

# Software Requirements Specification

## PriceEngine. Multi Organization Pricing Platform

**Version:** 1.0

**Status:** Draft

**Date:** February 2, 2026

**Prepared for:** PriceEngine Stakeholders and Development Team

**Prepared by:** Roy Chababi

---

## Table of Contents

1. Introduction
  - 1.1 Purpose
  - 1.2 Scope
  - 1.3 Definitions
  - 1.4 Overview of Document
2. Overall Description
  - 2.1 Product Perspective
  - 2.2 System Environment
  - 2.3 User Classes and Roles
  - 2.4 Assumptions and Constraints
3. System Features and Functional Requirements
  - 3.1 Multi Organization Architecture
  - 3.2 Authentication and Access Control
  - 3.3 Test Management
  - 3.4 Pricing Calculator
  - 3.5 External API Pricing Requests
  - 3.6 Logs and Audit Trail
  - 3.7 Analytics
4. External Interface Requirements
  - 4.1 User Interface
  - 4.2 API Interface
5. Non Functional Requirements
  - 5.1 Security and Data Isolation
  - 5.2 Logging and Traceability
  - 5.3 Performance
6. Glossary

---

# 1. Introduction

## 1.1 Purpose

This document specifies the software requirements for PriceEngine. It describes the system functionality, user roles, access control, and feature requirements for test management, pricing calculation, logs, and analytics.

## 1.2 Scope

PriceEngine is a shared platform that supports multiple independent laboratory organizations. Each organization uses the platform in isolation with its own users, tests, pricing data, logs, and analytics. The platform provides a pricing calculator for operational use and an analytics module for profitability modeling.

## 1.3 Definitions

- **Organization:** A laboratory entity using PriceEngine in an isolated environment.
- **Test Catalog:** The collection of tests maintained by a lab manager.
- **Panel:** A custom grouping of laboratory tests selected by a user for pricing and analysis.
- **Overhead Cost per Panel:** The allocated overhead derived from overhead cost and panel volume assumptions.
- **Anchor Test:** A test identified by the pricing rules as the primary driver of pricing logic.
- **Add On Test:** A supplementary test included in the panel.

## 1.4 Overview of Document

Section 2 provides the overall description of PriceEngine and its operating environment. Section 3 provides feature level functional requirements. Sections 4 and 5 specify interfaces and non functional requirements.

---

# 2. Overall Description

## 2.1 Product Perspective

PriceEngine is a multi organization platform that supports independent laboratories using a shared system while maintaining data and configuration isolation between organizations.

## 2.2 System Environment

- Web application with a centralized login page for all users
- Organization scoped data storage for tests, pricing, logs, and analytics
- External API access for pricing calculation requests

## 2.3 User Classes and Roles

### **Super User. PriceEngine Admin**

- Platform level administrator
- Manages organizations and global system configuration
- No access to lab operational data unless explicitly granted

### **Lab Manager**

- Manages laboratory specific configuration
- Responsible for: Test catalog, pricing, analytics and logs.

### **Lab Employee**

- Operational user
- Can access pricing calculator
- Does not manage pricing configuration or analytics

## 2.4 Assumptions and Constraints

- The platform supports staff user for lab employees. Individual employee accounts are not required.
- Access is role based rather than person specific.
- All data and pricing behavior are isolated per organization.

---

## **3. System Features and Functional Requirements**

### **3.1 Multi Organization Architecture. Functional Overview**

#### **Description**

Each laboratory organization has its own independent instance of all PriceEngine functionality. This includes organization specific user roles, test catalogs, pricing rules, analytics inputs and outputs, logs, and calculator results. Data, configurations, and pricing behavior are fully isolated per organization, ensuring that actions performed within one organization do not affect or expose data from any other organization.

Each organization operates in isolation with its own:

- Users
- Tests
- Pricing data
- Logs
- Analytics

#### **Requirements**

- The system shall support multiple laboratory organizations on the same platform.
- The system shall isolate users, tests, pricing data, logs, and analytics by organization.
- The system shall ensure that actions performed within one organization do not affect or expose data from any other organization.

