



UNIVERSITY OF MALAYA

SARJANA MUDA SAINS KOMPUTER DENGAN KEPUJIAN

UM VIRTUAL NETWORK COMPUTING (UMVNC)

KOK SOO LEONG

WEK 020087

(SESSION 2004/2005)

PROJEK ILMIAH TAHAP AKHIR

Under supervision of

Mr. LIEW CHEE SUN

Moderated by

Mr. ANG TAN FONG

This project is submitted to the Faculty of Computer Science and Information Technology, University of Malaya in partial fulfillment of the requirement for the Degree of Bachelor of Computer Science

Abstract

UMVNC is an application for FSKTM staff. Remote controlling software such as UMVNC has a variety of uses. It allows a person at a remote computer to assume control of another computer across a network, as if they were sitting in front of the other computer.

This thesis project is aimed to assist the operation and management of faculty of Computer Science and Information Technology. Network administrator or network technician can easily use UMVNC to control other PC. UMVNC can help to troubleshoot the computer of a distant less-technically-savvy relative. UMVNC can also be used in educational contexts, for example to allow a distributed group of students simultaneously to view a computer screen being manipulated by an instructor, or to allow the instructor to take control of the students' computers to provide assistance.

This thesis project has been developed on Windows XP Professional and the software development tools - Microsoft Visual Studio .NET 2003 Professional Edition. I used C#.NET to develop and enhance the features of UMVNC.

It is hoped that FSKTM and University of Malaysia can use the Virtual Network Computing Application to manage the network computers. Beside that, I hope this UMVNC application will be further developed and enhance features in the future to explore more functionalities that can be offered so that it can actually be put into use.

Acknowledgement

Throughout the duration of preparing this thesis project's proposal, many people have been lending helping hands, giving invaluable advices and encouragements to me, which enable this proposal to be completed on time. Here in the acknowledgement of the project's proposal, I would like to thank a few persons that have been very helpful to me.

First of all, I would like to take this opportunity to express my sincere gratitude and appreciations to my respected supervisor, Mr. Liew Chee Sun, for his advice and supervision throughout this preparation of this project's proposal. Without his patience and proper guidance, I would not have been able to complete the proposal for this project successfully.

I also want to thanks my respected moderator, Mr. Ang Tan Fong because he was spending the valuable time to evaluate my proposal. The comments given by him is to make the application even more mature and stable.

Thanks to Department of Computer Systems & Technology, FSKTM and all the lecturers and staffs have been taught me during my 3 years study at FSKTM in University of Malaya.

I also want to thanks all my friends, even though we are not in the same project but we have been working together all the time, help each other and many more.

Last but not least, I like to thank my family especially my father and mother for their warmth and support throughout my years of studies in University of Malaya (financial support). Their encouragement and words of wisdom have truly motivated me and strengthen my spirits in performing my best in every field I am involved in. Once again, thank you very much!

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENT	iii
TABLE OF CONTENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER 1: INTRODUCTION	1
1.1 Overview of Virtual Network Computing (VNC) application	1
1.2 Project Motivation	3
1.3 Project Objectives	5
1.4 Project Scope	6
1.5 Expected Outcome	8
1.6 Project Limitations	9
1.7 Project Schedule	10
1.8 Summary	11
1.9 Report Organization	12
CHAPTER 2: LITERATURE REVIEW	14
2.1 What is VNC	14
2.1.1 Agent	15
2.2 TCP/IP, Socket, Port, Microsoft Windows Services	17
2.2.1 TCP/IP	18
2.2.2 Socket	21
2.2.3 Port	23
2.2.4 Microsoft Windows Services	24
2.3 Case Study	28
2.3.1 RealVNC	28
2.3.2 Access Remote PC	29
2.3.3 Remote Administrator 2.2	31
2.3.4 Windows XP Remote Desktop Connection software	34
2.4 Operating System	36
2.4.1 Microsoft Windows 2000	37
2.4.2 Microsoft Windows XP	39
2.4.3 Linux	40
2.4.4 Unix	42
2.5 Programming Language and Technologies	43
2.5.1 Java	43
2.5.2 .NET Framework	47

2.5.3 Introduction of Microsoft .NET	47
2.5.3.1 VB.NET	51
2.5.3.2 C#.NET	52
2.5.3.3 C++.NET	53
2.5.3.4 J#.NET	54
2.6 Tools Development	55
2.6.1 Microsoft Visual Studio.NET 2003 Professional Edition	55
2.7 Database	57
2.7.1 XML	57
2.7.2 MySQL (Database Management System)	59
2.7.3 Microsoft SQL Server 2000	60
2.7.4 Oracle	61
2.8 Summary	63
CHAPTER 3: METHODOLOGY	64
3.1 Introduction	64
3.2 System Development Methodology	66
3.2.1 Spiral Model	67
3.3 Fact Finding Techniques	70
3.3.1 Library Research	70
3.3.2 Internet Surfing	70
3.3.3 Brainstorm	71
3.3.4 Other Reference	71
3.4 Summary	73
CHAPTER 4: System Analysis	74
4.1 Introduction	74
4.2 Functional Requirements of the System	75
4.2.1 Remote Controlling	76
4.2.2 Remote Shutdown or Restart	76
4.2.3 Remote Warning System	77
4.2.4 Network Monitoring Tool	77
4.2.5 Logging System	77
4.2.6 System Tray	78
4.3 Non-Functional Requirements Analysis	79
4.3.1 User-Friendly	79
4.3.2 Flexibility	80
4.3.3 Security	80
4.3.4 Correctness	80
4.3.5 Functionality	80
4.3.6 Reliability and Accuracy	81
4.3.7 Efficiency	81
4.3.8 Maintainability	81
4.4 Decision on Choosing Technology, Software and Development Tool	82
4.4.1 Development Framework	82
4.4.2 Development Platform (Operating System)	84
4.4.3 Programming Language	85
4.4.4 Database	90
4.5 Development Requirements	91
4.6 User client Requirements	92

CHAPTER 5: System Design	93
5.1 Introduction	93
5.2 Overview of UMVNC System Architecture Design	94
5.3 System Functionality Design	96
5.3.1 System Structure Chart	96
5.3.2 Data Flow Diagram	97
5.3.2.1 Context Diagram for UMVNC	99
5.3.2.2 Data Flow Diagram for UMVNC	100
5.4 Database Design	101
5.4.1 Data Dictionary	101
5.5 User Interface Design	102
5.5.1 Concepts of User Interface Design	102
5.5.2 UMVNC Graphical User Interface	106
CHAPTER 6: System Implementation	107
6.1 Introduction	107
6.2 Development Environment	108
6.3 Coding	109
6.3.1 XML Database (Application Setting)	109
6.3.2 C#.NET Remoting (VNC)	110
6.3.3 C# UDP Programming	111
6.3.4 System Logging	114
CHAPTER 7: System Testing	117
7.1 Introduction	117
7.2 Unit Testing	118
7.3 Module Testing	119
7.4 System Integration Testing	120
CHAPTER 8: System Evaluation	121
8.1 Introduction	121
8.2 System Strengths	122
8.2.1 Easy-to-use Application	122
8.2.2 System Transparency	122
8.2.3 Easy to deploy the agent application at the network	122
8.2.4 Organized and User Friendly Interface	123
8.3 Problem Encountered and Solutions	124
8.3.1 Lack of knowledge	124
8.3.2 Difficult to track the security problem occurred	124
8.3.3 Difficulties to choosing a development platform, programming language and tools	125
8.4 System Limitations	126
8.4.1 Authentication	126
8.4.2 Lack of global clock system	126
8.4.3 Lack of cancel action module	126
8.5 Future Enhancements	127
8.5.1 Authentication	127
8.5.2 Global clock system	127
8.5.3 Cancel action module	127

8.6 Conclusion	128
Reference	129
User manual	130

University of Malaya

LIST OF TABLES

Table 2.1	Agent Classifications	16
Table 4.1	Comparison of Microsoft .NET and J2EE	82
Table 5.1	Description of symbols used in DFD	97
Table 5.2	Description of symbols used in DFD (supplemental)	98
Table 5.3	XML Schema for Logging System	101
Table 6.1	The Development Tools	108
Table 6.2	Application Setting for UM Remote Agent	109
Table 6.3	.NET Remoting Coding	110
Table 6.4	UDP coding for remote shutdown, restart, logoff, Announcement	111
Table 6.5	System Logging Coding	114

List of Figures

Figure 1.1	Project Schedule	10
Figure 2.1	VNC Working Environment	14
Figure 2.2	Layers in the TCP/IP protocol architecture	19
Figure 2.3	Data encapsulation	19
Figure 2.4	Data structures	21
Figure 2.5	Connection Request	22
Figure 2.6	Connection Establish	22
Figure 2.7	The socket interface	23
Figure 2.8	RealVNC ScreenShot	28
Figure 2.9	Logo of RealVNC	28
Figure 2.10	Access Remote PC ScreenShot	30
Figure 2.11	Remote Administrator 2.2 ScreenShot	31
Figure 2.12	Windows XP Remote Desktop Connection software ScreenShot	34
Figure 2.13	Microsoft Windows 2000 ScreenShot	38
Figure 2.14	Microsoft Windows XP ScreenShot	39
Figure 2.15	Linux ScreenShot (running KDE Desktop)	41
Figure 2.16	.NET Framework	47
Figure 2.17	.NET Compilation	49
Figure 2.18	VB.NET	51
Figure 2.19	C#.NET	52
Figure 2.20	C++.NET	53
Figure 2.21	J#.NET	54
Figure 2.22	Microsoft Visual Studio.NET 2003 Professional Edition	55
Figure 2.23	MS Visual Studio.NET Working Environment	56

Figure 2.24	Logo of MySQL	58
Figure 3.1	System Development Process Model	65
Figure 3.2	Spiral Model	67
Figure 4.1	Scope of the Project	75
Figure 5.1	UMVNC System Architecture	94
Figure 5.2	Work Flow Diagram of UMVNC	95
Figure 5.3	System Structure Chart	96
Figure 5.4	Context Diagram for UMVNC	99
Figure 5.5	Data Flow Diagram for UMVNC	100
Figure 5.6	UMVNC Graphical User Interface	106
Figure 7.1	The Testing Process Flowchart	120

CHAPTER 1 INTRODUCTION

1.1 Overview of Virtual Network Computing (UMVNC) application

Virtual Network Computing (UMVNC) application are new in the field of computing science. With VNC, the user can remote access to any computers at the campus or computer lab which the computer has a agent running behind!. The user can access the computers from a remote location.

Networks (and network programming) have come a long way over the past 20 years. In the early days of network computing (the '80s), network programming was left to the advanced programmer, who typically built applications using the C programming language in (mostly) Unix environments.

Now, networks are everywhere, from large corporations to small home users. With so many computers connected together via networks, network-aware applications are an accepted necessity. Existing applications must incorporate network features to stay competitive in the marketplace, and adding network communication to applications is essential.

Network programs are used for everything from children's games to advanced corporate database systems.

Some of Features of VNC Application

- Remote tools let you run vital utilities remotely to solve problems and modify settings.
- Utilities you can use remotely include Command Prompt.
- Other remote capabilities shutting down, rebooting workstations, logging off users

Network programming has always been a key feature of the Microsoft Windows operating system. Unfortunately, you've had to know advanced C or C++ programming concepts to utilize the network programming features in Windows programs. Now, though, the .NET Framework languages simplify the task of adding network features to your applications. The .NET libraries provide many network classes that can integrate network programming.

The C# language has solved many of my network programming problems by allowing me to quickly prototype and deploy network applications using C# classes. Combining the C# Forms library to write the graphical code with the C# Socket library to write the networking code makes creating professional network applications simple. With C# network classes, what used to take a day to write often only takes an hour or less.

1.2 Project Motivation

The world before the invention of the network protocol (TCP/IP, UDP), before the rise of the Internet, Computer systems were usually standalone entities with no access to other systems. Applications were designed to use on one machine (workstation).

There was very little communication via computer within or between companies. Even interpersonal messages often had to be delivered by hand. The advent of networking have changed the way of information sharing and distributed computing helped millions of people to get involved in computing especially for the business purposes.

VNC empowers users to act as servers or clients. This has given a lot of control to common users or network administrator to control and manage the computer remotely without having to go to the target computer. What he has to do is launch the VNC application, then the VNC application will connect to the remote computer (with agent running behind). So, it will make the work more easy for network administrator or end users.

Actually I known this type of remote application when I was doing my industrial training at eGENTING SDN BHD during 16/2/2003 – 4/6/2004. My Supervisor mainly use the VNC application to control and manage the server without enter the server room! The invention of network application was make the people live more easily.

Due to the benefits and advantages of using VNC technologies, A **UMVNC** application is built to help network administrator or end user to manage or solve the typical problems of VNC applications.

University of Malaya

1.3 Project Objectives

This section will list the various objectives to be achieved in this project. The main objectives are as follow:

- To build a convenient and useful VNC application, it is meant to be use in University Malaya Local Area Network.
- To allow the network administrator or network technician to monitor what the students do at computer lab (view the remote computer screen) if he or she suspect the student.
- To help Lab technician reduces the time when shutdown each computer manually before leaving the computer lab.
- To allow network administrator to send warning message to the client computer
- To create a user-friendly VNC application.

1.4 Project Scope

This project will cover on the local area network of University of Malaya itself. It will bring attention to some stakeholders such as the faculty's staff and administrators, Head of Departments, Dean and students as users. The network administrator or network technician can easily manage all the computer in the lab.

Below are the scope of UMVNC:

- **Remote Controlling**

The network administrator or network technician can remotely control the target machine (computer), monitor the student's activity at computer lab. He also can change the setting of the remote host by using UMVNC application. This application help them to manage the network computer easily.

- **Remote Shutdown or Restart**

Remote Shutdown or Restart is one of the great feature integrated into UMVNC. It allow the network administrator or network technician to restart or shutdown all the computer remotely. With this feature the network administrator or network technician no need to shutdown computers manually. At the result, it's save time to manage the network computer.

- **Remote Warning System**

The network administrator or network technician can send a warning message to all the computer at local area network. At the client side, a warning message box will appear to let them know that the warning message sent by the network administrator or network technician.

- **Network Monitoring**

The network administrator or network technician are capable of monitoring the download and the upload speed, such as the transfer rate of the current network activity. Besides that, they network monitoring will calculate all the download and upload bytes.

- **Logging System**

The logging system is created to let network administrator or network technician to check what action has been performed last time. The logging system will store the data like what and when the action have been performed.

- **System Tray**

The UMVNC has a function to hide the application at the System Tray. A icon will be display at the System Tray.

1.5 Expected Outcome

- This VNC application,UMVNC application will be designed in a good Graphical User Interface (GUI) to let the network administrator or network technician use it easily without the study of the user manual extensively. An interactive, attractive and interesting presentation of user interface.

- UMVNC will improve the management of the network computer at the UM especially at Local Area Network (LAN).

- The remote functionality will make the work done easily without having to go to the target computer.

- To prove that the network functionality is the cutting-edge technology at the computer revolution in this new arena.

1.6 Project Limitations

- Inexperience skill

Since C#.NET is the latest technology and there are limitations especially in personal skill when using it to develop a whole new system.

- Time Constraint

This is a research project, so much time is needed to analyze and study on developing the system based on pure VNC technology and the mechanism. Besides that this project has to be handed out at next year 2005. Besides that, sufficient information has to be gathered before starting the project.

- Resources

Reference books for C#.NET are quite expensive in the market today. Resources such as coding and information about the latest technology hardly can be obtained. So the resource mainly gather from the internet and MSDN

- Human Resources

This project is individual project. I have to complete the project as its project scope is too large. This limitation will cause the system to be developed in a rush condition in order to develop it on time.

1.7 Project Schedule

In order to achieve the project objectives, a project schedule was determined to manage the time and tasks that must be accomplished with the development phases. Figure 1.1 shows the project schedule. With the proper project schedule planning, it will ensure that the project development process works out smoothly and leads to success. Figure shows the work breakdown and time allocation for each job that has to be done.

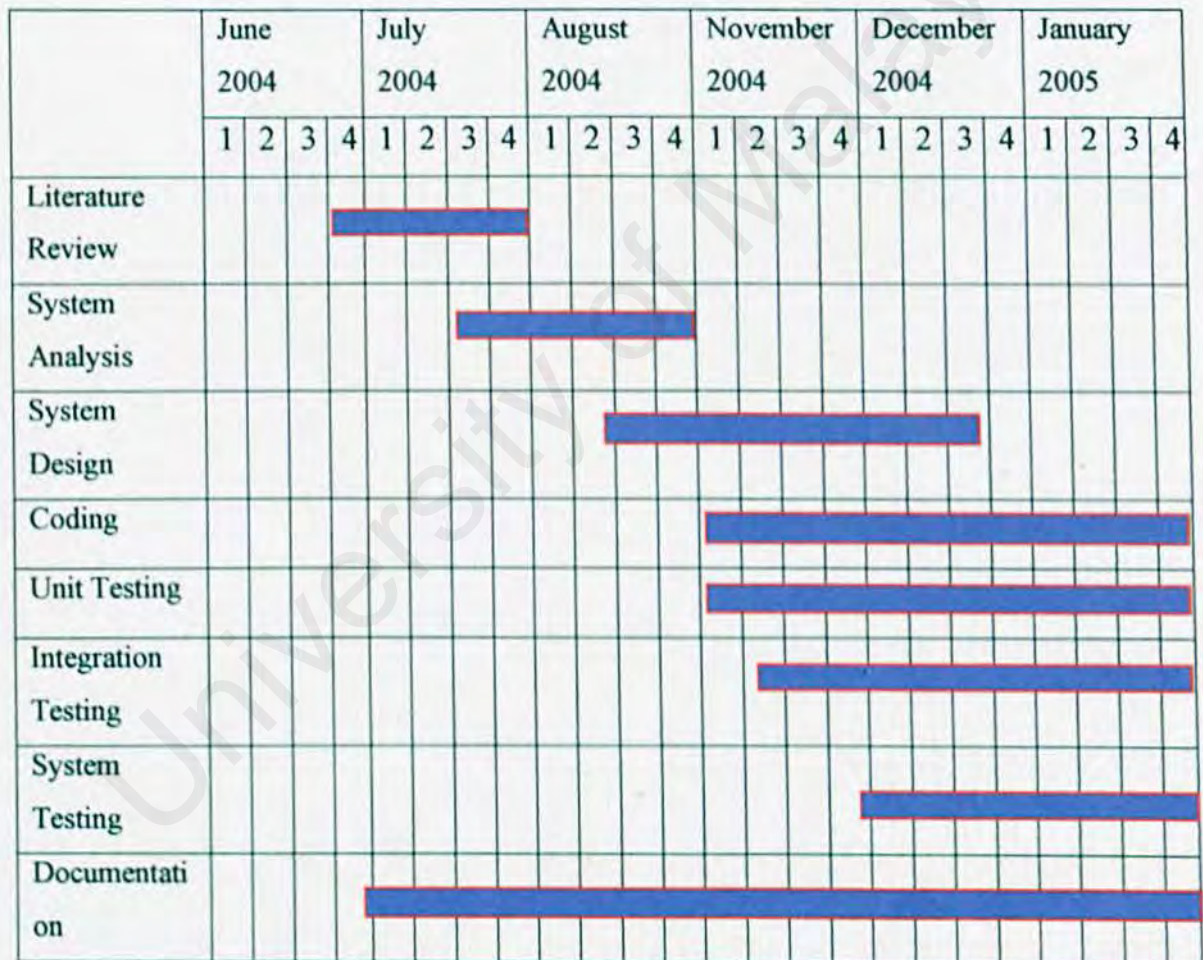


Figure 1.1: Project Schedule

1.8 Summary

Chapter 1 introduce the entire project, it includes the overview of VNC application, definition of the project, project motivation, project objective, project scope, expected project outcome and project schedule.

University of Malaya

1.9 Report Organization

This section contains brief introductory information of every chapter in this report to ensure a better understanding among readers about this project.

Chapter 1: Introduction

- gives a brief introduction to the whole project including the project overview, project motivation, project objectives, project scope, expected outcome, project limitations and project schedule to show the timeline of the project.

