Team Meeting Log

|  |  |  |  |
| --- | --- | --- | --- |
| Team Name | Meeting Log # | Date | Duration  1 hour |
| Team Members | Name | Contribution | Signature |
| Connor Morley | Documentation | Connor Morley |
| Jonathan Kim |  | Jonathan Kim |
| Julia Hoffmann | Class Diagram | Julia Hoffmann |
| Bryan Edman | Code Organization |  |
| Topics Discussed | | * Functional requirement documents * breaking things up into MVC | |
| Obstacles Encountered | | * how to delegate design vs. coding * missing team member | |
| Finished Items | | * 4 documents of functional requirements down to 2 * Assigned Class Diagram to Julia * Assigned design/ restructuring of code to Brian * Assigned library documentation to Connor * Assigned essay to Jon | |
| Unfinished Items | | * backlogs * new readme? | |
| Notes | | * worked out Functional requirements | |

Condensed FR doc 1

1. The system must provide a space for the user to input BasicML instructions.

2. The system must provide an accumulator which will act as a register that holds a value.

3. The system must provide a memory space with 100 locations.

4. The user should be able to perform simple math operations on the value in the accumulator with the value in a specific memory location.

5. The system should be able to prompt the user for an integer to read into memory.

6. The user should be able to store a BasicML instruction in a memory location.

7. The system could provide a brief header to introduce the basics of the system to the user

8. The system should execute the user program immediately after it is written to completion.

9. The program should jump to a specific place in memory, based on value in accumulator

10. The program should output what is in the accumulator once the instructions are read

11. The program should output all the values in memory once the instructions are read

12. The system should allow the user to store a word from the accumulator into memory.

13. The user should be able to signal to the system in some way to stop writing the program.

14. The program should output what is stored in a specific memory location

15. The user must be able to use a keyboard to input data into and operate the system.

Condensed FR Doc 2

FR1 System shall displays welcome message in the UVSim Prototype 1

FR2 System shall hold in memory up to 100 instructions

FR3 System shall read first two digits of the four digits of user’s current memory address and compares it to instruction

FR4 System shall allow a full program to be entered before running said program.

FR5 System shall distinguish between word instruction and operand (memory location)

FR6 System shall write any word from memory to the screen

FR7 System shall load a word from memory into accumulator

FR8 System shall store a word from accumulator into memory

FR9 System shall add a word from accumulator and a word in memory saving result in accumulator

FR10 System Shall subtract a word in memory from a word in accumulator

FR11 System Shall divide a word in accumulator by a word in memory

FR12 System Shall multiply a word in memory by a word in accumulator

FR13 System shall branch from operating on an instruction to an instruction in another memory location.

FR14 System shall branch to new memory location only if accumulator is equal to zero

FR15 System Shall Output Accumulator calculation and display sections of: Accumulator, Instruction Counter, Instruction Register, Operation Code, and Operand