# Sample inspection application – System documentation

Table of Contents

[Sample inspection application – System documentation 1](#_Toc449619589)

[History of modifications 1](#_Toc449619590)

[Objective 1](#_Toc449619591)

[Development team 1](#_Toc449619592)

[Target users 2](#_Toc449619593)

[Servers 2](#_Toc449619594)

[Basic architecture 2](#_Toc449619595)

[Source code 3](#_Toc449619596)

[Description of the database 4](#_Toc449619597)

[List of tables: 5](#_Toc449619598)

[List of views: 5](#_Toc449619599)

[List of UDF (User defined functions): 5](#_Toc449619600)

[How to add new part categories to the database 5](#_Toc449619601)

[Description of the code 6](#_Toc449619602)

[Customization of the model 6](#_Toc449619603)

[Customization of the controllers 6](#_Toc449619604)

[Customization of views 6](#_Toc449619605)

## History of modifications

|  |  |  |
| --- | --- | --- |
| Date | Name | Comment |
| 04/28/2016 | Luciano Jardim | Created |

## Objective

The Sample Inspection application records tests performed to verify if parts are compliant with minimum requirements.

Currently the application records tests done on Valves. However it can be extended to record tests done on blower, pilots, etc.

## Development team

* From the HHT Lake City plant
  + Justin Galke
  + Jay Lukowicz
  + Terrie Pfeilsticker (initial contact)
  + David Richmond (initial contact)
* For the HNI IT department
  + Molly Banes (initial contact)
  + Luciano Jardim
  + Yogi Naik (initial contact)

## Target users

HHT Lake City QA department will use the application to record the tests done on parts.

## Servers

* Production environment
  + IIS Web site
    - host header: SampleInspection.hearthnhome.com
      * IP: 172.20.11.47
    - Port: 80
    - Windows server: MUSHHT-WEBWB03P
  + SQL Server database
    - Name: SampleInspection
    - Windows Server: MUSHHT-SQLDB05P
    - Port: 1433
    - Instance: default
  + Application userid: HONI\ LKCSampleInspProd
  + Exchange distribution list that receives warnings: LAKECITYSAMPLEINSPECTION@hearthnhome.com
* Test environment
  + IIS Web site
    - host header: qa-SampleInspection.hearthnhome.com
      * IP: 172.20.11.43
    - Port: 80
    - Windows 2012 R2 server: MUSHHT-WEBWB03D
  + SQL Server database
    - Name: SampleInspection
    - Windows 2012 R2 Server: MUSHHT-SQLDB05D
    - Port: 1433
    - Instance: default
  + Application userid: HONI\ LKCSampleInspQA
  + Exchange distribution list that receives warnings: donotreply@hnicorp.com

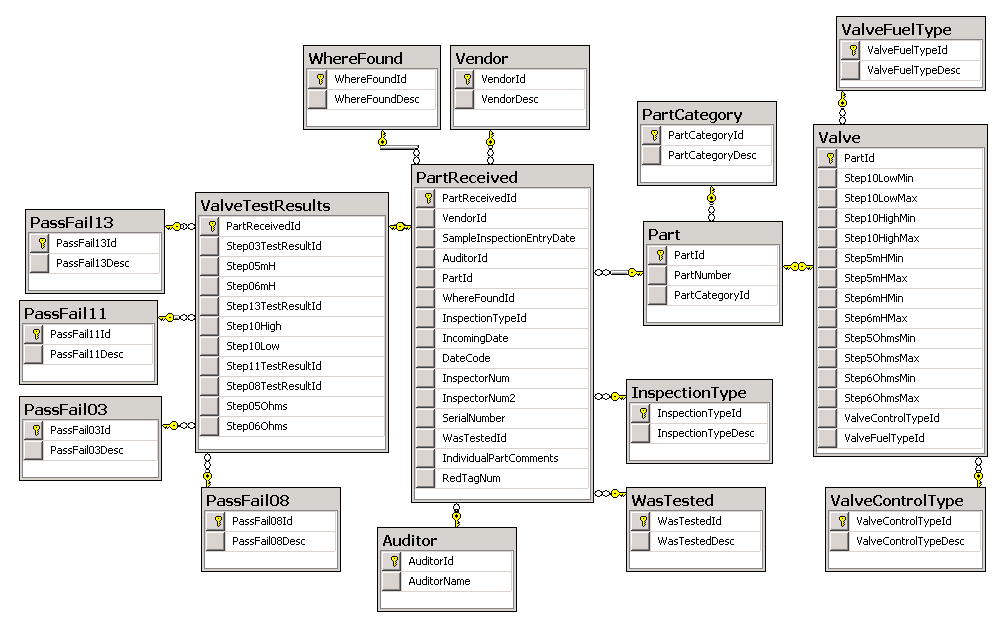
## Basic architecture

* Chrome 49 and Internet Explorer 11 web browsers
* IIS 8.5 Web server
  + C#
  + MVC 5
  + EF 6
  + Additional NuGet packages: PagedList and PagedList.Mvc
* SQL Server 2012 database

## Source code

**Will document location after application is in production**

## Description of the database



### List of database tables:

* Auditor
  + Team members that inspect the parts
* InspectionType
  + How part is going to be inspected
* Part
  + Parts used in the plant
* PartCategory
  + Category of the part, e.g. Valve
* PartReceived
  + Part to be inspected
  + Comments on columns
    - SampleInspectionEntryDate is when test was recorded in the database
    - IndividualPartComments and RedTagNum are usually used if the part fails one of the tests.
* PassFail03, PassFail08, PassFail11, PassFail13
  + States of the tests
* Valve
  + Describes a valve and the passing min/max values (ranges) for tests done with a valve needs
* ValveControlType
  + How a Valve is controlled, e.g. Manual or Stepped
* ValveFuelType
  + Fuel controlled by the valve, e.g. LP gas
* ValveTestResults
  + Result of all tests perform on Valves
* Vendor
  + Provider of a part
* WasTested
  + Describes if a part was tests (yes or no)
* WhereFound
  + Where in Lake City part is used

### List of database views:

* None of the views created is used by the application. They were created to facilitate the data migration.

### List of database UDF (User defined functions):

* None of the UDF create is used by the application. They were created to facilitate the data migration.

### How to add new part categories to the database

The database was designed to allow for more Part Categories to be added to it with minimum effort.

To add a new Part category:

* Add the new category to the PartCategory table
* Add the part numbers to the Part table
* Create new tables to represent the new category, similar to the Valve table
* Create new tables to represent the tests that need to be recorded, similar to the ValveTestResults table

## Description of the code

* The code was developed using Visual Studio 2015 update 2.
* It is based on C# MVC 5 and EF 6 model backed by SQL Server.
* Database first was used in the development of the model.

### Customization of the MVC model

* SampleInspectionMetadataClasses.cs has DataAnnotations created for the model, with emphasis on validating data maintenance on foreign keys and other data validations.
* SampleInspectionPartialClasses.cs has partial classes used to complement DataAnnotations and to validate values within tables that need to be compared.
* SampleInspectionValidation.cs contains multiple methods used in validations that required database access.

### Customization of the MVC controllers

* The controllers were initially generated with scaffolding.
* Most noticeable customizations were the integrations between Part and Valve, and PartReceived and ValveTestResults controllers. If new categories were to be created those would have to be integrated in a similar matter.

### Customization of MVC views

* The views were initially generated with scaffolding.
* Most complex customizations were done in the ValveTestResults Edit view. The user requested to get visual cues for each test done, green meaning passing, and red meaning failure. Also, the user requested to navigate primary using tab (instead of mouse).