|  |  |
| --- | --- |
| A blue sign with silver border  AI-generated content may be incorrect. | Functional SpecificationSPC Station Health Charts |

**Version:** 1.4

**Date:** October 7, 2025

**Status:** Implemented with Debug Tools

**Enhancement:** CSV Troubleshooting & Enhanced Error Reporting

## 1. Introduction

### 1.1 Purpose

This document specifies the functional behavior of the SPC Station Health Charts application from an end-user perspective. It describes what the system does, not how it does it (see Technical Spec for implementation details).

### 1.2 Scope

Desktop application for visualizing Statistical Process Control charts for airline maintenance metrics. Covers data loading, visualization, phase detection, and chart export.

### 1.3 Audience

* Product managers
* Quality assurance testers
* End users (maintenance managers, analysts)
* Training documentation authors

## 2. User Roles & Personas

### 2.1 Primary User: Maintenance Station Manager

**Name:** Sarah Chen  
**Role:** Maintenance Manager, Dallas Love Field (DAL)  
**Technical Skills:** Moderate (Excel power user, not a programmer)  
**Environment:** Corporate Windows laptop, no admin rights, restricted firewall

**Goals:**

* Monitor weekly maintenance KPIs
* Identify when process shifts occur
* Report trends to regional leadership
* Create charts for monthly presentations

**Pain Points:**

* Current Excel charts take 30+ minutes to update
* Manual calculation of control limits prone to errors
* Difficult to identify genuine process shifts vs. noise

### 2.2 Secondary User: Technical Operations Analyst

**Name:** Mike Rodriguez  
**Role:** Senior Tech Ops Analyst, Corporate HQ  
**Technical Skills:** Advanced (SQL, Python, data analysis)  
**Environment:** Workstation with Python installed

**Goals:**

* Analyze custom metrics not in standard reports
* Compare performance across multiple stations
* Validate statistical methodology
* Export charts for executive dashboards

**Pain Points:**

* Need flexibility to upload ad-hoc data
* Require statistically rigorous methods
* Must work offline (travels to remote stations)

## 3. Functional Requirements

### 3.1 Application Launch

#### FR-3.1.1: Windows Launch

**Trigger:** User double-clicks START\_DASHBOARD.bat  
**Preconditions:** Python 3.7+ installed

**Process:**

1. Batch file starts Python HTTP server on port 8000
2. Command Prompt window opens with server status
3. Default browser opens automatically to http://localhost:8000
4. Dashboard welcome screen displays

**Expected Output:**

╔════════════════════════════════════════════════════════════╗

║ Airline Tech Ops SPC Dashboard Server ║

╚════════════════════════════════════════════════════════════╝

✓ Server running on: http://localhost:8000

📋 Instructions:

1. Open your web browser

2. Navigate to: http://localhost:8000

3. Click 'Load Demo' or upload your CSV file

4. Press Ctrl+C to stop the server

⚠️ Keep this window open while using the dashboard!

**Success Criteria:**

* Server starts within 2 seconds
* Browser opens to dashboard
* No error messages displayed

### 3.2 Welcome Screen

#### FR-3.2.1: Initial Display

**When:** User navigates to http://localhost:8000

**Display Elements:**

* ✅ Application title: "SPC Station Health Charts"
* ✅ Subtitle: "Statistical Process Control • Wheeler's XmR Methodology"
* ✅ Two prominent buttons:
  + "📊 Load Test Data" (green, primary action)
  + "📤 Upload CSV (Auto-Detect Format)" (blue, secondary action)
* ✅ Description text: "Simulated test data • January 2023 to present • 4 maintenance metrics"
* ✅ Format support note: "Upload supports: timestamp,station,metric\_value OR station,measure,date,value"

### 3.3 Load Test Data

#### FR-3.3.1: Load Button Click

**Trigger:** User clicks "📊 Load Test Data" button

**Process:**

1. Button becomes disabled (prevent double-click)
2. Loading indicator displays: "Loading Test Data..."
3. Browser sends POST request to /api/load-actual
4. Server reads CSV files from input/ folder
5. Server processes data and returns JSON
6. Dashboard renders charts
7. Loading indicator disappears