Chapter 2: Literature Review

- studies on the events and problems that will occur before the project begins. It cores the results of literature survey on several existing VNC system. Besides that this chapter reviews the tools and languages that use to develop UMVNC.

Chapter 3: Methodology

- discuss the methodology in developing the system, and various techniques in solving problems throughout the whole development process.

Chapter 4: System Analysis

- discuss in details for the system requirements such as functional requirements, non-functional requirements, hardware and software requirements. It also explains the considerations in choosing various development tools and programming languages for this project.

Chapter 5: System Design

- provides details regarding the system design, database design and user interface design.

University of Malaya

CHAPTER 2 LITERATURE REVIEW

2.1 What is VNC?

VNC stands for Virtual Network Computing. It is remote control software which allows you to view and interact with one computer (or the "server") using a simple program (the "viewer") on another computer anywhere on the Internet.

The two computers don't even have to be the same type, so for example you can use VNC to view an office Linux machine on your Windows PC at home. VNC is freely and publicly available and is in widespread active use by millions throughout industry, academia and privately.

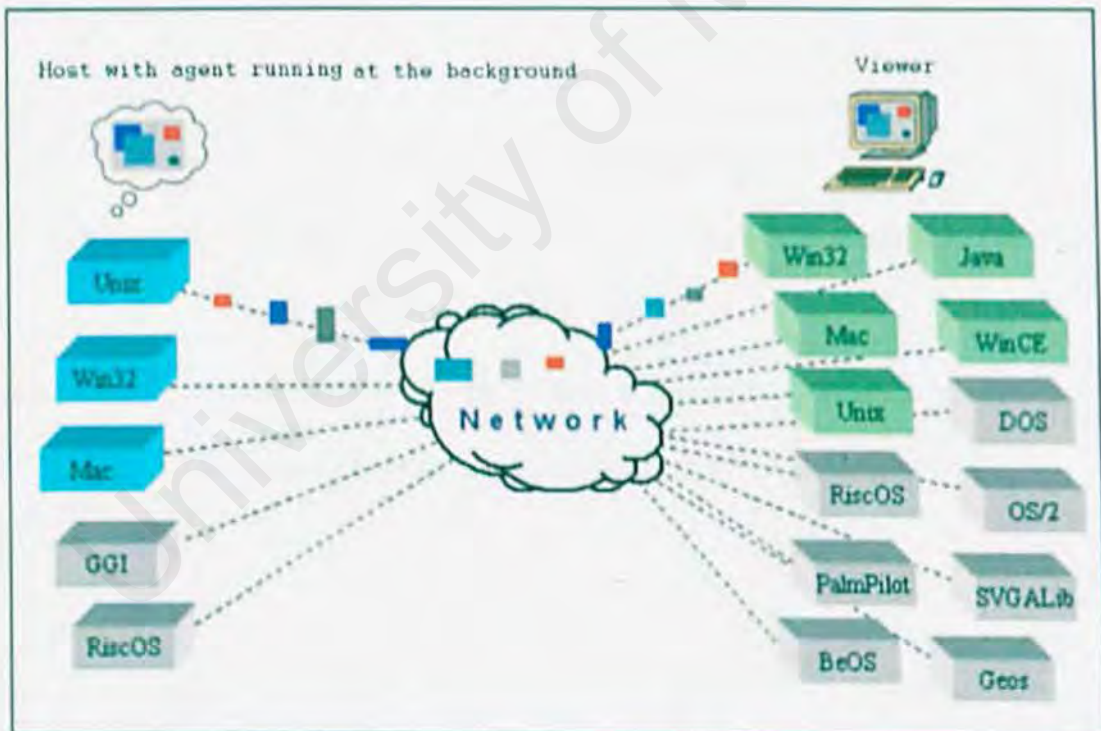


Figure 2.1: VNC Working Environment

VNC has a wide range of applications including system administration, IT support and helpdesks. It can also be used to support the mobile user, both for hot desking within the enterprise and also to provide remote access at home, or on the road.

The system allows several connections to the same desktop, providing an invaluable tool for collaborative or shared working in the workplace or classroom. Computer support within the geographically spread family is an ever popular use.

2.1.1 Agent

Agent software is a topic of growing interest to users and developers in the computer industry. Already, agents and wizards help users automate tasks such as editing and searching for information. A software agent is a program that performs tasks for its user. While this may sound just like any program, agents have somewhat special properties, which the enormous amount of media hype (and subsequent misuse of the term) has clouded in recent years. A good agent must have the characteristic as below:

Trust : the agent will do what you think it will do,

Personalizability : the agent can be either learn or be explicitly taught what to do for each individual user, and

Autonomy : the agent is allowed to take at least some actions on the user's behalf, without permission or perhaps even notification.

Property	Other Names	Meaning
Reactive	Sensing and acting	responds in a timely fashion to changes in the environment
autonomous		exercises control over its own actions
goal-oriented	pro-active purposeful	does not simply act in response to the environment
temporally continuous		is a continuously running process
communicative	socially able	communicates with other agents, perhaps including people
learning	adaptive	changes its behavior based on its previous experience
mobile		able to transport itself from one machine to another
flexible		actions are not scripted
character		believable "personality" and emotional state.

Table 2.1: Agent Classifications

There are, of course, other possible classifying schemes. For example, we might classify software agents according to the tasks they perform, for example, information gathering agents or email filtering agents. Or, we might classify them according to their control architecture. Agents may also be classified by the range and sensitivity of their senses, or by the range and effectiveness of their actions, or by how much internal state they possess.

Brustoloni's taxonomy of software agents [1991] begins with a three-way classification into regulation agents, planning agents, or adaptive agents. A regulation agent, probably named with regulation of temperature by a thermostat or similar regulation of bodily homeostasis, reacts to each sensory input as it comes in, and always knows what to do. It neither plans nor learns. Planning agents plan, either

in the usual AI sense (problem solving agent), or using the case-based paradigm (case-based agents), or using operations research based methods (OR agents), or using various randomizing algorithms (randomizing agent). Brusloni's adaptive agents not only plan, but learn. Thus there are adaptive problem solving agents, and so on, yielding a two layer taxonomy. Yet another possible classification scheme might involve the environment in which the agent finds itself, for example software agents as opposed to artificial life agents. And, there must be many, many more such possibilities

Typical applications of agents include information filtering and information discovery, although some are also used for entertainment. The Agents Group of the MIT Media Laboratory has built agents that do collaborative filtering of email or news filtering.

2.2 TCPIP, Socket, Port, Microsoft Windows Services

2.2.1 TCP/IP

Transmission Control Protocol (TCP) and Internet Protocol (IP) were developed by a Department of Defense (DOD) research Project to connect a number of different networks designed by different vendor into a network of networks (the internet). The name "TCP/IP" refers to an entire suite of data communications protocols. The suite gets its name from two of the protocols that belong to it: the Transmission Control Protocol and the Internet Protocol. Although there are many other protocols in the suite, TCP and IP are certainly two of the most important. TCP/IP Features

- Open protocol standards, freely available and developed independently from any specific computer hardware or operating system. Because it is so widely supported, TCP/IP is ideal for uniting different hardware and software, even if you don't communicate over the Internet.
- Independence from specific physical network hardware. This allows TCP/IP to integrate many different kinds of networks. TCP/IP can be run over an Ethernet, a token ring, a dial-up line, an FDDI net, and virtually any other kind of physical transmission medium.
- A common addressing scheme that allows any TCP/IP device to uniquely address any other device in the entire network, even if the network is as large as the worldwide Internet
- Standardized high-level protocols for consistent, widely available user services.

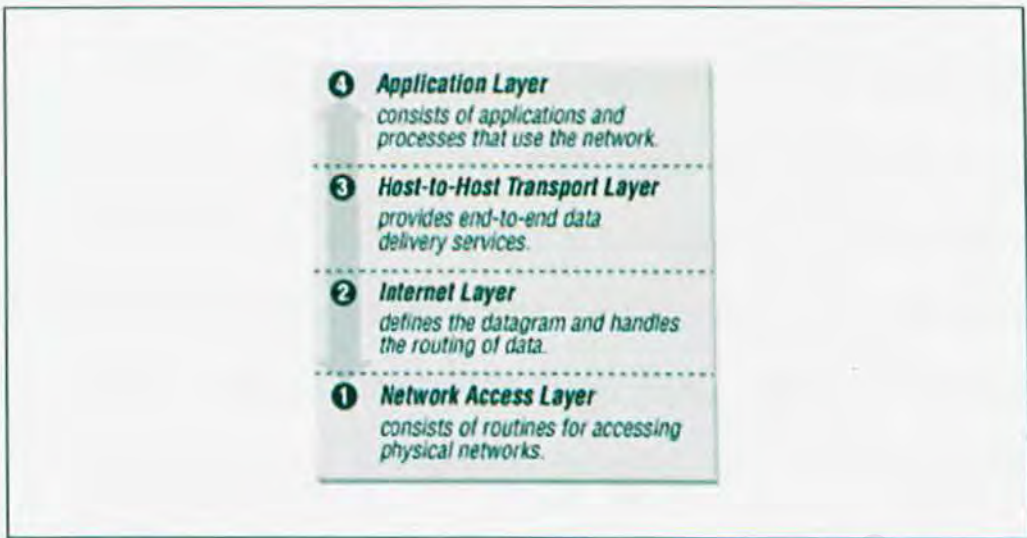


Figure 2.2: Layers in the TCP/IP protocol architecture

While there is no universal agreement about how to describe TCP/IP with a layered model, it is generally viewed as being composed of fewer layers than the seven used in the OSI model. Most descriptions of TCP/IP define three to five functional levels in the protocol architecture. The four-level model illustrated in Figure 2.2.

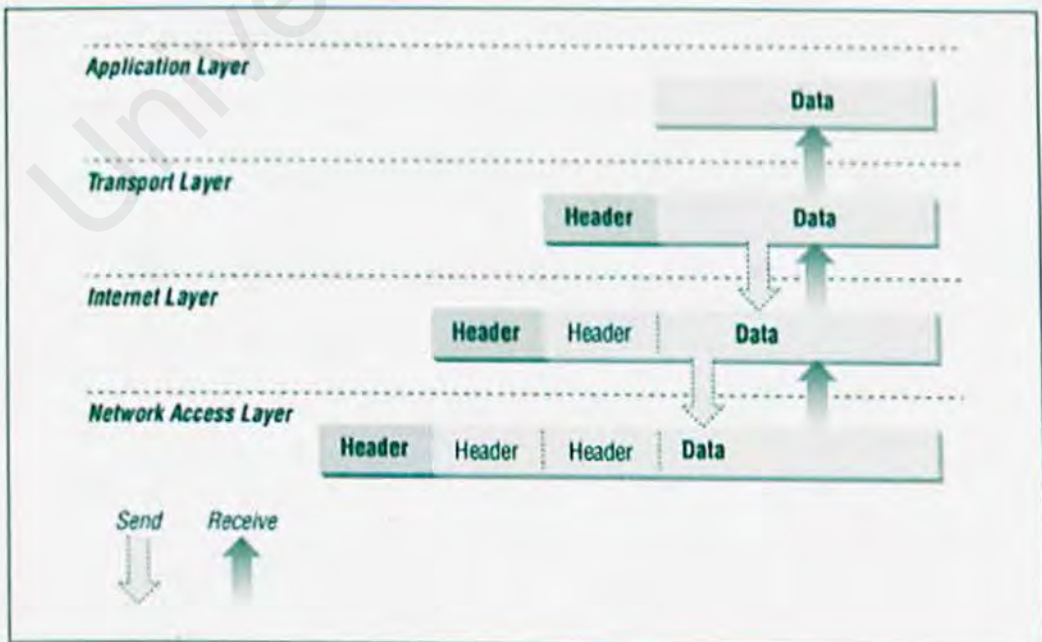


Figure 2.3: Data encapsulation

As in the OSI model, data is passed down the stack when it is being sent to the network, and up the stack when it is being received from the network. The four-layered structure of TCP/IP is seen in the way data is handled as it passes down the protocol stack from the Application Layer to the underlying physical network. Each layer in the stack adds control information to ensure proper delivery. This control information is called a *header* because it is placed in front of the data to be transmitted. Each layer treats all of the information it receives from the layer above as data and places its own header in front of that information. The addition of delivery information at every layer is called *encapsulation*. When data is received, the opposite happens. Each layer strips off its header before passing the data on to the layer above. As information flows back up the stack, information received from a lower layer is interpreted as both a header and data.

Each layer has its own independent data structures. Conceptually, a layer is unaware of the data structures used by the layers above and below it. In reality, the data structures of a layer are designed to be compatible with the structures used by the surrounding layers for the sake of more efficient data transmission. Still, each layer has its own data structure and its own terminology to describe that structure.

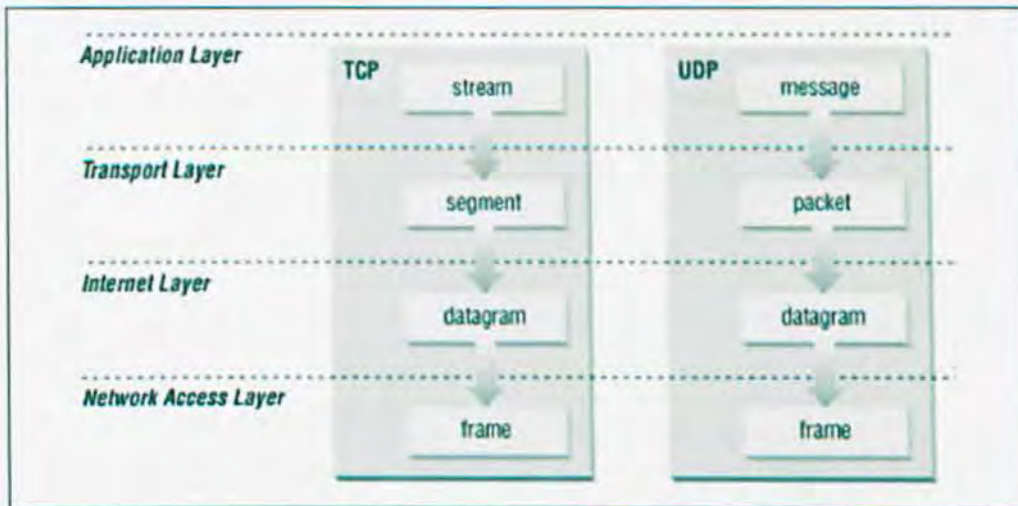


Figure 2.4: Data structures

Figure 2.4 shows the terms used by different layers of TCP/IP to refer to the data being transmitted. Applications using TCP refer to data as a *stream*, while applications using the User Datagram Protocol (UDP) refer to data as a *message*. TCP calls data a *segment*, and UDP calls its data structure a *packet*. The Internet layer views all data as blocks called *datagrams*. TCP/IP uses many different types of underlying networks, each of which may have a different terminology for the data it transmits. Most networks refer to transmitted data as *packets* or *frames*. In Figure 2.4 we show a network that transmits pieces of data it calls *frames*.

2.2.2 Socket

Definition: A socket is one endpoint of a two-way communication link between two programs running on the network. A socket is bound to a port number so that the TCP layer can identify the application that data is destined to be sent.

Normally, a server runs on a specific computer and has a socket that is bound to a specific port number. The server just waits, listening to the socket for a client to make a connection request.

On the client-side: The client knows the hostname of the machine on which the server is running and the port number to which the server is connected. To make a connection request, the client tries to rendezvous with the server on the server's machine and port.



Figure 2.5: Connection Request

If everything goes well, the server accepts the connection. Upon acceptance, the server gets a new socket bound to a different port. It needs a new socket (and consequently a different port number) so that it can continue to listen to the original socket for connection requests while tending to the needs of the connected client.

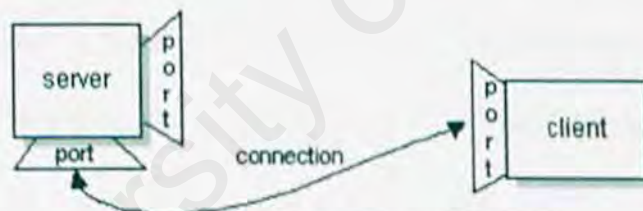


Figure 2.6: Connection Establish

On the client side, if the connection is accepted, a socket is successfully created and the client can use the socket to communicate with the server. Note that the socket on the client side is not bound to the port number used to rendezvous with the server. Rather, the client is assigned a port number local to the machine on which the client is running.

The client and server can now communicate by writing to or reading from their sockets.

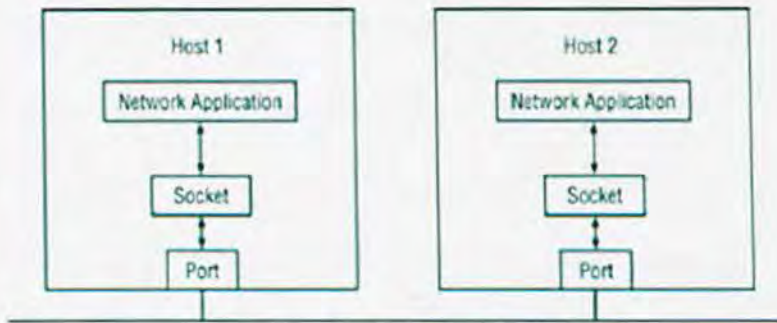


Figure 2.7: The socket interface

2.2.3 Port

Definition: A port number represents an endpoint or "channel" for network communications. Port numbers allow different applications on the same computer to utilize network resources without interfering with each other.

Port numbers most commonly appear in network programming, particularly socket programming. Sometimes, though, port numbers are made visible to the casual user. For example, some Web sites a person visits on the Internet use a URL like the following:

<http://www.mairie-metz.fr.8080/>

In this example, the number 8080 refers to the port number used by the Web browser to connect to the Web server. Normally, a Web site uses port number 80 and this number need not be included with the URL (although it can be).

In IP networking, port numbers can theoretically range from 0 to 65535. Most popular network applications, though, use port numbers at the low end of the range (such as 80 for HTTP). The port number is included as a field within the header of each IP packet.

Note: The term port also refers to several other aspects of network technology. A port can refer to a physical connection point for peripheral devices such as serial, parallel, and USB ports. The term port also refers to certain Ethernet connection points, such as those on a hub, switch, or router. Also Known As: port number, protocol number

2.2.4 Microsoft Windows Services

Microsoft Windows services, formerly known as NT services, enable developer to create long-running executable applications that run in their own Windows sessions. These services can be automatically started when the computer boots, can be paused and restarted, and do not show any user interface.

This makes services ideal for use on a server or whenever you need long-running functionality that does not interfere with other users who are working on the same computer. You can also run services in the security context of a specific user account that is different from the logged-on user or the default computer account.

You can easily create services by creating an application that is installed as a service. For example, suppose you want to monitor performance counter data and react to threshold values. You could write a Windows Service application that listens to the performance counter data, deploy the application, and begin collecting and analyzing data.

You create your service as a Microsoft Visual Studio .NET project, defining code within it that controls what commands can be sent to the service and what actions should be taken when those commands are received. Commands that can be sent to a

service include starting, pausing, resuming, and stopping the service, and executing custom commands.

After you create and build the application, you can install it by running the command line utility `InstallUtil.exe` and passing the path to the service's executable file, or by using Visual Studio's deployment features. You can then use the Services Control Manager to start, stop, pause, resume, and configure your service. You can also accomplish many of these same tasks in the Services node in Server Explorer or by using the `ServiceController` class.

Service applications function differently from many other project types in several ways. The compiled executable file that a service application project creates must be installed on the server before the project can function in a meaningful way. You cannot debug or run a service application by pressing F5 or F11; you cannot immediately run a service or step into its code. Instead, you must install and start your service, and then attach a debugger to the service's process.

Unlike some types of projects, you must create installation components for service applications. The installation components install and register the service on the server and create an entry for your service with the Windows Services Control Manager.

The `Main` method for your service application must issue the `Run` command for the services your project contains. The `Run` method loads the services into the Services Control Manager on the appropriate server. If you use the Windows Services project

template, this method is written for you automatically. Note that loading a service is not the same thing as starting the service.

