Software Engineering Concepts

SE - Lab 03

Topic: MVC Design Pattern.
Student learning time: 12 hours.

Lab Overview

This lab is helps you understand the Model-View-Controller (MVC) design pattern by developing a simple Student Management System using Java.

- You can refer to the MVC Design Pattern Documentation for more details on MVC concepts.
- Example code is available in the lesson PDF file under the name MvcJavaFXExample.zip.

Running the Example Code (10 points)

1. Download and extract the `MvcJavaFXExample.zip` file.
2. Open the project in your preferred Java IDE (e.g., IntelliJ IDEA, Eclipse, Visual Studio Code).
3. Ensure you have JavaFX libraries configured in your project build path.
4. Run the `Main.java` file to launch the Student Management System application.
5. Explore the application to see how the MVC pattern is implemented.
    - Explain how the Model, View, and Controller components interact with each other.
    - Identify the classes that represent each component of the MVC pattern.

Filter student by Email Domain feature (30 points)

Extend the existing Student Management System to include a filter functionality:

1. Implement a filter bar in the View component that allows users to input/select a student's email's domain name (e.g. @gmail.com).
2. In the Controller component, handle the filter logic to filter students based on the input.
3. Update the View to display the filter results dynamically as the user types or selects a domain.
4. Test the filter functionality to ensure it works correctly.

Failed Students feature (30 points)

Extend the existing Student Management System to include a feature that identifies and displays students who have failed:

1. Define a passing GPA threshold (e.g., 2.0).
2. In the Model component, implement a method to check each student's GPA against the passing threshold.
3. In the Controller component, create a method to retrieve the list of failed students.
4. Update the View to display the list of failed students.
5. Test the failed students feature to ensure it works correctly.

Switching Themes feature (30 points)

Extend the existing Student Management System to include a theme-switching functionality:

1. Design two themes (e.g., Light and Dark) with different color schemes.
2. In the View component, add a toggle button or dropdown menu to switch between themes.
3. In the Controller component, handle the logic to switch themes based on user input.
4. Update the View to apply the selected theme dynamically.
5. Test the theme-switching feature to ensure it works correctly.

Student names of your group members (5 points)

Change/Add the student names reflecting your group members in the Student Management System application.

Submission Instructions

1. Ensure all your code changes are committed to a version control system (e.g., Git).
2. Create a ZIP file of your project directory.
3. Submit the ZIP file through the designated submission portal before the deadline.
4. Include a README file with instructions on how to run your modified application and any additional notes. **Note:** Make sure to test all the new features you implement to ensure they work as expected