Fontys university of applied science | HBO ICT | Graduation Report

SCHEDULING SYSTEM

Philips Healthcare in Best

Huiwen Liang | 2116701

PREFACE

This report is written to present my work during the graduate project which took place at Philips Health Care department in Best, Netherlands. First of all, I want to thank Mr Jack Zijlmans` guidance as the university tutor during the 6-month period graduation project. Furthermore, I acknowledge Mr Lenssinck and Mr van Mierlo who chose and provided me a nice project as my graduation project. Especially Mr Hans, as the company tutor, who guided me through the project and gave me a lot of advisement which were very useful.

Contents

SUMMARY

The graduation project ‘Scheduling System‘ was performed at Philips HealthCare in Best, Netherlands. Philips healthcare manufactures medical devices and also medical IT. Installations of these devices in hospitals are done by service personnel who can do their work only when they receive proper training, which is provided by the training academy.

The initial situation of the project was that the Philips` instructors are using virtualisation to do their training courses. A course is created in a master template, a file system also called: datastore. When a course is coming, the instructor needed to make a number of copies according to the corresponding ‘Master DataStore’ which contains the training VM’s(virtual machines). These copies are given to service personnel to do their training.

Due to the authority, however, only the VMware administrator can make the copies of the ‘Master DataStore’ by using an existing copy application.
The problem was: we need instructors to create the copy datastores themselves instead of sending requests to the administrator. Additionally, the copy application needed to become an automated action instead of the manual action it was. Therefore, I was assigned to develop a scheduling system that allows all instructors to schedule their courses. The system must generate/delete the corresponding virtual machines according to the schedule time.

I followed the software developing methodology Waterfall model to work through the project. At the end of the project I successfully developed a scheduling system which basically consists of :
- a web application for user to schedule the courses,
- a windows service to implement necessary functions,
- a SQL database to store the necessary data.
- two Windows PowerShell applications to communicate with the VMware Vcenter server.
- a text file and a XML file to store necessary data

The hardest point of this project was finding a way to make each part of the system works logically, especially the windows service and Windows PowerShell applications, which are the heart of the entire system.

The system could be integrated as one application. However, the reason why I made each part separately was that if an error occurred, the administrator may find out and fix the problem easier without breaking other parts. Fortunately, the final system fulfills all the requirements. In addition, it is very easy to use.

Chapter 1 Introduction

Usually the courses are showed based on website explorer, power point, video or flash and etc. Have you ever heard that the courses was stored in an virtual machine? The training center of Philips Health Care in Best, however, is using this way for their training. The training center in Best receives a lot of employees from worldwide departments of Philips each year for a short term training. One or more computers working together, with corresponding operating systems, are needed for a service person during the course. In fact, it is impossible to provide a huge number of computers for all students in this case, because there are lot of courses start at the same time. Besides, some computers may need to be restored a new proper operate system for the next different course. To save the money and the time of work, the virtual machines were decided to be used to store the needed system for variety courses.

The VMware vSphere 5.1 which is VMware`s cloud computing operating system is used to manage all the virtual machines. In this system, the virtual machines can be created, removed and copied manually. However, only the administrator is having the permission to log in to the system and manage virtual machines. Each time a training course is going to start or end, the instructors need to send the create/delete requests to the administrator who will then implement the requests in the virtual environment. This leads to the result that the administrator needs to spend a lot of time to arrange the courses for instructors. Furthermore, if there are many courses take place at the same period, the work of virtual machines management will become very hard and complex.

My project assignment is to develop a scheduling system which allows for all instructors to schedule their courses, and the system can follow the schedules` properties to create/delete the virtual machines from the virtual environment automatically.

The basic strategy about developing this schedule system is :
Finding out the way to send creation or deletion commands to VMware vSphere environment so the virtual machines can be created or deleted automatically.
Then the platform is created where the users can make the schedules.
Finally, the previous two parts are integrated that the schedules` properties can be followed to send to correct commands to the virtualization environment.

In the following parts of the report, you will find the information about the company, where I did the graduation project in chapter 2. More details about the assignments were described in chapter 3. Chapter 4 gives the research information towards this project. Chapter 5 introduce the Waterfall model methodology which I followed during the project development.. The chapter 6,7,8,9 and 10 explains how the Scheduling system was developed in each project phase. To present a clear view of the final product, some code and screenshots are shown in these chapters after got the permission of the company. The last Chapter gives the conclusions about the whole project.

Chapter 2 The Company

2.1 Company Introduction

Philips Health Care was formerly known as Philips Medical Systems International B.V. The company was founded in 1896. Philips Healthcare in Best is the largest Philips location in the world. It is located at Veenpluis 4-6, 5680 DA Best,The Netherlands.

Figure 2.1 Company Planform

The company has more than 3,000 employees that work in diverse fields ranging from research and development to marketing and production.

Philips Healthcare offers diagnostic imaging systems, healthcare information technology solutions, and patient monitoring and cardiac devices. It operates through two divisions, General X-ray, and Cardiac and Monitoring Systems. The company also provides thermometers, blood pressure devices, and television monitors, as well as training and education, business consultancy, maintenance and repair, document, and e-care business services. It serves hospitals, research facilities, and outpatient centers, as well as healthcare consumers and providers.

2.2 Team and Organization Chart

The graduation project was provided by a sub department of Philips Health Care which is called CILS (clinical informatics learning services). The CILS is responsible for training the Philips field service engineers, so that they can install and maintain the Philips Health Care products. The following chart indicates the relationship between the project members and the company.

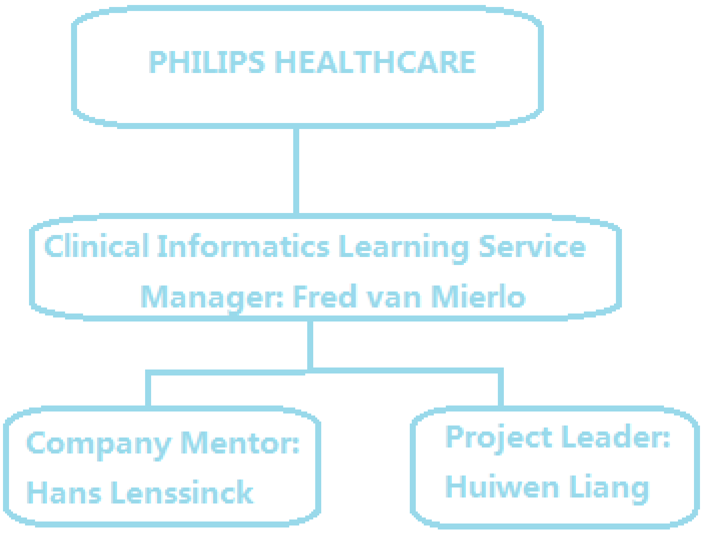


Figure 2.2: Organization Chart

2.3 My position

My position in the project was the project leader. I was the only member during project development stage, Mr Hans Lenssinck, as a project tutor, guided me to understand the project in the first few weeks. Then I was responsible to analyse, design and implement the project.

Chapter 3 Assignment Overview

3.1 Initial Situation

3.1.1 Background

The CILS (clinical informatics learning services) department of the Philips Heath Care is responsible for training the Philips field service engineers. Many computers with corresponding operating systems may be needed for each engineers or trainers during the training courses. To avoid the huge budget on the large amount of physical computers and maintain the stability of the operate system, virtualisation and virtual machines are used to store the divers system for the training course.

3.1.2 Working Environment

The company used VMware vSphere 5.1 to manage all the virtual machines. VMware vSphere leverages the power of virtualization to transform data centers into simplified cloud computing infrastructures and enables IT organizations to deliver flexible and reliable IT services.

VMware vSphere virtualizes and aggregates the underlying physical hardware resources across multiple systems and provides pools of virtual resources to the data center.[1]

The VMware vCenter Server, which is one of the VMware vSphere`s components, is the central point for configuring, provisioning, and managing virtualized IT. So the virtual machines can be easily created or deleted after users log in the vCenter.

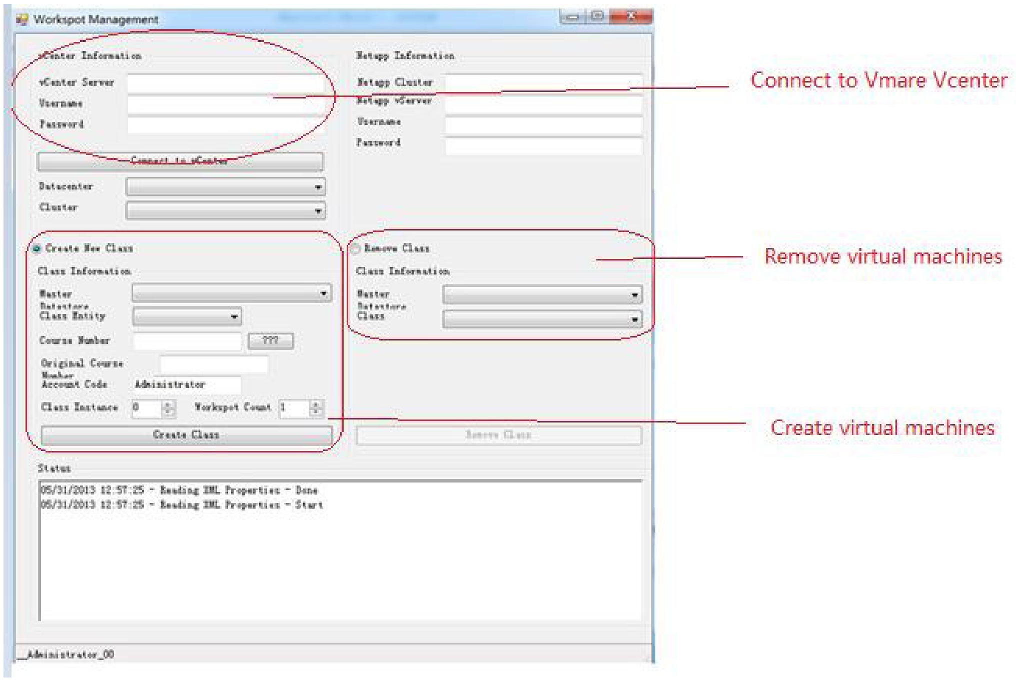
3.1.3 Problems

In diverse courses, the number of needed virtual machines for each trainer are different. In the vCenter, therefore, a list of folders have been created for all distinct courses which contain a number of corresponding virtual machines. Those folders are called Master-DataStores, which are static and unchangeable, during the class. These can therefore be considered as the content of the course and serve as template.

When a course is coming up, the number of copies of the specific Master DataStore must be created for each trainers. For example, if 14 service persons attend to a course, the corresponding Master-DataStore has to be copied 14 times in the vCenter. Each copy contains exactly the same set of virtual machines as the Master DataStore. The service persons work on the copies and the template will not be used at that moment.

Due to roles and permissions, only the administrator is allowed to log in the VMware vCenter. Therefor the instructors had to request the administrator to create the copies of the Master-DataStore before the start date of the course. In addition, the administrator was requested again to delete those copies after the courses` end time. This situation led to the result that the administrator spent a lot of time on arranging the virtual machines of the courses for instructors.

The following picture is the GUI of the ‘copy application’ that was used to create/delete the copies of the Master-DataStore by the administrator.



Account Code: Administrator
Class Instance: 0
Workspot Count: 1

```
05/31/2013 12:57:25 - Reading XML Properties - Done
05/31/2013 12:57:25 - Reading XML Properties - Start
```

_Administrator_00

Figure 3.1: The GUI of Copy Application

Each time the administrator creates or deletes the copies of the Master-DataStore, the vCenter server name, username and password must be given to connect to the VMware vCenter. After that, the administrator chooses one of the Master-DataStores and fills in the relevant data, such as, class name, number of copies, etc. As soon as the administrator clicks on the ‘Create’ and ’Delete’ button, the create or delete commands will be immediately implemented in the vCenter.

3.2 Purpose

The purpose of this project was to develop a platform which named ‘Scheduling System’ that could basically achieve the following goals:
- All users can independently schedule the courses in advance.
- The ‘Scheduling System’ can follow all the schedules` properties to automatically create/delete the copies of corresponding Master-DataStores in the VMware vCenter.
- All schedules` data are recorded in the database.

(Extra) Before the copies of Master DataSores have been deleted from vCenter, the usage of virtual machines` disk, CUP, network will be recorded in the database for the future billing (datamining capability).
By the end of this project, the end product must achieve all the basic goals, the extra goal will be nice to implement.

The project will help both administrator and instructors save a lot of time and work in configuring the virtual machines of the courses.

3.3 Company provides and unconstraints

To help me implementing the project well, the company provided me many things in hardware and software fields:
A laptop with company`s network.
An independent virtual machine with the Windows Server 2008 operate system in the VMware vCenter. This virtual machine will be the environment where I need to develop the ‘Scheduling System’
The SQL Server 2008 R2 which stored in my virtual machine.
The complete script of the existing ‘Copy Application’

During the project developing process, I was free to choose software and programming languages. I was responsible to design and implement the ‘Scheduling App’ independently.

CHAPTER 4 RESEARCH

4.1 Information Gathering Methods

This graduation project was the first project that I have to analyze, design and implement independently. So that I have done a lot of research during the project development. Most of needed information was gathered from the Internet.

Except searching information from the Internet, I also discussed with several experts in IT fields, who provided some ideas during the project development.

4.2 Techniques

4.2.1 C#

C# was the most used programming language during the project, which is also one of the main programming skills that I learned from school.

