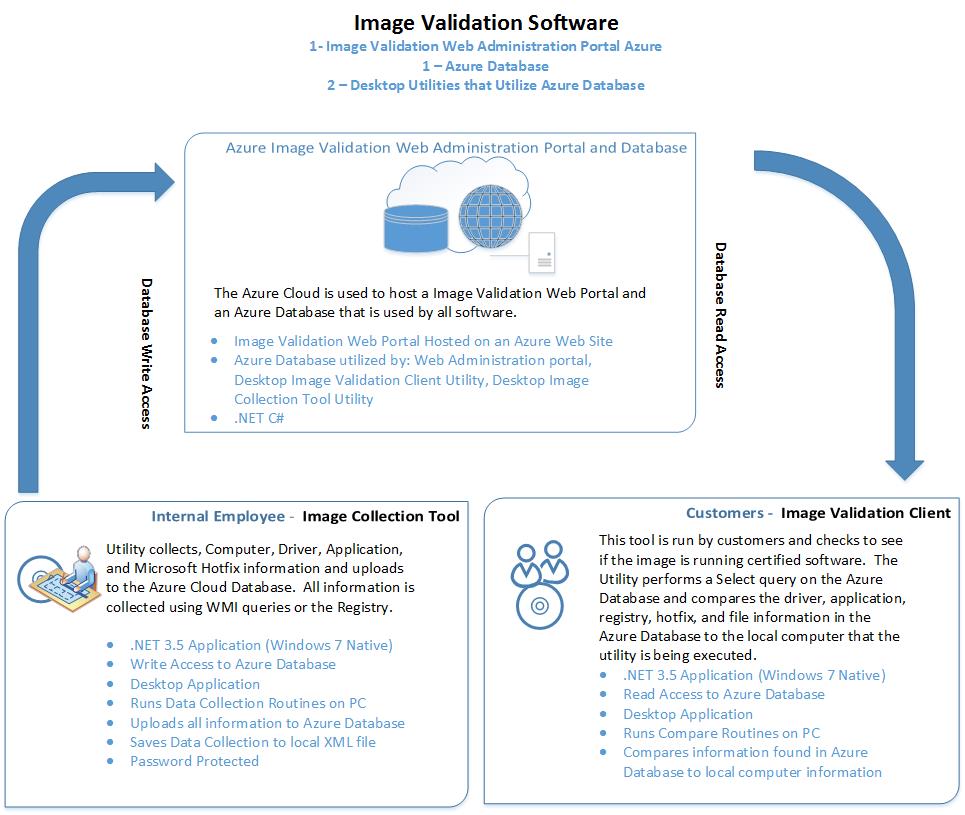
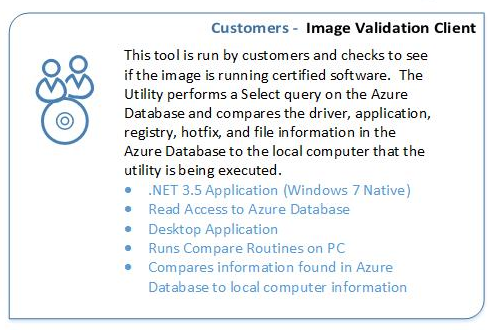
**IMAGE VALIDATION CLIENT UTILITY SPECIFICATION DOCUMENT**

**12/28/2012**

**Full Project Summary:**

****

**Image Validation Client Tool:**

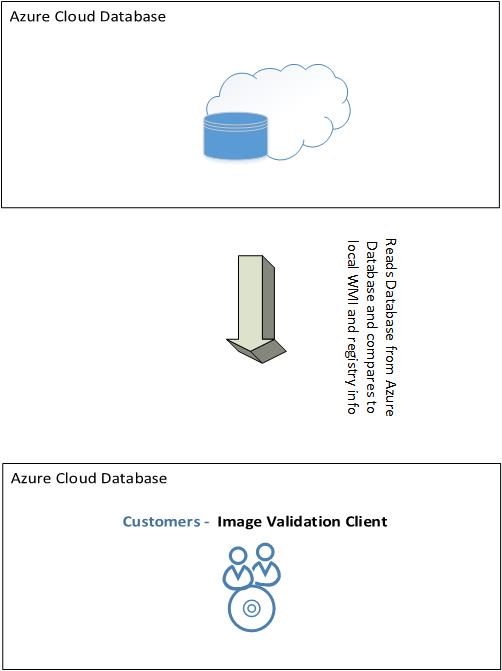
****

**Specifications:**

1. Reads (Selects) Data from the Microsoft Azure Database and compares it to local data
2. .NET 3.5 for native compatibility on Windows 7
3. Programmed in C#
4. WPF Desktop Application
5. Read Access to Azure Database

**Application Overview:**

This application runs selects information found in the Azure Database and compares that information with local information on the computer. The information locally is gathered from WMI queries and registry information.

