SZAKDOLGOZAT

Tanczenberger Attila

Debrecen

2009

Debreceni Egyetem
Informatikai Kar

Introducing Microsoft Office SharePoint 2007 and application in a business environment

Témavezető:
Jagó Jácint

Matematikus

Készítette:
Tanczenberger Attila
Programozó

Debrecen
2009

1 Preface

The main reason for choosing this rather new technology as subject of my thesis was my continued involvement with Microsoft SharePoint over the last two years. As a result I have been both analysing and applying this technology in real life scenarios and experienced the impact it has on business firsthand. Besides the reason of my working with this technology, another one that made me choose this product as grounds for my thesis was the fact that after my first encounter with it, I was truly fascinated by the scope and goals this solution has set for it self, and as well by the means it achieves these goals. Furthermore from my standpoint, the developer standpoint, the technology is a culmination of just about every technology Microsoft has developed over the years, which makes it a challenge to develop customizations to it, while using well known and well documented API to achieve it. It is a complex system, which requires knowledge ranged from simple ASP.NET and C# to XML and a variety of new technologies, like the Collaborative Application Mark-up Language (CAML). And the final, but by no means the least reason is that Microsoft SharePoint, more than any other technology for collaboration purposes, has been constantly in the limelight of the industry for the last two years.

As Michael Noel and Colin Spence put it in their book "It is rare for a technology product to attract as much attention as SharePoint has in recent years. The industry has historically paid little attention to new product suites, particularly those related to web design. SharePoint products and technologies, however, have managed to excite and rejuvenate industry followers, causing them to take notice of the ease of use, scalability, flexibility, and powerful document management capabilities within the product."[1] This thesis will try to elaborate briefly why this technology caused such a stir in the industry, and also why it became a preferred way for building collaboration workspaces within companies. Although the scope of this document is not big enough to cover all that can be said about Microsoft SharePoint technologies, it will try its best to give an accurate overview of the technology, as well as introduce the possibilities for business customers via an application example. The first part of the document will be explaining what exactly the Microsoft SharePoint technology is, what it incorporates and aspires to achieve. Furthermore it will supply the reader with a brief

[1] Michael Nolin – Colin Spence: SharePoint 2007 Unleashed, 2007 Sams Publishing, Ch 1, P 5.

overview of the most important parts of any application based on this technology. The second part of the document will supply the reader with a real life example of the application of SharePoint technology, as well as with examples of customization possibilities. The first part will try to focus on introducing the technology from the business angle, while the customizations will show a deeper insight into the workings of the SharePoint application.

2 Introduction to the line of Microsoft SharePoint Technologies

2.1 The business background which spurned the creation of Microsoft SharePoint 2007

Many organizational needs have been present over the years, all which have contributed to the creation of this technology. Although the scope of this document does not allow to elaborate all of them, it will try to list and discuss the one which a company can and will face during its lifetime.

- **Better Document Management** – Although most companies have a system for document storage and a way to share these documents for all employees, it has mostly been based off the file system of the machines these documents are stored at. This method, while being very easy to implement, makes it very hard to comply to security requirements set by the customers, or to the guidelines of the company itself. The need is there for an easier and more productive way for the employees to find the documents they need for their work, and most of the time only the documents they need. These requirements contain the ability to add metadata to the documents, a versioning system that's easy to use and efficient, and the ability to scale document access security on a more detailed level.

- **Collaboration between users** – Collaboration is the way of a streamlined communication and exchange of information and documents between the users of the technology. In SharePoint, this usually refers to the usage of Document Libraries, Discussion Boards, or the exchange of tasks. The most important point of this need is that companies do not want to acquire different technologies to achieve these goals, nor do they want their employees to learn a different method for collaborating each time. This is where one of the key features of SharePoint comes into play, the ability to integrate all features of the collaboration with the range of Microsoft Office products. This gives a very easy way to encapsulate the collaboration part in a users every day life, plus it's a well known and used technology, which leads to a minimal learning curve.

- **Intranet** – Nowadays every company has its own intranet in place, but most of the time it is regarded as a bulky and complicated way of communicating information to the employees, as with normal web applications every change has to go a long way to reach the intended targets. Usually such a request has to go all the way through the hierarchy, ending at the IT, who then realizes this request and puts it up on the intranet page. As a result this need usually arises in the Management circles of a company, who don't have the time or the patience to wait for their urgent messages or tasks to be put up on intranet.

- **Information Search** – This is a common problem faced in all companies. The mostly used way of gathering information and searching for the needed documents is writing an Email to the co-worker or superior to ask for the location of said documents. This forms the need of a centralized search engine, which is available to all users and allows them to find the documents key to their jobs in a familiar and easy way, without having to rely on other users. The engine should allow the user to search for certain words in the document, a timeframe of its creation, or other criteria, which could be attached to the document as metadata.

2.2 Better Document Management

The most used feature of SharePoint 2007 is the Document Library. Most users will exclusively use this list to do their everyday work. This is the solution SharePoint offers over the storage of documents on a shared drive or the file system. Employees working most of their time with documents will spend immense time interacting with this part of SharePoint, so it is very important for the Administrator of the Collaboration to put in a lot of time and effort to create this workspace as suitable for the end users as possible. Figure 2-1 shows a typical Document Library.

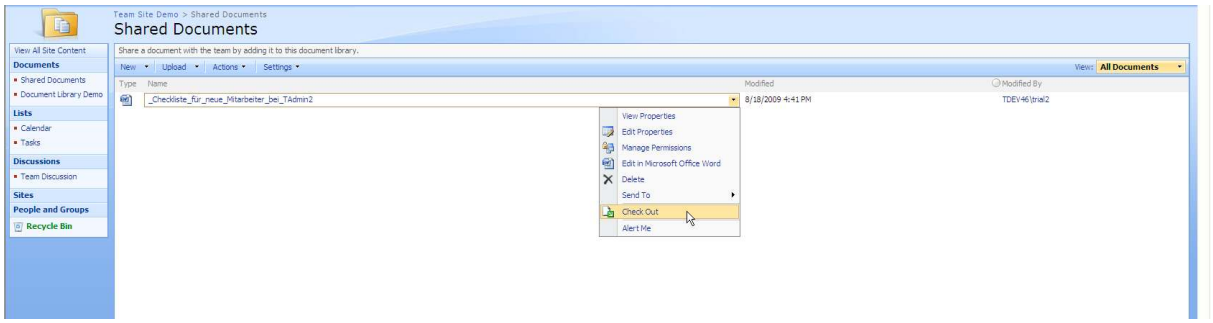

Figure 2-1

One of the most powerful features of SharePoint 2007 is the security trimming. This trimming applies not just to the documents themselves, but to the whole of the site and the document library. For example, if the user accessing this Document Library has only “Reader” privileges, then the view would change to Figure 2-2.

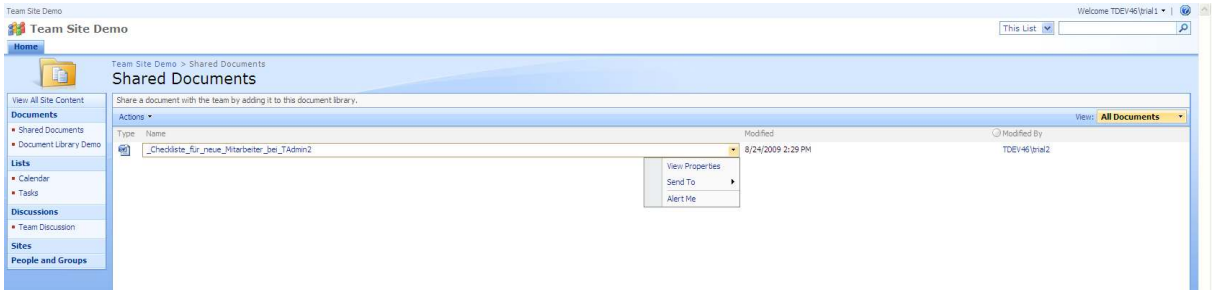

Figure 2-2

As we can see, only the Actions menu remains in the toolbar, as a user with “Reader” privileges has no right to add any document to this Library, nor rights to change the behaviour of this Library. But let us get back to Figure 2-1 and discuss the options SharePoint offers for the users of a Document Library. The toolbar offers ways to create a new document on the fly, based on templates stored for this library, the Upload menu allows the users to upload single documents, or multiple ones. The Actions menu provides additional functions, like opening this library in an Excel like Datasheet view, or to display the Library as a Web Folder via Windows Explorer. It is also possible via this menu to set Alerts for specific documents, so if a new version of this document is created, the user will be notified via Email. And last but not least it is also possible to attach this Document Library to the users Outlook application, and have the files available even when offline.

The context menu in Figure 2-1 offers a variety of functions as well. View Properties lets the user see any kind of extra information (metadata) that has been assigned to the document, such as the date of creation, the last date of change, version number etc. Edit Properties has the ability to edit this aforementioned metadata, if the user is supplied with the sufficing rights to do so. "Open in …" where the Office Application best suited for the file is displayed, gives the user the option to open the selected piece of information directly in the Office Application it was created in, allowing a quick and easy way to edit the information. Manage Permissions allows the fine grained setting of the documents accessibility.

The Administrators of the Document Library have additional options at their hand, via the Settings menu. As shown in Figure 2-3, they have the ability to alter many aspects of the document library, including the versioning settings, the overall permission sets for this library. Besides versioning, the administrator can decide whether an exclusive check-out of items is required before any changes are made to it, or if it is possible to create documents on the fly inside the library via templates. The administrator can also define which metadata is visible to the users and which are hidden, by creating custom views.

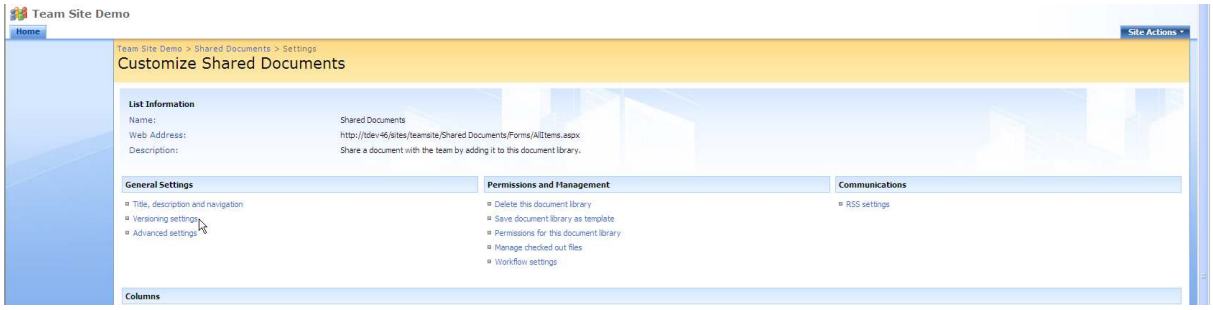

Figure 2-3

2.3 Collaboration between users

To allow the continual and easy information flow between users, SharePoint implements various Lists, which can be used in different aspects of business life. Although Document Libraries are easy to understand, because of their primary functions, the Lists are not that easy to define. SharePoint 2007 offers the following functionalities:

- **Announcements**
- **Contacts**
- **Discussion Board**

- **Links**
- **Calendar**
- **Tasks**
- **Project Tasks**
- **Issue Tracking**
- **Survey**
- **Custom List**
- **Custom List in Datasheet View**
- **KPI List**
- **Languages and Translators**
- **Import Spreadsheet**

All these Lists share a basic structure. They are built of columns and rows, like a spreadsheet. What they differ in is the contents of these columns and in some instances the representation of the Data contained in the rows. Following the most frequently used List types will be introduced.

