Heriot Watt University

MSc IT

Graduation Project

Osama M. Saied Mortada

101668699

Supervisor S. Priya

August 2011
ABSTRACT:

With the growing demand by organisations to automate and standardise the daily paper-work tasks and organise their internal shared documents, the demand of systems that simplify these tasks became an essential requirement for most companies. Having such a system based on the Web technology will allow remote users to externally access the portal from outside the organisation even if they use different operating systems. Internal users will also benefit from having the portal based on the web since they can also use their personal laptops or smart phones to access the portal. Such systems must abide strictly to organisations’ requirements and policy by allowing administrators to have full control over the system in terms of workflow and authentication. Also allowing the growth of new employees to join the portal without requiring any extra licenses is an essential feature. In small companies with one or two servers available, dedicating a server to host such application might not be possible; allowing a low cost plan for the portal will solve many deployment issues. The Web-based Office Automation Portal will be developed considering all previous points in order to achieve a complete and customisable solution for small to medium sized companies.
ACKNOWLEDGEMENT

I would like to thank Mrs Priya for her great support and willingness to answer any theoretical inquiries and questions I had. Her careful advice and guidance has supported me a lot throughout my entire project.

Many thanks for Prof. Stephen Gill for teaching me during the Master’s Degree period. I benefited a lot from him in the Database and the Advanced Software Engineering courses in doing this project.

I would like also to thank Mr Mohamed Rafeeq for helping me in the testing phase of the portal. He spent many hours in testing and reporting errors which made the portal more stable and reliable.

Last but not least, many thanks and love to my family and friends for their unconditional love and support and for making this project happen.
STATEMENT OF NON-PLAGIARISM

I, Osama Mortada confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed:

Date:
Table of Contents

Chapter 1: Introduction ........................................................................................................... 10
  1.1 Web Office Automation Portal (WOAP) ................................................................. 10
  1.2 Benefits of WOAP for multiple companies......................................................... 10
  1.3 Problem Statement ............................................................................................ 11
  1.4 Proposed Solution ............................................................................................. 12
  1.5 Objectives ........................................................................................................ 13

Chapter 2: Literature Review ............................................................................................... 14
  2.1 The Basics of Office Automation ........................................................................ 15
    2.1.1 Data Storage and Manipulation ..................................................................... 15
    2.1.2 Data Exchange .......................................................................................... 15
    2.1.3 Data Management .................................................................................... 15
  2.2 Office Automation Workplace ............................................................................... 16
  2.3 Web-Based Office Automation ............................................................................ 16
    2.3.1 Off The Shelf Office Automation Solutions ............................................... 17
    2.3.2 Custom-made Office Automation Solutions ............................................... 17
  2.4 Web Programming Languages Review ............................................................... 18
    2.4.1 Which scripting language is recommended to be used for this project? ...... 18
    2.4.2 PHP ....................................................................................................... 19
    2.4.3 ASP.Net (Active Server Page) .................................................................... 21
  2.5 RDBMS (Relational Database Management System) ........................................... 25
    2.5.1 Introduction ............................................................................................ 25
    2.5.2 Comparison between MS SQL & MySQL ................................................. 25
    2.5.3 SQL Stored Procedures VS Embedded SQL Statements ......................... 26
    2.5.4 SQL Injection: ....................................................................................... 26
  2.6 HTML & CSS ...................................................................................................... 28
  2.7 Rich User-Interface Tools .................................................................................... 29
    2.7.1 AJAX (Asynchronous JavaScript and XML): ........................................... 29
    2.7.2 3rd Party Tools: ..................................................................................... 29
  2.8 Possible Websites Structures .............................................................................. 30
    2.8.1 Linear Structure: .................................................................................... 30
    2.8.2 The Hierarchical Structure (Webbed): .................................................. 30
    2.8.3 The Spoke-and-Hub Structure: ............................................................... 31
    2.8.4 The Full Web Design Approach: ............................................................. 31
  2.9 Planning the Portal in Stages ............................................................................... 32
  2.10 Development Model ....................................................................................... 33
Chapter 3: Requirements Analysis ................................................................. 35
3.1 Current Business Process ........................................................................ 35
3.2 Deliverables ............................................................................................. 37
   3.2.1 Modules: ....................................................................................... 37
   3.2.2 User interface: ............................................................................... 38

Chapter 4: System Architecture & Design ...................................................... 39
4.1 Main Considerations: ............................................................................... 40
4.2 Multi-Tier Design: ................................................................................... 41
   4.2.1 Presentation Layer .......................................................................... 42
   4.2.2 Business Logic Layer (Database Access & Utilities classes) .......... 43
   4.2.3 Data Layer ...................................................................................... 43

Chapter 5: UML (ER, Activity, Use Case & Class Diagrams) ....................... 44
5.1 Entity Relationship Diagram (Database Design): .................................... 44
5.2 Business Logic Layer Design ................................................................. 46
   5.2.1 Activity Diagram (Authentication System): .................................... 46
   5.2.2 Use Case Diagram (Login System): ............................................... 47
   5.2.3 Employee-Related Classes: ............................................................. 48
   5.2.4 Domain-Related Classes: ................................................................. 49
   5.2.5 Requests Module: .......................................................................... 50
   5.2.6 Memos Module (including the GUI class diagram): ...................... 51
   5.2.7 Events Module: ............................................................................. 51
   5.2.8 Task Assigning Module: ................................................................. 52
   5.2.9 Messages Module: .......................................................................... 53
   5.2.10 Projects Module: .......................................................................... 53
   5.2.11 News Module: ............................................................................... 54
   5.2.12 Meeting Planner Module: .............................................................. 54
   5.2.13 Helper Classes: ............................................................................ 55
5.3 Presentation Layer: .................................................................56
  5.3.1 Certified 3rd Party Component: ........................................56
  5.3.2 AJAX Controls: .............................................................57

Chapter 6: Implementation & Challenges ...........................................58
  6.1 Choosing a design strategy for the whole portal: ......................58
  6.2 Multi-domain support: ..........................................................58
  6.3 Domain-Modules Registration System: ....................................60
  6.4 Dynamic Homepage Panels: ..................................................61
  6.5 Changing the interface (look & feel) dynamically: ....................62
  6.6 Users’ Authentication System: ...............................................63
  6.7 Dynamic Navigation Menu (based on user’s permissions): ..........65
  6.8 Facebook-like messaging system ..........................................67
  6.9 Meeting Planner Module .....................................................71
  6.10 Requests & Ticketing Module: .............................................76
  6.11 Leave Request with Work-Flow: ...........................................77
  6.12 Memos with Work-Flow Module: ........................................79
  6.13 Task Assigning System: .....................................................80
  6.15 Notifications System: ........................................................82
  6.16 Emailing System: .............................................................83
  6.17 Error Detection: ...............................................................85
    6.17.1 Method-Level Error Detection: ......................................85
    6.17.2 Application-Level Error Detection: ................................86

Chapter 7: Deployment & Testing .....................................................87
  7.1 Deployment: .......................................................................87
  7.2 Testing: ............................................................................88
    7.2.1 Tasks Module: ................................................................88
    7.2.2 Memos Module: ............................................................89
    7.2.3 Approving & Posting Memos: ..........................................89
    7.2.4 Requesting new meeting reservation: ..............................90
    7.2.5 Event Module: ..............................................................91
    7.2.6 Viewing Shared Files .....................................................92
    7.2.7 Uploading New Files & Creating New Subfolders ................92
    7.2.8 Updating Employee’s Profile ..........................................93
    7.2.9 Change Company’s Turnover Values ..............................93
    7.2.10 Help Desk Request .......................................................94
7.2.11 Purchase Request ........................................................................................................... 95
7.2.12 Payment Request ............................................................................................................. 95
7.2.13 Defining Groups (within domains) .................................................................................... 96
7.2.14 Defining Branches (within domains) ............................................................................... 96
7.2.15 Projects Module: ............................................................................................................. 97
7.2.16 Managing Domains Notifications .................................................................................. 98
7.2.17 Managing Employees’ Permissions ............................................................................... 98
7.2.18 Portal Admins’ Modules ................................................................................................. 99

Chapter 8: Evaluation ............................................................................................................... 100
8.1 Managers Feedback Form .................................................................................................. 100
8.2 Employees Feedback Form ................................................................................................ 101
8.3 Satisfaction Levels: .......................................................................................................... 102
8.4 The Overall opinion about the portal ............................................................................... 103
8.5 Key users’ opinions: ......................................................................................................... 103

Chapter 9: Professional, legal and ethical issues ..................................................................... 104

Chapter 10: Project Plan .......................................................................................................... 105
10.1 Schedule: ......................................................................................................................... 105
10.2 Gantt chart ....................................................................................................................... 106

Chapter 11: Risk Management ................................................................................................. 107

Chapter 12: Future Improvements ............................................................................................ 108
12.1 Integration with other software such as ERP: ................................................................. 108
12.2 Multi-Lingual Support: .................................................................................................... 108
12.3 Mobile Version: .............................................................................................................. 108

Chapter 13: Conclusion ............................................................................................................. 109

References ................................................................................................................................ 110
### Table of Figures:

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Web Technologies</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>The Linear Structure (Lowery, 2009)</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>The Hierarchical Structure (Lowery, 2009)</td>
<td>30</td>
</tr>
<tr>
<td>4</td>
<td>The Hub Structure (Lowery, 2009)</td>
<td>31</td>
</tr>
<tr>
<td>5</td>
<td>The Full Web Structure (Lowery, 2009)</td>
<td>31</td>
</tr>
<tr>
<td>6</td>
<td>Website Development Stages (Website Structure – What is Best?, 2010)</td>
<td>32</td>
</tr>
<tr>
<td>7</td>
<td>Waterfall Model</td>
<td>33</td>
</tr>
<tr>
<td>8</td>
<td>Agile Model</td>
<td>33</td>
</tr>
<tr>
<td>9</td>
<td>Agile Processing Model (Coutinho, 2009)</td>
<td>33</td>
</tr>
<tr>
<td>10</td>
<td>Prototype design of the portal</td>
<td>38</td>
</tr>
<tr>
<td>11</td>
<td>Portal High-Level Architecture</td>
<td>39</td>
</tr>
<tr>
<td>12</td>
<td>Layered design with a database class (MacDonald, Freeman and Mario, 2010)</td>
<td>41</td>
</tr>
<tr>
<td>13</td>
<td>Snapshot from Visual Studio that shows all projects included in our solution</td>
<td>42</td>
</tr>
<tr>
<td>14</td>
<td>ER Diagram</td>
<td>44</td>
</tr>
<tr>
<td>15</td>
<td>Activity Diagram</td>
<td>46</td>
</tr>
<tr>
<td>16</td>
<td>Use Case Diagram</td>
<td>47</td>
</tr>
<tr>
<td>17</td>
<td>Class Diagram - Employees Class</td>
<td>48</td>
</tr>
<tr>
<td>18</td>
<td>Class Diagram - Domain-related Classes</td>
<td>49</td>
</tr>
<tr>
<td>19</td>
<td>Class Diagram - Requests Module Class</td>
<td>50</td>
</tr>
<tr>
<td>20</td>
<td>Class Diagram - Leave Request Class</td>
<td>50</td>
</tr>
<tr>
<td>21</td>
<td>Class Diagram - Memos Module Class</td>
<td>51</td>
</tr>
<tr>
<td>22</td>
<td>Class Diagram - Events Module Class</td>
<td>51</td>
</tr>
<tr>
<td>23</td>
<td>Class Diagram - Task Assigning Class</td>
<td>52</td>
</tr>
<tr>
<td>24</td>
<td>Class Diagram - Messages Module Class</td>
<td>53</td>
</tr>
<tr>
<td>25</td>
<td>Class Diagram - Projects Module Class</td>
<td>53</td>
</tr>
<tr>
<td>26</td>
<td>Class Diagram - News Module Class</td>
<td>54</td>
</tr>
<tr>
<td>27</td>
<td>Class Diagram - Meeting Planner Module</td>
<td>54</td>
</tr>
<tr>
<td>28</td>
<td>Class Diagram - Helper Classes</td>
<td>55</td>
</tr>
<tr>
<td>29</td>
<td>Main Login Page for Domains</td>
<td>62</td>
</tr>
<tr>
<td>30</td>
<td>Blue Style Sheet</td>
<td>62</td>
</tr>
<tr>
<td>31</td>
<td>Red Style Sheet</td>
<td>62</td>
</tr>
<tr>
<td>32</td>
<td>Composing a new message window</td>
<td>68</td>
</tr>
<tr>
<td>33</td>
<td>Snapshot of replying to messages window</td>
<td>69</td>
</tr>
</tbody>
</table>
Figure 34 A Leave request made by an employee .......................................................... 77
Figure 35: Managers Feedback Form ................................................................. 100
Figure 36 Managers Feedback form ................................................................. 101
Figure 37 Employees Feedback form .............................................................. 101
Figure 38: Satisfaction Level (Employees Module) .............................................. 102
Figure 39: Satisfaction Level (Admin Modules) .................................................. 102
Figure 40: Satisfaction Level (General Consideration) ........................................ 102
Figure 41: The Overall opinion about the portal .................................................. 103
Figure 42 Project Plan - Gantt Chart ................................................................. 106
Chapter 1: Introduction

1.1 Web Office Automation Portal (WOAP)

The Web Office Automation Portal provides an internal communication bridge between employees within an organisation in order to automate most of the paper-based work. The portal should also allow managers and administrators to configure a work-flow system that abides to the organisation’s policy.

This system is also called Intranet; an Intranet is an internal system which allows information to be posted and available for everyone within the enterprise. Its main aim is to communicate employees inside their organisations. (Boyle and Lloyd, 1998).

This project provides a low-cost solution compared to other products in the market, it does not require any dedicated servers or licenses and the best of all it allows companies to tailor modules according to their needs.

The requirements analysis of this portal in terms of work-flow, functionalities, forms etc. has been thoroughly made in a local company based in UAE. Each module has been studied and designed carefully in order to ensure accurate results.

The portal is centrally hosted on the internet which allows more than one company to use it. It provides an option to enable/disable specific depending on companies’ needs.

1.2 Benefits of WOAP for multiple companies

Most of the modules implemented in this portal are standardised in order to be suitable for more than one company. These modules can help in automating most of the daily paper-based tasks.

In addition to the standard modules, custom modules are also available upon request. For example, the case study company has requested 2 custom modules, one for meeting room reservations which coordinates the reservation of the meeting rooms, and the second module that calculates and visualises (using charts) the current company’s profit (budgeted vs. actual).

Employees and managers can rely on this portal in many areas of their daily work. For example, managers can post information such as news, memos or new events whereas normal employees can view, comment and share the information.
1.3 Problem Statement

Currently, many companies use paper based solutions and sometimes small independent systems (Word, Excel or Outlook) in order to complete their internal daily tasks. In most cases it’s important to formalise, force work-flow, integrate and log such tasks which are difficult to achieve with unlinked or independent systems due to the difficulties of formalising or integrating such systems.

Below are some of the common required tasks by most companies:

1. Document-Expiry Management: Managing expiries and renewals of documents such as passports, visa, labour cards etc. is a major task within any company (HR Department). If the company manually checks these documents, it would be a very long time consuming task and sometimes accurate results are not guaranteed.

2. Posting HR memos: It’s an important task within any company, but currently many companies use Word or Outlook to write memos. This doesn’t guarantee proper work-flow and also would make it difficult to employees to keep track of old memos.

