**Software Testing Requirements Document**

1. **Functional Requirements**
   * **Connectivity**: Ensure stable connection with the Sensors.
   * **Safety**: Verify that the Account System functions correctly and securely.
   * **Correctness**: Validate that the app processes data accurately and estimates body status correctly.
   * **Liveness**: Confirm the ability to provide access to historical data.
2. **Performance Requirements**
   * **Precision**: Achieve high precision in data processing.
   * **Efficiency**: Minimize delay in data processing (aim for short response times).
   * **Resource Use**: Optimize for low power consumption and minimal storage/memory usage.
3. **Performance Attributes**
   * **Quantitative**:
     + Achieve at least 95% Leave-One-Out (LOO) accuracy in offline classification of general human activities using Respeck accelerometer.
     + Maintain real-time activity classification delay of 600ms or less.
   * **Qualitative**:
     + Ensure low memory usage.
     + Keep CPU consumption low during background operation.
4. **Robustness Requirements**
   * **Account System**: Implement robust user separation and prevent unauthorized access.
   * **Database System**: Safeguard against SQL injection and other forms of data breaches.
5. **Accessibility Requirements**
   * **User Interface**: Develop a responsive and user-friendly interface that displays processing results and history.
   * **Error Handling**: Provide clear error messages for illegal actions or errors without causing app crashes.
   * **Sensor Connectivity**: Prevent recognition processes from starting when devices are not connected and inform the user about the sensor's connection status.

**Test Approach**

**Low Memory Usage - Conduct Stress Testing on the App and Monitor Memory Status**

**Appropriateness**: Stress testing is crucial to evaluate how the app manages memory under extreme conditions. Monitoring memory status during these tests will help identify potential memory leaks or inefficiencies in memory usage.

**Responsive and Friendly UI - Sample User Test, Gathering Feedback from Users**

**Appropriateness**: The responsiveness and friendliness of a UI are subjective qualities best assessed by real users. User testing and gathering feedback provide direct insights into the user experience. It's important to ensure that the sample of users is diverse and representative of target audience.

**Accurate Estimation - Test with Additional Test Dataset; Test with Sensors in a Real Environment**

**Appropriateness**: Testing with an additional dataset and in a real environment with sensors is appropriate for assessing accurate estimation. Using an additional dataset ensures model is robust and works well with data that wasn't part of the initial training set. Testing in a real environment with sensors is crucial as it validates that the system works accurately under real-world conditions. This approach helps in identifying issues that may not be apparent in a controlled test environment.

**Detailed Test Plan**

1. **Low Memory Usage Test Plan**

The requirement of memory usage includes keeping the memory usage as low as possible, at the same time avoiding any memory leaking situation. Thus, the test set evolves to include scenarios simulating peak usage conditions, especially when new features are implemented, to continually assess and optimize memory efficiency.

1. **Responsive and Friendly UI Test Plan**

The target of the UI and UX design is to allow user to use the app with as less assistant from user manual as possible, providing user an smooth and intuitive using experience. So the plan is , when a new feature like viewing outstanding orders is introduced, create initial UI mockups and user stories, and then develop UI tests that assess the responsiveness and user-friendliness of this new functionality. Then, iteratively refine these tests based on user feedback and UI adjustments, ensuring that the UI remains intuitive and efficient as the project evolves.

1. **Accurate Estimation Test Plan**

Based on the target of body activity recognition accuracy (machine learning model’s accuracy), the model need to be tested both in controlled virtual environment and in realistic environment. For the virtual test environment, the test will be done by using sample test dataset (not included in the training dataset) and compute a series of evaluation metrics based on the result of the test. The real-environment test will be done by the testing member wearing the sensors, and performing a series of activities to compare the performed gesture and estimated gesture.