

Bachelor's thesis

Degree programme: Information Technology

Specialisation: Network Communication Technology

2011

YIMENG NIU

QUOTATION INFORMATION AUTOMATION SOLUTION



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

BACHELOR'S THESIS | ABSTRACT
TURKU UNIVERSITY OF APPLIED SCIENCES

Degree programme : Information Technology

Completion of the thesis: 30.4.2011

Total number of pages:45

Instructor: Patric Granholm

Author: Niu Yimeng

QUOTATION INFORMATION AUTOMATION SOLUTION

This thesis deals with a quotation information automation solution for Reka Kumi Oy.

In Reka Kumi Oy, the information needed for generating quotation sheet is spread in several different sheets in an Excel template file. Every time quotation sheet is generated, the sales department needs to fill in those sheets in the template, then gather the information needed for the quotation from different sheets, copy them to another file and then print it as the quotation sheet.

The company needed a more convenient way of generating quotation sheets and accessing quotation information.

The solution includes the following steps:

1. Making a 'Cover' page for the excel template. Collecting and calculating the information needed for generating quotation sheet into this 'Cover' page.
2. Creating a database to store the information of the 'Cover' page.
3. Using VBA code to transfer information from the Excel template to database.
4. Using C# to build a website to display and modify the quotation information.

This project is going to significantly reduce the time and labour consume by generating quotation sheet.

KEYWORDS: Information Technology, C#, Excel, VBA, Database, MySQL

Deposited at the Library of Turku University of Applied Sciences

Notation

Excel	Microsoft Excel
C#	Microsoft Visual C#
IIS	Microsoft Internet Information Services
VBA	Microsoft Visual Basic for Applications
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
MySQL	Opensource Structured Query Language
Control	A graphical user interface object which allows users to control the program.
DLL	Dynamic Link Library

Table of Contents

Abstract

Notation

Contents

1. Introduction.....	1
2. Project Overview.....	2
2.1 Background.....	2
2.2 Objectives and Solution of the Project.....	3
3. Implementing Platform and Server Environment	7
3.1 Operating System.....	7
3.2 Internet Information Services (IIS).....	7
3.3 Database.....	8
4. Development Tools and Technologies	9
5. Database Design	13
5.1 Designing Principles.....	13
5.2 Database Topology and Structure.....	15
6. Development Process.....	18
6.1 Microsoft Excel.....	18
6.2 Microsoft Visual Basic for Application.....	21
6.2.1 Change Detection.....	21
6.2.2 Connection String.....	23
6.2.3 Trim Function.....	23
6.2.4 Update to Database Function.....	24
6.2.5 Sheet Hiding and Displaying.....	25
6.3 C# ASP.NET.....	27
6.3.1 Default.aspx.cs Code Structure.....	30
6.3.2 Find.aspx.cs Code Structure.....	32
6.3.3 Result.aspx.cs Code Structure.....	34

6.3.4 GridView Control Function.....	36
7. Conclusion.....	39
References	

1. Introduction

Reka Kumi Co.,Ltd needs a solution to improve the efficiency of their current quotation generating process. Their needs are special and unique, since the company is using many old versions of software and there is no ready-made software and database that meet their needs. Hence, the author was asked to perform this project in order to reduce the amount of manual work for the sales department and provide an easier way of transferring, observing and modifying the information needed for quotations.

The author has previously worked in Reka Kumi Co., Ltd as an intern. The author has been using Microsoft Visual C# to write programs for Reka concerning product management, ERP (Enterprise Resource Planning) assistant, data migration and database maintenance.

2. Project Overview

2.1 Background

Reka Kumi Oy is one of the leading rubber manufacturers in Finland. Reka Kumi Oy produces a variety of rubber products including, but not limited to, moulded rubber with injection moulding, rubber-metal products and silicone hoses. Reka supplies clients all around the world.

The current quotation information generating process in Reka is considerably complex. Reka is using Microsoft Excel 2000/2003 and M-Files System for collecting and storing quotation information. The pricing and quantity information have to be calculated one by one. In the Calculus template (MS Excel template which stores in M-Files System, calculus document stores all the information needed for the quotation. Once the user creates a new Calculus document in the M-Files System by using the template, the basic information of product will automatically appears in the document.), there are several sheets containing different prices, materials, drawing, quantity information and so on. When a customer places an order, the sales department fills in all the necessary information in the template, so the sales personnel has to do a lot of calculations about price and quantity. Finally, they write the result and other information into quotation sheet. This process is time and labour-consuming. Hence, the efficiency of order processing and quotation making drops significantly. (The whole process is illustrated in Figure 1)

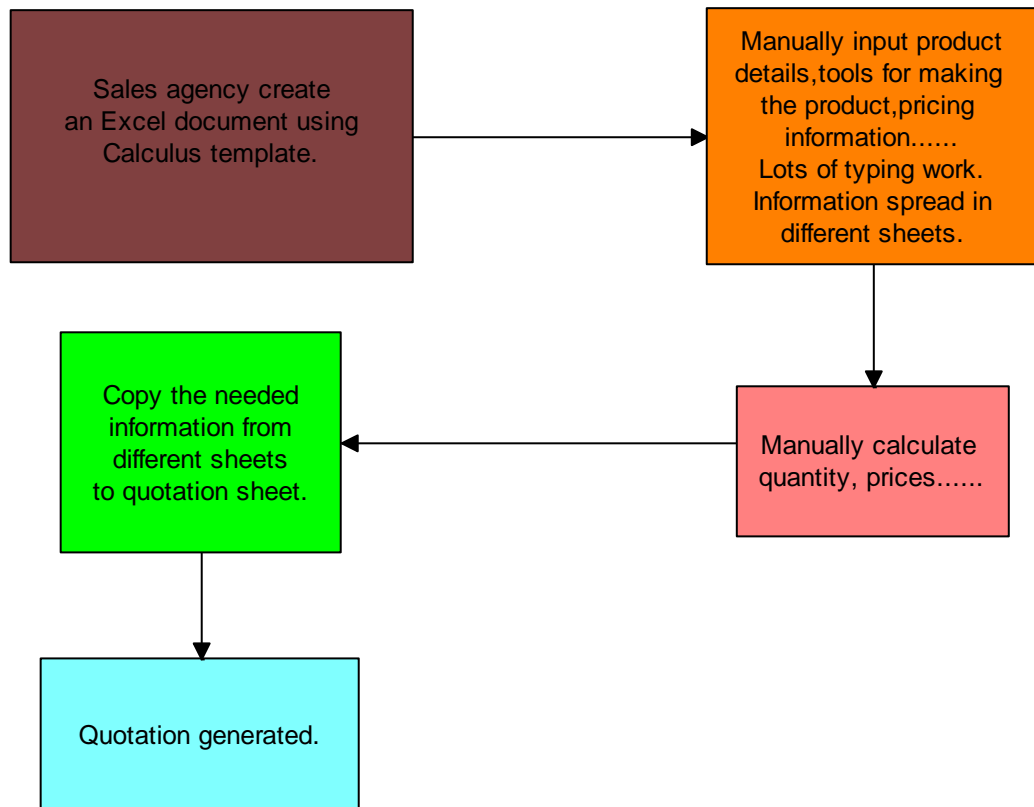


Figure 1. Current Quotation Generating Process

The author of this thesis was asked to find out a solution which speeds up the whole process and significantly reduces the labour consumption.

2.2 Objectives and Solution of the Project

After the author discussed and research with the IT Manager of Reka Kumi Oy and other related personnel, the author found out that the company needed an easier way of calculating, collecting, storing and transferring information necessary for quotations. The sales manager needed to access to the quotation information more efficiently and easily, in which case, a website which shows all the quotation information would be a good solution because it is easier to access a website than looking for documents (files) for user; and a website gives more visibility when looking for the collection of specific information.