****

**User and Role Based Security:**

**User Security:**

1. Each user can be assigned one security role.
2. Password should be at least 6 characters long and contain the following
   1. 1 Number
   2. 1 Uppercase Letter
   3. 1 Lowercase Letter

**Role Security:**

1. **Client: This account is used by the Validation Client Tool and only has read permission to the Azure Database. This account is unable to log into the Web Portal. This tool runs with this permission.**
2. Validator: This account has permission to use the Image Validation Collection Tool. The Image Validation Collection Tool uploads Computer Information data to the Azure Database. This account does not have permission to log into the web portal.
3. WebUser:This account has read permission to the Web Administration portal. This account is unable to access User Administration Page.
4. PowerUser: This account has all the permission of the Validator plus it has read and update permission to the Web Administration Portal. This account is unable to perform any delete operations. Unable to access the User Administration Page.
5. Admin: This account has full permissions. It can perform any operation.

**UML Database Design Diagram:**

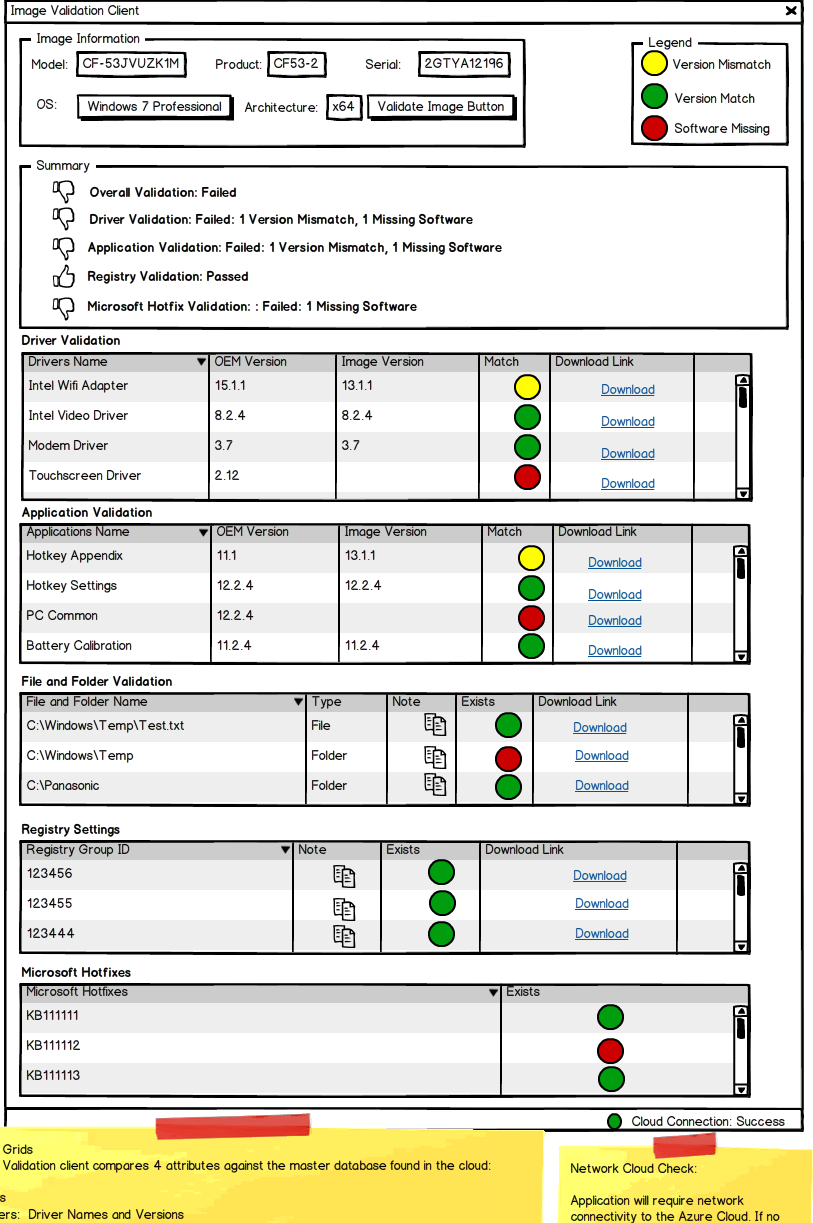
File: [UMLDiagram.jpg](file:///C:\Users\Administrator\Desktop\balsmiq\UMLDiagram.jpg)