C#(pronounced “c sharp”) is an Object-Oriented programming language from Microsoft that aims to combine the computer power of C++ with the programming ease of Visual Basic.[3] The purpose of C# is to precisely define a series of operations that a computer can perform to accomplish a task.

C# is more high level approach than C++ which is usually faster to develop in. It comes with a large framework of pre developed components, which makes it particularly useful for server side programming. On the cost of flexibility and runtime performance aspects, C# makes development faster and easier.

4.2.2 Asp.net

Asp.net was used to built an web application in the project. It is the next generation of Microsoft`s Active Server Page(ASP).[4] Asp.net provides a unified Web development model that includes the services necessary for developers to dynamically build Web pages on the fly by inserting queries to relational database in the Web page. It also provides a new programming model and infrastructure for more scalable and stable applications that help provide greater protection. Asp.net is a compiled, .NET-based environment which can be developed in any .NET compatible language, including Visual Basic, .NET, C# and Java Script. Developers can easily access the benefits of these technologies, which include managed common language runtime environment, type safety, inheritance, and so on.

The advantages of Asp.net are:
- Asp.net has multiple language abilities
- The Asp.net web applications are stored on the server, so it takes less execution time for complication.
- Asp.net makes development easy due to it can work on all .NET frameworks.
- Asp.net has security features like windows authentication and windows authorization, so the web application is more secure than any other.
- Asp.net pages are easy to maintain because HTML code and source code are combined in one application.

4.2.3 SQL

SQL stands for Structured Query Language which is used to communicate with a database. It is the standard language for relational database management which was used in the project. Many common relational database management systems use SQL, such as Oracle, Sybase, Microsoft SQL Server, Access, Ingres. Etc.[5]

Most database systems using SQL, have their own additional proprietary commands to communicate with the database. Standard SQL commands such as “Select”, “Insert”, “Update”, “Delete” can be used in every database system.

The advantages of using SQL are diverse. SQL queries can be used to retrieve large amounts of records from a database quickly and efficiently. Also, SQL databases use long established standards , which is being adopted by ANSI and ISO. Non SQL databases do not adhere to any clear standard. Furthermore, it is easier to manage database systems without having to substantial amount of code.

4.2.4 Microsoft Windows Service

Before introducing the windows service, I want to mention that the idea of using windows service as a background running application was gotten from one expert in IT field. This was used to build the main part of the final product in this project.

The Microsoft windows service. It formerly known as NT services, which enables users to create long running executable applications that run in their own Windows sessions. Comparing with other applications, the windows services have some particular features:

- they run in the background
- do not have a user interface that a user can click or tap on.[6]
- The services can be automatically started when the computer boots, it can be paused and restarted as well.

The windows services can be created as a Microsoft Visual Studio project with C# programming language. It is easy to create services by creating an application that is installed as a service. Suppose I want to monitor the start time and end time of the schedules and give the react when the time is expired. I could write a Windows Service application that keeps looking at the data in the database, monitor the time and deploy the reactions. The commands can be executed in the triggers of the service include starting, resuming and stopping the service.

Due to the services` stability, background and long time running, I used it in the project as the main part to track the information in the database and execute the corresponding commands on time.

4.2.5 Windows PowerShell

The ‘copy application’ was mentioned in Chapter 3, which was created in Windows PowerShell script. This application contains all commands that are used to communicate with VMware vCenter. In order to split out the useful commands from the ‘copy application’, I had to find out Windows PowerShell and how to use it.

Windows PowerShell is a task-based command-line and scripting language designed especially for system administration. Built on the .NET Framework, Windows PowerShell helps IT professionals and power user control and automate the administration of the Windows operating system and applications that run on Windows.[7] Built-in Windows PowerShell commands, called ‘cmdlets’. It lets the user manage the computer from the command line.
In addition, Windows PowerShell has a rich expression parser and a fully developed scripting language, it has the following features:

- Cmdlets is used to implement common system administration tasks, such as managing the registry, service, processes and event logs.
- It is a task-based scripting language.
- It supports for existing scripts and command-line tools.
- Powerful object manipulation capabilities.
- Extensible interface and independent software vendors.

4.3 Tools

4.3.1 VMware vSphere 5.1

VMware vSphere 5.1 is the tool that be used to configure and manage the virtualization environment in the company. It leverages the power of virtualization to transform datacenters into simplified cloud computing infrastructures and enable IT organizations to deliver flexible and reliable IT services. VMware vSphere virtualizes and aggregates the underlying physical hardware resources across multiple systems and provides pools of virtual resources of the datacenter.[1] The datacenter consists of basic physical building blocks such as x86 virtualization servers, storage networks and arrays, IP networks, a management server and desktop clients.

The VMware ESX and ESXi, one of VMware vSphere`s components which is a virtualization layer run on physical servers that abstracts processor, memory, storage and resources into multiple virtual machines. Another important component is the VMware vCenter, which is the central point for configuring, provisioning and managing virtualized IT environments. There are many more components are consisted such as vSphere Client, Virtual Machine File System, Virtual SMP and so on.

The vSphere client can be used to view, configure, and manage the key elements in the virtualization environment which include:
- Computing and memory resources called hosts, clusters and resource pools
- Storage resources called datastores
- Networing resources called networks
- Virtual machines.

It provides several interfaces for datacenter management and virtual machines access which includes: vSphere Client, vSphere Command-Line Interface, vSphere Management Assistant and Web Access. Due to roles and permissions, however, I was only able to access the virtual machine that I worked on by the VMware vSphere Web Access, which can also access vCenter Server through the Web browser by first pointing the browser to an Apache Tomcat Server set up by vCenter Server. The Apache Tomcat Server mediates the communication between the browser and vCenter Server through the VMware API.To access the virtual machine consoles through the Web browser, users can use the bookmark that vCenter Server creates. The bookmark first points to the vSphere Web Access. If the virtual machine is running, it also can be accessed by using its IP address on Windows Terminal Server.

4.3.2 Visual Studio 2010

Visual Studio is a very powerful software development environment, also known as an Integrated Development Environment(IDE). This tool was used to implement most of software developing during the project.

Visual Studio is a suite of applications created by Microsoft to give developers a compelling development environment for the Windows applications and

.NET platforms.[8] It can be used to write variety applications, such as console applications, windows applications, windows service, windows mobile applications, asp.net web application, WebService, Mobile application and etc. The developers are able to use many programming language in this environment, which includes C++, C#, VB.NET, J# and so on.

Visual Studio was stated offering in 1997, there have been numerous versions since its inception. Accompanying more and more versions had been released, the Visual Studio becomes more and more powerful.

The main features of Visual Studio are:
- IntelliSense: helps developer while programming by showing the available classes, methods and properties.
- Designers: Visual Studio including visual WYSIWYG designers for many applications development. These designers mate it much easier to get the application looking just right.
- Debugging: is the ability to step through the application line by line as it is executing.
- Organization: Visual Studio provides intuitive methods for organizing the various code files into project and the projects into solutions.

4.3.3 Internet Information Service(IIS)

Internet Information Services (IIS) for Windows Server is a flexible, secure and manageable Web server for hosting anything on the Web. From media streaming to web applications, IIS's scalable and open architecture is ready to handle the most demanding tasks.[9]
The newest version of IIS supports many protocols such as the HTTP, FTP,FTPS,SMTP,NNTP and etc.

The IIS publishes the normal HTML pages or Microsoft`s ActiveServer Pages(ASP). It also publishes the programs of Microsoft's Internet Server Application Program Interface (ISAPI) interface. ASPs and ISAPI programs run more efficiently than common gateway interface (CGI) and server-side include (SSI) programs.
In the project, IIS was used to host the ‘web application’ in the virtual machine. Thus, all instructors are able to access the ‘web application’ under their working environment fastly and easily.

4.3.4 Power GUI

PowerGUI is an extensible graphical administrative console for managing systems based on Windows PowerShell.[10] It is a very nice editor with color syntax highlighting, intellective showing the corresponding variables and functions, and debugging abilities.

In this project, the PowerGUI was used to analyze, rebuild and compile the PowerShell script application base on the ‘Copy Application.’

Chapter 5 Scheduling System Development Methodology

The software development method that I partly followed during this project was the Waterfall Model [2]. The waterfall model is a classic approach to the systems development life cycle model for software engineering. It describes a development method that is linear and sequential. The whole process of software development is divided into separate phases, which are showed as the following figure:

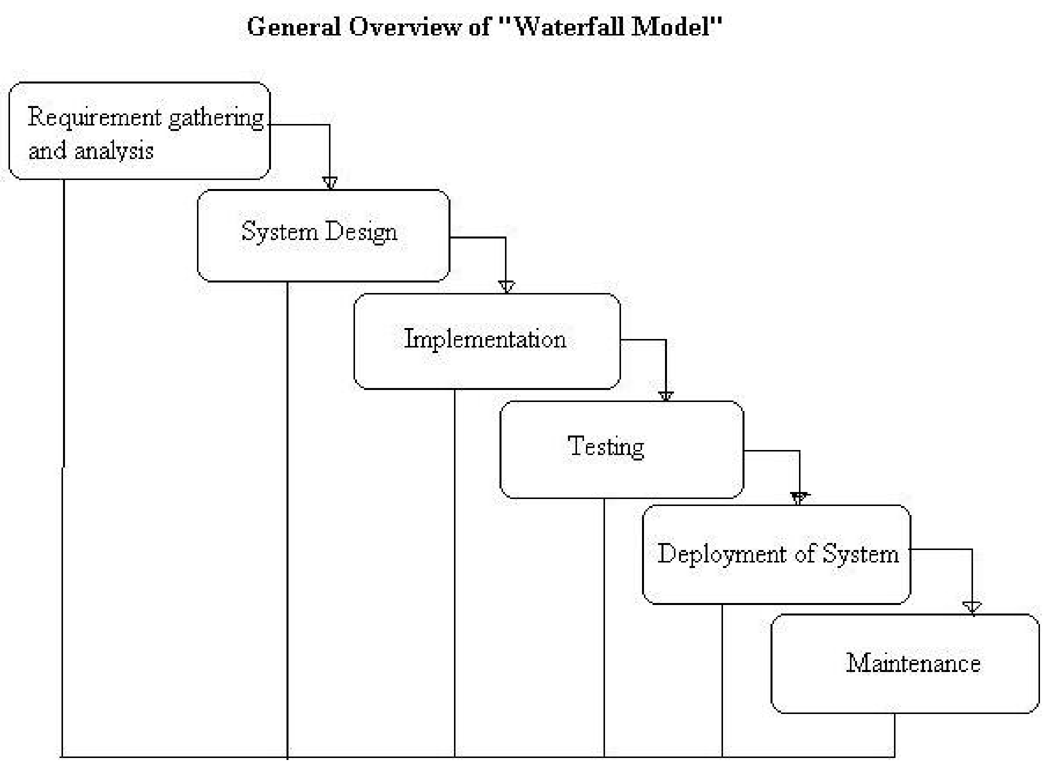

General Overview of "Waterfall Model"
Requirement gathering and analysis
System Design
Implementation
Testing
Deployment of System
Maintenance

Figure 5.1 General Overview of Waterfall Model.

The Waterfall Model is very simple and easy to understand and use. It gives a clear distribution of work and control which makes it easier to set a schedule for the tasks to be completed within a specific period. In addition, the project does not have to go through different iterative steps at the same time. The testing happens at the end of the phases, which helps in maintaining the quality of the project.

Drawing the advantages of the Waterfall Model, I devided the project development processes into the following phases:

5.1 Analysis and requirements gathering

In this phase, there are many meetings with the company tutor to understand the current situation, problems and specify the requirements. This is a crucial stage which can lead the following development into the right direction.

5.2 System Design

To make the implementation phase easy and structured, the requirements which specified in previous phase are broken down in to logical units. During this phase, the interrelation of various units of the software are identified and the connection are made in corresponding diagrams.

5.3 Implementation

This is the actual phase where the software development takes place. It is also known and coding and verification phase. Base on the previous design of the project, all the software program is written. Meanwhile the software is tested to check if the correct output is received. All parts of the project is integrated as one system in the end of this phase.

5.4 Testing & Debugging

The test plan with series of tests, which are run to check the performance of the integrated system, is made is this phase. Then the test plan is followed to carry out all the bugs, after that the bugs is removed, the system is tested again until all the bugs are removed and the system runs exactly as the company`s requirements.

5.5 Updating & Deployment

In the end of the project, the company tutor may want some extra changes or enhancements .The system then can be updated in this phase. If the changes make the basic objective of the project different, however, all previous processes is restarted. After all needed updates have been made, the final product is deployed to the clients side.

In the following chapters, all the details that I have done during each development phase are described.

Chapter 6 Analysis and Requirements Gathering

6.1 Define the project assignment

In the first few weeks of this project, I had many meetings with the company tutor, which helps me to understand the project assignment and specified the mandatory requirements. Simply say, the instructors of the company need to prepare a corresponding set of virtual machines for each student during the training courses. Only the administrator, however, is able to manage the virtual machines in the virtualization environment. There was an existing tool called ‘Copy Application’, was manually used by the administrator to create or delete the virtual machines from the virtualization environment. My project assignment was to develop a ‘Scheduling System’ to replace the original ‘Copy Application’ that allows not only the administrator, but also the instructors to schedule the virtual machines in advance for the courses.(for more details, see chapter 3)

6.2 Specified requirements

After understanding the project assignment, the requirements of the project was specified. In the following paragraphs, the requirements description are divided into two parts: functional requirements and non functional requirements.