2.3.1 Announcements

The Announcements list is used to share important information with all users or a group within a project. The entries can be set to be alive for a certain timeframe, which also automates the revoking of these entries after they have lost their importance. Figure 2-4 shows a typical Announcement on the front page.

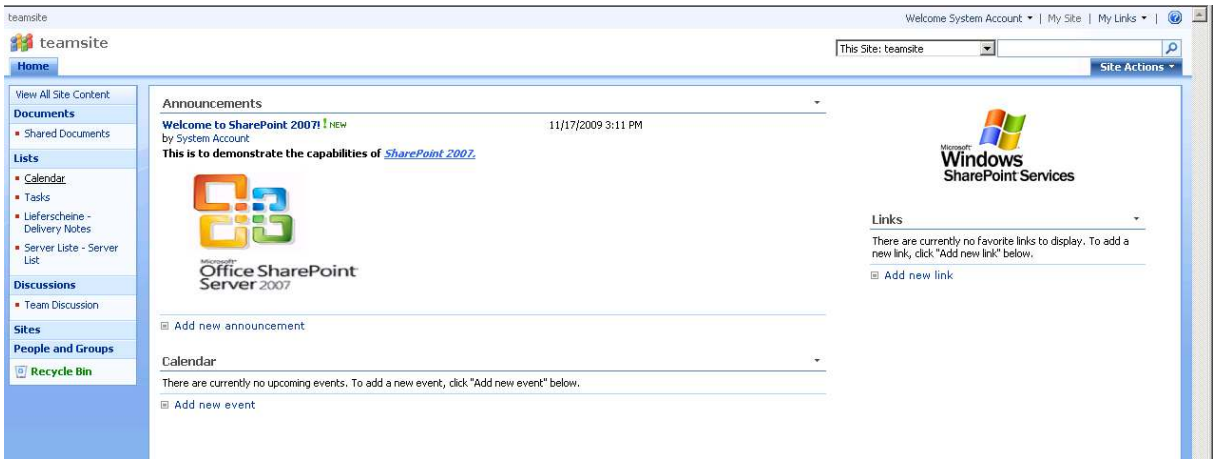

Figure 2-4

Other various benefits include:

- The entry can include pictures
- The text of the entry can be formatted with a wide arrange of formatting tools. The user can design the text just as easy as with any WSIWYG editor of his choice. Formatting Tools also include the option of directly working with HTML inside the announcements, and a spell checker is also available. Entries can include hyperlinks to internal or external locations.
- Lists can have versioning as well. Though not having major and minor version numbers, with the Content Approval it is possible to retain the draft version of Announcements
- The number of retained version can be limited

2.3.2 Contacts

The Contacts List is an excellent place to gather information about every member of a project or organizational unit. It holds the needed information of every user in one easily accessible place. Figure 2-5 shows a sample Contact List.

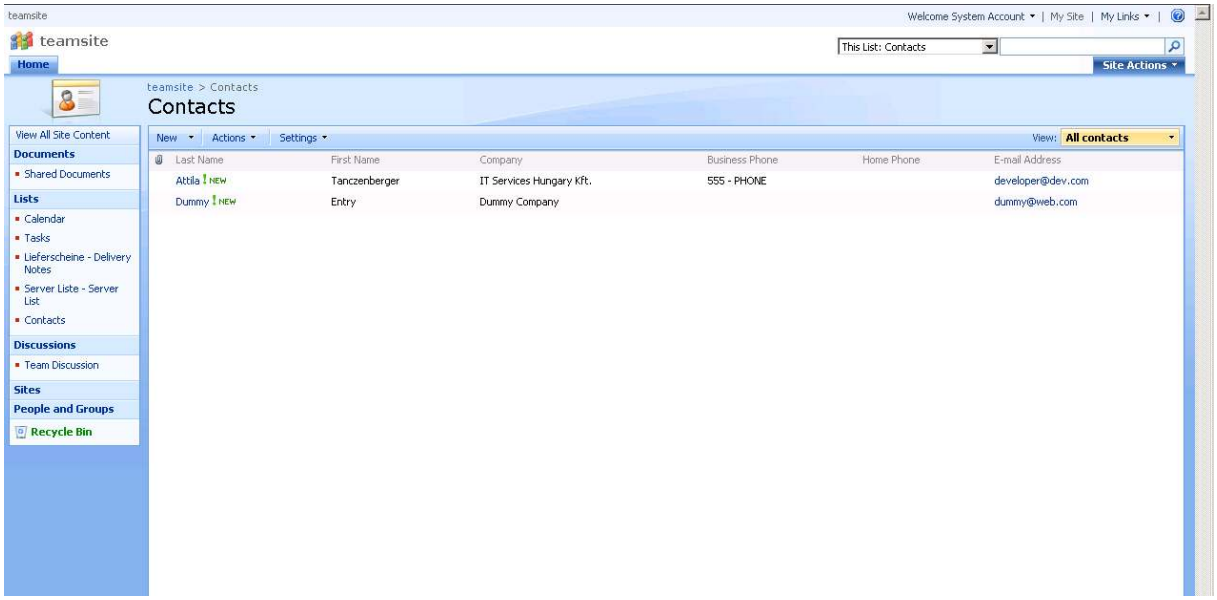

Figure 2-5

To add a new contact, one just has to use the New menu option and fill in the fields, as shown in Figure 2-6.

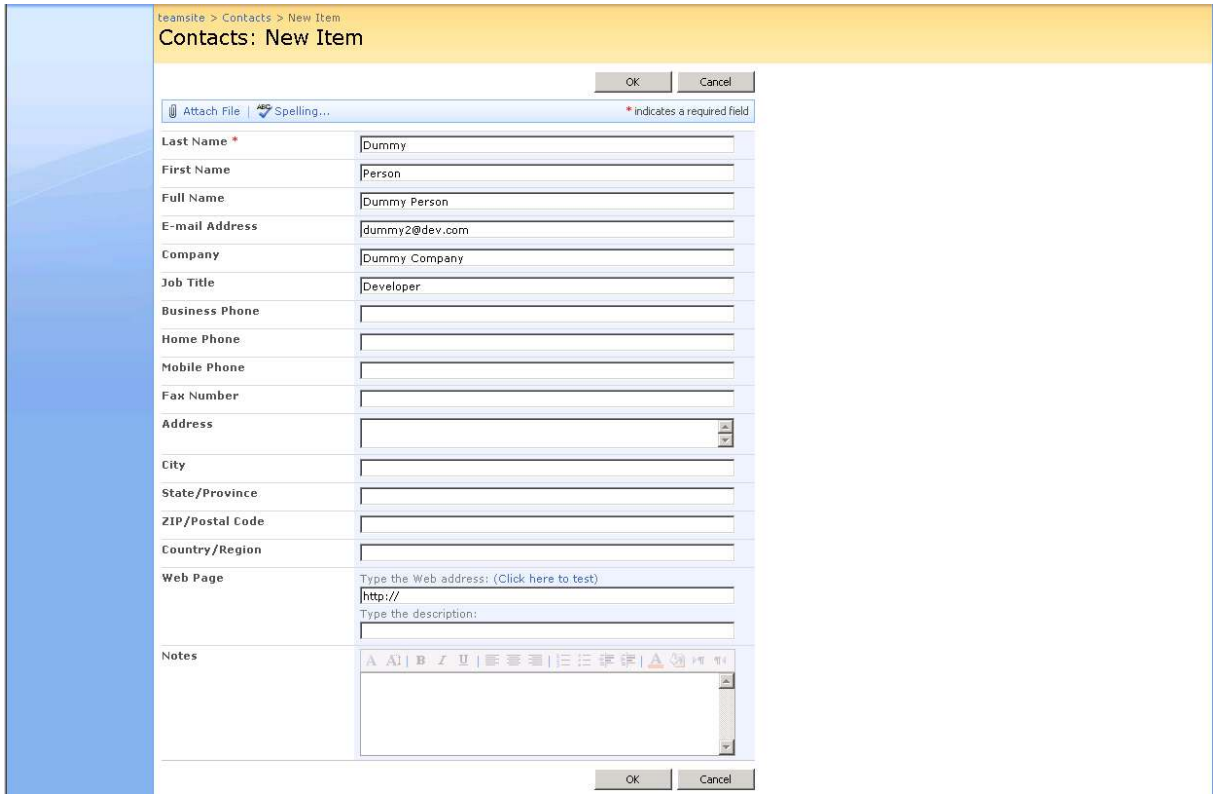

Figure 2-6

SharePoint 2007 can synchronize the contents of this list with the users Outlook Contacts list, by simply choosing the “Connect to Outlook” option from the actions menu. The result is shown in Figure 2-7.

