# **The Requirements file for the Java Capstone project- Group 2**

**Simulation Requirements**

* The system simulates energy consumption for multiple devices concurrently, ensuring realistic operations.
* Simultaneous energy consumption scenarios must be tested to validate system behavior under load.
* A TaskSimulator component is responsible for creating and managing task workflows based on configurable input parameters.

**Energy Management Requirements**

* The system balances energy in real-time between multiple sources (e.g., solar panels, batteries) and devices.
* Devices are prioritized based on their importance or energy needs, especially during resource constraints.
* Energy sources dynamically adjust their output to match current demand and supply conditions, ensuring efficient energy usage.

**Device and Source Configuration**

* The system integrates devices such as SmartLight and SmartThermostat alongside energy sources like Battery, SolarPanel, and GridConnection.
* Devices can be configured to set their operational priorities and activation states (on/off).
* Energy sources support customization for output capacity and automatic adjustments based on requirements.

**System Logging and Monitoring**

* The system logs critical events, including:
* Device activations or deactivations.
* Actions related to balancing energy across sources and devices.
* Errors and warnings encountered during operations.
* All error conditions, such as unavailable devices or invalid operations, are recorded for debugging and tracking purposes.
* A detailed log file is maintained to monitor system health and operational integrity.

**UI Requirements**

* The EnergyManagementDashboardUI provides an interactive dashboard for:
* Displaying energy consumption and the status of all available energy sources.
* Monitoring devices and their operational statuses.
* The dashboard includes interactive options to add, remove, and configure devices and energy sources.
* Users can view live system operations and modify configurations seamlessly.

**Configuration Management**

* The system initializes settings using a configuration file to ensure proper setup.
* It supports updating configurations during runtime without requiring a system restart, ensuring uninterrupted operations.

**Exception Handling**

* The implementation includes custom exceptions, such as:
* DeviceNotFoundException: Raised when a device cannot be found in the system.
* InvalidOperationException: Raised when an invalid action is attempted.
* These exceptions ensure robust error handling and allow the system to recover gracefully from unexpected situations.

**Testing and Validation**

* A comprehensive test suite is implemented to:
* Verify the correctness of the energy balancing logic under different scenarios.
* Validate the proper functioning of configuration file loading and runtime updates.
* Ensure that device and source integration works as expected.
* Tests cover edge cases, including resource constraints and simultaneous device operations.