3. Events planning: many companies need to keep track of important upcoming events, sometimes these events overlap in dates or are held at different cities or countries. Managing such task manually doesn’t guarantee proper planning by all employees. Sometimes due to the lack of information exchanging, some employees will not be aware of new events especially in companies with a large number of employees.

4. Assigning Tasks: If a company does not use a system which automates and tracks the task assigning process, it would be very difficult for managers and supervisors to keep track of each assigned task, the due date, and its status.

5. Ticketing & Requesting System: This includes many types of requests such as help desk, leave, payment, purchase and printing requests. Normally employees send emails for such requests with no option to track or comment on the status of their request.

6. Archiving and organising company’s documents: Documents such as presentations, logos, trading licenses, letter heads, template emails, certificates, forms etc. are usually required to be accessible by employees within companies. Storing such documents on local systems or unsorted on a local server doesn’t allow all employees (local and remote) to easily access these files.

7. Posting company’s news: Some companies especially the ones which deal with projects or tenders need to post their latest news and share it with their employees. Updating, sorting and keeping news accessible for employees is sometimes not an easy task. If news is posted directly on the company’s website, managers will be restricted to post some confidential news. And if it is distributed using emails, it would be difficult for employees to easily catch the latest news or the most important ones.
8. Out of office schedule: when an employee is about to leave for a vacation or a business trip, it’s important to notify or post such information to all employees, pending work that needs to be handed to other available employees, detecting conflict leaves within the same department, and sometimes employees in charge should be specified prior to the leave. For large companies, it’s very difficult to maintain such a schedule (planning to leave and currently on leave schedule) due to the vast number of employees within each department who applies for leave to HR independently and not through a centralised system.

Note: not all processes are included in the above list; the aim is only to clarify the nature of the problem which applies to many other processes.

1.4 Proposed Solution

Having a platform that links most of the internal daily tasks required by admins and employees in one solution would allow employees to easily understand the company’s processes, work-flow and to organise their daily work.

Such a system should integrate all tasks in one solution; this would enable managers and administrators to log and view activities for further decision and planning.

The system should also be accessible for all users (local and remote) in order to fit all employees and their situations. Therefore, a secure communication method should be used with an option to encrypt all critical information such as passwords.

It would also be very beneficial to have a system which could serve multiple domains (companies), this means to host the solution centrally on the cloud (internet) on a dedicated server which could serve all domains. This would simplify or almost eliminate any requirements to set up a new company and would also simplify fixing any bugs or updating the portal.

Finally, the portal should be user friendly and easy to use since employees will be using it on a daily basis and the friendly interface would encourage them to use it.
1.5 Objectives

1. To automate many of the daily tasks required by most businesses (discussed earlier). It should also act as a tool used by all employees and managers.

2. Multi-domain support; which means a portal that can serve multiple companies directly without any modifications. Once a company registers in the portal, admins should be able to define groups, branches, customise the look and feel, define work-flow, and manage permissions and notifications. This simplifies simulating the actual paper-based work-flow.

3. Enabling employees to easily create new accounts: this would speed up the set up process where employees can directly specify their personal details, upload their photos and documents copies (passport, visa etc.)

4. Critical information such as passwords should be secured and encrypted.

5. To allow dynamic and easy management for processes work-flow: managers & admins only should be provided with a control panel where they can control extended permissions.

6. Compatibility with several companies: This will be achieved by developing the portal in a dynamic manner. This will also allow future customisation to be added without affecting any existing modules.

7. Providing a CMS (Content Management System) that enables administrators to manage the portal in terms of security and assigning access privileges & permissions to users in addition to the ability to customise the look and feel.

8. Remote accessibility (outside the office) will also be provided since the portal will be based on WEB technology. Both local and remote users will be able to access the portal in the same manner.

9. Role Management: Each employee will be provided with an authentication level that controls the way they interact with the system.

10. Work-Flow: Each function that requires approval will be assigned to the work-flow system that controls its flow.

11. Privacy & Encryption: All employees’ related information should be safe & secured. Encryption will be used where necessary and accessing employee’s information should be restricted. The portal must follow the current company’s privacy policy.
In this chapter, theoretical and technical definitions of Office Automation Systems will be explained in order to promote the understanding and cycle of what office automation truly entails. Office automation relates to the different computer software and hardware that is used to electronically store, collect, create, edit, and portray office information needed for completing specific tasks. Three basic aspects of office automation systems that count towards a drastic infrastructure change include storage of raw data, data exchange and data management. Throughout these areas, hardware and software are combined to meet the requirements of basic functions. (Henderson, 2000). It’s a minimised version of the Internet inside an organisation. (Boyle and Lloyd, 1998: p4)
2.1 The Basics of Office Automation

2.1.1 Data Storage and Manipulation

Storing data in the terms of office automation usually refers to documents, forms and other records. Word processing packages permit the use of pure text and graphical data. The advantage of word processing over using a typewriter is that changes can be made without retyping the entire document. When changes are made, the file can be sent or printed for a hard copy.

Spreadsheet applications such as Microsoft Excel; allow the editing and processing numeric data. In a spreadsheet application, each value sits in a cell.

Desktop publishing (DTP) applications such Adobe InDesign have opened new horizons for text manipulation. It is the use of the computer and specialised software to create documents for desktop or commercial printing. DTP is the process of using the computer to produce documents such as newsletters, brochures, books, and other publications. Image-handling software is another facet of office automation. Images, or digital pictures, are representations of visual information. Visual information is an important complement to textual information. (Henderson, 2000)

2.1.2 Data Exchange

Exchanging and sharing soft data between employees is an important aspect of office automation systems. Systems that allow information to be shared by many users are also called groupware. Emails, voice mails and fax machine are example for such electronic data exchange. Real-time data exchange could be possible using specialised chatting & audio/video applications, such systems allow online conversation, video and audio streaming. (Henderson, 2000)

2.1.3 Data Management

Data management is essential for simplifying the management of electronic data. Task management, electronic scheduling and reminders systems are used for electronic management. Work groups can gain access to shared information by using an automated office system. (Henderson, 2000)
2.2 Office Automation Workplace

The section below summarises three essential topics that need further evaluation as per Henderson article in Office Automation:

1) Effects of office automation on individuals

This includes managers, engineers, secretaries and all employees involved with providing and sharing information. Two major issues are raised in this prospective; first is the training required to effectively use an office automation system, second is the workplace resistance since change is difficult for some workers.

2) Different tools used in automation today

Applications such as word processing and spreadsheets can provide a good mean for automating some office tasks; however they are not integrated enough in all business aspects and cannot provide a complete office automation system. Special tools are still required to fit into this gap in order to integrate business requirements.

3) How the work place has evolved due to office automation

Office Automation Systems considerations involve the budget, required equipment, training and security of data. However, it increases the workforce by allowing employees to work remotely at the same time with office employees, save time by automating most of daily tasks and allow the integration of different business aspects.

(Henderson, 2000)

2.3 Web-Based Office Automation

“\textit{In the wake of the astonishing growth in the World-Wide Web, many businesses discovered that the tools of the Internet could be used on private networks and that this offered distinct advantages over other methods of communicating within the company.}” (Craig, 1997)

Over the last decade, the web has become a central space for storing and sharing information between different applications in different domains. Information digitisation has been increased tremendously which resulted in an increase in digital information format. (Lively and Glenn, 1996)

Today’s office’s uses technology as a primary way for communication, employees use the web to find and share information. Companies are trying to use latest technologies in order to improve the productivity of their employees. On the other hand, workers expect to find new means that might help simplifying their daily tasks (Ellen, White and Pasewark, 2002)
2.3.1 Off The Shelf Office Automation Solutions

Many Office Automation packages are available in the market. These readymade packages can provide a low cost solution, save time and money. These packages are tested and they allow companies to automate most of the standard office tasks. However, in most cases these packages contain unwanted or extra features which are not required by the company and might lead to users getting frustrated. (Off The Shelf Applications Vs Custom Designed Software Solutions, 2011)

However, most off the shelf solutions lack the support of customisation (tailored needs); different companies might have different requirements and work-flows. They can be useful in some processes within the company but not for all processes and work-flows which might not be unified. Another disadvantage is the lack of supporting future requirements which might come after purchasing the package. They lack the integration with the company’s current system such as ERP or HR software. (Office Automation, n.d.)

2.3.2 Custom-made Office Automation Solutions

Custom-made solutions might provide similar functionalities as the off-shelf packages in terms of the automation of standard tasks. However, since these systems are developed specifically to serve a narrower range of companies, they should be powered with the support of custom needs and work-flow. Tailored office automations solutions should allow clients to fulfil most of their requirements.

A developer should analyse the client’s needs carefully during the design, this would save a significant amount of time later in the development phase. (Off The Shelf Applications Vs Custom Designed Software Solutions, 2011)

Choosing suitable tools to build the system is an important decision that developers should make. These tools include the programming language that fit with the system’s needs, and the database system that will be used to store the data. Developers should also consider a database system that allows the integration with any current database systems such as ERP; this would allow easier expansion of the portal in the future.
Many programming languages are available and used by developers. This section will consider two of the most used web technologies that could be used in building web portals (PHP & ASP.Net).

A complete technical review will be made on both technologies in terms of programming style, security, hosting, database connectivity, compatibility and performance.

2.4.1 Which scripting language is recommended to be used for this project?

There is no right or wrong answer to this question; PHP and ASP.Net are two powerful scripting languages which are widely used by professional developers. Each of these two technologies has large library of classes and APIs which allow developers create advanced and secured web applications.

However, when digging in depth into them, they are completely different languages in terms of coding style, platform, compiling and other measures. According to these measures which will be covered in this chapter, one of these two languages will be chosen to develop this project; the chosen language should fulfil the project’s requirement in the best possible way.
2.4.2 PHP

Preface

“One of the major reasons for PHP’s success as a web scripting language is its origins as a tool to process HTML forms and create web pages.” (David and Trachtenberg, 2002: p11)

PHP stands for Hypertext Pre-processor; it’s a scripting server-side language which is used with HTML to create dynamic web applications. Like any server-side language, PHP commands are executed at the server side then results are sent back to the user’s browser.

PHP is an Open Source technology, which means that it is available for developers at no cost. It’s also a cross-platform so it can be run on different operating systems such as Windows & Unix (IIS & Apache).

With PHP, developers can perform complicated programming tasks with the help of the PHP framework which contains dozens of distinct 3rd-party libraries. They can easily set HTTP headers which could be used to manage cookies, authenticate requests and preform some browser-related tasks such as redirecting the user to a new page. In addition to that, PHP provides connectivity to many databases such ad MySQL, Oracle and MS SQL. (Vasvani, 2009)

One of the best features in PHP is the ability to include the PHP script directly within the HTML code by including the script between PHP tags (covered below). The web server interprets any code between the two PHP tags and as a PHP script.

PHP coding style overview

PHP script should be included between the PHP tags: `<?php your code ?>`.

- Declaring variables:
  ```php
  $intVar = 5;
  $strVar = "hello world";
  ```

- Use of if/else:
  ```php
  if ($count > 0) { do something }
  else { do something else }
  ```

- Use of classes & functions:
  ```php
  class Test
  {
    var $count;
    function myFunction($parm1, $parm2) { your code }
    $returnValue = myFunction(5,25);
  }
  ```
**PHP is not pure OOL (Object Oriented Language):** PHP is an interpreted language. However it can be compiled using third party-tools. One of the major drawbacks of PHP is that it does not allow developers to write a pure object-oriented code. It lacks too many of the OOP standards.

PHP is similar to Perl; variables can be directly used without declaring them. Many data types such as Array, Strings, Integers and DateTime are supported. Functions and Voids are also supported and they could be included in separate files and link these files where needed. This allows reusability of functions and simplifies the debugging process.

**PHP Hosting & Performance:** PHP performs better when hosted on Apache server, although it can run on IIS web server but doesn’t provide the same performance. This is because PHP can be built as an Apache module which makes PHP lightweight and speedy.

Recently, Microsoft has improved the IIS (v7) which can provide a better performance for PHP scripts than the previous IIS version (v6).

**PHP Data Access:** Using PHP, connection to different databases such as MySQL and MS SQL is possible. However using PHP with MySQL provides the perfect combination in terms of compatibility and performance. MS SQL access can be faster with other technologies such as ASP.Net.

**PHP Tools:** Many IDEs (Integrated Development Environment) are available for PHP. Most of them are free to use such as NetBeans & Eclipse. However most of these IDEs require lots of add-ons in order to add support to some libraries and technologies such as the support of AJAX and some rich user interface controls. Till now there is no unified IDE that provides all these features for PHP.

**PHP Documentation:** Documentation is a very important measure when choosing a programming language. A solid documentation might save many hours for developers when trying to find samples or instructions on specific feature in the language.

As mentioned earlier, PHP is an Open Source technology and therefore it’s supported and documented by larger number of the community members. However, some libraries lack proper coverage and sometimes a difference in the documentation may be noticed.
2.4.3  ASP.Net (Active Server Page)

Preface

“When .NET first appeared, it introduced a small avalanche of new technologies. There was a whole new way to write web applications (ASP.NET), a whole new way to connect to databases (ADO.NET), new type safe languages (C# and VB .NET), and a managed runtime (the CLR).” (MacDonald, Freeman and Mario, 2010: p36)

ASP.Net is a Microsoft technology; it has started with a basic version named ASP which could not compete strongly with PHP at that time. Then ASP.Net (v1) was introduced to bring a complete new way of developing web applications. (Sanders, 2009)

ASP.Net is built on the Microsoft .NET Framework which consists of several related technologies, .Net Framework has made a revolution in the way that developers can access databases, build windows application and create dynamic web application. ASP.Net is a part of the .Net Framework.


ASP.Net coding style overview

<asp:TextBox id="myASPText" Text="Hello ASP.NET TextBox" runat="server" />
<asp:CheckBox id="myASPCheck" Text="My CheckBox" runat="server" />

Interact with these controls could be as follows:

myASPText.Text = "New text";
myASPCheck.Text = "Click!";

ASP.Net Tools

Having a reliable IDE for ASP.Net which is Microsoft Visual Studio makes it even easier for beginners to start using ASP.Net, and also this IDE allows experts to save a lot of development time by providing features that automate and generate the code.

ASP.Net Documentation

Documentation is provided by Microsoft directly and it’s sufficient and well organised. Many websites such as asp.net can also provide detailed example of many features on the .Net Framework.
ASP.NET is 100% Object-Oriented Language

“ASP.NET is the most complete platform for web development that’s ever been put together.” (MacDonald, Freeman and Mario, 2010: xxxiv)

With ASP.NET, all the technique of the OOP (object-oriented programming) can be exploited. Classes and interfaces could be reused, extending existing classes with inheritance and bundle useful functionalities in a distributable, compiled component (DLL).

With ASP.Net, developers can write the code behind using C# or Visual Basic which makes it easier for them to choose the language they prefer.

ASP.NET is compiled, not interpreted; the .NET application should pass two phases of compilation: firstly, the code written is compiled into an intermediate language called Microsoft Intermediate Language (MSIL). The second compilation happens just before the page is actually executed. The IL code is compiled into low-level native machine code. This is called just-in-time (JIT) compilation.

The following section shows some of the major aspects of ASP.Net technology (MacDonald, Freeman and Mario, 2010):

- ASP.NET is hosted by the Common Language Runtime (CLR): ASP.NET runs inside the runtime environment of the CLR, all namespaces, applications, and classes is referred to as managed code.

