Title: **IMPLEMENTING A WEB- BASED SINGLE-SIGN-ON**

By

NADHIRA YASMIN ZULKAPLI

(2003323669)

A project paper submitted to

FACULTY OF INFORMATION TECHNOLOGY AND QUANTITATIVE SCIENCE,

UNIVERSITI TEKNOLOGI MARA

In partial fulfillment of requirement for the

BACHELOR OF SCIENCE (Hons) IN DATA COMMUNICATION ADND NETWORKING

Major Area: NETWORK SECURITY

Approved by Examining Committee:

…….

(Pn. Norkhushaini Binti Awang) : Project Supervisor

……..

(Pn. Rozita Binti Yunus) : Examiner

UNIVERSITI TEKNOLOGI MARA

SHAH ALAM, SELANGOR
MAY 2006

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project that the original work is my own, except as specified in the references and acknowledgement and that the original work contained here in have not been taken or done by unspecified source or person.

……
(NADHIRA YASMIN ZULKAPLI)

ACKNOWLEDGEMENT

Alhamdulillah, in the name of ALLAH, The Al-Mighty, The Most Gracious and The Most Merciful Peace and Blessing of ALLAH. The Al-Mighty upon our beloved Prophet (Peace Upon Him), his entire relatives and all his companions and all those had followed.

Special thank to my supervisor, Puan Norkhushaini Awang, for her wonderful support, guidance and cooperation that had been given to me throughout the compilation of this project. To Puan Rozita Yunus who had been guiding the writing of this report. And also to all lecturers and Information Technology units of banks and government offices who were very helpful in providing me with valuable information and support.

I would like to extend a special thanks to my classmates on their support in sharing information and, precious advice and support during completing this project. The knowledge that has been shared is valuable for my future practice or usage especially in the working area.

Finally yet importantly, I would like to thank my family for their special support to me. Not forgetting all the people who had involve directly or indirectly towards the success of this project. Thank you again.

TABLE OF CONTENTS

	Page
DECLARATION	ii
ACKNOWLEDGEMENT	iii
TABLE OF CONTENTS	iv
LIST OF TABLES	vi
LIST OF FIGURES	vii
LIST OF ABBREVIATIONS	viii
ABSTRACT	ix
CHAPTER ONE: INTRODUCTION	
1.1 INTRODUCTION	1
1.2 PROBLEMS DESCRIPTION	3
1.3 OBJECTIVE OF THE RESEARCH	4
1.4 SCOPE AND LIMITATION	5
1.5 SIGNIFICANT OF RESEARCH	5
1.6 THESIS ORGANIZATION	6
1.7 CONCLUSION	7
CHAPTER TWO:	
2.1 INTRODUCTION	8
2.2 SIGN ON AND SINGLE SIGN ON	9
2.3 SINGLE SIGN ON OVERVIEW	10
2.3.1 WHAT IS SINGLE SIGN ON	10
2.3.2 SSO DESCRIPTION	12
2.3.3 HOW DOES SSO WORK	13
2.3.4 SECURE LOGIN WITHIN SSO	21
2.4 ENTERPRISE SSO AND WEB-BASED SSO	23
2.5 RELATED STUDIES	24

CHAPTER III: METHODOLOGY

3.1 INTRODUCTION 28
3.2 INFORMATION GATHERING 29
3.2.1 INTERNET 29
3.2.2 LIBRARY 29
3.2.3 INTERVIEW 30
3.2.4 OBSERVATION 30
3.3 DATA ANALYSIS 31
3.4 THE DEVELOPMENT OF SSO APPLICATION 31
3.5 TESTING PHASE FOR SSO APPLICATION 32
3.5.1 RESPONDENTS INVOLVEMENT 32
3.5.2 RESPONDENTS 33
3.5.3 TESTING 33
3.5.4 RECORDS 34
3.6 HARDWARE AND SOFTWARE REQUIREMENT 35
3.7 CONCLUSION 35

CHAPTER IV: FINDINGS AND RESULTS

4.1 INTRODUCTION 37
4.2 INTERVIEWS AND OBSERVATION 38
4.2.1 CONCLUSION 41
4.3 SSO EMAIL APPLICATION TESTING 42
4.4 CONCLUSION 44

CHAPTER V: CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION 45
5.2 RECOMMENDATION 47

LIST OF REFERENCE 49
APPENDIX 52-80

LIST OF TABLES

Table Ap 1: Summary of SSO Products 52
Table Ap 2: List of Group A members 56
Table Ap 3: List of Group B members 57

LIST OF FIGURE

Figure 2.1: Basic Transaction in the Kerberos Protocol 15

Figure 2.2: Cross Realm Referrals 16

Figure 2.3: Novell Authentication 18

Figure 2.4: Sign On to Multiple Systems 19

Figure 4.1: The respondents' age range 42

Figure 4.2: Total time taken by both groups 43

Figure Ap 1: An HTML page showing GSS applet usage 59

Figure Ap 2: SSO Main Login Page 59

Figure Ap 3: SSO Main Login Page (with characters) 60

Figure Ap 4: Welcome Page 60

Figure Ap 5: Main Email Account Page 61

Figure Ap 6: Email Account (account 01) 61

Figure Ap 7: Email Account Page (account 02) 61

Figure Ap 8: SSO Main Login Page (wrong ID or password) 61

LIST OF ABBREVIATIONS

SSO	Single Sign On
IT	Information Technology
API	Application Programming Interface
KDC	Key Distribution Center
TGT	Ticket Granting Ticket
ST	Service Ticket
TCO	Total Cost of Ownership
ID	Identification
userID	User Identification

ABSTRACT

Single Sign-On (SSO) is a technique that attempts to solve the "identity crisis" of this information age. Nowadays, most computer users have multiple user names and passwords for different domains, applications, and web sites. This is difficult for administrators to manage and challenging for the users to memorize. SSO in the simplest form is providing the user with a master password that has access to a database of different sign-on credentials, and to associate different, scattered accounts into a master account that is responsible for all the authentication and authorization processes. SSO systems have advanced from the initial stage of hiding a complex, multiple sign-on environment behind a single account on an authentication server, to more sophisticated implementations that involve polices, rules, and roles that determine a user's identity and level of rights.

CHAPTER I

INTRODUCTION

1.1 INTRODUCTION

Nowadays, technology is changing rapidly whether it is in hardware or software. Companies or individuals depends on internet to gain information and communicates with others. Besides that, they also use network connections in order to get resources from departments in the companies. Therefore, we are facing many problems. Technologies and people are dependable to each other. It is difficult to us to prevent them or one of them from developing or building new technology, when the new technology could actually give us more effective and convenient workload but at the same time will also give some negative effects.

In many organizations, users struggle with having to sign on multiple times to access different applications, Web portals, and servers. As the number of mandatory unique sign-on grows the burden on users to remember numerous usernames and passwords increases. The result is often unhappy users who innocently create serious security breaches by writing down usernames and passwords because they cannot remember them all. Frustration might also extend outside an organization. Business partners, field representatives, and customers might need to access multiple Web portals or applications from outside the

organization (typically over the Internet), and they might be subject to multiple sign-on requests. With the increasing use of distributed systems, users often need to access multiple resources to finish a single business transaction. Traditionally, users have had to sign on to all these systems, each of which may involve different usernames and authentication requirements.

