**Functional Requirements**

1. **Program Loading**
   * The system shall allow the user to load a program from a file.
   * The system shall display a message when a program is successfully loaded.
   * The system shall handle errors if the selected file is not a valid program file.
2. **Program Execution**
   * The system shall execute the loaded program when the "Run" button is clicked.
   * The system shall execute instructions sequentially unless directed by a branch instruction.
   * The system shall handle and display errors for invalid instructions.
   * The system shall halt execution if a halt instruction (opcode 43) is encountered.
   * The system shall prompt the user to continue or halt execution when a halt instruction is encountered.
3. **Input and Output Handling**
   * The system shall prompt the user for input when a read instruction (opcode 10) is executed.
   * The system shall display the output of a write instruction (opcode 11) in the output area.
   * The system shall log and display all executed read and write instructions.
4. **Memory Management**
   * The system shall initialize memory to a default size of 100 locations.
   * The system shall allow reading from and writing to specific memory locations.
   * The system shall reset memory to its initial state upon user request.
5. **User Interface**
   * The system shall provide a button to load a program file.
   * The system shall provide a button to run the loaded program.
   * The system shall provide a button to quit the application.
   * The system shall display all output and messages in a dedicated output area.
6. **Error Handling**
   * The system shall display error messages for invalid file selections.
   * The system shall display error messages for invalid instructions during execution.
   * The system shall handle division by zero errors gracefully.

**Non-functional Requirements**

1. **Usability**
   * The system shall have a user-friendly GUI that is easy to navigate.
   * The system shall provide clear and concise error messages to the user.
2. **Performance**
   * The system shall execute programs efficiently, with minimal delay.
   * The system shall handle large programs without significant performance degradation.
3. **Maintainability**
   * The system shall be modular and follow object-oriented design principles.
   * The system shall be documented with comments and a user manual.