- Type Safety: During the compilation phase of an application, .NET adds information that indicates details such as the available classes, their members, their data types. This extra layer of safety helps in avoiding many low-level errors.

- Multithreading: The CLR provides a pool of threads that various classes can use.

- Structured Error Handling: .NET languages offer structured exception handling. Separate exception handlers can be created to deal with different types of errors and they can also be nested in deeper levels.

- Automatic memory management and garbage collection: The CLR allocates space on the managed heap for that object. This memory doesn’t need to be cleared manually. As soon as the reference to an object goes out of scope or the application ends, the object becomes available for garbage collection. The garbage collector runs periodically inside the CLR, automatically reclaiming unused memory for inaccessible objects. This model saves from the low-level complexities.
ASP.Net web controls

ASP.NET web controls provide a higher level of abstraction and more functionality. A set of ready-made controls allow developers to save time by using them directly in the project (such as TextBox, Label, and Button). ASP.Net includes also some complex rendered controls (such as the Calendar, GridView and TreeView). These controls are programmable which means that developers can interact directly with these controls from the code behind. They are easy to learn and can fit easily with Windows-based application developers who are moving to the world of the Web, because many of the Windows controls property names are similar to ASP.Net Web Controls.

ASP.Net Hosting & Performance

Although an ASP.Net website can run on an Apache server using 3rd party plugins such as Mod_AsppDotNet and mod_mono 0.8, some tuning and workarounds will need to be made to solve incompatibility issues, in addition to the lack of supporting new .Net Framework versions.

IIS & ASP.Net are developed by Microsoft and they work perfectly together without any further add-ons or tuning. With the new IIS7, debugging high-performance applications is better than any other Web server. (Staples, 2007)

The graph below which is obtained from wrensoft.com shows that ASP.Net is currently the fastest scripting technology (Performance benchmarks , n.d.):

![Graph showing ASP.Net performance](wrensoft.com)

**Figure 1 Web Technologies**

ASP.Net Data Access:

ADO.Net (ActiveX Data Object) is a set of data-access APIs and it is part of the .Net Framework. It’s used to access and modify data in an easy and reliable way. ADO.Net provides direct support to MS SQL, and an indirect access to other database providers such as Oracle and MySQL. (ADO.NET Data Providers, n.d.)
ASP.Net Versions & Improvement:

- **ASP.NET 1.0 and 1.1**: It was released in January 2002 and its core idea was a model of web page design called web forms. The web form model is an abstraction that models the page as a combination of objects.

- **ASP.NET 2.0**: ASP.NET 2.0 kept the same core abstraction (the web form model) and concentrated on adding new, higher-level features. Some of these useful features include:
  - Master Pages: Master pages are reusable page templates which can be used to maintain a consistent design in pages (header, menu or footer).
  - Navigation: a framework for defining site maps that describe the logical arrangement of pages in a website.
  - Security and Membership: ASP.NET 2.0: automatically supports storing user credentials, a role-based authorisation, logging in, registering, and retrieving a forgotten password.
  - Data source controls: allows defining how the page interacts with a data source declaratively in the mark-up, rather than writing the equivalent data access code by hand.

- **ASP.NET 3.5**: This version included new features such as LINQ and AJAX.

- **ASP.NET AJAX**: Ajax is programming for a client-side technique that allows the page to call the server and update its content without triggering a complete post back (refresh). Ajax allows creating pages that look like running applications without post back (refreshes). (Smith, 2008)

- **ASP.NET 4**: Its latest version, the most significant features include:
  - Consistent XHTML rendering; allow clean, quirk-free XHTML standard.
  - Updated browser detection: Better-supported browsers include Google Chrome, Internet Explorer 8, Firefox 3.5, Opera 10, Safari 4, and the mobile browsers for the BlackBerry, iPhone, iPod, and Windows Mobile operating system. ASP.NET addresses different browsers’ issues in an intelligent way without the need for any workarounds.
  - Session state compression: this helps in performance improvement.
  - The Chart control: 2d & 3d graphs (including line, bar, curve, area, pie, doughnut, and point charts, complete with features like error bars and Bollinger bands).

- **ASP.NET MVC**: ASP.NET Model-View-Controller, it’s a different way of building web pages than the standard web forms model. The idea is that the application is separated into three logical parts.
  - Model: includes the application-specific business code & data-access logic
  - View: the Presentation layer that will be rendered to HTML
  - Controller: coordinates everything such as user interactions, updating the model, and passing the information to the view.
2.5  RDBMS (Relational Database Management System)

2.5.1  Introduction

As most web developers know, the most commonly used back-end data store technologies are MS SQL (Microsoft) and MySQL (Open Source) due to their ease of use. Fundamentally, both technologies are similar where both allow data storage and retrieval. But of course these two technologies have areas of differences which will be covered in this section.

2.5.2  Comparison between MS SQL & MySQL

**Licensing:** Currently both technologies are available for developers to start using at no cost. Microsoft has launched an Express version of MS SQL which is completely free to use for an unlimited period but with a limited database capacity. MySQL is completely free if used within an open source product, but it’s not free if used within a commercial product with closed-source (General Public License GPL). (Sanders Kaufman, 2003)

**Flexibility:** Both technologies are flexible and could be used with all common protocols such as Transact SQL which is commonly used protocol that provides a set of statements that developers can use to access and alter a database.

MS SQL has an advantage over MySQL in importing & exporting data, it’s easier for database administrators using MS SQL to directly export any table to excel or any other supported formats than MySQL.

In addition, MS SQL has better support for stored procedures (recently MySQL has added this great feature) and also MS SQL has better support for Transactions. (MySQL VS MsSQL – Data Management Comparison, 2010)

**Security:** Both technologies have good security measures but in different manners. In MySQL, the security is done on table-level which means if a user is allowed (or denied) on a specific table this means that this will apply on all columns within that table. In MS SQL, the security could be achieved to a column-level which is far more flexible. For example, a user can be allowed to read some columns within a table, while other columns in the same table are denied for the same user.

**Recovery:** MS SQL is much better in recovery than MySQL, if power fails unexpectedly on MySQL server. This might cause a complete loss and corruption of data (MySQL always assume uninterrupted operation). On the other hand with MS SQL it’s much more resistant to data corruption, data goes in multiple check points to ensure data-consistency at each phase. (Sanders Kaufman, 2003)
2.5.3 SQL Stored Procedures VS Embedded SQL Statements

Stored procedures are small programs written in the SQL server, some of the characteristics of stored procedure are:

- Stored procedures are saved in the databases server itself. This allows the program to split its processing into two parts. The application server will process the main application and the database server handles all stored procedures processing.
- Once a stored procedure is created on the server, it could be called from the application using its name.
- A procedural business-logic can easily be defined within the same stored procedures using multiple SQL statements.
- Improves security by omitting all the database access code from the source programme, thus avoiding the need to grant extra permissions to developers who do not necessarily require accessing the database, the DBA (Database Administrator) will have full control to define the data access code securely, creates the stored procedures and only allow access the created stored procedures.

The use of SQL Stored Procedures to define the data access queries instead of embedding the data access code into the application itself has many advantages such as:

2.5.4 SQL Injection:

“SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker”. (SQL Injection, n.d.)

Attackers usually insert a malicious code within an input field which is used to construct the SQL statement, and during concatenating the SQL query to build the final statement, the unwanted “malicious” query is executed.

For example, if the user is prompted to enter their name which will be used to get their full details, if they enter *Osama*, the basic query could look similar to the following:

```
SELECT * FROM Employees WHERE Name= 'Osama'
```

Assuming that the user enters: *Osama*; drop table Employees--

Then the following query is assembled by the script:

```
SELECT * FROM Employees WHERE Name= 'Osama'; drop table Employees--
```

The semicolon (;) indicates the end of the first query and the start of the second one. The double hyphen (--) indicates that the rest of the current line is a comment and should be ignored. When the server processes this query, it will first select all records from Employees table where Name is Osama. *Then, the table Employees is dropped.* (SQL Injection, n.d.)
How to protect your application from such attack?

In order to avoid SQL injection, all form values entered by users must be carefully checked, especially if the SQL statement is to be constructed dynamically based on the entered values.

The following section describes the coding best practice recommended by Microsoft:

- Use Type-Safe SQL Parameters: Parameters allow data types and length to be validated. Most importantly, the input value is treated as a literal value instead of an executable code.

- Use Parameterised Input with Stored Procedures.

- Never build Transact-SQL statements directly from user input.

- Not validated user input should never be concatenated. String concatenation is the most famous way for SQL injection.

- Check data-types of all entered values and enforce limits (min and max) where possible.

- Binary data, escape sequences, and comment characters should be ignored if found within any input string.

- Implement multiple layers of validation. Client-side validation can help in preventing basic SQL injection and other input mistakes where the next layer (server-side) can make another deeper check for the input.

- Reject input that contains these characters contained between brackets:
  (;) (‘) (--) (/\* ... */)

(SQL Injection, n.d.)
2.6 HTML & CSS

**HTML** stands for Hyper Text Mark-up Language. HTML is the main language for the World Wide Web (WWW). It's consisting of tags `<html>` which comes in pairs of opening and closing tags `<html>…</html>.

Web browsers use HTML to interpret and compose text, images and other objects into visual output. HTML allows many types of objects to be embedded by using the proper tags. Images, Flash clips, JavaScript, and many text styles can be used.

The latest version of HTML (v5) provides new features such as video/audio embedding some special tags for headers and footers. However currently many developers still use HTML version 3 & 4 since HTML 5 is still under development. (When will HTML5 be finished?, 2011)

Styling the HTML document can come in two ways. Embedded styling within the HTML document itself or by using an external document to define the appearance and layout of text and other objects.

CSS is the approved method by W3C for enhancing and formatting web pages. The main concept of CSS, is to separate the HTML (contents) from its styles and the formatting markups. It’s recommended to use CSS for styling HTML documents rather than embedding the styles in the HTML page. This isolation allows designers to maintain a consistent design across many pages by forcing multiple HTML documents to apply styles defined in that CSS file. (Meyer, 2000)

**Benefits of using CSS:**

- Normal users who do not have experience in HTML or CSS can change the page layout by changing some values in the CSS file (colour, font etc.).
- The isolation of content from its formatting provides clear HTML documents that contain contents only. The CSS document manages all styles and the appearance of the HTML document.
- One CSS style sheet can be used to style many pages. This simplifies the update process as once the CSS file is updated, all other HTML pages will be affected.
- CSS files are cached in the browser by default; this can improve the performance when loading pages.

(Meyer, 2000)
2.7 Rich User-Interface Tools

Website’s interface is one of the external key factors for its success. Providing rich user interface to visitors increases the amount of time that users might spend on the website and makes the experience with the website fun and useful.

AJAX is a strong technology used by Google (GoogleMaps) and other large websites for building web rich-user interface application. (Zakas, McPeak and Fawcett, 2007)

2.7.1 AJAX (Asynchronous JavaScript and XML):

It is an intermediate technology that works asynchronously within the web browser. It renders user’s request in a way that avoids unnecessary trips to the server (post backs). Trips to server are normally required to submit requests and receive responses, but with AJAX, requests are saved within the AJAX engine that handles them while allowing interaction with the application asynchronously and independently of any interaction with the server.

(Kyrnin, n.d.)

In addition, a set of AJAX controls (toolkit) are available for free download. These controls work natively within the AJAX platform. They allow developers to include rich user-interfaces, animations, elegant pop-ups and warning messages which are similar in a way to the normal desktop application. (MacDonald, Freeman and Mario, 2010)

AJAX & ASP.Net:

PHP doesn’t have a standard IDE that integrates AJAX plugins and toolkits. In many cases developers still have to write some JavaScript code in order to achieve the required results when using AJAX. (Introduction to Ajax for PHP Web Applications, n.d.)

The new versions of ASP.Net & Visual Studio have complete & native support for AJAX technology. All ASP.Net Web Controls are supported to work within the AJAX framework. Developers can include AJAX controls by simply dragging & dropping them to a Webpage.

2.7.2 3rd Party Tools:

There are many 3rd party companies that provide rich user-interface controls and add-ons for ASP.Net and AJAX. Currently, Component Art & ComponentOne are the strongest providers in the market. They provide powerful controls such as Charting, Spell Check & Editor controls which are not currently available in the standard Web Controls toolkit.
2.8 Possible Websites Structures

Most of professional websites worldwide are made using a simple layout in terms of content, links, images etc. This is for the user to obtain the desired information in an easier manner.

Research proves that internet users don’t exceed 3 clicks on a webpages till they get their information then abandon it to go to another source and continue collecting their information.

Below are four website structures provided by Lowery (Lowery, 2009):

2.8.1 Linear Structure:

The linear structure is applicable to the following cases:

1. Websites that sell a certain product.
2. Instructional websites that give a step by step guide to do tasks. Example; food recipes, writing an essay etc.
3. Websites with excess text such as stories, news etc. This type of websites would be mainly owned by a journalist or a similar person.
4. Websites with embedding keywords that actually link the user to another website containing information about the same topic. An excellent example of embedding websites would be Wikipedia. It has got almost 20% of the keywords used in its articles as embedded keywords. Such websites are very helpful for users that are searching for a specific topic but in really deep detail.

![Figure 2 The Linear Structure (Lowery, 2009)](image)

2.8.2 The Hierarchical Structure (Webbed):

Hierarchical navigational models emerge from top-down designs. These start with one key concept that becomes the home page. From the home page, users branch off to several main pages; if needed, these main pages can, in turn, branch off into many separate pages. Everything flows from the home page; it's very much like a company's organisational chart, with the CEO on top followed by the various company divisions.

The hierarchical Web site, shown in Figure 2, is best known for maintaining a visitors’ sense of place in the site. Some Web designers even depict the treelike structure as a navigation device and include each branch as a link. This enables visitors to quickly retrace their steps, branch by branch, to investigate different routes. This structure is used by many websites such as Facebook, eBay, Amazon, MySpace, Twitter & Wikipedia.

![Figure 3 The Hierarchical Structure (Lowery, 2009)](image)
2.8.3 The Spoke-and-Hub Structure:

The spoke-and-hub model performs very well. The hub is the first home page. The spokes projecting from the centre connect to all the major pages in the site.

This layout permits quick access to any key page in just two jumps - one jump always leading back to the hub/home page and one jump leading in a new direction.

The main drawback in this navigational mode is the required return to the home page.

2.8.4 The Full Web Design Approach:

This is the least structured approach for a Website but it takes the most advantage of the hyperlink capabilities. This navigational model enables every page to connect to every other page.

The full Web design works well for sites that are required to be linked to other sites in many areas (links within paragraphs, external resources, etc...).

The only drawback of using the full Web design model for a site design is that a visitor can really get lost. That’s why a site map is strongly recommended to be used with this model especially for large-scale sites.
2.9 Planning the Portal in Stages

![Website Development Stages](image)

Figure 6 Website Development Stages (Website Structure – What is Best?, 2010)

Well planning is an essential process, not only to cut development time considerably, but to make it far easier to achieve a uniform look and feel for a Web site, making it friendlier and easier to use. Developers must be aware of the considerations when designing a website (Sweeney, 2006)

What is to be delivered? The clearer a website maker is about their message, the more focused the Web site will be. It’s useful to try to state the purpose of the Web site in one sentence.

Who are the stakeholders? Quite often, a site's style is heavily influenced by a clear vision of the site's intended audience. For example, if the audience are mainly children, the design must be colourful, clear & uses large and clear typography.