Functional Requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs, which includes:

- User Roles Supported: the ‘Scheduling System’ must support two kind of user roles, which are administrator and instructors.

- User Log in: to be able to use the ‘Scheduling System’ , both administrator and instructors are required to log in the system with a specific user account and password.

- Registration: only the administrator (after log on) is able to register a new account with password for both administrator role and instructor role.

- Profile Editor: after users logs in the system, the personal information are able to be edited, such as First Name, Last Name, E-mail etc. The password can be changed in the profile editor as well.

- New Schedule Creation: users must able to create the new schedule by filling the following properties:

1. Class Entity: is the prefix of the course name that indicates the type of the course. The class entity statically consists four types, which are “CITC”, ”CSTC”, ”CIAC” and “CSAC”. Users have to choose one of them. These strings are read from an XML file.

2. Account Code: is the user account name used to distinguish who created the schedule. It automatically shows the current user account and unable to change on instructors side. For administrator side, however, this field is able to choose among all users` account that means the administrator can create a schedule for other users.
3. Choose Master DataStores: the Master DataStores represent all distinct courses that the system currently have. Users choose a Master DataStore they want to use as template for creating the copies.
4. Course Number: users give the course number that is used to specify the course name among all other existing courses. Two same course numbers cannot coexist at the same time.
5. Course Instance Number: is a unique number used to specify the different courses that using the same Master DataStore. This number will be given automatically.
6. Work Spots: users give the number of copies that need to be created.
7. Start Time: users give the starting date and time of the course.
8. End Time: user give the end date and time of the course.

- Schedules Execution: To guarantee all schedules can be executed before the course starting date, the schedule is always be executed 7 days in advance. If the starting date time is within 7 days since the current date time, the schedule will be executed immediately. As soon as the schedule is executed, the ‘Scheduling System’ creates the virtual machines automatically in the vCenter according to the schedule`s properties. The system deletes the virtual machines automatically of this schedule after 2 days of its end time.

- Schedules update: the schedules can be updated or deleted before its starting date time. During the ‘Start Time’ and ‘End Time’, only two properties are able to be updated, which are adding more ‘Work Spots’ and changing the end date and time. In this case, the administrator is still able to delete the schedule, and the relevant virtual machines will also be deleted as well.

- Schedules View: the instructors are able to view their own schedules. The administrators are able to view schedules from all users.

- Users Management: The administrators are able to block, active and delete all the instructors` accounts.

- Master DataStores Binding: The administrator is able to bind the Master DataStores from the vCenter by using the ‘Scheduling System’.

Non Functional Requirement specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. The non functional requirements of the project includes:

- Accessibility: the final system can only be accessed when the users are connecting the company`s intranet or relevant network. In addition, the interface of ‘Scheduling System’ is accessible for all users and the background of the system is only accessible for the administrators.

- Platform Compatible: the final product must be able to install and run in the Microsoft Windows 7 and Windows Server 2008 operate systems. It may nice to be supported in others environment.

- Deployment: after the project has been completed, I will be responsible to deploy the “Scheduling System” for the company using.

- Documentation: A documentation of the entire “Scheduling System” must be delivered along with the final product.

- Back up: In the end of the project, the correct running “Scheduling System” is required to make a backup for the company.

6.3 Understanding Working Environment

After the project assignment and requirements was specified, the development of the project then could be truly started. The first thing needs to be done is getting familiar with the working environment.

When the project was started, the company provided me a company laptop with Microsoft Windows 7 operating system. Although I am only able to connect to the company`s intranet by using this laptop, the Microsoft Windows 7 was not the real operating system on which the project was developed. To connect to the vSphere vCenter server easier and more convenient, my company tutor created a separate virtual machine in the vCenter with Microsoft Windows Server 2008 operating system for me where the “Scheduling System” was developed during the project. To access this virtual machine, I learned how to use the Web Access which is one of the VMware vSphere components(see VMware vSpher 5.1 research).

It seems that getting familiar with the VMware vSphere environment is very important in this phase. According to the advise from the company tutor, however, it was not necessary to make a lot of research for the virtualization environment. The existing ‘Copy Application’ already contains the completed and working functions that can connect to the VMware vCenterand provides creation and deletion of the virtual machines, which could be reused. I was supposed to focus on studying the “Copy Application” to rebuild useful applications for the entire “Scheduling System”.

6.4 Project Analysis and Preparation

In order to design and implement the project, the requirements need to be analyzed to prepare the necessary tools in this phase.

First of all, the “Scheduling System” is required to be accessed by all users with a ‘local user log in’ requirement. Creating a website as the platform for users to make their schedules could be an option.
Secondly, a database that is used to store the data of the schedules should be build. This database can then also be used for later datamining.

Considering my programming experience, the asp.net web application is the best choice for me to build the platform which can easily connect to the database.

Furthermore, the original ‘Copy Application’ needs to be rebuild to get the input data from other ways instead of the GUI. Thus the rebuild ‘Copy Application’ no longer needs to be operated manually. Before analyzing the ‘Copy Application’, I spent a few days to study the basic knowledge about the Windows PowerShell from an online tutorial website[11]. From the tutorial, I learned the variables, function structure and some basic commands. Later on, when I focused on the ‘Copy Application’ script, I found that some input values of the ‘Copy Application’ were read from the XML properties file, such as the ‘mac address’, ‘WorkArea Prefix’, 'NetApp server’ etc. which are the static values used to create the copies of Master DataStores. Then I realized that the GUI could be replaced by reading all necessary values from the XML file. In this case, the ‘Copy Application’ is closed automatically in the end of running. So if the schedule`s properties are overwritten in a Properties XML file, the rebuild ‘Copy Application’ will be able to create or delete the relevant virtual machines in the vCenter. Due to the requirements ‘Schedule Execution’ and ‘Master DataStores’, two new PowerShell applications are needed to be build base on the ‘Copy Application’, which are:
1. “Create_Delete” application
2. ‘Master DataStores Binding’ application.

The return value of Master DataStoring could be saved into a ‘Master DataStores’ text file. The PowerGUI software which suits for editing and compiling the PowerShell script is required. This part is considered as the base of the Schedule System’, which can directly connect to the vCenter and fulfill the virtual machines` creation and deletion.

Finally, a central control application is needed to monitor the start time and end time of the schedules and send the schedules data to the rebuilt ‘Copy Application’ in a logical way. This can be the heart part of the entire system.

To sum up, there is some software needed to be prepared for the virtual machine that I worked on, which includes:

1. Visual Studio 2010: building the asp.net web application and other possible applications.

2. Microsoft SQL Server 2008: storing the necessary data. The reason why I chose this database was that I had a lot of experience on SQL database and it was also used in the company.

3. Internet Information Service(IIS) Configuration: due to the previous accessibility requirements, IIS is the best option used to host the web application.

4. Complete Script of ‘Copy Application’: needs to be analyzed to build the Create_Delete application and Mater DataStores Binding application.
.
5. PowerGUI: is used to edit and compile the PowerShell script.

Chapter 7. System design

After having analyzed the requirements in the previous phase, the web application, IIS(Internet Information Service), SQL database, Create_Delete application, Master DataStores Binding application, a Master DataSotres text file and a PropertiesXML file had been involved in the 'Scheduling System'. The one last part needs is a monitor application that can connect all the separate parts as a complete system. It must be able to do the following :

- Monitor the database, especially the 'Start Time' and 'End Time' of the schedules.

- Be able to transfer the corresponding schedule`s properties into a XML file as soon as the schedule`s 'Start Time' or 'End Time' is expired.

- Be able to start up the Create_Delete application and Mater DataStores Binding application.

- Be able to check if the Create_Delete application or Master DataStores Binding application is current running.

In order to keep the 'Scheduling System' running well, the monitor application has to be always running in the background. The best option, after research, is building the monitor application as the Windows Service, which is called Scheduling Service.
.

7.1 Components relationship

Up to now, all needed parts of the Scheduling System are specified. The next step is to assemble the components and make them to be connected in a logical way. The following figure shows the relations design of the Scheduling System.

As the figure 7.1 shows, all components of the Scheduling System are made in the virtual machine where I worked on. Due to the web application is hosted by the IIS of the virtual machine, all users are able to login the web application, create and manage the schedules. The data generated from the web application are stored in the database which is monitored by the Scheduling Service. The Scheduling Service is responsible to execute the user`s requests and return the feedback information to the database, where the web application can read and present for users.

Tasks execution: When the schedule is executed, the Scheduling Service reads the schedule`s properties from database, and overwrite into a XML properties file first. Then the Create_Delete applications is started and executed the corresponding commands by reading the values from the XML file. In addition, when the Master DataStores Binding application is executed, the return values are stored into a text file. As soon as the application is finished running, the Scheduling Service is read data from the text file and update it into the database.

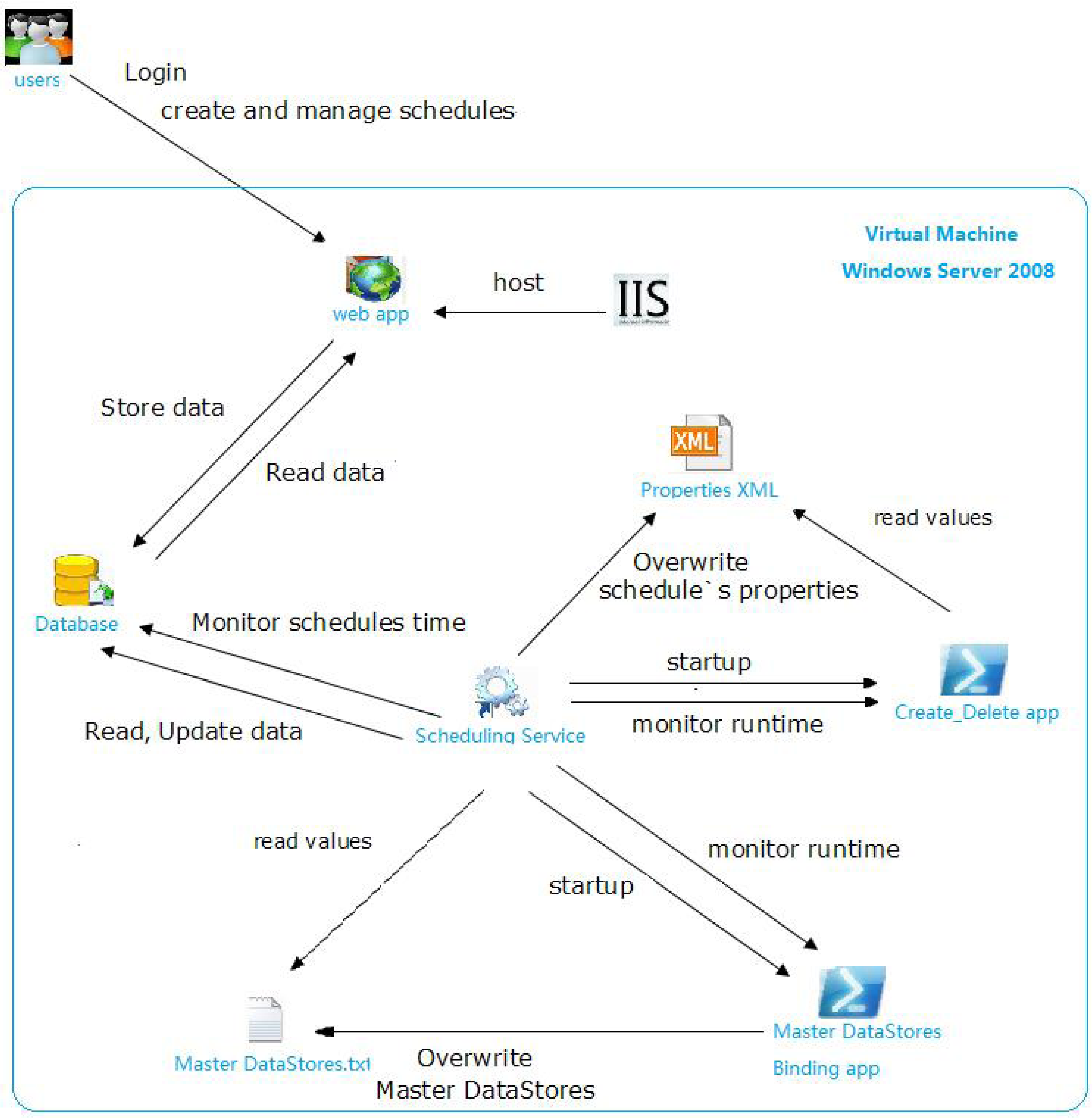


Figure 7.1 Scheduling System components connection

7.2 Web Applications Design

The Web Application is the GUI of the Scheduling System. It is also the only component where the interactions are taken place between users and the Scheduling System. So that the web application design must follow the functional requirements, which includes user roles supported, user login, registration, profile editor,new schedule creation, schedules update and user management; Accessibility and platform compatible of on functional requirements.

7.3 SQL Database Design

The database stores all necessary data of the Scheduling Application. According to the requirements, the database should at least contains user accounts, user profiles, schedules properties and Master DataStores information.

Scheduling Service Design: As the monitor of the Scheduling System, the Scheduling Service should contain two timers: a 'Start-End' timer and 'RunningTime' timer. The 'Start-End' is used to check the start time and end time of the schedules. if the time is expired, the timer will execute the corresponding commands. The 'RunningTime' timer is used to check the end time when the Create_Delete application or Master DataStores Binding application is started up. Because these two applications are running to communicate with vCenter, they can not start up together or start up twice at the same time. If there are more than one tasks need to be executed by these two applications at the same time, the 'Running Time' timer will make them start up one by one. In other aspects, the Scheduling Service contains all the control functions, such as overwrite data to XML file. Startup the applications, read and update data from database and so on.