The security of computer systems is a controversial issue from a practical viewpoint. The users expect the systems to help them in their daily work, which leads to an emphasis on ease-of-use. After all, if a system is difficult to use and slow in performing the tasks assigned to it, there is little incentive to use the system at all. In traditional PC systems, security has been of little or no concern because of the assumption that the systems are isolated from each other. With the very rapid growth of the Internet, PCs are no longer isolated. The Internet is very useful, and users are in a hurry to be connected. Once connected, large varieties of easy-to-use services are at the disposal of the users. On the other hand, so it would seem at first glance.

The Internet is indeed easy to use, but only once the user has registered him to obtain the necessary access privileges. The reasons for requiring registration vary, from the legitimate need to try to prevent misuse of a service to the more sinister attempts to silently and exactly create detailed profiles of users. No matter what the reasons are, the result is a multitude of authentication requests that face every user of the Internet today.

Different types of authentication methods are in use nowadays. The methods, their advantages and their shortcomings will be reviewed. We look at different compound solutions that attempt to provide a single sign-on (SSO) environment for the users.

A previously research done by Camillo Särs from the Department of Computer Sciences, Helsinki University of Technology in November 1998, was an attempt to do a systematic analysis of the different authentication methods. (Särs, 1998).

1.2 PROBLEMS DESCRIPTION

Intranet users are commonly required to use a separate password to authenticate themselves to servers they need to access in the course of their work. Multiple passwords are an ongoing headache for both users and system administrators. Users have difficulty keeping track of different passwords, tend to choose poor ones, and tend to write them down in obvious places. Administrators must keep track of a separate password database on each server and deal with potential security problems related to the fact that passwords are sent over the network routinely and frequently.

According to Protocom, password management can be a complex and frustrating task for end users, resulting in compromised security and lost productivity.

As IT, systems proliferate to support business processes, users and system administrators are faced with an increasingly complicated interface to accomplish their job functions. Users typically have to sign-on to multiple systems, necessitating an equivalent number of sign-on dialogues, each of which may involve different usernames and authentication information. (X/open, 1996).

1.3 OBJECTIVE OF THE RESEARCH

The objectives of the project are:

i. To study the concept of single sign on

ii. To develop a prototype model of an email single sign on application

iii. To define an API (application programming interface) for a single sign on

1.4 SCOPE AND LIMITATION

This research is focus on implementing a single sign on solution. We will also show different solutions used by other developer for their software. The scope of testing was done in a cyber café. The subject of study was the customers of the café. Interviews with users of certain organizations were also done to know the limitations of their own single sign on devices.

1.5 SIGNIFICANT OF RESEARCH

Password management is the most important thing in securing computer from unauthorized user access. It is difficult to manage a different password for every user. Therefore, by introducing single sign on solution into an organization, it may give more benefits. The benefits from this research are:

i. Give a simple administration to management team in managing and maintaining administrative task.

ii. Give a better administrative control, which is all-specific information is stored in a single repository.

iii. Improve user productivity by entering single password and user does not to remember multiple passwords.

iv. Single sign on provide secure authentication and provide a basis for encrypting the user's session with the network resources.

1.6 THESIS ORGANIZATION

The report contains five chapters in order to explain every phase that have been done though out this research. Below is the general view of each chapter:

CHAPTER 1: INTRODUCTION

This chapter explains about the introduction of the research, description of the problem, objectives of the research, research scope and significant of the project. This chapter also gives the short explanation about the methodology of this research and limitation and overview.

CHAPTER 2: LITERATURE REVIEW

This chapter contains the definition of pertinent terminologies that were used in this report. It also reviews the previous research related with this research.

CHAPTER 3: METHODOLOGY

This chapter explains the methodologies that were used during the period of this research being compiled and done in order to complete this research.

CHAPTER 4: FINDING

This chapter will show the detail numerous results and findings that we have panned in the previous chapters. This chapter also will look into several result and findings that we collected after the interview and analysis that has been done previously.

CHAPTER 5: RECOMMENDATION AND CONCLUSION

This chapter concludes the chapter and recommends the solution in order to solve the problem that may exist.

1.7 CONCLUSION

Single sign on is here to help in reducing the password management needs of the user in allowing them to access in multiple application in just remembering a password. We will also try to define the different approach used by other companies in developing a solution.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

Chapter 2 consists of reviews of other reports that are related and helpful in conducting our report. The review describes summaries, evaluation and clarification of this literature and matters arise. The information was gathered from websites, books, journals, articles and previous researches. The information can be helpful in compiling a good report. It also helps us in giving a guideline to our project. Users typically have to sign-on to multiple systems, necessitating an equivalent number of sign-on dialogues, each of which may involve different usernames and authentication information. System administrators are faced with managing user accounts within each of the multiple systems to be accessed in a coordinated manner in order to maintain the integrity of security policy enforcement. All the reviews of literature in this chapter are regarding the scope of the research project. It has been divided into several main areas related to this research.

2.2 SIGN ON AND SINGLE SIGN ON

There are a few definitions to the term of sign on. According to Google Define, sign on is the procedure by which the user starts working at a workstation. Besides that, sign on can also be define as to begin a working session

On the other hand, single sign on will give us the definition of letting a user log on once to a PC or network and access multiple applications and systems using a single password. Typically, single sign-on products (such as Computer Associates' eTrust Single Sign-On) authenticate the user at logon and present the available applications on the desktop. When the user selects an application, the SSO agent presents the authentication credentials in the background and they have access to that application without having to log-on separately. According to E-Government.govt.nz, it means the act of signing on once (providing a UserID and Password) thereby achieving access to multiple systems or e-services without having to re-establish the identity of the person. The definition given by Wikipedia.org states that Single sign-on (SSO) is a specialized form of software authentication that enables a user to authenticate once and gain access to the resources of multiple software systems.

2.3 SINGLE SIGN ON (SSO) OVERVIEW

2.3.1 What is Single Sign On?

Single sign on (SSO) is describes in several ways but the meaning is still the same. The Health Insurance Portability and Accountability Act of 1996 (HIPAA) describes that Single Sign On is a simple way to tie together proper user authentication and application access and enable proper privacy controls. One ID and password authenticate the user for all required application, such as prescription orders and patient records. Single sign on eliminates the need for health practitioners to remember multiple passwords, while retaining a high level of security for each application. A doctor can access patient's records, prescription information, and other medical data using a one-time authentication.

The single sign on authentication process in a client/server relationship where the user can enter one name and one password to log on to the applications. User will also have access to more than one application or access to a number of resources within an enterprise. Single sign on (will be later refer to as SSO) takes away the need for the user to enter further authentications when switching from one application to another. (webopedia.com, 2002), and quoting from Lubow, authentication is a complex problem (Lubow, October 2003).

In e-commerce, SSO is designed to centralize consumer financial information on one server - not only for the consumer's convenience, but also to offer increased security by limiting the number of times the consumer enters credit card numbers or other sensitive information used in billing. Microsoft's "Passport" SSO service (averaging over 40 million consumers and more than 400 authentications per second) is an example of a growing trend towards the use of Web-based SSO that allow users to register financial information once, shop at multiple Web sites, and feel more confident about security on the Web. (Waynforth, 2003)

Single Sign-On (SSO) systems enable users to authenticate a single time when establishing a desktop session, after which the SSO system handles any further authentication behind the scenes. (Särs, 1998)

While from mtechit.com, a SSO system is a set of software components, usually distributed over a network, which allow a user to log into his workstation once, and thereafter start applications and network login sessions without any further authentication. The initial login may be carried out using credentials, such as a user ID and password, or another technology, such as a *Public Key Infrastructure* or a *Smart Card*. (http://mtechit.com/resource,)

The ability for users to log on once to a network and be able to access all authorized resources within the enterprise. A single sign-on program accepts the user's name and password and automatically logs on to all appropriate servers. Single sign-on services such as Microsoft's Passport are increasingly being used for Web sites. (techweb.com, 2003)

From the website of iamsect (iamsect.ncl.ac.uk/term), Authentication is the process of determining a user's identity, usually by verifying a supplied username and password combination. SSO systems provide a means where authentication information can be shared between services, preventing a user from having to authenticate them multiple times. (iamsect.ncl.ac.uk, 2002)

2.3.2 Single Sign On Descriptions

White paper Window 2000 SSO explains that Single Sign On (SSO) is an optical user authentication security feature available. Authentication is a process in which a system attempts to validate a user-supplied name and password with an entry in a user account database on behalf of a secure application; SSO software increases an organization's level of security and decreases user account maintenance and administration in a heterogeneous intranet. Many intranet environments contain heterogeneous systems that offer secure applications and

independent user account databases. Typically, heterogeneous systems do not share user account information. As a result, a user must have a user account on each system from which to access secure applications.