Bandwidth Consideration: Webmasters must take into consideration the size of the web page, how long it takes to download the page on typical connections? Not all visitors are expected to have a fast internet connection. If including high quality graphics or videos, they shouldn’t be embedded directly on the page, instead users can be allowed to click and pop up a new page for that. A not like “512kbps or higher is recommended” can be included.

Update Mechanism: Sometimes a website can be developed in a short time but later, the updating process will be very time consuming due to lack of content management system planning. On the other hand, other websites might take more time in development but will be easily maintained in the future.

Professional web designers realise that time is an essential factor, it’s essential to stick to the dead line of the project and estimate the required resources accordingly.
2.10 Development Model

2.10.1 Waterfall Model:

In this model, gathering requirements comes at the first phase and accordingly, a proposed design will be made in the next phase. The developer can then start the actual development (coding) and testing. Maintenance will be a continuous process in order to maintain the application as long as it’s live and being used.

*This model is suitable if all requirements are available before starting the development.*

2.10.2 Agile Processing:

In this model, feedback is considered to be the main method of gathering the requirements rather than planning the application at once at the beginning. It is a circular model which means that once a process is complete, another process within the circular loop will start and so on until the project is completed.

One of the main advantages of using the Agile model is the ability to integrate business process with application’s processes. This integration will avoid one process being waiting for other processes, and this will definitely save time and improve the understanding of business needs by the developer and from the other side it will allow businesses to determine the requirements and changes accurately. (Coutinho, 2009)
2.11 Critical Analysis (Fitting our Needs in the Literature Review)

Based on the literature review and the portal requirements, the use of the following technologies/methods is recommended:

- ASP.Net v4, IIS 7, .Net Framework 4, MS SQL, AJAX in addition to some 3rd party controls.
- Agile Processing as the development model: it perfectly suits the nature of the project; it helps starting with the available information and then builds up as the project grows.
- The Hierarchical Website Structure for developing the interface. Some tools such as Photoshop and Illustrator will be used for designing the interface of the portal.

2.11.1 Why ASP.Net?

- It enables developers to write true object-oriented programs similar to Windows-based applications which is written using C++ or C#

- With ASP.Net the code can be separated from the HTML mark-ups using a code behind file; this gives the developer a cleaner place to write the code.

- ASP.Net is faster than PHP and JSP. (Performance benchmarks, n.d.)

- It’s well-supported and enhanced continuously by Microsoft

- ADO.Net provides perfect and native support to the MS SQL databases, XML and other database providers such as Oracle.

2.11.2 IIS with .Net Framework 4:

The company uses Windows based environment. All computers and servers use Windows operating system. IIS is installed by default on all Windows Servers so extra configuration is required.

Installing .Net Framework 4 will allow the use of new features provided by this version and it’s available for free downloading.

2.11.3 Why MS SQL?

It’s completely supported natively by ASP.Net (ADO.Net). In addition to that, the company has the latest version of MS SQL Server installed which currently serves the ERP application.

2.11.4 Why AJAX?

AJAX is considered as an intermediate technology which allows the link between server-side and client-side scripts. It’s completely free and supported directly by ASP.Net and Visual Studio 2010.
Chapter 3: Requirements Analysis

“Office tasks are related to the work of many parts of an organisation. Completing office tasks often requires judgements and making decisions. Understanding the organisation will help you make sound decisions in completing your work.” (Ellen, White and Pasewark, 2002)

The analysis of this portal in terms of work flow, required functions, forms etc. will be studied on-site by digging in depth into different departments in an organisation (or two) in order to understand the actual requirement of each department.

3.1 Current Business Process

Many small & medium sized companies are currently housing a paper based solution to complete their internal daily tasks. The following section highlights some of these tasks in order to realise how this portal will be very beneficial for them:

Assigning Tasks:

Managers can verbally assign tasks or by emails to employees within their teams. Following up with these tasks is done manually, no automated systems are used to check due dates or to monitor these tasks.

Posting HR Memos:

The HR manager writes the memo in any word processor and prints it in order to circulate it to all employees or can send it as an attachment in an email. It’s difficult to keep track of previous memos or search for a specific memo which was posted earlier.

Managing Events:

Many companies require calendars that keep track of upcoming events related to their scope of interests. Sometimes separate calendars are required by each division within a company which has different scope of interests.

Requests & Ticketing System:

Currently, requests are done verbally or using emails. There is no unified system for such requests.
Document-Expiry Management:

The HR personnel regularly checks all passports and other documents such as labour cards, then takes the necessary action if a document is about to expire.

Leave application requests:

1. An employee collects a leave application from the HR department
2. They handwrite the form fields
3. They request an approval from their supervisors
4. If approved, the employee submits the form to the HR department. If declined, a verbal reasoning is given to the employee.

Out of office schedule:

Since leave requests are done manually, the main disadvantage is that employees are not aware of the leave of other employees. An early notification is important in order to allow coordination between employees during the leave period.

Meeting Room Reservation:

First come first serve, there is no reservation system for the meeting hall. Employees can just walk in and start their meetings. Sometimes employees face problems with overlapped appointments where the room is already in use by other employees.

Posting News:

Divisions’ managers or the general manager send an email upon winning any new project or when major events occur. There is no unified platform for organising and saving such events.
3.2 Deliverables

3.2.1 Modules:

The following functions will be delivered by the end of the project. These functions are subject to a work-flow system that will manage all process.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Employees Registration</td>
<td>Authenticated &amp; Encrypted</td>
</tr>
<tr>
<td>Documents Search</td>
<td>for internal files shared by the company</td>
</tr>
<tr>
<td>E-Forms &amp; Requests</td>
<td>Leave application, business card request, IT help desk etc.</td>
</tr>
<tr>
<td>Employees currently on leave or</td>
<td>This schedule will be available for employees in order to plan and</td>
</tr>
<tr>
<td>planning to leave schedule</td>
<td>coordinate during the absence of other employees.</td>
</tr>
<tr>
<td>Documents-Expiry Notification</td>
<td>Admins and authorised users must be notified when a document is about to</td>
</tr>
<tr>
<td></td>
<td>expire.</td>
</tr>
<tr>
<td>Official Memos</td>
<td>HR Department can post memos. Approval is managed by the work-flow system</td>
</tr>
<tr>
<td>Events Management</td>
<td>Exhibitions, gathering, birthday alarms, events etc.</td>
</tr>
<tr>
<td>Task Assigning</td>
<td>Managers can assign tasks to their team, follow-up alarms and reports will</td>
</tr>
<tr>
<td></td>
<td>be available</td>
</tr>
<tr>
<td>Projects List</td>
<td>Current in hand projects, project type might vary according to the business</td>
</tr>
<tr>
<td></td>
<td>type</td>
</tr>
<tr>
<td>Internal Messaging</td>
<td>Facebook-like messaging system that maintains the in/out messages as threads.</td>
</tr>
<tr>
<td>Company News</td>
<td>Allows authenticated users to post news which will then be rotated on the</td>
</tr>
<tr>
<td></td>
<td>homepage.</td>
</tr>
<tr>
<td>Meeting Room Reservation</td>
<td>Coordinating the reservation the meeting room to avoid overlapped meetings.</td>
</tr>
<tr>
<td>Work-Flow System</td>
<td>This module will manage all other modules that require approvals or</td>
</tr>
<tr>
<td></td>
<td>authentications.</td>
</tr>
</tbody>
</table>

Sample work-flow:

- Employee logs in to the portal and fills an e-form
- Request is forwarded to his/her supervisor
- If rejected, a reason is provided and an email is sent to the employee
- If accepted, it is forwarded to the HR who continues the process in the portal, prints and signs
3.2.2 User interface:

Below is a prototype of the user interface design. It is noticeable that the body of the page is further divided into section which represents different modules on the same page.

![Prototype design of the portal](image_url)
Chapter 4: System Architecture & Design

The above figure shows how the portal is used by more than one company (domain). Each company is assigned to some modules as per its requirements, and sometimes, custom modules could be tailored to one or more companies according to their needs.

Each domain has one or more administrators. Administrators have full rights to configure the portal, assign permissions to employees and view reports.

Domains (companies) can use any modules offered by the portal, but domains have no link between each other. Each domain has its own data secured and isolated completely from other domains.
4.1 Main Considerations:

Before digging into the system’s architecture & design, the following section demonstrates a high-level explanation of the techniques used in building up this project:

1. The portal is designed and developed based on Modules. Each module will be completely separated and isolated from other modules in terms of database design and libraries used for development. This separation (modularity) will allow each module to grow and improve on its own in the future without affecting other modules.

2. Module-based design has other advantages such as building the large and basic modules separately and using them by sub modules which are developed & tailored for custom needs by companies.

3. Since all Modules are isolated and not based on each other or on a specific company, the portal will allow more than one company (Domains) to use the portal and share its functionalities.

4. An authentication system will be used to control what modules are visible for each company (domain). The authentication system is managed by the webmaster only.

5. Virtually, companies (Domains) will use the portal as if it was developed specifically for them. They will be able to completely customise it in terms of design, assigning permissions to its employees and sharing (download/upload) files within the portal with an option to enable and disable modules by companies (Domains) independently

6. Physically, the portal is centralised and deployed in the cloud. This will allow more than one company to subscribe in the portal and use its features (Modules) immediately without any additional customisation or modifications. This will also simplify the upgrade and debugging tasks where the webmaster will only need to update the source code or the database at one place affecting all Domains with the changes.

In stand-alone solutions, the webmaster will need to deploy new updates, bug fixes or backing up the database at each site where the portal is used. This will be very time consuming especially when the portal is used by large amount of companies.
4.2 Multi-Tier Design:

The portal is developed using a multi-tier model (Presentation, Business Logic & Data tier). This model was highly indicated in the development of this portal since the portal is built up using many web pages (aspx), a large number of data classes, business logic & utility classes in addition to a large number of SQL tables & queries.

It is difficult to maintain such requirements without dividing the application correctly into a layered design which separates each tier logically and then links them all together in the main project.

This multi-tier design was suggested and explained by (MacDonald, Freeman and Mario, 2010). It was very helpful and allowed architecting the project in a good way.

“In professional applications, database code is not embedded directly in the client but encapsulated in a dedicated class. To perform a database operation, the client creates an instance of this class and calls the appropriate method”. (MacDonald, Freeman and Mario, 2010)

Now it’s time to go a bit deeper into the system architecture and see how it works.

Figure 12: Layered design with a database class (MacDonald, Freeman and Mario, 2010)
Exploring the projects in our solution:

![Diagram showing project structure](image)

Figure 13 Snapshot from Visual Studio that shows all projects included in our solution

In the main solution, there are 2 Websites projects, 1 database class and 1 utilities class:

- **DatabaseComponent**: contains all code related to the data access
- **MyUtilities**: contains all helper classes such as email and MD5 hashing classes
- **Main**: which contains the whole portal and linked to all other libraries
- **Module_Careers**: it’s a custom module which is linked to the main project.

**Note:** Custom modules could be directly implemented in the “Main” project, and some custom modules have been already implemented in the “Main” such as the Projects Module. However, the Careers Module was very large and it has many pages (around 15 aspx pages) and tables and was preferred to separate it completely in a different project to avoid complication in the “Main” project.

### 4.2.1 Presentation Layer

The presentation layer contains all UI elements (user interface) such as:

- All WebPages (aspx) and UserControls (ascx)
- AJAX and 3rd party components (DLLs)
- All design elements, Style Sheets (CSS) and JavaScripts.
- It doesn’t include any data-related logic, SQL queries or other business logic classes. It’s just linked with the Business Logic Layer.
4.2.2 Business Logic Layer (Database Access & Utilities classes)

- Set of data classes which are compiled into a separate component (DLL) in order to allow multiple pages access it.

- Each database table is mapped into 2 classes. The *first class* (Data class) that defines the table’s properties as public fields, and the *second class* (Data Utility class) that consist of public stateless methods which define and wrap the SQL operations “queries” and populates a collection of the first class’s objects (using List or ArrayList). Stateless methods take all properties needed as parameters in the constructor; this simplifies the integration process with the other components.

- Defines the SQL operations required to access and manipulate the data in separate **Stored Procedure**. All Stored Procedures are saved physically in the database but they are considered to be part of the business logic layer because they operate according to the business logic requirements.

- This layer defines how the data is accessed rather than how it’s defined.

- In addition to data classes, this layer has also some **Utility classes** which are used by other components such as the Email class which is used by many pages in the presentation layer to send email notifications. And the Hashing class which is used to encrypt & decrypt strings and the Random Code Generator class which is used to generate random numbers used to verify new accounts.

- This layer has direct link to the **Data Layer**.

4.2.3 Data Layer

The Data Layer is only responsible for storing, accessing and maintaining data integrity.

Below are the main functionalities of this layer:

- It defines all data tables into the SQL Server.

- It defines all constrains (Primary Key, Foreign Keys, Default Values), this is very important in order to ensure proper indexing in each table.

- It defines **Referential Integrity** rules (Relations): These rules are used for maintain relations with specific properties between tables. For example a Relation between two tables could specify “Cascade Updates” or “Cascade Delete” which mean that foreign key should be updated/deleted automatically once the primary key changes. This will ensure the all records remain consistent in the database.

In the next section, each layer will be discovered in details from bottom to top (from data layer towards the presentation layer).
Chapter 5: UML (ER, Activity, Use Case & Class Diagrams)

5.1 Entity Relationship Diagram (Database Design):

Simplified ER diagram

Many modules are linked with the Domains & Employees tables; this is because each module is shared by more than one domain. On the other hand, many modules are required to log employees’ detail upon any request, that’s why the Employees table is included in almost all links.

A more detailed diagram is on the next page.
Detailed ER Diagram
5.2 Business Logic Layer Design

Not all modules are included in this section due to the similarity in design of some modules. Diagrams will be concentrating on the modules and the back-end design rather than GUI design in order to simplify the class diagrams. Only one class diagram of a GUI was included as a sample.

5.2.1 Activity Diagram (Authentication System):

![Activity Diagram](image)

Figure 15: Activity Diagram
5.2.2 Use Case Diagram (Login System):

Success Scenario:

1) The employee provides his/her login details
2) The authentication process takes place
3) If already registered, a further check will be made in order to get any extended permissions of the employee (i.e. allowed to post memos)

Unhappy Scenario:

2a. The employee is not registered
   - Show explanatory message

2a. The employee has provided wrong login details
   - Show explanatory message
5.2.3 Employee-Related Classes:

**CLASS DIAGRAM**

![Class Diagram - Employees Class](image)

Figure 17: Class Diagram - Employees Class
5.2.4 Domain-Related Classes:

**CLASS DIAGRAM**

Figure 18: Class Diagram - Domain-related Classes
5.2.5 Requests Module:

**STANDARD REQUESTS: CLASS DIAGRAM**

Figure 19: Class Diagram - Requests Module Class

**LEAVE REQUESTS: CLASS DIAGRAM**

Figure 20: Class Diagram - Leave Request Class
5.2.6 Memos Module (including the GUI class diagram):

**CLASS DIAGRAM**

Figure 21: Class Diagram - Memos Module Class

5.2.7 Events Module:

**CLASS DIAGRAM**

Figure 22: Class Diagram - Events Module Class
5.2.8 Task Assigning Module:

**CLASS DIAGRAM**

![Class Diagram - Task Assigning Class]

Figure 23: Class Diagram - Task Assigning Class
5.2.9 Messages Module:

**CLASS DIAGRAM**

![Class Diagram - Messages Module Class](image)

Figure 24: Class Diagram - Messages Module Class

5.2.10 Projects Module:

**CLASS DIAGRAM**

![Class Diagram - Projects Module Class](image)

Figure 25: Class Diagram - Projects Module Class
5.2.11 News Module:

**CLASS DIAGRAM**

![Class Diagram - News Module Class](image1)

Figure 26: Class Diagram - News Module Class

5.2.12 Meeting Planner Module:

**CLASS DIAGRAM**

![Class Diagram - Meeting Planner Module](image2)

Figure 27: Class Diagram - Meeting Planner Module
5.2.13 Helper Classes:

**CLASS DIAGRAM**

Note

![Class Diagram - Helper Classes](image)

**Figure 28: Class Diagram - Helper Classes**

**Note:**

1. The Email class is used to send emails by specifying all required details; it simplifies the initialisation process by wrapping all the required email’s classes into one class.