CHAPTER 8. IMPLEMENTATION

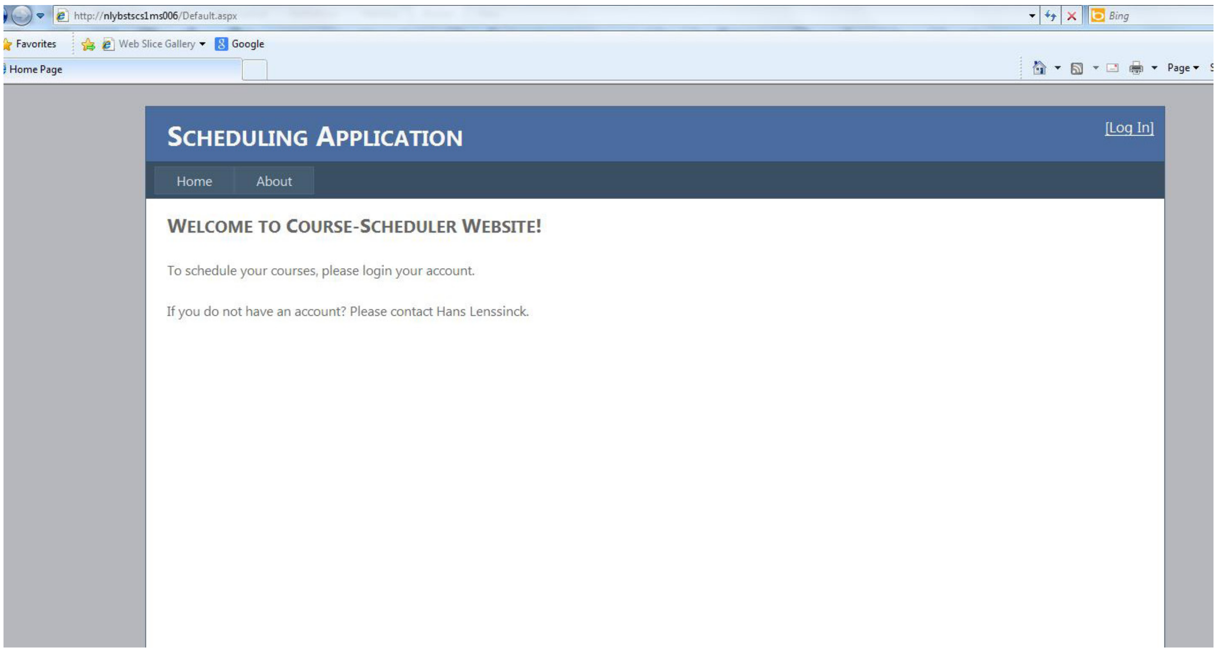
The implementation is to develop each separate component, and integrate them as a complete system according to previous requirements and design. During the implementation, the original design may have some small changes or improvement. In this chapter, The details about how each element of the Scheduling System was developed is given.

8.1 Web Application

I used Visual Studio to create the asp.net web application. During the development, I firstly created all needed pages according to the user requirements. After that, I integrated all pages by implement the background functions. Finally, I built the connection between the web application and the database

8.1.1 Structure & First pages

To make the web application to easily be used, I used a very simple and clear structure for each pages. The structure contains a title bar on the top, a navigation bar in the middle and the content frame in the bottom. When the page is changed, the structure remains the same and the content frame shows the corresponding content.



WELCOME TO COURSE-SCHEDULER WEBSITE!

To schedule your courses, please login your account.

If you do not have an account? Please contact Hans Lenssinck.

Figure 8.1.1 Home Page

As the figure 8.1.1 shows, for example, is the home page that firstly showed when user access to the web application. At this page, the blue bar with the 'SCHEDULING APPLICATION' is the title bar; the following darker one is the navigation bar and the white filed is the content frame. I put some general information here to let users get the first known about the web application. A notice is also given here to remind users log in the system before use It.

The login button is located on the top right corner. After user clicks on the button, the page is turned to login view(see figure 8.1.2). As mentioned above, the structure of the login page is exactly same as the home page. In the log in part, I created a function to check if the input username and password is correct, and to verify the user role. The instructors and administrators will see the different navigation bar after log in the system.

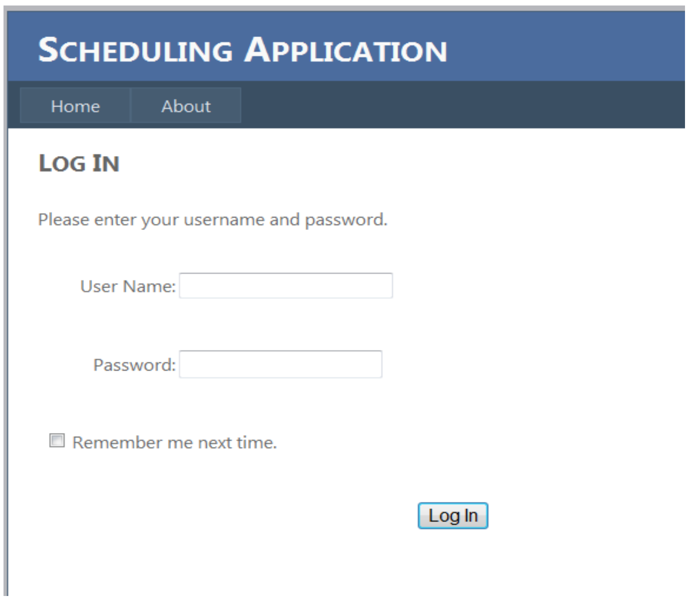
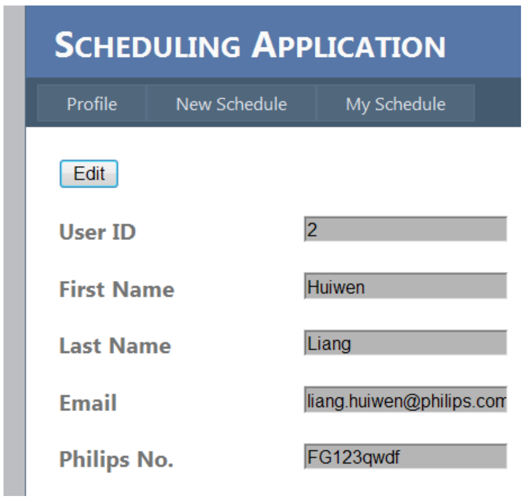


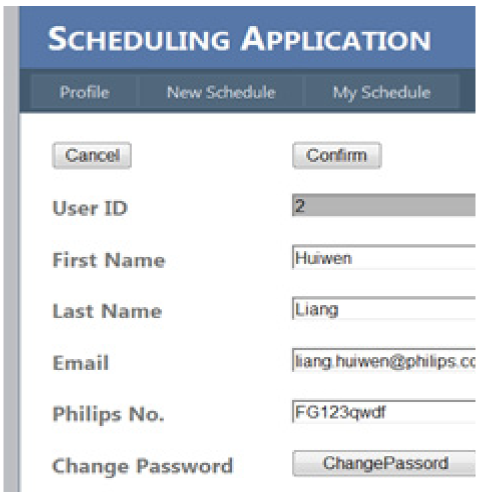
Figure 8.1.2 Log in view

8.1.2 Profile

The profile view is the first page after login the web application. User is able to midify their personal information and change the password at this page. In order to protect the profile data, I blocked all the text box as the defult set. The data will be modified after user clicks on the 'Edit' button(see figure 8.1.3).



User ID	2
First Name	Huiwen
Last Name	Liang
Email	liang.huiwen@philips.com
Philips No.	FG123qwdf



User ID	2
First Name	Huiwen
Last Name	Liang
Email	liang.huiwen@philips.cc
Philips No.	FG123qwdf
Change Password	ChangePassord

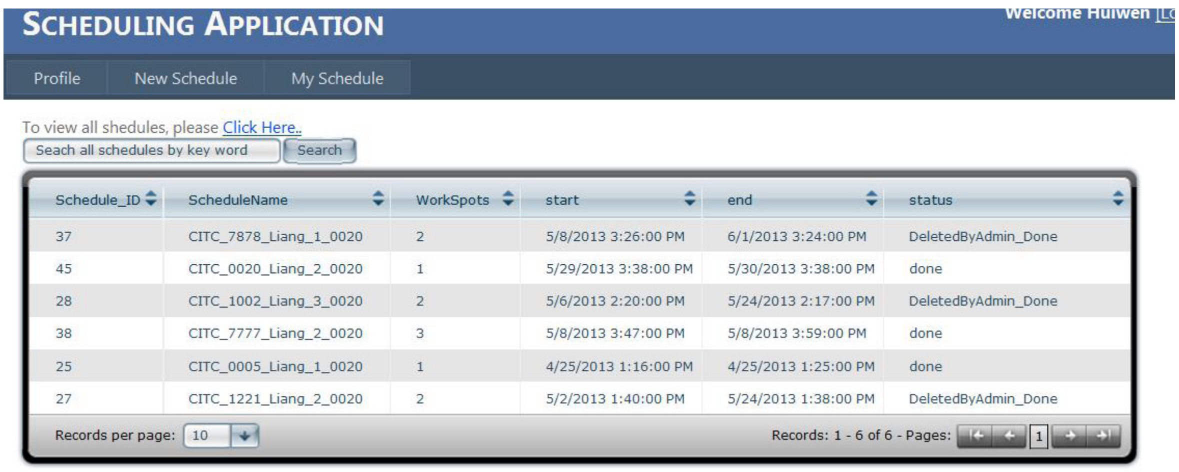
Figure 8.1.3Profile View

8.1.3 New Schedule

The New Schedule page is the most important part of the web application. It contains all the information about a schedule. In this page, user can create a new schedule by fully filling the corresponding data. I added some constraints on this page to make the schedule creation logically. Such the 'End Time' must be late than the 'Start Time'; the 'Workspots' fields must be a positive number. The instructors view on this part is different with the administrator, due to the administrator is able to create a new schedule for other users.

8.1.4 Schedule View

In the Schedule View , I created a grid view to show all the schedules that user can review and update the schedule`s properties. The grid view component was used from a third part called Obout(figure 8.1.5). The reason that I used the Obout grid was it has a nice interface, automatically reorder the columns, define pages and much more benefits compare with the original grid of asp.net. To find the specific schedule, I created small search engine above the data grid. User is able to search the schedule by typing a key word. Both administrators and instructors have a schedules view to review their schedules, but the administrators are able to view the schedules from all users(figure 8.1.4). To update a schedule properties, user need to select on the schedule, and give the confirmation.



To view all shedules, please Click Here..

Schedule_ID	ScheduleName	WorkSpots	start	end	status
37	CITC_7878_Liang_1_0020	2	5/8/2013 3:26:00 PM	6/1/2013 3:24:00 PM	DeletedByAdmin_Done
45	CITC_0020_Liang_2_0020	1	5/29/2013 3:38:00 PM	5/30/2013 3:38:00 PM	done
28	CITC_1002_Liang_3_0020	2	5/6/2013 2:20:00 PM	5/24/2013 2:17:00 PM	DeletedByAdmin_Done
38	CITC_7777_Liang_2_0020	3	5/8/2013 3:47:00 PM	5/8/2013 3:59:00 PM	done
25	CITC_0005_Liang_1_0020	1	4/25/2013 1:16:00 PM	4/25/2013 1:25:00 PM	done
27	CITC_1221_Liang_2_0020	2	5/2/2013 1:40:00 PM	5/24/2013 1:38:00 PM	DeletedByAdmin_Done

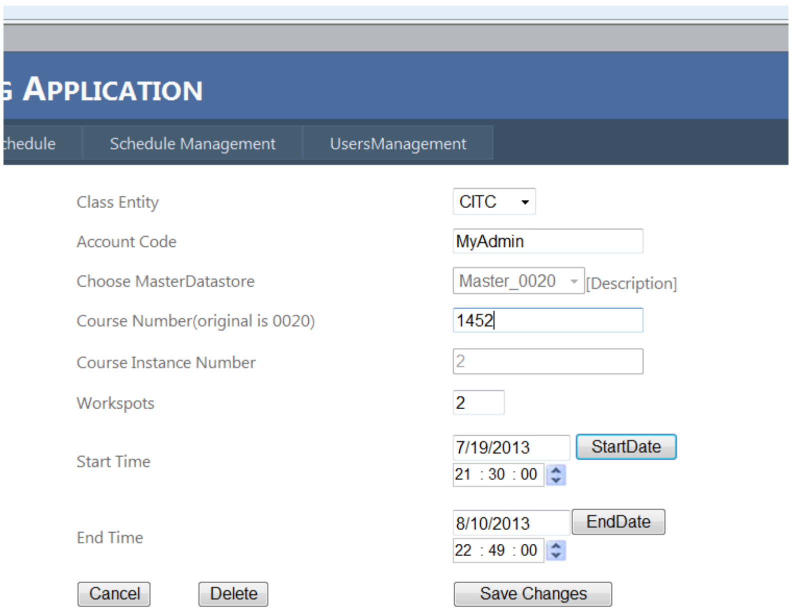
Records per page: 10   Records: 1 - 6 of 6 - Pages: 1

Figure 8.1.4 Update Confirmation

8.1.5 Schedule Update

After giving the confirmation in previous section,The update page is showed up. User is able to update the data from this page. The structure of the update page is same as the new schedule page. However, user can not to change the selected Master DataStores anymore.

