**High-Level Functionality**

A program designed to create a data repository and based on what each piece of data contains, it will perform several data management operations. Some of these operations include, user data entry, access to data, arithmetic functions, and basic conditional branching.

**User Stories**

Daniel is a student, who is looking to perform basic arithmetic calculations, so he can work on a school project.

Tabitha is a data entry clerk, who is looking to manage customer points, so they can do their job efficiently and satisfy the customer.

Michael is a device manager, who is in charge of tracking device check-in and check-outs. He is also color blind and can’t see the color green.

**Use Cases**

Actor: User

System Goal: Store the accumulator result in memory

Preconditions: User has finished an arithmetic operation

System stores the value in the accumulator at the specified memory address. If the memory address is out of range, the system calls an error.

System retrieves the stored value from memory and prints it to the terminal. Store User Input in Memory

Actor: User

System Goal: Store user input in memory

Preconditions: User wants to store a value in memory

System prompts the user for input and validates it.

System stores the validated input in the specified memory address.

Add or Subtract Number in Memory from Accumulator

Actor: User

System Goal: Perform addition or subtraction between memory and accumulator Preconditions: User wants to add or subtract a value in memory from the accumulator System retrieves the value from memory at the specified address.

System performs the addition or subtraction operation between the value in memory and the accumulator.

System stores the result in the accumulator.

Divide or Multiply Accumulator by Value in Memory

Actor: User

System Goal: Perform division or multiplication between accumulator and memory Preconditions: User wants to divide or multiply the value in the accumulator by a value in memory

System retrieves the value from memory at the specified address.

System performs the division or multiplication operation between the value in memory and the accumulator.

System stores the result in the accumulator.

Handle Large Arithmetic Results

Actor: System

System Goal: Handle large arithmetic results

Preconditions: The result of an arithmetic operation exceeds four digits

System handles the large result appropriately, such as by truncating, rounding, or displaying an error message.

Branch to a Distant Location

Actor: User

System Goal: Perform a branch to a distant location

Preconditions: User wants to branch to a location far away from the instruction pointer System performs the branch instruction to the specified location.

Branch to a Distant Location if Accumulator is Negative

Actor: User

System Goal: Perform a branch to a distant location if the accumulator is negative Preconditions: User wants to branch to a location far away from the instruction pointer if the accumulator is negative

System checks if the accumulator is negative.

If the accumulator is negative, system performs the branch instruction to the specified location. Branch to a Distant Location if Accumulator is Zero

Actor: User

System Goal: Perform a branch to a distant location if the accumulator is zero Preconditions: User wants to branch to a location far away from the instruction pointer if the accumulator is zero

System checks if the accumulator is zero.

If the accumulator is zero, system performs the branch instruction to the specified location. Halt the Program

Actor: User

System Goal: Halt the program

Preconditions: User wants to stop the execution of the program

System terminates the program execution.

Actor: User

System Goal: GUI User input

Preconditions: User wants to have GUI input for the operations of UVSim

System allows for a text box to be editable and saves user inputs to UVSim operations

System runs user inputs through UVSim

Actor: User

System Goal: GUI Output

Preconditions: User wants to have GUI output for the operations of UVSim

System allows for a text box to be viewable to the user for the output of UVSim.

Actor: User

System Goal: GUI Load file

Preconditions: User wants to use the GUI to load files into the UVSim VM.

System allows the user to click a load file button to select user files to load and view into the UVSim GUI.

Actor: User

System Goal: Make file edits/saves to user files

Preconditions: User wants to make file edits to user files and save them to file

System allows user text changes and is able to save them to file directory, up to 100 lines.

Actor: User

System Goal: Theme changes

Preconditions: User wants to make theme changes to the user preferred color scheme other than UVU’s traditional colors.

System allows for a clickable button to select new colors and save them to the gui and view immediately.

Actor: User

System Goal: Cut/Copy/Paste functions

Preconditions: User wants to be able to paste lines of new code to be saved and run through the UVSim VM.

System allows for cut/copy/paste functions in the memory editor view.

Actor: User

System Goal: Convert Four-Letter Words to Six-Letter Words

Preconditions: User wants to be able change all lines of memory to six letters without doing it manually line by line in UVSim VM.

System allows for automatic conversion with the click of a button.

Actor: User

System Goal: Save file as new file

Preconditions: User wants to be able to save the current working file as a new file in a certain location with a certain name.

System allows for SaveAs button that lets the user select the new file location and name.