---

### **3.2 Authentication and Access Control**

#### **Login Page. All Users**

- Centralized login page for all users

- Authentication handled by the platform
- Users are associated with an organization and a role
- Lab employees are treated as staff users
- Individual employee accounts are not required
- Access is role based rather than person specific

## Requirements

- The system shall provide a centralized login page for all users.
  - The system shall associate authenticated users with an organization and role.
  - The system shall enforce role based access for platform features.
- 

## 3.3 Test Management. Lab Manager

### Overview

The Test Management page allows a lab manager to maintain the laboratory's test catalog and pricing inputs.

### Bulk Upload

- Upload test catalog via Excel file using a predefined format
- Upload populates or updates the lab's test database
- Validation performed on upload:
  - Required fields present
  - No duplicate test IDs
  - Correct data types

### Test List View

Each test entry includes:

- Test ID

- Test Name
- Reagent Cost
- List Price

## Test Operations

- Add a new test manually
- Edit existing test information
- Delete a test
- Search and filter tests by:
  - Test ID
  - Test Name

## Requirements

- The system shall allow a lab manager to upload a test catalog via Excel file.
  - The system shall validate required fields, duplicate test IDs, and data types on upload.
  - The system shall allow adding, editing, deleting, searching, and filtering tests.
  - The system shall store test inputs including test ID, test name, reagent cost, and list price.
- 

## 3.4 Pricing Calculator. All Users

### Calculator Functionality

- Users can select any combination of tests to build a custom panel
- Submit the custom panel for pricing calculation
- Display calculated pricing results in real time

## Requirements

- The system shall allow users to build a panel by selecting tests from the laboratory's test catalog.
  - The system shall calculate and display the panel price immediately upon submission.
  - The system shall support lab employees accessing the calculator.
- 

## 3.5 External API Pricing Requests

### Description

The platform also supports external API requests for pricing calculations. API consumers can submit requests that include the laboratory organization identifier (organization number) along with the selected test data. The system processes the request within the context of the specified organization and returns the calculated panel price based on that organization's test catalog, pricing rules, and overhead configuration.

### Requirements

- The system shall accept external API requests for pricing calculations.
  - The system shall require an organization identifier (organization number or name) for each API request.
  - The system shall process API requests using the specified organization's configuration and data.
  - The system shall return the calculated panel price in the API response.
- 

## 3.6 Logs and Audit Trail. Lab Manager

### Overview

The Logs page provides visibility into all pricing activity performed by the lab.

### Logged Events

- All custom panel pricing requests
- Requests made through:
  - Calculator page

- External API

## Log Capabilities

- View pricing request history

Filter and search logs by:

- Date of request
- Panel composition. The set of laboratory tests selected to form the panel

Sorting and relevance behavior:

- The logs view is ordered by panel composition relevance first, then by time.
- When a user searches for tests S and X, results are ranked as follows:
  - Panels that match exactly S and X appear first.
  - Panels that contain S and X plus additional tests appear next.
  - Within the same relevance level, results are ordered by date of request, most recent first.

## Log Entry Details

- Timestamp
- Panel composition: the set of laboratory tests selected to form the panel
- Final calculated price

## Requirements

- The system shall log all custom panel pricing requests.
- The system shall log requests made through the calculator page and external API.
- The system shall allow lab managers to view, filter, and search pricing request history.
- The system shall include timestamp, panel composition, and final calculated price in each log entry. (name of client must never be logged).

---

## 3.7 Analytics. Lab Manager

### Panel. Definition and Purpose

A panel represents a custom grouping of laboratory tests selected by a user for pricing and analysis.

A panel is composed by selecting one or more tests from the laboratory's test catalog. The system calculates and displays the panel price immediately.

Each test within the panel contributes reagent cost, list price, and overhead allocation to the final panel price.

### Analytics Inputs

The analytics module requires inputs that allow the system to calculate the overhead cost per panel, which is necessary to determine whether a given panel is profitable.

User provides the following parameters:

- Current daily or monthly fixed overhead cost
  - Includes labor, equipment, facilities, and other operational expenses
- Current number of panels processed per day
  - Used to allocate current overhead across panels
- Expected future overhead cost
  - Allows modeling of cost changes due to growth, hiring, or infrastructure expansion
- Expected future number of panels processed per day
  - Used to simulate volume scaling and its impact on per panel overhead

These inputs are used to derive the overhead cost per panel by distributing fixed overhead across the total number of panels processed. This step is critical, as overhead cost per panel directly impacts the total cost of a panel and therefore its profitability.

