System Technical Roadmap

ArticuChic Team

April 8, 2025

Contents

1 Document Overview

1.1 Document Purpose

This document details the technical roadmap choices for each component of the system, including the UI Group, System Group, Data Group, and Analysis Group. The document aims to provide clear technical guidance for the development team, ensuring technical compatibility and collaboration efficiency between groups.

1.2 Project Background

This project is a comprehensive system with four main components working together: User Interface Group (UI), System Group, Data Group, and Analysis Group. The system allows users to access services through web pages, perform Bluetooth connections, data collection, page access, and login functions. The collected data is sent to the server via the network, where the server collects and stores the data, and simultaneously calls the Analysis Group's Java code for streaming analysis. The analysis results are stored and returned to the client for display.

1.3 Target Audience

This document is intended for the following readers:

- Project managers: To understand the project's technical architecture and roadmap
- Development team members: To understand the technical choices and implementation details of each group
- Testers: To understand the system's technical architecture and develop testing strategies
- Operations personnel: To understand system deployment and maintenance requirements

2 System Architecture Overview

The system adopts a layered architecture, mainly divided into frontend layer, service layer, data layer, and analysis layer. The responsibilities of each group are as follows:

- **UI Group**: Responsible for user interface design and implementation, providing user interaction functionality
- **System Group**: Responsible for system core architecture, API implementation, third-party service integration, system deployment, performance optimization, security assurance, and monitoring systems
- **Data Group**: Responsible for data model design, database implementation, data processing workflows, and data quality assurance
- **Analysis Group**: Responsible for data analysis algorithms, report generation, data visualization, and decision support models

The system data flow is as follows:

1. Users access the system through web pages
2. Web pages perform Bluetooth connection and data collection
3. Collected data is sent to the server via the network
4. The server collects and stores the data
5. The server calls the Analysis Group's Java code for streaming analysis
6. Analysis results are stored and returned to the client
7. The frontend renders the display data provided by the Analysis Group

3 UI Group Technical Roadmap

3.1 Frontend Technology Stack

- **Core Languages**:
  - HTML5: Page structure
  - CSS3: Page styling
  - JavaScript: Interaction logic
- **Frontend Frameworks**:
  - Vue.js/React: Building user interfaces
  - Bootstrap/Tailwind CSS: Responsive layout and UI components
- **Web Bluetooth API**:
  - Web Bluetooth API: Implementing browser connection with Bluetooth devices (official demo)
- **Data Visualization**:
  - Chart.js/D3.js: Charts and visualization
  - ECharts: Complex data visualization
- **State Management**:
  - Vuex/Redux: Frontend state management
- **Build Tools**:
  - Webpack/Vite: Module bundling and development server
  - Babel: JavaScript transpilation

3.2 Frontend Functionality Implementation

1. **User Authentication**:
   - JWT (JSON Web Token) authentication
   - OAuth 2.0 (if third-party login is needed)
2. **Bluetooth Connection**:
   - Using Web Bluetooth API to implement device discovery and connection
   - Implementing data collection and transmission
3. **Data Collection**:
   - Real-time data collection and transmission
   - Data caching and resumable transmission
4. **Data Display**:
   - Real-time data visualization
   - Historical data query and display
   - Analysis results display
5. **Responsive Design**:
   - Adapting to different devices (desktop, tablet, mobile)
   - Supporting different screen sizes

4 System Group Technical Roadmap

4.1 Backend Technology Stack

- **Core Language**:
  - Java: Main backend language
- **Web Frameworks**:
  - Spring Boot: Rapid development of web applications
  - Spring MVC: Handling web requests
- **API Design**:
  - RESTful API: Following REST architecture style
  - OpenAPI/Swagger: API documentation and testing
- **Security Framework (Optional)**:
  - Spring Security: Authentication and authorization
  - JWT: Stateless authentication

- **Message Queues**:
  - Apache Kafka: Streaming data processing
  - RabbitMQ: Message passing
- **Caching**:
  - Redis: High-performance caching
- **Monitoring and Logging**:
  - ELK Stack: Log collection and analysis
  - Prometheus + Grafana: System monitoring
- **Deployment and Containerization**:
  - Docker: Containerizing applications
  - Kubernetes: Container orchestration
  - Jenkins/GitLab CI: Continuous integration/deployment

4.2 System Functionality Implementation

1. **API Services**:
   - Providing RESTful API interfaces
   - Implementing API version control
   - Providing API documentation and testing tools
2. **Data Reception and Processing**:
   - Receiving data sent from the frontend
   - Forwarding data to Data Group and Analysis Group services
3. **Third-party Service Integration**:
   - Evaluating and selecting third-party services
   - Implementing API encapsulation
   - Providing unified interfaces
4. **System Deployment**:
   - Designing CI/CD workflows
   - Configuring automated build and testing
5. **Security Assurance**:
   - Identity authentication and authorization
   - Data encryption
   - Security auditing and monitoring

6. **Monitoring and Logging**:
    - System monitoring metrics design
    - Log collection and analysis
    - Alert mechanism implementation

5 Data Group Technical Roadmap

5.1 Database Technology Stack

- **Core Database Engine**:
    - MySQL Community Server 8.0.x: Main relational database
- **Database Interaction Layer**:
    - JDBC (Java Database Connectivity): Database connection
    - MySQL Connector/J: MySQL JDBC driver
    - HikariCP: High-performance connection pool (optional)
- **Database Migration Tools**:
    - Flyway: Database version control
- **Data Processing Tools**:
    - Apache Commons DbUtils: Simplifying JDBC operations
    - Spring JDBC (JdbcTemplate): Simplifying database operations
- **Backup and Recovery**:
    - mysqldump: Database backup
    - Binary logs: Incremental backup