The author considers that the most time/labour consuming part is inputting, calculating and transferring information. It could be more efficient if the computer handles those procedures. Since the information inputting cannot be avoided, more attention could be paid to the calculation and information transfer . So, it is necessary to have an automation to calculate, collect and transfer information. The automation should definitely reduce the time and labour consumption.

Based on above discussion and considerations, the solution is given as follows (Also see Figure 2):

1. Add a Cover sheet that contains all the information of the client, tooling, and products properties; calculate the price in the Calculus template.
2. Build a database for storing the quotation information.
3. Create a Macro by using Microsoft Visual Basic for Application (VBA) to transfer the quotation information to database.
4. Make a Microsoft C Sharp (C#) Web Application which queries the database and displays the returned information.

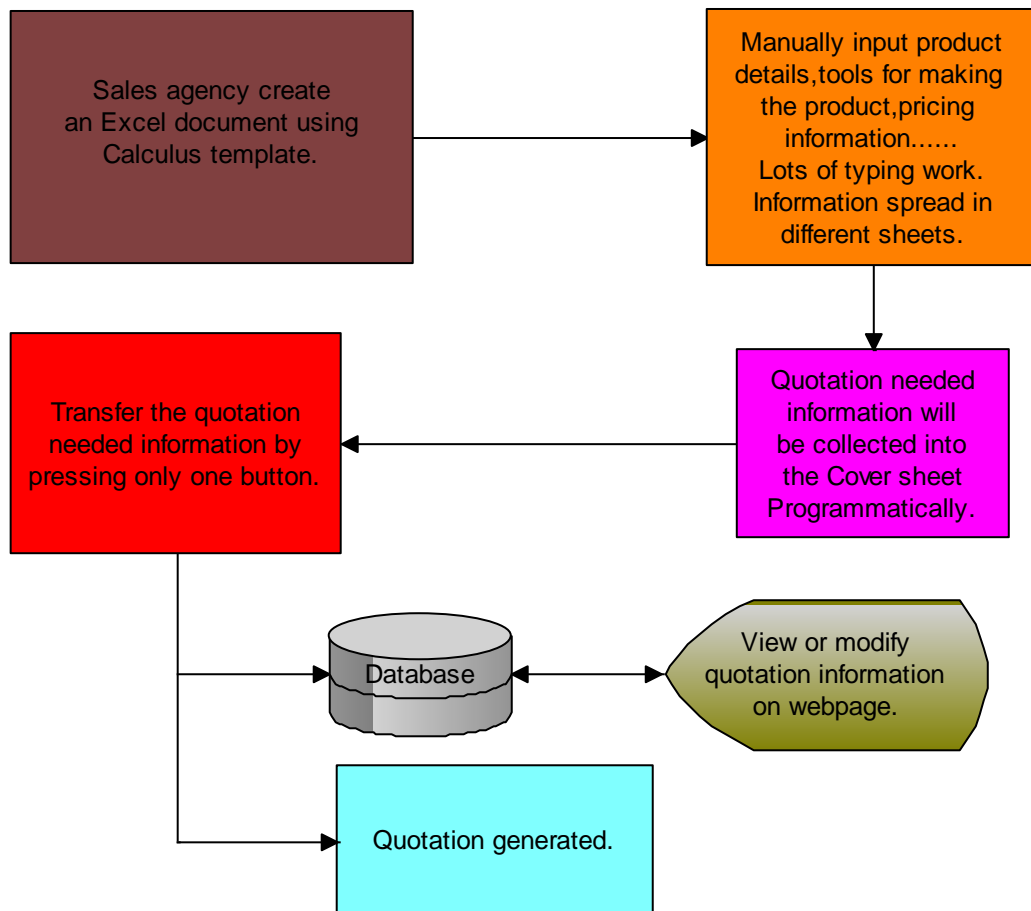


Figure 2. Evolved Quotation Generating Process

The reason of choosing VBA to handle the data transfer is VBA's high integration with MS Excel. The user could easily change the code and run a VBA macro in any computer which equipped with MS Excel 2003. Unlike other programming languages, VBA is a relatively simple programming language. It is easy to learn and implement. It comes with MS Excel automatically which means that the user does not need to install VBA separately in order to run VBA macros.

The disadvantage of using VBA macro to handle the data transfer is that Microsoft's products are not normally backward compatible. For instance, in most cases software which is designed for Windows 95 cannot run on Windows 2000. This means there is risk that the VBA macro for MS Excel 2003 designed by author may not able to run it on later version of MS Excel. The only way to avoid this is not using Microsoft's products but the Linux kernel operating system.

Apparently, switching to Linux is unrealistic currently for the company. The same applies to using other programming languages, all of which have the risk that the old code is not compatible with a newer version of Microsoft products.

According to above discussion, VBA could be the best tool to handle the data transferring part.

The reason of choosing Microsoft C# as the programming language to make Web Application is that the author has some experience in it. It would be easier for the author to perform this program with Microsoft C# than Java. When considering that .NET Framework is pre-installed in company computers, Microsoft C# application can be run directly. Java Runtime has to be installed separately if for running Java application.

3. Implementing Platform and Server Environment

This project was implemented based on the following platform and environment:

- Microsoft Windows 7 Ultimate 32 Bit (For development machine)
- Microsoft Windows 2008 Server R2 Enterprise 64 Bit (For LOTTA server)
- Microsoft Internet Information Services (For LOTTA server)
- MySQL Server (For LOTTA server)

3.1 Operating System

The author was using Microsoft Windows 7 Ultimate 32 Bit version as the operating system for the development machine.

Windows 7 is the current latest version of Windows operating system of Microsoft. It is widely used in personal computers (PC) including home and business use for laptop, tablet and media centre.

The server uses Microsoft Windows 2008 Server R2 Enterprise 64 Bit as operating system.

3.2 Internet Information Services (IIS)

The Internet Information Services – is a web server application and set of feature extension modules created by Microsoft for use with Microsoft Windows. It is the most used web server after the Apache HTTP Server: As of January 2011, it served 21.00% of all websites on the Internet and 16.22% of the one million

busiest websites on the Internet. IIS 7.5 supports HTTP, HTTPS, FTP, FTPS, SMTP and NNTP.[1]

IIS 7.5 is currently uses on the Reka server to provide information services.

3.3 Database

In this project, the author used MySQL as database. “MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases.”[2] More detailed information about database will be discussed in Chapter 5.

4. Development Tools and Technologies

This chapter introduces all the involved software and technologies implemented in this project.

Microsoft Visual C#

“C# (pronounced "C sharp") is a programming language that is designed for building a variety of applications that run on the .NET Framework. C# is simple, powerful, type-safe, and object-oriented. The many innovations in C# enable rapid application development while retaining the expressiveness and elegance of C-style languages. Visual C# is an implementation of the C# language by Microsoft. Visual Studio supports Visual C# with a full-featured code editor, compiler, project templates, designers, code wizards, a powerful and easy-to-use debugger, and other tools. The .NET Framework class library provides access to many operating system services and other useful, well-designed classes that speed up the development cycle significantly.”[3]

The author has previously some experience with MS C#. This is one of the reasons why MS C# plays an important role in this project.

Microsoft Visual Basic for Application

“Visual Basic for Applications (VBA) is an implementation of Microsoft's event-driven programming language Visual Basic 6 and its associated integrated development environment (IDE), which is built into most Microsoft Office applications. VBA enables programmers to extend the functions of Windows Application, especially for Microsoft Office software. Developers are able to use

it to build user defined functions, automate processes and access Win32 and other low-level functionality through DLLs. It supersedes and expands on the capabilities of earlier application-specific macro programming languages such as Word's WordBasic. It can be used to control many aspects of the host application, including manipulating user interface features, such as menus and toolbars, and working with custom user forms or dialog boxes.”[4]

In this project, VBA is used to collect data in Microsoft Excel and transfer data to the MySQL database.