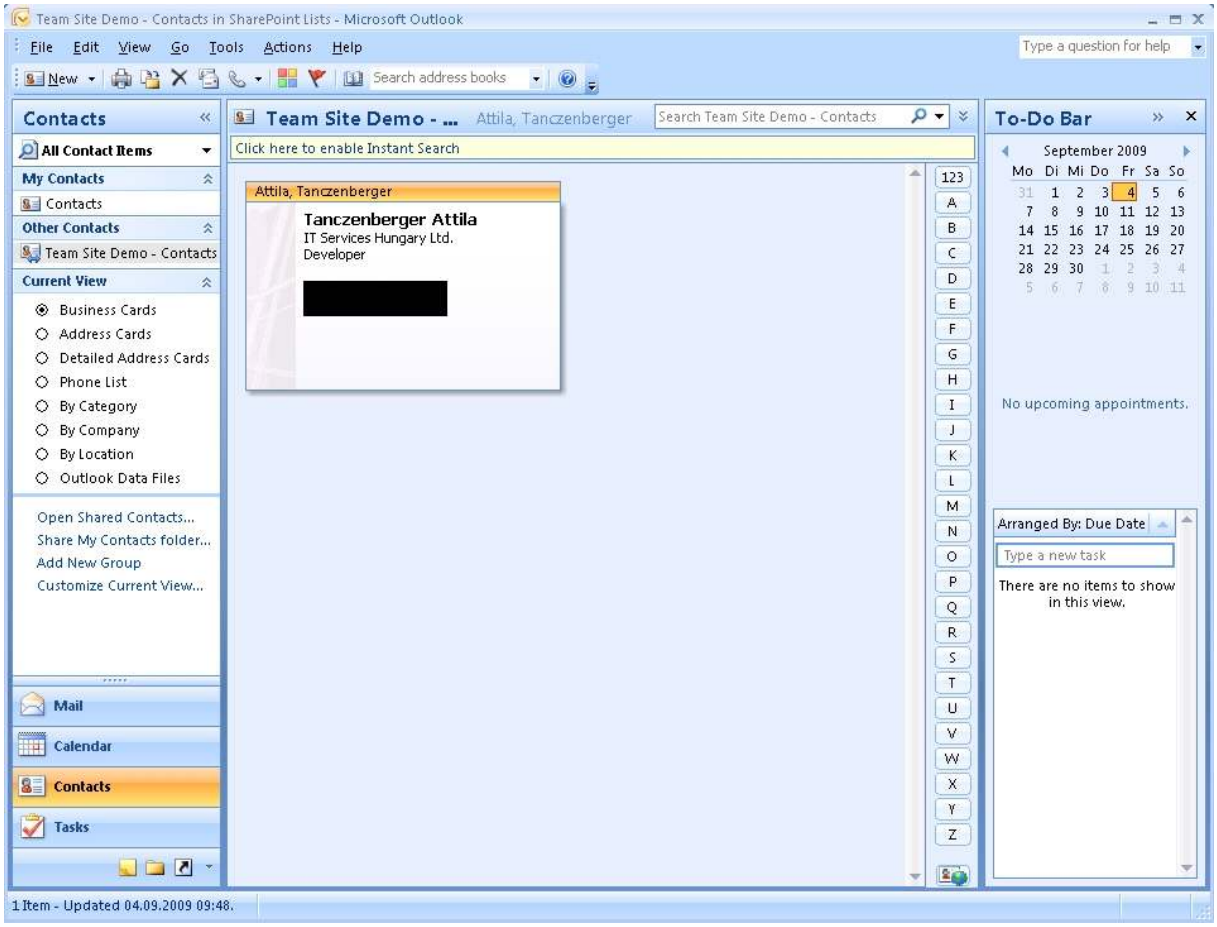

Figure 2-7

2.3.3 Calendar

The Calendar provides functionalities to coordinate absences and leaves of project members on an interface that is visible and editable for all members. It also offers the management an overview of their work resources. The presentation of the calendar is like the calendar most users see in Outlook 2007 (Figure 2-8), which adds to the familiarity and ease of use of this list.

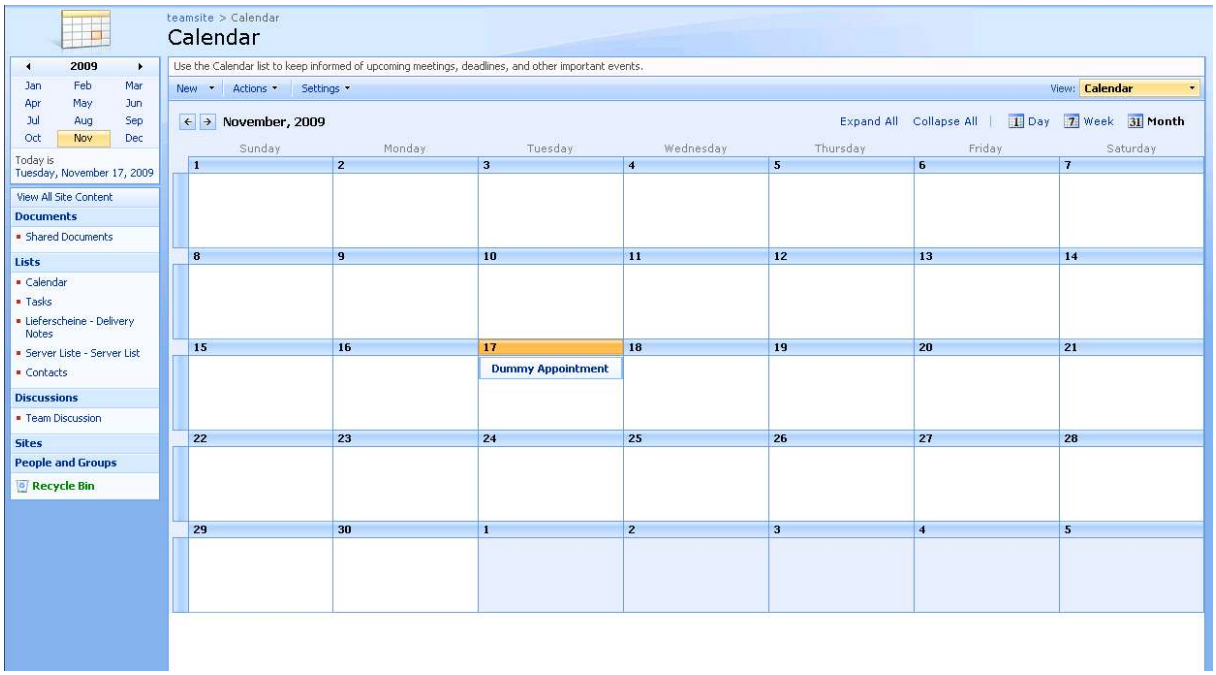

Figure 2-8

To add a new entry the user simply fills out the fields on the form appearing after clicking New in the toolbar (Figure 2-9)

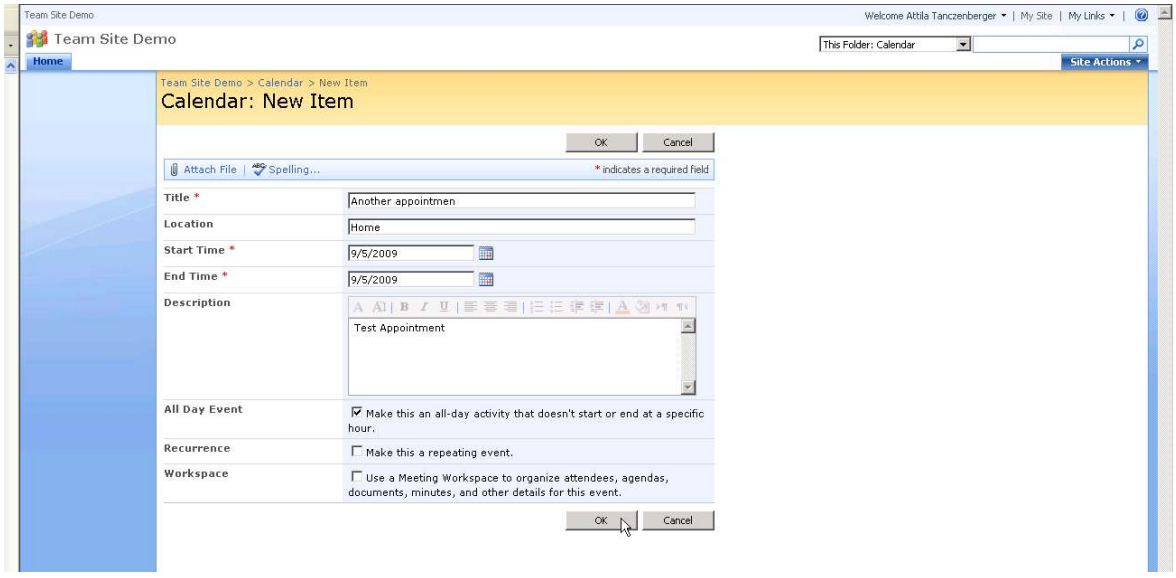

Figure 2-9

The Calendar can also be synchronized with Outlook, resulting in Figure 2-10.

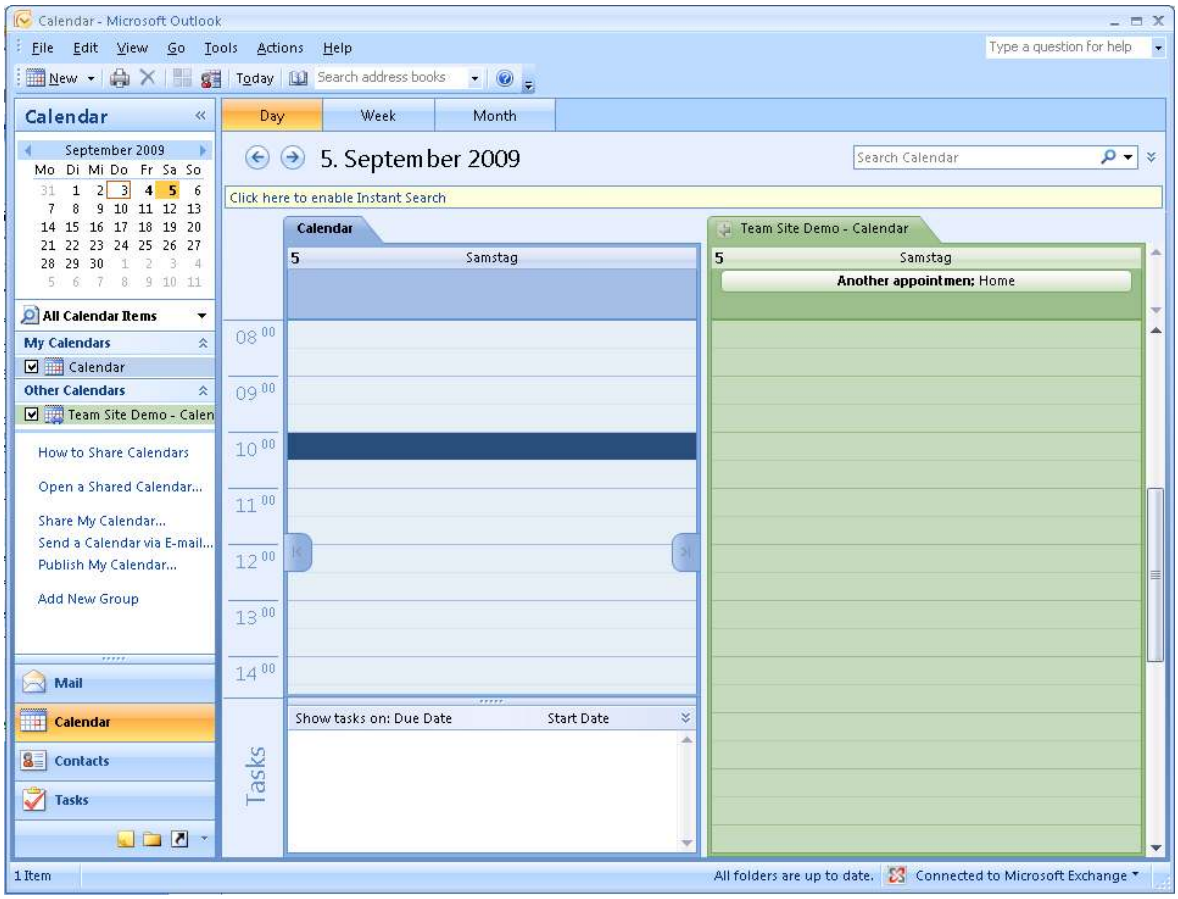

Figure 2-10

2.3.4 Discussion Board

SharePoint 2007 offers a Discussion Board similar to internet based forums, for the sake of communication between members, without the needs of using Emails to discuss problems that concern more than one individual. Users have the ability to limit the amount of information shown on a topic with the use of the Discussion Board views. The Subjects views only display the top level entries, while the threaded view shows all answers that have been added to the selected discussion. All answers are shown indented, providing a better overview of the information displayed. Though not as powerful as the forum engines created for this purposes, it is functional and should suit the needs of the company. Figure 2-11 shows a typical discussion board.

