**Technical Writing Project Cover Sheet**

Capstone Project Title: Creating An Online Certification Database Application \_\_\_\_\_\_\_\_\_

Student Name: Valentine Azbelle

Degree Program: Bachelor of Science, IT - Database Emphasis

Mentor Name: Lee Barker

Signature Block

Student's Signature:\_Valentine\_Azbelle\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Mentor's Signature:\_Lee\_Barker\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Capstone Project Waiver/Release Statement Acknowledgement**

It is the policy of Western Governors University (“WGU”) that student Capstone projects should not be based upon, and should not include, any proprietary or classified information or material belonging to your employer or any other organization (“Restricted Information”) without appropriate authorization.

Please confirm (by signing below) that you will complete (and upload into TaskStream) the IT Capstone Waiver Release form (verbiage is available in Appendix 2 of this document) indicating that your project does not include any restricted content. If you have included restricted content, please confirm that, in addition to the IT Capstone Waiver Release form, you will upload a suitable release letter giving you permission to use restricted information (A sample release letter is available in Appendix 3 of this document).

Valentine Azbelle 11.28.2014

Student’s Ink or Electronic Signature Date Signed**Table of Contents**

[Capstone Report Summary 1](#_Toc159213070)

[Review of Other Work 5](#_Toc159213071)

Project [Rationale and Systems Analysis 9](#_Toc159213072)

[Goals and Objectives 13](#_Toc159213073)

[Project Deliverables 18](#_Toc159213074)

[Project Development 21](#_Toc159213075)

[Addtional Deliverables 24](#_Toc159213076)

[Conclusion 24](#_Toc159213076)

[References 26](#_Toc159213076)

[Appendix 1: Competency Matrix 27](#_Toc159213077)

[Appendix 2: Project Timeline 29](#_Toc159213077)

[Appendix 3: Database Principle Layout 32](#_Toc159213077)

[Appendix 4: Simulation Report Sample 33](#_Toc159213077)

[Appendix 5: Search Page Screenshots 33](#_Toc159213077)

[Appendix 6: Search Results Screenshot 33](#_Toc159213077)

# Capstone Report Summary

As the world becomes increasingly dependent on data driven technology, businesses face the necessity of constantly upgrading their practices and procedures for collecting, storing, and transferring data. The amount of stored data steadily grows every year, and old technologies are simply unable to keep up with the new demands for data storage and access. The employee frustration and customer dissatisfaction is the usual outcome. This project focused on a non-profit organization dealing with those very problems.

NXYZ is a non-profit organization, which is the leading authority on energy performance rating in the window and door industry. It administers a uniform independent rating and labeling system for the energy performance of windows, doors, skylights, and attachment products. Its ratings are used by architects, builders, code officials, contractors, homeowners, and others in comparing different products, ensuring compliance with local codes, establishing performance requirements and standards, as well as leveling the playing field for the manufacturers by having an accurate method of showing the energy benefits of new designs and technology.

The NWYZ office is located in Washington DC. The certification program is being administered through four third party certification agencies located in different parts of the United States. The actual certification process consists of two parts: a computer simulation and physical testing performed by simulation and testing labs accordingly (not necessarily at the same location or even the same lab). The computer simulation is being run for several variants of the product (single pane, dual pane, with or without glass coating, etc.). With the new upgraded version of the simulation software and increased computing power the output of the simulation can be hundreds if not thousands of entries. One of those variants of the product is built by the manufacturer and sent to the testing lab for physical validation testing. The result of the simulation and the physical test get submitted to one of the third party certification agencies where a compliance engineer/inspector checks the simulation model and confirms that the validation test values match the simulation values for the given product within the allowed margin of error. If everything checks out the certification agency issues a printed and signed certificate to the manufacturer allowing them to label their products with the NXYZ logo and the approved simulated values.

Initially the data was being stored in a Microsoft Access database. It was split between the four certification agencies with each having only the data for the manufacturers who certified through that particular agency. The data from the simulation reports was supplied to the certification agency as a spreadsheet paper printout and entered into the database by hand. Every quarter the agencies sent their chunks of the database to the WXYZ office where the data was merged into a single database. A physical book was published annually with the values that were current prior to the publishing date as well as the new data submitted at the end of the year. The certification was valid for a four year period, after which the product line had to be re-certified.

This system was satisfactory when the number of the participating manufacturers was relatively small, the number of certified product lines for each manufacturer was in single (or at the most in low double) digits, and more importantly the number of the simulated product variants usually did not exceed twenty. Overtime that has changed dramatically. Now manufacturers have sometimes over forty product lines, and they simulate virtually every possible variant of the product regardless of whether there is even a slightest chance of actually producing it. The number of entries in the simulation reports coming out of the labs often gets into hundreds for each product line. This combined with the increased number of the participants in the program made manual data entry extremely impractical.

An even bigger issue was the availability of the certification to the manufacturers and public access to the certification data. Technically a program participant was allowed to label their products only when they have a printed and signed certificate from the certification agency. However some manufacturers preferred to etch the certification label into the glass. Which meant they could not start the production until they had the certificate. Furthermore oftentimes there was confusion due to the discrepancy between the labels on the product and the outdated data in the printed book - either the newly certified products had not made it into the book yet or the book was still showing the old values for the product whose values may have had improved with the new simulation upgrade.

In order to structure a proper plan of action we needed to fully understand the requirements. To accomplish this we met with key stakeholders including the head of NXYZ, the certification program administrator, the simulation and test lab representatives, the compliance engineers from the certification agencies, as well as some manufacturers (mainly those on the NXYZ board of directors). As they were directly involved in this process it was crucial that their interests and needs were included in the development plan. The biggest issue from the certification agencies side was the manual data entry and inefficient data transfer between the labs and the agencies. It was slow and prone to errors. Also the user interface in the existing MS Access application was cumbersome and inefficient. The manufacturers were primarily concerned with the timely delivery of the certification. They would like to be able to start labeling their products as soon as the certification agency verifies the simulation model and confirms the validation test. The NXYZ staff would like to get away from the database split into four parts and have a single database with personalized access for each certification agency so that they could only access their specific manufacturers' data. Another issue was the annually printed book. Everybody would like to make the data available in real time. The general consensus among all involved parties was the idea of moving to a more robust database and an online application.

With the information supplied by the stakeholders and with the NXYZ board of directors' approval the initial steps of planning and designing the new application commenced. A study was performed on the available database options. Namely MySQL vs. SQL Server vs. Oracle. The data flow was examined and changes were suggested to the policy concerning the certification procedure to optimize the efficiency. The user interface has been discussed and it was decided to redesign it to both make it more ergonomic and to accommodate the new data flow process.