Windows Service applications run in a different window station than the interactive station of the logged-on user. A window station is a secure object that contains a clipboard, a set of global atoms, and a group of desktop objects. Because the station of the Windows Service is not an interactive station, dialog boxes raised from within a Windows Service application will not be seen and may cause your program to stop responding. Similarly, error messages should be logged in the Windows event log rather than raised in the user interface.

The Windows Service classes supported by the .NET Framework do not support interaction with interactive stations, that is, the logged-on user. The .NET Framework also does not include classes that represent stations and desktops. If your Windows Service must interact with other stations, you will need to access the unmanaged Windows API.

The interaction of the Windows Service with the user or other stations must be carefully designed to include scenarios such as there being no logged on user, or the user having an unexpected set of desktop objects. In some cases, it may be more appropriate to write a Windows application that runs under the control of the user.

Windows Service applications run in their own security context and are started before the user logs into the Windows computer on which they are installed. You should plan carefully what user account to run the service within; a service running under the system account has more permissions and privileges than a user account.

University of Malaya

2.3 Case Study

2.3.1 RealVNC

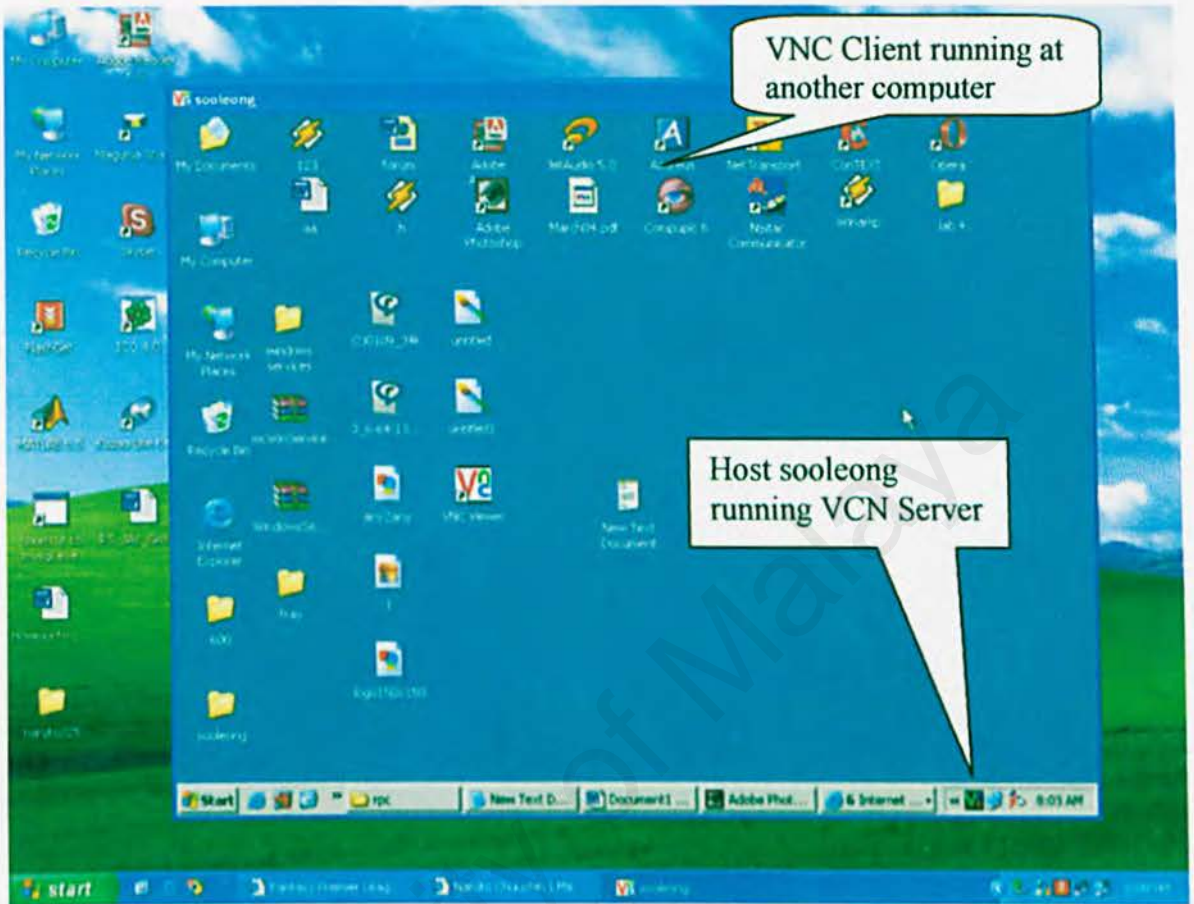


Figure 2.8: RealVNC ScreenShot



RealVNC is a UK company founded in 2002 by a team from the world-leading AT&T Laboratories in Cambridge. The company was established to commercially develop, enhance and promote VNC, the innovative remote access software with a widespread international user base.

The official Web site for RealVNC is

<http://www.realvnc.com/>

Figure 2.9: Logo of RealVNC

2.3.2 Access Remote PC

Access Remote PC is a computer program that lets you access your PC from another PC via Internet, LAN, or phone connection and work on your computer remotely as if you were sitting in front of it.

For Individuals

Control computers in your home network, access your home computer from office or work on your office computer from home, access your office or home computer while traveling. Invite someone to your PC to fix it remotely.

And Businesses

Your main office can access branch office desktops, or your branch office can access main office server. Your employees can access email, files, programs and network resources from home or the road. System administrators can remotely access and manage multiple PCs from anywhere. Your sales force can access applications and documents and continue to work while on the road.

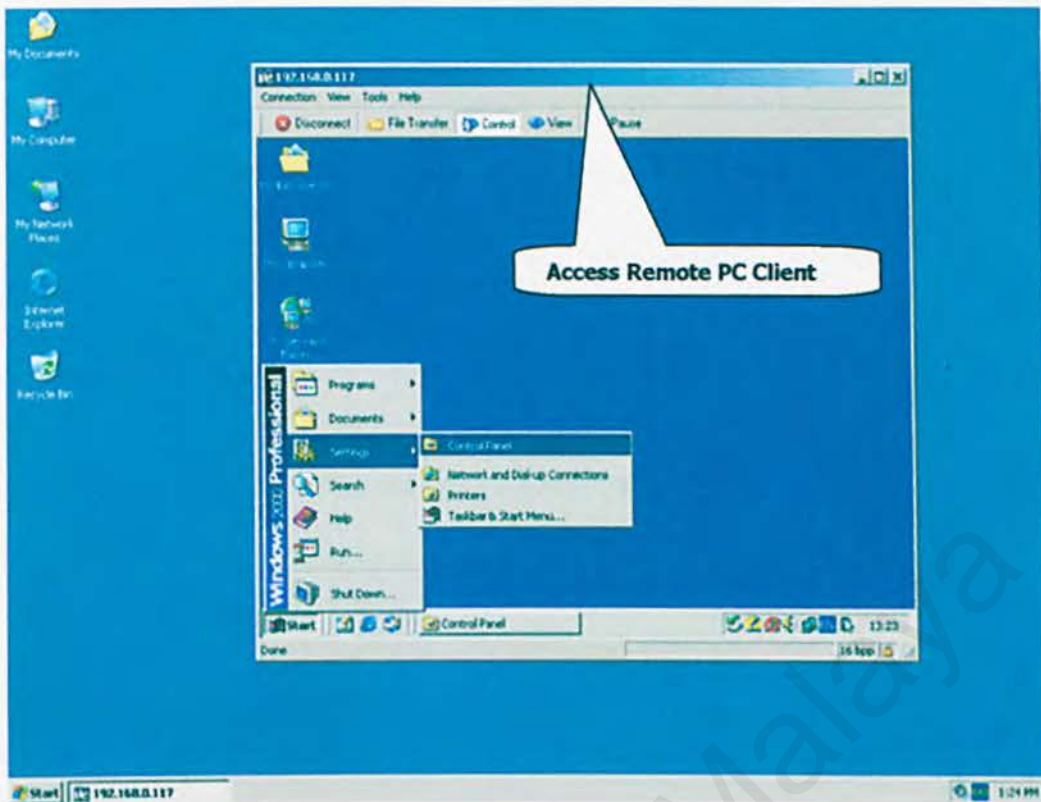


Figure 2.10: Access Remote PC ScreenShot

Some of the features / functionality::

- Lightweight (just 1 MB in size)
- Full screen option
- NT service mode
- Command-line interface
- Password protection
- Server icon in the system's tray
- Hide icon option
- List of connected users
- Access control
- Supports background operation

Cost \$39.95 (life time fee)

Buy licenses to use Access Remote PC on your home or company network or over the Internet via IP address. These licenses also cover Helpdesk peer-to-peer usage over IP (not firewall friendly).

The official Web site for Access Remote PC is <http://www.access-remote-pc.com/>

2.3.3 Remote Administrator 2.2

Radmin is an award-winning multifunction remote control program. Radmin enables you to remotely monitor or work on one or more network computers right from your own workstation. It is a complete remote control solution with such features as File Transfer, NT security, Telnet and Multilanguage Support built-in.

Remote Administrator (Radmin) gives you instant access to various remote resources through an Internet connection, over direct telephone lines and across multiple Windows platforms. Now you can monitor and manage PCs and servers in different locations anywhere in the world without leaving your desk.



Figure 2.11: Remote Administrator 2.2 ScreenShot

Functionality of Remote Administrator 2.2

- Operating systems support: Windows 9x/ME/NT4.0/2000/XP/2003*
- Radmin Server can work as a service under all supported operating systems, which allows you to logon and logoff a user remotely.
- Radmin supports Windows NT/2000/XP/2003* user level security. You can give the right of remote control to a specific user or user group.
- If Windows NT security support is switched off, access to a remote computer is controlled by password. Remote Administrator uses a challenge-response password authentication method similar to that used in Windows NT, but with more powerful security keys.
- Supports multiple connections on both server and client sides.
- Supports the controlling of multiple remote computers and the viewing of multiple sessions on one screen.
- Transfers files to or from a remote computer. With Radmin you can easily drag and drop files via Explorer-like interface.
- Lets you remotely shutdown a computer without having to connect in remote screen mode.
- View modes include Full-Screen, Scaled and Windowed. Full-Screen mode lets you see the remote screen on the entire screen of your display.
Scaled mode lets you see the scaled remote screen in a window with a user defined size.
- Radmin uses a video hook kernel mode driver under Windows NT 4.0 to improve performance. This lets you work on a remote computer with an incredible 'real-time' update speed (hundreds of screen updates per second).

The Win2000/XP/2003 compatible version of the driver will be included in the coming version of Radmin.

- Radmin server provides Telnet access to remote computers when it's Server runs on Windows NT/2000/XP/2003.
- Has multilingual support using one file per language so it is not necessary to download and re-install Radmin for each language.
- Radmin supports High Resolution modes of up to 2048 X 2048 X 32bit color.

Cost \$ 35.00 (Single License)

The official Web site for Remote Administrator 2.2 is

<http://www.famatech.com/>

2.3.4 Windows XP Remote Desktop Connection software



Figure 2.12: Windows XP Remote Desktop Connection software ScreenShot

Remote Desktop in Windows XP Professional provides remote access to the desktop of your computer running Windows XP Professional, from a computer at another location. Using Remote Desktop you can, for example, connect to your office computer from home and access all your applications, files, and network resources as though you were in front of your computer at the office. When run, this software allows older Windows platforms to remotely connect to a computer running Windows XP Professional with Remote Desktop enabled.

This package provides flexible deployment options of the full Terminal Services Client, including auto-repair through Windows Installer technology and application publishing via IntelliMirror™ management technologies or Microsoft Systems Management Server (SMS).

Note: The Remote Desktop Connection software is pre-installed with Windows XP. To run it, click Start, click All Programs, click Accessories, click Communications, and then click Remote Desktop Connection. This software package can also be found on the Windows XP Professional and Windows XP Home Edition product CDs and can be installed on any supported Windows platform. To install from the CD, insert the disc into the target machine's CD-ROM drive, select Perform Additional Tasks, and then click Install Remote Desktop Connection

System Requirements

- Supported Operating Systems: Windows 95, Windows 98, Windows ME, Windows NT, Windows Server 2003, Windows X

2.4 Operating System

An operating system is made up of software instructions that lie between the computer hardware (disks, memory, ports, and so on) and the application programs (word processors, Web browsers, spreadsheets, and so on). At the center is the kernel, which provides the most basic computing functions (managing system memory, sharing the processor, opening and closing devices, and so on). Besides the kernel, an operating system provides other basic services needed to operate the computer, including:

File systems

- The file system provides the structure in which information is stored on the computer. Information is stored in files, primarily on hard disks inside the computer. Files are organized within a hierarchy of directories.

Device drivers

- These provide the interfaces to each of the hardware devices connected to your computer. A device driver enables a program to write to a device without needing to know details about how each piece of hardware is implemented. The program opens a device, sends and receives data, and closes a device.

User interfaces

- An operating system needs to provide a way for users to run programs and access the file system. Linux has both graphical and text-based user interfaces.

System services

- An operating system provides system services, many of which can be started automatically when the computer boots.

Without an operating system, an application program would have to know the details of each piece of hardware, instead of just being able to say, “open that device and write a file there.”

2.4.1 Microsoft Windows 2000

Windows 2000 is a commercial version of Microsoft’s evolving Windows operating system. Previously called Windows NT 5.0, Microsoft emphasizes that Windows 2000 is evolutionary and “Built on NT Technology”. Windows 2000 designed to appeal to small business and professional users as well as to the more technical and larger business market for which the NT was designed.

The windows 2000 product line consists of four products:

- Windows 2000 Professional: aimed at individuals and businesses of all sizes. It includes security and mobile use enhancements. It is the most economical choice.
- Windows 2000 Server: aimed at small-to-medium size businesses. It can function as a Web server and /or workgroup (or branch office) server. It can be part of a two-way symmetric multiprocessing system. NT 4.0 servers can be upgraded to this server.
- Windows Advanced Server: aimed at being a network operating system server and / or an application server, including those involving large database.

This server facilitates clustering and load-balancing. NT 4.0 servers with up to eight-way SMP can upgrade to this product.

- Windows 2000 Datacenter Server: designed for large data warehouses, online transaction processing (OLTP), econometric analysis, and other applications requiring high-speed computation and large databases. The Datacenter Server supports up to 16-way SMP and up to 64 GB of physical memory.

Windows 2000 is reported to be more stable (less apt to crash) than Windows 98/NT systems. A significant new feature is Microsoft's Active Directory, which, among other capabilities, enables a company to set up virtual private networks, to encrypt data locally or on the network, and to give users access to shared files in a consistent way from any network computer.

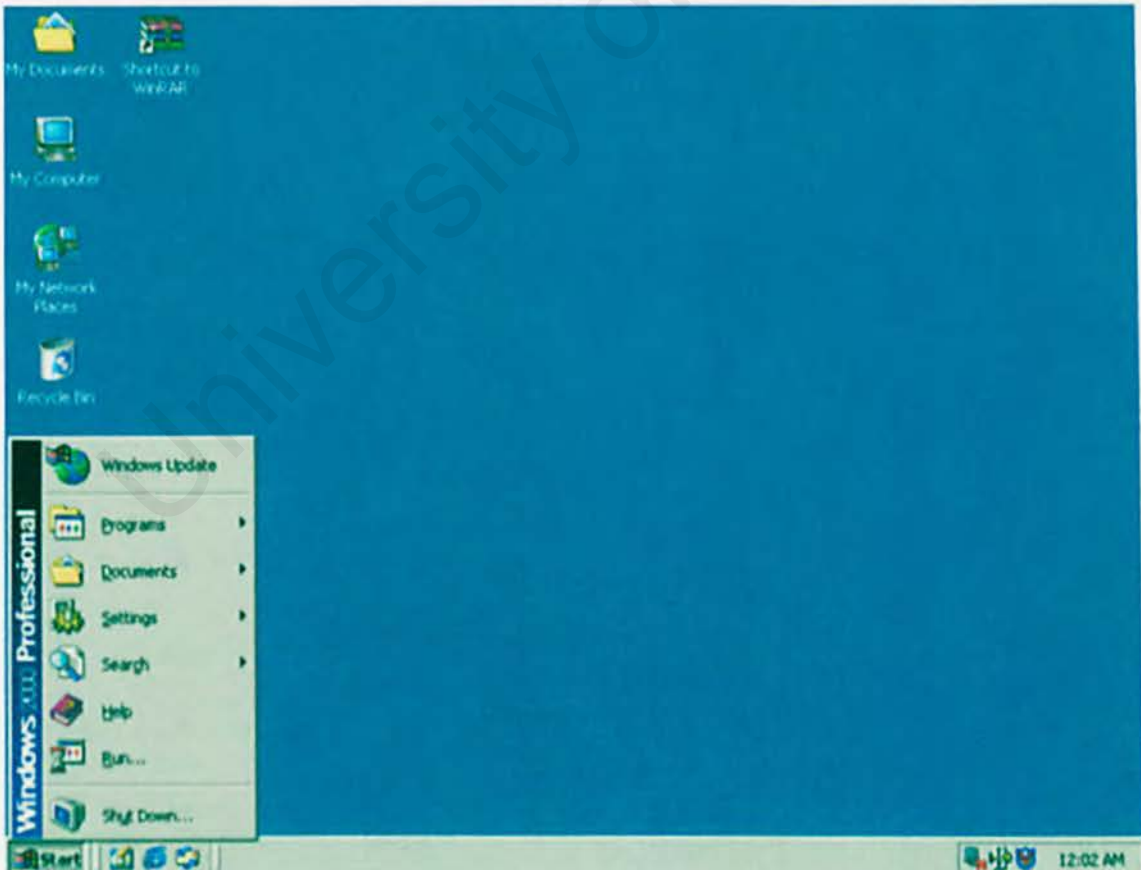


Figure 2.13: Microsoft Windows 2000 ScreenShot

2.4.2 Microsoft Windows XP

Windows XP is the latest version of Windows desktop operating system for PC. Microsoft and trade publication writers view Windows XP as the most important version of Windows since Windows 95. Windows XP is built on the desktop that will also make it easier for users to scan or import images and to acquire music files on the Web and transfer them to portable devices.

The new Windows allows different family members to use their own desktop and personal sets of files. In addition to the “My Computer” and “My Documents” views provided in Windows 2000, Windows XP users see “My Music” and “My Pictures”. The Start Menu has been redesigned to make the most-used programs easiest to find. Windows XP comes in a Professional version and a Home Edition version.



Figure 2.14: Microsoft Windows XP ScreenShot

2.4.3 Linux

Linux is a free operating system that was created by Linus Torvalds when he was a student at the University of Helsinki in 1991. Torvalds started Linux by writing a *kernel* — the heart of the operating system — partly from scratch and partly by using publicly available software. Torvalds then released the system to his friends and to a community of “hackers” on the Internet and asked them to work with it and enhance it. It took off.

Today, there are hundreds of software developers around the world contributing software to the Linux effort. Because the source code for the software is freely available, anyone can work on it, change it, or enhance it. On top of the Linux kernel effort, the creators of Linux also drew on a great deal of system software and applications that are now bundled with Linux from the GNU software effort (GNU stands for “GNU is Not UNIX”), which is directed by the Free Software Foundation (FSF).

There is a vast amount of software that can be used with Linux, all of which includes features that can compete with or surpass those of any other operating system in the world.

Unlike Windows and other proprietary systems, Linux is publicly open and extendible by contributors. Because it conforms to the Portable Operating System Interface standard user and programming interfaces, developers can write programs that can be ported to other operating systems.

Linux comes in versions for all the major microprocessor platforms including the Intel, PowerPC, Sparc, and Alpha platforms. It's also available on IBM's S/390. Linux is sometimes suggested as a possible publicly developed alternative to desktop predominance of Microsoft Windows. Although Linux is popular among users already familiar with UNIX, it remains far behind Windows in numbers of users.



Figure 2.15: Linux ScreenShot (running KDE Desktop)

2.4.4 Unix

UNIX (often spelled “Unix” in news media) is an operating system that originated at Bell Labs in 1969 as an interactive time-sharing system. Ken Thompson and Dennis Ritchie are considered the inventors of UNIX. The name (pronounced YEW-ninks) was a pun based on earlier system, Multics.