Figure 2-11

2.3.5 Survey

Surveys are an integral part of every business. Whether it is an internal survey for employee matters, or one for the customers to take part in, a company has to have the ability to create, monitor and analyse these surveys. SharePoint offers a solution for this, in form of the Survey List. Creating a new Survey is easy and user friendly, and happens within the SharePoint UI context, without the need for the maker of the survey to learn any new technologies. The user clicks new in the Survey Lists interface, and fills in the information, as shown in Figure 2-12. and 2-13.

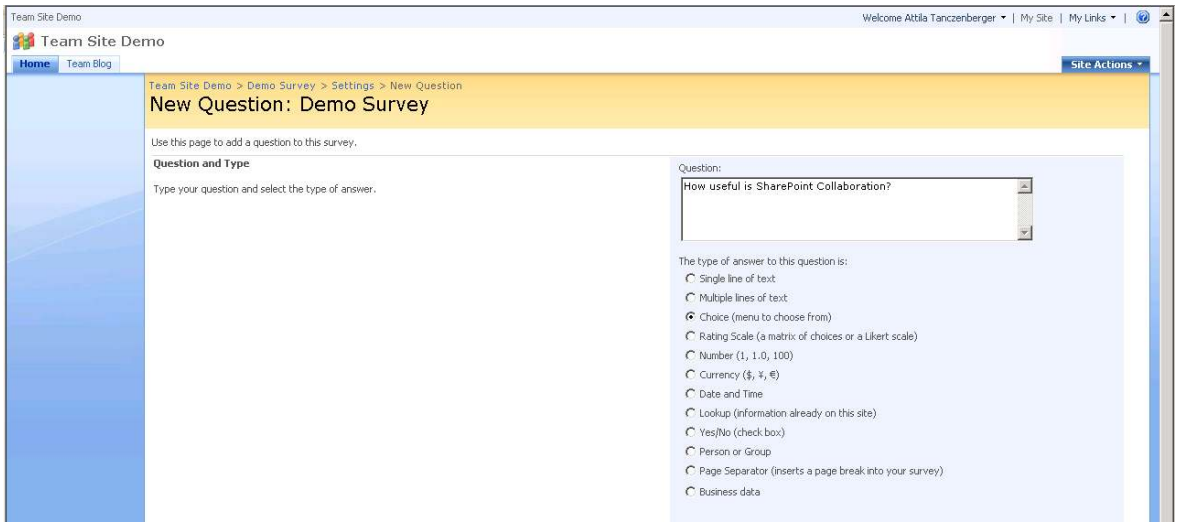

Figure 2-12

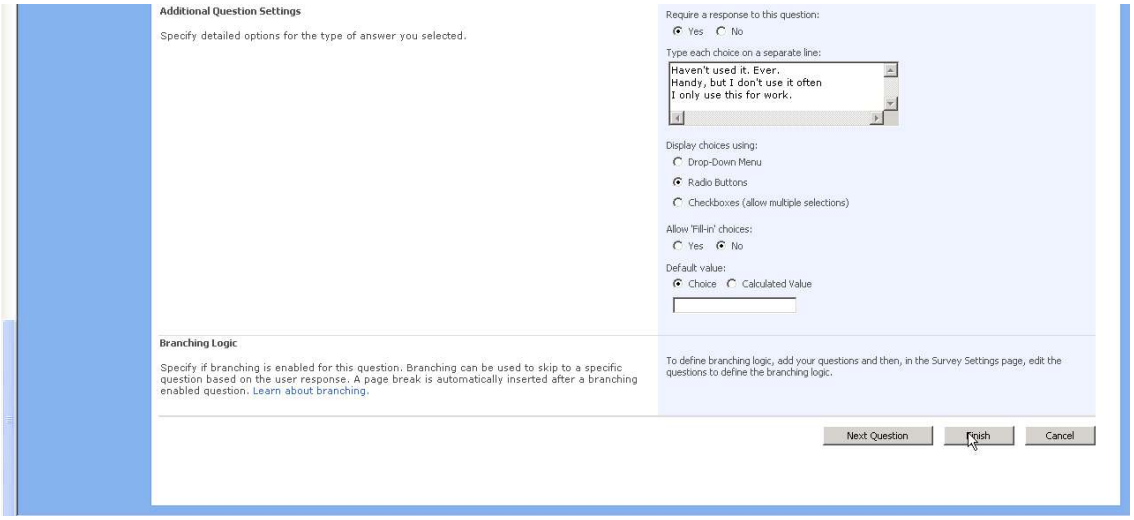

Figure 2-13

After the Survey has been created, the visitors can fill it out. What they see is shown in Figure 2-14.

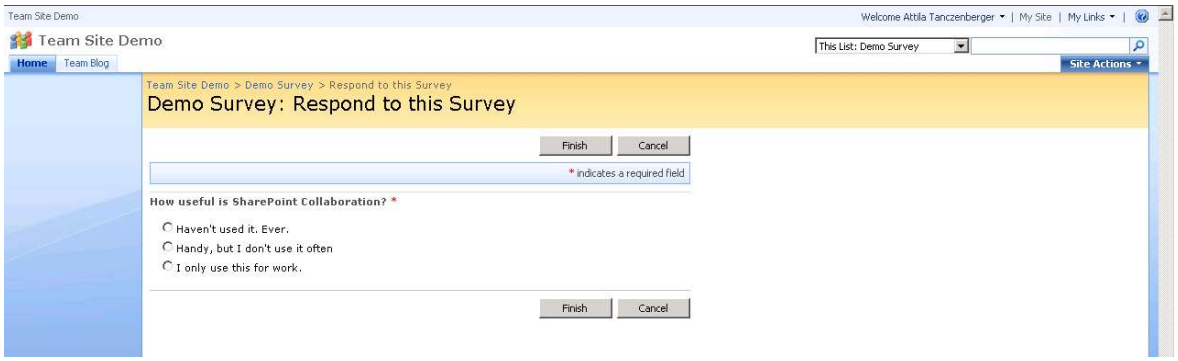

Figure 2-14

2.3.6 Tasks List

Tasks are an important part of any business action. They are crucial in organising the day-to-day actions of employees, the steps needed to take in projects and business transactions, and to successfully leverage work amongst the employees. SharePoint 2007 offers the Tasks List as a solution to handle this obstacle in the company. Figure 2-15 shows a typical Tasks List. The fields can be expanded if the current complexity is not enough to tackle the company needs.

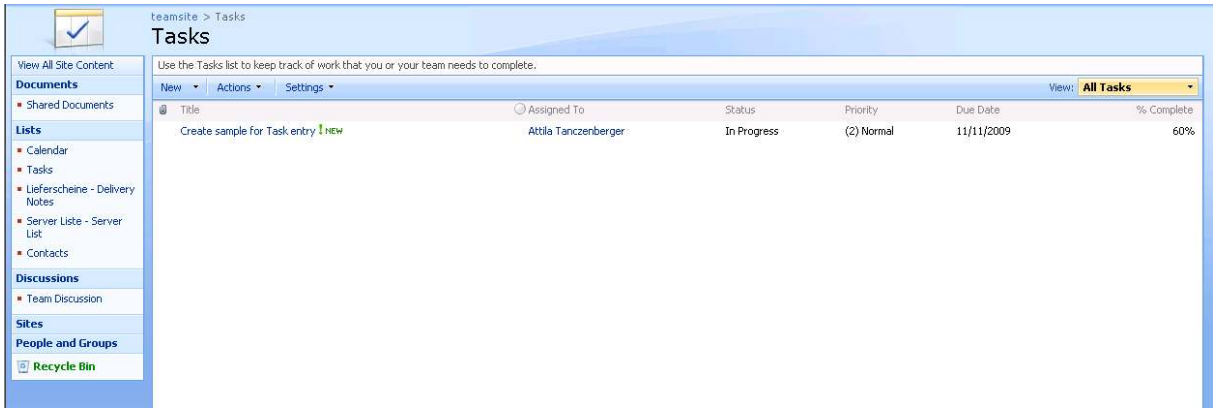

Figure 2-15

The new Task Item (Figure 2-16) is created within the SharePoint UI, and is easy to navigate. Most of the fields are fill-in textboxes or Dropdown menus, which provide an accessible and neatly sorted amount of information. The most important field to be filled out is the "Assigned to" field. This field can contain the user name of any of the Collaboration users or a whole Group of users allocated within the Collaboration Workspace. The field has two helper objects; one is a tool to authenticate the entry give, while the other opens up a search interface, to help the creator of the task find the appropriate person.

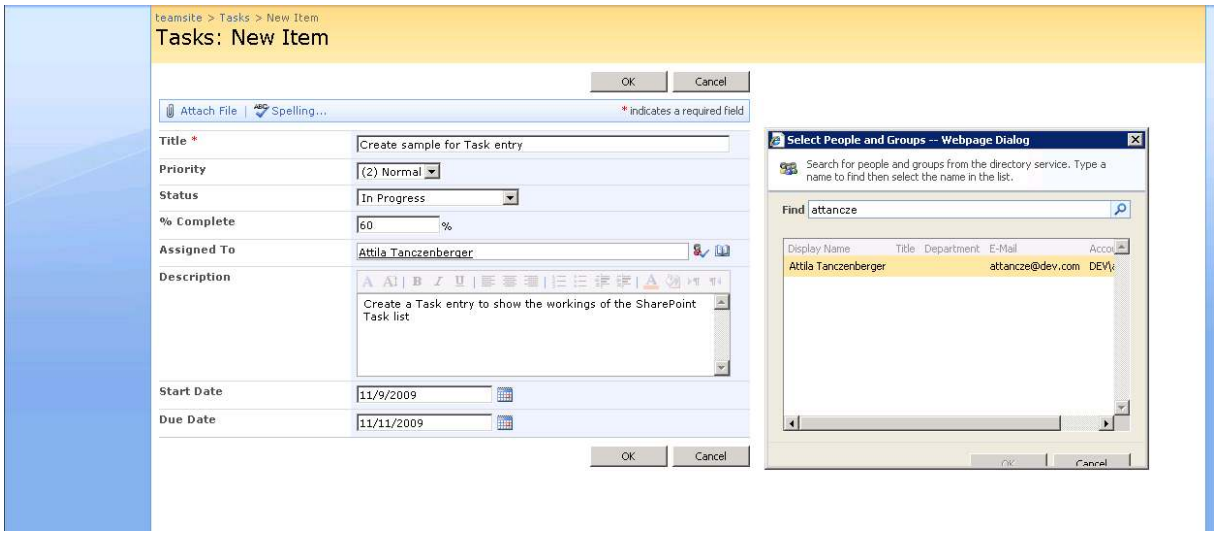

Figure 2-16

Optionally it is possible to set the task in a way, that every assignment of ownership triggers a mail to be sent to the new owner. And as with most of the Lists, the Task list has an option to synchronize with Outlook 2007 and add the given Tasks to the responsible person's task list. And even in the UI view of the tasks within SharePoint, the use of the built in filtered views allows the users to easily access their own tasks, or check for the ones that must be completed in the near future.