This design we proposed highligthed the use of the Systems Development Life Cycle (SDLC) methodology and its offshoot designed specifically for database development - the Database Life Cycle (DBLC). The SDLC consists of Planning, Analysis, Detailed Systems Design, Implementation, and Maintenance. Similarly DBLC consists of Database initial study, Database design, Implementation and loading, Testing and evaluation, Operation, and Maintenance and evaluation. Completion of each phase of this design style ensured the successful execution and operation of the application. The results presented constitute the first four phases for both methodologies: Planning, Analysis, Systems Design, and Implementation for SDLC and Planning, Analysis, Detailed Systems Design, Implementation for DBLC. Operation and Maintenance phases are left out of the scope in relation to the project.

The project commenced on January 5, 2015 with a kickoff meeting with the staff of NXYZ. A larger meeting that included principal stakeholders from all parties involved was held the next day and the proposal was presented. The two following weeks encompassed a majority of the research and analysis including travel to the manufacturing facilities, the simulation and test laboratories, the certification agencies, and attending a membership meeting. All that was crucial to understanding the certification flow and developing the correct logistical base, which included the online interface and the supporting database. Knowing that there can be various requirements changes along the way we allotted ample time for the software development and implementation. We were able to complete the project on the planned date May 12, 2015.

The main objective of this project was to deliver an improved data access to all involved parties - the certification agencies, the manufacturers, and the public - in a solution that surpasses the board of directors' expectations and was within budget requirements. At the termination of this project we had provided a user-friendly online application allowing availability of the certification data in real time while at the same time cutting down on manual data entry and thus reducing errors and work time.

# Review of Other Work

A very important issue to address in our project was to select the appropriate database for the solution. We needed to compare various database solutions to see which one was the best fit for the job or even whether we needed to switch from MS Access that was the software used initially. We looked at MS Access vs. MS SQL Server vs. MySQL vs. Oracle. We also briefly looked into the NoSQL solution. From the start MySQL seemed like the most appropriate solution due to its streamlined features designed for speed of execution and no cost. However we decided to assess other available alternatives. Here's what we have found.

First we have looked at MS Access to see whether we could possibly keep it as the back end database for the online certification application. In reality Access isn't really a database. It's just a front end/user interface for the JET/ACE database engine. There are significant limitations associated with JET/ACE database engine:

- Not a true Client Server

- Only supports 255 concurrent users

- Has a 2 GB file size limit

- Has virtually no security

Taking in consideration the above limitations - especially a low concurrent user count and the low security level - it was clear a different database engine had to be found.

Next we looked at three major RDBMS: Oracle, MS SQL Server, and MySQL. All three are very capable pro level database solutions. A brief comparison can be laid out in the following table handily put together by Lee (2013):

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Oracle | MySQL | SQL Server |
| Interface | GUI, SQL | SQL | GUI, SQL, Various |
| Language support | Many, including C, C#, C++, Java, Ruby, and Objective C | Many, including C, C#, C++, D, Java, Ruby, and Objective C | Java, Ruby, Python, VB, .Net, and PHP |
| Operating System | Windows, Linux, Solaris, HP-UX, OS X, z/OS, AIX | Windows, Linux, OS X, FreeBSD, Solaris | Windows |
| Licensing | Proprietary | Open source | Proprietary |

Oracle is probably the most powerful of the three. It has numerous "bells and whistles" such as Oracle Advanced Analytics platform, data reduction to enhance security of sensitive data, new database handling for archiving FDA, several enhancements to Oracle Application Express (a rapid-development tool allowing users to build web applications using SQL and PL/SQL), etc. As such it carries a significant licensing cost, as well as the cost of an expensive DBA required to maintain the database. Frankly the extended capabilities of Oracle RDBMS would be an overkill for this project and there was no reason to spend the extra money.

This left us with MS SQL Server vs. MySQL. There is a wealth of information online on the advantages and shortcomings of both, each having a loyal following. We were leaning towards MySQL from the start but did our due diligence researching SQL Server.

"Microsoft SQL Server is aimed at corporate and enterprise markets, with the pluses of functionality, feature completeness, and inclusion of things they think you may need like graphical user, administration and data modeling interfaces. Fine tuning of user permissions are already included in the product. Microsoft SQL Server is proprietary closed-source software, and you are at the mercy of a company. You however gain in compatibility with Microsoft Windows and Office, you have some better tools, more wizards, and MS-Access just plain works better with it, esp. with date fields. Part of this is the "Apple Principle", where you control the OS and the applications so everything works together with a smoothness not possible with MySQL." (Agnew, 2013)

The differences between the security models in SQL Server and MySQL can be summarized as "Microsoft SQL Server's security is based on controlling what you can get to, MySQL's is based on controlling who gets in".

- Microsoft SQL Server works with sql server logins, trusted logins, and/or Windows domains.

- MySQL totally assumes Internet access, and has username@ipaddress formatting, enabling one to lock down a user to one ip address, or lock out all but one subnet from access at all. Both dbms systems can lock down users by login, database, table, column, and view.

Some important differences are:

- MySQL cannot put security on one particular stored procedure but can put security on who can access any stored procedure, and can lock the data by securing the underlying data.

- Microsoft SQL Server can fine-tune on who can run which stored procedure, as well as lock the underlying data against access.

The major difference between SQL Server and MySQL is the database engine. SQL Server uses SYBASE storage engine. It is slower than MySQL's default MyISAM engine. Not to mention that MySQL can use at least 10 different storage engines, which provides this database with great flexibility and allows it to be custom tailored to the specific project.

Other areas of comparison:

"Backups and Recovery:

Both systems can do log-based recovery and backups.

- Microsoft SQL Server has backup and recovery systems builtin. Here at VCU we use IBM's Tivoli to backup databases w/o using backup files.

- MySQL can dump to flat SQL files at pre-determined intervals, it is up to you to put them somewhere that is safe. There is no Tivoli client for MySQL, one has to backup the flat SQL files. The format of these can be made ANSI by a MySQL setting for reading into Microsoft SQL Server.

Replication:

Both systems can do log-based replication.

- Microsoft SQL Server has several replication schemes, snapshot, transactional and merge.

- MySQL replication is based on the transaction log only.

Report Generation:

- Microsoft SQL Server has TONS of report generating capacities, plus Microsoft Access can natively access SQL Server databases.

- MySQL is very sparse in this."

(Agnew, 2013)

MS SQL Server sounded very close to what we were looking for. However the license for the Standard Edition (the least acceptable version for our purposes) carries the cost of $5000. The Enterprise Edition, which is what we would have recommended as a future-proof solution costs $20000. The board of directors did not want to spend that kind of money. Considering MySQL was free and provided the same or better capabilities in the key areas of this project it has been decided to proceed with MySQL implementation.