In the case that if current date time is later than schedule`s start time, the 'Start Time' will be disable to changed and 'Delete' button will be disable as well. Only administrator can delete the schedules after it started.

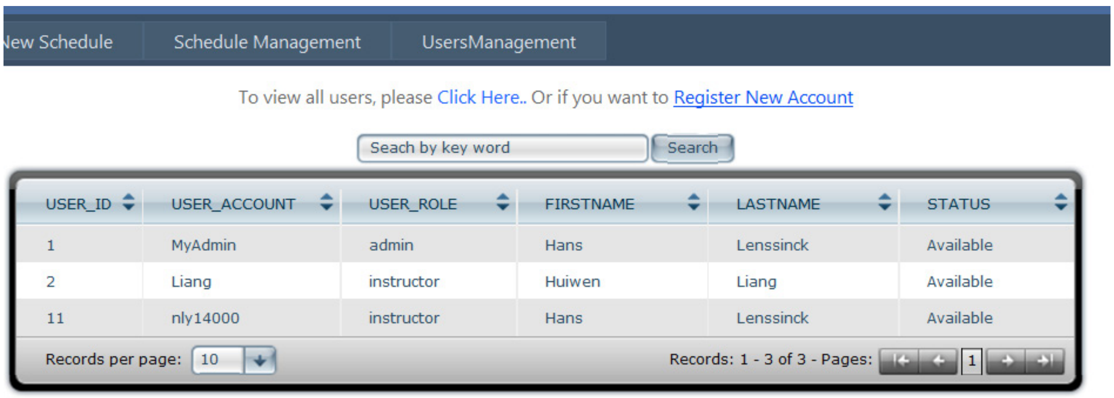


Class Entity	CITC
Account Code	MyAdmin
Choose MasterDatastore	Master_0020 [Description]
Course Number(original is 0020)	1452
Course Instance Number	2
Workspots	2
Start Time	7/19/2013 21 : 30 : 00
End Time	8/10/2013 22 : 49 : 00

Figure 8.1.5 Update View

8.1.6 User Management

This page is specifically for the administrator users. All registered user accounts are reviewed in this page(figure 8.1.6), and the administrator is able to block(figure 8.1.7), active and delete the account by selecting it. Meanwhile, a new account can be registered after user clicks on the 'Register New Account' link.



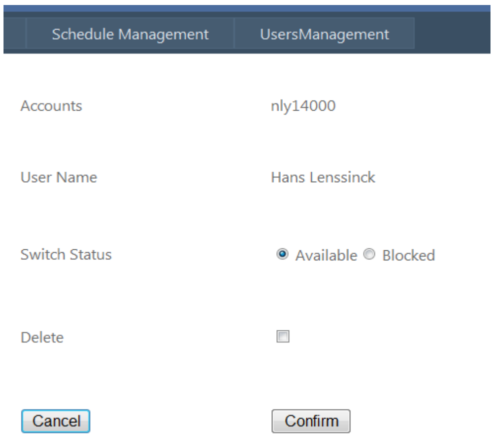
To view all users, please Click Here.. Or if you want to Register New Account

USER_ID	USER_ACCOUNT	USER_ROLE	FIRSTNAME	LASTNAME	STATUS
1	MyAdmin	admin	Hans	Lenssinck	Available
2	Liang	instructor	Huiwen	Liang	Available
11	nly14000	instructor	Hans	Lenssinck	Available

Records per page: 10   Records: 1 - 3 of 3 - Pages: 1

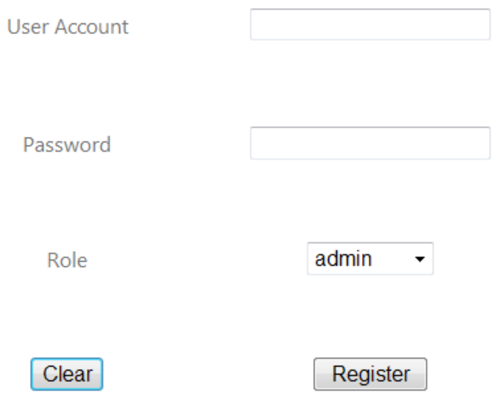
Figure 8.1.6 Users View

In the registration page, the administrator give the new account code, password and choose the user role to create a new account. The new account can not be creates when using the same user name.(figure 8.1.8)



Accounts	nly14000
User Name	Hans Lenssinck
Switch Status	[✓ Available] Blocked
Delete	☐

Figure 8.1.7 Update View



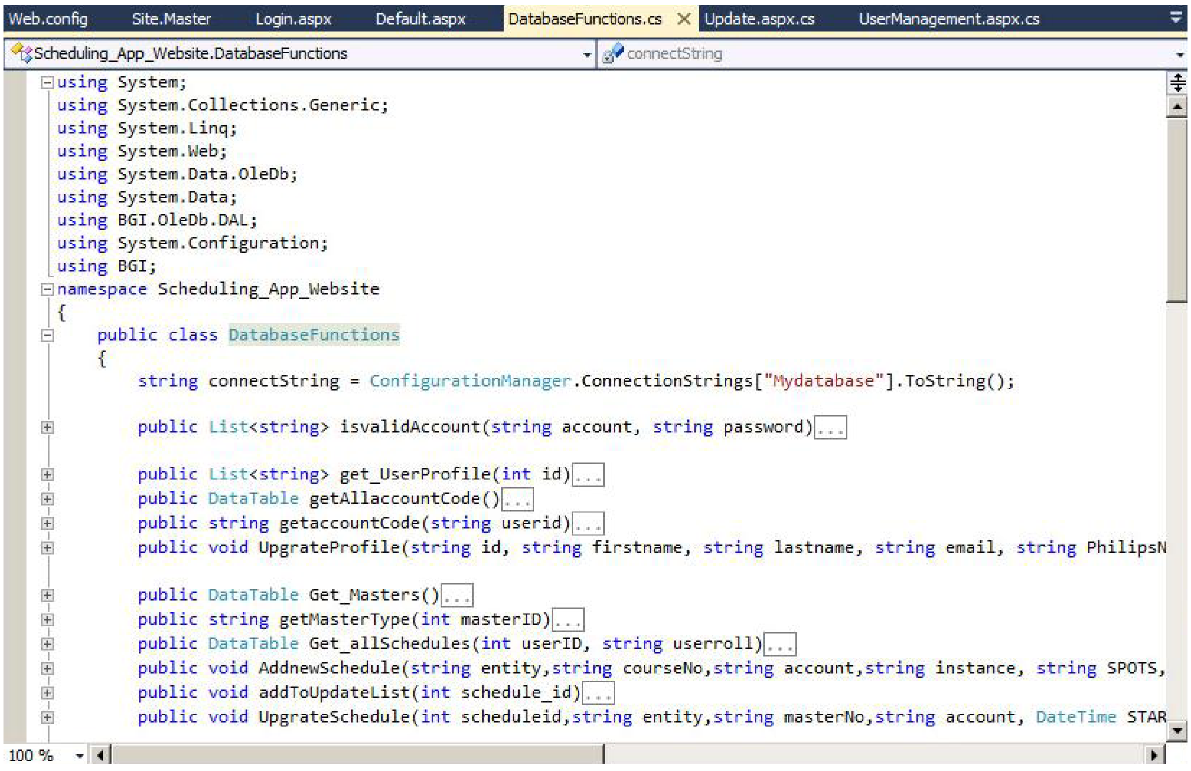
User Account	
Password	
Role	admin

Figure 8.1.8 Update View

In the case that if current date time is later than schedule`s start time, the ‘Start Time’ will be disable to changed and ’Delete’ button will be disable as well. Only administrator can delete the schedules after it started.

8.2 Connect to Database

In the web application development, I created a separate class to store all the functions that are used to communicate with the database. Thus the management of the functions is more easy and convenient. Although the database was used Microsoft SQL server, I used the ‘oledb’ query technique instead of normal SQL query to communicate with the database. To do so, the ‘BGI’ and ‘BGI.OleDb.DAL’ references need to be used in the Class.(see figure 8.2.1)



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data.OleDb;
using System.Data;
using BGI.OleDb.DAL;
using System.Configuration;
using BGI;
namespace Scheduling_App_Website
{
    public class DatabaseFunctions
    {
        string connectString = ConfigurationManager.ConnectionStrings["Mydatabase"].ToString();

        public List<string> isvalidAccount(string account, string password)...

        public List<string> get_UserProfile(int id)...
        public DataTable getAllaccountCode()...
        public string getaccountCode(string userid)...
        public void UpgrateProfile(string id, string firstname, string lastname, string email, string PhilipsN

        public DataTable Get_Masters()...
        public string getMasterType(int masterID)...
        public DataTable Get_allSchedules(int userID, string userroll)...
        public void AddnewSchedule(string entity,string courseNo,string account,string instance, string SPOTS,
        public void addToUpdateList(int schedule_id)...
        public void UpgrateSchedule(int scheduleid,string entity,string masterNo,string account, DateTime STAR
```

Figure 8.2.1 functions in the class

The ‘BGI.OleDb.DAL’ referenced was gotten during my last internship at the KSD company. The reference, which was developed by the KSD company, has a lot of advantages. For example, the question mark ‘?’ can be used to present the values in the query string. It makes the query string more easier to be edit. Moreover, the values in the query can be added with the columns name, which make sure the values are assigned correctly. Compare to SQL query, the oledb query execution contains more return types.

8.3 SQL Database

The figure 8.3.1 shows all the tables that been created in the Database. These tables can be divided as two categories: the mandatory tables and session tables. The mandatory tables are must be created to store the systems information, which includes an ‘ACCOUNTS’ table to store user account data, a ‘ALL_SCHEDULES’ table to keep the schedule`s properties, a ‘MASTERS’ to store all the existing Master DataStores` name and a ‘PROFILE’ table to save the users information. On the other hand, the session tables were created to store the relevant session ID which may be generated during the Scheduling system running. In this part, I created a ‘UpdateTable’ to store the schedule ID that been selected to update in the web application; as the same point, the ‘UpdateUserID’ table is used to store the user ID when the administrator manages the user accounts; the last table ‘WaitingList’ stores the ID of schedules that need to be execution. In the following paragraph, I give the details of each table.

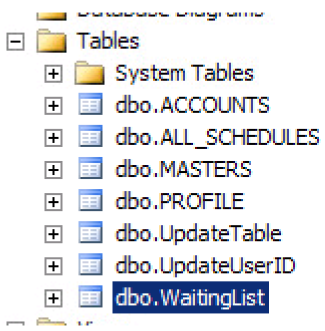
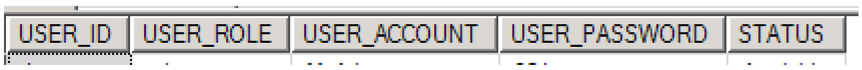

Tables
System Tables
dbo.ACCOUNTS
dbo.ALL_SCHEDULES
dbo.MASTERS
dbo.PROFILE
dbo.UpdateTable
dbo.UpdateUserID
dbo.WaitingList

Figure 8.3.1 Database

8.3.1 ACCOUNTS

The ‘ACCOUNTS’ table is related with the user login and registration pages of the web application. When the user logs in the web application, the ACCOUNTS table is checked to verify if the typed user name and password are correct, and the status is ‘AVAILABLE’. As well as when the administrator creates a new account, the relevant data is stored in this table.



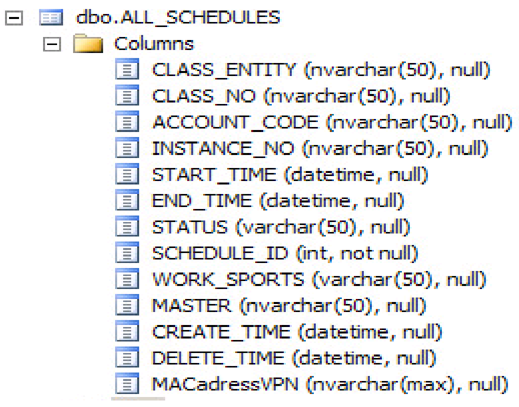
USER_ID	USER_ROLE	USER_ACCOUNT	USER_PASSWORD	STATUS

Figure 8.3.2 columns of Accounts table

The properties of this table is showed as figure 8.3.1 . The USER_ID is a unique number used to identify the user, which is referred as the foreign key in the PROFILE and UpdateUserID table. According to the users management in the web application, the administrator can block and active normal users. So that I added the 'STATUS' property that indicates if the user account is blocked for available. The user is not able to log in with the blocked status.

8.3.2 ALL_SCHEDULES

After a new schedule is created, the properties of schedule are stored into the ALL_SCHEDULES table. Besides, the 'ACCOUNT_CODE', 'CREATE_TIME', and 'DELETE_TIME' are also added in this table. The 'ACCOUNT_CODE' (username) specified the user who created the schedule. It could be used the USER_ID from
ACCOUNTS table, but the the username is also unique in the table and it needs to be used as a part of the schedule name. In addition, due to the schedules execution requirements, the 'CREATE_TIME' and 'DELETE_TIME' are used to record the real execution time of the schedule. These two properties are not showed for users.

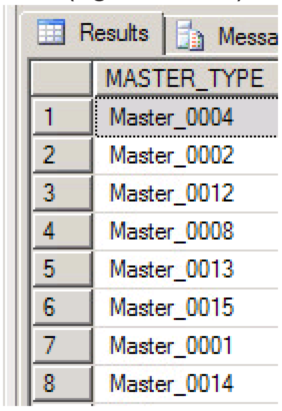