2.4 Intranet

As shown in the previous chapter, SharePoint 2007 provides collaboration solutions to just about every part of a successful environment. This all is packaged into a web based portal service, which is ideal for usage as an intranet service. But what a company really is striving for is an intranet that can be tailored not only in function, but also in look and feel to their own standards. SharePoint has a solution for this need as well.

2.4.1 Theming

To apply the corporate identity of one's company to their intranet is a basic need for everyone. SharePoint offers an easy and accessible way of doing this. From the top-level site of a SharePoint Collaboration to the last list everything is governed by a web-based design.

The foundation of it is ASP.NET. It uses a Master Page for the overall look and feel, which is inherited by any sub-site or List created within SharePoint. Colour schemes and pictures used throughout the collaboration are added with Cascading Style Sheets. And to ease the application of these for the Administrator, a collection of a master page, the images and CSS files belonging to it can be put together into one solution, a "theme". These themes can then be deployed on the server, and are then readily available for the Administrators to apply them to his site. As the scope of this document, to provide a wide overview of SharePoint Technologies, is not deep enough to cover the whole of this segment, the hows of creating a theme solution are omitted here.

2.4.2 Custom Information placement

As every man differs from another, so does our needs in the display and amount of information we perceive as useful on a collaboration site. Especially if talking about intranet, in the past it was very difficult, to tailor the nodes for employees to their liking. Enter SharePoint 2007, where the possibility exists of personalized version of Pages. Every page in the Collaboration (give the sufficient rights, of course) can be personalized by each user to a certain extent. As the pieces of information are shown in ASP.NET Web Parts, it is easy to allow a user to have a personalized version of the top page for example. He can access the "Edit Page" function, and place Web Parts containing his Tasks, Documents or Appointments in one of the provided Web Part Zones. This version of the page will only be visible to him, and does not alter the way any other users sees the top page. Every version is stored separately inside SharePoint and so every user can set up his work environment, without having to go to lengths contacting IT about certain changes, from which most cannot even be granted, as it would make things difficult for other users.

2.5 Information Search

The capability of search in a collaborative environment is imperative for and efficient and user friendly working environment. SharePoint 2007 offers a vast search engine with superior capability to cover this need in just about every aspect of its collaboration solutions. The different versions offered inside the SharePoint product family can be used to scale and select the needed search options for a company. Figure 2-17 shows the advanced search page of

SharePoint 2007. It is apparent that this interface is very intuitive and offers the users an easy way to pinpoint the information they need to find.

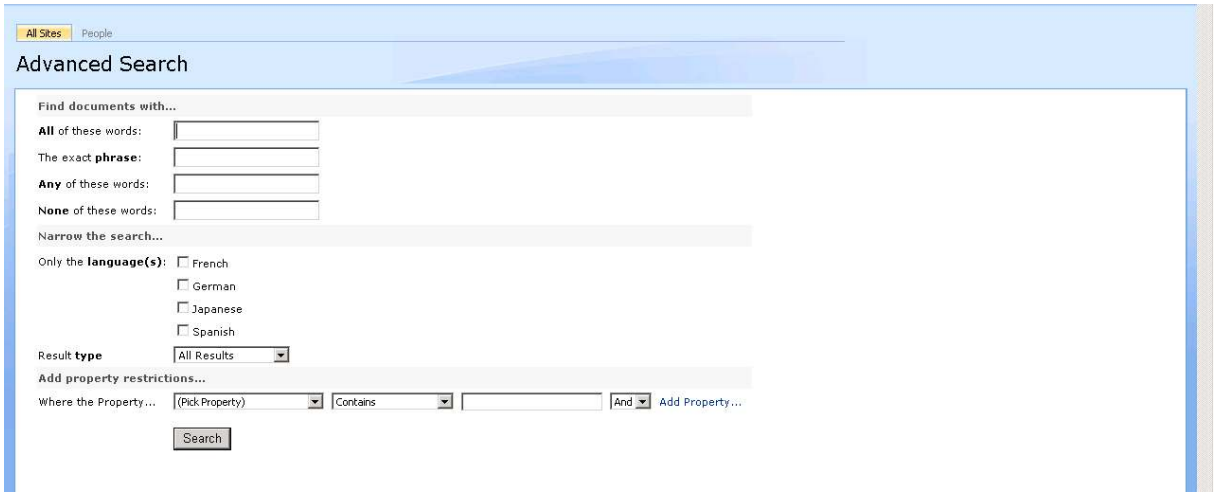

Figure 2-17

This interface allows the user to include or exclude words at the search at will, as well as the ability to restrict the search to certain languages. The type of expected results can be altered to, for example one can limit the search results to Word Documents or Excel Tables. The last option is the ability to search for information with the usage of the metadata added to each document or item in a List. The metadata that has been indexed appears in the dropdown list and can be selected by the user (named Property). Then the relation of the user added information of the Property can be selected (Contains, Begins with, etc.) and in the last field the user enters the phrase he is searching for. The results are displayed as a web form, where each result item is shown with its name, various properties, as well as the first rows of information contained within, as shown in Figure 2-18

Figure 2-18

Changing the search page to the “Peoples” tab, the user can use the capabilities of the SharePoint search engine to search for user related information.

3 Real World application

The previous chapter focused on the introduction to the SharePoint product line, and this part will show what the application of this product in real life looks like. The application example we are using uses only a few of the functions described in the previous chapter, but will contain a number of customized features, as in most cases the out-of-the-box solutions will not meet the demands set by the company. After a short description and requirements of the example it will mainly focus on the customized parts of the project, to show the possibilities of SharePoint customization.

3.1 Description of the example

X Company wants to create a centralized research study database. After careful examination of the current market, it decides to utilize SharePoint Search Server 2008 as the base product to achieve this goal. The reasons are as follows:

- Normally if a customer requests a study from the company, it is done via Email of telephone. After that the contacted employee checks his hard drive or uses other means to get the needed study. This approach is not flexible of efficient enough
- Market researchers spend much time in acquiring said documents, but don't have the time to add every property of these documents to a separate storage location for search purposes
- Even if the database described before would exist, there is no guarantee that every employee would know about its existence.
- Said employees work on a need-to-know basis, so all sources of studies are not known to them

After leveraging these needs the company decided that the functions of SharePoint 2007 are the ones needed to solve all these problems, and create a centralized Document Storage for the research study documents.

Requirements are as follows:

- Centralized Document Libraries for study data, utilizing Access Restrictions
- Management of users and user groups
- Corporate Identity

- Multilanguage capability
- Specialized Search capability throughout the whole Application
- Upload rules for the adding of studies to the Document Library
- Automatic metadata extraction upon upload

As the first three of the requirements are met by SharePoint by default, the product for realization of the project was set.

3.2 Planning the structure of the project

In accordance with the requirements set by the contractor, the following set-up was decided upon. One Site would be used to accommodate all the functions of the project, available in both German and English language. Two Document Libraries will be used, for Primary and Secondary studies.

The last three requirements required customization of the standard SharePoint interface, and following actions have been decided:

- A special Event Handler will be created to handle the metadata extraction, upload rules and Access restrictions which had to be enforced
- To accommodate said Event Handler, a SQL database will be created to hold the additional information required for metadata extraction
- Other commonly used metadata will be stored in SharePoint Lists.
- To accommodate the Search specialization requirements, a customized Search interface will be implemented, along with a custom Search Result interface
- Multilanguage will be solved by making the web parts available in both languages. Built in functionalities rely on the language locale of the underlying server.

3.3 Out of the box elements

Figure 3-1 and Figure 3-2 show the two document libraries implemented via out-of-the box capabilities.

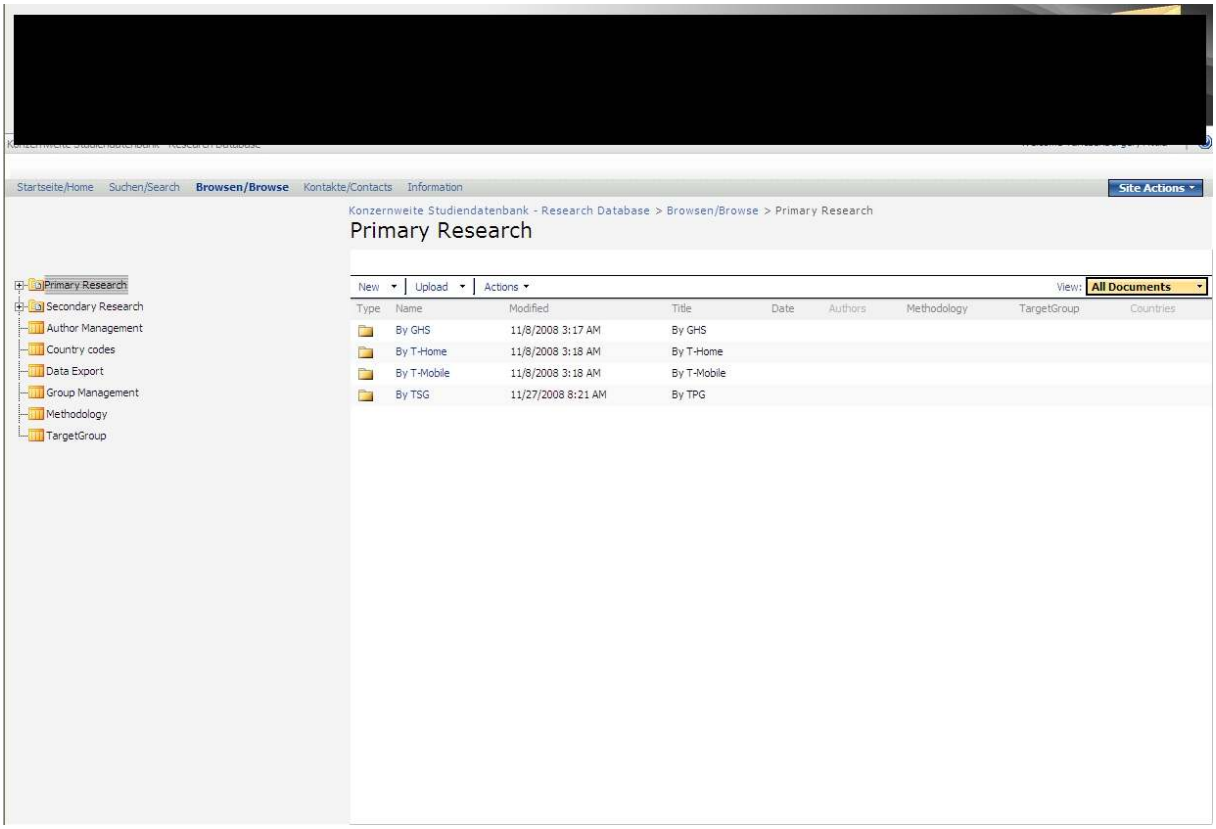
Startseite/Home
Suchen/Search
Browsen/Browse
Kontakte/Contacts
Information
Site Actions
Konzernweite Studiendatenbank - Research Database > Browsen/Browse > Primary Research
Primary Research
Primary Research
Secondary Research
Author Management
Country codes
Data Export
Group Management
Methodology
TargetGroup
New
Upload
Actions
View:
All Documents
Type
Name
Modified
Title
Date
Authors
Methodology
TargetGroup
Countries
By GHS
11/8/2008 3:17 AM
By GHS
By T-Home
11/8/2008 3:18 AM
By T-Home
By T-Mobile
11/8/2008 3:18 AM
By T-Mobile
By TSG
11/27/2008 8:21 AM
By TPG