Unix is a layered operating system. The innermost layer is the hardware that provides the services for the OS. The operating system, referred to in Unix as the kernel, interacts directly with the hardware and provides the services to the user programs. These user programs don't need to know anything about the hardware. They just need to know how to interact with the kernel and it's up to the kernel to provide the desired service. One of the big appeals of Unix to programmers has been that most well written user programs are independent of the underlying hardware, making them readily portable to new systems. User programs interact with the kernel through a set of standard system calls. These system calls request services to be provided by the kernel. Such services would include accessing a file: open close, read, write, link, or execute a file; starting or updating accounting records; changing ownership of a file or directory; changing to a new directory; creating, suspending, or killing a process; enabling access to hardware devices; and setting limits on system resources.

Unix is a multi-user, multi-tasking operating system. You can have many users logged into a system simultaneously, each running many programs. It's the kernel's job to keep each process and user separate and to regulate access to system hardware, including cpu, memory, disk and other I/O devices.

2.5 Programming Language and Technologies

2.5.1 Java

In discussing Java, it is important to distinguish between the Java programming language, the Java Virtual Machine, and the Java platform. The Java programming language is the language in which Java applications (including applets, servlets, and JavaBeans components) are written. When a Java program is compiled, it is converted to byte codes that are the portable machine language of a CPU architecture known as the Java Virtual Machine (also called the Java VM or JVM). The JVM can be implemented directly in hardware, but it is usually implemented in the form of a software program that interprets and executes byte codes.

The Java platform is distinct from both the Java language and Java VM. The Java platform is the predefined set of Java classes that exist on every Java installation; these classes are available for use by all Java programs. The Java platform is also sometimes referred to as the Java runtime environment or the core Java APIs (application programming interfaces). The Java platform can be extended with optional standard extensions. These extension APIs exist in some Java installations, but are not guaranteed to exist in all installations.

The Java programming language is a state-of-the-art, object-oriented language that has a syntax similar to that of C. The language designers strove to make the Java language powerful, but, at the same time, they tried to avoid the overly complex features that have bogged down other object-oriented languages, such as C++. By keeping the language simple, the designers also made it easier for programmers to write robust, bug-free code. As a result of its elegant design and next-generation

features, the Java language has proved wildly popular with programmers, who typically find it a pleasure to work with Java after struggling with more difficult, less powerful languages.

The Java Virtual Machine, or Java interpreter, is the crucial piece of every Java installation. By design, Java programs are portable, but they are only portable to platforms to which a Java interpreter has been ported. Sun ships VM implementations for its own Solaris operating system and for Microsoft Windows (95/98/NT) platforms. Many other vendors, including Apple and various Unix vendors, provide Java interpreters for their platforms. There is a freely available port of Sun's VM for Linux platforms, and there are also other third-party VM implementations available. The Java VM is not only for desktop systems, however. It has been ported to set-top boxes, and versions are even available for hand-held devices that run Windows CE and PalmOS.

Although interpreters are not typically considered high-performance systems, Java VM performance is remarkably good and has been improving steadily. Of particular note is a VM technology called just-in-time (JIT) compilation, whereby Java byte codes are converted on-the-fly into native-platform machine language, boosting execution speed for code that is run repeatedly. Sun's new Hotspot technology is a particularly good implementation of JIT compilation.

The Java platform is just as important as the Java programming language and the Java Virtual Machine. All programs written in the Java language rely on the set of predefined classes that comprise the Java platform. Java classes are organized into

related groups known as packages. The Java platform defines packages for functionality such as input/output, networking, graphics, user-interface creation, security, and much more.

A class is a module of Java code that defines a data structure and a set of methods (also called procedures, functions, or subroutines) that operate on that data.

It is important to understand what is meant by the term platform. To a computer programmer, a platform is defined by the APIs he or she can rely on when writing programs. These APIs are usually defined by the operating system of the target computer. Thus, a programmer writing a program to run under Microsoft Windows must use a different set of APIs than a programmer writing the same program for the Macintosh or for a Unix-based system. In this respect, Windows, Macintosh, and Unix are three distinct platforms.

Java is not an operating system. Nevertheless, the Java platform--particularly the Java 2 Platform--provides APIs with a comparable breadth and depth to those defined by an operating system. With the Java 2 Platform, you can write applications in Java without sacrificing the advanced features available to programmers writing native applications targeted at a particular underlying operating system. An application written on the Java platform runs on any operating system that supports the Java platform. This means you do not have to create distinct Windows, Macintosh, and Unix versions of your programs, for example. A single Java program runs on all these operating systems, which explains why "Write once, run anywhere" is Sun's motto for Java.

There is a Java-based operating system, however; it is known as JavaOS.

The Java platform is not an operating system, but for programmers, it is an alternative development target and a very popular one at that. The Java platform reduces programmers' reliance on the underlying operating system, and, by allowing programs to run on top of any operating system, it increases end users' freedom to choose an operating system.

In addition to evolving the Java platform over time, Sun is also trying to produce different versions of the platform for different uses. The Standard Edition is the only version currently available, but Sun is also working on the Java 2 Platform, Enterprise Edition (J2EE), for enterprise developers and the Java 2 Platform, Micro Edition, for consumer electronic systems, like handheld PDAs and cellular telephones.

2.5.2 .NET Framework

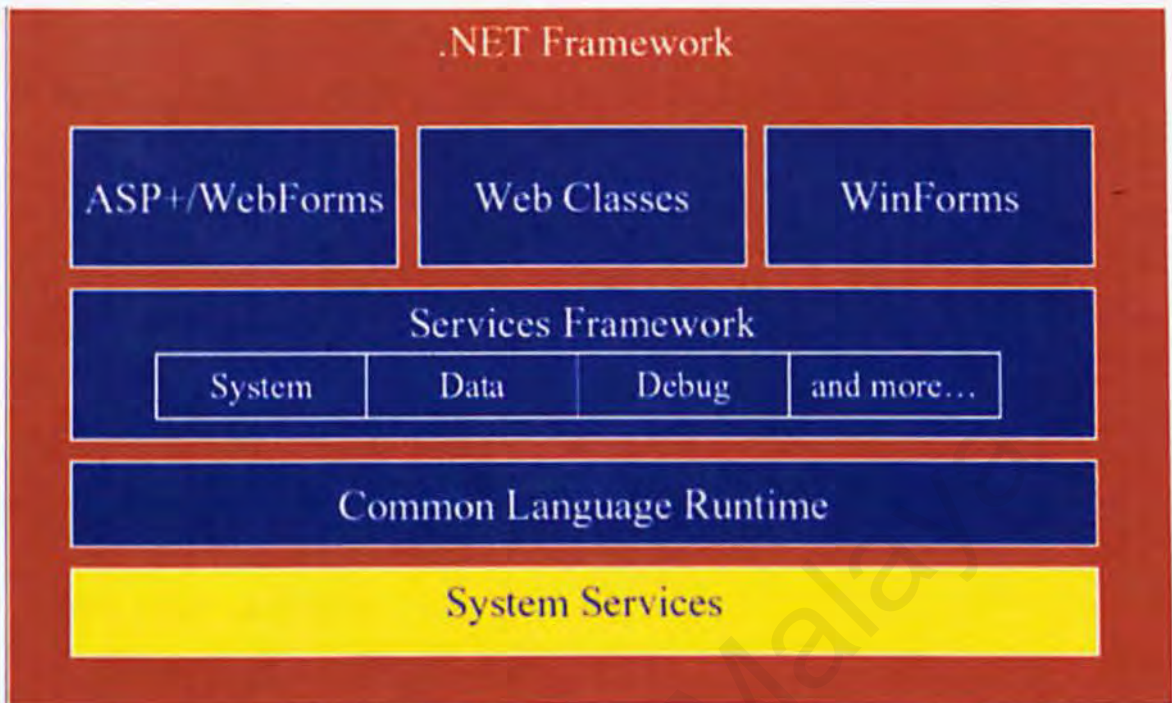


Figure 2.16: .NET Framework

2.5.3 Introduction of Microsoft .NET

.NET is a whole new programming environment, designed for the internet, leverages existing skill sets, takes Internet programming to the next level

.NET components

- .NET framework
- .NET servers
- .NET Architecture
- Common Language Runtime
- .NET System Class Library
- Web Forms and ASP+
- Web Services
- Win Forms
- Deployment and Configuration
- ADO+
- Languages

.NET Goals

- Consistent and simplified programming model
- Cross-platform interoperability
- Multi-language support
- Code reuse
- Automatic resource management
- Type safety
- Debugging
- Error handling
- Deployment (Elimination of DLL hell)
- Security

Common Language Runtime

- Consistent support for all languages
 - o Intermediate Language (IL)
 - o Interoperability between languages
- Just-In-Time (JIT) compilers
- Type Safety
 - o Code that can't violate the memory constraints of the runtime (no pointers, etc.)
- Managed vs. Unmanaged code
- Metadata

As has been mentioned already, the CLR is, like the Java virtual machine, a runtime environment that takes charge of resource management tasks (memory allocation and garbage collection) and ensures the necessary abstraction between the application and the underlying operating system.

In order to provide a stable platform, with the aim of reaching the level of reliability required by the transactional applications of e-business, the CLR also fulfills related tasks such as monitoring of program execution. In DotNet-speak, this involves "managed" code for the programs running under CLR monitoring, and "unmanaged" code for applications or components which run in native mode, outside the CLR.

The CLR watches for the traditional programming errors that for years have been at the root of the majority of software faults: access to elements of an array outside limits, access to non-allocated memory zones, memory overwritten due to exceeded sizes.

Compilation

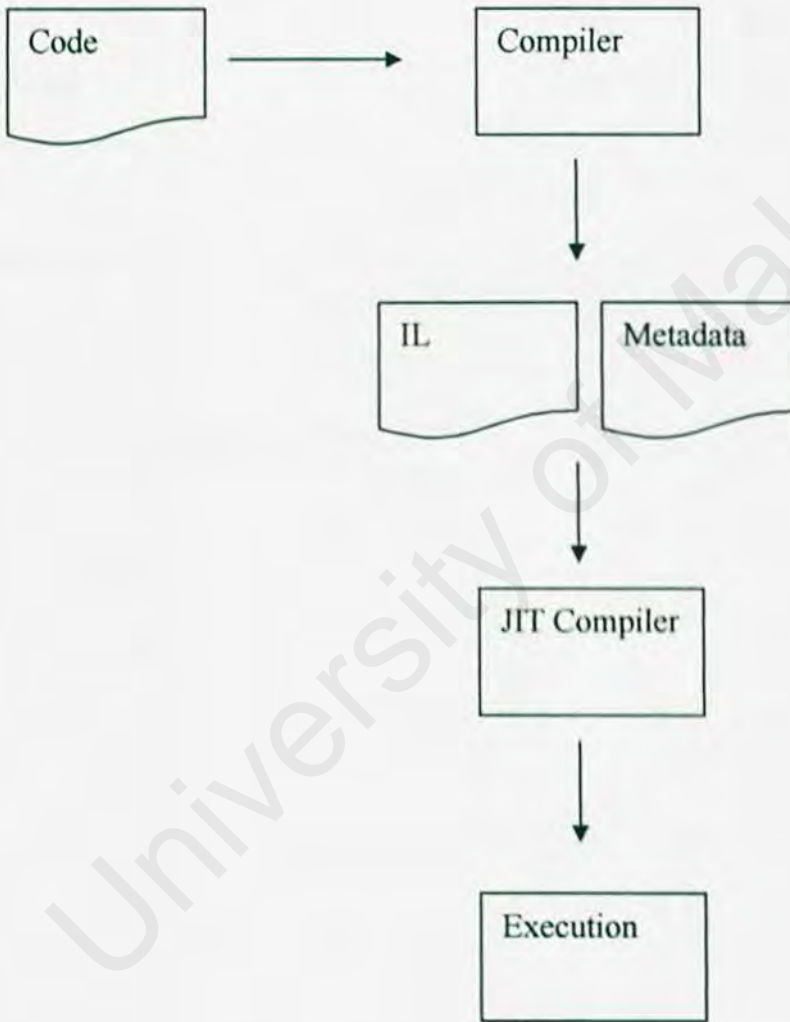


Figure 2.17: .NET Compilation

.NET System Class Library

Code DOM	NewXML
Collections	Reflection
Components	Resources
Configuration	Runtime
Core	Security
Data	Service Process
Diagnostics	System
Directory services	Text
Drawing	Threading
Globalization	Timers
IO	Web
Messaging	WinForms
Net	XML

.NET Programming Languages

- VB.NET
- C#.NET
- C++.NET
- J#.NET

2.5.3.1 VB.NET



Visual Basic® .NET 2003 lets you quickly write dependable, robust software. It delivers the tools you need to build applications that will power today's organizations and drive the next generation of Windows, Web, and mobile software.

Figure 2.18: VB.NET

Features

- Solve today's development problems more rapidly
- Quickly create powerful Windows and Web applications, easily deploy and maintain applications, and access data from a wider range of scenarios.
- Build powerful Windows applications in less time
- Build more responsive, powerful, and easy-to-deploy applications with control anchoring, an in-place menu editor, and more.
- Rapid Application Development for the Web
- Use the Web Forms designer to quickly build true thin-client Web applications that intelligently render on any browser and on any platform.
- Easily deploy and maintain applications
- Difficulties with DLLs are a thing of the past with side-by-side versioning, XCOPY deployment, and auto-download of Windows applications.
- Fast development for mobile Web devices
- Visual Basic .NET 2003 features support for more than 200 mobile Web devices including mobile phones, pagers, and personal digital assistants (PDAs).

2.5.3.2 C#.NET



Visual C#® .NET is an innovative language for building .NET-connected software for a variety of platforms and devices. With the language's familiar syntax and integrated environment, Visual C# .NET eases the development of .NET-connected software.

Figure 2.19: C#.NET

Features

- Join the forefront of software design
- C# includes a number of innovative component-oriented language features, letting you more readily create robust, durable software.
- Increase your overall productivity
- Improved IntelliSense® statement completion helps you write code more efficiently with on-the-fly syntax and error checking.
- Reduce the learning curve
- Leverage a strong C++ heritage modeled on C++ syntax. C# code is immediately comfortable to C++ and Java programmers regardless of skill level.
- Take advantage of a robust debugger
- Debugger windows give you precise control over application testing, and immediate Window IntelliSense makes your debugging more productive.
- Construct compelling .NET-connected solutions
- Build a variety of applications, from device solutions to datacenter servers. Use code obfuscation technology to protect your intellectual property.

2.5.3.3 C++.NET



Visual C++® .NET is a powerful toolset for creating best-in-class Windows®-based and .NET-connected applications and components using the C++ development language. A robust environment, it enables efficient editing and debugging of source code.

Figure 2.20: C++.NET

Features

- Create highly tuned .NET-connected applications
Use Managed Extensions to C++ and compiler support for optimized Microsoft Intermediate Language (MSIL) code generation.
- Move existing code to .NET at a pace you define
Existing C++ code compiles for .NET without the need to rewrite it. Visual C++ supports intermixed managed and unmanaged code and data.
- Build modern C++ code and library sources
Compile modern C++ code that uses advanced template features, including Partial Template Specialization and Partial Ordering of Function templates.
- Utilize libraries to incorporate advanced features
Visual C++ .NET 2003 includes a variety of libraries encapsulating an abundance of data types and programming patterns.
- Join a mature, intelligent developer community
Exchange code and knowledge with the vast population of hardcore C++ developers, and leverage your C++ skills across multiple computing environments.

2.5.3.4 J#.NET



Visual J#® .NET is a tool for Java developers who want to build applications and services on the .NET Framework, joining more than 20 previously announced languages with its ability to target the .NET Framework and first-class XML Web services.

Figure 2.21: J#.NET

Features

- Easily build .NET-connected software

Visual J# .NET provides the familiarity of the Java-language syntax, so you can reuse your existing skills to build .NET-connected applications.

- Build a broad range of compelling solutions

Create rich Windows, Web, and even mobile Web applications that dynamically render on more than 200 types of mobile device browsers.

- Take advantage of a first-class code editor

Efficiently organize your code with advanced editing capabilities such as IntelliSense®, code outlining, color syntax highlighting, and more.

- Receive the help you need with Dynamic Help

Dynamic Help gives you context-sensitive help with samples and documentation based on the task at hand.

- Get on the fast track to .NET development

Gain access to settings, updates, and the MSDN Online Community from the Start Page.

2.6 Tools Development

2.6.1 Microsoft® Visual Studio® .NET 2003 Professional Edition



Visual Studio® .NET 2003 Professional lets you rapidly build a broad range of applications for Windows, the Web, and mobile devices, addressing today's most challenging development needs.

Figure 2.22: Microsoft Visual Studio.NET 2003 Professional Edition

Features

- Quickly build software for Windows and the Web
Intuitive visual designers for building professional software let you reuse your existing skills across multiple application types.
- Build high-performance database applications
ADO.NET class libraries provide the infrastructure for high-performance data access to SQL Server, Oracle, and other data sources.
- Simplify application deployment and maintenance
No Touch Deployment enables applications to be deployed as easily as Web pages, and side-by-side component versioning eliminates DLL overwrites.
- Connect to a variety of systems and applications
Industry-leading support for Web service creation and consumption enables you to quickly build and employ standards-based business-logic components.
- Leverage existing investments in code and skills
Get multiple options for reusing existing code, including the Visual Basic® Upgrade Wizard, COM Interoperability, and more.
- Build device-based applications in record time
Support for smart devices enables design, development, debugging, and

automatic deployment of applications to more than 200 wireless mobile devices.

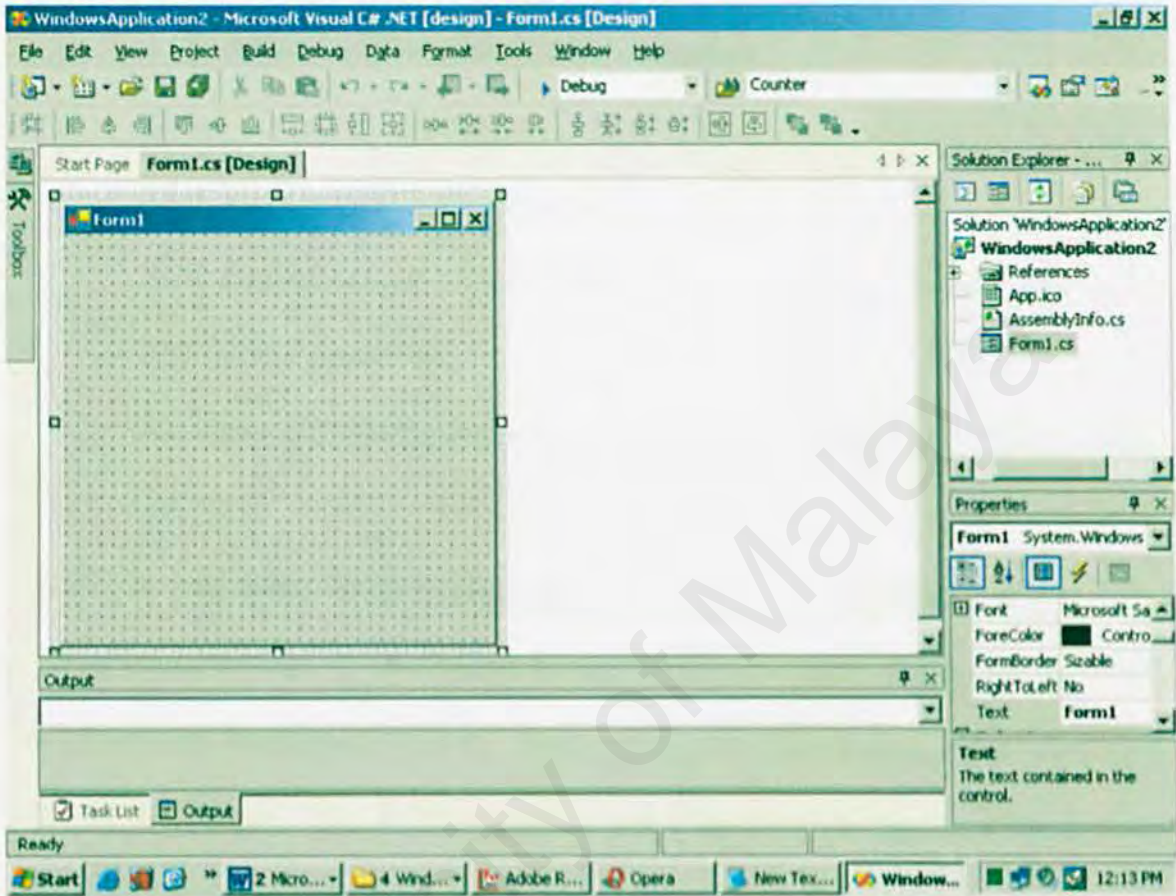


Figure 2.23: MS Visual Studio.NET Working Environment

2.7 Database

2.7.1 XML

What is XML?

