**Team 23**

**Requirements Specification Document**  
**Interactive Real-Time Visualisation Dashboard for Rakusens**

**1. Introduction**

Rakusens is a traditional food manufacturer based in Yorkshire that is actively modernizing its processes by leveraging AI/Big Data technologies. A key part of this transformation involves developing an **interactive, real-time visualization dashboard** for monitoring temperature data from multiple sensors. By utilizing this dashboard, operators will be able to make informed decisions regarding temperature adjustments, ultimately improving product quality and energy efficiency.

This specification document is based on an initial scenario and discussions with the client, outlining the key functionalities, database requirements, and user interface specifications.

* 1. **Team Expertise**

Our team is constructed with diverse individuals expertising in different roles to ensure a successful real-time visualisation dashboard such as:

* Front-end: HTML, CSS, Javascript, PHP, …
* Back-end: Python, …
* Database: SQL, …
  1. **Rationale of Topic Choice**
* **Increase product quality –** This would allow operators to adjust temperatures in real-time
* **Enhance operation transparancy –** This would allow user to view historical data tracking and manage trends
* **Integrate AI-Driven anomalies –** This would detect temperatures and flag anomalies using machine learning.
* **Increase energy efficiency –** This would detect temperature in real-time which would decrease the used of excessive heating/cooling

**2. Functional (and Non-functional) Requirements**

**2.1 Actors**

The Rakusens Dashboard (RD) will have three types of users:

* **Administrator** – Manages user accounts and system configurations.
* **Operators** – Monitor temperature data, receive real-time alerts, and make adjustments accordingly.
* **General Users** – Have access to view visualizations but with restricted functionality.

The system will provide a user interface with a main menu containing multiple options and will allow for a smooth user experience across various devices.

**2.2 Functions and Their Description**

**Login/Logout for RD Administrators and Operators**

* + Users will log in with a username and password.
  + Credentials will be validated, and unsuccessful attempts will prompt re-entry.
  + Users will log out when ending their session.

**User Registration and Authentication**

* + The administrator can register new operators and users by providing unique usernames and passwords.
  + Users can reset passwords if necessary.

**Real-time Temperature Data Monitoring**

**Data Retrieval**

* + The system will retrieve real-time temperature readings from multiple sensors installed in different production areas.
  + Data will be stored in an SQL database and accessed via an API.

**Interactive Visualisation**

* + Users can view temperature data through **interactive graphs and dashboards**, powered by open-source libraries such as **Plotly.js**.
  + Data will be aggregated by sensor and displayed with historical trends.

**Traffic-Light System for Anomalies**

* + A Machine Learning model will analyse temperature data and flag readings that deviate significantly from normal trends.
  + Anomalies will be **color-coded (green, yellow, red)** based on severity.

**Real-time Alerts and Notifications**

* + Operators will receive **immediate alerts** if a temperature reading falls outside the optimal range.
  + Alerts can be delivered via **web notifications, emails, or SMS** (optional feature).

**Historical Data and Reports**

* + Users can generate **reports on past temperature trends** for analysis and compliance documentation.
  + Export options include **CSV, PDF, and Excel formats**.

**System Performance and Maintenance**

* + The administrator can perform **system health checks**, manage database storage, and optimize data retrieval speeds.

**2.3 Non-functional Requirements**

* **Scalability** – The system should handle increasing amounts of sensor data efficiently.
* **Security** – All data should be encrypted, and role-based access control (RBAC) should be implemented.
* **Cross-Platform Compatibility** – The dashboard should be **responsive** and accessible via desktops, tablets, and mobile devices.
* **Performance** – The system should provide real-time data updates with minimal latency.

**3. Data Description**

**Database Tables:**

1. **Temperature Readings**
   * Sensor ID (Primary Key)
   * Timestamp
   * Temperature Value
   * Status (Normal/Warning/Critical)
2. **Users**
   * User ID (Primary Key)
   * Username
   * Password (Encrypted)
   * Role (Admin/Operator/General User)
3. **Alerts**
   * Alert ID (Primary Key)
   * Sensor ID
   * Timestamp
   * Alert Level (Low/Medium/High)
   * Action Taken

**4. Interface**

**Rakusens Dashboard Main Menu**

* **Administrator**
* **Operator**
* **General User**
* **Stop**

**Administrator Panel**

* **User Management** (Add/Edit/Delete Users)
* **System Settings**
* **View Reports**
* **Logout**

**Operator Dashboard**

* **View Real-time Temperature Data**
* **Acknowledge Alerts**
* **Generate Reports**
* **Logout**

**General User View**

* **View Historical Temperature Trends**
* **Basic Dashboard Access**
* **Logout**

**5. LSEPI and Risk Assessment**

**Legal, Social, Ethical, and Professional Issues:**

* **Data Privacy** – Ensuring compliance with **GDPR** regulations for user authentication and data storage.
* **Security Risks** – Unauthorised access should be prevented through encrypted credentials and/or **multi-factor authentication (MFA)**.
* **Operational Risks** – System failures could lead to incorrect temperature monitoring, potentially affecting food quality. A robust **backup strategy** will be implemented.

**6. Work Plan**

A detailed **Gantt Chart** will outline the project timeline, including milestones such as API development, front-end design, database setup, and system testing.

**7. GitHub**

* **GitHub Repository for the Prototype** – [Link]
* **GitHub for Meeting Minutes** – [Link]
* **GitHub for NDA** – [Link]

**8. Peer Review**

Each team member’s contribution will be evaluated based on assigned tasks, achievements, and effort, using a grading scale of **1 to 10** (where 1 = minimal contribution, 10 = outstanding contribution).

**END**