Figure 3-1

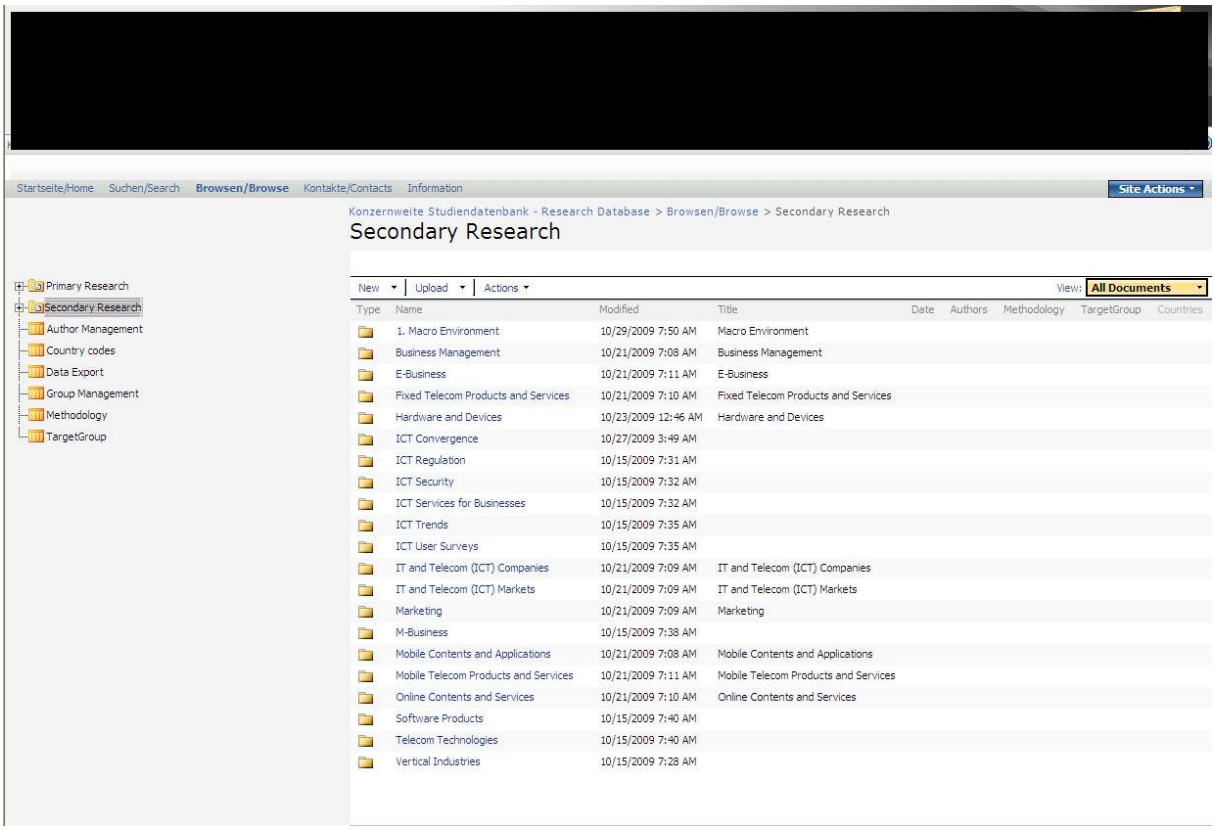

Figure 3-2

To store metadata widely used over the project, several Custom Lists have been created, using the built in List template.

3.4 Customization

3.4.1 Upload rules and metadata extraction

3.4.1.1 Preface

Uploading documents needs special attention with the **name** of the documents. There is a standard which must be kept.

Example:
2006-01_TMO+ipsos_Terminal Basics QUANT+B2C_AT+CZ+DE+NL+UK.doc

The filename is a composition of **metadata**. Metadata are words separated with "_" sign. The metadata are used for setting up the property fields of the uploaded file.

The first segment represents the date in the 'yyyy-mm' form.

In the second segment are defined the authors, in the '<authorname>[+authorname]' form. It has to be given at least one author name, in case of more, the others have to be separated by the '+' delimiter character.

The third segment contains the document's title, followed by a " " (space) delimiter character and by the methodology name and target group name. The last two segments are optional, if both are given, they have to be separated by the '+' delimiter character. The target group name can't be given without the methodology name; however the methodology name can be set without defining the target group name.

The last segment contains country codes delimited by the '+' separator.

Figure 3-4 shows the properties after metadata extraction.

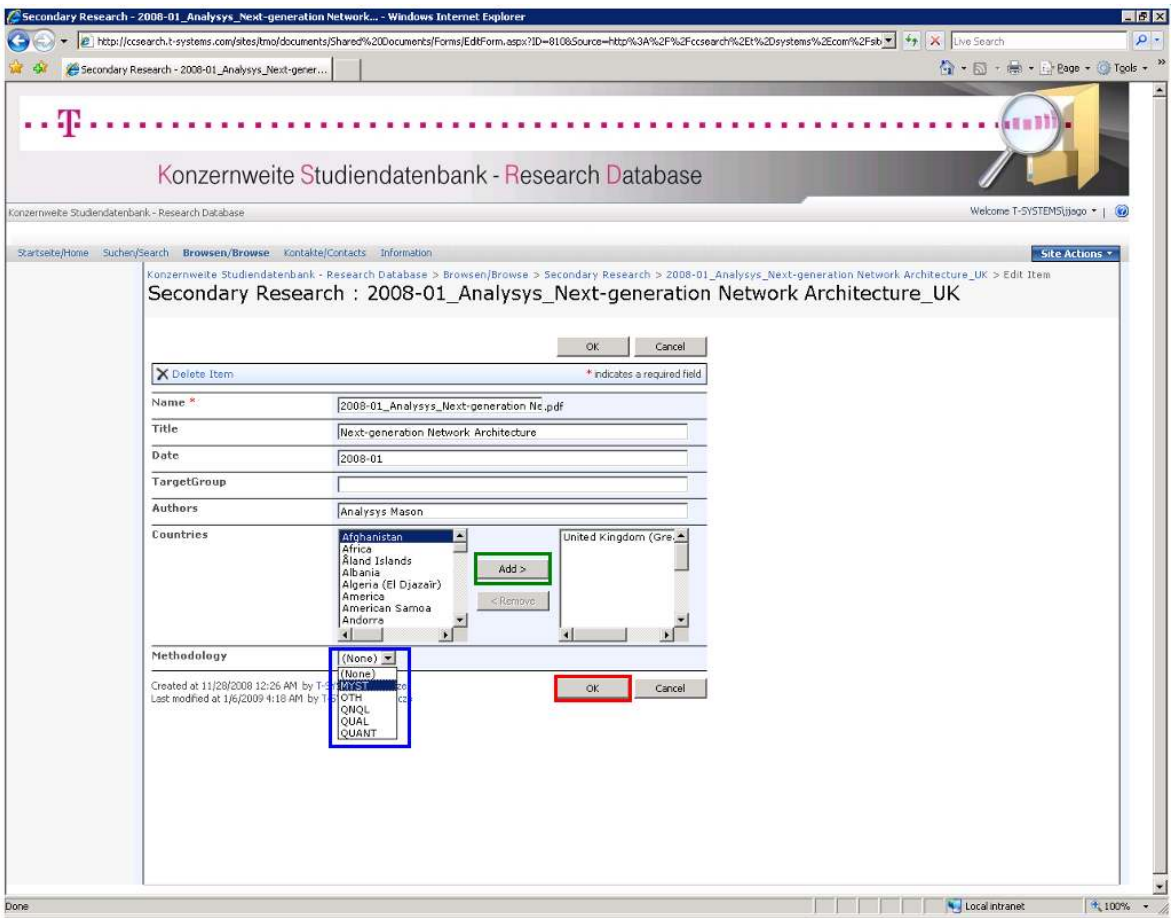

Figure 3-3

3.4.1.2 Implementation

To achieve this functionality a so-called Event Handler has been attached to the document libraries, which monitor any change on them. If a file is being uploaded, it fires and Upload and Update event, which is then handled by this Event Handler.

3.4.1.2.1 ItemAdding

```
void ItemAdding(SPItemEventProperties properties)
```

If the SPItem is not a directory, the method sets the current SPItem's metadata, excluding the title.

3.4.1.2.2 ItemAdded

```
void ItemAdded(SPItemEventProperties properties)
```

Sets the title metadata and the roles for this file if the uploader is in the 'provider' group (in this version the roles are set in every cases)

3.4.1.2.3 ItemUpdating

```
void ItemUpdating(SPItemEventProperties properties)
```

Checks if the user changes the metadata and verifies the updated fields. If the updated data is not correct, it cancels the update.

3.4.1.2.4 ItemUpdated

```
void ItemUpdated(SPItemEventProperties properties)
```

Sets the title metadata and the roles for this file if the uploader is in the 'provider' group (in this version the roles are set in every cases)

3.4.1.2.5 AuthorCalculator

```
ArrayList AuthorCalculator(string p)
```

Gets the author nicknames from the “p” string. Using these nicknames, it gets the authors full names from the database and puts them into a list.

A string that contains author names. Author names are separated by the '+' delimiter.

3.4.1.2.6 CountryCalculator

```
bool CountryCalculator(string countryCodesString)
```

CountryCalculator decides if the country codes are correct in the “countryCodesString” string.

Checks the country code abbreviations (in string countryCodesString) that are in SharePoint’s "Country codes" list.

<param name="countryCodesString">a string that contains country code abbreviations. Country codes are separated by the '+' delimiter </param>

<returns>If all country code are in SharePoint’s "Country codes" list, it returns a true value, otherwise a false one</returns>

3.4.1.2.7 TitleCalculator

```
string[] TitleCalculator(string p)
```

Using the “p” string, returns a string array containing 3 emelments as it follows: 'real' title, methodology and targetgroup.

3.4.1.2.8 filenameSplit

```
string[] filenameSplit(string fullFilename, int mode,
SPItemEventProperties properties)
```

Gets the file name from the “fullFilename” string which is an url string. Splits the filename by the '_' character and returns a string array containing the metadata.

fullFilename: file name that is an url string.

Mode: Sets the behavior of filenamesplit function. If mode==1 then fullfilename is an url. If mode==0 the fullfilename is exactly the file name.

Properties: Is a SPItemEventProperties object, that is needed to decide if the filename is a directory or not.

Return value: It returns a string array that contains the metadata.

3.4.1.2.9 getMetadata

```
void getMetadata(string[] metaadat, ref DateTime date, ref
ArrayList authorNames, ref string[] titleName, ref Boolean
countries, SPItemEventProperties properties)
```

Sets the ref parameters using the metadata string array.

