. It includes methods like display_students_and_courses, display_items, display_sorted_courses, view_my_enrollments, and view_available_courses.

8. Utils: A class containing utility methods such as sort_students_by_name, filter_students_by_course, sort_courses_by_name, and generate_unique_id.

9. AddObject: A class responsible for adding new objects (students, administrators, and courses) to the system. It includes methods like add_student, add_course, add_admin, and enroll_student_by_id_and_course_code.

10. Interface: A class responsible for displaying the interface, handling user inputs, and interacting with other classes to perform necessary tasks. It includes methods like show_interface, display_admin_menu, and display_student_menu.

Description of special functions and/or algorithms used:

1. generate_unique_id(cls): A method in the Utils class that generates a unique 10-digit ID number for a Person. It ensures that the generated ID is not already assigned to any existing instances of the class.

2. Sorting and filtering: The Utils class contains methods for sorting and filtering students and courses. For example, sort_students_by_name sorts students alphabetically by their names, while filter_students_by_course filters students based on the courses they are enrolled in.

Instructions on how to start and work with the program:

Ensure that all the required modules and classes are imported properly.
Initialize the main Interface class:
interface = Interface()
Start the program by calling the show_interface method:
interface.show_interface()

Or in this case, just run the main method.

Follow the prompts to enter your ID number. If you are an administrator, you will be presented with the admin menu, which allows you to add students, courses, and other administrators, view lists of students and courses, and perform various sorting and filtering tasks. If you are a student, you will be presented with the student menu, which allows you to view your course enrollments, view available courses, and enroll in new courses. To exit the program, enter '0' at the main menu. The program