# Project Rationale And Systems Analysis

As it is imperative to follow a proper design methodology, the System Development Life Cycle (SDLC) was used for the project. Within that specifically for the database we used the Database Development Life Cycle. Both methodologies consist of several iterative steps and are similar. The *Planning* and the *Database initial study* phases were essential to the success of the project. Executing a thorough analysis helped us evaluate the existing system, and collecting the user requirements assured that the application was designed to meet the demands placed on it by the stakeholders. Next, the *Detailed system design* phase, which incorporated the *Database design* phase, mapped out the concept for the application build including defining the detailed system specifications, creating the conceptual design for the database, selecting the appropriate DBMS software, laying out the database logical design, and then creating the database physical design. Armed with the conceptual design for the application we could proceed onto the next phase of *Implementation*. This is where all the coding, testing, and debugging was performed, the database software was installed, the database was created, loaded, and filled with the data. Next we concentrated on the database *Testing and evaluation* phase where we fine-tuned the database through comprehensive testing in conjunction with the application to make sure the database performance was optimized for speed and the application worked with the database seamlessly. During the DBLC's *Operation* phase we would insure that the database produces the required information flow. The *Maintenance* phase for both the application and the database would conclude our project. The *Operation* and *Maintenance* phases are out of the current scope for this project.

To get a clear picture of what needed to be done we spoke with representatives of each entity involved in using the database. The issues were different depending on whom we talked to. However there was one common theme - a slow and awkward data flow.

In order to properly evaluate the data flow we followed the certification procedure from start to finish. We visited the manufacturing facilities, the simulation labs, the testing labs, the certification agencies', and the NXYZ office in Washington DC to observe how the users interacted with the previous data solution.

The process started with the manufacturer sending an AutoCAD drawings of their window design to the simulation lab. The drawings were usually printed and mailed. The simulation lab engineer (for simplicity called a simulator) then would recreate the drawings the best they could in the simulation drawing software called Window. It was a very crude approximation of the original drawings. At the time of our analysis an old version of Window was used. It had a poor graphical interface and couldn't even draw curves - the simulator had to approximate curved surfaces with drawings of different size blocks. The created drawing would be loaded into the simulation software called Therm. Therm would produce four values for each modeled variant of the window: two U-values (the thermal energy transmittance coefficient) - for the residential and non-residential sizes, solar heat gain coefficient (SHGC), and visual transmittance (VT). The results were output in a Microsoft Excel spreadsheet and then printed out. The printed copy was stapled to the drawings from the manufacturer and a CD with the simulation files, and then mailed to the certification agency. Independently of the above the manufacturer built a physical sample of the window (usually the cheapest variant), which was boxed up and shipped to the test lab for validation. The test took a couple of days. After it was completed a report was printed out with the results - the U-value for the given size (usually not the size simulated - the test option had to be simulated in addition to the standard simulation sizes). The test report was mailed to the certification agency. Once the agency had received both the simulation and the test reports, the compliance engineer reviewed the simulation, the validation test values, and then - if there were no errors - proceeded to enter the data from the simulation report into the Access database by hand via VB forms. The simulators weren't even allowed to use the native MS Access interface as they were kept out of the raw tables for security. Realistically, even if they were allowed in the tables considering that each lab submitted the data in their own format, there was no way to simply copy and paste the data from an Excel spreadsheet into an Access table. We were shocked at how inefficient this system was. When reviewing the data for about a dozen product lines we found three errors introduced by manual data entry - one happened in the simulation lab when the simulator was typing the report copying the data from Therm into Excel and the other two crept in when the compliance engineer was entering the data from the Excel spreadsheet into the Access database. After the data had been entered a report was printed out, signed by the certification engineer, and mailed to the manufacturer. Which officially permitted the manufacturer to place labels on the product, thus concluding the certification process.

However this was not the entirety of the system functionality. Our analysis happened around the time when the annual book was being published. And we observed manufacturers desperately trying to get their products certified before the submission deadline for the book (otherwise they would have to wait another year to get their products into the publication). The workload on the labs and especially on the certification agencies increased tenfold. There was a lot of frustration on all sides as well as a disappointment and resentment when some of the product lines did not make the deadline simply because it was humanly impossible to process so many product lines in such a short time.

It was evident that such system could not sustain the certification program, which was growing in both the number of the participants and the number of products they certified. The major hindrance appeared to be the absence of an instantaneous access to real time data - something an online application would provide. Clearly the entire process had to be overhauled to allow the data remain in the digital domain as much of the process as possible, starting with the drawings submission.

This led us to meet with the simulation software developers who were actually working on the new versions for both Window and Therm. We have suggested that Window should be able to import AutoCAD drawings and allow the simulator to trace the drawing lines instead of trying to recreate the entire drawing from scratch. This suggestion was enthusiastically accepted and the corresponding changes were released in the new version of Window. We have also requested for Therm to have the ability to output data in a format compatible with MySQL database, which was also accomplished.

With those tools being finally available we could proceed to map out the new data flow. Which goes as follows. The manufacturer would submit the digital files of the products' AutoCAD drawings to the sim lab. The simulator would import the files directly into Window and trace the lines of the submitted drawings. This would make the job of the simulator immensely easier and less time-consuming. All the simulator would need to do is fill in the specified materials and the air cavities, and then define the boundary conditions for the simulation. The results of the simulation would be exported out of Therm directly into the database table where they would wait for an approval from the certification agency. The compliance engineer at a certification agency then could access those results on his/her end over the VPN immediately after the simulation was completed. The test results would also be entered directly into the database by the test lab. Therefore the simulation could be validated (or rejected) automatically based on the permitted margin of error. No manual data entry. All the compliance engineer would have to do is to match the sim report to the test report and highlight the product in the sim report which was the test option. Once the simulation files have been reviewed, and if the test validates the simulation, the compliance engineer can release the data in the database to become visible to the public. A certification report would be automatically generated and e-mailed to the manufacturer, which would allow them to label their products literally within seconds of the products being certified.

The printed publication would become obsolete as all of the certification data becomes available online as soon as it is approved by the certification agency. No more submission deadlines and no more waiting a year for a product to appear in print and become accessible to the public.