Microsoft .NET Framework 4.0

“The .NET Framework (pronounced dot net) is a software framework for Microsoft Windows operating systems. It includes a large library, and it supports several programming languages which allow language interoperability (each language can use code written in other languages). The .NET library is available to all the programming languages that .NET supports.”[5]

Version 4.0 is currently the latest version for .NET Framework. It needs to be installed both in the development computer and the server in order to perform C# applications and VBA code. However, users can work with older versions of .NET Framework since the VBA for MS Excel 2003 requires at least the 1.0 version.

Microsoft ASP.NET 2.0

“Microsoft® ASP.NET is a unified Web development model. It includes various necessary services for generating enterprise level applications. ASP.NET is provided as a part of the .NET Framework. When programming with ASP.NET, the developers are able to access the classes in the .NET Framework. ASP.NET

pages execute on the server and generate markup such as HTML, WML or XML that is sent to a desktop or mobile browser. It was first released in January 2002 with version 1.0 of the .NET Framework, and is the successor to Microsoft's Active Server Pages (ASP) technology. ASP.NET is built on the Common Language Runtime (CLR), allowing programmers to write ASP.NET code using any supported .NET language.”[6]

The ASP.NET library is packaged into 'System.Web.dll'. It provides webpage handling, expansion and the communication of application of HTTP Tunnel. ASP.NET is a development platform other than a programming language.

Microsoft ODBC

“In computing, Open Database Connectivity (ODBC) provides a standard software interface for accessing database management systems (DBMS). The designers of ODBC aimed to make it independent of programming languages, database systems, and operating systems. Thus, any application can use ODBC to query data from a database, regardless of the platform it is on or the DBMS it uses. This is accomplished by using an ODBC driver as a translation layer between the application and the DBMS. The application thus only needs to know ODBC syntax, and the driver can then pass the query to the DBMS in its native format, returning the data in a format the application can understand.”[7]

In this project, an ODBC connector was used to handle the data transfer between MS Excel and MySQL. It builds up the connection between each other and send MySQL query.

Microsoft Excel

“Microsoft Excel is a commercial spreadsheet application written and distributed

by Microsoft for Microsoft Windows and Mac OS X. It features calculation, graphing tools, pivot tables and a macro programming language called Visual Basic for Applications. It has been widely applied spreadsheet for these platforms, especially since version 5 in 1993. Excel forms part of Microsoft Office. The current versions are 2010 for Windows and 2011 for Mac.”[8]

In Reka Kumi Co., Ltd, the sales departments use ready-made Excel templates to input and calculate quotation information.

HTML

“HTML, which stands for HyperText Markup Language, is the predominant markup language for web pages. HTML is the basic building-block of webpages.”[9] It is used for information structuring for example, title, paragraph and list. It also, to some extent, describes the appearance and semantics of documents.

M-Files System

“M-Files System is an easy to use solution that helps small to medium-size businesses make dramatic gains in efficiency and productivity by improving the way they organize and manage their business documents, information and processes.”[10]

5. Database Design

5.1 Designing Principle

Principles of Table Designing:

Standardization

The standardization of data helps to eliminate the redundancy of database. There are few different kinds of standardization. However, the Third Normal Form (3NF) is usually recognized as the best for balancing the performance, expansibility and integrity of data. In other words, the principle for designing a database table which obeys the 3NF standard is 'One Fact In One Place'. A table should only include the basic attributes of itself. The table needs to be decomposed when it contains the attributes which do not belong to it. Foreign Key connects and represents the relation between tables.

Data Driver

Data Driver provides a more convenient way for maintenance and the change of designing strategy. It enhances the flexibility and expansibility of the whole system. For instance, when the user interface carries out work flow tasks (Sending E-mail, Printing letter, Modifying records), the data that generated by the tasks can be stored into database. If the process is data-driven, the user carries more responsibilities to maintain the work flow process.

Considering the Possible Changes

When designing a database, the designer must consider what data fields could possibly change in the future. For example, the surname for woman could change after she gets married. So, it is better to create a field to store the surname in the table. Additionally, the designer creates starting date and ending

date fields, in order to track the change of this data entry.

Principles of Field Designing

A matured database table shall contain the following 3 useful fields:

RecordCreationDate

RecordCreator

RecordVersion

The above 3 fields help to precisely locate the reason of missing data or null record. Hence, it is important to have those fields in significant tables.

Implement multiple fields for address and phone number data

Normally it is not enough to describe an address within one line. It is necessary to separate an address into street address, house number, city, province, country and postcode fields; A phone number can be divided into area code and phone number fields. This provides better flexibility and visibility.

Apply role object to define a field belongs to a category

When defining an object which belongs to a specific category or has a specific role, it applies a role object for a specified time relation in order to achieve the self-documentation. For example, George Brown, Accountant was promoted to become George Brown, Financial Manager and finally he became George Brown, CEO. The database designer can describe this change with the 'Person' and 'Person_Type' objects. The database designer only needs to change the value of 'Person_Type' which related to George Brown, then have a Date/Time field to record when it happened. The 'Person_Type' field contains all the possible occupations (Accountant, Financial Manager, Engineer, Director, CEO and so on).

5.2 Database Topology and Structure

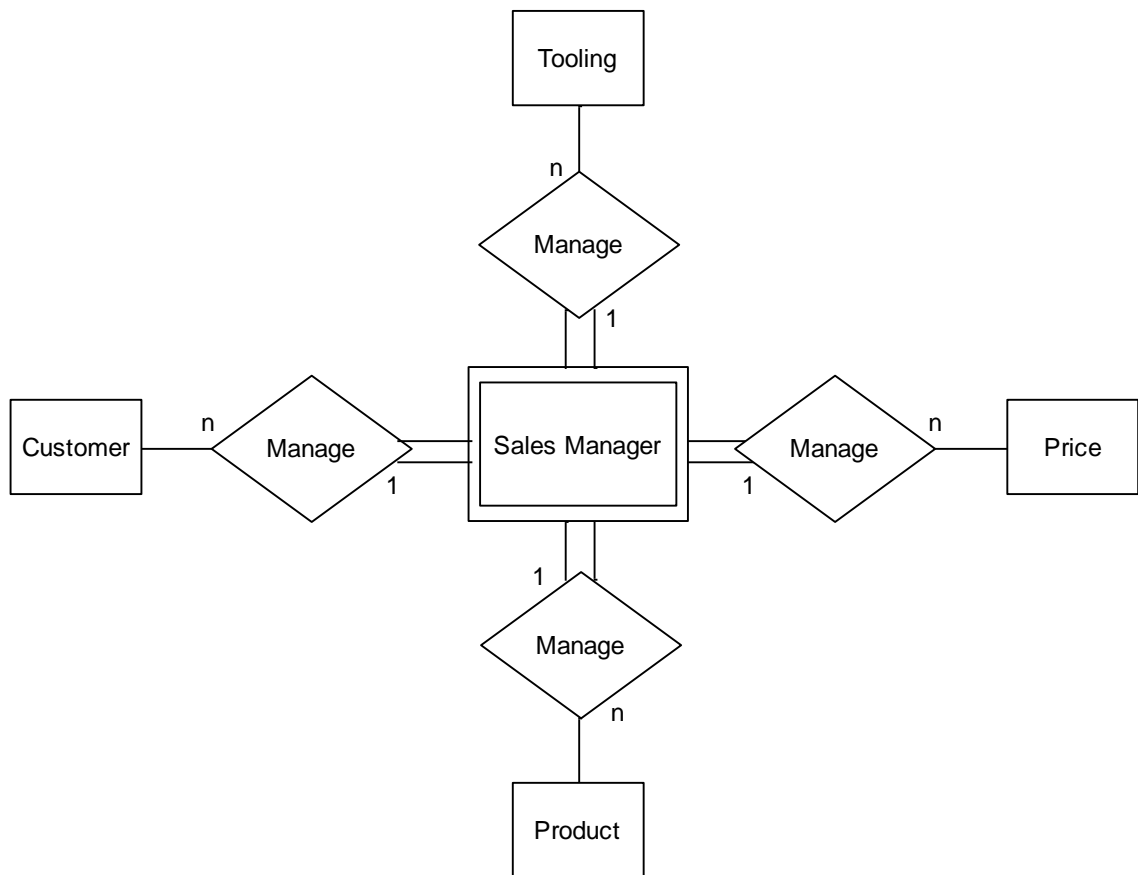


Figure 3. Database Entity-Relationship Model

There are totally five tables in the database. The above graph shows the relation between each table. Each table has 'M-Files ID' field marked as primary key and foreign key. Because a customer could have different orders with different products, a sales manager could have different customers, assigning the Unique 'M-Files ID' as primary key and foreign key prevents duplicated entries.

