Updated User Stories and Use Cases

User Stories:

As a student**,** I want to write and execute BasicML programs so that I can become familiar with machine language and gain a better understanding of computer architecture. By using the BasicML simulator, I will develop my coding skills as well as be more prepared for exams.

As an educator**,** I want to analyze the BasicML programs submitted by my students so that I can assess their understanding of key concepts in computer architecture and provide useful and constructive feedback. This will help to make the understanding of this abstract subject easier to grasp and establish a good foundation of computer architecture knowledge.

As a student, I want to be able to load a BasicML file into the UVSim program and execute it, so that I can easily test and debug my BasicML programs without manually entering instructions into the system.

As a student, I want to save BasicML files that I build using UVSim, so that I can revisit, edit, and reuse my work without starting from scratch, ensuring consistency and efficiency in my workflow.

As a student, I want to change the colors of the user interface, so that I can personalize the program to my preferences or adjust the color scheme to reduce eye strain during prolonged usage.

As a student, I want to open multiple windows of UVSim at the same time, so that I can run, compare, or debug multiple BasicML programs simultaneously without having to close or interrupt my current session.

Use Case 1: Load and Execute a BasicML Program

Actor: Student
Scenario: This is the core functionality of UVSim. Users need a reliable way to load and execute BasicML programs to test, debug, and observe how their instructions behave.

1. The user launches the UVSim program.
2. The user selects the "Load File" option.
3. The user browses their system to locate and select the desired BasicML file.
4. UVSim validates the file format and loads the instructions into memory.
5. The user clicks "Execute."
6. UVSim executes the program, displaying outputs or errors in the console.

Use Case 2: Debug a BasicML Program

Actor: Student
Scenario: Debugging capabilities enhance the program's usefulness by allowing users to step through their code, identify errors, and understand program behavior. This improves user learning and troubleshooting.

1. The user enables the "Debug Mode" in UVSim.
2. The user steps through the program line by line.
3. UVSim displays the current instruction, memory state, and accumulator value.
4. The user identifies errors and modifies instructions.
5. The user saves changes and re-executes the program.

Use Case 3: Compare Two BasicML Programs

Actor: User, UVSim Program
Scenario: Comparing programs simultaneously is a powerful feature. It enables analyzing and comparing differences in behavior, performance, and output.

1. The user launches UVSim and opens two separate windows.
2. The user loads one BasicML program in each window.
3. The user executes both programs independently.
4. The user compares outputs and memory usage displayed in each window.