```
dbo.ALL_SCHEDULES
  Columns
    CLASS_ENTITY (nvarchar(50), null)
    CLASS_NO (nvarchar(50), null)
    ACCOUNT_CODE (nvarchar(50), null)
    INSTANCE_NO (nvarchar(50), null)
    START_TIME (datetime, null)
    END_TIME (datetime, null)
    STATUS (varchar(50), null)
    SCHEDULE_ID (int, not null)
    WORK_SPORTS (varchar(50), null)
    MASTER (nvarchar(50), null)
    CREATE_TIME (datetime, null)
    DELETE_TIME (datetime, null)
    MACadressVPN (nvarchar(max), null)
```

Figure 8.3.2 Schedule propoties

8.3.3 MASTERS

The 'MASTERS' table records all the current Master DataStores name in vCenter. Each time the Master DataStores Binding application is processed, it returns a list of Master DataStores name into a text file. After that, the Scheduling Service read the text file and overwrites the data into MASTERS table. (figure 8.3.3)



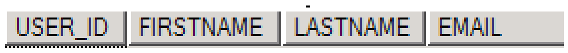
	MASTER_TYPE
1	Master_0004
2	Master_0002
3	Master_0012
4	Master_0008
5	Master_0013
6	Master_0015
7	Master_0001
8	Master_0014

Figure 8.3.3 content of Masters table

When user is creating a new schedule in th web application, all Master DataStore names are presented in a drop down list component.

8.3.4 PROFILE

he PROFILE table is simply storing the basic information.Because of the company can not specified what profile information is needed before using the Scheduling System. This table may be expanded in the future.



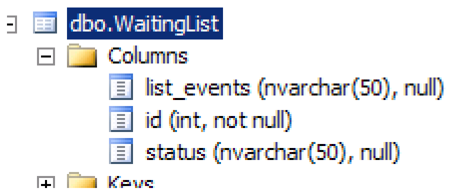
USER_ID	FIRSTNAME	LASTNAME	EMAIL

8.3.5 UpdateTable & UpdateUserID

Due to the weakness that using the Obout grid to show the schedules and users data, the parameters can not be transferred when user selected on a row. To simply say, when the user wants to update the information by selecting row on the Obout grid, the update page is showed, but the system does not know which schedule or user is needed to be updated. So I created these two tables to store the temporary schedule_id or user_id. When the user selected on a data row, the corresponding unique id is stored into the table. Then the web application, before the update page shows, reads the unique id from these tables and get its relative data. As soon as the update page is unloaded, the temporary id in the table is removed.

8.3.5 WaitingList

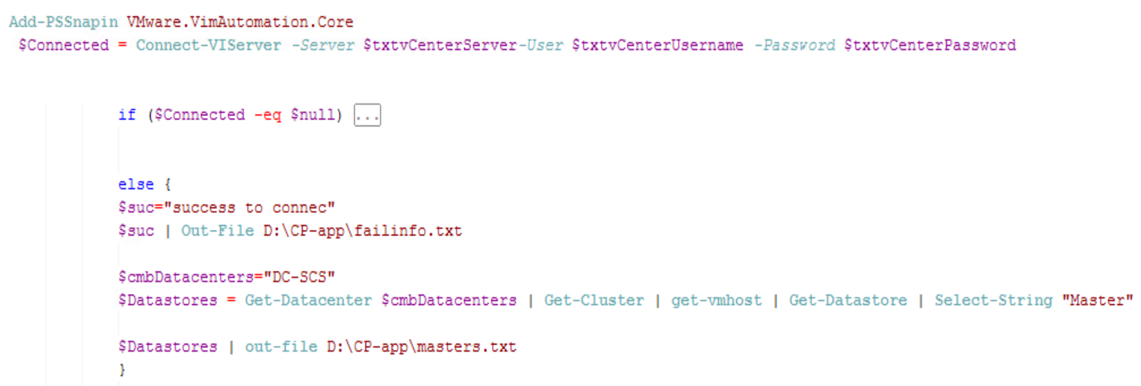
The ‘WaitingList’ table is created to store the Master DataStores binding and schedules execution commands that need to be executed immediately. As mentioned above, the the Master DataStores Binding application and Create_ Delete application can not start up at the same time. So I created this table to store the commands information which will be executed one by one in the ‘first come, first out’ sequence by the Scheduling Service. The WaittingList was first made as a temporary Array in the Scheduling Service. However considering the Array may disappear if the Service stopped or restarted, storing the data in a data table is more safe and stable.



```
dbo.WaitingList
  Columns
    list_events (nvarchar(50), null)
    id (int, not null)
    status (nvarchar(50), null)
  Keys
```

8.4 Master DataStores Binding Application

After studied and understood the code of the original ‘Copy Application’, I built the Master DataStores Binding application with a very short and simple code. Firstly, I referred the cmdlet of VMware to the application by add ‘Vmware. VimAutomation.Core’, which is shown as the first line of figure 8.4.1 .



```
Add-PSSnapin VMware.VimAutomation.Core
 $Connected = Connect-VIServer -Server $txtvCenterServer-User $txtvCenterUsername -Password $txtvCenterPassword

          if ($Connected -eq $null) ...


          else {
          $suc="success to connec"
          $suc | Out-File D:\CP-app\failinfo.txt

          $cmbDatacenters="DC-SCS"
          $Datastores = Get-Datacenter $cmbDatacenters | Get-Cluster | get-vmhost | Get-Datastore | Select-String "Master"

          $Datastores | out-file D:\CP-app\masters.txt
          }
```

Figure 8.4.1 Master DataStore Binding script

The VMware cmdlet contains all the functions that used to communicate with the vCenter. Such as the 'Connect-VIServer', on the second line, is the first function that used to connect the vCenter. The last two lines are used to get the current Master DataStores, and overwrite to the master.txt file. The code was edited on the PowerGUI software, which was also used to compile the script into an application. Thus the Master DataStores application can be started up by the Scheduling Service.

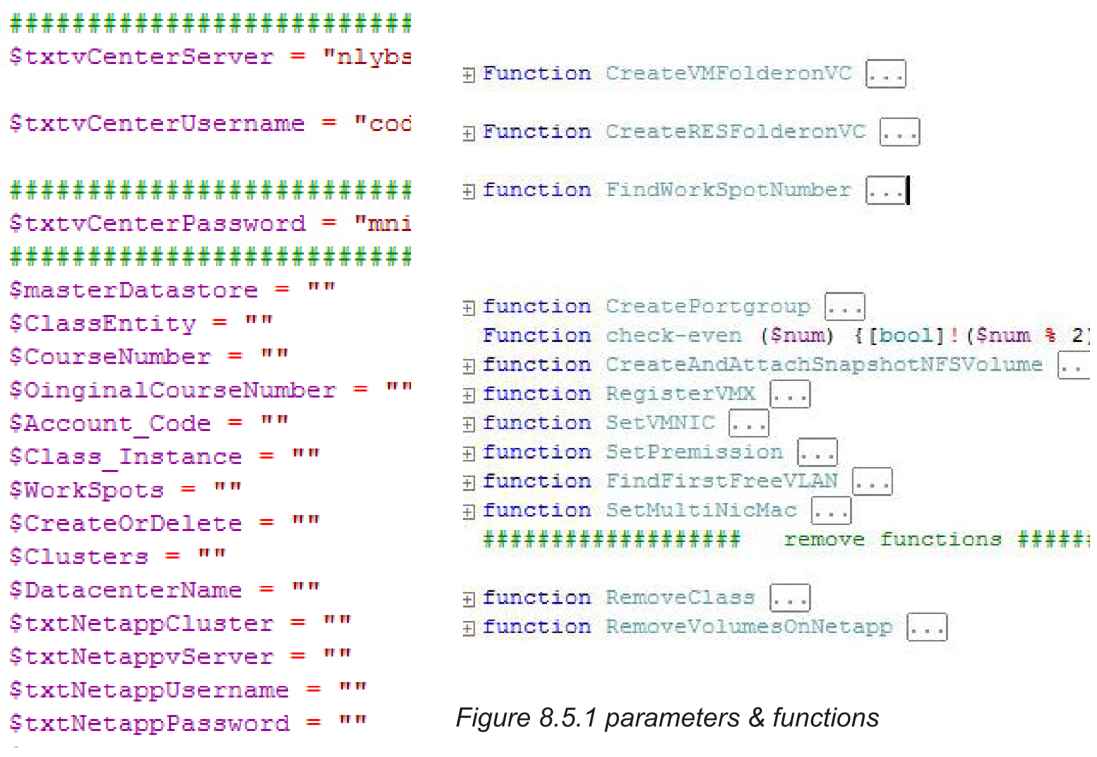
8.5 Create_Delete Application

The Create_Delete application is used to create and delete the copies of Master DataStores in vCenter. It directly reflects if the execution of schedules is successful or not. Comparing with the Master DataStores, this application is much more complex.

8.5.1 Split Out Functions

Before making this application, I dig into the script of Copy Application to find out all the mandatory parameters and functions, that needed to complete an process of the creation and deletion commands. After that, I copied all needed functions into the Create_Delete application script and declared the relevant parameters with the empty values.(see figure 8.5.1)

In the Copy Application, all these parameters` value were assigned by the GUI, which caused the value must be set manually. To avoid this situation, the parameters value must be assigned automatically when the Create_Delete application starts up. So the next step is making the input value from the XML file instead of the GUI.



```
##########################
$txtvCenterServer = "nlybs

$txtvCenterUsername = "cod

##########################
$txtvCenterPassword = "mni
##########################
$masterDatastore = ""
$ClassEntity = ""
$CourseNumber = ""
$OinginalCourseNumber = ""
$Account_Code = ""
$Class_Instance = ""
$WorkSpots = ""
$CreateOrDelete = ""
$Clusters = ""
$DatacenterName = ""
$txtNetappCluster = ""
$txtNetappvServer = ""
$txtNetappUsername = ""
$txtNetappPassword = ""
```

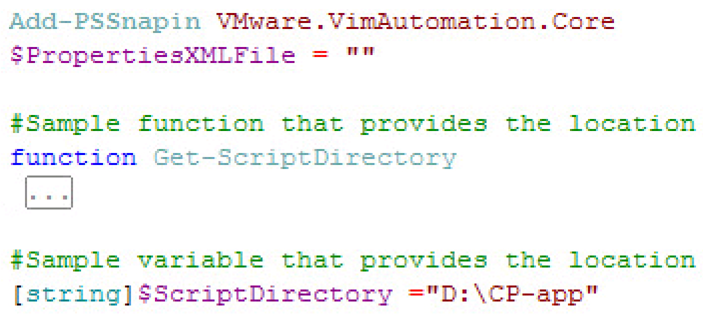
```
⊞ Function CreateVMFolderonVC [...]

⊞ Function CreateRESFolderonVC [...]

⊞ function FindWorkSpotNumber [...]


⊞ function CreatePortgroup [...]
  Function check-even ($num) {[bool]!($num % 2)
⊞ function CreateAndAttachSnapshotNFSVolume [..
⊞ function RegisterVMX [...]
⊞ function SetVMNIC [...]
⊞ function SetPremission [...]
⊞ function FindFirstFreeVLAN [...]
⊞ function SetMultiNicMac [...]
  ##################   remove functions #####

⊞ function RemoveClass [...]
⊞ function RemoveVolumesOnNetapp [...]
```

Figure 8.5.1 parameters & functions

In the Copy Application, all these parameters` value were assigned by the GUI, which caused the value must be set manually. To avoid this situation, the parameters value must be assigned automatically when the Create_Delete application starts up. So the next step is making the input value from the XML file instead of the GUI.

8.5.2 Read From XML File

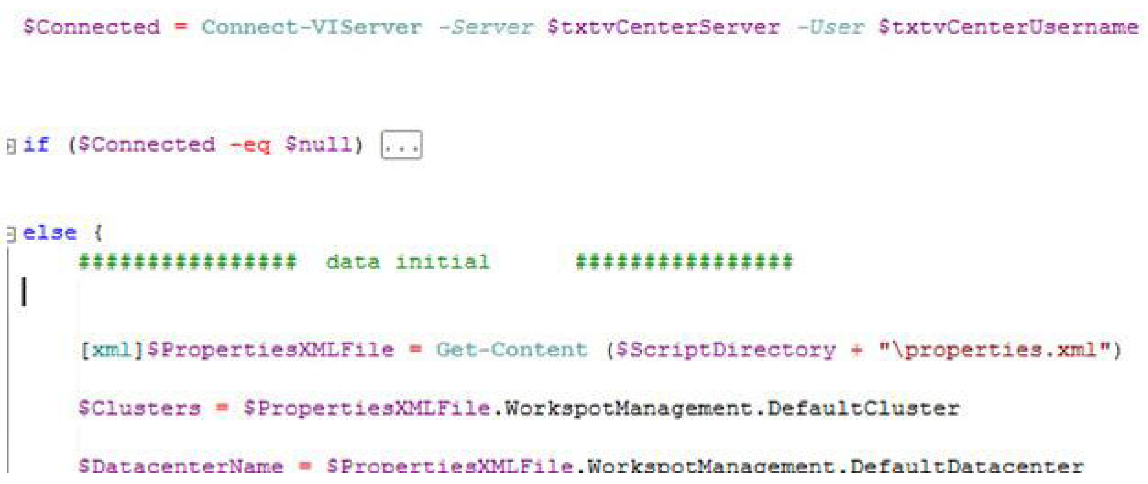
In this section, I declared a '$PropertiesXMLFile' parameter to store content of the properties XML file, and created a function to find the root of the XML file.(see figure 8.5.2)



```
Add-PSSnapin VMware.VimAutomation.Core
$PropertiesXMLFile = ""

#Sample function that provides the location
function Get-ScriptDirectory
 [...]

