Cover page

Team 23

Interactive Real-Time Visualisation Dashboard for Rakusens

23039430/1 AFZAL, Ehtasham [eafzal2@bradford.ac.uk](mailto:eafzal2@bradford.ac.uk)   
23015945/1 BISWAS, Kian [k.biswas@bradford.ac.uk](mailto:k.biswas@bradford.ac.uk)   
23040217/1 KHALID NACIRI, Lina [l.khalidnaciri@bradford.ac.uk](mailto:l.khalidnaciri@bradford.ac.uk)   
21007967/3 KHAN, Tayyib [t.khan34@bradford.ac.uk](mailto:t.khan34@bradford.ac.uk)   
23045736/2 NAZAR, Mahad [m.nazar2@bradford.ac.uk](mailto:m.nazar2@bradford.ac.uk)   
23021200/1 QAZI, Areesha [a.qazi5@bradford.ac.uk](mailto:a.qazi5@bradford.ac.uk)   
22008496/2 RASHEED, Adam [a.rasheed6@bradford.ac.uk](mailto:a.rasheed6@bradford.ac.uk)

Software Requirements Specification (SRS)

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements for the Interactive, Real-Time Visualisation Dashboard for Rakusens. This system will provide a web-based solution to monitor and visualise sensor data in real time for both tablets and desktops.

1.2 Scope

The system will include:

* A web-based dashboard for visualizing real-time and historical sensor data.
* Data aggregation and anomaly detection using an ML model.
* User authentication with admin, operator, and general user roles.
* Integration with a PostgreSQL database for historical data storage.
* Traffic-Light System for anomaly detection.
* Real-time alerts and notifications.

1.3 Team Expertise

Our team consists of individuals with expertise in:

* Front-end: HTML, CSS, JavaScript, PHP.
* Back-end: Python, Node.js.
* Database: SQL, PHPMyAdmin, XAMPP.

1.4 Rationale of Topic Choice

* Increase Product Quality – Enables operators to adjust temperatures in real-time.
* Enhance Operational Transparency – Allows users to view historical data and track trends.
* Integrate AI-Driven Anomalies – Uses machine learning to detect temperature deviations.
* Increase Energy Efficiency – Reduces excessive heating/cooling based on real-time detection.

1.5 References

 **Client-Provided Documentation:** System requirements, authentication, data handling, and machine learning integration. Client provided documentation on 7/2/2025

 **Project Proposal:** Outlines the scope, objectives, and key deliverables.

 **Authentication System:**

* User roles: Regular user (production operator) and admin (manager).
* User registration approval by admin.
* Password reset functionality.

 **Data Handling & Visualization:**

* Responsive web-based dashboard for both tablet and PC.
* Real-time and historical sensor data visualization with statistical summaries.
* Traffic Light System for anomaly detection.

 **Real-Time Data & Simulation:**

* No live data stream; real-time data is simulated using probabilistic/statistical methods, replayed past data, or random number generation.
* API structure: SQL database for historical data, simulator for real-time data.

 **Traffic Light System & Machine Learning Integration:**

* ML model provided for anomaly detection.
* Python-based model using the **Prophet** library for time series forecasting.
* API service for predictions and anomaly detection, accessible via REST API.
* Thresholds for green/amber/red anomaly categorization.
* ML model accuracy is **not** part of this project but can be improved in the future.

 **Directory Structure & Data Storage:**

* **Data Folder:** Contains cleaned (line4.csv, line5.csv) and raw uncleaned sensor data.
* **Machine Learning Folder:** Stores ML models, Prophet forecasting models, and an example usage Jupyter notebook.

 **External References:**

* Prophet Time Series Forecasting Library – <https://facebook.github.io/prophet/>

2. Functional (and Non-functional) Requirements

2.1 Actors

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

* Administrator – Manages user accounts and system configurations.
* Operators – Monitor temperature data, receive real-time alerts, and make adjustments accordingly.

2.2 Functions and Their Description

User Authentication

* Login/logout for administrators and operators.
* User registration and authentication by administrators.
* Password reset functionality.

Real-Time Temperature Data Monitoring

* System retrieves real-time temperature readings from multiple sensors.
* Data is stored in an SQL database and accessed via an API.

Interactive Visualization

* Users can view temperature data through interactive graphs and dashboards (e.g., Plotly.js).
* Data is aggregated by sensor and displayed with historical trends.

Traffic-Light System for Anomalies

* ML model flags temperature anomalies using a color-coded system (green, yellow, red).
* Alerts are triggered for deviations from normal trends.

Real-time Alerts and Notifications

* Operators receive alerts via web notifications, emails, or SMS (optional).

Historical Data and Reports

* Users can generate reports on past temperature trends (CSV, PDF, Excel export options).

System Performance and Maintenance

* Admins can monitor system health, manage storage, and optimize data retrieval.

2.3 Non-functional Requirements

* Scalability – Handle increasing sensor data efficiently.
* Security – Implement encryption and role-based access control (RBAC).
* Cross-Platform Compatibility – Responsive design for desktop, tablet, and mobile.
* Performance – Real-time updates with minimal latency.

3. Data Description

3.1 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

4.1 Rakusens Dashboard Main Menu

* Administrator
* General User

4.2 Administrator Panel

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

4.3 General User Dashboard

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

5. LSEPI and Risk Assessment

5.1 Legal, Social, Ethical, and Professional Issues

* Data Privacy – Ensure GDPR compliance for user authentication and data storage.
* Security Risks – Prevent unauthorized access with encrypted credentials and/or multi-factor authentication (MFA).
* Operational Risks – Implement a robust backup strategy to prevent system failures that may affect temperature monitoring.
* User training plan – include training for operators to ensure they are able to use the interface correctly to prevent issues such as incorrect temperatures.

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. All ideas are added to the shared GitHub.

7. GitHub

* GitHub Repository for the Prototype – [https://github.com/Hedgeturd/ep-team-repo.git]
* GitHub for Meeting Minutes – [https://github.com/Hedgeturd/ep-team-repo/tree/9361881878c1c3f48f9e82044f8f8e7a1dd54cc0/team]
* GitHub for NDA – [https://github.com/Hedgeturd/ep-team-repo/blob/28ea73b79c7c3b3008ef5b638192885d31435fac/docs/NDA%20student-CEC%20-%20Ehtasham%20Afzal.pdf]

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).

Kian Biswas – as Secretary Kian was an important role to the team as he voluntarily took charge of monitoring the teams meetings and observing all the roles in the team. He specialised in API working as the backbone of the API. 10/10

Ehtasham Afzal – worked as the team leader, Ehtasham made sure that each member of the team was given an opportunity to include ideas and contribute to the project. He specialised on the front end of the project whilst contributing ideas to API. 10/10

Areesha Qazi – was given the role to work solely on the SQL side of the project. This was vital to ensure that the data that was given to run with the project 10/10

Mahad Nazar – worked on the front end of the project. Mahad specialised in the login ensuring security and worked with SQL. Also, 10/10

Lina Khalid – as the team speaker, Lina’s included her ideas of the project during teams meeting. She specialised mainly on the front end of the project. 10/10

Adam Rasheed – Adam solely worked on the machine learning of the project to ensure that that it would run with the rest of the project. He contributed to teams meetings. 10/10

Tayyib Khan – Tayyib was able to observe the whole team and give key ideas of how the project should be laid out. His main role was to help with the API. 10/10