Datamonitor stated that single sign on increases user productivity by minimizing work disruption when logging on to a new application or platform. It also frees time, which would otherwise be spent on helpdesk calls or on trying to remember forgotten passwords.

Evidian highlights that SSO relieves Window users from the necessity to handle dozens of passwords a day, and enable them to connect seamlessly from their desktop to all the applications they need, whether local, mainframe, client-server or web.

2.3.3 How does single sign on works?

The basic of single sign on can be implemented in some environments. At this moment, many companies in security are trying to develop hardware and software based on the concept of single sign on. Single sign on that was provided in Windows 2000 using Kerberos authentication protocol, which is the default authentication protocol in Windows 2000. The use of the Kerberos protocol provides significant improvements in administrative ease, security, and network performance.

Microsoft Windows NT distributed Security Services (1998) explains that the Kerberos protocol is based upon the idea of tickets where its encrypted data packets issued by a trusted authority called Key Distribute Center (KDC). A ticket vouches for a user's identity, as well as carrying other information. A KDC provides tickets for all of the users in its area of authority, or realm. In Windows 2000, every domain controller is a KDC, and the realm of a domain controller corresponds to its domain. Figure 2.1 explains detailed transaction in the Kerberos protocol.

The operation of the protocol is simple, as shown in Figure 2.1. At logon time, the user authenticates to a KDC, which provides it with an initial ticket called Ticket Granting Ticket (TGT). When the user needs to use a network resource, his user session presents the TGT to the domain controller and requests a ticket for the particular resource, called Service Ticket (ST). He represents the ST to the resource, which grants his access.

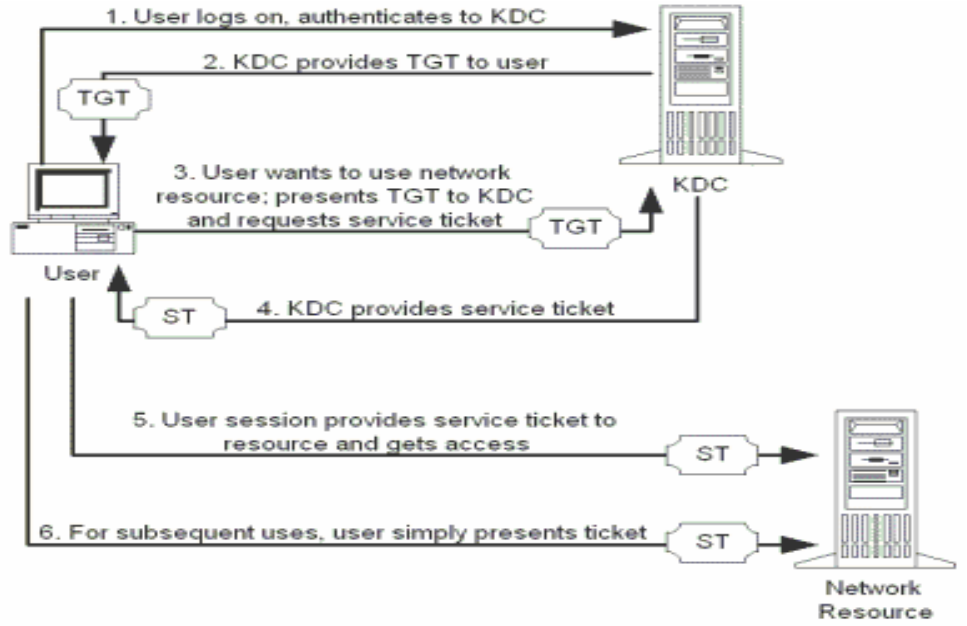

Figure 2.1: Basic Transactions in the Kerberos Protocol (Adopted from Microsoft Windows 2000 Server)

Figure 2.1 shows the basic transaction in the Kerberos Protocol. The figure was adopted from Microsoft webpage. It was adopted from the architecture of Microsoft Window 2000 Server.

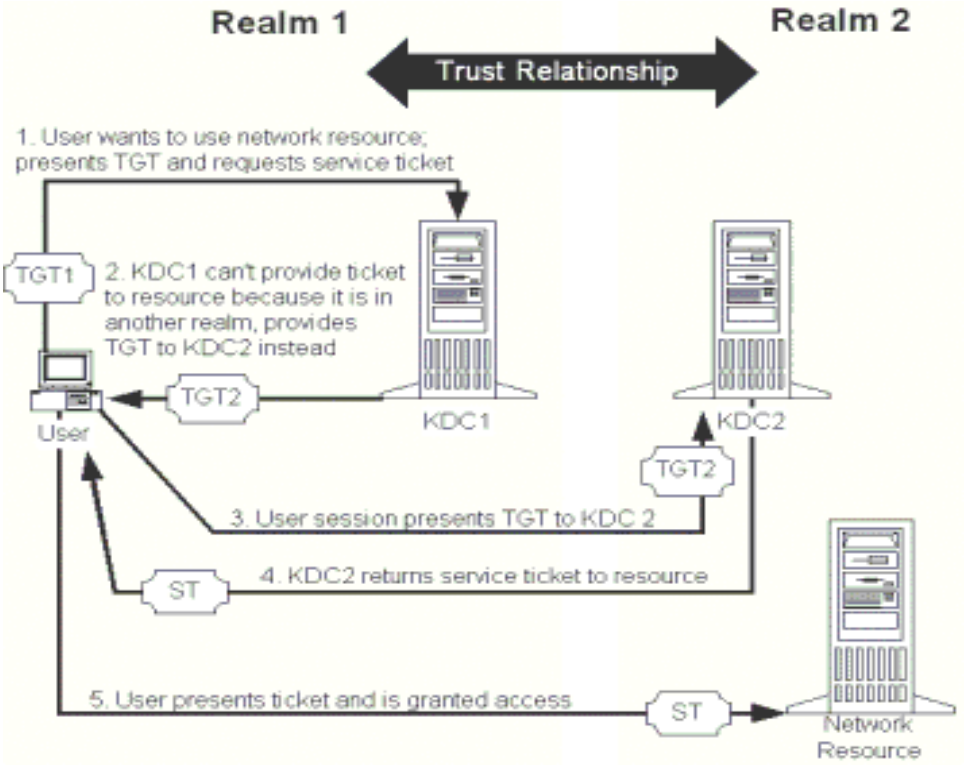

Figure 2.2: Cross-Realm Referrals (Adopted from Novell Inc)

Otherwise, Novell.Inc also provides single sign on to be implemented in their system to authenticate the login. The following outlines the authentication processes for Novell SecureLogin:

- **Step 1** – User logs in to workstation, entering password or other authentication method. (i.e., token, smart card or biometric).
- **Step 2 –** Authentication information is sent from the workstation to the directory. (Supports Novell

eDirectory™ and other LDAP v3 directories, Active Directory*, and NT Domains).

