# Quality Ticket System

Project Scope Development Plan

1/29/2025

| | |
|---|---|
| **Prepared By:** Trevor Keller | **Estimated Start Date:** 2/1/2025 |
| **Project Leader:** Karl Jentzen | **Estimated Completion Date:** 12/31/2025 |

## 1. Project Overview

**1.1 Description:** This project aims to redesign and optimize our current quality assurance program and workflow to improve efficiency, usability, and scalability. This overhaul will involve analyzing existing quality processes, identifying inefficiencies, and implementing modern tools for a better user experience. The final product will provide a structured, efficient, and scalable quality ticketing system that enhances the overall system.

**1.2 Justification:** Our current quality system has become inefficient, prone to errors, outdated, and limited scalability, leading to increased manual workload and reduced operational efficiency. By modernizing our workflow and design, we aim to increase efficiency, improve compliance, and support scalability. This will lead to higher product quality, faster issue responses, and an overall improvement in quality workflow.

## 2. Objectives

**2.1 Database Migration**
- Transition from the current PHP-based database to Microsoft SQL Server.

**2.2 User Interface Modernization**
- Replacing the existing PHP-system with a modern React + TypeScript frontend.
- New system integrates seamlessly into the current hub system.
- Ensure mobile compatibility to support users on various devices.

**2.3 Enhanced System Efficiency**
- Optimized system performance for ticket creation, updates, viewing, and closing.
- Introduce new features that enhance ticket tracking and increase resolution speed.
- Improve reports,logs, and summary capabilities.

## 3. Deliverables

**3.1 Migrated Microsoft SQL Database**
- Optimized Microsoft SQL Server database integration.

**3.2 New React + TypeScript User Interface**
- A new front-end application built using React + TypeScript, replacing old PHP UI.
- Application integrates seamlessly into the current hub system.
- Fully responsive UI, ensuring compatibility with all devices.

**3.3 Optimized Ticket Management System**
- Performance optimized ticket management system support:
    - Creation, updates, viewing, and closing of tickets.
- Implementation of new tracking/notification features to improve resolution time.
- Enhanced reporting and summary capabilities, allowing users to generate reports.

**3.4 Functionality Testing**
- Verify proper workflow of all ticket functionality (creation, viewing, updates, closing).
- Ensure the new UI is compatible with all desktop, and mobile devices.

## 4. Scope

### 4.1 In Scope

- **Database Migration Scope**
  - Migration from PHP-based database to Microsoft SQL Server.
- **User Interface Enhancements Scope**
  - Development of new front-end UI using React + TypeScript.
  - Integration of new system into existing hub system.
  - Full mobile responsiveness across devices.
- **System Performance & Feature Enhancements Scope**
  - Performance optimization for usage of tickets.
  - Implementation of new ticket tracking/notification features.
  - Enhanced reporting and summary generation capabilities.
- **Comprehensive Testing**
  - Functional Testing including ticket lifecycle and database transactions.
  - UI testing to confirm responsive UI across devices.

### 4.2 Out of Scope:

- **Major Workflow Redesign:** The core workflow of the current Quality Ticket System will remain the same, with only efficiency improvements.
- **Legacy Support System:** The project will focus on the new system.

## 5. Requirements:

### 5.1 Database & API Requirements

- The system must use Microsoft SQL Server as the primary database.
- A Node.js API must be implemented to handle communication between the frontend and the SQL database.
- API responses must be in JSON format.

### 5.2 User Interface Requirements

- The frontend must be built using React + TypeScript.
- The UI must be styled using Tailwind CSS and ShadCN components. Along with using the correct color scheme: rgb(59,130,246), rgb(255,255,255) rgb(31,41,55). The component should not have a width greater than 1300px.
- The UI must integrate seamlessly into the existing hub system.
- The application must be fully responsive and support desktop and mobile devices.

### 5.3 Core Features

- Users must be able to manage tickets (create, update, view, close).
- The system must include a notification system (email), to alert users of ticket status.
- Ability to generate reports and visual overview of tickets.

### 5.4 Testing & Deployment

- Functional testing must cover the entire ticket lifecycle (creation, viewing, updates, closing).
- UI testing must ensure a responsive UI across desktop and mobile devices.