- XML stands for EXtensible Markup Language
- XML is a markup language much like HTML.
- XML was designed to describe data.
- XML tags are not predefined in XML. You must define your own tags.
- XML is self describing.
- XML uses a DTD (Document Type Definition) to formally describe the data.

XML, or Extensible Markup Language, is a markup language that you can use to create your own tags. It was created by the World Wide Web Consortium (W3C) to overcome the limitations of HTML, the Hypertext Markup Language that is the basis for all Web pages. Like HTML, XML is based on SGML -- Standard Generalized Markup Language. Although SGML has been used in the publishing industry for decades, its perceived complexity intimidated many people that otherwise might have used it (SGML also stands for "Sounds great, maybe later"). XML was designed with the Web in mind.

Now let's look at a sample XML document. With XML, you can assign some meaning to the tags in the document. More importantly, it's easy for a machine to process the information as well. You can extract the postal code from this document by simply locating the content surrounded by the `<postal-code>` and `</postal-code>` tags, technically known as the `<postal-code>` element.

```
<address>
  <name>
    <title>Mrs.</title>
    <first-name>
      Mary
    </first-name>
    <last-name>
      McGoon
    </last-name>
  </name>
  <street>
    1401 Main Street
  </street>
  <city state="NC">Anytown</city>
  <postal-code>
    34829
  </postal-code>
</address>
```

There are three common terms used to describe parts of an XML document: tags, elements, and attributes. Here is a sample document that illustrates the terms:

- A tag is the text between the left angle bracket (<) and the right angle bracket (>). There are starting tags (such as <name>) and ending tags (such as </name>)
- An element is the starting tag, the ending tag, and everything in between. In the sample above, the <name> element contains three child elements: <title>, <first-name>, and <last-name>.
- An attribute is a name-value pair inside the starting tag of an element. In this example, state is an attribute of the <city> element; in earlier examples, <state> was an element

XML can be used to exchange data. In the real world, computer systems and databases contain data in incompatible formats. One of the most time consuming challenges for developers has been to exchange data between such systems over the

Internet. Converting the data to XML can greatly reduce this complexity and create data that can be read by different types of applications.

XML can be used to store data. XML can also be used to store data in files or in databases. Applications can be written to store and retrieve information from the store, and generic applications can be used to display the data.

2.7.2 MySQL (Database Management System)



Figure 2.24: Logo of MySQL

MySQL is a multi-user SQL **relational** database server. It can run on most unix platforms, windows and OS/2. Relational databases store information in tables with rows and columns. SQL (Structured Query Language) is high-level, declarative database language used for defining and querying relational databases. MySQL, the most popular open source SQL database, is developed and provided by MySQL AB. MySQL AB is a commercial company the builds its business providing services around the MySQL database. The MySQL software delivers a very fast, multi-threaded, multi-user, and robust SQL database server. It is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software. MySQL is a trademark of MySQL AB For more information about MySQL visit the MySQL website. <http://www.mysql.org>.

2.7.3 Microsoft SQL Server 2000

Microsoft SQL Server 2000 is a full-featured relational database management system (RDBMS) that offers a variety of administrative tools to ease the burdens of database development, maintenance and administration. Microsoft SQL Server 2000 consist of the six of the more frequently used tools: Enterprise Manager, Query Analyzer, SQL Profiler, Service Manager, Data Transformation Services and Books Online.

Enterprise Manager is the main administrative console for SQL Server installations. It provides you with a graphical "birds-eye" view of all of the SQL Server installations on your network. You can perform high-level administrative functions that affect one or more servers, schedule common maintenance tasks or create and modify the structure of individual databases.

Query Analyzer offers a quick and dirty method for performing queries against any of your SQL Server databases. It's a great way to quickly pull information out of a database in response to a user request, test queries before implementing them in other applications, create/modify stored procedures and execute administrative tasks.

SQL Profiler provides a window into the inner workings of your database. You can monitor many different event types and observe database performance in real time. SQL Profiler allows you to capture and replay system "traces" that log various activities. It's a great tool for optimizing databases with performance issues or troubleshooting particular problems.

Service Manager is used to control the MSSQLServer (the main SQL Server process), MSDTC (Microsoft Distributed Transaction Coordinator) and SQLServerAgent processes. An icon for this service normally resides in the system

tray of machines running SQL Server. You can use Service Manager to start, stop or pause any one of these services.

Data Transformation Services (DTS) provide an extremely flexible method for importing and exporting data between a Microsoft SQL Server installation and a large variety of other formats. The most commonly used DTS application is the "Import and Export Data" wizard found in the SQL Server program group.

2.7.4 Oracle

Oracle9i Database Server describes the most recent major version of the Oracle Relational Database Management System (RDBMS) family of products that share common source code. Leveraging predecessors including the Oracle8 release that surfaced in 1997, the family includes:

- Personal Oracle, a database for single users that's often used to develop code for implementation on other Oracle multiuser databases
- Oracle Standard Edition, which was named Workgroup Server in its first iteration as part of the Oracle7 family and is often simply referred to as Oracle Server
- Oracle Enterprise Edition, which includes additional functionality

In 1998, Oracle announced Oracle8i, which is sometimes referred to as Version 8.1 of the Oracle8 database. The "i" was added to denote added functionality supporting Internet deployment in the new version. Oracle9i followed, with Application Server available in 2000 and Database Server in 2001. Oracle has focused development

around a single source code model since 1983. While each database implementation includes some operating system-specific source code, most of the code is common across the various implementations. The interfaces that users, developers, and administrators deal with for each version are consistent. Features are consistent across platforms for implementations of Oracle Standard Edition and Oracle Enterprise Edition. As a result, companies have been able to migrate Oracle applications easily to various hardware vendors and operating systems while leveraging their investments in Oracle technology. From the company's perspective, Oracle has been able to focus on implementing new features only once in its product set, instead of having to add functionality at different times to different implementations.

2.8 Summary

In this chapter, many information and resource has to be gathered and reviewed. Internet is a good medium to get all those information. Anyway, it can be summarized that any type of information that is related or linked to this project has been reviewed in order to understand to system that going to be develop. Moreover, review has been done on current exiting system and system that has close similarity to this project in order to adopt some beneficial features. To ease the system development, review on tools of various kinds to identify which of these tools is suitable.

CHAPTER 3 METHODOLOGY

3.1 Introduction

Numerous standards and process models apply to the software development industry. Developing computer systems (software) is a complex task. It requires a disciplined approach to the methods used. A process is a series of steps involving activities, constraints and resources that produce an intended output of some kind. Usually, a process involves a set of tools and techniques. Software development normally involves the following stages:

- Requirements analysis and definition
- System design
- Writing the programs (program implementation)
- Unit testing
- Integration testing
- System testing
- System delivery
- Maintenance

Each stage itself is a process (or collection of processes) that can be described as a set of activities. Each system development process begins with system requirement (input) and process flow until finish as a product (output) (see Figure 3.1).

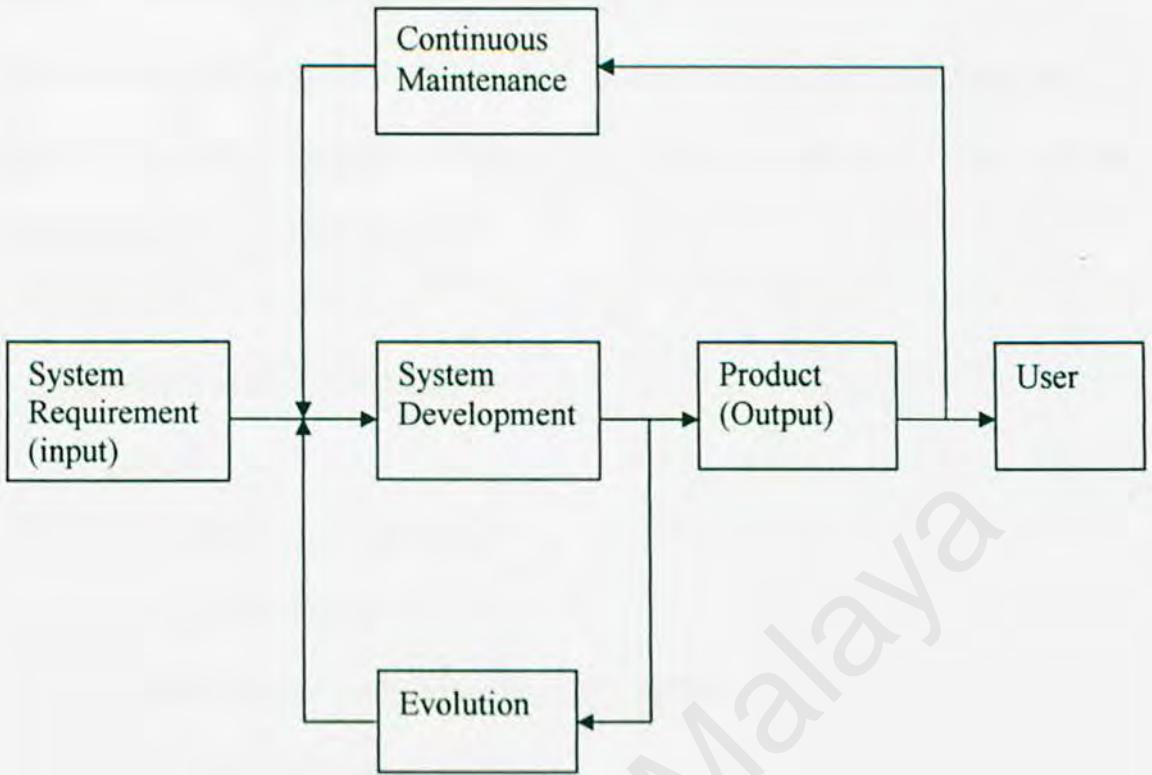


Figure 3.1: System Development Process Model

3.2 System Development Methodology

Several methodologies are available. Each model has advantages and disadvantages depend on scale, complexity, budgeted, time constraints and stuff. Some of the Methodology are available as:

- Waterfall Model
- Waterfall Model with Prototype
- V Model
- Operational Specification Model
- Transformation Model
- Dynamic System Development Method (DSDM)
- Context Driven Testing
- Spiral Model
- Rapid Development
- Extreme Programming (XP)
- Iteration-and-Increment Life Cycle Model

There are so many methodology available. The System Development Methodology that I choose in my software development system is Spiral Model.

3.2.1 Spiral Model

The spiral model, illustrated in Figure 3.2, combines the iterative nature of prototyping with the controlled and systematic aspects of the waterfall model, therein providing the potential for rapid development of incremental versions of the software. In this model the software is developed in a series of releases with the early stages being either paper models or prototypes. Later iterations become increasingly more complete versions of the product.

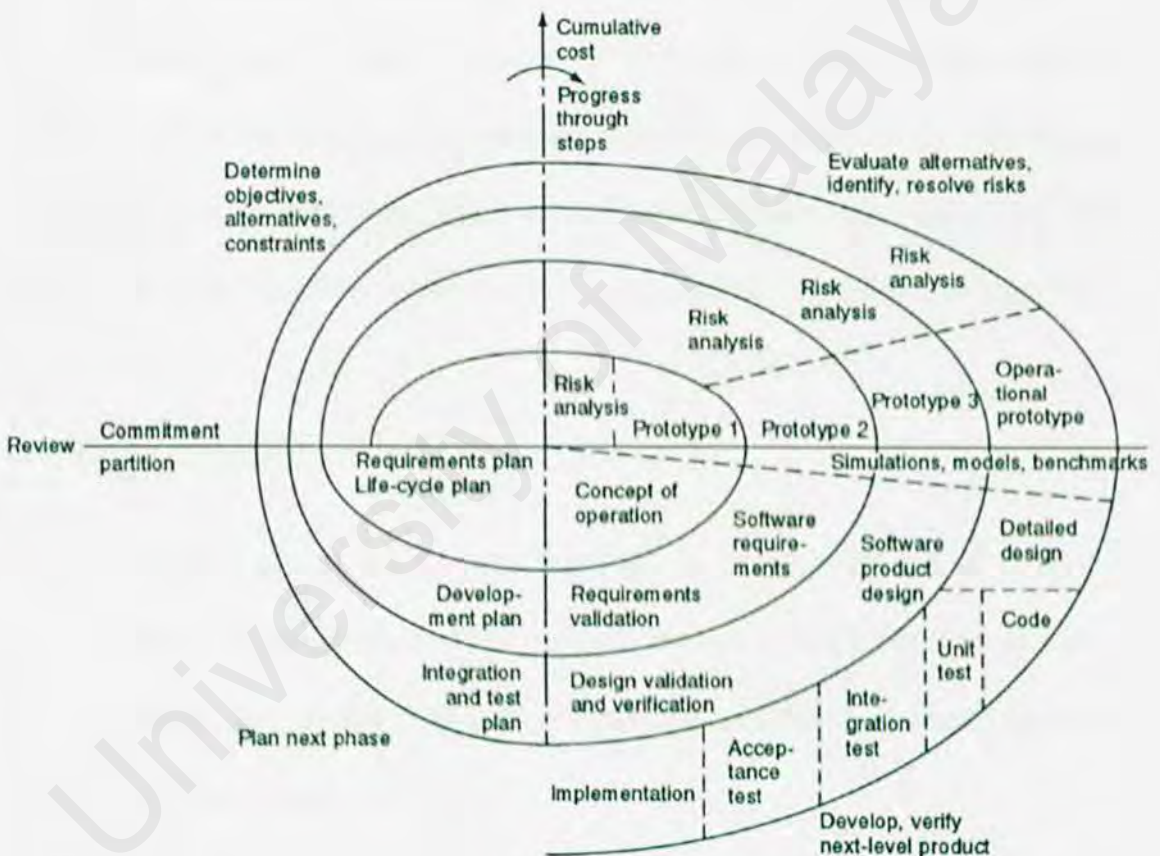


Figure 3.2: Spiral Model

As illustrate in Figure 3.2, the model is divided into a number of task regions.

The evolutionary process begins at the centre position and moves in a clockwise direction. Each traversal of the spiral typically results in a deliverable. For example, the first and second spiral traversals may result in the production of a product specification and a prototype, respectively. Subsequent traversals may then produce more sophisticated versions of the software.

An important distinction between the spiral model and other software models is the explicit consideration of risk. There are no fixed phases such as specification or design phases in the model and it encompasses other process models. For example, prototyping may be used in one spiral to resolve requirement uncertainties and hence reduce risks. This may then be followed by a conventional waterfall development.

Advantages of the Spiral Model

- The spiral model is a realistic approach to the development of large-scale software products because the software evolves as the process progresses. In addition, the developer and the client better understand and react to risks at each evolutionary level.
- The model uses prototyping as a risk reduction mechanism and allows for the development of prototypes at any stage of the evolutionary development.
- It maintains a systematic stepwise approach, like the classic life cycle model, but incorporates it into an iterative framework that more reflect the real world.
- If employed correctly, this model should reduce risks before they become problematic, as consideration of technical risks are considered at all stages.

Disadvantages of the Spiral Model

- Demands considerable risk-assessment expertise
- It has not been employed as much proven models (e.g. the WF model) and hence may prove difficult to 'sell' to the client (esp. where a contract is involved) that this model is controllable and efficient. [More study needs to be done in this regard]

University of Malaya

3.3 Fact Finding Techniques

Through the methodology phase, information is obtained by a process called requirement determination which is also known as fact-finding. Fact-findings are essential in order to have a thorough understanding of the system to be developed. The fact-finding techniques used include library research, interview, Internet surfing, observation and using a questionnaire.

3.3.1 Library Research

Library is a good place to find the resources. Related books were borrowed from library to have a deeper understanding on system design and implementation of the VNC application.

3.3.2 Internet Surfing

The internet is a informative place where student can find a lot of information online. With search engines such as Google, Excite, Yahoo and Search, relevant information sites can be viewed with only a click away, provided users key in the related keywords in the search box provided. There are many sites available that provide information as guidelines in determining the system requirements.

3.3.3 Brainstorm

During the requirements elicitation, discussions with the supervisor have been carried out in order to discuss about the requirements of the system. Abundance of ideas about the design and requirements of the system have been figured out.

3.3.4 Other References

Besides that, a lot of studies also have been done through the reading material such as books, magazines and journal. Furthermore, some researches also have been carried out in the document room by reading the senior's thesis. From there, quite a lot of ideas about the system requirements and its design have been arisen.

3.4 Summary

The methodology used to guide the development of the system is the Spiral Model. Requirement analysis is also being defined. There are two types of the requirements defined, the functional requirements and the non- functional requirements. Analyzing the type of the system will help in developing the system.

University of Malaya

CHAPTER 4 SYSTEM ANALYSIS

4.1 Introduction

System analysis is an explicit formal inquiry carried out to help someone (referred to as the decision maker) identify a better course of action and make a better decision than he might otherwise have made. The characteristic attributes of a problem situation where systems analysis is called upon are complexity of the issue and uncertainty of the outcome of any course of action that might reasonably be taken. Systems analysis usually has some combination of the following: identification and re-identification) of objectives, constraints, and alternative courses of action; examination of the probable consequences of the alternatives in terms of costs, benefits, and risks; presentation of the results in a comparative framework so that the decision maker can make an informed choice from among the alternatives.

After doing research and study on current software development tools that have been carried out and discussed in Chapter 2, a most suitable tool for the system development have been decided after considering their benefits and advantages. These tools include application platform and operating system, programming languages, and technologies.

System analysis plays an important role in order to meet the requirements of the system. The system definitely cannot be accomplished without the existence of these needed requirements. These requirements are further divided into two categories; functional and non-functional requirements. Besides that, software and hardware requirements have been identified to ease the development process.

4.2 Functional Requirements of the System

Functional Requirements is a services provided to the user and to other systems. Functional Requirements describes an interaction between the system and its environment. Further, the functional requirements, the system can be partitioned into a set of logical minimally related pieces, each of which describes some way in which the system will function. This technique is known as use case. The main modules of UMVNC applications are illustrated in figure below:

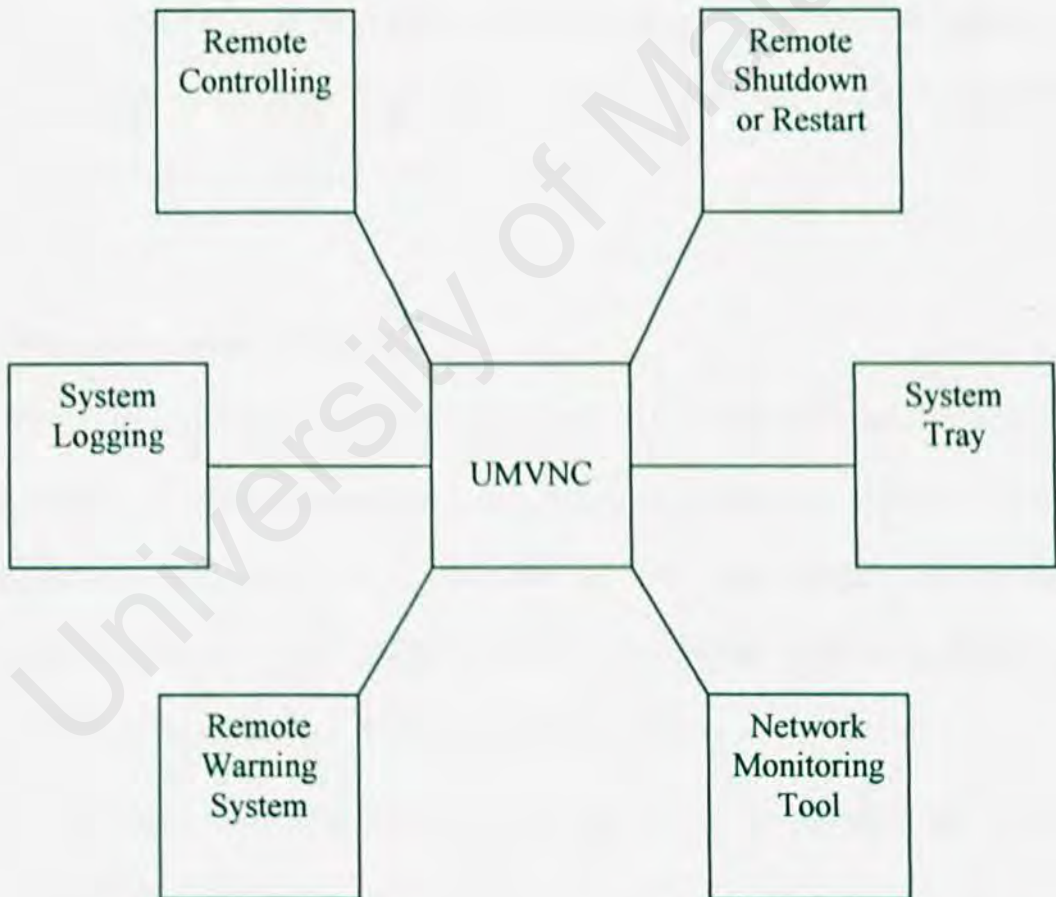


Figure 4.1: Scope of the Project

4.2.1 Remote Controlling

The network administrator or network technician can remotely control the target machine (computer), spy the student's activity at computer lab. He also can change the setting of the remote host by using UMVNC application. This application help them to manage the network computer easily. The mechanism is as below

- The target host must running the agent at the background without the notification of the student or with the notification also can.
- The network administrator or network technician use his computer with UMVNC application running to connect to the target host.
- If the connection is established, network administrator or network technician can view the computer screen of the student, monitoring, controlling even he can change the computer setting remotely.

