**Meeting Report**

**Date:** February 3, 2025

1. **Schedule**:
   * Weekly meetings are set for Mondays at 5:00 PM, with flexibility for additional discussions as needed.
2. **Programming Language**:
   * The project will be implemented in Python due to its simplicity and documentation support.
3. **Source Control**:
   * GitHub will be used for version control and task management. Ashton will create the repository.
4. **Role Assignments**:
   * Luis is the Scrum Master for this milestone. Responsibilities will rotate in future milestones.
5. **Project Understanding**:
   * The UVSim will simulate a BasicML virtual machine with a focus on executing commands in a console-based app.
   * Memory will be represented as a 100-word array, and an accumulator register will be implemented as a separate variable.
   * Users will provide a program (BasicML commands in a txt file) that the simulator will execute sequentially starting from location 00.
6. **Objectives for the Week**:
   * Complete pseudocode for the main loop and individual commands.
   * Draft user stories and use cases alongside pseudocode development.
7. **Task Assignments**:
   * Luis: Main loop, ADD, and SUBTRACT functions.
   * Ethan: READ, WRITE, LOAD, and STORE functions.
   * Ashton: BRANCH, BRANCHNEG, and BRANCHZERO.
   * Milly (if participating): DIVIDE, MULTIPLY, and HALT functions.

**Tasks Created**

**Development Tasks**

* **Main Loop**:
  + Implement the main function to iterate through memory, identify instructions, and execute corresponding commands.
* **Command Functions**:
  + **READ (10)**: Input value from the keyboard to memory.
  + **WRITE (11)**: Output a value from memory to the console.
  + **LOAD (20)**: Move a value from memory into the accumulator.
  + **STORE (21)**: Save the accumulator value into memory.
  + **ADD (30)**: Add a memory value to the accumulator.
  + **SUBTRACT (31)**: Subtract a memory value from the accumulator.
  + **DIVIDE (32)**: Divide the accumulator by a memory value.
  + **MULTIPLY (33)**: Multiply the accumulator by a memory value.
  + **BRANCH (40)**: Jump to a specific memory location.
  + **BRANCHNEG (41)**: Jump to a specific memory location if the accumulator is negative.
  + **BRANCHZERO (42)**: Jump to a specific memory location if the accumulator is zero.
  + **HALT (43)**: Terminate program execution.
* **Memory Management**:
  + Simulate memory as a 100-element array.
  + Ensure the accumulator is implemented as a separate variable.

**Documentation Tasks**

* **User Stories**:
  + Draft at least two user stories to describe the simulator's functionality from a user's perspective.
* **Use Cases**:
  + Create 10–15 use cases to define the system's functionality.

**Testing Tasks**

* Plan unit tests for each use case.
* Include both success and failure scenarios for each command.