Metaadat: metadata

date: in/out datetime parameter

authorNames : in/out ArrayList parameter

titleName: in/out string array parameter

countries: in/out boolean parameter

properties: it is needed to open the database connection

3.4.1.2.10 isDuplicated

```
bool isDuplicated(string filename, SPItemEventProperties
properties)
```

Checks if the file is uploaded in SharePoint

If the file exists in SharePoint it returns true otherwise false.

3.4.1.2.11 OpenARCIOBConnection

```
void OpenARCIOBConnection(SPItemEventProperties properties)
```

Opens the database connection and fills the local methodology, targetgroup, countrycodes lists.

Properties: It is needed to get the server name.

3.4.1.2.12 OpenARCIOBConnectionElevated

```
void OpenARCIOBConnectionElevated(SPItemEventProperties
properties)
```

Opens the database connection and fills the local methodology, targetgroup, countrycodes lists.

The difference between this function and the upper one is that this one can be called only in runelevated function.

3.4.1.2.13 SetMetadata

```
void SetMetadata(SPListItem current, DateTime date, ArrayList
authorNames, string[] titleName,string mode)
```

Updates the current item, sets it's value using the metadata. It is used in itemupdated.

3.4.1.2.14 SetMetadataAdding

```
void SetMetadataAdding(SPItemEventProperties properties,SPWeb
webElevated, DateTime date, ArrayList authorNames, string[]
titleName)
```

Updates the current item, sets it's value using the metadata. It is used in itemadding.

3.4.1.2.15 SetRoles

```
void SetRoles(SPListItem current, SPWeb web, ArrayList
authorNames)
```

Sets the current item's role. Uses the previously opened database connection.

3.4.1.2.16 isDocumentProvider

```
private bool isDocumentProvider(Guid sitidd,
SPItemEventProperties properties)
```

Checks if the uploader has provider role.

3.4.2 Search Web Part

3.4.2.1 Preface

The goal of the Search Web Part is to offer an easy and fast interface to select the appropriate search options, which then are forwarded to the Search Result Web Part to execute the search query and display the results. The

Search Web Part encompasses to functions. First it is a UI for the user, and according to the settings the user takes on this UI, the Web Part builds the Search Query and forwards it, as well as other crucial pieces of Information. All functions contained in the Web Part are used to

1) populate the various selection options in the UI
2) language settings
3) and finally to create the search query

Figure 3-4 shows the custom Web Part.

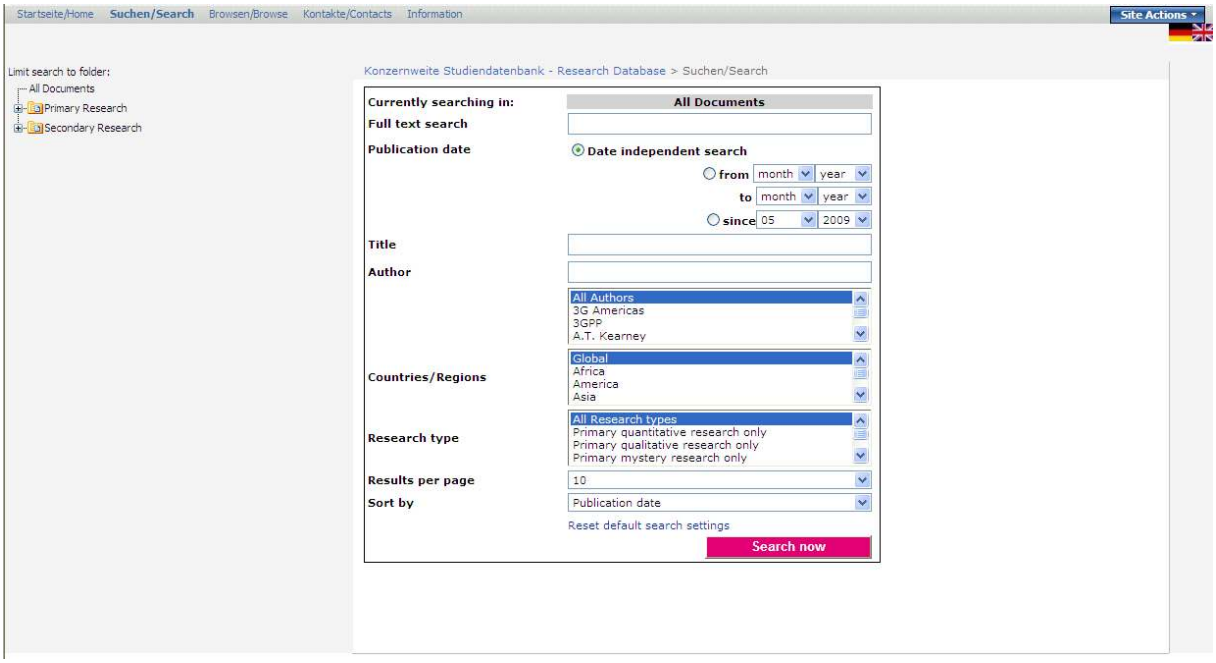

Figure 3-4

3.4.2.2 Implementation

3.4.2.2.1 OnInit()

This function is used to build up the structure of the UI, and attach the Event Handlers when pressing buttons to the UI. Depending on the Session variable *Language*, passed down from the starting page, the labels for the lists and all text is displayed in English, or German. This function also calls to several auxiliary functions, which populate the Lists. These functions are: lstResearchTypeFill (), lstCountriesFill (), lstAuthorFill (), which will be explained further on. The OnInit function creates the content for the Date and Sorting lists. Furthermore this is the part of the Code, where the Tree view for narrowing the search is created. This is found in the region

3.4.2.2.2 TreeViewCustom

First the function clears the left navigation of the page from all previous controls. Then a custom Tree View is created and using the recursive function buildSubNode () populated with every document library and directory level within them. Finally the Tree view is configured for size and styles, and then placed in the left navigation area.

3.4.2.2.3 buildSubNode(TreeNode tn, SPFolderCollection folders,string siteURL)

This is a recursive function for building a Tree view from every directory in the document library. As parameters it takes the parent node which should be extended, the collection of its folders and the URL of the site of the root document library. The function browses through all of the folders in the folder collection. For each folder it creates a new tree node. If this folder has subfolders, the function calls itself with the new tree node as first parameter and the folders subfolder collection as second. Once all subfolders have been added, the tree node is added to the parent node.

3.4.2.2.4 lstCountriesFill()

This procedure is used to fill the country selection list box in the UI. First it checks if the UI is displayed in English or German, and depending on that sets the ColumnName variable to one of these two possible values:

- Long Name, if the UI is in English
- Long Name German, if the UI is in German

This variable is used to extract the corresponding country name from the SharePoint list, where the countries are located. Then the procedure impersonates the System Account, and opens the list where the countries are saved to. It fills to SortedList objects with the information from the SharePoint list: one with the Regions, and one with the normal countries. The procedure uses the Sorted List because the countries are automatically sorted this way with their names. Then the procedure loads all entries from the Region's sorted list into the list box. It the searches for

the “Global” Region and places it at the very first place. Finally the procedure loads all other countries into the list box.

3.4.2.2.5 lstAuthorFill()

This procedure populates the author list box. It connects to the KSDB Database and uses the stored procedure “authorFill” to query all the authors. It fills a DataTable with it, and this table is then bound to the list box. Finally the item “All Authors” is added at the first place of the list box

3.4.2.2.6 lstResearchTypeFill()

This procedure fills the Research type list box. For the time being all connection types between target groups and methodologies are entered manually.

3.4.2.2.7 btnReset_Click(object sender, EventArgs e)

This function resets all fields in the UI to their standard.

3.4.2.2.8 langDe_Click(object sender, ImageClickEventArgs e)

This function sets all labels and text in the Page to German

3.4.2.2.9 langEng_Click(object sender, ImageClickEventArgs e)

This function sets all labels and text in the Page to English.

3.4.2.2.10 QueryBuilder(string FreeText, string Title, string Author, int[] Countries, int[] Authors,int[] ResearchTypes)

This function is the heart of the Web Part, as it is responsible for the building of a correct and functioning Enterprise Search Query for the Search Engine to process. Because we are using a lot of wildcard detection and full-text indexing, the FullTextSQLQuery is the only way to get our search results. It functions similar to a standard SQL Search statement. This function is used to build one, depending on the choices the User makes in the UI.

The beginning of each SQL Statement is the same, namely:

```
SELECT Rank,Size,FileExtension,PublicationDate,path,Title, Authors, Countries, Methodology, TargetGroup FROM SCOPE() WHERE "scope" = 'All Sites' AND ContentType = 'Document'
```

The "scope" sets the target where the search engine will search for documents, and the ContentType designates the types of items to be included in the search results. From here additional options to the query are added according to the UI settings.

If the user selects a directory from the left Treeview different from 'All Documents', the following is added to the statement: `AND CONTAINS(path,'\"" + path + "*\"')` where the path variable stores the selected directories full URL.

If the user added some text into the full text textbox, the following is added: `AND CONTAINS('\"" + FreeText + "\"')`, where the FreeText variable contains the text entered.

If the user checked ‚Date independent search' nothing is added, otherwise the functions checks if a ‚from...to' option or the ‚since' option is checked.

- if the date checked is 'from...to' the following statement is added: `AND ((NOT PublicationDate >= '" + stringTo + "') AND (PublicationDate >= '" + stringFrom + "'))`, where stringTo is the 'to date' and stringFrom is the 'from date'
- if the date checked is 'since', the following is added: `AND PublicationDate >= " + stringSince.ToString()`, where stringSince is the Date entered by the user

Additionally, if ‚include Archive' is not checked, this constraint is also added: `AND PublicationDate >= " + archivestring + " "`, where archivestring is a variable that is 3 years before the current date. This constraint is needed, because if 'include Archive' is not checked, the user doesn't want to see documents older then 3 years.

If the user entered text in the Title text box, the function decides if the title contains the '*' wildcard.

- if it does, this is added: `AND CONTAINS(Title,'\"" + Title + "\"')` to ensure a full-text search
- else the added code is: `AND CONTAINS(Title,'" + Title + "')`

If text is added to the author text box, the function searches for all corresponding authors and adds them to the statement:

`AND (CONTAINS(Authors,'\"" + authornames[0] + "\"')` for the first author, and `OR CONTAINS(Authors,'\"" + authornames[i] + "\"')` for every one beside the first.

Then the function checks the selections in the author list box. If other then 'All authors is selected', it adds the authors to the statement. First it checks if additional authors have been added using the text box. If not, it adds the first selected author from the list via `AND (CONTAINS(Authors,'\"" + lstAuthor.Items[Authors[0]].Text + "\"')`. If there are more each of them is added via `OR CONTAINS(Authors,'\"" + lstAuthor.Items[Authors[i]].Text + "\"')`. If the text box has been used, the entry for the first author in the list changes to `OR CONTAINS(Authors,'\"" + lstAuthor.Items[Authors[0]].Text + "\"')`. Finally missing parentheses are set.