The UML Database Design gives a rough draft of a proposed database design for the Image Validation Software System. This can be updated / corrected based on the Developers experience in Database design (normalization), Security and MVC Development.

**GUI Design and Layout:**

**Mockup:**

Below is a draft mockup. If the window height is too large the comparison areas (driver, applications, registry, file and folders, hotfixes) can be placed into separate tabs.

****

**Azure Database Query:**

1. The query should first look if there is an exact Model match with IsPrimaryModel set to True. If no result then…

Example:

* 1. Select \* From Computer Where Model Like “CF-53JVUZK1M” AND IsPrimaryModel = “1”

2. Then a query should search if there is a Product match with IsPrimaryProduct set to True. If no result then…..

Example:

1. Select \* From Computer Where Product Like “CF53-2” AND IsPrimaryProduct = “1”

3. Return a screen that states that the Database doesn’t have a record for that computer.

**GUI Information:**

**Display Computer Information in the Image Information Group from Local Computer Information:**

|  |  |
| --- | --- |
| WMI Query Information | ("SELECT \* FROM Win32\_OperatingSystem", |
| SOurce: |
| WMI WIN32\_Win32\_OperatingSystem |
| Caption: |
| OSArchitecture: |

|  |  |
| --- | --- |
| WMI Query Information | ("SELECT \* FROM Win32\_BaseBoard", |
| SOURCE |  |
| WMI WIN32\_Baseboard |
| pRODUCT |
| Manufacturer |

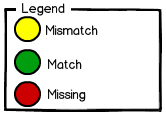
|  |  |
| --- | --- |
| WMI Query Information | ("SELECT \* FROM Win32\_ComputerSystem", |
| SOURCE | **DESTINATION** |
| WMI WIN32\_COMPUTERSYSTEM |
| Model |

|  |  |
| --- | --- |
| WMI Query Information | ("SELECT \* FROM Win32\_BIOS", |
| SOURCE | **DESTINATION** |
| WMI WIN32\_BIOS |
| SERIALNUMBER |

The GUI compares the 5 main areas:

1. Driver Information (Azure Database Driver Table to local WMI information)
2. Application Information (Azure Database Application Table to local registry information)
3. File and Folder Information (Azure Database FileFolder Table to local file information)
4. Registry Information (Azure Database Registry Tables to local registry information)
5. Microsoft Hotfix (Azure Database Hotfix Table to local WMI information)
6. **Perform Driver Information Comparisons between Azure Database and Local WMI Driver Information.**

Run a WMI Query on the local computer using the Win32\_PnPSignedDriver class and compare to the Driver Azure table. If the results match display a green circle, if the driver is missing display a red circle and if the drivers are mismatched present a Yellow Circle. (SEE GUI Mockup for Details)

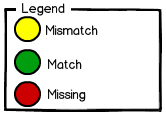


Comparison should only use the Yellow Highlighted Rows

|  |  |
| --- | --- |
| WMI Query Information | ("SELECT \* FROM Win32\_PnPSignedDriver WHERE infname LIKE OEM\*.inf", |
| Compare local WMI Data with | **Azure Database Driver Table** |
| WMI Win32\_PnPSignedDriver | **Azure Database Driver Table** |
| CompatID | CompatID |
| Description | Description |
| DeviceClass | DeviceClass |
| DeviceID | DeviceID |
| DeviceName | DeviceName |
| DriverDate | DriverDate |
| DriverProviderName | DriverProviderName |
| DriverVersion | DriverVersion |
| FriendlyName | FriendlyName |
| HardWareID | HardWareID |
| InfName | InfName |
| IsSigned | IsSigned |
| Manufacturer | Manufacturer |
| Name | Name |
| PDO | PDO |
| Signer | Signer |

1. **Perform Application Information Comparisons between Azure Database and Local Registry Information.**

Search the local application uninstall registry keys and compare to the Application Azure table. If the results match display a green circle, if the application is missing display a red circle and if the applications are mismatched present a Yellow Circle. (SEE GUI Mockup for Details)



Comparison should only use the Yellow Highlighted Rows

**Note:**

In addition to performing the normal comparison the application comparison must check to see if the application has a hardware driver dependency. Certain applications require that certain devices are installed. For example a wireless cell modem software requires that a wireless cell modem is installed. The software shouldn’t be installed if the device is not installed. The query should use the applicationdriverdependency table to check if the application requires a certain device.