# Goals and Objectives

This project was intended to achieve two primary goals. The first one was to overhaul the existing certification process in order to drastically reduce the time it takes to certify a product with NXYZ while simultaneously eliminate data transfer errors introduced by manual data entry. The second was to make the certification data available to all concerned parties in real time. These goals were accomplished by completing the following objectives:

* Build a database capable of handing the increased amounts of data and large numbers of concurrent users
* Create an online application capable of accessing the database and displaying the data to the users in an easy to navigate format
* Create a user interface allowing the certification agencies access to their specific data and control over its release to the public
* Automate output of the data from the simulation software into the database thus eliminating manual data entry
* Revise the certification procedure to be in line with the newly adopted technology
* Meet proposed deadlines and remain within the budget scope

To attain the desired result all objectives had to be completed successfully. Each objective represented a crucial piece of the fundamental change to the certification process and the program as a whole. A failure to execute any one of the objectives would result in the failure of the entire project.

Before we could proceed with writing the code for the database we needed to determine which database software we were going to use to house our data. After comparing various database solutions we agreed on MySQL - mainly because it was free but had virtually the same functionality and performance as the other two candidates - Oracle and MS SQL Server. With that out of the way we moved onto the second phase of DBLC methodology - *Database design*.

In order to have a clear picture of what our database would look like we first mapped out a logical design for it. We decided what kind of tables had to be in the database, what kind of attributes (columns) they would have, and drew diagrams with the logical relationships between them. Armed with the schema diagram we began writing the actual SQL code that would create the tables and columns we mapped out.

Our main table - the one containing the actual product data - turned out to be too large and we needed to denormalize it in order to speed up the performance. We also added the necessary indexes for the search function to be implemented through our online application. In order to preserve raw data and prevent users inadvertently modifying it, we created user views, which are to be used for data access. This also allowed us to separate the data between the four certification agencies so they could only view and modify the data for the manufacturers that were assigned to them. Various constraints (such as primary/foreign key pairs, NOT NULL, and unique) were added where necessary in order to preserve data integrity. A user cannot enter a product number into the Products table if the manufacturer to whom that product belongs isn't listed in the Manufacturers table (no more orphan data as it was sadly the case with the previous application).

Concurrently we were developing the front end application. We paid special attention to the interface. A user won't care about how well the database is designed on the back end if the front end is poorly thought out and awkwardly built (which in part explains the popularity of MS Access - it has an excellent user interface). After speaking with the labs and the certification agencies we have come to a conclusion that the main interface needed to resemble the original simulation submission sheet - i.e. the specific data is to be presented in a table with the header for the common data. The compliance engineers made a request to allow for as much of the table as possible to be visible on the screen at one time. This ruled out the frames for the interface design (implemented via divs, of course), which is what we had originally conceived. This request was problematic because there is no way to know in advance how many records would be returned by the search, and if the number of records is too large, the loading time for the page could be unacceptably long. Ultimately we came up with a compromise - a paged design where if the number of records exceeds a specified figure (user defined) the table will split and be displayed on more than one page. In order to prevent an accidental call for too many records on one page we placed safeguards in place such as the default number of records to be returned on one page to be 200. This figure can be changed if desired to be more or less but to no more than 1000 (which in our test runs took at least 20 seconds to load, and that was the longest time we deemed acceptable). Also one cannot return ALL records from the database - the search must have at least one argument. From what we've observed during the *Analysis* phase of the project there is rarely the need to return more than 1000 records at a time and therefore limiting the number of records displayed on one page to a thousand does not impede the workflow in the vast majority of cases.

The validation test report page was redesigned completely from the test page in the previous application. One minor (but a very important) improvement was the way the fields are rearranged on the page and how the cursor moves when tabbed between the fields. In the previous application clearly not much thought was given to how the user actually interacts with the page. The fields, among which were a couple of drop-down boxes, were arranged vertically in two columns but the tabbing went left-right instead of following the logical sequence between the fields. As a result the user had to constantly move from the alphanumeric keyboard to the numeric keyboard to the mouse. We redesigned the page to preserve the logical sequence and minimize the movement between input devices. Furthermore, we built in the ability to type in the drop-down boxes (before absent from the app) and thus eliminate the necessity to use the mouse altogether.

A special search function was built in the database and installed on the test report page designed to find the test product in the simulation report. According to the revised procedure the simulation lab has to mark the option on the simulation report which corresponds to the physical sample that was tested. A column of the TINYINT(1) data type (MySQL Boolean data type) was created in the Products table, which if checked indicates that the record is for the test option. This is necessary as the test sample usually is built to a real life size and not to the standard simulation size. With the test option search function the user only has to enter the simulation report corresponding to the test report, and the values from both reports are automatically displayed and compared. If the test does not validate the simulation the data is locked. The certification agency is prevented from issuing the certificate for the given product line and releasing the simulation data to the public. The simulation report is automatically rejected and a notification is sent to the sim lab that the report must be revised and resubmitted. If the test report validates the simulation a button "Certify" becomes available to the certification agency. The compliance engineer still has to review the simulation files before certifying the product, and once the files are reviewed, pressing the "Certify" button issues a certificate to the manufacturer (via e-mail or, if preferred, prints out a copy for mailing) and releases the data for public access.

When it came to merging Therm with MySQL originally we thought we would have to import Therm's output in a spreadsheet format. Which MySQL is definitely capable of but it's a somewhat awkward solution. The Therm developers graciously saved us a lot of work by simply creating a function for output in .frm format. This allowed us to merge Therm with MySQL seamlessly.

With regards to changing the certification procedure itself we had to overcome some hurdles. A non-profit trade association like NXYZ is governed by its members and all policy changes happen under the general vote at the quarterly meetings. In our experience any change - even if it benefits literally everybody - is usually met at least with some resistance. Some of it is due to internal politics, some - due to people's lack of understanding of the new technology and wanting to have a tangible object like a printed copy, some - due to the general inertia and opposition to change. What we are suggesting removes the element of a physical signature from the process, and a lot of participants appear to be uncomfortable with that. But we are confident that in the end the common sense (and the desire to avoid losing money while waiting for weeks for the certification) will prevail, and the appropriate policy changes will be voted in.

# Project Deliverables

The key deliverables for this project were the back end database and the front end application.

The database was created in MySQL and included tables for the manufacturers, product lines, products, validation tests, as well as other tables that are maintenance related. The design is such that the data from the simulation report is initially held in a Temp\_products table until it is validated by the data in the Test\_report table and verified by the certification agency. Once the product is certified the sim report data is copied into the Products table, from which the data becomes visible to any user via a read-only view.

The certification cycle lasts 4 years. After which the product line must be recertified, i.e. re-simulated and re-tested. An archiving functionality was built into the database, which automatically archives the products whose simulation report is over four years old. However many manufacturers often take advantage of updated simulation software (if it yields better results) and recertify their products in the middle of their current certification term. In that case the new data replaces the old data, which is automatically archived, and the certification term is counted from the date of the new simulation report.

