**Test Procedure Document for Smart Home Energy Management System (SHEMS)**

**Introduction**

This document outlines the proposed test procedures for the Smart Home Energy Management System (SHEMS). It details the testing of individual components as well as the integration testing sequence to ensure a robust and reliable system.

**Component Testing**

**1. Data Collection Service:**

* **Unit Tests:**
  + Verify data is correctly received from IoT sensors.
  + Test handling of various data formats.
  + Simulate sensor failures and ensure the service logs errors appropriately.
* **Performance Tests:**
  + Measure data processing speed under various loads.
  + Ensure minimal latency in data capture and storage.

**2. Data Processing Service:**

* **Unit Tests:**
  + Validate correct parsing and interpretation of raw data.
  + Test calculation accuracy for energy consumption metrics.
  + Ensure error handling for corrupt or incomplete data.
* **Performance Tests:**
  + Benchmark data processing times under different loads.
  + Test scalability by simulating increasing numbers of data inputs.

**3. Control Service:**

* **Unit Tests:**
  + Verify the correct execution of commands sent to appliances.
  + Test scheduling functionality for various time settings.
  + Simulate command failures and ensure proper error logging.
* **Integration Tests:**
  + Ensure seamless communication with IoT devices.
  + Validate that commands result in expected changes in appliance states.

**4. Notification Service:**

* **Unit Tests:**
  + Test notification triggers for various events (e.g., unusual consumption patterns).
  + Verify the delivery of notifications through different channels (email, SMS, app alerts).
* **Load Tests:**
  + Simulate high volumes of notifications and ensure timely delivery.
  + Measure system performance under peak loads.

**5. User Interface Service:**

* **Unit Tests:**
  + Validate the correctness of data displayed on the dashboard.
  + Ensure user interactions (e.g., setting schedules, viewing reports) function as expected.
* **Usability Tests:**
  + Conduct user testing sessions to gather feedback on interface usability.
  + Test compatibility with different browsers and devices.

**6. Database:**

* **Unit Tests:**
  + Verify data integrity and correctness of CRUD operations.
  + Ensure indexing and queries perform efficiently.
* **Backup and Recovery Tests:**
  + Test the reliability of database backup and recovery processes.
  + Simulate database failures and verify data restoration.

**Integration Testing**

**Order of Integration Testing:**

1. **Data Collection and Data Processing Services:**
   * Ensure data flows correctly from sensors to processing.
   * Validate processed data accuracy.
2. **Data Processing and Database:**
   * Confirm processed data is correctly stored in the database.
   * Test retrieval of data for reporting and control purposes.
3. **Database and User Interface Service:**
   * Verify the UI accurately displays data from the database.
   * Ensure real-time updates are reflected correctly on the dashboard.
4. **Control Service and IoT Devices:**
   * Validate command transmission and execution on IoT devices.
   * Test scheduled operations and real-time control.
5. **Notification Service Integration:**
   * Ensure notifications are triggered based on data from the processing service.
   * Verify end-to-end notification delivery to users.
6. **Full System Integration:**
   * Conduct end-to-end testing from data collection to user interaction.
   * Simulate real-world usage scenarios to validate overall system behavior.
   * Perform load and stress testing to ensure system stability under peak conditions.

**Test Procedures**

**1. Unit Testing:**

* Conducted by individual developers during the development phase.
* Utilize automated testing frameworks (e.g., Jest for JavaScript, JUnit for Java).

**2. Integration Testing:**

* Performed after unit testing completion of all components.
* Use integration testing tools and frameworks (e.g., Postman for API testing).

**3. System Testing:**

* Conducted after successful integration testing.
* Involves comprehensive end-to-end testing of the entire system.
* Includes performance, load, and stress testing.

**4. User Acceptance Testing (UAT):**

* Involves real users testing the system in a controlled environment.
* Gather feedback on system usability and functionality.
* Make necessary adjustments based on user feedback before final deployment.

**Conclusion**

The proposed test procedures ensure thorough validation of each individual component and their interactions within the Smart Home Energy Management System (SHEMS). By following a structured testing process, we aim to deliver a reliable, efficient, and user-friendly system.