Health Dashboard Requirements Specification

1. Overview

1.1 Purpose

This document defines the requirements for a **Health Dashboard** feature that provides real-time monitoring of a Raspberry Pi-based Flight Tracking system status. The purpose is to ensure system stability and performance while enabling early detection of operational anomalies.

1.2 Scope

- Real-time display of system status information collected from Raspberry Pi on GUI
- Information updates at 1-second intervals
- Warning alarms when system status anomalies occur
- Presentation in Menu List format

2. Functional Requirements

ID	Function Description	Detailed Content
FR-01	CPU Usage Display	Display current CPU usage (%) and maximum usage
FR-02	Memory Usage Display	Display current memory in use (MB) and maximum memory (MB)
FR-03	CPU Temperature Display	Display current CPU temperature (°C) and maximum temperature
FR-04	Disk Usage Display	Display current disk usage (%) and maximum usage
FR-05	Uptime Display	Display system uptime (DAY HH:MM:SS)
FR-06	Data Latency	Time taken for the Display program to receive data transmitted from Raspberry Pi
FR-07	Update Interval Setting	All information is automatically updated every 1 second
FR-08	Menu List UI	Each item is organized in menu list format within the GUI
FR-09	Warning Alarm Function	Display warning and timestamp when abnormal threshold values are met next to GUI items
FR-09	Warning Alarm Function	Warning alarm criteria and display methods when system status abnormalities occur: • CPU Usage: When 80% or higher is maintained for 5 seconds / High CPU Load • Memory Usage: When 80% or higher / Memory Shortage • CPU Temperature: When 70°C or higher / High Temperature • Disk Usage: When 90% or higher / Disk Space Shortage • Data Latency: When 500ms or higher / Data Reception Delay (Timestamp not displayed separately)
FR-10	General Error Display	Items and Display: 1. When CRC fails 3 consecutive times / Network Error 2. When initial Connect fails / Connection Failed 3. When network connection is lost / Network Disconnected

3. Non-Functional Requirements

ID	Item	Description
NFR-01	Performance	Data collection and display delay shall be within 500ms
NFR-02	Reliability	When data transmission/reception fails, the system shall maintain previous normal values and display warning messages to users. CRC errors occurring 3 consecutive times shall be considered as 'Network Error'
NFR-03	Scalability	Future expansion possible for GPU usage, network traffic, etc.
NFR-04	Usability	UI shall be designed intuitively and user-friendly, and key information shall be recognizable within 3 seconds
NFR-05	Maintainability	Modular structure for easy function-specific maintenance
NFR-06	Testability	All functions shall be verifiable through automated or manual testing, and test coverage shall be 80% or higher

4. System Interface

- **Data Collection Method**: Using Raspberry Pi internal scripts or system APIs (`/proc`, `vcgencmd`, `uptime`, `df`, `free`, `sensors`, etc.)
- **Data Transmission Method**: Local IPC or TCP/HTTP communication (optional)
- **GUI Integration Method**: Integration with Embarcadero C++Builder-based GUI

5. UI Requirements

- **Display Format**: Menu list format
- **Display Items**: Current value / Maximum value for each item
- **Update Method**: Automatic updates every 1 second
- **Example**:

```
[System Health]                                          Latency : 123ms
- CPU Usage: 23% / 100%
- Memory Usage: 512MB / 2048MB
- Disk Usage: 45% / 100%
- CPU Temp: 52°C / 85°C
- Uptime: x days HH:MM:SS
```

6. Test Requirements

- **Normal Operation Test**: Verify that each item is displayed correctly
- **Update Test**: Verify that values are updated every 1 second

- **System Status Anomaly Alarm Test**: Verify occurrence of system status anomaly alarms
- **Error Handling Test**: Verify warning message output when sensors malfunction