A rounding function is used for displaying data that has finer than 0.01 precision. Therm outputs data in 0.00001 precision, and while the data in the Products table is stored in this precision, the view displaying it to the public rounds it up to two significant digits as it is the precision required by the NXYZ documents and is used for the values displayed on the labels.

Various security measures were implemented. Initially the board of directors wanted to keep the servers in the headquarters office. We have done an analysis and showed them that having an outside hosting service is a much more practical solution. And less expensive in the long run. Therefore for hosting the database an outside hosting service was contracted. The service provides separate servers to host the database and the front end application, and handles the general online security. Access to the database is granted via VPN with individual login and password for each user. Internally different privileges are enabled for each user depending on their role thus ensuring that the users see only the data they are supposed to see and have no way to access the data they don't own or weren't granted the permission for.

The front end application was written in ASP.NET. The UI was created in accordance with the requests of the users. The product data is displayed in a tabular format resembling that of the spreadsheet from the simulation report to facilitate a seamless transition from the printed out copies of Excel spreadsheets to a fully digital data flow. The test report page allows automatic comparison of the test values and the simulation values for the specified sample. The simulation data is automatically rejected or accepted based on the result of that comparison. In the case of the latter the certification agency has an option to press the "Certify" button, which issues and e-mails an electronic certificate to the manufacturer, as well as copies the data into the Products table making it visible to the public via the read-only view created for this purpose. The certificate can also be printed out for mailing.

The centerpiece of the application is the search page. Users can search the database by manufacturer name, manufacturer number, product line name, product line number, and product number. The search incorporates drop-down lists with a type-ahead function. The number of records displayed on one page is limited to 200 by default. A user can change that figure to up to 1000. If the number of records returned exceeds the specified limit the extra records are displayed on another page, which can be accessed by clicking on the page number appearing at the top of the page in the header. The header is always visible but is kept small to allow for as much of the table underneath it to be visible without scrolling.

The application has a reports page accessible only to the NXYZ staff and the certification agencies. Based on the privileges and permissions granted the users are able to view and print various statistics reports, such as a number of the active manufacturers in the program, the number of the current product lines, the number of products per manufacturer, the upcoming expiration dates, and the like. A function automatically notifying the manufacturers of their expiring certifications was also implemented. It sends an e-mail to the manufacturer contact listed in the database with the list of expiring certifications 3 months and 1 month before the expiration date.

The application is hosted by an outside service on a different server from the database. The online availability of the website is maintained by the service. The service also provides numerous backup options as well as test servers for testing updates.

# Project Development

During the initial *Planning* and *Analysis* phases of the project we interviewed the stakeholders at different levels of the organization - from the manufacturers and the labs to the certification agencies and the NXYZ staff. We made sure to include everyone who was involved in the certification process in our discussion about the new application to gain different perspectives and to approach the database and the front end app design with the goal of making it equally user-friendly for everybody. Our investigation allowed us to thoroughly understand the process and the data flow, which was meticulously documented . This documentation was used to make informed decisions on how to implement the new software. We researched multiple means to achieving the goal of improving the data flow and data storage and provided the justification to the management on why changes were necessary and what was needed to complete those objectives.

Early on we were faced with an issue we had not anticipated initially. Once we had a clear picture of the data flow it became fairly obvious to us that the new software implementation alone cannot solve the problem. The very process of certification had to change. NXYZ is a large organization comprised of hundreds of manufacturers from both the US and overseas. The NXYZ staff does not have the authority to make a change like that - the order of the certification process is written into the organization's bylaws. Only the general membership can approve a change of such significance in the quarterly meeting by a vote. Luckily such meeting was scheduled at the end of January and we were invited to attend. We made our case to the membership but to our surprise the proposed changes were met with a lot of resistance. People appeared to be set in their ways and were extremely reluctant to accept changes even though it greatly benefited them either in reduced costs, or in decreased wait time, or in a more efficient and productive labor. Claims as ridiculous as "What if I don't have a computer" were made by some members in opposition to moving the certification entirely into the digital domain. However after two days of deliberations and with several amendments our proposed changes to the certification process were accepted and the corresponding language was written into the organization's documents. Now we could start working on the software.

The first week of February we began the *Detailed system design* phase and within it - the *Database design* phase. Once we compared available options for the solution and worked out a single comprehensive strategy that addressed all of the objectives we arrived at in the *Planning* and *Analysis* phases we proceeded to design our application. We drew a logical design for the database allowing us to see clearly what kind of tables we would need and how they would be connected within the relational database. We then wrote the code needed to create the database and the tables along with the appropriate constrains, indexes, sequences, and views. Concurrently we worked on the front end application. The GUI interface was first laid out in accordance with the requests from the stakeholders and then created as an executable file. The corresponding code was written to create functionality within the GUI including the correct tabbing through the text boxes and the security checks.

With the design in place we moved onto the *Implementation* phase of the project. First we found and contracted an outside hosting service provider. Once allocated a dedicated server we installed the database software and loaded the database, then filled it with the data. The front end app was connected to the database. We checked that all the dropdown boxes were showing the correct options and where that was not the case we went back and corrected the code. We set up a dummy website with the new code and gave the select users access credentials to see if what we have done is to their liking. There weren't a lot of complaints - a testament to how exhaustively thorough our research was during the *Planning* phase - although we did have to change a few minor things. Mainly some people have discovered that they didn't like the way things looked on their screens - even though the screens were done exactly the way they initially requested - and we had to make adjustments that addressed these new requirements. For example, the test report page initially had two columns of text boxes. Some users at certification agencies made a request to fit everything into one column. Which we did.

After the initial design was accepted by the users and signed off on by the NXYZ staff we moved on to the next phase - *Testing and evaluation*. Due to our unexpected deviation from the timeline with the need for the membership approval of the fundamental process changes we were running a week behind. On the 1st of April we began exhaustive testing on both the front end code and the database. We wrote numerous unit tests to make sure every possible scenario with erroneously entered data was being handled by the software. We ran the security checks on the database to ensure it was protected from all known attacks. We tuned the database to return the search results in the shortest time and then tested the accuracy of the results. We worked hard to make up the lost time and were able to catch up to our initial schedule during this and the next phases.

With all the phases complete we delivered the final product to management and officially closed the project on May 11 as planned. It was immediately evident how much more efficient the new system was. The certification process took days instead of months and the data was free of human error. The huge file cabinets containing thousands of paper simulation and test reports eventually disappeared from the certification agency offices as all documentation was passed along in the digital domain from start to finish. The inspectors were freed of manual data entry and now could concentrate on reviewing the simulation files, which was their main purpose. And most importantly the consumers now had instant access to the manufacturers' current data. Everybody at NXYZ was happy with the result that we were able to achieve in the completion of this project.