Field Name	Data Type	Description
M-Files ID	INT(10)	M-Files ID
salesmanager	VARCHAR(45)	Name of sales manager

Figure 4. Sales Manager Table Structure

Field Name	Data Type	Description
M-Files ID	INT(10)	M-Files ID
Customer	VARCHAR(45)	Customer Name
Customerno.	VARCHAR(45)	Customer Number
Customergroup	VARCHAR(45)	Customer Group
CuslteNam	VARCHAR(45)	Customer Item Name
CuslteNum	VARCHAR(45)	Customer Item Number
CuslteDra	VARCHAR(45)	Customer Item Drawing
RCD	DATETIME	Record Creation Date
RC	VARCHAR(45)	Record Creator
RV	VARCHAR(45)	Record Version
OS	VARCHAR(45)	Order Status

Figure 5. Customer Table Structure

Field Name	Data Type	Description
M-Files ID	INT(10)	M-Files ID
Tooldes	VARCHAR(45)	Tool Description
Cavityno.	VARCHAR(45)	Cavity Number
NOT	VARCHAR(45)	Number of Tools
Toolp	VARCHAR(45)	Tool Price
TTC	VARCHAR(45)	Total Tooling Cost
RCD	DATETIME	Record Creation Date
RC	VARCHAR(45)	Record Creator
RV	VARCHAR(45)	Record Version

Figure 6. Tooling Table Structure

Field Name	Data Type	Description
M-Files ID	INT(10)	M-Files ID

RE	VARCHAR(500)	Rubber Elastomer
Hardness	VARCHAR(45)	The Hardness of Rubber Part
Reinforcement	VARCHAR(500)	Reinforcement
WT	VARCHAR(45)	Wall Thickness
PV	VARCHAR(500)	Product Review
Tolpuls	VARCHAR(45)	Tolerance (+)
Tolminus	VARCHAR(45)	Tolerance (-)
RCD	DATETIME	Record Creation Date
RC	VARCHAR(45)	Record Creator
RV	VARCHAR(45)	Record Version
Tyyppi	VARCHAR(45)	Product Type
Origin	VARCHAR(45)	Location of the Manufacture
Deviations	VARCHAR(45)	Product Deviations

Figure 7. Product Construction Table Structure

Field Name	Data Type	Description
M-Files ID	INT(10)	M-Files ID
SP1AQ	VARCHAR(45)	Step Price 1 Annual Quantity
SP1MDQ	VARCHAR(45)	Step Price 1 Minimum Delivery Quantity
SP1PP	VARCHAR(45)	Step Price 1 Price/PCs
SP2AQ	VARCHAR(45)	Step Price 2 Annual Quantity
SP2MDQ	VARCHAR(45)	Step Price 2 Minimum Delivery Quantity
SP2PP	VARCHAR(45)	Step Price 2 Price/PCs
.....
SP6PP	VARCHAR(45)	Step Price 6 Price/PCs
YE	VARCHAR(45)	Yearly Euro
Proto	VARCHAR(45)	Proto Price

Figure 8. Pricing Information Table Structure

6. Development Process

This chapter introduces the software for developing this project and the development process. The following 4 software are involved:

- Microsoft Excel 2003
- Microsoft Visual Basic for Application
- Microsoft Visual Studio 2010 (C#)
- MySQL Server

MySQL Server will not be discussed separately in this chapter, since the process of MySQL is fully combined with other three programs.

6.1 Microsoft Excel

The Cover page includes customer information (See. Figure 9), tooling information (See. Figure 10), product construction information (See. Figure 11) and pricing information (See. Figure 12).

	A	B	C	D	E	F
1	Cover sheet for a quotation			M-Files ID	130001	
2				Date	0	
3	Sales Manager					
4						
5	Customer	0	0	Customer Group		
6						
7	Customer Item Name	0		Poland		
8						
9	Customer Item Number	0				
10						
11	Customer Item Drawing	0				
12	(+ issue)					
13						
14	Reka Product Number	0				

Figure 9. Customer Information

15	Tooling				
16	Tool Description				
17					
18	Cavity Numbers				
19					
20	Number of Tools				
21					
22	Tool Price				
23					
24	Total Tooling cost				

Figure 10. Tooling Information

25	Product construction				
26	Rubber Elastomer				
27					
28	Hardness		Sh		
29					
30	Reinforcement				
31					
32	Wall Thickness		mm	tol	(+) (-)
33					
34	Product review				
35					

Figure 11. Product Construction Information

36	Pricing	Yearly Euro			0
37					
38		Proto Price			
39					
40			Annual Quantity	Minimum Delivery Quantity	Price/PCs
41					
42		Step Price 1.	100	10	
43					
44		Step Price 2.	100	10	
45					
46		Step Price 3.	100	10	
47					
48		Step Price 4.	100	10	
49					
50		Step Price 5.	100	10	
51					
52		Step Price 6.	100	10	
53					

Figure 12. Pricing Information

	A	B	C	D
1	Cover sheet for a quotation			M-Files ID
2				Date
3	Sales Manager			
4				

Figure 13. Sales Manager Value by M-Files

	A	B	C	D	E
34	Product review				
35					
36	Pricing	Yearly Euro			0
37					

Figure 14. Yearly Euro Value by formula Calculation

Only few of the above information are obtained by user inputting. Most of the information can be generated by formula calculation and M-Files System. For instance, the 'Sales Manager' value is generated the by M-Files System; the 'Yearly Euro' value is calculated by formula. (See Figure 13 and 14)

As Figure 14 shows, Excel calculates 'Yearly Euro' by multiplying the value of C15 (MVX sheet) cell with the value of E42 (Cover sheet) cell. So, this means that every change the user made on other sheets might affect the Cover sheet.

When the user makes changes on any sheets, the tab color of the edited sheet and Cover sheet will change to red in order to remind the user to update the new data to database. Then, the user presses 'UPDATE TO OVERVIEW' button to process the update. After the changes have been stored into database, the tabs colors changes back to green.

There is a sheet named 'Feasibility' in this Excel template. The sales manager requires that the Feasibility sheet must be complete before the user makes any change to other pages. The IT Manager of Reka suggested that all other sheets should be hidden before the user complete 'Feasibility' sheet, to prevent the user from skipping the Feasibility sheet. The Product Manager thought that the 'Feasibility' sheet should contain an 'Emergency Button' which could skip to fill in this sheet when necessary, the tab color of 'Feasibility' sheet becomes pink which reminds the user about the incomplete sheet.

6.2 Microsoft Visual Basic for Application (VBA)

6.2.1 Changes Detection

In this section, the sheet change detection is separate into two parts:

- Change detection for Cover sheet
- Change detection for other sheets

The reason why the changes detection is divided into two parts is that when user makes changes on other sheets, it does not necessarily affect the Cover sheet. In other words, the user makes changes on other sheets and the changes do not affect the quotation information. Only the tab color of the modified sheets changes to red. When the changes affect the Cover sheet, the tab color of Cover sheet will change to red.