- **Step 3** – A successful login confirms user's identity in the directory.
- **Step 4** – User launches an application and the application requests user authentication.
- **Step 5** – Novell SecureLogin automatically detects the authentication request, checks the policy for that application, and sends the user's credentials for that application.
- **Step 6** – Novell SecureLogin presents the application with the users' credentials specific to that application. The application then authenticates the user and enforces its access control policy on the user.

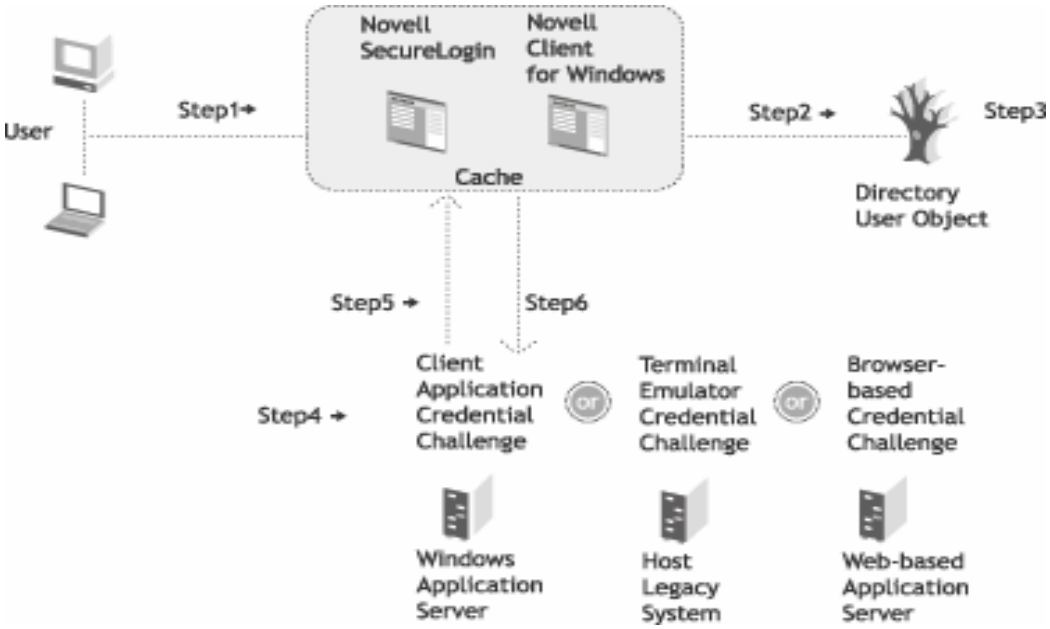

Figure 2.4: Novell authentication (Adopted from Novell Inc., 2004)

Novell SecureLogin provides the most comprehensive array of features, benefits and capabilities for delivering secure, flexible, and easy-to-use single sign on. Figure 2.4 shows the architecture of Novell authentication, which was adopted from Novell Incorporated.

On the other hand, Open Group also describes the approach of single sign on as illustrated below in Figure 2.3:

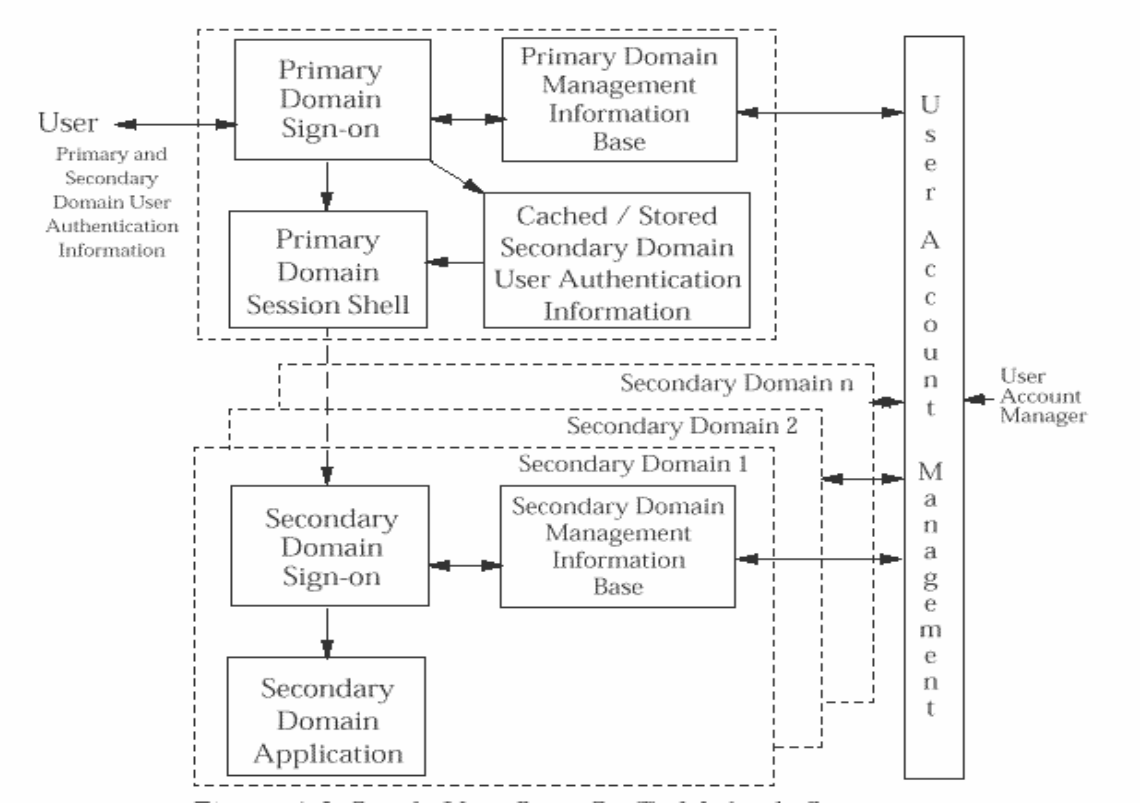

Figure 2.3: Sign on to Multiple Systems (Adopted from Open Group, 1998)

Historically a distributed system has been assembled from components that act as independent security domains. These components comprise individual platforms with associated operating system and applications. These components act as independent domains in the sense that an end-user has to identify and authenticate himself independently to each of the domains with which he wishes to interact. This scenario is illustrated above. The end user interacts initially with Primary Domain to establish a session with that primary domain. This is termed the Primary Domain Sign-On in the above diagram and requires the end user to

supply a set of user credentials applicable to the primary domain, for example a username and password. The primary domain session is typically represented by an operating system session shell executed on the end user's workstation within an environment representative of the end user (e.g., process attributes, environment variables and home directory). From this primary domain session shell, the user is able to invoke the services of the other domains, such as platforms or applications. To invoke the services of a secondary domain, an end user is required to perform a Secondary Domain Single-On. This requires the end user to supply further set of user credentials applicable to that secondary domain. An end user has to conduct a separate sign-on dialogue with each secondary domain that the end user requires to use. The secondary domain session is typically represented by an operating system shell or an application shell, again within an environment representative of the end user. From the management perspective, the legacy approach requires independent management of each domain and the use of multiple user account management interfaces. Considerations of both usability and security give rise to a need of to coordinate and where possible integrate user sign-on functions and user account management functions for the multitude of different domains now found within an enterprise.