2. The LogError class is used to log any application-level errors to a text file.

3. The RandomCode class is used to generate random codes, these codes are used when verifying email addresses.

4. The Hash class is used to Encrypt/Decrypt passwords and other critical information.
5.3 Presentation Layer:

The following technologies were used in the presentations layer in order to provide richer user interface and better experience using the portal:

5.3.1 Certified 3rd Party Component:

Component Art (Chart Control)

Used to visualise data into 3d bars with options to compare 2 values on the same bar such as comparing the full budget with actual achievements.

Component Art (Auto-Complete Combo Box)

Very handy control which mixes the TextBox with the ComboBox. Users can search either by selecting a value from the list, or by typing the first letters causing all the matches to be automatically filled in the list.

Component Art (News Rotator)

Rotators are widely used in websites, but this control has some advantages such as the ability to bind directly ADO.Net DataSets, DataTables, XML or arrays.

The developer has the option to control the rotation speed, page size and some other parameters.
CK Editor

It’s currently one of the most famous controls used on the web for rendering HTML contents. This control allows users to input and format text as HTML complaint.

CK Finder

This control is produced by the same company as the CK Editor (FCK), it allows user to download/Upload files directly within the browser.

It also allows developers to apply security checks such as disabling the upload button if the user is authenticated.

5.3.2 AJAX Controls:

AJAX simplifies the mix of server-side technologies such as ASP.Net with the client-side technology (Javascript). AJAX enables developers to use a set of tools (designed for ASP.Net). These tools deliver many client-side functions. Developers can control these controls using server-side code (i.e. C#).

Some AJAX controls were used in order to enhance the user interface such as:

6 Calendar: A handy tool that pops up a small calendar that allows users to choose dates easily.

7 Confirmation boxes (Yes/No): Used when a critical action is about to occur such as deleting a record.

8 Popup Windows: Used when a minor action is required such as showing a small form that allows users to change their passwords.

9 Progress notification: Used to indicate a background process. It can also show a friendly message such as “please wait … “. This important because AJAX events are done in the background and thus users will not be able to notice the current status.
Chapter 6: Implementation & Challenges

In this section, the main modules will be explained in terms of development, database access and the logic used to achieve the required results. It will also cover the most critical decisions which have been taken during the development.

Some modules are not included in this section due to the similarity in design and development.

6.1 Choosing a design strategy for the whole portal:

The most critical and important decision was to decide how to architect the whole portal. Many ideas were available, "Shall the database queries and table definitions be embedded in the same project? OR "To follow a multi-tier architecture which divides the project logically into three main layers (Presentation, Business Logic & Data Layers).

A recommendation suggested by Macdonald in his latest book was followed. It sated to:

1. Define the data tables, views, constrains and referential integrity rules in the database (data layer).

2. Write all database methods and table-to-objects mapping in a separate project (DatabaseComponent.dll) and write all database queries as Stored Procedures in the database itself (business logic).

3. Keep all webpages, interface elements and third party components, Java Scripts and CSS in a separate project (presentation layer)

6.2 Multi-domain support:

At the beginning, the aim was to develop this portal to be used internally in companies. This would require some minor modifications every time the portal will be installed in a company such as changing the appearance to match the company's identity, disabling some modules which are not applicable to that company, defining company's divisions and branches and finally setup the database server and the web server to run the portal.

After finishing around 70% of the portal, it was a preferable idea to make the portal support multiple domains to avoid all the above obstacles. Company owners can then change the appearance and choose the modules they need in their companies themselves. Updating or fixing a bug in the portal would also be much easier since it has to be done once and all domains will be affected. A company reporting an error will benefit all other companies too after an update is done.
The challenge was to build up a module which links each domain with specifically selected modules. This will help assigning the required modules depending on the companies' needs since some of the modules might not be applicable for all domains (companies) such as projects-related module and other custom modules.

Another obstacle was faced during upgrading the portal which was to support multiple domains. Allowing companies to assign extended permissions to some employees was a must such as allowing the HR department to add, edit and delete records in the "Memos Module" where normal employees can only read.

Also finding a way to dynamically assign different Style Sheets to domains was important. This useful in order to allow companies to change the appearance of the portal (colours, H1, H2 etc.). This problem was solved by saving the required CSS file path in the Domains table itself and once the portal is loaded a JavaScript code will amend the <header> tag to use the selected “Style Sheet” path from the database.

A separate shared folder was required to be created dynamically on the server once a domain is created. The shared folder is used by the "shared files" module that allows the company to upload and download the commonly used files.

In order to allow companies to use their own email server host address, port number, require SSL or not, sender email address and the email display name, all SMTP server details of each domain were saved separately in the Domains table.

URL Rewriting was required in order to redirect registered companies directly to their domain page using a friendly URL. Registered companies will be provided with permanent URLs which directly take them to their domain page. The domain name will be appended at the end of the URL which defines the requested domain.

For example if a company called BMC Gulf with domain name bmc-gulf.com, the URL for that company will look as http://office.pixelstree.com/bmc-gulf.com. URL Rewriting internally translates the appended domain name into a parameter which will be used to fetch domain’s details from the database. Employees need only to enter the first part of the email & the password in order to login. The following snapshot shows the login page for BMC Gulf employees:
6.3 Domain-Modules Registration System:

The figure above shows how the Super Admin can assign modules to different domains, the red titles shows the main tabs in the navigation menu. The Super Admin has access to a secured control panel from which a portal-level management can be done.

The query below is used to assign modules for domains. 2 tables are linked in this module, Domains & DomainsModules tables. Selected ModuleIDs are passed into one variable separated using comma to the stored procedure along with a unique DomainID; the stored procedure then splits the ModuleIDs and loops accordingly to insert the values.

```
DECLARE @count int
DECLARE @str varchar(MAX)
DECLARE @spot smallint
BEGIN
    DELETE FROM DomainsModules
    WHERE (Domain = @Domain)
END
WHILE @ModuleIds <> ''
BEGIN
    SET @spot = CHARINDEX(',', @ModuleIds)
    IF @spot>0
        BEGIN
            SET @str = CAST(LEFT(@ModuleIds, @spot-1) AS INT)
            SET @ModuleIds = RIGHT(@ModuleIds, LEN(@ModuleIds)-@spot)
        END
    ELSE
        BEGIN
            SET @str = CAST(@ModuleIds AS INT)
            SET @ModuleIds = ''
        END
    INSERT INTO DomainsModules(Domain, ModuleId)
    VALUES (@Domain,@str)
END
END
```
6.4 Dynamic Homepage Panels:

Panels on the homepage are dynamically created according to modules registered by the company. Each panel is created in a separate user control (ascx) and then added to the homepage at the run time.

Note: some panels are visible by default for all companies such as Notifications panels, and some are optional. Before adding a panel to the homepage, a check is made testing whether the company has registered for the main module that represents the panel or not.

```csharp
private void AddPanels()
{
    // GETS COMPANIES REGISTERED MODULES
    ModuleDB moduleDB = new ModuleDB();
    List<ModuleDetails> modules = moduleDB.GetDomainInHomePanels(Session["domain"]).ToString();

    // Start from the Panel 2 because the first panel is default on the position 1
    int counter = 2;

    // Loop and add panels
    foreach (ModuleDetails module in modules)
    {
        ContentPlaceHolder cph =
            (ContentPlaceHolder)this.Master.FindControl("ContentPlaceHolder1");
        Panel pnl = cph.FindControl("Panel" + counter.ToString()) as Panel;
        UserControl usercontrol =
            (UserControl)LoadControl("HomePanels/" + module.HomePanel + ".ascx");

        pnl.Controls.Add(usercontrol);
        pnl.Visible = true;
        counter++;
    }

    if (counter > 5) Panel63.Visible = true;
}
```
6.5 Changing the interface (look & feel) dynamically:

![Main Login Page for Domains](image1)

*Note: The above login page is skipped if full URL including a domain name were provided.*

Once a company logs in, the following code is used in order to determine the Style Sheet to be used (CSS).

```csharp
protected void Page_Load(object sender, EventArgs e)
{
    // Change Style sheet dynamically
    string styleSheetPath = "/StyleSheets/" + ((DomainDetails)Session["DomainDetails"]).StyleSheetFile;
    AddCss(styleSheetPath, this.Page);
}

public static void AddCss(string path, Page page)
{
    Literal cssFile = new Literal()
    {
        Text = @"<link href="" + page.ResolveUrl(path) + @" type=""text/css"" rel=""stylesheet"" />
    }
    page.Header.Controls.Add(cssFile);
}
```

The `Session["DomainDetails"]` contains all details such as the assigned style sheet.
6.6 Users’ Authentication System:

An administrator can assign extended permissions to employees such as posting memos, uploading files and managing specific requests such as help desk tickets. Administrators can only assign permissions to modules which are registered under their domains. For example if the company is not registered for the events module, admins will not be able to see any extended permissions related to that module.

If the user is set to Administrator, they will get domain-level authorities which will allow them to control other users’ permissions as well as to configure the domain settings such as company’s divisions and branches.

The following database design is used to assign permissions to employees:
When assigning permissions to employees, the following SQL query is used which optimises the insert process; Permissions IDs are passed all at once using a comma separated list (@ParmID), and then looping within the stored procedure in order to perform the actual insert operation:

```sql
@EmpEmail varchar(30),
@PermID varchar(100) -- we pass ID separated with commas ,
AS

DECLARE @count int
DECLARE @str varchar(MAX)
DECLARE @spot smallint

BEGIN
  DELETE FROM EmployeesPermissions
  WHERE (EmpEmail = @EmpEmail)
END

WHILE @PermID <> ''
BEGIN
  SET @spot = CHARINDEX(',', @PermID)
  IF @spot>0
    BEGIN
      SET @str = CAST(LEFT(@PermID, @spot-1) AS INT)
      SET @PermID = RIGHT(@PermID, LEN(@PermID)-@spot)
    END
  ELSE
    BEGIN
      SET @str = CAST(@PermID AS INT)
      SET @PermID = ''
    END

  BEGIN
    INSERT INTO EmployeesPermissions(EmpEmail, PermID)
    VALUES (@EmpEmail, @str)
  END

END

Explanation:

The stored procedure splits the permissions list using the comma “,” and then assigns the permission ID to the employee in the EmployeesPermission table.

Note: the use of the DELETE statement at the beginning of the stored procedure, it’s used to remove all previous permissions prior assigning the new permissions.
6.7 Dynamic Navigation Menu (based on user’s permissions):

The users’ authentication system controls the navigation menu of all employees and sometimes controls the view of each page. For example, if an employee is assigned to some extended permissions, they will be able to see these extra permissions under the "Manage" tab in the navigation menu. Some permissions don't have separate pages under the manage tab, instead extra controls will be visible on the same page that all employees can access.

The following code is used to build up the navigation menu dynamically:

```csharp
private void buildMenu()
{
    DataSet ds = moduleDB.GetDomainModules(Session["Domain"].ToString(), true, false);

    ds.Relations.Add("NodeRelation", ds.Tables[0].Columns["ModuleId"],
                    ds.Tables[0].Columns["ParentModuleId"]);

    foreach (DataRow dbRow in ds.Tables[0].Rows)
    {
        if (dbRow.IsDBNull("ParentModuleId"))
        {
            ComponentArt.Web.UI.MenuItem newItem = CreateItem(dbRow);
            MenuNavigator.Items.Add(newItem);
            PopulateSubMenu(dbRow, newItem);
        }
    }

    // Build lat tab "Manage"
    BuildManageTab();

    // Save Menu in session to save time
    Session["MenuItems"] = MenuNavigator.Items;
}

private void PopulateSubMenu(DataRow dbRow, ComponentArt.Web.UI.MenuItem item)
{
    foreach (DataRow childRow in dbRow.GetChildRows("NodeRelation"))
    {
        ComponentArt.Web.UI.MenuItem childItem = CreateItem(childRow);
        item.Items.Add(childItem);
        PopulateSubMenu(childRow, childItem);
    }
}
```
private ComponentArt.Web.UI.MenuItem CreateItem(DataRow dbRow)
{
    ComponentArt.Web.UI.MenuItem item = new ComponentArt.Web.UI.MenuItem();
    item.Text = dbRow["Text"].ToString();
    item.NavigateUrl = dbRow["NavigateUrl"].ToString();
    if (dbRow["LookId"].ToString() != "") item.LookId = dbRow["LookId"].ToString();
    if (dbRow["SelectedLookId"].ToString() != "") item.SelectedLookId = dbRow["SelectedLookId"].ToString();
    if (dbRow["ChildSelectedLookId"].ToString() != "") item.ChildSelectedLookId = dbRow["ChildSelectedLookId"];  
    item.Look.RightIconUrl = dbRow["RightIcon"].ToString();
    return item;
}

// Build Manage Tab
private void BuildManageTab()
{
    int lastMenuItem = 0;

    // get and bind menu items only if the MenuItems session is not exist
    EmployeeDetails emp = (EmployeeDetails)Session["EmployeeDetails"];

    // - Write user name on the first menu item
    MenuNavigator.Items[0].Text = emp.FullName;

    // fill the "Manage" menu item according to employee's assigned permissions
    List<PermissionDetails> allEmpPerms = (List<PermissionDetails>)Session["EmployeePermissions"];

    // NOTE: we add all permissions which have special page to a separate collection
    List<PermissionDetails> specialPerms = new List<PermissionDetails>();
    foreach (PermissionDetails empPerm in allEmpPerms)
    {
        if (empPerm.NavigateUrl != "")
            specialPerms.Add(empPerm);
    }

    // Now we have list of permissions which have separate pages
    if (specialPerms.Count > 0)
    {
        ComponentArt.Web.UI.MenuItem item = new ComponentArt.Web.UI.MenuItem();
        item.Text = "Manage";
        item.LookId = "TopItemLook";
        item.SelectedLookId = "TopItemSelected";
        item.ChildSelectedLookId = "TopItemSelected";

        MenuNavigator.Items.Add(item);

        // add sub items to the last tab "Manage" that will include admin pages
        lastMenuItem = MenuNavigator.Items.Count - 1;

        foreach (PermissionDetails empPerm in specialPerms)
        {
            ComponentArt.Web.UI.MenuItem subItem = new ComponentArt.Web.UI.MenuItem();
            subItem.Text = empPerm.Name;
            subItem.NavigateUrl = empPerm.NavigateUrl;

            // add to Manage menu item
            MenuNavigator.Items[lastMenuItem].Items.Add(subItem);
        }
    }
}
6.8 Facebook-like messaging system

This was inspired by Facebook messaging system which shows the whole conversation in one thread. This system is very user friendly and clear for grouping messages. Below are the trickiest parts faced during the development of this module.

**Grouping all related messages within a conversation into one thread**

This was the first challenge faced in this module. Each message had to be assigned to (MsgID) and ThreadID. The MsgID is a unique field in the table (Auto increment integer) and is used as the primary key. The ThreadID will be the same for all messages within a same conversation. This allowed grouping messages using the ThreadID while the MsgID remains unique throughout the whole table.

**Allowing more than one party participate in a conversation**

Another obstacle faced was to allow more than one party participate in a conversation (similar to Facebook). This was the most difficult part to implement. The following questions had to be answered. "Shall a separate message be created for each party to allow them manipulate the message view separately such as set read and unread and delete the message without affecting the view of other parties?" OR "Is it better to have the record created once with proper properties used to allow each party manipulate the message view separately but on the same table record?"

The second option was followed which will not create separate instance of the same record for each party. A comma separated list of recipients was used for parties who have read the message, parties who have deleted the message and parties who have deleted the message from their sent items.

So for example, once a party deletes a message from their inbox, their name will only be added to the comma separated list of (HiddenFor) field in the table.

Setting a message to read for specific party, means adding their names to the (ReadFor) field in the table. This happens once the user clicks on a main thread, and then all related messages will be shown and will be set as read for that user. Read & Unread flags were used to show notifications on the users' homepage and also to show unread messages in different colours in the list.

Note: 3 fields control the message visibility, these fields have a default value set to “None”, and then values are appended separated by commas.

*ReadFor:* Used to indicate whether the message is read or unread

*HiddenForInbox:* Used to indicate users who have deleted the message from their inbox

*HiddenForSent:* Used to indicate users who have deleted the message from their sent items.

The following page shows some of the major queries used to achieve the above mentioned requirements.
Composing a new message query:

```
-- 1) Insert the message
INSERT INTO Messages
    (FromEmail, ToEmail, Body, Subject)
VALUES     (@FromEmail, @ToEmail, @Body, @Subject)

-- 2) Get the auto-field MsgID
SET @MsgID = @@IDENTITY

-- 3) Use it to set ThreadID = MsgID
UPDATE Messages SET ThreadID = @MsgID WHERE MsgID = @MsgID

-- NOTE
-- Setting ThreadID = MsgID will simplify selecting thread's messages later
-- where the main message and related replies will have the same ThreadID
```

Explanation:

This query shows how messages are grouped under the same ThreadID once the first message is created. The ThreadID for the first message (main) will be equal to the MsgID, and then all future replies will have the same ThreadID.
Replaying to a message query:

![Figure 33: Snapshot of replying to messages window](image)

```sql
INSERT INTO Messages (ThreadID, FromEmail, ToEmail, Subject, Body)
VALUES (@ThreadID, @FromEmail, @ToEmail, @Subject, @Body)
```

**Explanation:**

This query shows how replies get grouped under the main message using the ThreadID. Once a reply is made, the ThreadID of the main message will be passed as a parameter to the insert query of the reply message.

Now, all related messages share the same ThreadID. The ThreadID could be used as the unique key for all these related messages.

Last thing was to complete the query mentioned above by setting the message to Unread for all users in order to allow them get notified about the new reply:

```sql
UPDATE Messages
SET HiddenForInbox = 'None', HiddenForSent = 'None'
WHERE ThreadID = @ThreadID
```
Reading messages query:

```sql
SELECT Messages.*, Employees.FirstName, Employees.LastName, Employees.ExtPhoto
FROM Messages INNER JOIN Employees ON Messages.FromEmail = Employees.Email
WHERE ThreadID = @ThreadID
ORDER BY SentOn
```

```sql
UPDATE Messages
SET ReadBy = Cast(ReadBy as varchar(MAX)) + ', ' + @UserEmail
WHERE (ThreadID = @ThreadID) AND (ReadBy NOT LIKE '% ' + @UserEmail + '%')
```

The above query is used to get all messages within a thread using ThreadID. Then the user who read the message their email will be appending to the comma separated list in `ReadFor` field:

**Showing a list of main threads only using the latest sent message from each thread query:**

The last thing was to implement viewing messages by employees. The aim was to show a list of main threads only by showing the newest message from each thread; a brief of the message with the picture of the sender was included.

```sql
SELECT Messages.*, Employees.FirstName, Employees.LastName, Employees.ExtPhoto
FROM Messages INNER JOIN Employees ON Messages.FromEmail = Employees.Email
WHERE Messages.SentOn IN
    (SELECT MAX(Messages.SentOn)
     FROM Messages
     WHERE Messages.ToEmail LIKE '%$' + @ToEmail + '%$'
     AND Messages.HiddenForInbox NOT LIKE '%$' + @ToEmail + '%$')
ORDER BY SentOn DESC
```

The above query was one of the trickiest past faced. The snapshot below shows a user’s inbox page who has received 2 messages from “Osama Mortada”. These 2 messages are different threads; each of them has its own messages.

The light blue shading indicates that the message is unread.

---

**Osama Mortada | Sent On: 08/August/2011 03:13 AM**

Subject: URGENT request

Message: Please complete the required tasks as per our conversation today.

---

**Osama Mortada | Sent On: 08/August/2011 03:11 AM**

Subject: How is your project

Message: Hi kareem, I wanted to ask you about your project.
6.9 Meeting Planner Module

Allowing employees to make recurrent meeting reservations such as weekly, daily etc.: 

A very handy feature made in the room meeting reservation section was to allow employees make recurrent meeting reservations which means specifying a specific date and time for a meeting and repeating it for a specific period of time. For instance, Mr.Daniel works as a Human Resources manager in a company using the office automation portal and needs to make a weekly meeting every Monday for the duration of 3 months till a major project is completed with all the employees to discuss specific topics about the project. Mr.Daniel goes onto the meeting room reservation section in the office automation portal and is firstly asked about the beginning date of the meeting, the title of the meeting which in Mr.Daniel's case it is "Project Discussion", then the beginning and ending timings of the meeting which are 2pm to 3pm. He is then asked about the number of recurrences which in Mr.Daniel's case would be every week.

Finally Mr.Daniel is asked about the duration of the required recurrences. Since Mr.Daniel needs to have every Monday reserved for the duration of three months (till the project is done), he chooses the duration 3 months. The duration is actually for how long the meeting reservation will keep recurring.

Mr.Daniel also has the capability of deleting one occurrence or deleting all the recurrences at once so no need to get back and delete each and every one.
Detecting overlapping meetings and rescheduling:

Since the meeting room reservation option is available to all employees at any time, overlapping is expected. After Mr. Daniel filled in all the details of his meetings and clicked on "Save reservation", he was notified by a friendly message telling him the number of successful recurrences. Only one was not successful and overlapped a meeting made by another employee. Mr. Daniel may either reschedule it or negotiate with the other employee to change his meeting reservation.

Daily, weekly, monthly and yearly meetings view:

For future reservations, Mr. Daniel can view all the reservations made for the meeting room either for this day, this week, this month or this year which will help him avoid choose reserved timings. A trick was faced while making the reservation view of the weekly meetings which was excluding the days that have already passed. For example it is Tuesday today and Mr. Daniel wants to view the reservations of this week, so Sunday and Monday today had to be excluded since they have already passed. Tuesday, Wednesday and Thursday only need to be viewed.

The query used to detect meeting conflicts:

```sql
SELECT MeetingRoom.*, Employees.LastName, Employees.FirstName
FROM MeetingRoom INNER JOIN Employees ON MeetingRoom.ReservedBy = Employees.Email
WHERE (MeetingRoom.Domain = @Domain) AND
      (DAY(MeetingRoom.EndDate) - DAY(@Date)) AND
      (MONTH(MeetingRoom.EndDate) = MONTH(@Date)) AND
      (YEAR(MeetingRoom.EndDate) = YEAR(@Date))
ORDER BY MeetingRoom.StartDate
```
The following method demonstrates the logic used to achieve all the above mentioned challenges:

```csharp
protected void InsertRecurrentMeeting(DateTime startDate, DateTime endDate)
{
    string currentOwner = "";

    DateTime LastRecurrentDate = new DateTime();
    DateTime FirstRecurrentDate = new DateTime();

    // counter used to increment reservation periods
    int increment = Convert.ToInt16(DropDownListRecurrency.SelectedValue);

    // count successful insertions
    int countInserted = 0;

    // count errors & description of the errors
    int errorsCounter = 0;
    string errorsDesc = "";

    FirstRecurrentDate = Convert.ToDateTime(TextBoxDate.Text);

    if (DropDownListDuration.SelectedValue != "Till end of this year")
        LastRecurrentDate =
            FirstRecurrentDate.AddDays(Convert.ToInt16(DropDownListDuration.SelectedValue));
    else
        LastRecurrentDate =
            Convert.ToDateTime("31/December/" + DateTime.Now.Year.ToString());

    // Generate UniqueIdentifier, this will be used by all reservations as MeetingID
    Guid guid = Guid.NewGuid();

    Explanation:
    
    In the first part of the method, some counters and variables are created which is used to verify the parameters of the requested meeting. These variables are used in the next part in order to make the actual insertion.
for (DateTime i = First Recurrent Date; i <= Last Recurrent Date; i = i.AddDays(increment))
{
// check if the date has any conflicts with previous meetings,
// get name of current owner in case of conflict
    currentOwner =
        db.GetReservationOwnerByDateRange(Session["Domain"].ToString(), startDate, endDate);

// If new reservation does not conflict with previous meetings
if (currentOwner == "")
{
// Check if the selected date exceeds this year
    if (i.Year > DateTime.Now.Year)
    {
        errorsCounter++; // Increment error counter
        errorsDesc += errorsCounter + " . Next year not allowed " + i.ToString("dd MMM yyyy") + 
            " <br />
;
    } else
    {
// add record
    try
    {
        emp = (EmployeeDetails)Session["EmployeeDetails"]; // Get employee details

        db.InsertMeetingReservation(Session["Domain"].ToString(),
            guild, TextBoxTitle.Text, emp.Email, startDate, endDate,
            DropDownListRecurrency.SelectedItem.Text, GetReceiptsList());

        countInserted++;
    }

    // in case of any sql error; rare case
catch (ApplicationException ex) { errorsDesc += ex.Message; }
    }

// Else, there is owner, add error to err string
else
{
// compose error with full details
    errorsCounter++; // Increment error counter
    errorsDesc += errorsCounter + " . " + startDate.ToString("dd MMM")
        + " for " + currentOwner + "<br />
;
}

// Increment the counter
    startDate = startDate.AddDays(increment);
    endDate = endDate.AddDays(increment);
}

Explanation:

Here, a loop is made according to the requested period, then a check is made for testing whether the requested date has any conflicts or not, and if everything went well, the (insertCounter) will increment otherwise the (errorCounter) will increment.
Finally, the finalisation is made by displaying the results on the user’s GUI similar to the following:

4 reservation(s) accepted

5 conflict(s) found:
1. 25 August for kareem@bmc-gulf.com
2. 01 September for kareem@bmc-gulf.com
3. 08 September for kareem@bmc-gulf.com
4. 15 September for kareem@bmc-gulf.com
5. 22 September for kareem@bmc-gulf.com
6.10 Requests & Ticketing Module:

Employees can make *four different types of requests*, after submitting any of these requests; employees in charge (according to the request type) will be notified using emails and notifications on the homepage. Below are two sample requests:

**IT helpdesk:** This request is commonly used in almost all companies; the employee would explain the problem and set its priority, then whoever is responsible will get notified in order to take action.

![Image of IT helpdesk request form]

**Purchase request:** This module has some more features than the other requests modules. It allows an employee to add more than one item to the request with an option to attach documents which will then be sent to admins as an email notification.

![Image of purchase request form]

*The system automatically calculates the subtotal of each item and finally calculates the total of all items.*
6.11 Leave Request with Work-Flow:

After an employee has made a leave request, a notification appears on their supervisor’s homepage. The supervisor has the right to either accept it or decline it.

If the request is cancelled, no further action is taken whereas if it is accepted, a notification is sent to someone with authority in the company such as the HR manager to complete the leave request steps such as the leave salary, air tickets etc.

This was achieved using the (Status) field in the leave table. By default, the request will be in “Pending” mode once created which will allow supervisors only to get notified in order to approve it or decline it.

Note: optionally, authorised HR personnel can create leave applications for technicians or employees who do not have access to the portal.

Calculating leave and remaining days (excluding weekends):

```csharp
TimeSpan ts = dtTo - dtFrom;

// keep a copy of dtFrom before altering its value
DateTime dt = dtFrom;

int weekends = 0;

while (dt < dtTo)
{
    if (dt.DayOfWeek == DayOfWeek.Friday || dt.DayOfWeek == DayOfWeek.Saturday) weekends++;
    dt = dt.AddDays(1);
}

// show leave days (without weekends)
LabelLeaveDays.Text = (ts.Days - weekends).ToString();

// get remaining leave days
```

Explanation:

In the method above, after the TimeSpan is calculated using the provided leave date and return date, the while loop detects and counts the weekends within the TimeSpan range in order to calculate a pure leave days.
Once the request is approved by the supervisor, the flag will be set to “Approved” which will allow the HR personnel who is in charge of completing leave requests to get notified to complete the process.

Note: the same page is used for admins and supervisors to approve and complete leave requests, but according the logged in user permissions, the proper logic will apply.
6.12 Memos with Work-Flow Module:

In the office automation portal, posting memos is an extremely easy process to carry out. It is done over two steps. The first step is adding the memo by someone having authority in the company such as the HR executive manager. The second and final step is the approval of the HR manager. Once the added memo is approved, it is instantly posted for everyone in the portal to see.

This work-flow system was achieved using one flag in the Memos table (IsApproved) which indicates the status of the memo. Only authorised managers can set this flag to True and only authorised HR personnel can create a new memo:

```csharp
private void CheckSecurity()
{
    // set admin panel to hidden
    PanelAdmins.Visible = false;

    List<PermissionDetails> perms = (List<PermissionDetails>)Session["EmployeePermissions"];
    foreach (DatabaseComponent.PermissionDetails perm in perms)
    {
        if (perm.Name == "Add/Edit Memos")
        {
        }
        if (perm.Name == "Memos Management")
        {
            - PanelAdmins.Visible = true;
        } break;
    }
}
```

**Note:** by default all admin controls are hidden for normal users, and only if the logged in employee has sufficient privileges, admin controls will be enabled accordingly.

*The above authentication technique is used in most modules.*
6.13 Task Assigning System:

This module is very important within any company which doesn’t have a task assigning system. It allows managers and supervisors to assign tasks within their teams, follow up tasks and set the deadlines. The following work-flow is used in this module:

1. An employee can assign a task to one or more other employees.
2. They need to choose the deadline of the task.
3. The task appears on the notification panel of each user who is a member of the task.
4. Task members can communicate and interact together within the assigned task.
5. Task owner can set the task status to closed or on hold according to the achieved results.

The trick in this module was to allow multiple users to interact within the same task, and this was achieved using a separate table that links all employees who are members of a specific task with the task record in the Tasks table:

During the development, optimisation of the `Insert` operation was highly indicated, since once a task is requested with multiple employees involved (i.e. 10 employees), the `Insert` operation will be required to occur for each employee in the `TaskTransactions` tables (10 times).

An easy approach was followed which was to use a comma separated list of all task’s members in one field (MembersEmails) instead of having a separate table for the same. This is useful when extra properties about the link are not required.