The following example code is for detecting changes in the Cover sheet. If any change has been detected, the color of the 'UPDATE TO OVERVIEW' button and the Cover sheet tab color change to red.

Program 1. Cover Sheet Change Detection

```
Private Sub Worksheet_Change(ByVal Target As Range)

    If Not Intersect(Target, & _
        Range("B3:B52,C3:C52,D42:D52,E1:E52,F32")) Is Nothing Then

        On Error Resume Next

        Application.EnableEvents = False

        CommandButton1.BackColor = &HFF&

        Sheet1.Tab.ColorIndex = 3

        Application.EnableEvents = True

        On Error GoTo 0

    End If

End Sub
```

The change detection for other sheets checks if there are any changes in the sheets other than the Cover sheet. If a change is detected, the button and tab color will change the same way as in the Cover sheet change detection. (See Program 1 and Program 2)

Program 2. Other Sheets Change Detection

```
Private Sub Workbook_SheetChange(ByVal Sh As Object, ByVal Target
As Range)

    Static AncAdress As String, AncCell As Variant

    AncAdress = Target.Address

    AncCell = Target.Value2

    If AncAdress <> "" Then

        ActiveSheet.Tab.ColorIndex = 3

        Module1.sheetname = ActiveSheet.Name

        i = Module1.sheetname

    End If

End Sub
```

```

        Sheets(i).Tab.ColorIndex = 3

        Sheet1.CommandButton1.BackColor = &HFF&

        Sheet1.Tab.ColorIndex = 3

    End If

End Sub

```

6.2.2 Connection String

A connection string is used for building up the connection between Excel and the database. In this project, the company uses the MySQL database. The connection adapter is ADODB. Here is an example of the connection string below.

In this example, the server name is "lilli". The company has a DNS server which translates the domain name "lilli" into an IP address.

Program 3. Connection String

```

Private Sub ConnectDB()

    Set oConn = New ADODB.Connection

    oConn.Open "DRIVER={MySQL ODBC 5.1 Driver};" & "SERVER=lilli;" & _
        "DATABASE=quotation;" & "USER=käyttäjä;" & "PASSWORD=passi;" & _
        "Option=3"

End Sub

```

6.2.3 Trim Function

In Visual Basic for Application, the apostrophe symbol is used as comment which means that the code written after "'" symbol (including "'" symbol) will not be compiled and executed.

In practical writing, the apostrophe is a punctuation mark, and sometimes a diacritic mark, in languages that use the Latin alphabet or certain other alphabets. [11] In many cases, it is not possible to avoid using this mark. This is why the Trim Function was created. (See Program 4)

Program 4. Trim Function

```
Function esc(txt As String)
    esc = Trim(Replace(txt, "'", "\"'))
End Function
```

In VBA and many other programming languages, the \ mark is added to those marks which mean different functions between programming language and practical writing. Hence, ambiguity can be avoided.

6.2.4 Update to Database Function

This function builds up the connection between Excel and the MySQL database. It also handles the tab color changes. Here is a simple example of inserting information into the MySQL database. (See Program 5)

Program 5. Update to Database Function

```
Private Sub InsertData()
    Set rs = New ADODB.Recordset
    ConnectDB
    With Sheet1
        rowCursor = 3
        strSQL = "REPLACE INTO calculus SET id='" & esc(.Cells(1, 5))
        "', customernumber='" & esc(.Cells(rowCursor + 2, 3)) & "'"
        rs.Open strSQL, oConn, adOpenDynamic, adLockOptimistic
    End With
    CommandButton1.BackColor = &HE0E0E0
    Sheet1.Tab.ColorIndex = 4
End Sub
```

This function builds up a new ADO DB Recordset. It is an object which collects a set of record from database table. The 'ConnectDB' function is used to establish the connection (See 7.2.2). The 'strSQL' string contains the query which is going to be executed by MySQL. In MySQL, there are several different ways of inserting or modifying data. For Example:

- INSERT ... SET...
- INSERT ... VALUES ...
- REPLACE INTO ...

The INSERT ... SET... and INSERT ... VALUES ... functions always insert data without deleting old records. In this case, another function which modifies data is needed.

Hence, the author prefers to use the REPLACE INTO function, because it inserts data when the target record does not exist and it modifies and updates data when the target record exists, so that it does not need to use other functions to check the existence of target record.

Finally, if the MySQL query can be executed successfully which means the information has been stored into database, the tab color and button color will change to green which indicates the process was executed successfully.

6.2.5 Sheet Hiding and Displaying Function

When user creates a new document by using this template, all the sheets except 'Feasibility' sheet should be hidden by default. In 'Feasibility' sheet, the user is expected to fill in the feasibility study information. The trigger to show other sheets is the feasibility decision selection box.

Here are those four decision selection boxes which can be selected by the user.

- Feasible

- Feasible With Approved Review
- Feasibility Study Not Done Yet
- Not Feasible

If the last option 'Not Feasible' is selected, those hidden sheets will not show up. This indicates that the order cannot be performed due to some reasons.

The third option 'Feasibility Study Not Done Yet' is the emergency button which is requested by the Product Manager. If this option is selected, all sheets will show up. Then, the tab color of this sheet remains pink.

The above four options have CheckBoxes which are linked to the cells. Once a CheckBox is checked, it translates the CheckBox value from Boolean to logic text. The text it displays depends on the language version of Excel. In the English Excel version, the cell shows 'True' or 'False'.

The following code shows an example of how it judges to show or hide sheets. (See Program 6)

Program 6. Sheet Hiding and Displaying

```
Sub CheckBox28_Click()
    If Sheet3.Range("E39").Value = True Then
        MsgBox "Error. Please first uncheck 'Feasibility study not done yet'"
        Sheet3.Range("E37").Value = False
    Else
        If Sheet3.Range("E37").Value = True Then
            showpage
        ElseIf
            Sheet3.Range("E37").Value=False And
            Sheet3.Range("E36").Value = False Then
                hidepage
        End If
    End If
End Sub
```

If the user clicks on 'Feasible' or 'Feasible With Approved Review' button, Excel will check whether the 'Feasibility Study Not Done Yet' has been forgotten to uncheck. The user gets a pop-up window that states the problem if there was one.

6.3 Microsoft Visual Studio 2010 (C#)

In this chapter, the author focuses on the key functions of this website. Only part of the necessary codes is presented here.

With C#, it is easy to build a multi-functional website. It provides direct access to the database and GridView to display the result that is returned by the database.

The webpage provides the following fetching method (See Figure 15. Main Interface):

- By Item Number
- By Customer Name
- By Customer Number
- By Customer Group
- By Sales Manager

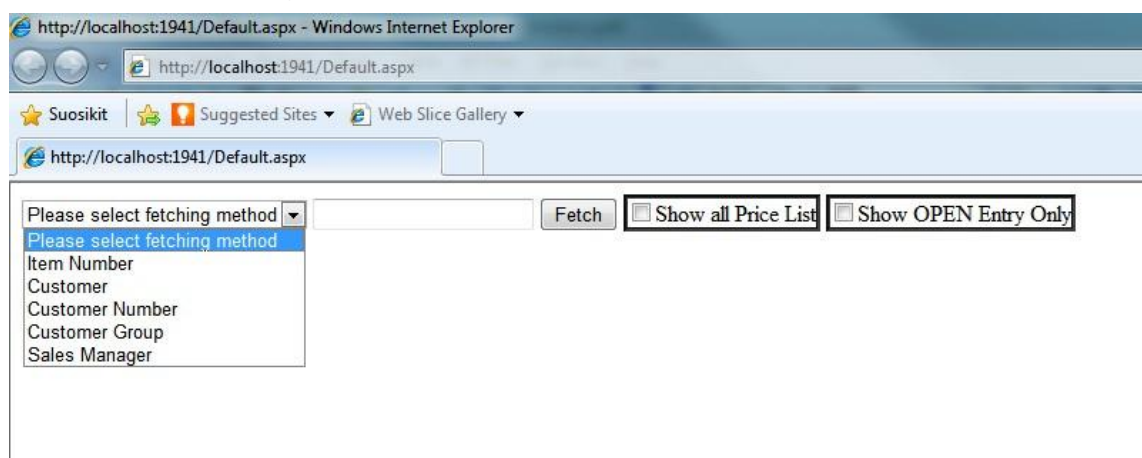


Figure 15. Main Interface

After selecting the fetching method (Ex. Customer), the user shall give the keyword (Ex. Customer Name) in the TextBox. The following two types of keywords are acceptable:

- Full name/number
- Part of the name/number (any character which consist of the name or number)

If the full name/number has been given, the browser will direct to a specific page and display the result.