4.2.2 Remote Shutdown or Restart

Remote Restart and Shutdown is one of the great feature integrated into UMVNC. It allow the network administrator or network technician to restart or shutdown all the computer remotely. With this feature the network administrator or network technician no need to shutdown computers manually. At the result, it save time to manage the network computer. The mechanism is as below

- The target host must running the agent at the background without the notification of the student or with the notification also can.
- The network administrator or network technician just send a signal to the target network (IP multicast), then all the computer in the network will shutdown or restart. A confirmed message will be sent back.

4.2.3 Remote Warning System

The network administrator or network technician can send a warning message to all the computer at local area network. At the client side, a warning message box will appear to let them know that the warning message has been sent by the network administrator or network technician. The mechanism is as below

- The target host must running the agent at the background without the notification of the student or with the notification also can.
- The network administrator or network technician just send a warning message to the target host. The agent of the target host will accept the message then a warning message box will be prompted. So that the student will see the warning message send by network administrator or network technician.

4.2.4 Network Monitoring Tool

The network administrator or network technician are capable of monitoring the download and the upload speed, such as the transfer rate of the current network activity. Besides that, they network monitoring will calculate all the download and upload bytes. This is just a simple tool to monitoring the network activity of the local computer.

4.2.5 Logging System

The logging system is created to let network administrator or network technician to check what action has been performed last time. The logging system will store the data like what and when the action have been performed.

4.2.6 System Tray

The UMVNC has a function to hide the application at the System Tray. A icon will be display at the System Tray.

University of Malaya

4.3 Non-Functional Requirements Analysis

Non-Functional Requirements describe how the system will do it. Non-functional requirements describe the constraints and restriction of the system that limit the choices for constructing a solution to the problem. These constraints narrow the selection of language, platform, or implementation techniques or tools. The UMVNC must ensure certain VNC application qualities like user-friendly, flexibility, security, correctness, functionality, reliability and accuracy, efficiency as well as maintainability. The following non-functional requirements have been considered for UMVNC.

4.3.1 User-Friendly

The user interface and design of the system should be user-friendly to enhance the interaction between users and the system. The suitable and meaningful graphics, images, buttons are required to enhance the user friendliness of the system. Adequate messages and response given by the system are required to provide a good impression to a user in using the system. The interface design should apply to consistent fashion where all visual information must be organized according to a design standard that is maintained throughout all screen displays. Apart from that, input mechanisms are constrained to a limited set that are used consistently throughout the application.

4.3.2 Flexibility

The system should be flexible and able to fulfill all user requirements. The system, software and hardware should have the capability to take advantage of the new technologies and resources and can be implemented in the changing environments. It must be able to cater for all types of information.

4.3.3 Security

The system should provide certain level of security to ensure only the authorized user can access to any functionality in UMVNC. The logon process should be implemented to allow only network administrator and network technician to use the UMVNC.

4.3.4 Correctness

A program or system must operate correctly or it provides little value to its users. Correctness is the degree to which the software performs its required function. To ensure this application quality, lots of testing and trial-and-errors will be carried out.

4.3.5 Functionality

The functionalities stressed here are the remote controlling and remote shutdown restart, which is very important in any VNC application that deals with remote host. The network administrator or network technician can full use the functionality provided.

4.3.6 Reliability and Accuracy

Reliability is referred to the expectation of a system to perform its intended function accurately. The system should be able to handle unexpected error and give proper response to the user to enhance the reliability, accuracy as well as robustness of the system. It is closely related to correct link processing, error recovery and user input validation and recovery. This quality is essential as it indicates how far users will be confident in the implementation of the new computerized system in getting daily minutes processing done.

4.3.7 Efficiency

Undeniable, efficiency is the main key for implementing the new meetings management system. Efficiency is understood as the ability of a process procedure to be called or accessed unlimitedly to produce similar performance outcomes at an acceptable or credible speed. Efficiency is measured base on response time performance and transfer rates.

4.3.8 Maintainability

The sections within the system should be easy to handle to ensure the maintenance can be done regularly. System maintenance accounts would require more effort if the system is not designed according to good programming practices. Maintainability is the ease with which a program can be corrected if an error is encountered, adapted if its environment changes, or enhanced if the customer desires a change in requirements. Besides, evolutionary of the system will easy to be done.

4.4 Decision on Choosing Technology, Software and Development

Tool

4.4.1 Development Framework

Microsoft .NET Framework

Microsoft .NET has many powerful feature that programmer will looking for. It provide a very easy way to develop a new software. Microsoft .Net Framework is chosen as the development framework because it is specifically design and has unique features in developing a VNC application. It provides convenience and easier way to establish a fully operable system without any obstacle.

The platform technologies used are different in both .NET and J2EE.

Table 4.1: Comparison of Microsoft .NET and J2EE

Feature	Microsoft .Net	J2EE
Object-oriented programming language; single inheritance from one base class	C#	Java
Runtime system manages memory references and allocations, controls execution	CLR + metadata, runtime attribute	JVM + secured operations
Reusable components JavaBeans + info on how to use and deploy	.DLLs	CLR
A large library of base components	.NET common components	JDK
Web page development framework	ASP.NET, Web Services	JSP, Servlets
Database access framework	ADO.NET	JDBC, SQLJ
GUI design	Win Forms and Web Forms	Java Swing
Middleware and Application Integration	DCOM	EJB, J2EE, XML
Mobile, PDA and Smart Devices	.NET Mobile Edition	J2ME

Cross language development	18 languages	2 or 3 languages
Vendor Neutrality	Not vendor neutrality, as it is tied to Microsoft	Not totally vendor neutrality
Interoperability	Stronger technology neutral eCollaboration strategy as .NET is designed to interoperate between platforms.	The scope of J2EE is limited to J2EE vendors and COBRA vendors.
Scalability	Can support a large amount of workload with lower cost.	IT is not proven that it can handle these workload at any price.
Framework	.NET platform includes on eCommerce framework called Commerce Server	There is no equivalent vendor-neutral framework in J2EE. Have to build a new application form scratch.

Table 4.1 compares each and every features that both platform have to act a development framework. From the table, clearly is shown that Microsoft .NET can achieve more advantages in developing the system if compare to J2EE. Development process will be smoothly proceeding till the completion of the system.

The .Net initiative promises to make development of distributed components within the Windows platform a lot easier to do, with big improvements coming in six to nine months with the release of Visual Studio.NET, ASP.NET and ADO.NET. Over the next year, Windows XP, formerly called Whistler, will appear, complete with some .Net Web Services and building block services. But as with Office 10 and the .NET Enterprise Servers, the incorporation of complete support for such .Net facilities as managed executables, ASP.NET, ADO.NET, JScript.NET, Visual Studio for Applications.NET, and so on, will have to wait for the next iteration of Windows.

As a conclusion, Microsoft .NET framework is chosen as a development framework.

4.4.2 Development Platform (Operating System)

Microsoft Windows XP Professional

Microsoft Windows XP Professional has a very powerful, user-friendly and interactive Graphical user interface (GUI) and stable compare to the previous version of the windows OS. It is chosen as the development platform for the software developing in UMVNC. This is because it's strengths and reliability compares to the other operating system like linux and unix. Microsoft Windows XP Professional is built to work with a series of microprocessor from the Intel Corporation that share the same or similar sets of instructions.

The network computer lab has the powerful computer with Microsoft Windows XP operating system installed. Windows XP is commonly use nowadays and it is easy to get in the faculty laboratories. Therefore, the implementation of the proposed system can be done easily and effectively without facing the issue of operating system.

Besides that, one of the features like System Recovery will automatically restore the previous setting when computer is crashing. It make a good job especially for those who are not a computer troubleshooter. Windows's graphical user interface (GUI) also makes it easy to setup application without having to enter a command.

4.4.3 Programming Language

Visual C# .NET

Visual C# .NET 2003 is the modern, innovative programming language and tool for building .NET-connected software for Microsoft Windows, the Web, and a wide range of devices. With syntax that resembles C++, a flexible integrated development environment (IDE), and the capability to build solutions across a variety of platforms and devices, Visual C# .NET 2003 significantly eases the development of .NET-connected software.

I will choose Visual C#.NET to develop the software because C# is a great language, and I have found it to have a number of advantages over other programming language. The reasons to choose C# as below

- **Foreach** : this is a simple, but powerful addition to a programming language, and something grabbed straight from the Visual Basic world. Operating in a similar fashion to a C-style "for" loop, it allows you to iterate over the contents of a collection without having to access elements in an array by index
- **Delegates and Events**: In Java, events are modeled after the "Listener" design pattern. Objects which generate events provide a set of methods to add or remove "listener" objects. These listener objects implement a particular interface specified by the event generating object. Events are generated when the event generating object calls methods on the registered event listener object. NET follows the Listener pattern, but with a new twist. First, .NET includes "delegates" which are a special type of object that wraps a method, static or member, that corresponds to a particular signature. Microsoft calls

them "typesafe function pointers," which seems an apt description. A method wrapped inside an event delegate can be passed to an event property (which encapsulates the add and remove logic in one location) that is declared as part of an event generating class, and which is clearly marked in metadata as an event registration point. For instance, through reflection, you can ask any class for the events it supports by calling `GetEvents()` on its `Type` object (which is similar in concept to the Java class object), irrespective of naming convention.

- **No Exception Specifications/Support for Nested Exceptions:** In my experience as a Java programmer, it is very common to get an exception that is NOT the real cause of the problem. I track down the location in source code from which the exception was thrown (using stack trace information), only to find that the exception was thrown within a catch block, effectively hiding the lower level exception (and, potentially, the real cause of the problem) from me.. This is an accidental byproduct of a rule which says you have to declare in every method the exception(s) that will be thrown. Since this list can grow quite large for methods that call large numbers of exception throwing methods, it is common practice to specify a common base exception for your library, creating derived exceptions from that base as needed. Lower level exceptions are trapped within a method, and the application throws a new exception that meets the requirements of the exception specification.
- **Properties:** Properties are a common programming paradigm. Java programmers specify ordinary methods as properties by using the naming convention of "getXYZ" for property retrieval and "setXYZ" to set the value. Development environments know to look for this naming convention and

treat values that follow the convention as properties. In contrast, .NET in general and C# in particular includes explicit support for properties as separate entities from methods. System.Type includes methods for retrieving properties (System.Type.GetProperties and System.Type.GetProperty).

- **Indexers:** C# allows objects to be treated like arrays through the use of an indexer. The indexer can take any type, and return any type of value. For instance, a collection object can provide access to its contents by index (an integer), by name (a string), or some arbitrary 3rd object type. The presence of indexers in C# explains why ArrayList and Hashtable can be accessed using array-style syntax. This should appeal to C++ programmers accustomed to STL (Standard Template Library) conventions. C# doesn't include support for generics (C++ templates are an example of generics), but that's coming, and in a form that might avoid the code bloat issues associated with C++ templates.
- **A preprocessor:** They are used to manage conditional compilation. Conditional compilation allows programmers to include optional code into the final executable based on the presence of certain preprocessor constants. For instance, a common use of conditional compilation is for the inclusion of debug-related code. Such debug code might perform operations that you wouldn't want present in the released program. Conditional compilation allows this code to be included or excluded as necessary. C# preprocessor support would be familiar to anyone accustomed to the C/C++ preprocessor. Constants are defined using #define, or as part of the build environment. Conditional statements use the #if, #else, #elif and #endif statements.

- **The using code block:** In languages such as C++, you can determine exactly when memory will be freed for reuse. Resources can be released immediately through use of the delete operator, or at the end of a particular scope (such as a method declaration) for objects declared on the stack. With Java and .NET, however, this is not the case. Memory is managed by the runtime, which means you cannot know exactly when your objects will be destroyed. This is a problem in cases where you are using resources that need to be freed for reuse as soon as possible. Good examples include open files with exclusive read/write access, and database connections retrieved from a shared pool. The common paradigm in Java is to define a "close()" method on objects that require such immediate cleanup. This method needs to be called in all cases, whether or not a particular block of code has exited normally. The standard way to do this is to place the call to close in the "finally" clause of a "try" exception handling block.
- **More flexible source code organization:** Java classes must be defined in a file named after the public class it contains (only one of which can exist in the file). Furthermore, the programmer must place the source file in a directory structure that matches the package associated with the class. For instance, if the package directive at the top of the page reads "turtleneck.core", your source files must usually be located in a directory named turtleneck\core. I've never liked this restriction. For small applications, you don't need such a strict code hierarchy, and either way, it should be up to developers to decide where to place their files. C# gets rid of this requirement. This might lead to longer source code files, as developers might choose to put more than one public

class in a source file. C# remedies this issue through use of the "#region" precompiler directive.

- **Really simple native invocation:** This will make Java purists scream, as Java isn't supposed to need native invocation. Perhaps that deep-seated belief is what guided the Java development team to turn native code execution into a programmer's version of the Chinese water torture. JNI is a HORRIBLY messy C API, and anyone who likes it probably hangs around New York's Little Italy spitting half-chewed Milk Duds at Tony Soprano look-a-likes. There are LEGITIMATE REASONS for needing to call into native code, if for no other reason than that we don't live in a world where everything can be done in managed code (read: inside a Java VM or the .NET runtime). C# makes it really simple to specify a library function for use in a C# class.
- **Attributes:** Attributes aren't exclusive to C# as they are defined as part of the .NET CLI. However, since C# is a .NET language and makes their use almost trivial, they are worth a mention. ALL bracketed information will be stored as part of the assembly as serialized data in the case of custom attributes, or as special flags on the object for special attributes (such as "Serializable") which are directly supported by the .NET runtime. Attributes can be applied at the class, interface, struct, method, property, parameter and return value level.

At the Conclusion, I will choose Visual C#.NET to develop the software. Visual C#.NET have a better IDE compare to any Java IDEs. Though C#, like Java, runs through a virtual machine, the .NET virtual machine is far superior more than Java virtual machine.

C#, for all intensive purposes, is the premier language of .NET. It was written explicitly to support features found in .NET, and has become the de facto language used in .NET-oriented training materials.

In other words, most .NET programmers will come across C# code frequently, and will acquire a strong familiarity with the language.

4.4.4 XML

The database system that I use to develop my application is XML. Here are the reason why I choose XML.

- XML is a meta language
- XML is text-based and easy to read
- XML is ideal for structured documents
- XML is presentation neutral
- XML is multilingual
- XML helps businesses to integrate
- XML is open

4.5 Development Requirements

The following are the hardware requirements and software requirements for the development of the UMVNC. It is the minimum requirement in order to develop the UMVNC application. The faster machine sure will give u the faster throughput

Hardware requirements:

- PC with at least Pentium III 600 MHz processor
- At least 256 MB of Random Access Memory (RAM)
- At least 20GB of hard disk space.
- Fast Network Interface card.
- 17 inch CRT Monitor
- Other standard computer peripherals.

Software requirements:

- Microsoft Windows XP Professional Edition
- Microsoft Visual Studio .NET 2003 Professional Edition
- Microsoft .NET Framework 1.1
- Adobe PhotoShop 6.0 for image editing

4.6 User client Requirements

The following are the hardware and software requirements for the user in order to function properly with UMVNC

Client Hardware Requirements

- PC with at least Pentium 200 MHz and above
- At least 128 MB of Random Access Memory (RAM)
- Fast Network interface card.
- At least 10 GB hard disk space
- 15 inch CRT Monitor
- Other standard computer peripherals

Client Software Requirements

- Microsoft Windows 98 Second Edition or above
- Microsoft .NET Framework 1.1 or above

CHAPTER 5 SYSTEM DESIGN

5.1 Introduction

System design is a process of transforming the problem into a solution. It can be referred as the description of a solution. The description of a system may change during the system development life cycle since the nature of the solution may change as the solution is described or implemented.

In this phase, the overall architecture for this project will be determined and documented. It includes features and components of the system, data design, architectural design, user interface design and procedural design of the system. The purposes of this phase are to transform the requirements into a functional system and to change the abstract logical model to the concrete physical implementation.

