**User Stories**

User Story 1:

As a computer science student, I need a software simulator so that I can learn and understand the basic operations of a simple virtual machine, including loading and storing values in memory, performing arithmetic operations, and branching to different parts of the program, enhancing my learning, and preparing me for advanced topics in computer science.

User Story 2:

As a computer science teacher, I need a software simulator so that I can demonstrate the fundamental concepts of computer architecture and programming to my students in an interactive and hands-on manner, allowing them to write and execute programs on a virtual machine and observe the changes in memory and the accumulator.

**Use Cases**

Branch

Actor: Programmer

System: UV Sim

Goal: Jump to different instruction

1. Programmer inserts branch instruction into program
2. Programmer loads program into UVSim
   1. Error: branch address is negative
      1. System stops loading program, outputs error message, and terminates
3. System runs program and reaches branch instruction
4. System loads program counter with branch address
5. System jumps to new address immediately

Branchneg

Actor: Programmer

System: UV Sim

Goal: Jump to different instruction if value in accumulator is negative

1. Programmer inserts branchneg instruction into program
2. Programmer loads program into UVSim
   1. Error: branch address is negative
      1. System stops loading program, outputs error message, and terminates
3. System runs program and reaches branch instruction
4. System reads accumulator
5. If accumulator contains negative value, system loads program counter with branch address
6. System moves to address in program counter

Branchzero

Actor: Programmer

System: UV Sim

Goal: Jump to different instruction if value in accumulator is zero

1. Programmer inserts branchzero instruction into program
2. Programmer loads program into UVSim
   1. Error: branch address is negative
      1. System stops loading program, outputs error message, and terminates
3. System runs program and reaches branch instruction
4. System reads accumulator for a positive zero value
5. System reads accumulator for a negative zero value
6. If accumulator contains any zero value, system loads program counter with branch address
7. System moves to address in program counter

Halt

Actor: Programmer

System: UV Sim

Goal: Terminate execution of instructions

1. Programmer inserts halt instruction into program
2. Programmer loads program into UVSim
   1. Error: program does not contain any halt instructions
      1. System stops loading program, outputs error message, and terminates
3. System runs program and reaches halt instruction
4. System immediately terminates execution of instructions

ADDITION

1. The UVSim begins iterating through its memory contents

2. An addition operation is reached in the memory – “3016”

3. The system retrieves the value in memory location 16

4. The accumulator is incremented by the retrieved value

Subtraction

1. The UVSim begins iterating through its memory contents

2. A subtraction operation is reached in the memory – “3199”

3. The system retrieves the value in memory location 99

4. The accumulator is decremented by the retrieved value

Multiplication

1. The UVSim begins iterating through its memory contents

2. A multiplication operation is reached in the memory – “3345”

3. The system retrieves the value in memory location 45

4. The accumulator is multiplied by the retrieved value

Division

1. The UVSim begins iterating through its memory contents

2. A division operation is reached in the memory – “3209”

3. The system retrieves the value in memory location 09

4. The accumulator is divided by the retrieved value

Load

Actor: Programmer

System: UVSim

Goal: Load a word from memory into the accumulator

1. Programmer inserts load instruction into program
2. Programmer loads program into UVSim
3. System runs program and reaches load instruction
4. System reads operand from instruction
5. System loads word from memory address specified by operand into accumulator
6. System moves to next instruction

Store

Actor: Programmer

System: UVSim

Goal: Store the value in the accumulator into a memory location

1. Programmer inserts store instruction into program
2. Programmer loads program into UVSim
3. System runs the program and reaches the store instruction.
4. System reads the value currently in the accumulator.
5. System stores the value from the accumulator into the specified memory location.
6. Execution continues with the next instruction in the program.

Read

Actor: Student (end user)

System: UVSim

Goal: Read a word from the keyboard into a specific location in memory.

1. Create file with read instruction

2. Load a file with instructions into the program

3. System reads the file and expects a user input

4. User types an input

5. Program accepts the input and saves it into a memory location specified by the instruction

Read

Actor: Student (end user)

System: UVSim

Goal: Write a word from a specific location in memory to screen location

1. Create file with read instruction

2. Load a file with instructions into the program

3. Program reads the file and expects a memory location

4. Program retrieves value at that location

5. Program displays the value