If part of the name/number has been given, the browser will direct to a page named 'Find.aspx'. This page shows a list of name/number which includes the keywords. Here, the user can select the one he or she wants. (See Figure 16. Fuzzy Inquiry)

If the CheckBox 'Show all Price List' is checked, the full price list will be displayed in the result. There are totally six prices fields in the database. Normally, the result contains one price per id only and other information if this CheckBox is not checked.

All the quotation information has three statuses:

- OPEN
- WIN
- LOST

If the user wants to see the entries with 'OPEN' status only, the CheckBox named 'Show OPEN Entry Only' should be checked. Otherwise, it shows all entries by default.

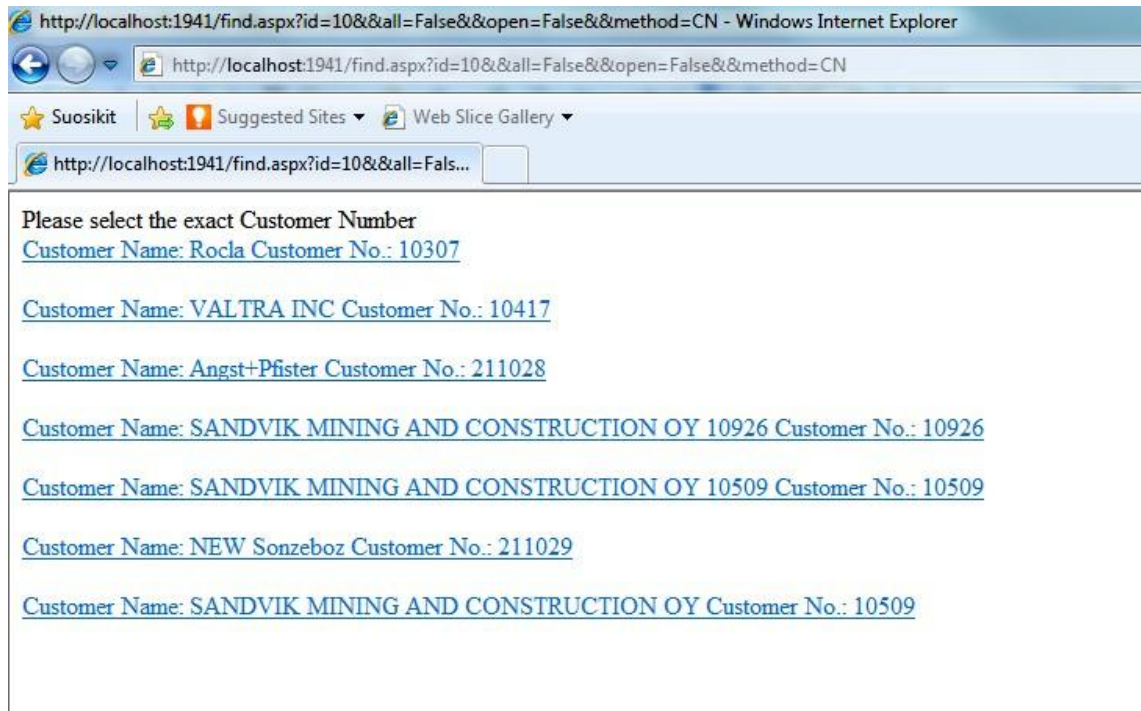


Figure 16. Fuzzy Inquiry

After the user has selected the desired entry, the page shows the result.

Here is a flow chart describing the whole process (See Figure 17. Process Flow).

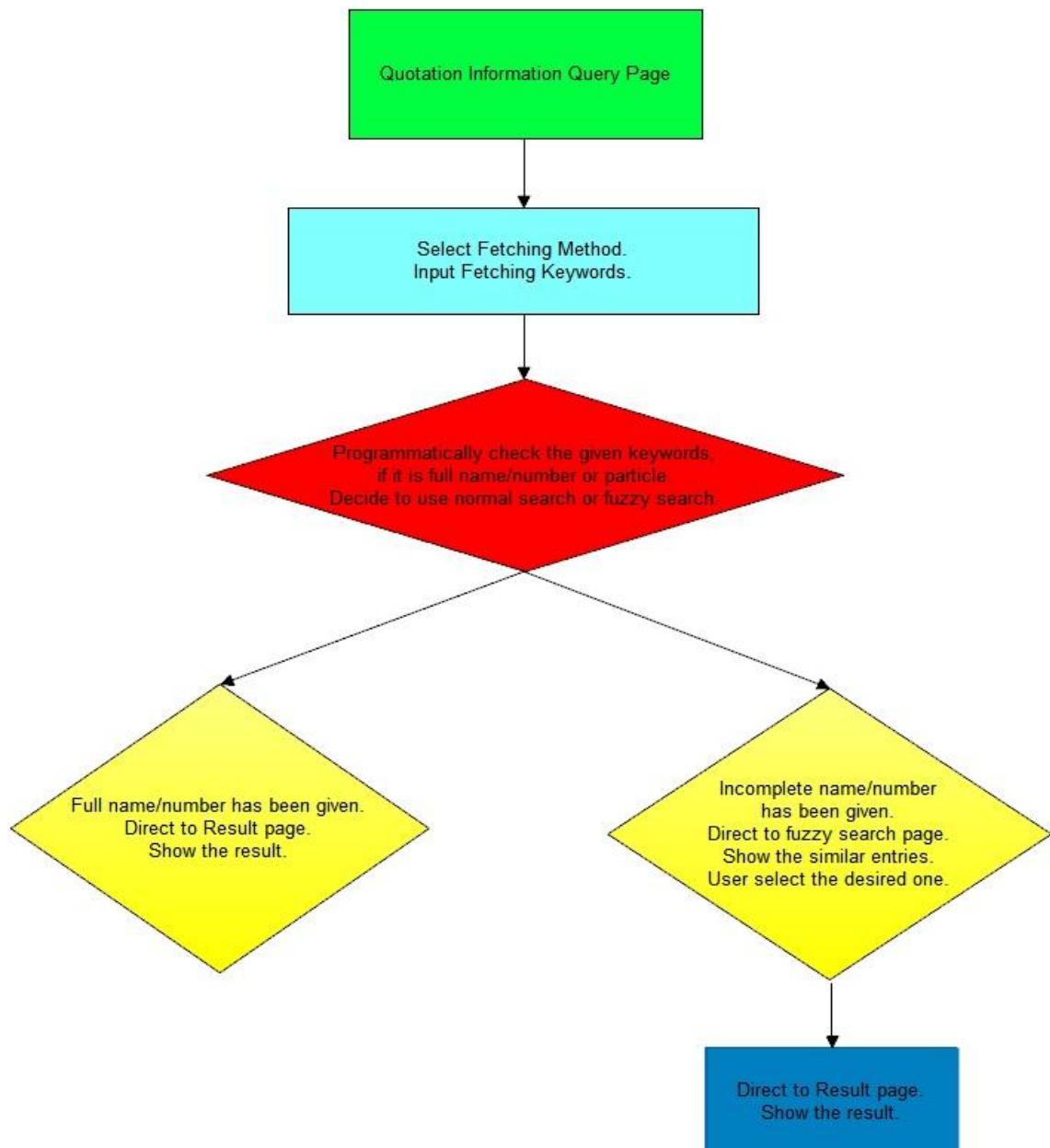


Figure 17. Process Flow

6.3.1 Default.aspx.cs Code Structure

Default.aspx is the homepage of this website. In this page, the user selects the fetching method and input the keyword. After TextBox is filled in with the keyword, the user shall press the 'Enter' key or click the 'Fetch' button to continue

Program 7. The 'Fetch' Button Method Example

```
if (DropDownList1.SelectedItem.ToString() == "Customer Number")
{
    string para = "CN";
    checkparameter(para);
}
```

(See Program 7) In the 'Fetch' button method, this function checks the selected item name in the DropDownList and passes it as a parameter to the next method which is called 'checkparameter' (See Program 8).

