**Functional Requirements**

**UVSim**

File load button: allow user to click the file destination through their Operating system

Run System button: allow the user to run the file selected from a button in the GUI

Runs Assembly-Like code: Takes BasicML, machine like code, and runs it like a basic computer might.

**GUI**

Memory display: displays the memory from the file

Accumulator display: displays what is stored in the accumulator

A step-through button: steps through the function and highlights each element of the

register/console/memory as it encounters it in the function

Error pop-up messages: messages can be exited with a click and the user can re-enter, or re-load whatever the case may be

File step-through/debugger: Divide the file into stepwise functions and display to the user which line is about to be run and what the last line’s output was after it was run.

Step-through/Debugger STOP button: Halt the current line of code being run through the step-through function and leave the debugger.

File viewer: A viewport for the user to see the loaded file that is ready to be run

Exit button: An exit button that terminates the program upon clicking

Main Window GUI: Contains the main file viewport, console, and toolbar

Toolbar GUI: Contains main function buttons such as File Load/Save/Run/Delete/Etc.

**IO Operations**

Console input: Allow user to type input codes into the console through a clean GUI box part of the main screen

Console output: Display for the user the output of the machine during and after the code is run.

**Non-Functional Requirements**

Visual appeal: The program is aesthetic but does not sacrifice too much efficiency or usability.

Efficient - simple and quick to use, doesn’t have unnecessary load time (longer than 2 seconds)

Intuitive - the user should not require many instructions to use it, but it does provide instructions that are easy to follow