#Sample variable that provides the location
[string]$ScriptDirectory ="D:\CP-app"
```

Figure 8.5.2 propertiesXMLFile parameter

As soon as the vCenter is connected, I used the 'Get-Content' function to import the XML file`s content into the declared parameter '$PropertiesXMLFile'. Then all the necessary values can be assigned to the proper parameters. The figure 8.5.3 shows the code about reading XML File and assigning the values.

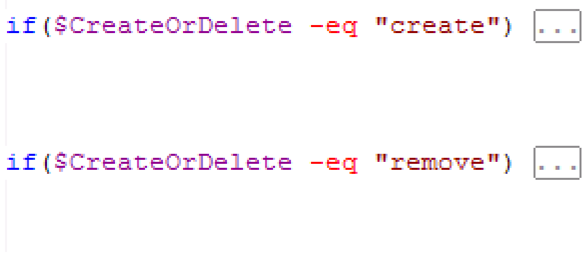


```
$Connected = Connect-VIServer -Server $txtvCenterServer -User $txtvCenterUsername

if ($Connected -eq $null) ...

else {
    ################  data initial     ################

    [xml]$PropertiesXMLFile = Get-Content ($ScriptDirectory + "\properties.xml")

    $Clusters = $PropertiesXMLFile.WorkspotManagement.DefaultCluster

    $DatacenterName = $PropertiesXMLFile.WorkspotManagement.DefaultDatacenter
```

Figure 8.5.3 Get XML file content

8.5.3 Verify Command Type

In the properties XML file(from previous step), there is a parameter called 'CreateOrDelete' , which indicates whether the application is run to create or delete the copies of Master DataStores. The functions are used in each command are different. So that I created the following functions to verify the command type.



```
if($CreateOrDelete -eq "create") ...


if($CreateOrDelete -eq "remove") ...
```

8.5.4 Create Command

If the command was verified as 'create', the following functions are executed. Firstly, a VM folder and Resources folder are created to store all the relevant stuff of the executed schedule. After that, the worksports(number of copies) are created one by one. For each workspots, a series of functions are used to create the folder, portgroup, snapshot, and set MAC address, permissions and so on. During studied the script of the original Copy Application, I found that the code of both creation and deletion parts can be reused in the Create_Delete application. So except the parameters of the functions were changed, most of code in creation and deletion part were copied from the Copy Application.

8.5.5 Deletion Command

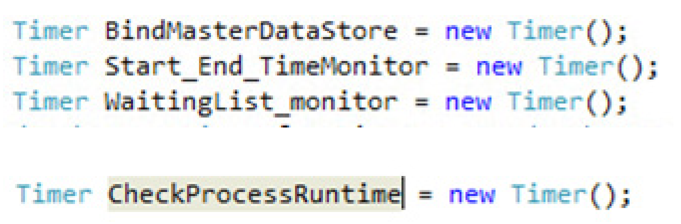
The code in the deletion part is very simple. After defined the folder name of the created schedule, a 'RemoveClass' function is called to remove the entire folder with the content in it. Then the other function 'Remove VolumesOnNetapp' is used to clean up all the revenant data of the schedule, such as resource pool, snapshots and etc.

8.5.6 Integrated and Debug

After finished the code editing, I integrated all the code and tested the script in the PowerGUI by manually giving the values in the properties XML file. In the first few times, it always executed errors. To find and resolve the errors, I put some breakpoint in the suspicious code line and used the debug function of the software to detect the problems. Until all the problems and bugs were fix, I used the PowerGUI software to compile the script into the application for the Scheduling System.

8.6 Scheduling Service

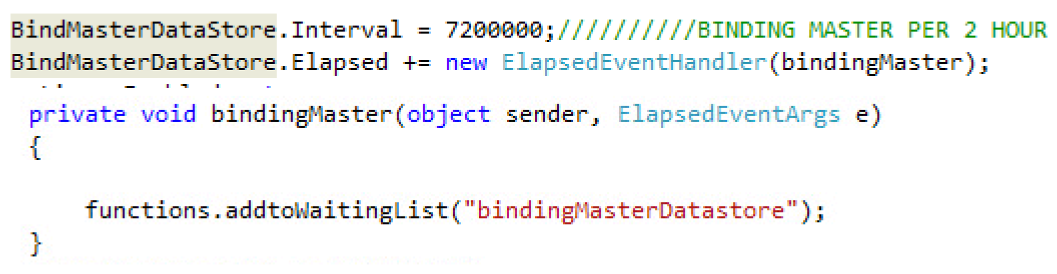
The scheduling service plays the role of the central controller in the Scheduling System. It was connected with almost all of other components, such as the database, properties XML file, Create_Delete application and so on. The Scheduling service is responsible to execute user commands and schedules in a correct and timely way. To make the service run logically, I created 4 Timers with corresponding functions in the service. They are shown in figure 8.7.1. In the C# programming, the Timer is used to set a event `s trigger with a certain time. For example, If I set the interval time as 10 seconds, and the execution event is starting up the notepad. After the Timer is run, it will start up a notepad each 10 seconds. In the following sections, I introduced what each timer is and how does it works.



```
Timer BindMasterDataStore = new Timer();
Timer Start_End_TimeMonitor = new Timer();
Timer WaitingList_monitor = new Timer();

Timer CheckProcessRuntime = new Timer();
```

8.6.1 BindMasterDataStore Timer

According to the user requirements, the Scheduling System must be able to automatically binding the Master DataStore from vCenter. This Timer is used to add the binding command to the WaitingList table in the database. Due to we never know when there has a changes in the vCenter, I set the interval of this Timer as two hours. Which means the system will automatically bind the Master DataStores once per two hours. (as the same as the web application, I created a class ’function’ to store all the necessary functions that are used to communicate with the database.)



```
BindMasterDataStore.Interval = 7200000;//////////BINDING MASTER PER 2 HOUR
BindMasterDataStore.Elapsed += new ElapsedEventHandler(bindingMaster);

 private void bindingMaster(object sender, ElapsedEventArgs e)
 {

     functions.addtoWaitingList("bindingMasterDatastore");
 }
```

8.6.2 Start_End_TimerMonitor Timer

The Start_End_TimerMonitor Timer is used to monitor the start time and end time of the schedules. If the time is expired, the relevant schedule ID will be stored into the the WaitingList table in the database. To avoid executing the same schedule commands twice, I changed the 'STATUS' of the schedule as soon as it been added into the Waiting List table.

8.6.3 CheckProcessRuntime Timer

When the Master DataStore Binding application or the Create_Delete application is started up, it will takes some time until the application is finished. Due to these two application can not neither run twice nor run at the same time, I created the CheckProcessRuntime Timer to check, from Windows Process Manager, if there is any process of these application is running. As soon as the process is finished, it will executes some corresponding functions that are introduced where the Timer is used in the next section.

8.6.4 WaitingList_monitor Timer

This Timer is used to check if there is any command existing in the WaitingList table. If so, the command will immediately be executed. If there are more than 1 commands stored in the table, the Timer will execute them one by one.

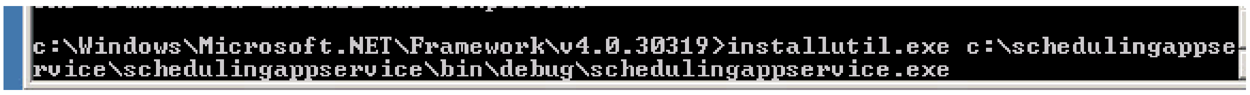
Due to the previous two Timers, the commands in the data table have two types that are executed by either the Master DataStore Binding application or the Create_Delete application. So that when one or more commands are detected by the Timer, the system will firstly check if there is any command`s status is 'running'. If there is no command is running, the first command is selected, and its status is changed from 'waiting' to 'running'. After that, the system checks if there is any process of the Master DataStore Binding application or the Create_Delete application existing in the Windows Process Manager. If the result is negative, the system will execute the command according to the command type.

If the command is executed by the Master DataStore application, the system will start the application up directly. Then the CheckProcessRuntime Timer is started to run.   As soon as the CheckProcessRuntime detects the application is finished, a 'ReadDataStores' function ,that I created in the service, will be called to read the return data from the Master DataStores text file and overwrite into the Master table in database. Lastly, the CheckProcessRuntime Timer is stopped and the command is removed.

On the other hand, if the command is executed by the Create_Delete application, the system will firstly gather all the relevant data of the schedule, and overwrite into the properties XML file. Then the application is started up and the CheckProcessRuntime Timer is turned on. As well as another command type, when the Timer detects the process is finished, the command is deleted and the Timer is turned off again. What the difference is that the schedule`s status is also changed in order to indicate that the schedule has been executed with this command.

After all the Timers were created, I made the BindMasterDataStore Timer, Start_End_TimerMonitor Timer and WaitingList_monitor Timer automatically started running when the Scheduling service starts up.

To install the scheduling service, I firstly built the scheduling service application and execute an 'installutil.exe' command with the proper root, which is shown as the figure 8.7.4.



