

# PerSpection App

**EDITED** Design: Service Layers

## Owner

Jarrett Zanders

## Overview

The PerSpection app will allow users to build test inspections before official inspection. The application frontend and backend will be hosted through [Render.com](https://render.com) and will be built using a Postgres Object-Relational Database. RESTful APIs Requests and responses will be sent as JSON documents.

## Hosting Platform

Render.com

## Project URL

<https://perspectiveapp.onrender.com>

## GitHub URL

<https://github.com/SWDV-Capstone/PerSpectiveApp.git>

## Database Type

Object-Relational Database

## Database Connection

HTTP protocol to all HTTP requests and responses.

## Database Base URL

<postgres://admin:KqAiCAkiyXEs0ivOiaGbgzAWNBytlVNM@dpg-co07ge8cmk4c73b650tg-a.ohio-postgres.render.com/perspectiveapp>

## Service Layers

1. Management Layer for CRUD operation endpoints.
2. Data Persistence Layer for data storage and retrieval.
3. Reporting Layer for reporting analytics.
4. Validation layer for input controls
5. Authentication layer for application access/navigation.
6. User policy layer for UI roles and rights.
7. Automation layer for event triggers.

## Users

**Create User:** Select “Create Account” for login and click “Save”.

```
async function createUser( data ) {  
  const res = await axios.post( baseUrl, some_column: 'someValue' )  
}
```

**Edit User:** Select user’s “Edit” icon and click “Save”.

```
async function editUser( data ) {  
  const res = await axios.patch( baseUrl + '/' + data.id, { some_column: 'someValue' } )  
}
```

## Inspections

**Create Inspection:** Select “Add New Inspection” and click “Add” icon.

```
async function createInspection( data ) {  
  const res = await axios.post( baseUrl, some_column: 'someValue' )  
}
```

**Edit Inspection:** Select Inspection’s “Edit” icon and click “Save”.

```
async function editInspection( data ) {  
  const res = await axios.patch( baseUrl + '/' + data.id, { some_column: 'someValue' } )  
}
```

**Display Inspections:** Created Inspections will be displayed on the Inspections page.

```
async function fetchInspections() {  
  const res = await axios.get( baseUrl + '?_sort=inspection_name&_order=asc' )  
  inspectionList.value = res.data  
}
```

## Points

**Create Points:** Select “Add Inspection Point” and click “Add” icon.

```
async function createPoint( data ) {  
  const res = await axios.post( baseUrl, some_column: 'someValue' )  
}
```

**Edit Point:** Click Point’s “Edit” icon and click “Save”.

```
async function editPoint( data ) {  
  const res = await axios.patch( baseUrl + '/' + data.id, { some_column: 'someValue' } )  
}
```

**Display Points:** Created point will be displayed as accordion of inspection parent.

```
async function fetchPoints() {  
  const res = await axios.get( baseUrl + '?_sort=point_name&_order=asc' )  
  pointList.value = res.data  
}
```

**Delete Point:** Click Point’s “Delete” icon.

```
async function deletePoint( data ) {  
  const res = await axios.delete( baseUrl + '/' + data.id )  
}
```

### Example: JSON Object (Inspection)

```
{
  "inspection_id": 1001,
  "inspection_name": "Kitchen: Temperature Control",
  "points_count": 2,
  "inspection_group": "Kitchen Monthly",
  "inspection_interval": 30,
  "inspection_due_date": "01-02-2024",
  "inspection_start_date": "01-01-2024",
  "inspection_end_date": null,
  "inspection_status": "incomplete",
  "user": [{
    "user_id": 101,
    "user_isAdmin": true
  }],
  "Points": [
    {
      "point_id": 1001,
      "point_name": "Thermometers working correctly",
      "point_status": "measured",
      "Measurements": [{
        "measurement_id": 1,
        "measurement_value": "Good",
        "measurement_date": "01-01-2024"
      }]
    },
    {
      "point_id": 1004,
      "point_name": " Cold food below 40°F ",
      "point_status": "not-measured",
      "Measurements": [{
        "measurement_id": null,
        "measurement_value": null,
        "measurement_date": null
      }]
    },
  ],
}
```