The next list to check is the country list. If the 'Global' entry is selected, nothing is added. Else the first country is added in this statement: `AND (CONTAINS(Countries,'\"" + lstCountries.Items[Countries[0]].Value + "\"')` for every other country the statement used is `OR CONTAINS(Countries,'\"" + lstCountries.Items[Countries[i]].Value + "\"')`, finally the 'Global' is added via `OR CONTAINS(Countries,'\"Global\"'))`

Finally the entries in the research type list box are analyzed. If not 'All research types' is selected, the selected items are taken one-by-one and processed. Each item can contain a methodology and a target group. These are extracted, and added. If only a methodology is given, the statement used is `AND CONTAINS(Methodology,'\"" + split[0] + "*\"')`. If there are target groups as well, the statement is extended with `AND (CONTAINS(TargetGroup,'\"" + split[1] + "\"')`". Additionally, if the target group is 'B2B' or 'B2C' `OR CONTAINS(TargetGroup,'\"BCB\"')` is added, to enclose documents which belong to both target groups. Using this logic, if the target group is 'BCB', both 'B2B and 'B2C' are added via `OR CONTAINS(TargetGroup,'\"B2B\"') OR CONTAINS(TargetGroup,'\"B2C\"')`

For every selected item after the first `OR CONTAINS(Methodology,'\"" + split[0] + "*\"')` and `OR CONTAINS(TargetGroup,'\"" + split[1] + "\"')` are used. Finally missing parentheses are added.

3.4.3 Search Result Web Part

3.4.3.1 Preface

The Search Result Web Part takes the created search query from the Search Web Part, executes the query and displays the results as a GridView for the user on a separate page. To achieve this, the results are created in an embedded class, and forwarded to an ObjectDataSource, which is the used as the DataSource of the GridView. Figure 3-5 shows how the Search Result will be presented to the end user.

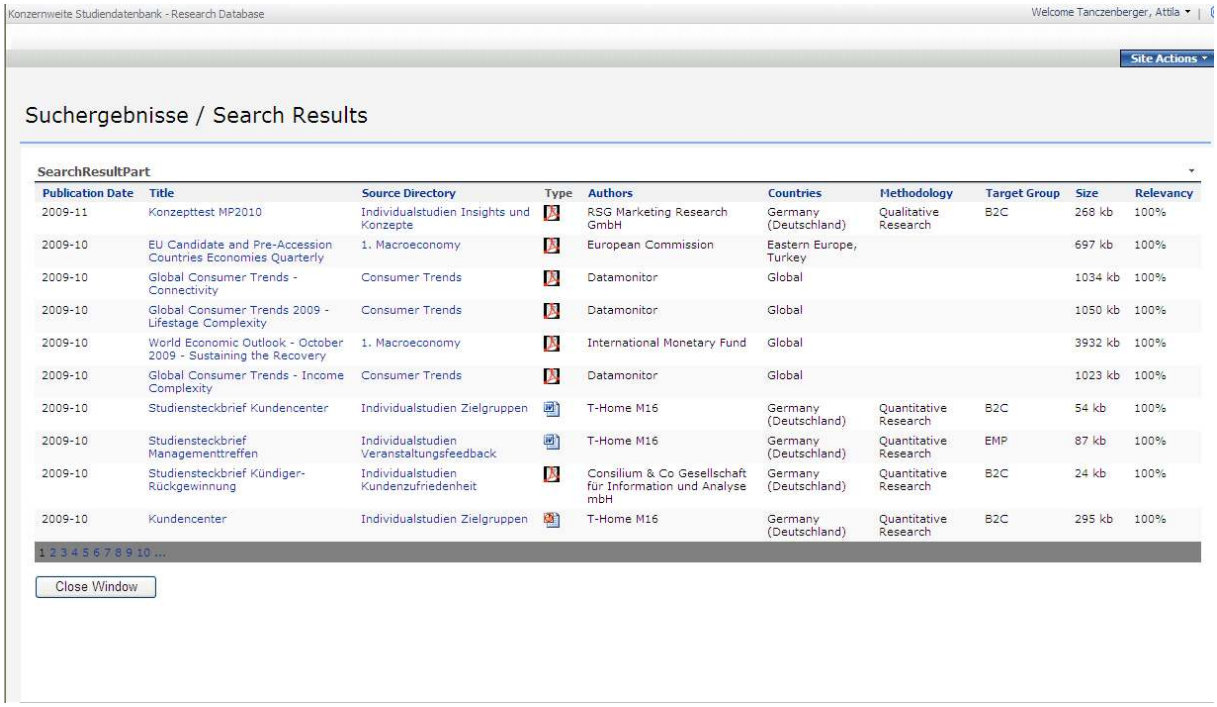

Figure 3-5

3.4.3.2 Implementation

3.4.3.2.1 CreateDataView(string chk1, string param, string sortColumn, string Language, string siteUrl, string pagesize)

Using the *Language* variable, the function sets the *ColumnName* variable to

- “Long Name” if its value is EN
- “Long Name German” if its value is DE

Then two SortedList objects are created from the country list to be used in the result table. The function then takes the query statement, which is supplied in the *param* variable and executes an Enterprise SQL query, then stores the result in a ResultTableCollection. What we need are the relevant results, which we store in a ResultTable. The function then takes this result table, and copies it into a DataTable for further expansion. After this the function expands the table with 6 new colums, which are:

- “DocIcon” for the document icon
- “CountryList”, which is used to translate the country codes to the full country names
- “FileSize” to calculate the file size from bytes to kilobytes
- “Relevancy” to display the Relevancy of the result in percentage
- “dirName” to add a link to the directory where the document is located
- “dirHeader” to store the name of the directory where the document resides

Then the function cycles through all the rows of the data table, and fills in these columns. First to fill the “FileSize” column, the “Size” column is divided by 1024 and is written in the “FileSize” column. To calculate the “Relevancy”, the “Rank” Column is divided by 1000 and multiplied by 100, to get the percentage. To create the url for the document icon, we take the extension of the document, and form the URI like this: `queryDataTable.Rows[i]["DocIcon"] = site.Url + "/images/ic" + queryDataTable.Rows[i]["FileExtension"] + ".gif"`, for every document has an image, which is located in the images folder of the root site. The file is called *ic<extension>.gif*. Before it we put the URL of the root site, which is contained in *site.URL*. To fill in the “CountryList” we split up the original “Countries” field, extract the codes, and search them in the SortedLists. If found, we add them to the “CountryList” column. We extract the “dirHeader” from the second string taken from end of the “path” property, and the “dirName” is built together the same way, we only remove the last string of the “path” property along the “/” characters.

3.4.3.2.2 OnInit()

The function takes the Session parameters received from the Search Web Part and stores them to create the ObjectDataSource. Depending on the *Language* session variable the Web Part configures a Boolean variable to German or English. Then the GridView is configured. Each of its column receives the name of its DataField from the ObjectDataSource, as well as their Header and SortExpression. Sort expressions are responsible for the column sorting in the results. The Headers are configured by using the Boolean value and set corresponding to it to English or German. After this the GridView is built together from its Columns and placed on the page. The function also

removes unnecessary controls from the Page, leaving only the Result GridView. Then an instance of the ObjectDataSource is created. As its SelectMethod, the method which builds up the DataSource we select the CreateDataView. We add the extracted parameters to the DataSource as SelectParameters. After the instance has been created we add it to the page's controls and set it as the DataSource of the GridView.

4 Summary

This paper tried to give a brief overview of the SharePoint Technologies, as well as give an example of the customization possibilities of said application. What I tried to show is that this technology is notably capable of tackling today's business needs of a unified and centralized collaboration platform. It can be used to create a workspace in the company intranet, and every employee is able to use it with the smallest amount of training necessary. It is vastly customizable, and even the out-of-the-box features allow a great deal of scaling and tailoring for every size of company to meet their requirements. The SharePoint line of products is offered in different versions, and it is mostly assured that every company not matter the size will find a solution set that suits its needs. Although the scope of the paper didn't allow a very deep dissection of the product line, it is handy as an overview to anyone, who is just trying to get familiar with the technology.

The first part of the paper focused on the technology itself, giving a brief overview of the most notable and for business most important properties. Although it might seem shallow, as an administrator or user, there is not much more to know about it. It is one of the greatest pros of the technology that everyday users, and even the Administrator responsible for the running of the site does not need to be bogged down by problems or previous knowledge of any of the involved elements, besides the ones that are normally the working environments of a server administrator. Even rudimentary customization and setting can be done by only using the UI, so the normal users never have to look behind the flashy WebPages or UI driven menus.

In the second part of the paper I tried to model a project as a sample for real world applicability of the product line. I tried to emphasize that the default settings might not suit everyone, but the room SharePoint leaves for customization is immense. Here I tried to show SharePoint from the other perspective, the one of the programmer and web developer. As a developer it is a challenge to be in contact with so many programming styles and technologies combined into one application, but thanks to a well designed SDK and the amount of coverage SharePoint has received in these two years it is not impossible to get on par with the product line. There is a clear cut line, where the usage of the Object model ends, and the work for the web designer begins, and so it is not likely that the developer will have to take over the work of a web designer, or vice versa.

But as with any technology, it is not without its flaws, and despite the scalability and the abundance of tutorials and programming guides out there, sometimes the developer can pretty

much hit rock bottom while customizing SharePoint. The shear size and complexity of it requires a lot of different technologies to work in tandem, and for the developer it brings out the need to be knowledgeable about a lot of them, to successfully handle any requests the customer might throw at him. Although built around ASP.NET, SharePoint utilizes a host of other functionalities as well, for example SQL for the database layer, or XML based advanced mark-up languages.
This huge amount of needed knowledge however has its pros as well. If one looks on the internet, a lot of smaller companies have evolved around providing third party extensions to the SharePoint 2007, filling the gaps the default product might not cover, and giving other companies, who only wish to utilize this system but not develop for it, the opportunity to additionally purchase any component that they need.

For the last two years I've been tasked with many different parts of web development, and I have to say it is still a challenge sometimes to get a request done, even after so much accumulated experience. Albeit it is a rewarding challenge, as I see that the future of collaboration workspaces is the same as SharePoint is going to take. Even as I wrote this paper, the new version in accordance with Office 14 is going to be released, with a host of new features and probabilities, to make this product line even more appealing to business partners everywhere. Overall I do feel lucky of being tasked to get in contact with this technology, and I hope that this paper will shed some light on the Microsoft SharePoint Product line and its possibilities.

5 Bibliography

5.1 Books

Michael Noel – Colin Spence: SharePoint 2007 Unleashed, 2007 Sams Publishing

Kevin Hoffman – Robert Foster: SharePoint 2007 Development Unleashed, 2007 Sams Publishing

Patrick Tisseghem – Lars Fastrup: Inside the Index and Search Engines: Microsoft Office Sharepoint 2007, 2008 Microsoft Press

5.2 Web Articles

http://www.c-sharpcorner.com/UploadFile/Chandresh.P/EventHandler04292009070324AM/EventHandler.aspx - Implementing Event Handler in SharePoint, Chandresh Pansuriya, 2009.04.29

6 Acknowledgements

I would like to offer my thanks to Jagó Jácint and Márton Ágnes, who have taken the task of guiding me through the planning and realisation of this paper. Furthermore I'd like to thank all my colleagues, with whom I have worked together these last two years implementing and learning the ways of SharePoint Development. Last, but not least I would like to thank all my friends who gave me the courage and resolve to write up this paper.