**Expected Timeline:**

* Loading indicator appears: < 100ms
* Data processing: < 2 seconds
* Chart rendering: < 1 second
* **Total:** < 3 seconds

#### FR-3.3.2: Test Data Content

**Data Specifications:**

* **Time Range:** January 2, 2023 to current week (140+ weeks)
* **Stations:** 3 (Austin/AUS, Dallas/DAL, Houston/HOU)
* **Measures:** 4
  1. Maintenance Cancels
  2. Maintenance Delays
  3. Scheduled Maintenance Findings
  4. Unscheduled Maintenance
* **Chart Count:** 24 total (3 stations × 4 measures × 2 chart types)

### 3.4 Station Selection

#### FR-3.4.1: Station Dropdown

**Location:** Top-right of dashboard  
**Default Value:** First station alphabetically (AUS)

**Options:**

* "All Stations" (shows all station charts)
* "AUS" (Austin)
* "DAL" (Dallas Love Field)
* "HOU" (Houston Hobby)

**Behavior:**

* Dropdown updates after data loads
* Selecting station filters charts dynamically
* Page does not reload
* Charts re-render smoothly

### 3.5 Chart Visualization

#### FR-3.5.1: Chart Components

Each chart displays:

**Title Bar:**

* Station code (e.g., "DAL")
* Measure name (e.g., "Maintenance Cancels")
* Chart type: "(X Chart - Individuals)" or "(mR Chart - Moving Range)"

**Chart Area:**

* X-axis: Date (weekly intervals)
* Y-axis: Metric value (auto-scaled)
* Data points: Blue dots connected by lines
* Out-of-control points: Red dots
* Control limits: Horizontal lines (UCL, CL, LCL) in different colors
* Phase boundaries: Vertical dotted lines (if multiple phases)

**Legend:**

* UCL (Upper Control Limit)
* CL (Center Line / Mean)
* LCL (Lower Control Limit)
* Data Points

**Action Button:**

* "💾 Save PNG" button below chart

#### FR-3.5.2: X Chart (Individuals)

**Purpose:** Monitors process location (average level)  
**Y-axis Label:** Metric name (e.g., "Maintenance Cancels")  
**Data Points:** Actual weekly values

**Control Limits:**

* Calculated using 2.66 sigma
* Based on moving range
* Recalculate per phase

**Visual Indicators:**

* Points within limits: Blue
* Points outside limits: Red (signals special cause)
* Line connects sequential points

#### FR-3.5.3: mR Chart (Moving Range)

**Purpose:** Monitors process variation (consistency)  
**Y-axis Label:** "Moving Range"  
**Data Points:** Absolute difference between consecutive values: |X[i] - X[i-1]|

**Control Limits:**

* UCL calculated from average moving range
* CL = average moving range
* LCL typically 0 (ranges cannot be negative)

**Note:** mR chart has one fewer data point than X chart (first point has no previous value)

### 3.6 CSV Upload

#### FR-3.6.1: File Selection

**Trigger:** User clicks "📤 Upload CSV" button

**Process:**

1. Hidden file input dialog opens
2. User selects .csv file
3. File name displays briefly
4. Loading indicator: "Processing [filename]..."

**Accepted File Types:** .csv only  
**Max File Size:** No explicit limit (tested up to 10MB)

#### FR-3.6.2: Format Auto-Detection

**Supported Formats:**

**Format A: timestamp,station,metric\_value**

timestamp,station,metric\_value

2023-01-02,Austin,2.5

2023-01-02,Dallas,3.1

2023-01-09,Austin,2.8

* **Measure name:** Inferred from filename (e.g., maintenance\_cancels.csv → "Maintenance Cancels")
* **Station names:** Auto-mapped (e.g., "Dallas" → "DAL", "Houston" → "HOU")