# Additional Deliverables

Included as appendices to this report are items relevant to the core of the project. The competency matrix and project timeline are appended as Appendix 1 and 2 accordingly. Provided the NXYZ database principle layout (Appendix 3), a sample of the simulation report, which contains the data to be transferred into the database (Appendix 4), progressive screenshots of the search page in the online application (Appendix 5), and the search results screenshot (Appendix 6).

# Conclusion

The project commenced and concluded under budget and on schedule. With the exception of the unexpected involvement in the organization's policy change everything went more or less smoothly and as expected. Once the application was fine-tuned, tested, and delivered, the immense time and labor savings were immediately evident. A lot of previously necessary menial labor has become obsolete - the manufacturers no longer had to check their data for accuracy 3 times (for each manual data transfer) and the engineers did not have to waste hours on tedious and error-prone typing. The manufacturers could have the certificate for a product line literally seconds after the certification was granted by the IA and start labeling their product immediately. The consumers viewed current data and not something that possibly hasn't been valid for almost a year. The NXYZ staff has received tools to track and manage the data in a much more efficient manner via specifically tailored built-in database queries. Because we were thorough in our research and made an effort to interview all participants of the process we were able to discover the need and meet all the demands expected. Once again we learned the value of having proper documentation. This helped prevent uncontrollable scope creep and keep us on track, especially when new requirements were introduced. We worked very hard to complete the project on time despite losing a week on issues not related to the software development. Which has shown us the importance of preserving our timeline. It allowed us to stay under budget and deliver the project before the deadline. All these experiences afforded us a solid foundation to achieve success in future projects.

# References

O’Hearn, S. (2010). *OCA Oracle Database SQL Certified Expert Exam Guide – (Exam 1Z0- 047).* New York, NY: Oracle Press.

Ramklass, R., & Watson, J. (2008). OCA *Oracle Database 11g: SQL Fundamentals I: Exam Guide (Exam 1Z0-051).* New York, NY: Oracle Press.

Van Der Lans, R. F., & Cools, D. (2012). *SQL For MySQL Developers.* New York, NY: TRN Murach, J. (2012). *Murach's MySQL.* New York, NY: Mike Murach & Associates

Schumacher, R. (2013). Why Move to MySQL from Microsoft SQL Server? *MySQL.* Retrieved from http://dev.mysql.com/tech- resources/articles/move\_from\_microsoft\_SQL\_Server.html

Lee, J. (2013, November 30) Oracle vs. MySQL vs. SQL Server: A Comparison of Popular RDBMS. *Udemi/Blog.* Retrieved from https://www.udemy.com/blog/oracle-vs-mysql-vs-sql-server/

Agnew, J. (2013, March 25 ) A comparison of MySQL vs Microsoft SQL Server, how to decide which to use? *VCU.edu.* Retrieved from http://www.people.vcu.edu/~agnew/Misc/MySQL-MS- SQL.HTML

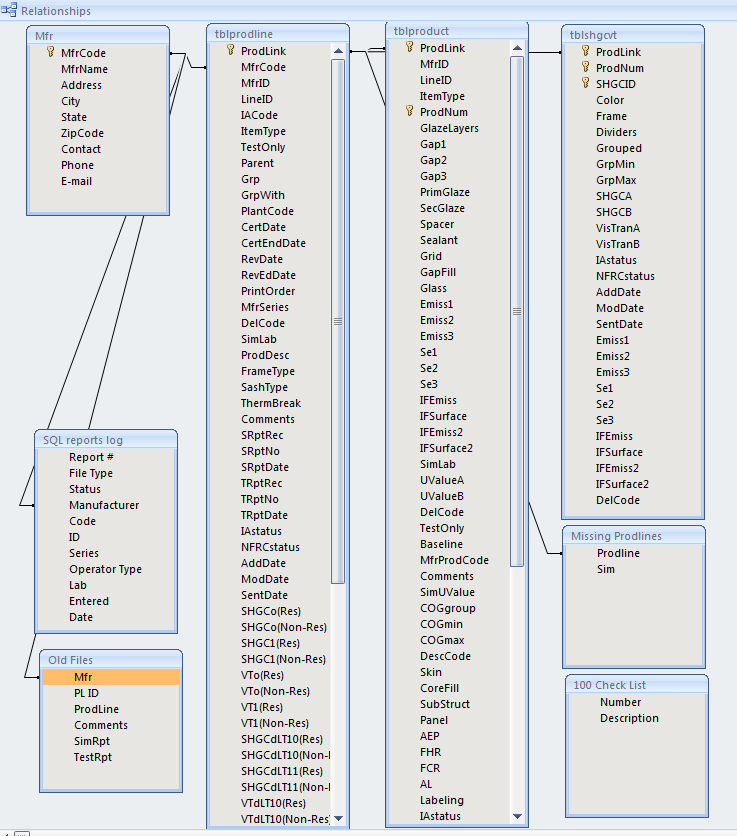
**Appendix 1: Competency Matrix**

|  |  |  |
| --- | --- | --- |
| **Domain/Subdomain** | **Competency** | **Explanation** |
| Leadership and Professionalism | Project Scope | Understanding the project scope is the first thing one must do before starting a project. I have described and explained the key components of the project scope. This has ensured the successful completion of the capstone project. |
| Language and Communication | Mechanics | In the capstone report have correctly applied rules of grammar, syntax, and punctuation. |
| Language and Communication | Documenting Sources | In the capstone report all references and in-text citations follow APA style formatting. |
| Language and Communication | Adaptation | The capstone project was designed to adequately address the needs of the NXYZ trade association. |
| Quantitative Literacy | Constructing Arguments & Reasoning | As part of the capstone project in order to determine an appropriate DBMS software I have compared 3 candidate applications, as well as the application used previously, and demonstrated that the best DBMS for the job was MySQL. |
| Language and Communication | Inquiry and Research | In my research for the best fitting DBMS software I studied numerous sources concerning the available options for the database solution and distilled the information into a concise comparison demonstrating the reason for the DBMS chosen in my capstone project. |
| Upper Division Collegiate Level Reasoning and Problem Solving | Problem Identification and Clarification | During the *Analysis* phase of the capstone project I observed and analyzed the previous certification process from start to finish and correctly identified its multiple flaws and issues. |
| Upper Division Collegiate Level Reasoning and Problem Solving | Reaching Well-Founded Conclusions | After identifying the problems of the previous certification process I have arrived at a viable solution to those problems - such as creating an online database application. This was the basis for the successful completion of the capstone project. |
| Leadership and Professionalism | Project Planning | A well thought-out plan was paramount to the success of the capstone project. I created a project plan based on the SDLC and DBLC methodologies. I also created a timeline for the project. |
| Leadership and Professionalism | Strategic Planning | I have explained in depth how the NXYZ certification process must be overhauled in order to make it more efficient and to allow NXYZ have even more impact on the window and door industry. This explains the need for this capstone project. |
| Database | Relational Database Design and Application | In the capstone project I reviewed and selected the appropriate database design, as well as identified design solutions that addressed the application needs. I created the conceptual and logical designs for the database, and then built a fully functional application. |

