

Functional Requirements (FRs)

1. **Program Input** – The system must allow users to input a UVSim machine code program manually through a text field.
 2. **Program File Handling** – Users must be able to load and save UVSim programs from/to a file in .txt format.
 3. **Execution Control** – The system must provide controls to **start, pause, stop, and reset** program execution.
 4. **Step Execution** – The user must be able to execute a single instruction at a time for debugging purposes.
 5. **Breakpoints** – Users must be able to set breakpoints where execution will automatically pause.
 6. **Memory Management** – The GUI must display **all memory locations** in a structured format and allow users to inspect or modify values.
 7. **Register Display** – The system must visually display the accumulator and instruction counter, updating them in real time.
 8. **Error Handling & Reporting** – The system must detect and display errors such as **invalid instructions, memory overflows, and divide-by-zero errors**.
 9. **Input/Output Simulation** – The application must support **simulated input and output**, displaying results in a dedicated console area.
 10. **Instruction Validation** – When users enter instructions manually, the system must validate them before execution to prevent invalid commands.
 11. **Logging Execution History** – Users must be able to **view a history log** of executed instructions for debugging.
 12. **Undo Last Action** – The system must provide an option to **undo** the last executed instruction before continuing.
 13. **Code Editing Support** – Users must be able to edit their UVSim code directly within the application before execution.
 14. **Dark Mode / Light Mode** – The GUI must support **a toggle between light and dark themes** for better accessibility.
 15. **Help & Documentation** – The system must include an integrated **Help section or tooltips** explaining UVSim commands and GUI functions.
-

Non-Functional Requirements (NFRs)

1. **Performance Efficiency** – The system must execute a single instruction in **under 100 milliseconds**, ensuring smooth user interaction.
2. **Cross-Platform Compatibility** – The application must run on **Windows, macOS, and Linux** with no major modifications.
3. **Accessibility Compliance** – The GUI must be **fully navigable via keyboard shortcuts** and **support screen readers** for visually impaired users.