2.3.4 Secure Login within Single Sign On

Passlogic describe many companies are evaluating or implementing strong authentication technologies, such as smart cards or biometrics. When deployed with SSO, those technologies secure access to all network applications and resources, including logon to Windows. Nevertheless, when the authenticators are not available, for example because the user misplaced the smart card, the fallback is usually the Windows password for log on. As users probably cannot remember, that password enables them to pick a new Windows password and be on their way in less than a minute. With single sign on companies, pave the way for a seamless and affiancing deployment of strong authentication.

Novell® Nsure™ act as one of the component that secure identity management solution family, Novell SecureLogic significantly reduces password security risks, providing users secure single sign-on to all of the business systems that they are authorized to access. It does away with the need for users to remember multiple passwords, safeguards authentication credentials, and enables you to implement strong passwords policies while reducing over 90% of password security vulnerabilities, Additionally, the single sign-on functionality inherent to Novel SecureLogic enables organizations to realize significant return on investment by reducing help desk costs and improving user productivity. Novell SecureLogin facilitates and automates the enforcement of

password strengthening policies. It protects your confidential data from the prying eyes of rogue administrators. It has the flexibility to allow you to require more stringent authentication methods for highly sensitive business systems. Novell SecureLogin simplifies and automates you password management so you can safeguard your enterprise resources from malicious intruders.

Extended Sign on Kerberos in Windows includes APIs that allow a front-end server to obtain a Kerberos session ticket on behalf of a user, then to submit that session ticket to a back-end server when requesting access to a database or other information repository. Using this mechanism, an application developer does not need to embed authentication credentials into the front- end where they could be compromised by an attacker, thus making the entire systems more secure. This is an arcane branch of security, and one that developers are only just now starting to implement, but it holds lots of promise for distributed authentication in front-end/back –end database applications.

2.4 ENTERPRISE SSO AND WEB-BASED SSO

In any client/server relationship, single sign on (pronounced SING-uhl SAIN-awn) is a session/user authentication process that permits a user to enter one name and password in order to access multiple applications. The single sign on, which is requested at the initiation of the session, authenticates the user to access all the applications they have been given the rights to on the server, and eliminates future authentication prompts when the user switches applications during that particular session.

According to Powertech, **Enterprise single sign on** (E-SSO), also called legacy single sign on, after primary user authentication, intercepts login prompts presented by secondary applications and automatically fills in fields such as login ID or password. E-SSO systems allow for interoperability with applications that are unable to externalize user authentication, essentially through “screen scraping”,

Adding to it, Powertech also adds that **Web Single sign on** (Web SSO), works strictly with applications and resources accessed with a web browser. Access to web resources is intercepted, either using a web proxy server or by installing a component on each targeted web server. Cookies are most often used to track user authentication state, and the Web SSO infrastructure extracts user identification information from these cookies, passing it into each web resource.

2.5 RELATED STUDIES

Single sign on has been developed by many organizations especially from corporate organization. Some of them that were developed are beneficial for us in completed our research.

The BNX Identity Management Suite (2000) manages application sign-on and strong user authentication in one integrated enterprise solution. This allows organizations to increase security, users' convenience and operational efficiency and quickly realize a significant Return on Investment. Unlike any other solution available today, BNX support for Windows, Web and terminal-emulated environments, Policy-based, centrally managed authentication and single sign-on (SSO) with consolidated administration.

V-GO SSO is first Universal Single Sign-On solution, which works with all applications application, without a lengthy and complex implementation and development process. Whether we are deploying strong authentication, implementing an enterprise-wide identity management initiative or simply focusing on the sign-on challenges of a specific group of users, v-GO SSO's constraints. V-GO SSO's patented client-side intelligence, combined with superior directory integration and simple, yet powerful, administrative consol, delivers the benefits of faster, more secure sign-on in days- not months.

Enterprise users can enjoy sub-second single sign-on while connected or disconnected to the corporate network, while roaming between computers, or while sharing kiosk with multiple users. V-GO SSO supports any type of user authentication from smart cards to biometrics and integrates with any LDAP directory or Microsoft® Active Directory®, ADAM and SQL server. This allows an organization to apply the security it needs, as we require it, while relying on the infrastructure we already trust. Our users can log-on to our network and easily and securely access all their applications from desktops, laptops and kiosks.

Encentuate TCI is an enterprise identity management platform that provides fortified SSO easily and rapidly. To provide true single sign-on, it takes an unconventional approach to solving the problems instead of conventional centralized authentication with server-side integration.

Encentuate TCI uses an innovative, agent-driven, server-managed architecture. The Encentuate AccessAgent auto-learns all user passwords. It signs the user on to the application changes. In addition, Encentuate TCI fortifies the individual application passwords to avoid any keys-to- the-kingdom problems. Support for hardware authentication further enhances security.

Novell® SecureLogin is comprised of multiple integrated security systems that provide authentication and single sign-on to networks and

applications throughout an organization. The goal is to provide a single entry point to the corporate network and its resources for the users, increase security, and improve compliance with corporate security policies.

Microsoft provides an integrated, comprehensive, easy- to-use SSO capability as part of Microsoft® Windows ® 2000 operating systems, thus allowing dramatic reduction in the total cost of ownership (TCO) for a computing enterprise by improving user and administration productivity while improving security. The greatest benefits are derived from implementing benefits even when deploying Windows in heterogeneous networks. Furthermore, because SSO for Windows 2000 interoperates with so many other vendors' operating systems, it is the best choice to serve as an SSO hub in heterogeneous networks. Microsoft using Kerberos protocols in implementing the usage of single sign on (SSO) on their systems. Kerberos used ticket granting Ticket (TGT) to authenticate users. Beside that, they also used trust relationship in getting trusted from server.

AccessMaster offers administration functions ranging from administration of user rights up to administration of keys and certificates, along with robust authentication and applications access authorization functions. It is designed to meet the Enterprise's security needs. By incorporating into its administration tool all the function of a PKI, AccessMaster offers a transition between management of security policies based on passwords and those based on

certificates. In just a few clicks from their AccessMaster administration console, the administrator groups manage the creation, renewal and revocation of user access rights and use of their certificates.

CHAPTER III

METHODOLOGY

3.1 INTRODUCTION

This chapter explains about the method and process used for research to collect and gather sufficient information related to this thesis. Different methodologies approach was used in order to gather related information. Each methodology has its own strength and weaknesses depending on what type of information that we want to collect. Different methodology used, has its own different kind of methods in collecting data.

In this project, we want to develop an application that represents Single Sign On solution. Certain steps were taken into consideration before we can list down the methodology we want to use.

After several discussions, we have decided to use these methodologies. All information about the hardware and software requirements and process involved will be described further in the chapter. The methodologies are information gathering, data analysis, implementation phase, and testing phase. Figure 3.1 shows the general overview of methodology techniques that were used in this stage.

3.2 INFORMATION GATHERING

After we have confirmed with our project title, we begin to collect data that is relevant to this project.

3.2.1 Internet

The internet is a major source of information for this project. It provides us with the information we need. From the internet, we found research and project done previously by different researchers in this topic. Many universities and organization have developed their own single sign on solution, even though in different language and context. Several of them are focus on using smartcards as alternative to passwords. There were also research on different kind of authentication such as biometric, iris recognition and voice recognition.

3.2.2 Library

The data collection from the library focus on books, journals, and previous thesis projects relevant to our title and the software we want to use. The software we used is Macromedia Dreamweaver version eight. This software allows our application to have an easy to use interface and will be an ease to the users too.