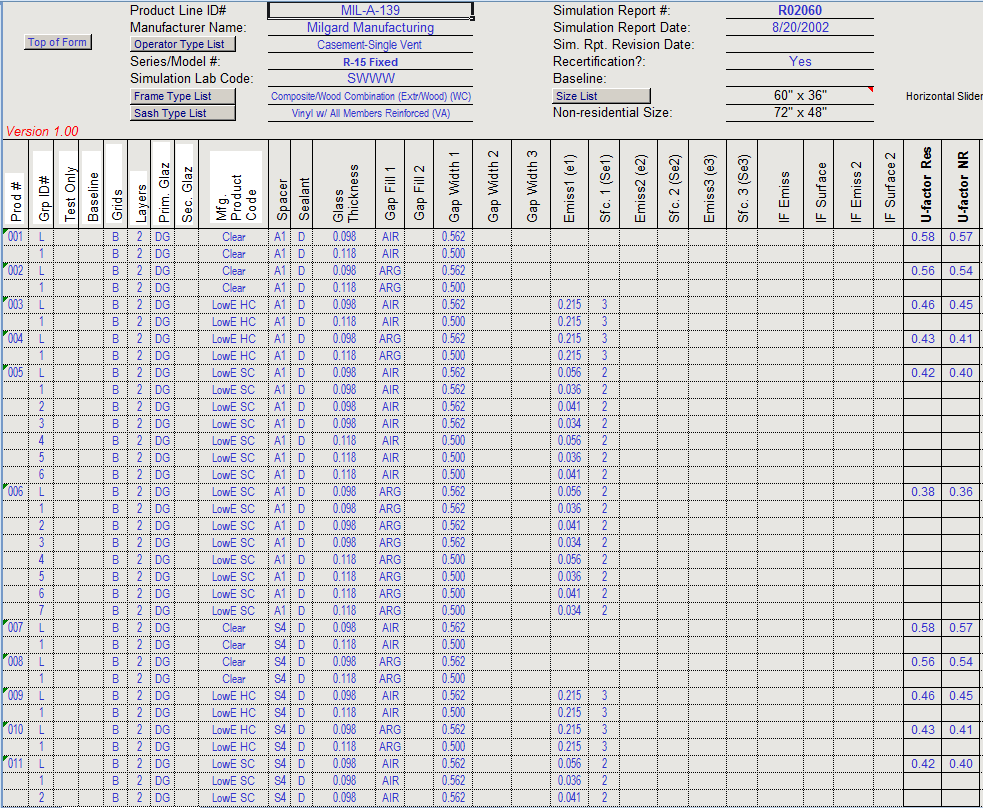
# Appendix 2: Project Timeline

|  |  |  |  |
| --- | --- | --- | --- |
| **Project Deliverable or Milestone** | **Duration** | **Planned Start Date** | **Planned End Date** |
| **Project Phase: Planning and Database Initial Study** | | | |
| Initial assessment | 1 day | 01/05/2015 | 01/05/2015 |
| Meet with stakeholders | 1 day | 01/06/2015 | 01/06/2015 |
| Define objectives | >1 day | 01/07/2015 | 01/07/2015 |
| Define the scope and boundaries | >1 day | 01/07/2015 | 01/07/2015 |
| Obtain signoff | >1 day | 01/07/2015 | 01/07/2015 |
| **Project phase: Analysis** | | | |
| Obtain user requirements | 1 day | 01/08/2015 | 01/08/2015 |
| Existing system evaluation | 14 days | 01/09/2015 | 01/23/2015 |
| Meet with stakeholders | 2 days | 01/24/2015 | 01/25/2015 |
| **Project phase: Detailed System Design and Database Design** | | | |
| Create a conceptual database design | 2 days | 01/26/2015 | 01/27/2015 |
| DBMS software selection | 1 day | 01/27/2015 | 01/27/2015 |
| Create logical database design | 2 days | 01/28/2015 | 01/29/2015 |
| Create detailed system specification | 3 days | 01/30/2015 | 02/01/2015 |
| Create physical database design | 14 days | 02/02/2015 | 02/16/2015 |
| Attend the general membership meeting | 3 days | 02/24/2015 | 02/27/2015 |
| **Project Phase: Implementation** | | | |
| Contract an outside hosting service | 2 days | 03/01/2015 | 03/02/2015 |
| Coding, testing, and debugging | 30 days | 03/03/2015 | 04/01/2015 |
| Create GUI for sim and test report pages | 2 days | 03/04/2015 | 03/05/2015 |
| Create the search page for the products | 2 days | 03/06/2015 | 03/07/2015 |
| Create reports page | 1 day | 03/08/2015 | 03/08/2015 |
| Install the DBMS | 1 day | 03/09/2015 | 03/09/2015 |
| Create the database | 1 day | 03/10/2015 | 03/10/2015 |
| Migrate the data into the database | 2 days | 03/11/2015 | 03/15/2015 |
| Install the application | 1 day | 04/02/2015 | 04/02/2015 |
| **Project Phase: Testing And Evaluation** | | | |
| Test the database | 5 days | 04/03/2015 | 04/08/2015 |
| Fine-tune the database | 5 days | 04/03/2015 | 04/08/2015 |
| Fine-tune the application | 5 days | 04/03/2015 | 04/08/2015 |
| Evaluate the database and the application | 2 days | 04/09/2015 | 04/10/2015 |
| **Project Phase: Operation and Maintenance** | | | |
| Produce the required information flow | 5 days | 04/11/2015 | 04/16/2015 |
| Introduce changes | 23 days | 04/16/2015 | 05/09/2015 |
| Make enhancements | 23 days | 04/16/2015 | 05/09/2015 |
| **Project Phase: Closing** | | | |
| Release the database and the application to NXYZ | 1 day | 05/10/2015 | 05/10/2015 |
| Receive written customer approval | 1 day | 05/11/2015 | 05/11/2015 |
| Perform lessons learned analysis | 1 days | 05/12/2015 | 05/12/2015 |