**Format B: station,measure,date,value**

station,measure,date,value

AUS,Maintenance Cancels,2023-01-02,2.5

DAL,Maintenance Cancels,2023-01-02,3.1

AUS,Maintenance Cancels,2023-01-09,2.8

* **Measure name:** Taken from measure column
* **Station codes:** Used as-is

### 3.7 Chart Export

#### FR-3.7.1: Save PNG Button

**Location:** Below each chart  
**Label:** "💾 Save PNG"  
**Trigger:** User clicks button

**Process:**

1. Button click captured
2. Canvas element converted to PNG data URL
3. Temporary download link created
4. Browser "Save As" dialog appears
5. User chooses save location
6. File saves

**Default Filename:** [Station]\_[Measure]\_[ChartType].png  
**Example:** DAL\_Maintenance\_Cancels\_X\_Chart.png

#### FR-3.7.2: PNG Image Specifications

* **Resolution:** Matches canvas size (~1200×600 pixels)
* **Format:** PNG (lossless)
* **Background:** White
* **Quality:** Suitable for printing and presentations
* **File Size:** Typically 50-150 KB per chart

**Image Contents:**

* Complete chart (data, axes, labels, legend)
* Title clearly visible
* Control limits labeled
* Legend included

## 3.8 CSV Troubleshooting Tools

### 3.8.1 CSV Format Checker

**Purpose:** Diagnose CSV format issues before upload  
**Location:** http://localhost:8000/csv\_format\_checker.html

**Features:**

* Auto-detects CSV format (A or B)
* Validates column names and data types
* Identifies decimal format issues (commas vs. periods)
* Provides specific fix suggestions
* Shows sample data analysis

**Usage:**

1. Open CSV Format Checker in browser
2. Paste CSV content into text area
3. Click "Check Format" to analyze
4. Review results and apply suggested fixes

### 3.8.2 Enhanced Error Reporting

**Server Console Debugging:**

* Shows detailed CSV parsing information
* Displays each row being processed
* Identifies specific parsing errors
* Reports total valid data points found

**Client-Side Error Messages:**

* Clear format requirements
* Specific error descriptions
* Actionable fix suggestions

## 4. Acceptance Criteria

### 4.1 Launch Criteria

* ✅ User can start application with single double-click
* ✅ No installation required beyond Python
* ✅ Dashboard loads without errors
* ✅ Test data displays correctly

### 4.2 Visualization Criteria

* ✅ Both X and mR charts display for each measure
* ✅ Control limits calculated using 2.66 sigma
* ✅ Phase boundaries marked when shifts detected
* ✅ Out-of-control points highlighted in red
* ✅ Charts scale appropriately to data range

### 4.3 Export Criteria

* ✅ PNG download triggers on button click
* ✅ Image includes complete chart with labels
* ✅ Filename descriptive and valid
* ✅ Image quality suitable for presentations

## 5. Out of Scope

### Explicitly Excluded from V1.0:

* ❌ Real-time data updates
* ❌ Data persistence / database
* ❌ Multi-user collaboration
* ❌ User authentication
* ❌ Mobile responsive design
* ❌ Internationalization (English only)
* ❌ Accessibility features (screen readers, etc.)
* ❌ Advanced SPC rules (beyond Rule #1 and #4)

## Approval

|  |  |  |  |
| --- | --- | --- | --- |
| **Role** | **Name** | **Date** | **Signature** |
| Product Owner | Tech Ops Manager | 2025-10-06 | ✅ Approved |
| QA Lead | Quality Assurance | 2025-10-06 | ✅ Tested |
| End User | Station Manager | 2025-10-06 | ✅ Accepted |

**Document Version History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Changes** |
| 1.0 | 2025-10-06 | Development Team | Initial functional specification |
| 1.3 | 2025-10-07 | Development Team | Graphics update: Southwest Tech Badge logo, logo test gallery, improved branding |
| 1.4 | 2025-10-07 | Development Team | CSV troubleshooting tools, enhanced error reporting, debug logging |