**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.

**Use Cases**

When the user is getting ready to do an arithmetic operation they need the other portion of the problem in the accumulator to be loaded with the expected value.

When the user is finished with an arithmetic operation, they need the answer of the problem that is in the accumulator to be stored in memory with the expected value.

The program will get the address, and access the memory at that location. If it is out of range it will call an error, otherwise it will store the value at that location in a variable, then print that variable to the terminal.

The program will get the address, and access the memory at that location. If it is out of range it will call an error, otherwise it will store the value at that location in a variable, then print that variable to the terminal.

The program will get the address, then ask the user for an input. If the input is invalid it will ask again, then it will store the input into memory.

The program will get the address, then ask the user for an input. If the input is invalid it will ask again, then it will store the input into memory.

When the user wants to add or subtract a number in memory from a location to the value in the accumulator.

When the user wants to divide the value in the accumulator by a specific value in memory or multiply the same two values.

If the program encounters an added, subtracted, divided, or multiplied result that is larger than four digits.

When the user needs to branch to a location far away from the instruction pointer.

When the user needs to branch to a location far away from the instruction pointer only if the accumulator is negative.

When the user needs to branch to a location far away from the instruction pointer only if the accumulator is zero.

When the user needs to halt the program.