```
c:\Windows\Microsoft.NET\Framework\v4.0.30319>installutil.exe c:\schedulingappse-
rvice\schedulingappservice\bin\debug\schedulingappservice.exe
```

Figure 8.7.4 Install service command

Filially, I set the service to run automatically when the operate system starts up.

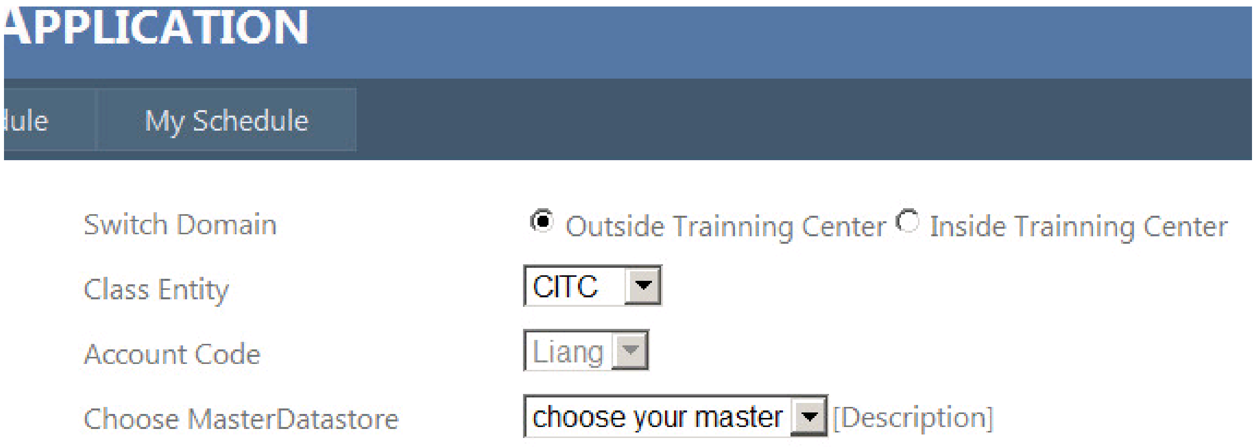
CHAPTER 9. TESTING & DEBUGGING

After all the components were created. I followed the requirements to make a series tests to check the functionality of the Scheduling System and remove all the detected bugs. See appendix C for more details about the test plan.

During the test plan was making, I found a defect on the Scheduling Service. Which was when the schedule is executing, the virtual machine may meet some accidents that leads to the result of the operate system is shut down or restarted. In this case, although the scheduling service will run again after the operate system starts up, there will be a command with the 'running' status in the WaitingList table which can not be finished anymore, and all other waiting commands can not be executed as well. To avoid this problem, I added a new function, called 'Assignment_Redo', into the service

CHAPTER 10. UPDATING & DEPLOYMENT

At the end of the project, I was requested to add a new feature into the Scheduling System which was when a new schedule is creating, the user should be able to switch the outside or inside domain of the course(see figure 10.1). This domain will cause using the different MAC address in the Create_Delete application.



Switch Domain	[✓ Outside Trainning Center] Inside Trainning Center
Class Entity	CITC
Account Code	Liang
Choose MasterDatastore	choose your master [Description]

Figure 10.1 Domain Switch

After spent a day to make the update changes, the final scheduling system had been completed. The last step that I need to do is deploying the Scheduling System for the company users. To achieve the accessibility of the requirements, I published the web application on the Internet Information Service(IIS), and started up the Scheduling Service.

CHAPTER 11. CONCLUSION

Overall, although the project was much more complex than I expected, I successfully developed the Scheduling System which achieved all the basic goals.

Looking back to the initial requests from the company, the Scheduling System must allow all users to independently schedule their courses in advance. Meanwhile, the schedules data need to be recorded into a database and the system must be able to automatically create and delete the VM`s in the vCenter according to the schedules` properties.

In the final Scheduling System, there is a web application that allows all users to login and manage their schedules. The schedules were designed to create with a specific start time and end time, thus user is able to schedule the courses in advance. Moreover, a SQL database is consisted in the Scheduling System. All the schedules properties and others necessary information are recorded in to the SQL database. From the web application, I created a page which users can review their own schedules. Lastly, a scheduling service and 'Create_ Delete' application, which are the main components of the Scheduling System, are responsible to monitor the schedule`s time and execute the corresponding commands in time.

The only pity thing of this project was I did not fulfil extra request by the end of the project, which was after the copies Master DataStores are deleted, the system will record the usage of CPU,network,storage and etc. Even though I did not complete this part, I found the access to visit the vCenter database which contains all the relevant data of the virtual machines. I believe this goal will be achieved in the future.

EVALUATION

I have learned many things after completed this project. Firstly, I got more experience on the techniques of C#, asp.net and SQL database. Especially the the skill of C# programming language, which was learned and used most at the school, has improved a lot. In addition, I learned many new things, such as the Internet Information Service(IIS), VMware vSphere virtualization, Windows PowerShell and Windows Service.

In the beginning, I thought the hardest part of this project for me was creating the Create_Delete application and the Binding Master DataStore application. Because the Windows PowerShell was quite new for me. I had to learn this new programming language before looking at the code of the original copy application. After I learned some basic knowledge of the PowerShell script, however, I found it was not that hard to understand the code of the copy application. So that I spent less time than expected to finish the Scheduling System.

Another interested thing that I learned was how does Windows Service work and how to build a service application by the Visual Studio 2010. I may be able to create some useful service for my operate system in the future. For instance, removing the junk files periodically or making an auto alert for my own schedules and etc.

Finally, looking back the process of the project development. I spent too much time on the system analyzing and requirements gathering. This was caused by the reason that I made a very cursory project plan in the beginning of the project. Now I realized the importance of making a good project plan before starting development of the project. I will pay more attention on project plan in the future project development.

APPENDIX A: GLOSSARY

VM: virtual machine
Esx: Elastic Sky X
Esxi: Elastic Sky X Integrated
Vcenter: virtual center
IIS: Internet Information Service
Cmdlet: command let
WSYIWYG: What you see is what you got
XML: eXtensible Markup Language
SQL: Structured Query Language

APPENDIX B: PROJECT PLAN

Introduction

This project plan is written for the graduation project which offered by Mr.Fred Mierlo from Philips Healthcare department. The project takes place at the department of Philips Medical Systems Nederlands B.V. in Best gebouw QS-2 on the term from Feb 11 till Jul 8 2013. During the period, Mr Hans Lenssinck will guide me through the project. The purpose of this project plan are following:
Specify the project assignment is and company expecting
Planning how to get start the project.
For university tutor will keep an eye on the content of the assignment and following the steps during the graduation project.

The project plan is divided into two main parts, project statement, project phasing. In the project statement, all the roles during the project and the assignment detail are described. The project phasing indicates all the needed activities to achieve the end of the project along with corresponding terms.

Project statement

- Formal client
  Mr Hans Lenssinck will be the formal client. I will discuss with him to work out what they expected in the end of the project and Hans will help me to work though the project.

- Project leader
  Due to I am the only person who will be responsible to fulfil whole project, so I am the the project leader of my self.

Current situation

To conveniently give the courses for Phillips trainees, the courses have been made based on the virtual machines. Each different virtual machine of the course is considered as the master data-store which is static and reliable. As soon as an instructor wants to start a course with 10 student for example, he needs to arrange 10 instances of such a master course for each student. Then he asks Mr Hans to make 10 instances of that master data-store who has the virtual machine copy app. The instances of the VM master data-store have to be configured manually by using the copy app which also need to be removed manually after the instances are not going to be used.

Project justification

The copy app now is the only one tool that can generate and remove the instance of master virtual machine data store which has to be done manually by Mr Hans. Once the number of master data store increased, there will be a lot of work need to deal with the copy app which is very inconvenient and wasting time. Further more, Hans wants some data to be stored in database, such as name of instructor who arranged the course, the number of courses, number of used storage etc. To avoid current problems and improve management of the courses, I was requested to do the followings :

- To build a scheduling system, which can be used to schedule the virtual machine courses in the future.
- Building a database to record corresponding information
- The system must be able to automatically create and delete the copies of corresponding master data stores in the vCenter according to the schedule`s time.
- Making a documentation of after scheduling system is built.

Project product

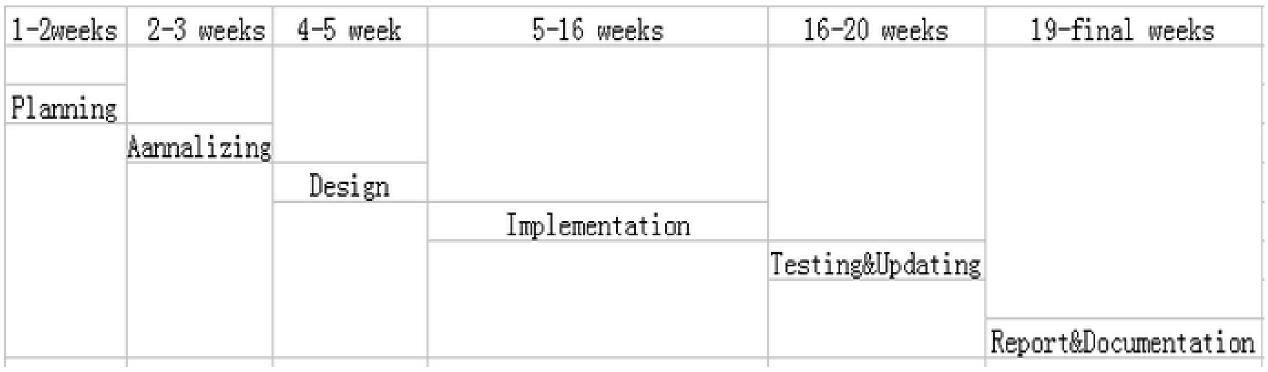
The final product will be a working scheduling system which at least contains the requested functions , and a documentation of the scheduling system

Project deliverables and non deliverables

Deliverable :
- The scheduling system,
- A documentation of the system
- All the relevant code.

Non deliverable:
- No comments.

Project phasing



1-2weeks	2-3 weeks	4-5 week	5-16 weeks	16-20 weeks	19-final weeks
Planning					
	Aannalizing				
		Design			
			Implementation		
				Testing&Updating	
					Report&Documentation

Planning:
- specifying the assignment, client`s expectation, setting up needed hardware and software and making the project plan..

Analysis and requirements
- Analyzing the current situation, and specifying the user requirements

Designing:
- designing the scheduling system and possible GUI.

Implementation:
- Implementing the scheduling system according to the requirements. and design

Testing and debugging:
- Testing the functions and removing bugs.

Report and documentation:
- Making the documentation for the application and finish the report for school.

APPENDIX C: TEST PLAN

Introduction

This document is written to test the functions in the windows service which is a crucial part of the scheduling system. As the project description above, the windows service plays the role to implement all schedules` assignments by monitoring the data from SQL database in the background. The service is considered as the heart of the scheduling system, thus the functions in the service must work in the proper way.

Resources required

To complete the test tasks, the following resources are required:

1. Asking admin to make sure that the virtual machine is power on and the scheduling windows service is running.

2. A computer with Philips network.

3. The scheduling web application domain name(http://nlybstscs1ms006/)

4. An account to login the scheduling web application

5. An administrator who can login and view the status in Virtual Center.

The resource 1 is to make sure that the scheduling system is all connected. Due to the scheduling web app was published on the Internet Information Service (IIS) base on Philips Internet in the virtual machine, a computer with Philips network is needed to connect to the scheduling web app. The address of the scheduling app is assigned as the name of the virtual machine, thus the domain name to achieve the scheduling app will be the name of the virtual machine. To start using the scheduling web app, an account which can be registered by administrator will be needed. Furthermore, all the test tasks have to be tested with an administrator who can check the result in Virtual Center.

Testing Tasks List

1.Schedule create
2.Admin creates schedule for other instructor.
3.Schedule delete
4.Schedule update: delete before its start time
5.Schedule update: delete in the ‘duration’ status by admin
6.Schedule update: add more work spots in the ‘duration’ status
7.Scheduling update: change the end time.

8.Multiple schedules create/delete at the same time
9.Using correct class instance number test
10.Using correct work area number test.
11.Register
12.System continuity
13.Binding Master-Data-Stores test

Test Goal, Processes, Expected Result and Result.

Task 1
Goal: To check if the class is created with the corresponding properties.
Processes: user login the scheduling web application, and creates a new schedule which starts in few minutes. As soon as the schedule`s start time is expired, the administrator checks if the class is generating in the vCenter.
Expected result: a number of corresponding Master DataStores will be created in the vCenter 7days in advance of the schedule`s start time. If the start time is less than 7 days, it will be created immediately.
Test result: as the expecting .

Task 2
Goal: To check if the administrator can create a schedule for the instructor.
Processes: Administrator login the scheduling web application, and creates a new schedule for another instructor by selecting another 'account code'.
Expected result: a new schedule will be added into database with an instructor`s account code. The instructor can view this schedule after login the web application
Test result: as the expecting

Task 3
Goal: To check if the class was removed automatically after 2days of the schedule`s end time.
Processes: creating a new schedule and set the proper end time. After 2 days of the end time, the administrator checks the vCenter if the class was removed.
Expected result: the schedule will be removed after 2days of the end time.
Test result: Due to 2 days was too long, I set the end time to 5 minutes later of the end time, and it was removed on time.

Task 4
Goal: To check if the schedule is totally deleted in the database.
Processes: user creates a new schedule and delete it before the schedule`s start time.
Expected result: Due to the schedule had not been executed, the schedule will be removed from database.
Test result: as expecting.

Task 5
Goal: After a schedule`s start time, the ‘status’ of the schedule will turn to ‘duration’ until its end time. In this case only administrator can delete the schedule. This test is to check if the admin can delete the schedule during its ‘duration’ status.
Processes: in the ‘schedule management’, administrator clicks on a schedule with the ‘duration’ status to open the update view, then administrator deletes the schedule.
Expected result: the schedule will be put into the deleting process immediately, and few minutes later, the corresponding class will be removed from vCenter.
Test result: as expecting

Task 6
Goal: to check if user can add more work spots after the class has been created.
Processes: in the update view of a schedule whose status is ‘duration’, user fills in the additional work area number, and saves the changes.
Expected result: the corresponding number of work area will be created immediately under the same class folder in vCenter.
Test result: as result

Task 7
Goal: to check if user can delay the end time of a schedule.
Processes: in the update view of a schedule whose status is ‘duration’, user changes the end time and saves the changes.
Expected result: the end time of the schedule will be changed as well in the database.
Test result: as the expecting

Task 8
Goal: to check if the processes works well when more than 1 schedule need to be processed at the same time.
Processes: create/delete two or more schedules at the same time.
Expected result: all processes will be put in the “waitingList” table in SQL database, and the processes will be executed 1 by 1.
Test result: as the expecting

Task 9
Goal: to check if the class instance number is distinct.
Processes: create two schedules using the same Master-Data-store.
Expected result: the class instance will always be different during all ‘active’ schedules.
Test result: as the expecting

Task 10
Goal: to check if the system assigns an available Work Area number to the classes in vCenter.
Processes: create two schedules with 1 work area and check their WA number; delete the first schedule and check the WA number; create another schedule with 2 work area and check the WA number.
Expected result: suppose the first schedule`s WA number is 1, then the second`s will be 2. As soon as the first schedule has been deleted, the WA number 1 is available again. So the WA number of the third schedule will be 1 and 3.
Test result: as the expecting

Task 11
Goal: check if the administrator can register an account for instructor.
Processes: Administrator log in the web application, and register an account. Administrator log out and login in with the new account.
Expected result: the new account is available to log in the web application.
Test result:as the expecting

Task 12
Goal: To check if the current running task start over when the Virtual Machine restarted.
Processes: Stop the service and restart it again; or restart the virtual machine.
Expected result: the unfinished task will be start over
Test result: as the expecting

Task 13
Goal: To check if the system can bind the mater data stores from vCenter when there is a new master data store is created.
Processes: First create a new master data store in vCenter. There are two ways to check if the system can bind the master data store: run the 'BindingMasterDatastores' application manually or waiting 2 hours, the system will automatically bind the masters from vCenter per 2 hours. After that, user login the web application and try to create a schedule with new Master DataStore.
Expected result: In the creating new schedule view, the new Master DataStores will exist in the master drop down list. User will be able to
Test result: as the expecting.

APPENDIX D: REFERENCE

[1]VMware vSphere Introduction
Copyright © 2009 VMware, Inc.
http://www.VMware.com/pdf/vsphere4/r40_u1/vsp_40_u1_intro_vs.pdf

[2] Explanation of waterfall model
http://searchsoftwarequality.techtarget.com/definition/waterfall-model

[3]Definition of C#
http://cplus.about.com/od/introductiontoprogramming/a/cshbeginners.htm

[4]Introduction to Asp.net
http://msdn.microsoft.com/en-us/library/4w3ex9c2(v=vs.71).aspx

[5] What is SQL
http://www.sqlcourse.com/intro.html

[6]What is Windows Service
http://www.7tutorials.com/what-are-windows-services-what-they-do-how-manage-them

[7]Windows PowerSehll
http://technet.microsoft.com/en-us/library/bb978526.aspx

[8]Information about Visual Studio
http://www.windowsdevcenter.com/pub/a/windows/2005/08/22/whatisVisualStudio.html

[9]Internet Information Service
http://www.iis.net/home

[10]About PowerGUI
http://wiki.powergui.org/index.php/FAQ

[11]Tutorial of Windows PowerShell
http://lesca.me/archives/powershell-tutorial-basics.html