

INFO5100 20240 Application Engineer & Dev SEC 30 Fall 2023 [OAK-2-TR]

Application Specification for Blood Monitor Project

Team name:

Yin and Yang

Team members:

Shuhan Ji: 002857464

Rushikesh Prajapati: 002809751

Zhongwen Yang: 002816147

Yiran Chen: 002839117

Yazhen Han: 002950305

Yuchen Zhou: 002833031

System Definition

Project Overview

The Blood Monitor application is designed to monitor and analyze key blood pressure metrics, including Diastolic Pressure, Systolic Pressure, and Heart Rate. The system is built using Vue.js for the front end and Spring Boot for the back end, providing a user-friendly interface for real-time tracking and personalized health monitoring.

Key Features

- **Data Visualization:**
 - Real-time display of Diastolic Pressure, Systolic Pressure, and Heart Rate data using Vue.js and EChart library.
 - Chart.js utilizes data representation to generate graphs for Diastolic Pressure, Systolic Pressure, and High Blood Pressure. The data is sourced from a database, and the resulting graph is responsive and user-friendly, facilitating easy analysis based on date and time information.
- **Date Range Filtering:**

- Users can filter data by selecting a specific one-week date range using Vue components.

- **Metric Switching:**

- Users can switch between different blood pressure metrics, such as Diastolic Pressure, Systolic Pressure, and Heart Beats, using navigation buttons within Vue components.

- **Real-time Updates:**

- Data visualizations update in real-time based on the user-selected date range.

Requirements

Non-functional Requirements

- **Usability:**

- The user interface shall be intuitive and user-friendly.
- Response time for data retrieval and visualization shall be within acceptable limits.

- **Reliability:**

- The application shall handle errors gracefully and provide clear feedback to users.
- The system shall be available and reliable for use at least 99% of the time.

- **Security:**

- User data, especially health-related information, shall be stored securely.
- The application shall adhere to data protection regulations and best practices.

Design Document

Architecture

The Blood Monitor application follows a client-server architecture.

- **Frontend:**

- Developed using Vue.js and Element UI for the user interface.
- Utilizes the Charts.js library for dynamic data visualization.

- **Backend:**

- Built using Spring Boot for the server-side configuration.
- Implemented CRUD operations on data using Spring Data JDBC and Spring JPA.

Database Schema

The application's database will store user data, including date, Diastolic Pressure, Systolic Pressure, and Heart Rate.

- **Table: blood_pressure_data**

- Columns:

id,date,time,systolic_pressure,diastolic_pressure,heart_beats,avg_SP,avgDP,avg_HB

- Triggers: calculate_average

APIs

- **GET /bloodpressure/{startTime}/{endTime}**

- Retrieves blood pressure data within the specified date range.
- Parameters: startTime, endTime.
- Response: JSON array of bloodpressure data and status code.

- **GET /bloodpressure**

- Retrieves all blood pressure data .
- Response: JSON array of bloodpressure data and status code.

User Interface Flow

- User selects a date range using the Date Picker component.
- Application fetches relevant blood pressure data from the server based on the selected date range.
- Data is displayed using ECharts charts.
- Graphs are generated using data time and blood pressure measurements to represent the information visually.
- Users can switch between Diastolic Pressure, Systolic Pressure, and Heart Beats using navigation buttons.

Task Allocation

- **Frontend:**
 - **Yazhen Han:**
 - A. Packaged the request files including encapsulating Axios requests(configured request interceptor and response interceptor)
 - B. Configured a state management system using Vuex
 - C. Coordinated with the backend by calling API
 - D. Created a table with various health metrics for data visualization(systolic pressure,diastolic pressure, heart beats, avg systolic pressure etc..)
 - E. Added a date filter to display data for specific dates
 - F. Configured a proxy to address cross-origin issues in vue.config.js
 - **Shuhan Ji:**
 - A. **Health Data Visualization Module:**
 - Implement a date range filter component allowing users to select a one-week timeframe.
 - Integrate the ECharts library for creating interactive line charts to visualize diastolic pressure data.
 - Develop navigation buttons facilitating the switch between different blood pressure metrics (Systolic Pressure, Diastolic Pressure, Heart Beats).
 - Implement functionality to filter data for a specific one-week period based on user selections.
 - Ensure ECharts charts provide real-time representation of diastolic pressure over the selected timeframe.
 - B. **Error Handling Component:**
 - Develop a component for displaying user-friendly messages in case the selected date range is not one week.
 - C. **User Interface (UI) Design:**
 - Design the overall layout and styling of the user interface to ensure a cohesive and visually appealing experience.
 - D. **Integration Testing:**

Conduct integration testing to ensure that all components work seamlessly together and meet the specified requirements.

E. Documentation:

Create documentation for the frontend components, including usage instructions and any relevant information for future development or maintenance.

- Rushikesh Prajapati:

Presentation for Introduction of this project and wrote chart.js

- **Backend:**
- Yuchen Zhou, Yiran Chen
- **Database:**
- Zhongwen Yang

Conclusion

This application specification outlines the system definition, requirements, and design document for the Blood Pressure Monitor project. It provides a foundation for development and collaboration among team members involved in the project.