Program 8. The 'Checkparameter' Method Example

```
public void checkparameter(string para)
{
    if (para == "CN")
    {
        string columnname="customernumber";
        string prequery = "SELECT distinct
customer,customernumber FROM quotation.calculus where customernumber
like '%" + TextBox1.Text + "%'";
        string strProvider = "Data Source= lilli ;
Database=quotation; User ID=xxx; Password=passi";
        MySqlConnection mysqlCon = new
MySqlConnection(strProvider);
        mysqlCon.Open();
        MySqlDataAdapter customerListAdapter = new
MySqlDataAdapter();
        customerListAdapter = new MySqlDataAdapter(prequery,
mysqlCon);

        DataSet DS = new DataSet();
        customerListAdapter.Fill(DS);
        mysqlCon.Close();
        if (DS.Tables[0].Rows.Count>1)
        {
            string url;
            url = "find.aspx?id=" + TextBox1.Text.ToString() +
"&&all=" + CheckBox1.Checked.ToString() + "&&open=" +
CheckBox2.Checked.ToString() + "&&method=CN";
            Response.Redirect(url);
        }
    }
}
```

```

else
{
    string url;
    url = "Result.aspx?id=" +
DS.Tables[0].Rows[0][columnname].ToString() + "&&all=" +
CheckBox1.Checked.ToString() + "&&open=" +
CheckBox2.Checked.ToString() + "&&method=CN";
    Response.Redirect(url);
}
.....
}

```

The 'Checkparameter' method accepts the parameter passed from the 'Fetch' button. In this function, it uses 'If' clause to check the parameter. Then, it decides which database query string to use. After querying the database, results are returned and stored into a DataSet named DS.

If the user has given a correct and complete keyword, there will be only one row of database query result, since the 'DISTINCT' function is used in the query. The browser will be directed to the 'Result.aspx' page which shows the quotation information.

Alternatively, if an incomplete keyword has been given in the query string, there could be more than one row for the result. The browser will be directed to the 'Find.aspx' page with passing the keyword as parameter in order to find out the correct number or name.

6.3.2 Find.aspx.cs Code Structure

This page refers to the Fuzzy Inquire function which aims to find all similar and related entries from the database based on the passed keyword.

In the 'Page_Load' function which loads methods when entering this page, the 'FindSimilarEntries' method is loaded (See Program 9). The loaded method

checks the parameter which is passed from the 'Default.aspx' page. Then it decides which query string is going to be used. This method basically uses the same way as 'Checkparameter' in 'Default.aspx' when it looks for the correct keywords.

Program 9. The 'FindSimilarEntries' Method Example

```
public void FindSimilarEntries()
{
    string prequery="";
    if(Request.QueryString["method"].ToString()=="CN")
    {
        .....
        int resultcount = DS.Tables[0].Rows.Count;
        for (int i = 0; i < resultcount; i++)
        {
            linkbutton(DS.Tables[0].Rows[i]["customer"].ToString(),
            DS.Tables[0].Rows[i]["customernumber"].ToString(),
            Request.QueryString["method"].ToString());
        }
        .....
    }
}
```

After a list of similar keywords is returned from database, this function checks the amount of results and assigns to the amount an integer variable named 'resultcount'. Another method named 'LinkButton' will be called certain times depending on the amount of results.

The 'LinkButton' method is used for generating LinkButtons programmatically. It generates a list of correct keywords for use to choose (See Program 10).

Program 10. The 'LinkButton' Generating Method Example

```
public void LinkButton(string name, string number, string method)
{
    form1.Controls.Add(new LiteralControl("<br>"));
    LinkButton lbtn = new LinkButton();
    if (method == "CN")
    {
        lbtn.Text = "Customer Name: " + name + " Customer No.: "
+ number;
        lbtn.CommandArgument = number;
    }
    if (number == "SM")
    {
        lbtn.Text = "Sales Manager : " + name;
        lbtn.CommandArgument = name;
    }
    .....
    lbtn.Command += new CommandEventHandler(this.lbtn_Click);
    form1.Controls.Add(lbtn);
    form1.Controls.Add(new LiteralControl("<br>"));
}
```

The parameters in the 'LinkButton' method, obtained from 'FindSimilarEntries' method, indicate the keyword type (Customer Name, Customer Number, Item Number, ...) and the keyword values. Finally, the program calls the 'LB_Event' method to add the Button_Click event to each LinkButton. Hence, those LinkButtons are assigned with unique HyperLinks towards to the 'Result.aspx' page in order to display the quotation information.

6.3.3 Result.aspx.cs Code Structure

There are two GridView Controls in this page. One is for displaying all the information, the other displays less information which is easier to see.

The main function of this page is the 'GetResult' method which is used for fetching quotation information (See Program 11).

Program 11. The 'GetResult' Method Example

```
public void GetResult()
{
    if (Request.QueryString["all"].ToString() == "True")
    {
        string query = "";
        string strProvider = "Data Source= lilli ;
Database=quotation; User ID=xxx; Password=passi";
        MySqlConnection mysqlCon = new
MySqlConnection(strProvider);
        mysqlCon.Open();
        if (Request.QueryString["method"].ToString() == "CN")
        {
            DropDownList1.SelectedIndex = 3;
            string asiakas =
Request.QueryString["id"].ToString();
            query = "SELECT ID,salesmanager as 'Sales
Manager',qnumber as 'Qnumber',Customer,
.....
'MDQ6',ROUND(REPLACE(stepprice6pp,',','.'),2) as 'PP6' FROM
`quotation`.`calculus` where customernumber='" + asiakas + "'";
            if (Request.QueryString["open"].ToString() ==
"True")
            {
                query = "SELECT ID,salesmanager as 'Sales
Manager',qnumber as 'Qnumber',
.....
FROM `quotation`.`calculus` where customernumber='" + asiakas + "' and
(status=' ' or status is null)";
            }
            .....
            for (int i = 0; i < DS.Tables[0].Rows.Count; i++)
            {
                string Str =
DS.Tables[0].Rows[i]["ToolPrice"].ToString().Trim();
                double Num;
                bool isNum = double.TryParse(Str, out Num);
                if (isNum)
                {
                    toolprice =
Convert.ToInt32(DS.Tables[0].Rows[i]["ToolPrice"].ToString()) +
toolprice;
                }
                else
                {
                    continue;
                }
            }
            .....
        }
    }
}
```



```

        Labell1.Text = "Total lines: " +
DS.Tables[0].Rows.Count + "          Total tool Price: " + toolprice + "
Total Yearly Euro: " + yearlyeuro;
        .....
    }

```

When a correct keyword is given or selected by user, the program starts to fetch quotation information by using the keyword. It loops the DataTable to eliminate non numerical characters for the total tool price and yearly euro.