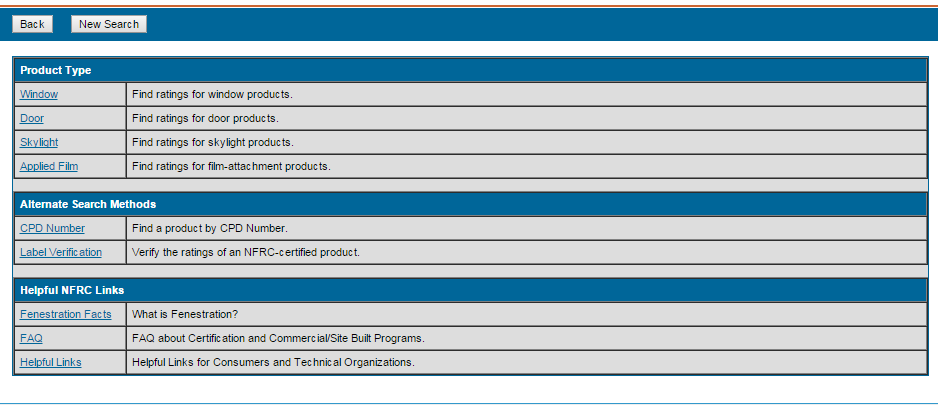
# Appendix 3: Database principle layout

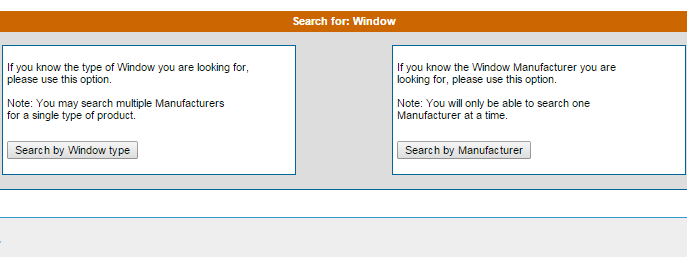


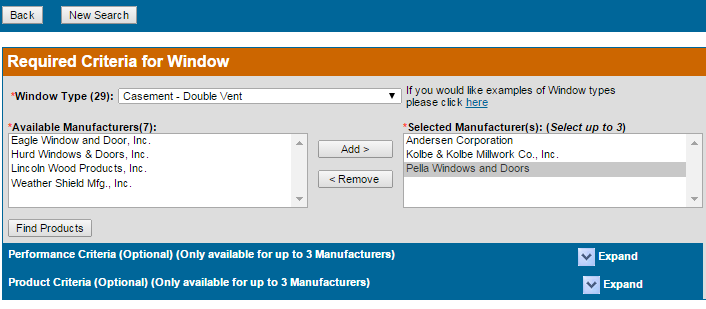
# Appendix 4: Simulation report sample

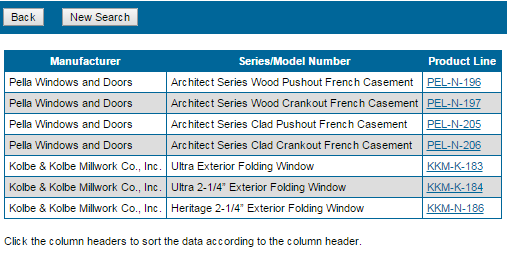


# Appendix 5: Search page screenshots

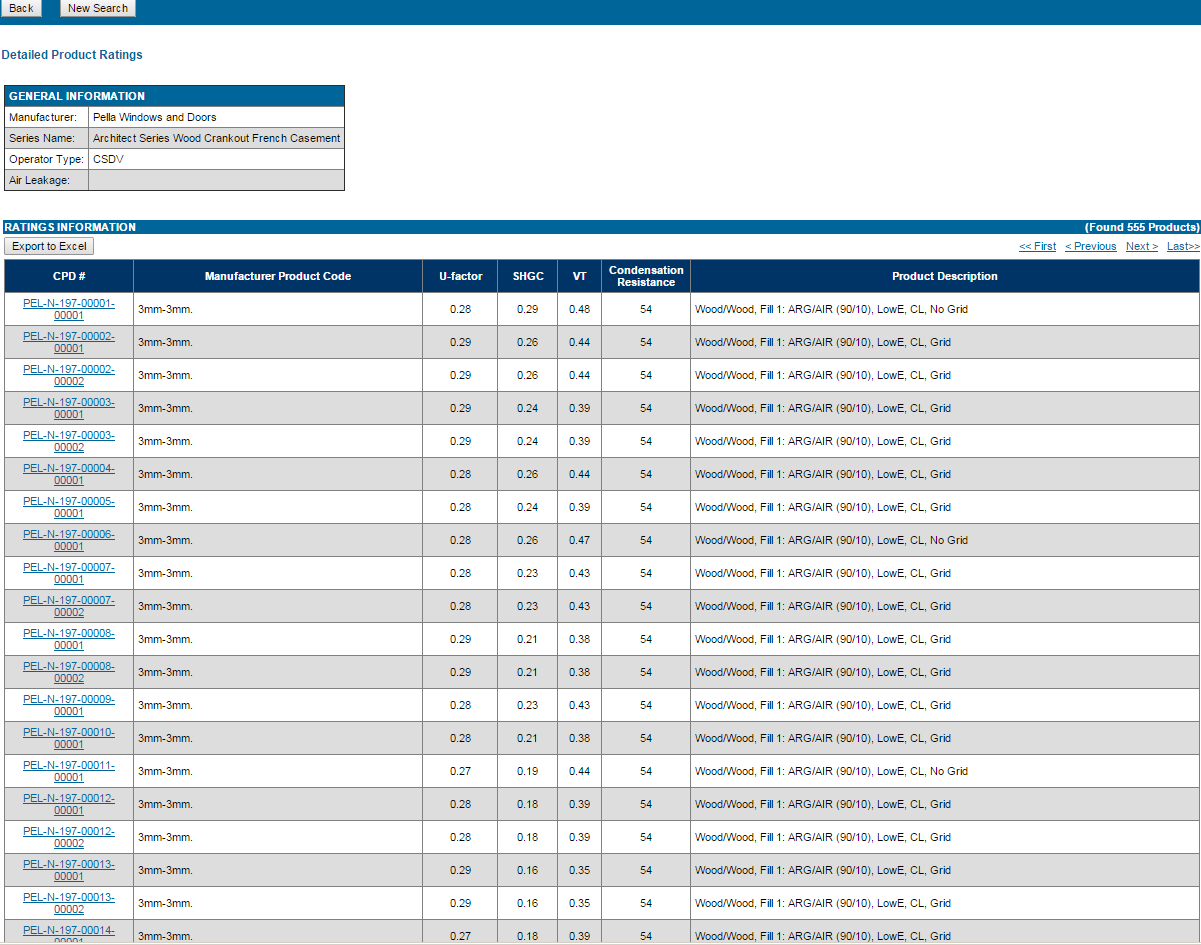








# Appendix 6: Search results screenshot



# 