The objectives of system design are listed below:

- Meet User Requirements
- Ease of Use
- Specify Logical Design Elements
- Software Specification

UMVNC is designed in the following design principal

- User Friendly
- Modularity
- Cohesion
- Flexibility

5.2 Overview of UMVNC System Architecture Design

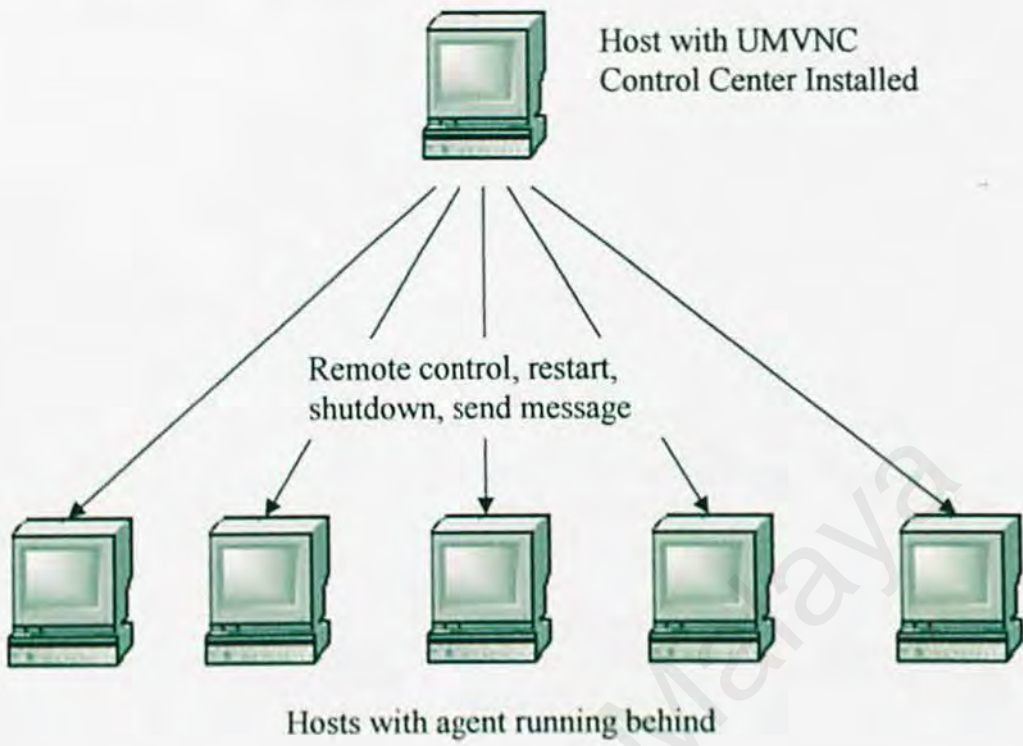


Figure 5.1: UMVNC System Architecture

Work Flow Diagram

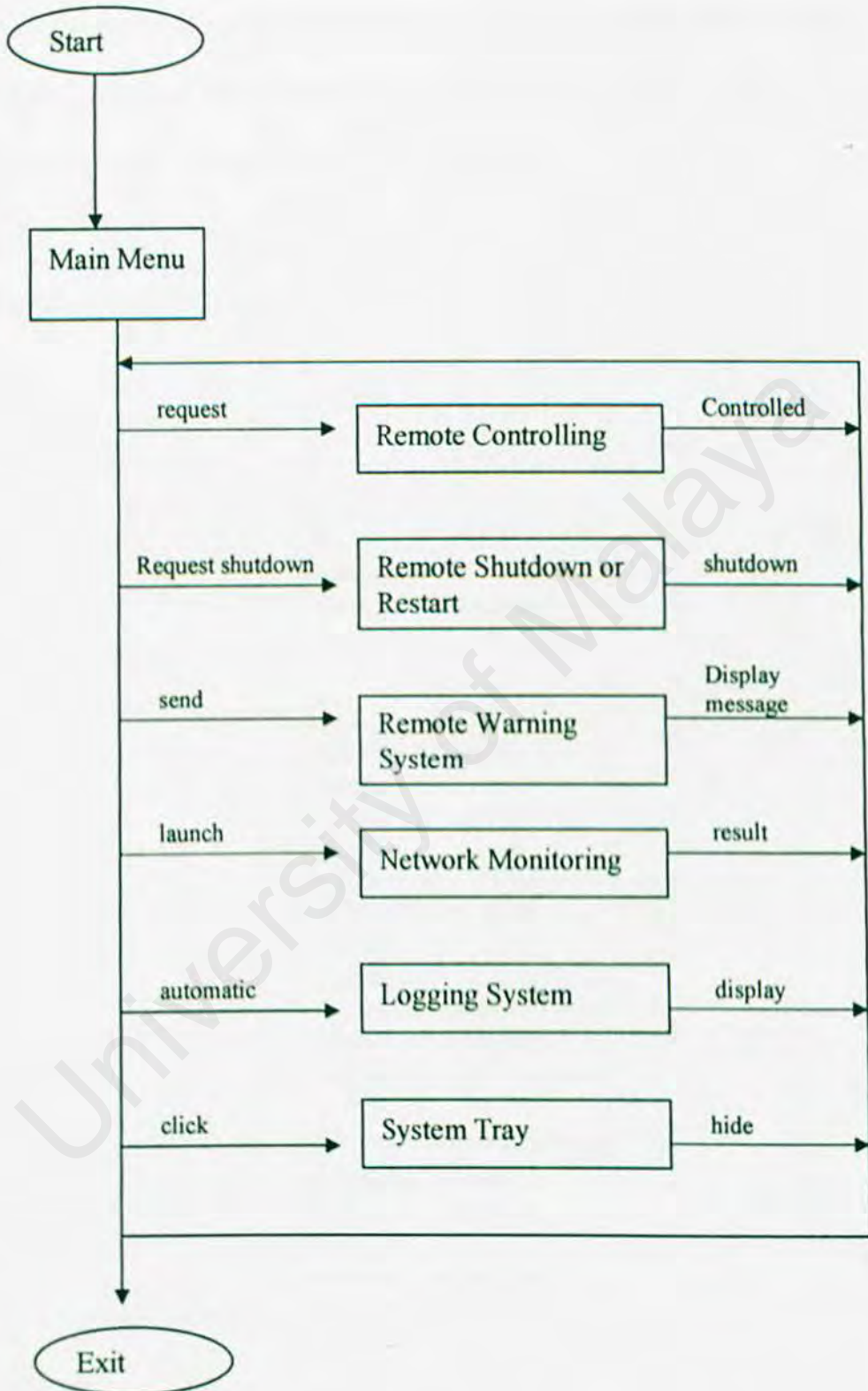


Figure 5.2: Work Flow Diagram of UMVNC

5.3 System Functionality Design

System functionality design is based on the system requirements stated in Chapter 3. It translates the system requirements into system functionality. This design focuses on the system structure design and data flow diagram.

5.3.1 System Structure Chart

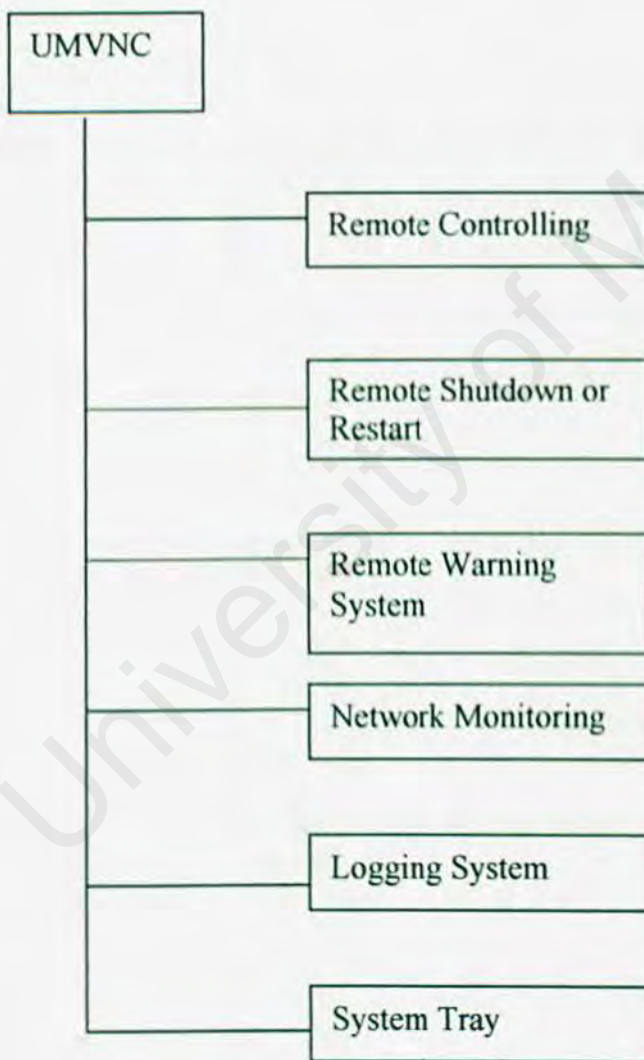
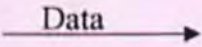

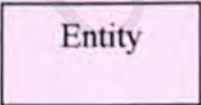



Figure 5.3: System Structure Chart

5.3.2 Data Flow Diagram

Data Flow Diagram (DFD) is concerned with designing a sequence of functional transformation that converts inputs into required output. The diagrams illustrate how data flows through a system and how the output is derived from the input through a sequence of functional transformations. Table 4.1 shows the description of symbols used in DFD.



Table 5.1 Description of symbols used in DFD

<i>Component</i>	<i>Description</i>
	<p>Data Flow - Represent the flow of data or into from one object to another.</p> <ul style="list-style-type: none"> - Arrow denotes the direction of data flow - Each data flow is labeled with the name or details of the info represented by the data flow.
	<p>Data Store - Hold data for a time within the system.</p> <ul style="list-style-type: none"> - Comprise two sections: <ol style="list-style-type: none"> 1. Identifier information 2. Description of the data stored
	<p>Entity - Any objects in the real world, for example person.</p>
	<p>Process - Transform the input data to output data</p> <ul style="list-style-type: none"> - Represent by the rectangle shape - Comprise two or three section: <ol style="list-style-type: none"> 1. Top section contains a description of the

	<p>process.</p> <p>2. Lower section contains the physical location or computer program info.</p>
--	--

In order to simplify and clarify what the data flow diagram is portraying, there are supplemental conventions as shown in Table 5.2.

Table 5.2 Description of symbols used in DFD (supplemental)

<i>Component</i>	<i>Description</i>
	Use a slash in the corner to indicate a duplicate entity
	Use a "hoop" to illustrate that data flow lines cross but not intersect.

5.3.2.1 Context Diagram for UMVNC

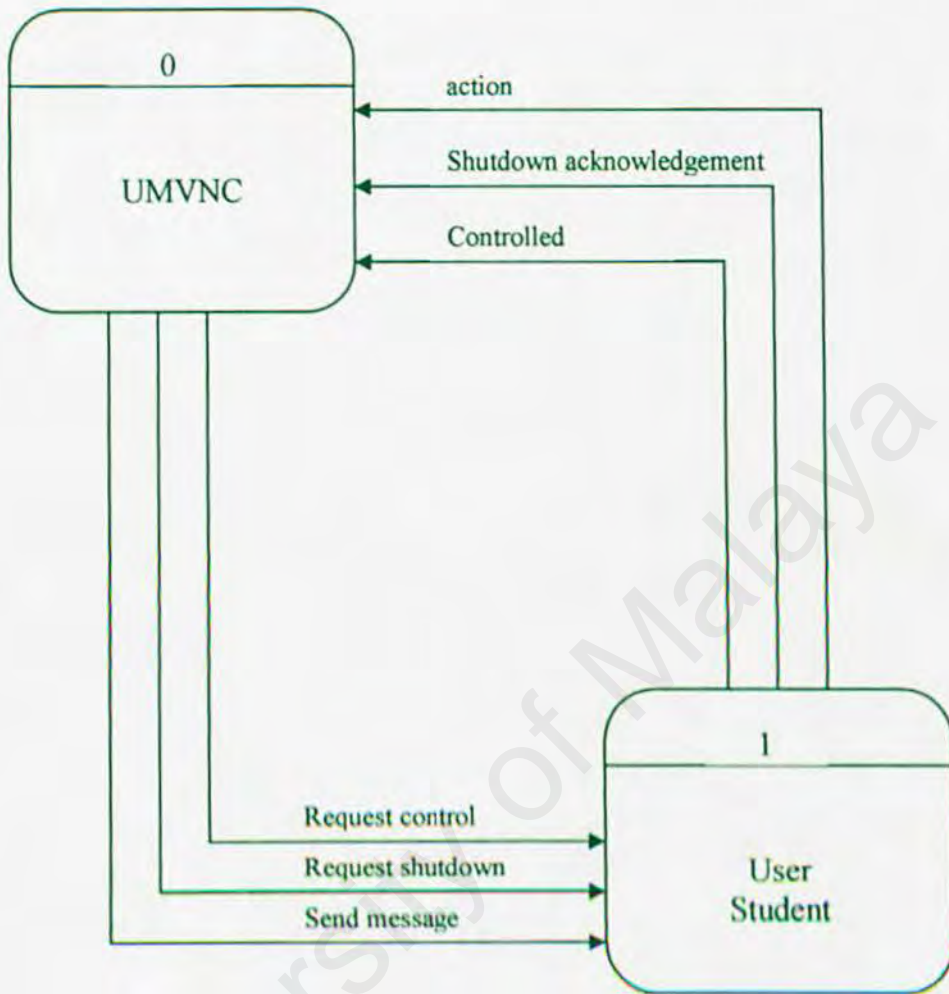


Figure 5.4: Context Diagram for UMVNC

5.3.2.2 Data Flow Diagram for UMVNC

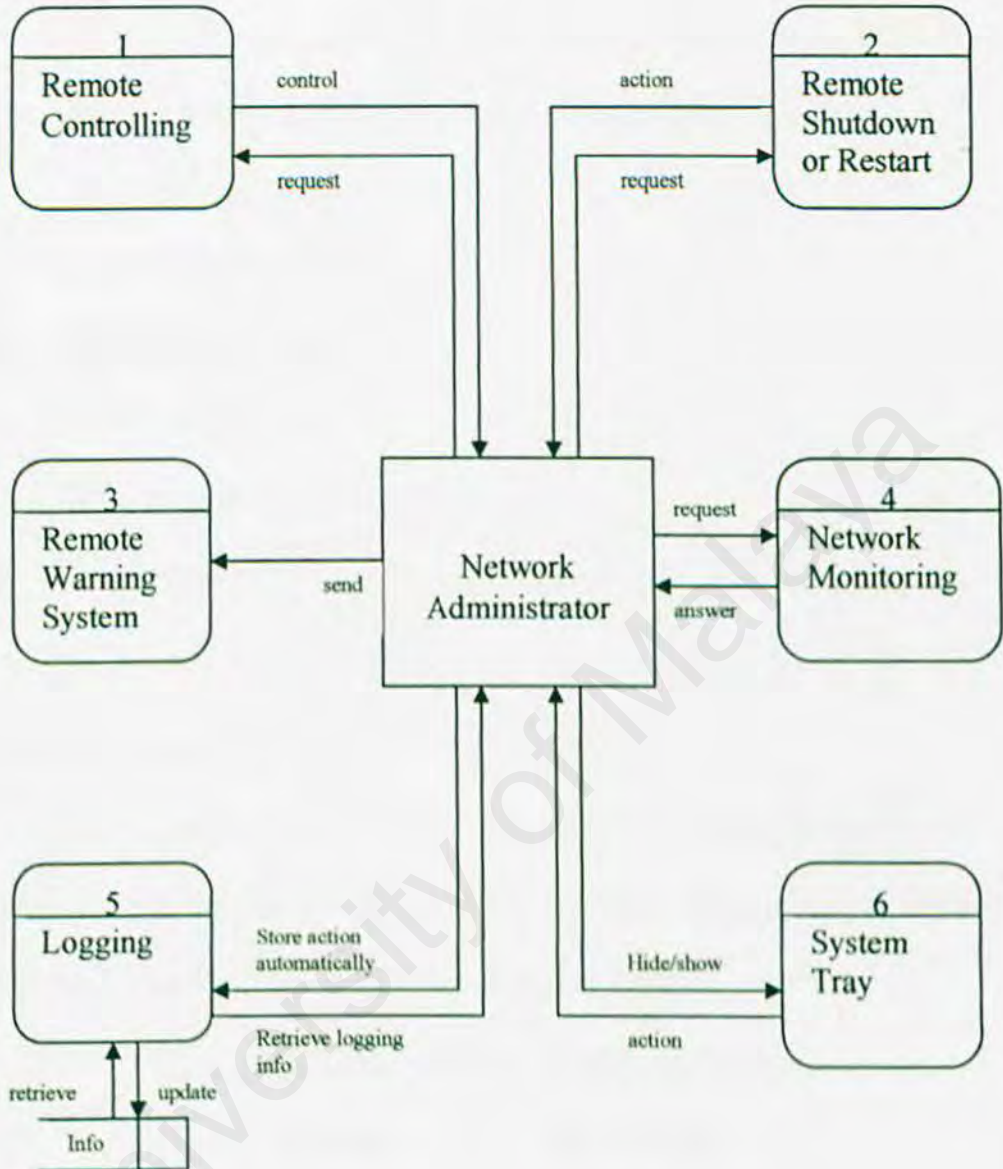


Figure 5.5: Data Flow Diagram for UMVNC

5.4 Database Design

Regardless of the design techniques to be used, well-designed data can lead to the better program structure and modularity, and reduce the procedural complexity. The database is constructed using XML. The design objectives are:

- i) Purposeful information retrieval
- ii) Small size of database
- iii) Efficient data storage
- iv) Data availability
- v) Efficient updating and retrieval
- vi) Data integrity

5.4.1 Data Dictionary

Data dictionary explicitly represents the relationship among the data objects and constraints on the elements of a data structure. The fields, data type, field size and the description of fields for each table are shown as follow:

Table 5.3: XML Schema for Logging System)

Element	Description
activity	What action has been performed
date	Date of the action
time	Time of the action

5.5 User Interface Design

User interface design describes how the software communicates with the humans who use it. The UMVNC graphical user interface is designed to make the usage more easier and a lot of the functions have been integrated into the application.

5.5.1 Concepts of User Interface Design

- **Learnability vs. Usability**

The goal of the user interface should be foremost in the design process. Consider the example of a visitor information system located on a kiosk. In this case it makes perfect sense that the primary goal for the interface designers should be ease of operation for the first-time user. The more the interface walks the user through the system step by step, the more successful the interface would be. User interface design is not a "one size fits all" process. Every system has its own considerations and accompanying design goals. The Requirements Phase is designed to elicit from the design team the kind of information that should make these goals clear.

- **Intuitiveness**

It is generally perceived that the most fundamental quality of any good user interface should be that it is intuitive. The problem is that "intuitive" means different things to different people. To some an intuitive user interface is one that users can figure out for themselves. There are some instances where this is helpful, but generally the didactic elements geared for the first-time user will hamper the effectiveness of intermediate or advanced users.

A much better definition of an intuitive user interface is one that is easy to learn. This does not mean that no instruction is required, but that it is minimal and that users can "pick it up" quickly and easily. First-time users might not intuit how to operate a scroll bar, but once it is explained they generally find it to be an intuitive idiom. Icons, when clearly unambiguous, can help to make a user interface intuitive. But the user interface designer should never overlook the usefulness of good old-fashioned text labels. Icons depicting portrait or landscape orientation, for example, are clearly unambiguous and perhaps more intuitive than the labels themselves, but without the label of "orientation," they could make no sense at all. Labels should be concise, cogent, and unambiguous. A good practice is to make labels conform to the terminology of the business that the application supports. This is a good way to pack a lot of meaning into a very few words. Designing intuitive user interfaces is far more an art than a science. It draws more upon skills of psychology and cognitive reasoning than computer engineering or even graphic design. The process of Usability Testing, however, can assess the intuitiveness of a user interface in an objective manner.

- **Consistency**

Consistency between applications is always good, but within an application it is essential. The standard GUI design elements go a long way to bring a level of consistency to every panel, but "look and feel" issues must be considered as well. The use of labels and icons must always be consistent. The same label or icon should always mean the same thing, and conversely the same thing should always be represented by the same label or icon. In addition to consistency of labeling, objects should also be placed in a consistent manner.

- **Simplicity**

The complexity of computers and the information systems they support often causes us to overlook Occam's Razor, the principle that the most graceful solution to any problem is the one which is the most simple. A good gauge of simplicity is often the number of panels that must be displayed and the number of mouse clicks or keystrokes that are required to accomplish a particular task. All of these should be minimized. The fewer things users have to see and do in order to get their work done, the happier and more effective they will be.

- **Prevention**

A fundamental tenet of graphic user interfaces is that it is preferable to prevent users from performing an inappropriate task in the first place rather than allowing the task to be performed and presenting a message afterwards saying that it couldn't be done. This is accomplished by disabling, or "graying out" certain elements under certain conditions.

- **Forgiveness**

One of the advantages of graphic user interfaces is that with all the options plainly laid out for users, they are free to explore and discover things for themselves. But this requires that there always be a way out if they find themselves somewhere they realize they shouldn't be, and that special care is taken to make it particularly difficult to "shoot themselves in the foot." A good tip to keep users from inadvertently causing damage is to avoid the use of the Okay button in critical situations. It is much better to have button labels that clearly indicate the action that will be taken.

- Aesthetics

Finally, it is important that a user interface be aesthetically pleasing. It is possible for a user interface to be intuitive, easy to use, and efficient and still not be terribly nice to look at. While aesthetics do not directly impact the effectiveness of a user interface, users will be happier and therefore more productive if they are presented with an attractive user interface.