In this page, the program uses two GridView Controls to display either the full price or the single price which depends on the user selection. If the user clicks on 'Show all Price List', it shows the full price and vice versa.

6.3.4 GridView Control Functions

A GridView displays the values of data source in a table where each column represents a field and each row represents a record. The GridView Control enables the user to select, sort and edit these items (See Figure18. GridView Control at a glance).

	File Link	Qnumber	Yearly Euro	Status	Type	Source	Sales Manager	Customer	Customer No.	Customer Item Number	Customer Item Name
Edit File	Databour	Databour	OPEN ▼	Moulded ▼	Finland ▼	Databound	Databound	Databound	Databound	Databound	Databound
Edit File	Databour	Databour	OPEN ▼	Moulded ▼	Finland ▼	Databound	Databound	Databound	Databound	Databound	Databound
Edit File	Databour	Databour	OPEN ▼	Moulded ▼	Finland ▼	Databound	Databound	Databound	Databound	Databound	Databound
Edit File	Databour	Databour	OPEN ▼	Moulded ▼	Finland ▼	Databound	Databound	Databound	Databound	Databound	Databound
Edit File	Databour	Databour	OPEN ▼	Moulded ▼	Finland ▼	Databound	Databound	Databound	Databound	Databound	Databound

Figure18. GridView Control at a glance

In this GridView Control, the author applied the following functions.

- Edit and Save functions are enabled for 'Qnumber', 'Yearly Euro', 'Status', 'Type' and 'Source' columns.
- Price List HyperLink
- File HyperLink

The above functions are currently fairly adequate for the sales department to enrich the information for single/multiple items.

In order to enable the edit and save function, it is necessary to add 'TemplateField' and 'ItemTemplate' to the aspx source code. The 'AutoGenerateColumns' function should be disabled since the amount of columns is fixed and data is bound to those columns (See Program 12).

Program 12. Edit and Save Function Example

```
<asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False"
    onrowcancelingedit="GridView1_RowCancelingEdit"
    onrowupdating="GridView1_RowUpdating"
onrowediting="GridView1_RowEditing">
    <Columns>
        <asp:TemplateField HeaderText="Qnumber">
            <ItemTemplate>
                <asp:TextBox ID="txtQnumber" runat="server"
                    Text = '<%#
DataBinder.Eval(Container.DataItem, "Qnumber") %>'
Width="60px"></asp:TextBox>
            </ItemTemplate>
        </asp:TemplateField>
        .....
    </Columns>
</asp:GridView>
```

The 'Price List' HyperLink opens a new window to show the full price list. It passes the 'Item ID' to the PriceList.aspx page and fetches pricing information

there (See Figure 19).

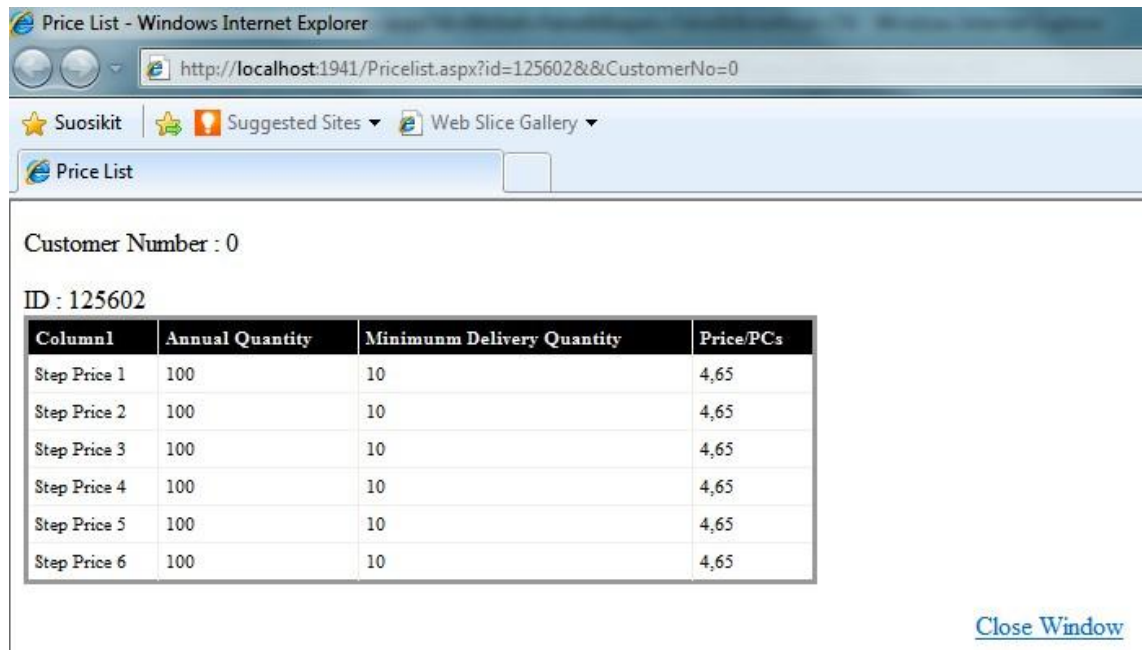


Figure 19. Pricing Information Window

There is a File HyperLink function which links to the related Excel document that the item information originates from. All the Excel documents are stored in the M-Files System. The M-Files System allows to access files by using M-Files Links which is the easiest way to quickly locate and open the needed files. M-Files Links appear like 'http://lotta/mfiles/mfiles.pl?id={0}'. When accessing the related Excel document, the M-Files ID is passed to the M-Files Server. This ID is unique in the M-Files and identifies a unique document.

7. Conclusion

This project has now been completed. The author achieved a great improvement of the effectiveness of the previous quotation generating process by modifying and enhancing the previous Excel templates and creating a database and web pages. This project has impacted the quotation generating process very much. It has reduced the amount of manual typing work and price calculation; in addition, it has provided a more visible and rapid way of viewing, checking and modifying quotation information.

For generating quotation information, there is a better way to process all the quotation information including input, store, view, and modify on WebPages, namely, integration. However, this requires buying licences for new programs which could cost lots of money; at the same time, the old data cannot be accessed by new programs. In this case, Reka did not want to have this 'Better' solution. So, the current solution is the best for this situation.

References

- [1] Wikipedia, Internet Information Services, Consulted 21.4.2011
http://en.wikipedia.org/wiki/IIS7#Version_7.0
- [2] Wikipedia, MySQL, Consulted 21.4.2011
<http://en.wikipedia.org/wiki/MySQL>
- [3] MSDN, Visual C#, Consulted 21.4.2011
[http://msdn.microsoft.com/en-us/library/kx37x362\(v=VS.90\).aspx](http://msdn.microsoft.com/en-us/library/kx37x362(v=VS.90).aspx)
- [4] Wikipedia, Visual Basic for Application, Consulted 21.4.2011
http://en.wikipedia.org/wiki/Visual_Basic_for_Applications
- [5] Wikipedia, .NET Framework, Consulted 21.4.2011
http://en.wikipedia.org/wiki/.net_framework
- [6] Wikipedia, ASP.NET, Consulted 21.4.2011
<http://en.wikipedia.org/wiki/ASP.NET>
- [7] Wikipedia, Open_Database_Connectivity, Consulted 21.4.2011
http://en.wikipedia.org/wiki/Open_Database_Connectivity
- [8] MSDN, What is Excel, Consulted 21.4.2011
<http://office.microsoft.com/en-us/excel-help/basic-tasks-in-excel-2010-HA101829993.aspx?CTT=5&origin=HA010370218>
- [9] Wikipedia, HTML, Consulted 21.4.2011
<http://en.wikipedia.org/wiki/Html>
- [10] Motive System, The Document Management Solution for Your Business, Consulted 21.4.2011
<http://www.motivesys.com>
- [11] Wikipedia, Apostrophe, Consulted 21.4.2011
<http://en.wikipedia.org/wiki/Apostrophe>