In order to retrieve each member’s tasks, the LIKE operator is used as follows:

```sql
SELECT Tasks.*, Employees.FirstName, Employees.LastName
FROM Tasks INNER JOIN Employees ON Tasks.Owner = Employees.Email
WHERE Tasks.Status = @Status AND IsHidden = 0 AND
  (Tasks.MembersEmails LIKE '%@EmployeeEmail + %') OR
  (Tasks.Owner = @EmployeeEmail)
```
6.14 Document-Expiry Detection System:

Document expiry system is a system that regularly checks the expiry dates of documents. It is helpful for Human Resource managers since it automatically notifies them by email and homepage notifications when a passport or any other document is about to expire to take the required action.

The following code was used to get any documents which will expire in 30 days:

```sql
SELECT FirstName, LastName, ExpDatePassport, ExpDateVisa, ExpDateLabour
FROM Employees
WHERE ExpDatePassport < DATEADD(day, 30, GETDATE()) OR
    ExpDateVisa < DATEADD(day, 30, GETDATE()) OR
    ExpDateLabour < DATEADD(day, 30, GETDATE()) AND Domain = @Domain
```

Note: the restriction in the WHERE clause (Domain = @Domain). This is important because as mentioned earlier, this portal is shared between multiple domains.

**Documents Expiry (in 30 days)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Passport expiry date</th>
<th>Visa expiry date</th>
<th>Labour expiry date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abdul Latheef</td>
<td>01/Feb/2005</td>
<td>15/Jun/2012</td>
<td>20/Feb/2012</td>
</tr>
<tr>
<td>Osama Mortada</td>
<td>30/Jul/2011</td>
<td>28/Apr/2013</td>
<td>23/Nov/2019</td>
</tr>
<tr>
<td>Osama Mortada</td>
<td>30/Jul/2011</td>
<td>28/Apr/2013</td>
<td>23/Nov/2019</td>
</tr>
</tbody>
</table>
6.15 Notifications System:

This system groups and counts all related activities of logged in users. It shows the new, unread messages, pending tasks and birthday notifications.

For authorised users, it shows any pending requests such as the IT Help desk; purchase requests, payment requests etc. For supervisors, it shows pending leave requests made by a member of his team.

This section allows instant view of all recent activities made by users themselves or pending actions waiting for completion by the user.

The following code demonstrates how notifications are created and shown on users’ homepage, two sample notifications are included (pending memos & and new messages):

```csharp
// Count Pending Memos (if authorized to approve memos)
foreach (PermissionDetails empPerm in empPerms)
{
    if (empPerm.Name == "Approve Memos")
    {
        count = memoDB.CountPendingMemos(Session["Domain"]).ToString();
        if (count > 0)
        {
            LabelNotifications.Text +=
            "<a class='notificationItem' href='Memos.aspx'>" +
            "<img src='..\images\notification\memo.png' style='float:right' /"" >" +
            count.ToString() + " Pending Memo(s)</a>";
        }
    }
}

// Count Unread Messages
count = msgDB.CountUnreadMessages(emp.Email);
if (count > 0)
{
    LabelNotifications.Text +=
    "<a class='notificationItem' href='Messages.aspx'>" +
    "<img src='..\images\notification\msg.png' style='float:right; border:0px' /"" >" +
    count.ToString() + " Unread Message(s)</a>";
}
```

Note: the whole notifications system is displayed into one Label control, which renders HTML tags the defined CSS classes.
6.16 Emailing System:

ASP.Net has some built-in methods for sending emails. Different classes are used to create the message object (From, To, Subject and body), handle attachments, apply custom credentials and a class for initialising the server object (SMTP server details).

In this project, a separate method was used which wraps all the required methods in one class. This simplifies the process of initialising and sending email messages.

Constructor:

```csharp
public Email(string domain, string smtpServer, string smtpEmail, string smtpPassword,
int smtpPort, bool isSSL, string emailDisplayName)
{
    // initialize the SMTP Client
    smtp = new SmtpClient(smtpServer, smtpPort);
    smtp.UseDefaultCredentials = false;
    smtp.Credentials = new NetworkCredential(smtpEmail, smtpPassword);
    smtp.EnableSsl = isSSL;

    // initialize the Email Message
    msg = new MailMessage();
    msg.From = new MailAddress(smtpEmail, emailDisplayName);
    msg.BodyEncoding = System.Text.Encoding.UTF8;
    msg.IsBodyHtml = true;

    // Set Domain
    Domain = domain;
}
```

Method 1 (Sending an email message with attachment and fixed body):

```csharp
// Send using direct text for body
public void Send(string recepients, string subject, string body, ArrayList attachments)
{
    msg.To.Add(recepients);
    msg.Subject = subject;
    msg.Body = body;

    // if Attachment
    if (attachments != null)
    {
        foreach (Attachment t in attachments)
        {
            msg.Attachments.Add(t);
        }
    }

    ActualSend(msg);
}
```

The text of the body is passed directly to this method; it also accepts an Array of attachments.
Method 2 (Sending an email message without attachments and dynamic body):

```csharp
public void Send(string recipients, string subject, string[] parms, string[] values, string fileName)
{

    for (int i = 0; i < parms.Length; i++)
    {
        body = body.Replace(parms[i], values[i]);
    }

    msg.To.Add(recipients);
    msg.Subject = subject;
    msg.Body = body;

    ActualSend(msg);
}
```

**The dynamic text file looks as follows:** The $name variable indicates a replacement of a dynamic value passed to the method above.

```html
<h2>New Profile Created</h2>
<br />
Profile for <b>$name</b> has been created successfully.
<br />
<h4>Office Automation Portal</h4>

<h5>-- No Reply --</h5>
```

Method 3 (Actual sending): all previous methods are used only to construct the message, the below method is used to execute the actual SMTP sending operation. Since the SMTP transfer might be interrupted for any reason such as a slow internet connection or any packet loss, this method attempts sending the message three times. Whenever a failure occurs, it repeats the process before it returns an indication of a complete failure.

```csharp
private void ActualSend(MailMessage msg)
{
    try { smtp.Send(msg); } // First
    catch
    {
        try { smtp.Send(msg); } // Second
        catch
        {
            try { smtp.Send(msg); } // Third
            catch { /* do nothing */ }
            finally { msg.Dispose(); }
        }
    }
}
```
6.17 Error Detection:

6.17.1 Method-Level Error Detection:

**Try, catch & finally:** During development, the `try/catch` blocks were used when required. This would allow detection of expected errors. The `finally` block was also used when a method finalisation is important. For example, when a method opens a connection to the database, the `finally` block guarantees that the connection is closed whether the method completed successfully or not.

Since the portal contains more than one project, exceptions are thrown from down to top. For example, all methods in DatabasComponent.DLL throw their exceptions directly to the caller method including detailed description of the error; the caller method would then handle the error as required. The code below shows the use of the “`try, catch & finally`” blocks in a method within the DatabaseComponent project:

```csharp
try
{
    con.Open();
    cmd.ExecuteNonQuery();
}
catch (SqlException err)
{
    // Replace the error with something less specific.
    // You could also log the error now.
    throw new ApplicationException("Data error." + err.Message);
}
finally
{
    con.Close();
}
```

**TryParse:** Where possible, the TryParse(object) method is used. This method allows the early detection of errors when converting (casting) critical objects such as converting String to DateTime or to Decimal. The code below explains the use of this method in detecting invalid Date format:

```csharp
DateTime dtFrom;
DateTime dtTo;

bool d1 = DateTime.TryParse(TextBoxDateOfLeave.Text, out dtFrom);
bool d2 = DateTime.TryParse(TextBoxDateOfReturn.Text, out dtTo);

if (d1 && d2)
{
    if (dtTo > dtFrom)
    {
        TimeSpan ts = dtTo - dtFrom;
```
6.17.2 Application-Level Error Detection:

What happens if an unhandled error has occurred?

ASP.Net framework provides an Application-Level error detection mechanism through the use of the Global.asax file. The Global.asax file has some default methods, one of these method fires when an unhandled exception occurs. The following code snippets demonstrate the use of this method:

```csharp
void Application_Error(object sender, EventArgs e)
{
    Exception objErr = Server.GetLastError().GetBaseException();
    string err = "Error Caught in Application_Error event\n" +
        "Error in: " + Request.Url.ToString() +
        "\nError Message:" + objErr.Message.ToString() +
        "\nStack Trace:" + objErr.StackTrace.ToString();

    // Send report by email
    // MyUtilities.Email.Send("admin", "Office Automation Portal | Bug", err);

    // clear server errors to avoid catching the error in web.config
    Server.ClearError();

    // Log the error
    MyUtilities.LogError.WriteEntry(Environment.NewLine + "-------------------" +
        + DateTime.Now + "-------------------" +
        + Environment.NewLine + err + Environment.NewLine);

    // show friendly error page
    //Response.Redirect("~/Error.aspx", true);
    Response.Write(err);
}
```

The above method appends any unhandled error to a text file; the following static method is used for logging errors:

```csharp
public static void WriteEntry(string error)
{
}
```
Chapter 7: Deployment & Testing

7.1 Deployment:

The portal is currently accessible at a temporary URL ([http://office.pixelstree.com](http://office.pixelstree.com)). This URL will be changed later with a permanent domain-name.

The portal is centrally hosted on the cloud (internet), it’s available for all companies to use without any special requirements from their side. However, it could also be hosted internally within a company upon request.

During the development, once a module is completed, it will be uploaded for testing. This will ensure that each function is tested separately and the feedback will be collected accordingly. This will also help employees to understand the portal functionalities and to simplify the training task.

Some modules could not be tested separately which are connected to other modules or require a work-flow system. These modules will be uploaded together at the final stage.

Deployment & Testing Stages (Agile Methodology):

- The portal will be deployed in the cloud
- Training will be provided for employees
- Collecting feedback from management & employees
- Updating the system where necessary

Server-Side Requirements:

Below are the requirements for the server which will be used to host the portal:

- MS SQL database server 2008 or higher
- IIS 7 or higher web server with .Net Framework 4.
- 2 GB RAM
- Dual Core processor (quad-code is recommended)

Client-Side Requirement:

- Web-browsers such as Internet Explorer (v6 or higher), Firefox, Google Chrome or Safari.
7.2 Testing:

7.2.1 Tasks Module:
7.2.2 Memos Module:

Add Memos

Memos Module:

7.2.3 Approving & Posting Memos:
### 7.2.4 Requesting new meeting reservation:

**Meeting Planner**

- **Required Date:** 25 Aug 2011
- **Meeting Title:** Test overlapping meeting
- **Start Time:** 1 PM
- **Finish Time:** 4 PM
- **Recurrences:** Once

**25, AUGUST**
- Every Week
- Business-IT Alignment
- Kareem Essam

**01, SEPTEMBER**
- Every Week
- Business-IT Alignment
- Kareem Essam

**03, SEPTEMBER**
- Once

---

**Meeting overlapping detection**

**Meeting Planner**

Oops, meeting room already reserved for kareem@bmc-gulf.com at the selected period. Please change the period.

**Meeting Planner**

Start time should be less than end time.

---

**Incorrect time detection**
7.2.5 Event Module:

Add Events

<table>
<thead>
<tr>
<th>Event Name</th>
<th>GITEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>15/October/2011</td>
</tr>
<tr>
<td>To</td>
<td>20/October/2011</td>
</tr>
<tr>
<td>Remarks</td>
<td>IT Related exhibition</td>
</tr>
</tbody>
</table>

- Group of Interest:
  - Accounts
  - Administration
  - BMC Abu Dhabi
  - Design
  - Dubai Contracting
  - Human Resources
  - Information Technology
  - Logistics
  - Marketing
  - Plastic Welding Technology
  - Purchasing
  - TF
  - The Finest Bathroom
  - Trading

Save

Events

<table>
<thead>
<tr>
<th>FROM</th>
<th>TO</th>
<th>Event</th>
<th>Group of Interest</th>
<th>Remarks</th>
<th>Edit</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oct 15</td>
<td>Oct 20</td>
<td>GITEX</td>
<td>Information Technology</td>
<td>IT Related exhibition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jul 21</td>
<td>Jul 24</td>
<td>Cityscape</td>
<td>Accounts, Logistics</td>
<td>The other day I was working on a code behind form and I needed to dynamically load a UserControl and access its properties. I was stumped, but with a little help from the codejunkies at asp.netgen.com I was able to move on. Here is a little sample to demonstrate this technique.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jul 18</td>
<td>Jul 22</td>
<td>Egyptscape</td>
<td>Logistics</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Add Events
7.2.6 Viewing Shared Files

Shared Files

On double click, open selected document

7.2.7 Uploading New Files & Creating New Subfolders

Shared Files
7.2.8 Updating Employee’s Profile

<table>
<thead>
<tr>
<th>Personal Details</th>
<th>Work Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Title</strong></td>
<td><strong>Hire Date</strong></td>
</tr>
<tr>
<td>Mr.</td>
<td>1 February 1991</td>
</tr>
<tr>
<td><strong>First Name</strong></td>
<td><strong>Job Title</strong></td>
</tr>
<tr>
<td>Osama</td>
<td>Assistant Accountant</td>
</tr>
<tr>
<td><strong>Last Name</strong></td>
<td><strong>Dept./Division</strong></td>
</tr>
<tr>
<td>Morlada</td>
<td>Dubai Contracting</td>
</tr>
<tr>
<td><strong>Email</strong></td>
<td><strong>Branch</strong></td>
</tr>
<tr>
<td><a href="mailto:osana@bmc-gulf.com">osana@bmc-gulf.com</a></td>
<td>Abu Dhabi</td>
</tr>
<tr>
<td><strong>Login Password</strong></td>
<td><strong>Reports To</strong></td>
</tr>
<tr>
<td>Change</td>
<td>Shady Morlada</td>
</tr>
<tr>
<td><strong>Nationality</strong></td>
<td><strong>Work Cell</strong></td>
</tr>
<tr>
<td>Syria</td>
<td>34024</td>
</tr>
<tr>
<td><strong>Birth Date</strong></td>
<td><strong>Extension</strong></td>
</tr>
<tr>
<td>5 August 1981</td>
<td>123</td>
</tr>
<tr>
<td><strong>Home Phone</strong></td>
<td><strong>Passport Exp Date</strong></td>
</tr>
<tr>
<td>3442342</td>
<td>30 July 2011</td>
</tr>
<tr>
<td><strong>Emergency Contact</strong></td>
<td><strong>Visa Exp Date</strong></td>
</tr>
<tr>
<td>Shady</td>
<td>28 April 2013</td>
</tr>
<tr>
<td><strong>Emergency Phone</strong></td>
<td><strong>Labour Exp Date</strong></td>
</tr>
<tr>
<td>31454234</td>
<td>23 November 2019</td>
</tr>
<tr>
<td><strong>Address</strong></td>
<td><strong>Passport Copy</strong></td>
</tr>
<tr>
<td>The Gardens</td>
<td>Choose File</td>
</tr>
<tr>
<td><strong>Upload Photo</strong></td>
<td><strong>Labor Card Copy</strong></td>
</tr>
<tr>
<td>No file chosen</td>
<td>Choose File</td>
</tr>
<tr>
<td><strong>Upload CV</strong></td>
<td><strong>Employee Number</strong></td>
</tr>
<tr>
<td>No file chosen</td>
<td></td>
</tr>
</tbody>
</table>

