This project is a straightforward University Management App built with Avalonia using the MVVM pattern. It lets students and teachers manage and view academic subjects, with all data stored in a JSON file. The app includes login functionality, role-based access, and unit tests to ensure the core logic works as expected. Plus, we've added a course search bar to meet the second bonus point requirement.

Project Overview

1. Structure & Data Models
    - Models: Student, Teacher, Subject classes store the data needed (IDs, names, enrolled/created subjects, etc.).
    - Data Persistence: A data.json file holds all entities. DataService.cs handles loading and saving this JSON data.
    - Views & ViewModels:
        - LoginView / LoginViewModel for user authentication.
        - StudentView / StudentViewModel for viewing and enrolling/dropping subjects.
        - TeacherView / TeacherViewModel for creating, editing, and deleting subjects.
        - MainWindow / MainWindowViewModel as the core window controlling navigation.
2. Functionality
    - Student Role:
        - Log in with valid credentials.
        - Enroll in and drop subjects.
        - View all available subjects and personal enrolled subjects.
    - Teacher Role:
        - Log in with valid credentials.
        - Create, edit, and delete subjects they own.
        - View a list of their subjects.
    - Search Bar for Courses:
        - Both students and teachers can search subjects by name.
        - Implemented in the ViewModels with a filter method that updates the displayed list of subjects.
3. Testing
    - Unit Tests: Located in the UniversityManager.Tests project. We used xUnit to verify core business logic (e.g., enrollment, data loading, and subject creation/deletion).
    - The tests ensure correct handling of user actions and proper updates to the JSON data.

4. Bonus Points

    - First Bonus Point: We covered all main requirements (role-based features, JSON persistence, and at least one test from each testing section).
    - Second Bonus Point: We completed additional unit testing coverage and added the course search bar feature.

Contributions

1. Mihnea Stefan Tudor

    - Implemented the data models (Student, Teacher, Subject) and DataService logic.
    - Ensured JSON loading/saving is integrated with the ViewModels, keeping data in sync across the application.
    - Coordinated application architecture to follow MVVM best practices.
2. Alexandra Corina Gubauceanu

    - Developed the UI in Avalonia XAML, creating LoginView, StudentView, TeacherView, and MainWindow.
    - Set up command bindings and data bindings to connect the Views and ViewModels.
    - Designed the layout and navigation flow, ensuring a clean user experience.
3. Mihai Adrian Briceag

    - Handled unit testing using xUnit, creating tests for enrollment, subject management, and data persistence.
    - Added the search functionality for courses, allowing both students and teachers to filter subjects by name.
    - Assisted in bug fixes and final integration to ensure all components work smoothly together.

Summary

Our University Manager app checks all the boxes for both Student and Teacher roles, with JSON-based data storage and solid unit testing. We also added a course search bar to filter by name, meeting the second bonus point requirement.

This project is the result of great teamwork, showcasing Avalonia MVVM development, strong collaboration, and clean, well-tested code.