3.2.3 Interview

After several discussions, we have agreed to choose interview as method to collect data from the other companies and government agencies that have been using single sign on application in the office. The interviewees were asked several questions related to their experience of time before the implementation of single sign on and after. Some companies were also approached as to know whether they are familiar with this solution and to know whether they are willing to adopt this solution to their workstations.

3.2.4 Observation

Observation will also be conducted in order to gain information. Observation will be done in agencies that have applied the application to their servers or workstations.

3.3 DATA ANALYSIS

After all the information needed to develop our application is sufficient, we will go through to the data analysis method. Data should be analyze and study such as:

- The method of creating the application
- The human memory limitation
- Type of testing should be conducted to the respondents
- Codes for the single sign on application development

3.4 THE DEVELOPMENT OF SINGLE SIGN ON APPLICATION

Before we start the development process, we should learn to use the Macromedia Dreamweaver software. We need to find out the users' need and expectation in order to design the application. The application must be completed before we can start the testing phase. However, we will limit our tasks until the email section. The application will still show the method of single signing on to an email server with multiple userIDs and passwords.

During the log on, the user will required to remember only one master userID and one password. The user will then enter this main email page with the master userID and at the same time he had successfully enter to the emails accounts that is registered under the master userID.

3.5 TESTING PHASE FOR SINGLE SIGN ON APPLICATION

The testing phase for single sign on application starts after the development of the application is completed. Based on the previous works done by researchers in this topic, we have agreed to test the application with individuals who owns multiple email accounts and private agencies that have not yet apply this solution.

3.5.1 Respondents Involvement

The respondents participate in the testing are individuals from the Fastlink Cyber Café situated in Taman Sri Serdang. We included both men and women in this test. Because of the time constraints, most of the respondents involved in this test are secondary school students and Universiti Putra Malaysia's students. Their age range is from 16 to 25 year olds.

3.5.2 Respondents

There are 10 respondents involved in the single sign on application testing. The respondents were divided into two groups. Each group consists of ten respondents with five men and five women each. Before we started the testing, all of the respondents were required to write down their two email userIDs and passwords accounts on a sheet of paper provided in front of them.

The two groups were name as Group A while the other one is Group B. Group A will log on to their email accounts, one by one. As for Group B, they will be given a master userID to try to log on to their accounts.

3.5.3 Testing

There two stages in this testing. The same methods of testing are applied onto both groups. In the first stage, they are required to memorize the passwords (Group A) and Group B will memorize their new master userID and password. We gave each respondent two minutes to memorize the given orders.

In the second stage, we asked them to log on to their emails accounts. However, the twist of this stage is Group A will need to log on to each accounts, one by one. As for group B, they try to log on using the

master userID and the password. One respondent from each group is tested simultaneously. This is done to record the time they used to log on their emails.

After that, each of them was asked on their perception of the single sign on application.

3.5.4 Records

Respondents were informed that their email addresses and passwords were only for the record of this project. We need to keep a written record of the participant's email accounts for this education purposes.

3.6 HARDWARE AND SOFTWARE REQUIREMENT

In this thesis, we only used one personal computer. The computer is used to develop the single sign on application. The software we used is Macromedia Dreamweaver version 8 and Microsoft SQL Server 2000 Personal Edition. This software allows the developer to create application with user interfaces. Computer application with interface is easy to use and provide comfort ability to the respondents.

Before installing the Macromedia Dreamweaver software and Microsoft SQL Server 2000 Personal Edition, there is several hardware requirements need to be fulfill to ensure better performance:

- Pentium 4 or above
- 256 Mb RAM or above
- Window 2000/NT/XP
- At least 400Mb free space

3.7 CONCLUSION

In our daily life, we follow the rules and guidelines. This was also applied in the methodology of this project. The methodology is used as a guideline

throughout the completion of this project. The methodology must be followed systematically in order to achieve the objectives.

Information gathering has been chosen as one of the methodologies in this research because a lot of information can be reviewed and gathered throughout process. The resources can be gathered from internet, library and interviews. The internet is useful whenever we need to review the white papers or previous researcher from foreign universities and agencies. By subscribing to the white paper newsletters, we will be informed on new submitted paper to the group. This shows the time can be used wisely, as we do not have to be on the internet, searching for information for a long time.

Interviews with system administrators and individual related to IT, gave us a new perspective of the network and system world of today. They also help in giving ideas on the testing stage of this application.

The quantity of respondents during the single sign on application test is very important. A large quantity of respondents can provided us with more accounts to show the limitation of multiple sign on compared to the single sign on. The demographic information such as age is also important, as they are the major user of multiple email accounts (range age of 15-30 year olds). Our single sign on application is a standalone application. Therefore, we can use only one computer as a server and client.

CHAPTER IV

FINDING AND RESULTS

4.1 INTRODUCTION

In this chapter, we will discuss on the findings and results from the observations, interviews and single sign on application testing, we had conducted before. In the first part, we will discuss the perspective of user of the single sign on and the non-users. The interview was done with respondents of Petroleum Nasional Malaysia (Petronas), Maybank Malaysia Berhad, Texas Instruments Malaysia, and Bank Pertanian Malaysia.

Later on, we will look into the single sign on application testing. We want to compare the time took by each group in log in on to their respective email accounts. The last part is the conclusion from all what we had discussed though out this chapter.

4.2 INTERVIEW AND OBSERVATION

The single sign on solution is not a new technology in this part of the world. In this section, we are going to analyze and discuss the results of the interviews and its significances to our project. The questions asked during the interview can be found in Appendix B. The questions asked during the interviews vary from each company or personnel.

Some large company has even applied to this technology since 1999. However, most of the current using companies are multinational and international companies. Besides that, in a company, the single sign on technology is not widely used by all staff or departments.

The respondents or officers participated in these interviews are from different departments. The respondents were closely chosen from the same education background of education which is degree or higher. All of them are IT literate, which means their job descriptions are related to computer and technology. All of them are 35 years old or older. The questions are divided into two different sections. On the first section, we asked about their knowledge or experience in using single sign on and the second section is about their multiple email accounts.

In our interview with an officer (Officer A) of Petroleum Malaysia (Petronas), she claimed that this technology is only applied to only certain level in a department. However, Officer A only owns one email account of *xxxx@petronas.com*. Therefore, she cannot give a sufficient answer to the question of her experience using the multiple email account experience. Other officers that were interviewed by us had given positive feedback to the use of this technology as it improves user's productivity and gives better administrative control to the users.

As for banks (Maybank and Bank Pertanian Malaysia), this technology is widely used especially by their managers. An interview with a branch manager, he suggested that this use of technology could also be applied to schools or even cyber café. Therefore, it shows that, with this technology, people can really save time and effort on log in onto their accounts. A branch manager has the responsibility to log on to several servers. He can log on to the transaction server where under that server there are current accounts, saving accounts, etc. He also has the responsibility of his branch loan server. Even though, the main server is monitored by the system administrator but each of the managers have their own privilege set by the system administrator. The managers hold email account with the addresses of *xxxxx@bpm.com* or *xxxxx@maybank.com*.

The respondents were asked of the method, they used to save their passwords or ID. All of them claimed to save it, as none of them has to write down the ID and password. The ID and password as they only have to remember one ID and password. This shows that one of the objective of single sign on in minimize the usage of writing down the IDs and passwords has been achieved.

Most of these respondents never changed their passwords unless it is necessary. This is because some companies apply a policy of changing the employee's passwords every three or four months.

