**Functional Requirements:**

**1. Dashboard Display**  
1.1.Sensors should be organised logically for better readability and visualisation.  
1.2. Display live sensor data and historical data.  
1.3. Display filtering options for the user to apply on the historical sensor data.

**2. Display Statistics**  
2.1. When specific sensors are clicked on, specific sensor data should be displayed to the user.  
2.2. Statistics include sensor charts, pie charts, minimum, maximum, average values, and anomalies for each sensor.

**3. Implement Signup Process**  
3.1. Upon registration, the user enters their email, temporary password assigned by the admin, password, and confirmation of password.

**4. Implement Login System**  
4.1. Login: Users should be able to log in using email and password after being approved by the admin.  
4.2. Password Security: Implement a hashing technique to securely store passwords in the database.  
4.3. Reset Password: Users should be provided with the option to reset their passwords if they have forgotten theirs.

**5. Define User Roles and Access Privileges**  
**5.1. Production Operators:**  
5.1.1. Must first register to access the system.  
5.1.2. After registration approval from the admin, they can then log in using email and password.  
5.1.3. Can view historical data.  
5.1.4. Can apply filters to refine sensor data displayed to them.  
5.1.5. Can view live sensor data.  
5.1.6. Can view statistical analysis for sensor data.  
5.1.7. Can access the forgot password functionality.  
5.1.8. Can access the user help guide.  
5.1.9. Can apply light/dark mode.

**5.2.Production Managers:**  
Has the same access privileges as production operators but also have admin functionalities including:  
5.2.1. Approving/declining new user registrations.  
5.2.2. Optionally removing/adding sensors.  
5.2.3. Assigning temporary passwords for registration.  
5.2.4. Resetting passwords for users who have forgotten their passwords.

**6. API Development**  
6.1. Interact with database:  
6.1.1. The system must be able to read and process historical sensor data from CSV files stored in the database.  
6.1.2. The system will authenticate user logins by verifying the credentials and existence of the user stored in a separate database table, which includes emails and hashed passwords.

6.2. Simulate Real-Time Data:  
6.2.1. The API should simulate real-time data, either using synthetic data, replaying historical data, or implementing a random number generator. The most suitable approach will be selected based on its effectiveness for the system.  
6.2.2. Simulated real-time data must be displayed and updated live on the dashboard at specific time intervals.

6.3. Filtering: Allow filtering of data by specific time ranges to refine displayed data.

**6.4. Implement Traffic Light System:**  
6.4.1. The Traffic Light System should be integrated as a separate functionality within the API.  
6.4.2. Anomaly Detection: Flag sensor data points with a traffic light system (green, yellow, red) based on deviation from expected values.  
6.4.3. Anomalies will be detected either using static thresholds derived from the statistical analysis of the sensor data, or by integrating the ML model.

**7. Web Application Responsiveness**  
7.1. Device Compatibility: The web application must be responsive and accessible on both desktop and tablet devices.

**8. Machine Learning Model Integration (Optional)**  
8.1. Predict Sensor Values: The provided pre-trained ML model will predict the expected sensor values for a given timestamp, with these predictions being used internally within the traffic light system. They will not be presented to the user as actual sensor data but will instead serve as a reference for system operations.  
8.2. Integration: The model should be integrated into the backend via an API or Python code.

**Non-Functional Requirements:**

**1. Light/Dark Mode (Optional)**  
1.1. Mode Toggle: Implement a toggle to allow users to switch between light and dark modes.

**2. Flexibility for Future Sensor Additions (Optional)**  
2.1. Scalable Model: The system must be able to accommodate the addition of new sensors without requiring major changes or experiencing a drop in performance.  
2.2. Dynamic Sensor Display: The frontend must dynamically adjust to new sensor types without hardcoded changes.

**3. Performance**  
3.1. Fast Response Times: The web application should load quickly and update sensor data in real-time without delays.

**4. Scalability**  
4.1. Expandable Architecture: The system should be designed to easily add more sensors or integrate new features in the future.

**5. Reliability**  
5.1. Consistent Data: Simulated real-time data displayed on the dashboard should be consistent with the actual sensor readings and historical data.

**6. Usability**  
6.1. User-Friendly Interface: The application should have a clear, intuitive interface that is easy to navigate.  
6.2. Accessibility: Ensure the application follows best practices for accessibility (e.g., colour contrast, font readability).  
6.3. Help Sections: Provide user guidance through a help section to support new or less-technical users.

**7. Maintainability**  
7.1. Modular Codebase: The code should be modular and easily maintainable, with clear separation of concerns.  
7.2. Flexible Configuration: The system should allow easy configuration and addition of new sensors or features without extensive rewrites.  
7.3. Adherence to Coding Best Practices: Standard coding practices, such as using clear and descriptive variable names, should be consistently applied to ensure ease of collaboration.

**8. Data Privacy**  
8.1. Anonymity of User Data: Store only the necessary user data (e.g., hashed passwords) and avoid storing sensitive personal information.

**ACTORS:**

### **1. Production Operators:**

Production Operators will access the system after registering and receiving approval from the admin. Their role is primarily to view sensor data, including both real-time and historical data, as well as statistical analysis. They will be able to apply filters to refine the data based on time intervals. Operators can also access the help guide to understand the system’s features and reset their passwords if necessary. Additionally, they can toggle between light and dark modes for better user experience. However, their privileges are limited, and they cannot add or remove sensors, nor approve new user registrations.

### **2.Production Managers:**

Production Managers have all the privileges of Production Operators, with the added responsibility of overseeing user registrations. They have the authority to approve or decline new user accounts and manage sensor configurations by adding or removing sensors. Admins also can assign temporary passwords for new users and reset passwords for existing users. Their elevated privileges allow them to configure the system more flexibly while maintaining user access control, ensuring the system can be adapted to operational needs.

### **3. Platform (Web Application):**

The platform is a user-friendly, web-based application that acts as the central hub for visualising and interacting with sensor data. It features a dynamic and responsive dashboard where users can view both live and historical data from sensors, along with comprehensive statistical analysis. The platform allows users to filter historical data based on specific time intervals to refine the information they see. It supports visualisations like charts and graphs, as well as anomaly detection through a traffic light system, which flags data points based on deviations from expected values. Additionally, the platform is designed to be fully responsive, ensuring seamless access on both desktop and tablet devices. It also offers a light/dark mode toggle for user preference and a help guide to assist users in navigating the system. The platform communicates with the backend API to fetch and display real-time data, process filtering options, and update data at specified intervals. It ensures that users have a smooth and informative experience while interacting with the sensor data.

A diagram of a production manager

AI-generated content may be incorrect.**USE CASE DIAGRAM:**