5.2 Data Functionality Implementation

1. **Data Model Design**:
    - Designing database table structures
    - Defining field types and constraints
    - Designing indexes and relationships
2. **Data Access Layer Implementation**:
    - Using JDBC to implement data access
    - Implementing connection pool management
    - Implementing transaction management
3. **Data Quality Assurance** (Optional):

- Data integrity checking
- Data consistency maintenance
- Data backup and recovery

4. **Database Optimization**:
   - Query optimization
   - Index optimization
   - Table structure optimization
5. **Data Security**:
   - Data encryption
   - Access control
   - Audit logging

6 Analysis Group Technical Roadmap

6.1 Analysis Technology Stack

- **Core Language**:
  - Java: Main analysis language
- **Data Processing Frameworks**:
  - Apache Spark: Large-scale data processing
  - Apache Flink: Streaming data processing
- **Machine Learning Libraries**:
  - Weka: Machine learning algorithms
  - DL4J (Deep Learning for Java): Deep learning
- **Statistical Analysis Libraries**:
  - Apache Commons Math: Mathematical and statistical calculations
  - JFreeChart: Chart generation
- **Data Visualization**:
  - JFreeChart: Java chart library
  - XChart: Simple chart library
- **API Integration**:
  - Spring Boot: Providing analysis service API
  - RESTful API: Interacting with System Group

6.2 Analysis Functionality Implementation

1. **Data Analysis Algorithms**:
    - Implementing data preprocessing algorithms
    - Implementing feature extraction algorithms
    - Implementing classification and regression algorithms
    - Implementing clustering algorithms
2. **Streaming Data Processing**:
    - Real-time data reception and processing
    - Streaming analysis algorithm implementation
    - Real-time result generation and transmission
3. **Report Generation**:
    - Designing report templates
    - Implementing data filling logic
    - Generating PDF/Excel reports
4. **Data Visualization**:
    - Designing visualization components
    - Implementing data-to-chart mapping
    - Generating interactive charts
5. **Decision Support Models**:
    - Implementing prediction models
    - Implementing recommendation systems
    - Implementing anomaly detection
6. **Analysis Result Storage**:
    - Designing analysis result storage structure
    - Implementing result storage and retrieval
    - Implementing result version management

7 System Integration and Interaction

7.1 Inter-component Interaction

- **UI Group and System Group**:
    - Communication through RESTful API
    - Data exchange using JSON format

  - Implementing WebSocket real-time communication

- **System Group and Data Group**:
  - Database access through JDBC
  - Connection pool management for database connections
  - Transaction management implementation

- **System Group and Analysis Group**:
  - Calling analysis services through RESTful API
  - Asynchronous communication using message queues
  - Implementing streaming data processing

- **Data Group and Analysis Group**:
  - Accessing shared data through JDBC
  - Implementing data export and import

7.2 Data Flow

1. **Data Collection Flow**:
   - Users connect to Bluetooth devices through web pages
   - Web pages collect device data
   - Data is sent to System Group API via network
2. **Data Processing Flow**:
   - System Group receives data and performs initial processing
   - Data is stored in Data Group's MySQL database
   - Data is forwarded to Analysis Group for streaming analysis
3. **Analysis Result Flow**:
   - Analysis Group generates analysis results
   - Results are stored in Data Group's MySQL database
   - Results are returned to UI Group through System Group API
   - UI Group renders analysis results

8 Deployment Architecture

8.1 Development Environment

- **Development Tools**:
  - IntelliJ IDEA: Java development
  - Visual Studio Code: Frontend development

  - Git: Version control

- **Development Servers**:
  - Local development environment
  - Development server

- **Testing Environment**:
  - Unit testing: JUnit
  - Integration testing: Spring Test
  - End-to-end testing: Selenium

8.2 Production Environment

- **Server Architecture**:
  - Application server: Deploying System Group and Analysis Group services
  - Database server: Deploying MySQL database
  - Web server: Deploying frontend application

- **Containerized Deployment**:
  - Docker containers: Encapsulating applications
  - Kubernetes: Container orchestration

- **High Availability Design**:
  - Load balancing: Nginx
  - Database master-slave replication
  - Service redundant deployment

- **Monitoring and Operations**:
  - System monitoring: Prometheus + Grafana
  - Log management: ELK Stack
  - Alert system: AlertManager

9 Risks and Challenges

9.1 Technical Risks

- **Web Bluetooth API Compatibility**: Different browsers have varying levels of support for Web Bluetooth API
- **Real-time Data Processing Performance**: Streaming data processing may face performance challenges
- **Data Security**: Security risks in sensitive data transmission and storage
- **System Integration Complexity**: Different technology stacks between groups may present integration challenges

9.2 Mitigation Strategies

- **Technology Selection Evaluation**: Thoroughly evaluating the pros and cons of each technical solution
- **Prototype Validation**: Conducting prototype validation for key functionalities
- **Incremental Development**: Adopting agile methods to gradually improve functionality
- **Continuous Integration and Testing**: Establishing comprehensive CI/CD workflows and testing systems
- **Technical Documentation**: Detailed recording of technical decisions and implementation details

10 Conclusion

This document details the technical roadmap for each component of the system, including the UI Group, System Group, Data Group, and Analysis Group. By clearly defining the technical choices and implementation strategies for each group, it provides clear technical guidance for the development team, ensuring technical compatibility and collaboration efficiency between groups.

The system adopts a layered architecture, with components working together through well-defined interfaces. The frontend uses HTML, CSS, and JavaScript to implement the user interface, the backend uses Java and Spring Boot to provide API services, the data layer uses MySQL and JDBC for data storage and access, and the analysis layer uses Java and data processing frameworks to implement data analysis and visualization.

By following the technical roadmap in this document, the project team can efficiently develop and deploy the system to meet user needs and business objectives.