An addition to this, the respondents owns a smartcard, which the main purpose is to mark their working time and attendance. They need this card to enter other building (Texas Instrument Malaysia, 2005) or just as an identification.

However, the personnel's of a department in Texas Instrument Malaysia claimed that they are not familiar to this technology and willing to know more of this technology. After a discussion with them, we found out that they are using single sign on especially when sharing information with their main headquarter in America. After we had explained to them the meaning of single sign on then we can continued our interview.

4.2.1 Conclusion

The purpose of this interview is to understand the different single sign on application one company used than the other. By having this interview, we can understand that this technology is beginning to widely enter the market. At this moment, most of the small, medium industries have not applied this technology, as most of them do not see the usage of computer as a medium of communication and technology. From the interview, we can know that most of the interviewed are now happy with the single remembrance of ID and password. Even though their password consists of alphanumeric, it is not a problem to any of the respondents. They also knew that they should not write their passwords on a paper because somebody might steal it. The respondents are not changing their password unless it is necessary or policy.

So, with that reasons, we agree of single sign on can help in remembering the respondents userID and passwords.

4.3 SINGLE SIGN ON EMAIL APPLICATION TESTING

In this chapter, we are going to discuss the result from the testing phase. The testing was conducted in April 2006 at a cyber café in Taman Sri Serdang. The respondents were the customer of this café. There were 10 men and 10 women. They were between 16 to 25 year olds. The Figure 4.1 below shows the age and quantity of each age.

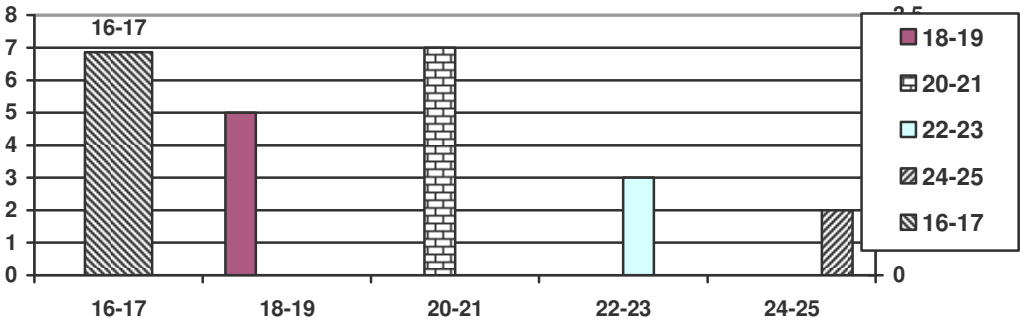

Figure 4.1: The respondents' age range.

The respondents were divided into two groups. They are multiple sign on group (Group A) and single sign on group (Group B). Before we start the test, we asked them to write two of their email ID and passwords. Please refer to Appendix C to see the list of userID and passwords used.

Before we started the test, members of group B were given a master userID and master password to let them log onto the application. Briefly explained by us, we asked them to rememorize the passwords. After two minutes, one respondent from each group is asked to log on to their account. The time took by the respondents was recorded for future used. The respondents were required to log onto two email accounts to show that by having single sign on, we can minimize the time. Our recorded time only took from the start of the input of the userID and password until the respondent entered 'Enter" to the page. We eliminated the time between the "Enter" actions until the page appear on the desktop. This is to make sure that we have the precise recorded time. Figure 4.2 shows the time taken by both groups with Group A (264 seconds) and Group B (112 seconds).

At the end of the testing phase, both of the groups agreed that by using single sign on email application, they could minimize the time used in typing their userID and passwords.

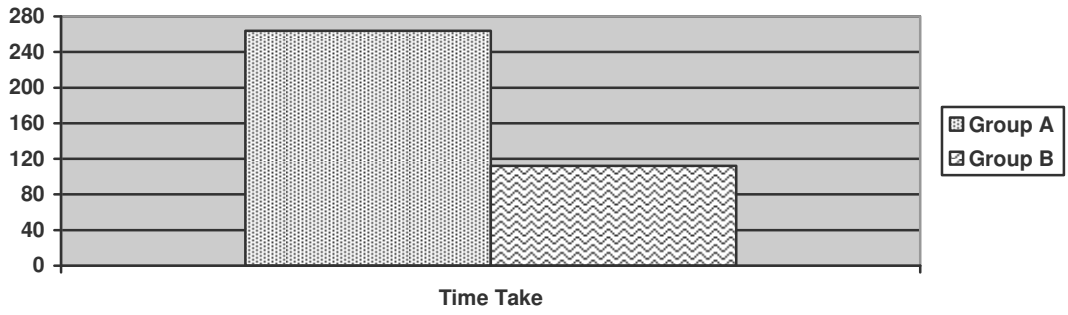

Figure 4.2: Total time taken by both groups to log onto the application

4.4 CONCLUSION

In this chapter, we have discussed about the findings and results from the interview and email application testing we have conducted before. From the interviews hat have been conducted, we found out that single sign on is widely used in these companies but not widely used in other industries. Most companies that are currently using single sign on are more focus on signing on to the desktop or workstations. However, the smartcard are still widely used in the companies that we have visited. It is found that the respondents do not write their passwords or ID on any paper. They also do not change their password unless it is necessary. If they do have to, they were already given the reminder of do not write their password at any surface or paper.

On the second part, the main conclusion is that by applying to the single sign on email application, we have minimized the time-constraint to log on to several emails accounts. With this, single sign on password is easier to remember than the multiple sign on passwords.

CHAPTER V

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

In today's world, hardware and software development are growing rapidly. Companies compete with each other in producing the best software or hardware and variety types, shapes and sizes for their products and services. A few years back, most of the single sign on can only be found in the form of software but not today. In today's world, we can find hardware to be used as a single sign on device. The devices are smartcard, retina recognition, voice recognition, biometric, USB and others to represents single sign on. The main issue will always be on the security when implementing single sign on.

In the first chapter, we have described about the problem statement, project objectives, project scope and limitation, and the significance. In the problem description, we discussed on the user problems of having to remembering several userID and passwords. The rise of Internet has caused the users to memorize even more of userID and passwords.

In the second chapter, we have discussed on the literature review. The literature reviews were about the working flow of the single sign on from

different developers. We had also combined related literatures in this chapter. Besides that, there is a section to compare between enterprise SSO and web-based SSO.

In the third chapter, we discussed on the methodologies used. The methodologies used are information gathering, data analysis, implementation phase and testing phase. The most important phase will be the information gathering phase as that is the core of this project.

In the fourth chapter, we discussed the findings and results of the testing phase. In this chapter, we discussed the finding that we found during our interview and observation session with our respondents. Also in this chapter is our finding during the testing phase.

The last chapter concludes our thesis. In this chapter, we discussed the conclusions and recommendations for the future development. The recommendations are to help the future improvement of the related projects.

Different companies applied different device of single sign on. A high profile company like Petroleum Nasional (Petronas) will always have a different security compare to banks like Maybank and Bank Pertanian Malaysia. The purpose of security of one's organization is different from others. As for Petronas, they are protecting the information of their oil and expenditures while

the banks will always want to protect the information of their customers and credits within the bank. At the end, all of these organizations do have the same opinion, which is the smart card is one of the most reliable sign on device and it does not cost a fortune too.

With the used of our single sign on email application, we have shown that we can easily create our own application with the knowledge of programming language such as C or even php. The most important thing is we have to know the main idea behind this project, which is to show a single sign on application.

5.2 RECOMMENDATIONS

This project has detected some problems during developing this project. They are:

i. Most organizations are not willing to show us how their single sign on application is done.

ii. System administrators were not willing to share knowledge in terms of single sign on devices limitations.