Azure Application Query Step Example:

1. Check the azure database to see if the application has a device dependency
2. If the application does have a device dependency check to see if the device exists in the local computer
3. If the device does exists, perform the comparison
4. If the device does not exist, don’t perform the comparison and remove this application from the GUI.

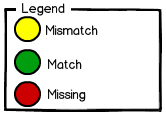
|  |  |
| --- | --- |
| Registry: |  |
| x64 OS Registry Keys: (These Registry KEYS MUST BE USED on X64 OS)  HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\Microsoft\Windows\CurrentVersion\Uninstall  HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall  x86 OS Registry Keys:  HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall | |
| Compare the local registry keys | **WITH THE AZURE DATABASE APPLICATION TABLE** |
| REGISTRY KEYS ABOVE | **Azure Database Application Table** |
| DisplayName | DisplayName |
| DisplayVersion | DisplayVersion |
| Publisher | Publisher |
| VersionMinor | VersionMinor |
| VersionMajor | VersionMajor |
| Version | Version |
| HelpLink | HelpLink |
| HelpTelephone | HelpTelephone |
| InstallDate | InstallDate |
| InstallLocation | InstallLocation |
| InstallSource | InstallSource |
| URLInfoAbout | URLInfoAbout |
| URLUpdateInfo | URLUpdateInfo |
| Comments | Comments |
| AuthorizedCDFPrefix | AuthorizedCDFPrefix |
| Contact | Contact |
| EstimatedSize | EstimatedSize |
| Language | Language |
| ModifyPath | ModifyPath |
| Readme | Readme |
| UninstallString | UninstallString |
| SettingsIdentifier | SettingsIdentifier |

4. **Perform File and Folder Comparisons between Azure Database and Local File Information.**

|  |  |
| --- | --- |
|  |  |
| Compare local file information with | **Azure Database FileFolder Table** |
|  | filefolderid |
|  | computerid |
| Check on local computer to see if location exists | location |
|  | type |
|  | note |

If a note exists display the note icon in the Grid which if clicked displays the note. (see mockup)

If the comparison results in a match display a green circle. If the comparison results in it missing display a red circle.

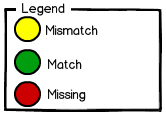
****

**5. Perform Registry Comparisons between Azure Database and Local Registry Information.**

|  |  |
| --- | --- |
| Comparisons should be on the data highlighted in yellow. |  |
| Azure Database Tables (registry, registrygroup, registrygrouping) | with local computer reigstry |
| registry table |  |
| registryid |  |
| key | Compare with local registry |
| value | Compare with local registry |
| valuedata | Compare with local registry |
| datatype | Compare with local registry |
| registrygroup table |  |
| registrygroupid | Compare with local registry |
| filename | Compare with local registry |
| note |  |
| registrygrouping table |  |
| registrygroupid |  |
| registryid |  |
| computerid |  |

If a note exists display the note icon in the Grid which if clicked displays the note. (see mockup)

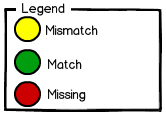
If the comparison results in a match display a green circle. If the comparison results in it missing display a red circle.

****

**6. Perform Microsoft Hotfix comparison between the Azure Database (Hotfix Table) and Local WMI Hotfix Information.**

|  |  |
| --- | --- |
| WMI Query Information | ("SELECT \* FROM Win32\_QuickFixEngineering) |
| Compare local WMI Data with | **Azure Database Hotfix Table** |
| WMI Win32\_QuickFixEngineering | **Azure Database Hotfix Table** |
| CSName | CSName |
| Description | Description |
| HotFixID | HotFixID |
| InstallDate | InstallDate |
| InstalledBy | InstalledBy |

If the comparison results in a match display a green circle. If the comparison results in it missing display a red circle.

****