Your account has been updated successfully.
Back to home page

7.2.9 Change Company’s Turnover Values

<table>
<thead>
<tr>
<th>Division Name</th>
<th>Budgeted Value (millions)</th>
<th>Actual Value (millions)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC Abu Dhabi</td>
<td>34.50</td>
<td>25</td>
</tr>
<tr>
<td>Dubai Contracting</td>
<td>43.30</td>
<td>23.90</td>
</tr>
<tr>
<td>The Finest Barch</td>
<td>empty</td>
<td>20</td>
</tr>
</tbody>
</table>

Save
7.2.10 Help Desk Request

Help Desk Request | New

Problem Title: My PC is very slow
Priority: High

Problem Description: [Blank]

Help Desk Request | View

My PC is very slow
Priority: High
Request Description: Please solve my problem I can't work.
(Test for Polsys)

Monitor
Priority: High
Request Description: Please fix my monitor I can not see anything
Requested On: 31/July/2011 03:35 PM | Updated On: 08/August/2011 03:44 AM
7.2.11 Purchase Request

<table>
<thead>
<tr>
<th>Serial</th>
<th>Description</th>
<th>Unit</th>
<th>Quantity</th>
<th>Unit Price</th>
<th>Sub Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HP Laptop</td>
<td>PCS</td>
<td>2</td>
<td>2,200.00</td>
<td>4,400.00</td>
</tr>
<tr>
<td>2</td>
<td>LCD Screens</td>
<td>PCS</td>
<td>2</td>
<td>600.00</td>
<td>1,200.00</td>
</tr>
</tbody>
</table>

7.2.12 Payment Request

<table>
<thead>
<tr>
<th>Purpose</th>
<th>text purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Division</td>
<td>Dubai Contracting</td>
</tr>
<tr>
<td>Vendor</td>
<td>Sharaf</td>
</tr>
<tr>
<td>Payment Mode</td>
<td>Cash</td>
</tr>
<tr>
<td>Currency</td>
<td>AED</td>
</tr>
<tr>
<td>Amount</td>
<td>3000</td>
</tr>
<tr>
<td>Payment Date</td>
<td>27/Aug/2011</td>
</tr>
<tr>
<td>Priority</td>
<td>High</td>
</tr>
</tbody>
</table>
7.2.13 Defining Groups (within domains)

Add Groups

<table>
<thead>
<tr>
<th>Group Type</th>
<th>Department</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Accounts</td>
</tr>
</tbody>
</table>

Save

Groups

- Accounts
- Administration
- BMC Abu Dhabi (Division)
- Design
- Dubai Contracting (Division)
- Human Resources
- Information Technology
- Logistics
- Marketing
- Plastic Welding Technology
- Purchasing
- TFF (Division)
- The Finest Bathroom (Division)
- Trading

Delete

7.2.14 Defining Branches (within domains)

Add Branches

<table>
<thead>
<tr>
<th>Branch</th>
</tr>
</thead>
</table>

Save

Abu Dhabi

Dubai

Delete
7.2.15 Projects Module:

Projects

<table>
<thead>
<tr>
<th>Division</th>
<th>BMC Abu Dhabi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Country</td>
<td>Angola</td>
</tr>
<tr>
<td>City</td>
<td>City</td>
</tr>
<tr>
<td>Area</td>
<td>57876</td>
</tr>
<tr>
<td>Application</td>
<td>Evaporation Pond Lining</td>
</tr>
<tr>
<td>Value</td>
<td>7,777.00</td>
</tr>
<tr>
<td>Est. Cost</td>
<td>999.00</td>
</tr>
<tr>
<td>Retention %</td>
<td>8</td>
</tr>
<tr>
<td>Guarantee</td>
<td>kjkjkjkjk</td>
</tr>
<tr>
<td>Remarks</td>
<td>ikjkjkjkjk</td>
</tr>
<tr>
<td>Quotation No.</td>
<td>98</td>
</tr>
<tr>
<td>Customer No.</td>
<td>98798</td>
</tr>
<tr>
<td>Contractor</td>
<td>cont</td>
</tr>
<tr>
<td>Client</td>
<td>client</td>
</tr>
<tr>
<td>Payment Terms</td>
<td>50% Upon order confirmation</td>
</tr>
</tbody>
</table>

Edit Projects

Project No.: 1005
Project Name: Palm
Division: BMC Abu Dhabi
Open Date: 01/Jan/1991
Actual Start Date: 01/Jan/1991
Expected End Date: 01/Jan/1991
Country: Angola
City: City
Area (m²): 57876
Application: Evaporation Pond Lining

Value: 7,777.00
Estimated Cost: 999.00
Retention %: 8
Guarantee: kjkjkjkjk
Quotation No.: 98
Customer No.: 98798
Contractor: cont
Client: client
Payment Terms: 50% Upon order confirmation

Save
7.2.16 Managing Domains Notifications

7.2.17 Managing Employees’ Permissions
7.2.18 Portal Admins’ Modules

Adding a New Domain

Updating Domains
Chapter 8: Evaluation

8.1 Managers Feedback Form

This form will be used by managers in order to evaluate the admin part of the portal. The admin part is designed to provide a content management system (CMS) for the portal.

**Office Automation Portal**

**Managers Feedback Form**

How long do you spend in average using the portal?

How much does the portal fulfill your daily needs?

Do you need more training on how to use the portal?

Is the user interface friendly enough?

Do you have any suggestions or ideas for improving the application?

If you are facing any problems with the application, please specify below:

Which functions or modules you use the most?

- [ ] Posting HR Memos
- [ ] Handling HR Requests
- [ ] Projects Management
- [ ] Posting to Discussion Board
- [ ] Events Management
- [ ] Documents Management
- [ ] Task Assignment
- [ ] Assets Management
- [ ] Meeting Room Reservations

Employee Name:  

Date:  

Department:  

Figure 35: Managers Feedback Form
8.2 Employees Feedback Form

This form will be used by employees in order to evaluate the portal’s general modules. More than one copy of this form could be provided weekly.

![Image of Employees Feedback Form]

If you are facing any problems with the application, please specify below:

Which functions or modules you use the most?

- [ ] HR Forms & Policies
- [ ] Discussion Board
- [ ] Events
- [ ] Home Page Posts & News
- [ ] IT Forms
- [ ] Messaging
- [ ] Projects Search
- [ ] Following up My Tasks
- [ ] Accounting Forms
- [ ] Documents Search
- [ ] Meeting Room Reservations

Employee Name: ___________________________ Date: ____________
Department: ______________________________

![Image of Employees Feedback Form]
8.3 Satisfaction Levels:

**Figure 38: Satisfaction Level (Employees Module)**

**Figure 39: Satisfaction Level (Admin Modules)**

**Figure 40: Satisfaction Level (General Consideration)**
8.4 The Overall opinion about the portal

![Pie chart showing overall opinion about the portal]

Figure 41: The Overall opinion about the portal

8.5 Key users’ opinions:

“I consider this portal as an important tool of communication between employees, managers and supervisors. It allows automation of many of our paper based work. It is complementary to our ERP application and I hope we can integrate them together in the near future.”

Reda Ashkar – General Manager – BMC Gulf

“I think this portal simplifies many of our daily tasks, some modules are very important for the HR Department such as the Memos, Leave Requests and the Documents-Expiry. I hope that we will benefit from this portal in all our company’s department after proper training of our employees.”

Mohamed Shady, HR Manager – BMC Gulf

“As a Division manager who supervises a team of around ten employees, I find this portal very helpful for assigning and following up tasks. I like also the meeting planner and the events management modules; they provide centralised pool where all employees can easily access information.”

Fabian Beermann – Division Manager – BMC Gulf

“I was working as a tester for this portal, I used all modules thoroughly and I am impressed with the various features that the portal provides. I found the portal very reliable, fast and user friendly which will encourage our employees to use it on daily basis. I admired also that all critical information are encrypted and secured because the portal is hosted externally on the cloud.”

Mohammed Rafeeq – IT Administrator – BMC Gulf
Chapter 9: Professional, legal and ethical issues

Professional:

Although this web office automation portal is designed to serve a range of companies, only employees within the same company will be allowed to access and share their company’s information.

Some features of this portal were designed to be used by admins and management only such as assigning employees’ level of access, configuring the workflow, posting HR memos, reviewing forms and requests, monitoring the discussion board, assigning tasks etc.

This portal is currently designed to work in English language only. All employees should consider English input during the use of the portal.

The portal currently doesn’t contain features that suite disabilities however, these features could be added in the future according to companies’ needs.

Legal:

Some modules such as Leave Requests are currently abiding to the UAE Law in terms of calculating the allowed annual leave days, Week-ends, Official Holidays and the calculation of the leave salary. However if these modules are required to be used outside the UAE, it could be easily customised to abide with laws of other countries.

Ethical:

All employees’ sensitive data such as passwords, messages etc. will be secured and saved in an encrypted form in the database. Even admins will not have access to this data by any means. Admins will have the right to disable access of certain employees or remove their related data by an official request from the management.

The portal allows internal messaging system that could be used between employees to send and receive messages similar to the normal email systems. However, this feature was designed to support business-related communications and should only be used for this purpose.
Chapter 10: Project Plan

10.1 Schedule:

<table>
<thead>
<tr>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Office Automation Portal Plan</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report Preparation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abstract, Introduction</td>
<td>5 days</td>
<td>Feb 1</td>
<td>Feb 5</td>
</tr>
<tr>
<td>Literature Review</td>
<td>45 days</td>
<td>Feb 25</td>
<td>Apr 28</td>
</tr>
<tr>
<td>Project Goals</td>
<td>3 days</td>
<td>Mar 8</td>
<td>Mar 10</td>
</tr>
<tr>
<td>Risk Management</td>
<td>3 days</td>
<td>Mar 11</td>
<td>Mar 15</td>
</tr>
<tr>
<td>Presentation Preparation</td>
<td>5 days</td>
<td>Mar 16</td>
<td>Mar 22</td>
</tr>
<tr>
<td>Requirement Analysis</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Get detailed requirements, collect forms and other material</td>
<td>4 days</td>
<td>May 1</td>
<td>May 4</td>
</tr>
<tr>
<td>Draft UML Diagrams</td>
<td>6 days</td>
<td>May 2</td>
<td>May 8</td>
</tr>
<tr>
<td>Main Interface Prototype (HTML)</td>
<td>5 days</td>
<td>May 9</td>
<td>May 13</td>
</tr>
<tr>
<td>Function-specific Prototype (HTML)</td>
<td>3 days</td>
<td>May 17</td>
<td>May 19</td>
</tr>
<tr>
<td>Development</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database Creation (SQL)</td>
<td>2 days</td>
<td>Mar 23</td>
<td>Mar 24</td>
</tr>
<tr>
<td>Database Design (ER Diagram)</td>
<td>2 days</td>
<td>May 20</td>
<td>May 22</td>
</tr>
<tr>
<td>Coding/Testing</td>
<td>41 days</td>
<td>May 25</td>
<td>Jul 20</td>
</tr>
<tr>
<td>Evaluation &amp; Testing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deployment on a real environment</td>
<td>8 days</td>
<td>Jul 21</td>
<td>Aug 1</td>
</tr>
<tr>
<td>Final Retouches</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Updating the final report</td>
<td>6 days</td>
<td>Aug 1</td>
<td>Aug 7</td>
</tr>
</tbody>
</table>
Figure 42: Project Plan - Gantt Chart

- Abstract, Introduction
- Project Goals
- Risk Management
- Presentation Preparation
- Requirement Analysis
- Get detailed requirements, collect forms and other material
- Draft UML Diagrams
- Interface Design
- Main Interface Prototype (HTML)
- Function-specific Prototype (HTML)
- Database Creation (SQL)
- Database Design (ER Diagram)
- Coding/Testing
- Evaluation & Testing
- Deployment on a real environment
- Final Retouches
- Updating the final report
## Chapter 11: Risk Management

<table>
<thead>
<tr>
<th>Risk</th>
<th>Risk Prevention</th>
<th>Severity</th>
<th>Likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Insufficient knowledge of company work flow</strong></td>
<td>Must conduct a thorough company analysis in order to understand each step.</td>
<td>Severe This can cause unexpected results.</td>
<td>Unlikely</td>
</tr>
<tr>
<td><strong>Delayed Feedback</strong></td>
<td>Use a wider customers’ feedback from more than one company.</td>
<td>Severe Can result in delaying development</td>
<td>likely</td>
</tr>
<tr>
<td><strong>Lack of clear information</strong></td>
<td>Double checking with different departments.</td>
<td>Severe This can cause unexpected results.</td>
<td>likely</td>
</tr>
<tr>
<td><strong>Lack of knowledge by employees using the system</strong></td>
<td>Must conduct a detailed training and provide employees with a manual for the system.</td>
<td>Severe This can cause employees not to use the system</td>
<td>likely</td>
</tr>
<tr>
<td><strong>Insufficient Testing</strong></td>
<td>Test system vigorously by all possible users to ensure it works exactly as it should. Test using different test data and scenarios.</td>
<td>Very Severe Will cause errors to reside in the systems and cause problems when used.</td>
<td>Very Unlikely</td>
</tr>
<tr>
<td><strong>Unexpected Application Errors</strong></td>
<td>Separate modules to simplify debugging. Thorough debugging for each module separately.</td>
<td>Very Severe Can result in delaying development</td>
<td>Unlikely</td>
</tr>
<tr>
<td><strong>Sickness</strong></td>
<td>To finish as much as possible at an early stage.</td>
<td>Very Severe Will delay progress of project and cause late submission.</td>
<td>Likely</td>
</tr>
</tbody>
</table>
Chapter 12: Future Improvements

12.1 Integration with other software such as ERP:

In the future, this application could be expanded easily by adding new modules, customising current ones and could also be connected to other external systems such as ERP application or HR software. The application will consider modularity during the development, which means each module, will be separately developed and deployed. This will enable admins to disable or enable features according to their needs, and will also prevent the customisation of one feature from affecting other modules.

12.2 Multi-Lingual Support:

One of the good features that could be implemented is the support for the Arabic language; this will enable companies who use the Arabic language as their primary language in business to use the portal.

12.3 Mobile Version:

Mobile users have become an important player in the web, allowing such users to access a light version of the portal which will increase its usability. iPhones and iPads will be considered due to the huge increase of their use. Software development of these Apple devices became much easier than before, now Adobe Flash can be used to develop applications for iPhone & iPad. (Bansod, 2010)
Chapter 13: Conclusion

During designing and analysing the requirement of this portal, it was very helpful to compare and understand different technologies and tools that could be used to build the portal. Being employed in a company which was in need of most of the implemented modules, this helped me to understand the detailed process and the work-flow of each module.

Being an ASP.Net developer and after completing a sufficient study on object oriented programming and UML designing at the university, this helped me to easily translate the portal requirements using proper object oriented techniques. These techniques were very important to use in such a big project in order to clearly design and architect the portal.

The multi domain support was a big improvement to this portal which might allow me in the future to expand it in order to support a wider range of companies with different needs.

During the year I spent in my Master's degree, I continuously had to develop programs and submit assignments. This was very important to keep me on the right development and working atmosphere. The courses I learnt in my Master’s Degree enriched my knowledge and helped in linking different techniques based on a solid knowledge.

Although every section of this portal was a challenge and required deep thinking, I have enjoyed the whole journey in developing it.

I hope I can market this portal in the near future. This will be after completing enough time of real environment testing with at least three or four companies for duration of at least six months.

Finally, I am very proud of taking the opportunity to develop such a useful project for the dissertation of my Master’s Degree. I hope this is the beginning of a real improvement in my career which will hopefully continue.
References


When will HTML5 be finished? (2011), [Online] [17 April 2011].