University of Malaya

5.5.2 UMVNC Graphical User Interface

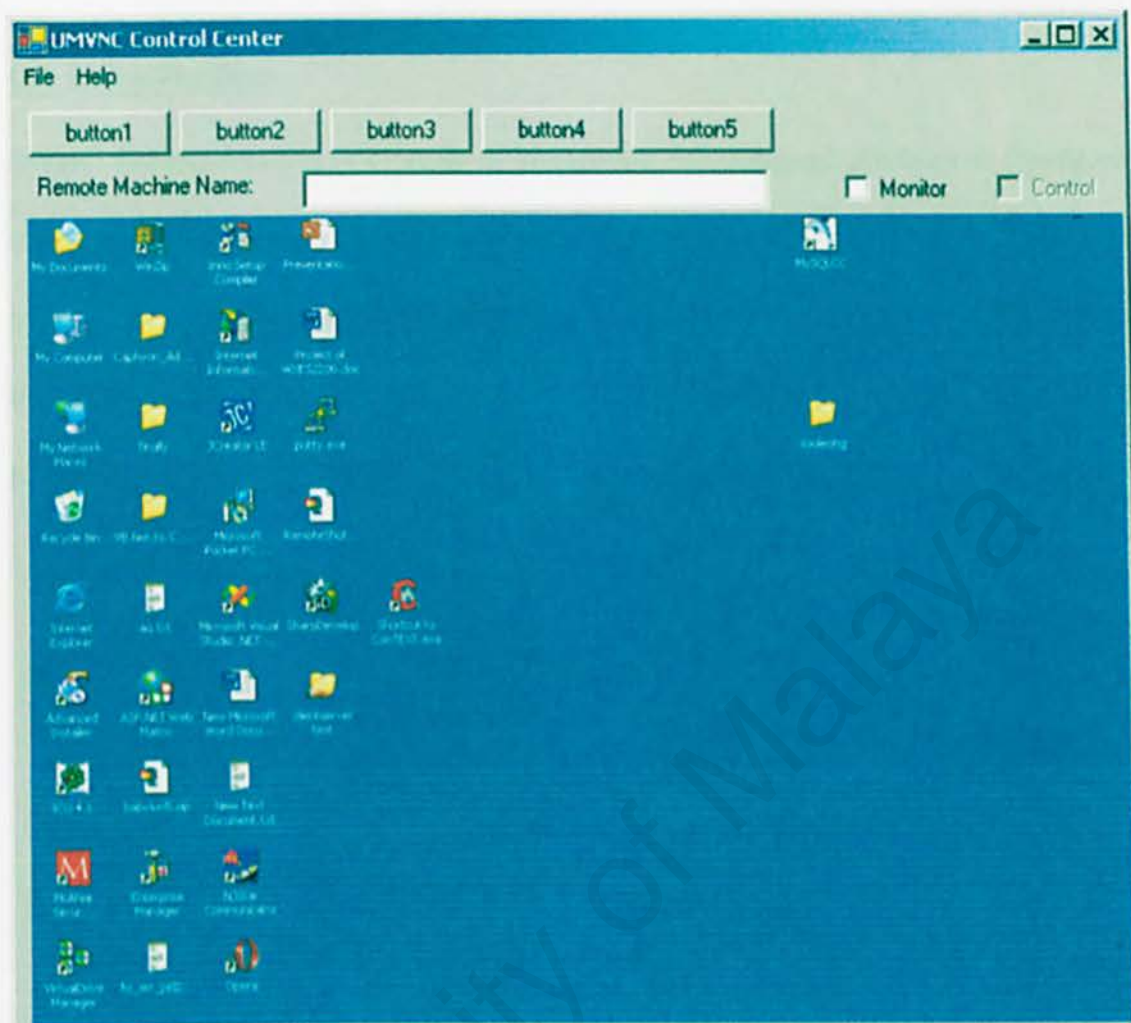


Figure 5.6: UMVNC Graphical User Interface

CHAPTER 6 SYSTEM IMPLEMENTATION

6.1 Introduction

System implementation is a phase of integrating the designed modules or functions of the system based on the requirements. The process of coding is started in this phase where it took most of the time in developing this system. In order to achieve the suggested system, the appropriate tools and languages are needed to code the program. A number of hardware and tools were chosen in this case.

University of Malaya

6.2 Development Environment

In order to maximize the development process an appropriate tools have been selected. Below are the tools:

Table 6.1: The Development Tools

Development Tools	Function	Description
Microsoft Windows XP	Development Platform	Operating System
Microsoft Visual Studio .Net	Programming Tool	System Development Tool
Adobe Photoshop 7.0	Image Creation Tool	To create image
Adobe ImageReady 7.0	Animation Creation Tool	To create gif animation
Icon Editor 5.2	Icon Creation Tool	To create UMVNC icon

6.3 Coding

6.3.1 XML Database (Application Setting)

For UM Remote Agent, I used XML Database to store the configuration setting because it is easy to deploy the application at the network without having to recompile the application.

Table 6.2: Application Setting for UM Remote Agent

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="UDPIpMulticast" value="224.100.0.1" />
    <add key="UDPIpMulticastPort" value="5555" />
    <add key="vncPort" value="9000" />
    <add key="shutdownTimer" value="10" />
    <add key="restartTimer" value="15" />
    <add key="logOffTimer" value="20" />
  </appSettings>
</configuration>
```

6.3.2 C# .NET Remoting (VNC)

.Net Remoting used to establish 1-1 connection between the client and server (agent)

Table 6.3: .Net Remoting Coding

```
public void vncListening()
{
    // Read appSettings from App.config
    string VNCPORT = ConfigurationSettings.AppSettings["vncPort"];
    int intVNCPort = Int32.Parse(VNCPORT);
    TcpServerChannel channel = new TcpServerChannel(intVNCPort);
    ChannelServices.RegisterChannel(channel);
    RemotingConfiguration.RegisterWellKnownServiceType(typeof
    (RemoteObject), "RemoteObject", WellKnownObjectMode.SingleCall);
}
```

6.3.3 C# UDP Programming

for remote shutdown, restart, log off, announcement

Table 6.4: UDP Coding for Remote Shutdown, Restart, Log off, Announcent

```

public void udpListening()
{
    // Read appSettings from App.config/LMRemoteAgent.exe.config
    string UDPIpMulticast =
    ConfigurationSettings.AppSettings["UDPIpMulticast"];
    string UDPIpMulticastPort =
    ConfigurationSettings.AppSettings["UDPIpMulticastPort"];

    int UDPIpMulticastPort = Int32.Parse(UDPIpMulticastPort);
    UdpClient sock = new UdpClient(UDPIpMulticastPort);
    sock.JoinMulticastGroup(IPAddress.Parse(UDPIpMulticast), 50);
    IPEndPoint iep = new IPEndPoint(IPAddress.Any, 0);
    while(true)
    {
        byte[] data = sock.Receive(ref iep);
        string stringData = Encoding.ASCII.GetString(data, 0,
data.Length);
        if(stringData == "Shutdown")
        {
            string strIP = iep.ToString();

            int a;
            a = strIP.IndexOf(":");

            string temp;
            temp = strIP.Substring(0,a);

            shutdownForm shut = new shutdownForm();

```



```
        shut.sourceIP(temp);

        shut.ShowDialog();
    }
else if(stringData == "Restart")
{
    string strIP = iep.ToString();
    int a;
    a = strIP.IndexOf(":");
    string temp;
    temp = strIP.Substring(0,a);
    restart Restart = new restart();
    Restart.sourceIP(temp);
    Restart.ShowDialog();
}
else if(stringData == "Logoff")
{
    string strIP = iep.ToString();
    int a;
    a = strIP.IndexOf(":");
    string temp;
    temp = strIP.Substring(0,a);

    logOff lo = new logOff();
    lo.sourceIP(temp);
    lo.ShowDialog();
}
else
{
    // System logging
    logging announcementLogging = new logging();
```

```
announcementLogging.systemLogging(stringData);  
  
announcement Announcement = new announcement();  
Announcement.message(stringData);  
Announcement.ShowDialog();  
  
}  
  
}  
  
}
```

University of Malaya

6.3.4 System Logging

To record activities that has been performed

Table 6.5: System Logging Coding

```

using System;

using System.Windows.Forms;

using System.IO;

namespace fsktm.networking
{
    /// <summary>
    /// Summary description for logging.
    /// </summary>
    public class logging
    {
        public logging()
        {
            //
            // TODO: Add constructor logic here
            //
        }

        public void serviceStartLogging()
        {
            string filePath =
Path.GetDirectoryName(Application.ExecutablePath) + @"logging.txt";

            FileStream fs = new FileStream(filePath,
FileMode.OpenOrCreate, FileAccess.Write);

            StreamWriter sw = new StreamWriter(fs);
            sw.BaseStream.Seek(0, SeekOrigin.End);

            string content = DateTime.Now.ToString() + " " +

```

```

"UM Remote Agent is started! \n";

        sw.WriteLine(content);

        sw.Flush();

        sw.Close();

    }

    public void serviceStopLogging()

    {

        string filePath =

Path.GetDirectoryName(Application.ExecutablePath) + @"\logging.txt";

        FileStream fs = new FileStream(filePath,

FileMode.OpenOrCreate, FileAccess.Write);

        StreamWriter sw = new StreamWriter(fs);

        sw.BaseStream.Seek(0, SeekOrigin.End);

        string content = DateTime.Now.ToString() + " " +

"UM Remote Agent is stopped! \n";

        sw.WriteLine(content);

        sw.Flush();

        sw.Close();

    }

    public void systemLogging(string message)

    {

        string filePath =

Path.GetDirectoryName(Application.ExecutablePath) + @"\logging.txt";

        FileStream fs = new FileStream(filePath,

FileMode.OpenOrCreate, FileAccess.Write);

        StreamWriter sw = new StreamWriter(fs);

        sw.BaseStream.Seek(0, SeekOrigin.End);

        string content = DateTime.Now.ToString() + " " +

```

```
message + "\n";  
  
        sw.WriteLine(content);  
  
        sw.Flush();  
  
        sw.Close();  
  
    }  
  
}
```

University of Malaya

CHAPTER 7 SYSTEM TESTING

7.1 Introduction

System Testing is required to ensure the system runs accordingly to its specifications and in line with the user's requirement and expectations. A few users are given the opportunity to try out the system so as to trace any foreseen errors or misunderstanding before the system is implemented.

Generally, there are 3 types of testing strategy that are performed in the system testing:

- Unit Testing
- Module Testing
- System Integration Testing

7.2 Unit Testing

Another name for unit testing is program testing. This type of testing normally focuses on small unit of codes for example a subroutine or functions, which are independent of one another to locate errors.

During the unit testing process, it enables the tester to detect errors in coding and logical mistakes that are contained within that module itself. The interactions between modules are initially avoided during this testing, as each is tested separately.

All function on each button is examine to ensure it perform the entitles output such as hyperlink to the right page, call the right function to execute, display the correct message according to the error and eliminates all the syntax faults occurred.

7.3 Module Testing

Module Testing is done when the individual functions in that module are integrated. This is only done once the integration takes place as to ensure that all the units in the module functions well and accordingly.

In module testing, each of the modules is tested as an independent component. In this case, black box testing approach has been used. The system can be treated as a “black box” where its characteristic is determined through a study of the collection between the inputs and outputs.

7.4 System Integration Testing

System testing takes place when modules or sub systems are integrated so that the testing is done on a created larger system. This type of testing is important to see if the end product is working perfectly once all the modules and functions are integrated.

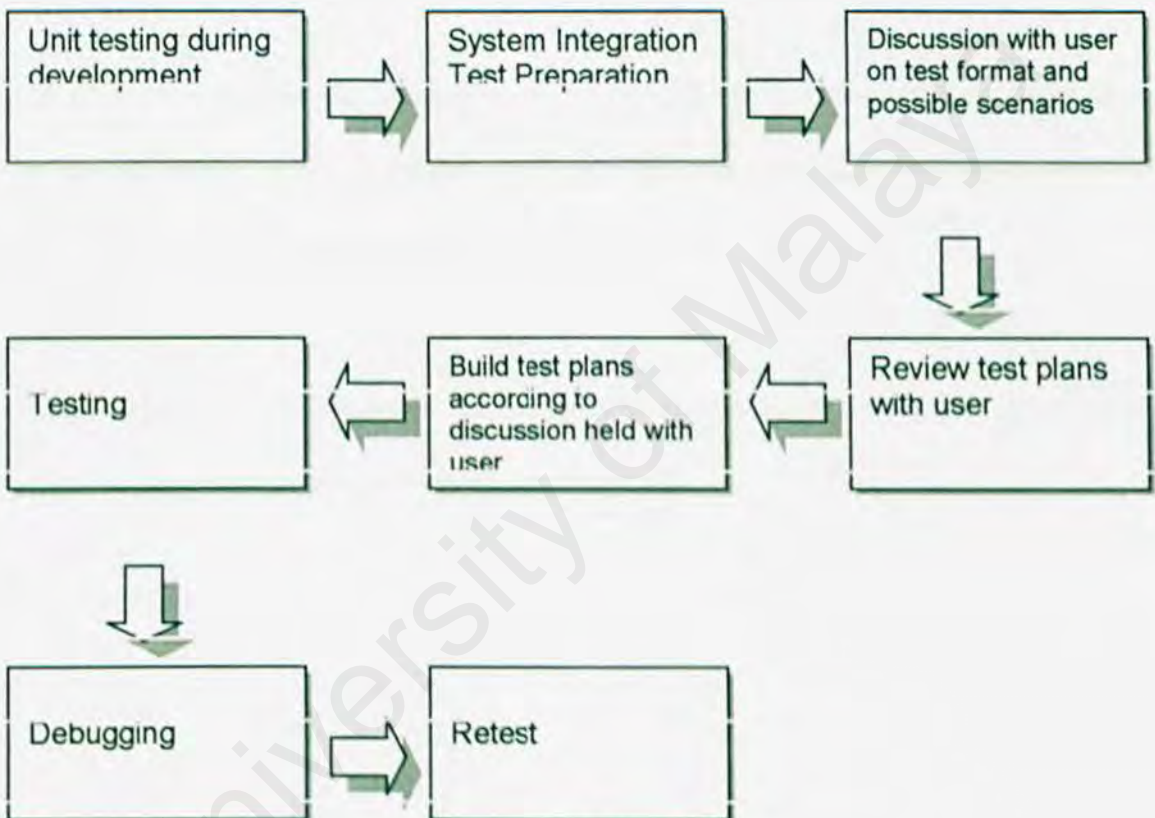


Figure 7.1: The Testing Process Flowchart

CHAPTER 8 SYSTEM EVALUATION

8.1 Introduction

During the development process, several problems were encountered in hardware, software interfaces, and logic errors in programming the required functions of the system. Most of the problems were solved at the end and some will be solved in due time.

The system's strengths and limitations were evaluated by a variety of users. Basically, improving of the present system and potential enhancements are based on the suggestions and evaluation of results from these users.

This chapter will also present the knowledge and experience gained through the process of developing and implementing UMNVC.

8.2 System Strengths

As this project is about building a VNC application, here is some strength that this program has managed to reveal:

8.2.1 Easy-to-use Application

Ease-of-use is most important aspect in this system. This system specially designed to control the target host (agent) effectively. It eliminates as many time-consuming and resource-consuming tasks as possible. On the other hand, this system provides user friendly interface, no training is needed to learn how to use the UMVNC. The network administrator will find UMVNC is a easy tool to manage the network computer.

8.2.2 System Transparency

System transparency refers to the conditions where the network administrator do not have to know the working code, how the system structure, and anything related to the system built. They are just required to know how to communicate with the user interface of the system.

8.2.3 Easy to deploy the agent application at the network

The configuration of the application is saved at the xml database file make it easy to deploy the agent application at the network computer without having to recompile the application. The environment is dynamic so that it will meet the changes of the network computer for the future.

8.2.4 Organized and User Friendly Interface

All the modules in UMVNC have its own and standardized user interface, no command line is needed and it is very easy to learn the application. Standardization in user interface design is making UMVNC easier to understand and reduce the complexity of the system when user want to use the application.

University of Malaya

8.3 Problems Encountered and Solutions

Several problems have occurred during the development and implementing of UMUMVNC. Some problems have been discovered and solutions have been sought during testing and reference check on the information through Internet, C# programming reference books and the guide given by the supervisor. Encountering with these problems has been proven to be valuable experience and guideline in the future.

8.3.1 Lack of knowledge

The very fast challenge in developing VNC application such as UMVNC is to master and to get familiar with programming language. For instance, to get mastering in using visual studio .NET by using c# .NET and VB .NET as the programming language, debugging and compiling file. By referring to website msdn.microsoft.com, lots of article available there. Besides that lots of valuable information can be gained from codeproject website.

8.3.2 Difficult to track the security problem occurred

When the lab computer changed to Microsoft Windows XP Service Pack 2. All the internet application is block by default setting of the Windows Firewall. At the early time, I really don't know how to solve the problem but fortunately with the guide given by Mr. Ling Teck Chaw I knew that how to enable the application to the listening mode.

8.3.3 Difficulties in choosing a development platform, programming language and tools for UMVNC applications

There are several platform and technologies available that we can use to develop a UMVNC application such as Java and C# .NET. Besides the platform, choosing a suitable programming language and tools was a critical process as all tools and the programming language have their strengths and weakness. In addition, the availability of the required tools for development is also a major consideration.

Getting information from internet helps in making the decision of choosing platform and development tools. By choosing C# .NET as the platform have some advantages. C# .NET is a powerful programming language but equip with easy to use user interface.

8.4 System Limitations

There are few system limitation which are highlighted below:

8.4.1 Authentication

The authentication should be integrated into UMVNC. So that only the authorize user can use the application. Without the authentication, it will post a security risk to the network.

8.4.2 Lack of global clock system

The global clock system should be implemented to give the system clock of the network computer synchronized so that the system logging will record the activities performed correctly.

8.4.3 Lack of cancel action module

When the network administrator issue a remote shutdown, restart, log off command the UMVNC should have the ability to cancel the action. So that it will make the network computer stay intact.

8.5 Future Enhancements

The current UMVNC can be upgraded so that new and sophisticated functions and features can be added on. Generally, there are a few enhancements that can be done to this system in future and they are as below.

8.5.1 Authentication

With the authentication, the UMVNC will be more secure. So that an authorize use can use the UMVNC application.

8.5.2 Global clock system

With the global clock system, the system logging will record all the action performed more precisely and perfectly. It is more easy when the network administrator to track the record.

8.5.3 Cancel action module

With this cancel action module, the network administrator can roll back the action performed so that the network computer won't be shutdown, restart, log off if the network administrator press action wrongly.

8.6 Conclusion

After implementation, UMVNC should be evaluated. Problems that encountered were analyzed and the appropriate solutions were taken carefully. Overall, UMVNC strengths fulfill the functional and non-functional requirements as planned at the start of this project. Although there some constraints or limitation in UMVNC, but there can be the future enhancement. Knowledge and experience gained through this project also being evaluated.

University of Malaya

REFERENCE

- [1] Introduction of VNC
<http://www.realvnc.com>
- [2] MSDN
<http://msdn.microsoft.com/>
- [3] Introduction of agent
<http://www.aaai.org/AITopics/html/agents.html>
<http://www.msci.memphis.edu/~franklin/AgentProg.html>
- [4] Introduction of .Net Framework
http://www.internetworld.com/dotnet.php?inc=in_focus/031501/03.15.01dotnetframework8.htm
- [5] Overview of MySQL
http://www.idevelopment.info/html/MYSQL_overview.shtml
- [6] Introduction of System Design
<http://cfg.cit.cornell.edu/cfg/design/concepts.html>
- [7] Introduction of User Interface Design
http://www.chorussw.com/products-gui_dc.htm
- [8] Introduction of XML
<http://www2.softwareag.com/Corporate/Company/xmlycompany/7goodreasons.asp>
- [9] Redhat Linux Official Homepage
<http://www.redhat.com>
- [10] Microsoft Official Homepage
<http://www.microsoft.com>
- [11] Pressman, Roger S. (2001) *Software Engineering: a practitioner's approach* – 5th edition. McGraw-Hill.
- [12] Sommerville. , I. *Software Engineering . , 5th Ed.* Addison Wesley Publishers Ltd, 1998.
- [13] Kendall, K. , Kendall, J. *System Analysis and Design, 4th Ed.* Prentice Hall, Inc. , 1999.
- [14] Microsoft .NET Homepage
<http://www.microsoft.com/net>

- [15] The Code Project
<http://www.codeproject.com/>
- [16] The forum for .NET developers
<http://www.dotnetforums.net>
- [17] Mandel, T. (1997) *The Elements of User Interface Design*, Wiley.
- [18] Myers, Thomas A (1999). *Professional Java XML Programming with Servlets and JSP*. Birmingham, UK: Wox Press. 2-33
- [19] KDE desktop
<http://www.kde.org>
- [20] Network Programming Concept
<http://www.c-sharpcorner.com/Networking.asp>