**Design Document**

**User Stories**

1. As an educational client, I want the software to be easy to use, so that the students who are using it can more easily learn from it and will want to use my product.
2. As a computer science student, I want to be able to know why my BasicML isn’t working if there’s an error, so that I can still use the program and see at least some of my code working.

**Use Cases**

1. System will read a word from the keyboard into a location in memory.
2. System will write a word from a location in memory to the screen.
3. System will load a word from a location in memory into the accumulator.
4. System will store a word from the accumulator into a location in memory.
5. System will add a word from a location in memory to the word in the accumulator.
6. System will subtract a word from a location in memory from the word in the accumulator.
7. System will divide the word in the accumulator by a word from a location in memory.
8. System will multiply a word from a location in memory to the word in the accumulator.
9. System will branch to a location in memory.
10. System will branch to a location in memory if the accumulator is negative.
11. System will branch to a location in memory if the accumulator is zero.
12. System will pause the program when Halt is called.
13. System will provide errors for infinite loops in branching.
14. System will provide errors for overflow.
15. System will ensure that only inputs that are in the (+/-)#### format are accepted.
16. System will allocate 99 locations in memory.