# Feedback implementation

I made the application more dynamic by not limiting or minimizing the time period and hours spent. The feedback said if the hours and credits go above certain predefined numbers, the app becomes irrelevant. Therefore, I reversed this approach by removing those limitations to make the application more flexible and applicable to a broader range of scenarios. This way, the application becomes more dynamic and user-friendly, accommodating a wider range of use cases.

I corrected my logic and calculation of the hours and work weeks. My calculation was initially subtracting from the number of weeks instead of the class hours per week, so I fixed the calculation. Also, my work week was not changing because of how I was calculating it so I fixed that as well.

I improved on my UI by using User Control instead of a Window to allow easier navigation for the user.

I presented all my classes on the UML class diagram as I had missed some of them in my part 1.

I switched to the MVVM (Model-View-ViewModel) architectural. MVVM promotes a clear separation of concerns within the application. It divides the code into three components: Model, View, and ViewModel. Each component has a specific responsibility, making the codebase more organized and maintainable.

Model: Represents the data logic of the application. It is responsible for data manipulation, validation, and communication with external services and databases.

View: Represents the UI components. It is responsible for displaying data and capturing user input. In MVVM, the View is passive and delegates its interactions to the ViewModel.

ViewModel: Acts as an intermediary between the Model and the View. It contains the presentation logic, handles user input, and exposes data to the View. It does not have direct knowledge of the View it serves.

Commands: MVVM uses commands to handle user interactions. Instead of directly wiring events to methods in the code-behind, commands in the ViewModel are bound to UI elements, providing a cleaner separation of UI and logic.