### Analytics Outputs

The analytics page presents a detailed financial breakdown for the selected panel under both current and projected operating conditions.

- **Profitability Forecast**
  - Displays current and future profitability of the panel based on user provided

assumptions

- Enables comparison across different processing volumes and overhead cost structures

- **Pricing Algorithm Breakdown**

- Identification of anchor test(s) within the panel
- Identification of add on tests
- Detailed per test breakdown including:
  - Reagent cost
  - Allocated overhead contribution
  - List price or applied pricing rule

- **Total Reagent Cost**

- Aggregated reagent cost across all tests in the selected panel
- Represents the direct variable cost of performing the panel

- **Total Overhead Cost**

- Allocated labor and operational overhead applied to the panel
- Calculated based on the overhead cost per panel derived from analytics inputs

- **Gross Profit Margin**

- Calculated margin for the panel under current and projected scenarios
- Allows lab managers to quickly assess whether the panel is profitable

The panel pricing algorithm calculates the final panel price based on test list prices, reagent costs, marginal overhead allocation, and predefined pricing rules. The algorithm distinguishes between a primary **anchor test** and secondary **add-on tests** to avoid linear price stacking while preserving minimum margin requirements.

## **Step-by-Step Pricing Logic**

1. **Input**

- The algorithm receives a list of test identifiers representing the selected panel composition.

2. **Test Retrieval and Ordering**

- All tests associated with the provided test IDs are retrieved from the database.
- Tests are sorted in descending order by list price.
- The test with the highest list price is designated as the anchor test.

### **3. Anchor Test Pricing**

- The anchor test contributes its full list price to the panel price.
- Its reagent cost plus a marginal overhead value is included in the internal cost calculation.

### **4. Add-On Test Pricing**

- Each remaining test is treated as an add-on test.
- For each add-on test, the system computes:
  - A discounted price equal to the test's list price multiplied by a marginal discount factor.
  - A floor price equal to three times the sum of reagent cost and marginal overhead.
- The final price for the add-on test is the maximum of the discounted price and the floor price.
- This ensures that each test remains profitable while preventing excessive price stacking.

### **5. Additional Fixed Charges**

- A fixed per-request donation amount is added to the panel price.
- A revenue share amount per request is also added.

### **6. Final Output**

- The algorithm returns the total calculated panel price.

### **7. Implementation reference**

For code-level details on pricing breakdowns and panel price calculation logic, refer to [pricecare/backend/calculator](#) in the MVP implemented in Go.

Github repo: <https://github.com/andrewchababi/pricecare.git>

## **Requirements**

- The system shall allow lab managers to create or select a panel for analysis.
  - The system shall accept overhead and volume inputs used to derive overhead cost per panel.
  - The system shall display profitability forecast, pricing algorithm breakdown, total reagent cost, total overhead cost, and gross profit margin.
- 

# **4. External Interface Requirements**

## **4.1 User Interface**

- Centralized login page
- Test Management page for lab managers
- Logs page for lab managers
- Analytics page for lab managers
- Pricing Calculator page for all users

## **4.2 API Interface**

- External API endpoint that accepts organization identifier and selected test data
  - Returns calculated panel price
- 

# **5. Non Functional Requirements**

## **5.1 Security and Data Isolation**

- Organization data, configurations, and pricing behavior are fully isolated per organization.

- Role based access is enforced for all features.

## 5.2 Logging and Traceability

- Pricing requests are logged with timestamp, panel composition, and final calculated price.
- Logs are searchable and filterable by lab managers.

## 5.3 Performance

- The system displays calculated pricing results in real time for calculator submissions.
- 

# 6. Glossary

- **Organization:** Independent laboratory entity using PriceEngine
- **Panel:** Custom grouping of tests selected for pricing and analysis
- **Anchor test:** Test used as the pricing driver in algorithm breakdown
- **Add on test:** Supplementary test included in the panel
- **Overhead cost per panel:** Fixed overhead allocated across panels based on volume inputs