However, to solve this problem, we have two recommendations to it. Those are:

i. Develop specific profiles or guidelines that describe how the standards should be used in particular applications.

ii. Provide free software and free CAs so people can set up a test SSO with or without cost.

iii. Provide a “Helpbook” with easy steps for developing an easy to guide any single sign on software applications (home emails, personal computers at homes, and etc)

REFERENCE:

Britt, P. (January 2005) "*Bank System and Technology: Who Goes There*?" (http://www.banktech/articles71u7.com) retrieved on 1 February 2005

Carroll, M. (July 2004) "*Protocom Development System: What is Window Finder*" (www.protocom.com/supportsso.html) retrieved on 23 February 2005.

Carroll, M. (October 2004) "*Protocom Development System: The Differences between Single Sign On in Directory mode and Standalone mode*" (www.protocom.com/supportsso.html) retrieved on 23 February 2005.

Carroll, M. (December 2004) "*Protocom Development System: The Recommended Setting for a Corporate Single Sign on Deployment*" (www.protocom.com/sso.services.html) retrieved on 23 February 2005.

Carroll, M. (January 2005) "*Protocom Development System: Single Sign on Enabling Internet Banking*" (www.protocom.com/intbanking.html) retrieved on 20 February 2005.

Carroll, M. (October 2004) "*Protocom Development System: How Can Single Sign On Enable Telnet? Using Terminal Launcher*" (www.protocom.com/index.product.html) retrieved on 20 February 2005.

Carroll, M. (January 2005) "*Protocom Development System: How to ensure Single Sign on Starts before All Enabled Application*" (www.protocom.com/index.secure_login.html) retrieved on 23 February 2005.

Carroll, M. (January 2005) "*Protocom Development System: How to centrally administer Single Sign On*" (www.protocom.com/index. article 291r.html) retrieved on 20 February 2005.

Carroll, M. (January 2005) "*Protocom Development System: What is the Minimum Length of the Passphrase Answer*" (www.protocom.com/index/customer_support.knowledge.html) retrieved on 28 February 2005.

Carroll, M. (January 2005) "*Protocom Development System: Single Sign on Enabling Internet Banking"* (www.protocom.com/index./article 291e.html) retrieved on 28 February 2005.

Dunne, C. (September 2003) "*Build an implement single sign on solution*"

(http://www-106.ibm.com/developerworks/java/library/wa-singlesign/?ca=dgr-lnxw914CASss0#author1) retrieved on 24th February 2005.

Evidian "*Evidian: Identify and Access Management for the Banking Environment*" (www.evidian.com/security/index/article7889) retrieved on 9 March 2005.

Gerson, V. (November 2004) "*Bank System and Technology: No Traffic Jams*" (http://www.banktech.com) retrieved on 1 February 2005.

Lubow, E. (October 2003) "*Single Sign On: Go to Open Source*" (http://www.linuxsecurity.com) retrieved on 28 January 2005.

McKay, A. (2002) "*Plone.org: Single Sign On in Windows Domains*" (http://plone.org/documentation/how-to) retrieved on 10 March 2005.

Musthaler, L.(January 2002) "*Network World Fusion: The Holy Grail Of Single Sign On*" (www.nwfusion.com/security/holygrail462.com) retrieved on 25 February 2005.

Protocom Development System "*Protocom Development System: Benefits*" (www.protocom.com/index.html) retrieved on 23 February 2005.

Protocom Development System "*Protocom Development System: Single Sign on Overview*" (www.protocom.com/index.html) retrieved on 23 February 2005.

Runkel.S (May 2004) "*Quadtech: Leveraging Legacy Equipment in Banking Environment*" (www.quatech.com/products/products.banking235e23h2.php) retrieved on 28 February 2005.

Tao, J. (October 2002) "*Implement Single Sign On with JAAS*" (http://www.devx.com/devx/JAVA/Article.6353) retrieved on 20 January 2005.

Taylor, L. (May 2002) "*Jupiter Research: Understanding Single Sign on Part 1*" (www.jupiterevents.com/journal56-661.htm) retrieved on 3 March 2005.

Taylor, L. (May 2002) "*Jupiter Research: Understanding Single Sign on Part 2*" (www.jupiterevents.com/journal56-662.htm) retrieved on 3 March 2005.

Taylor, L. (May 2002) "*Jupiter Research: Understanding Single Sign on Part 3*" (www.jupiterevents.com/journal56-663.htm) retrieved on 3 March 2005.

Thurman, M (December 2003) "*Single Sign on Effort Falls Short: Security Manager's Journal Page 1*" (www.computerworld.com/security_topics/security.htm) retrieved on 5 February 2005.

Thurman, M (December 2003) "*Single Sign on Effort Falls Short: The No-Name Log-in Page 2*" (www.computerworld.com/security_topics/security.htm) retrieved on 5 February 2005.

Thurman, M (December 2003) "*Single Sign On Effort Falls Short: Page 3*" (www.computerworld.com/security_topics/security.htm) retrieved on 5 February 2005.

Webopedia: Word-Single Sign On (www.webopedia.com/network/security/references/sso) retrieved on 17th January 2005.

Davis F. D. (1989). *Perceived usefulness, perceived ease of use, and user acceptance of information technology.* MIS Quarterly, 13(3), 319-340.

Elliza J. (2004). *Designing and implementing online testing system base on cognitive domain of Bloom's Taxonomy. 21-32.*

Nor Asyikin S. *(2004). A Study of the usage of single sign on.*

Zanaria J. *(2004). An interactive approach to memorable passwords.*

Suhaida A. *(2004). Evaluation of the implementation of E-CRM in internet banking industry.*

APPENDIX A

In the below table is the summary of single sign on products. Some of the products are used (applied) to organization this Malaysia.

Product Name	**Description**
AccessMaster Single Sign-On (Evidian)	Can handle tens of thousands of internal or remote users and provide 24x7 service. Its unique architecture allows it to be easily deployed to access all systems and applications, without changing any target.
AccessMatrix Universal Sign-On (i-sprint)	AccessMatrix Universal Sign-On (USO) is a non-intrusive Enterprise SSO solution that enables organizations to achieve single sign-on to multiple applications and systems. In most organizations today, users are often required to remember many IDs and passwords in order to perform their various job functions. By deploying our enterprise single sign-on solution, our clients will improve staff and customer satisfaction, resulting in improved productivity and reduced administration costs. Client/server, host-based, Java-based and web-based applications are supported without source code changes. Unlike other single sign-on products, manual client software installations are not required on the users' desktops.
Entrust/SignOn (Entrust Technologies)	Eliminates the need for multiple logins to Windows and secure applications. Entrust/SignOn means fewer passwords to remember, fewer help desk calls due to forgotten passwords, and easy to use security.
OneSign (Imprivata)	Imprivata® OneSign(tm) is an easy, smart and affordable Enterprise SSO appliance that lets organizations quickly and effectively automates password management and user authentication policies. OneSign's breakthrough Application Profile Generator (APG(tm)) technology enables secure, seamless SSO to ALL applications, without the need for modifications or custom scripting. Customers benefit from dramatically reduced costs, increased employee productivity, stronger password security, and increased regulatory compliance.
Passwerks (ASG Technologies)	A Network Access Control and Security product designed specifically for Service Providers in the

	Telecommunications industry. Includes single sign-on, password management, and network administrator interface capabilities.
SecureLogin (Novell)	Novell® SecureLogin 2.5 is a directory-based authentication solution that extends single sign-on access to virtually every application within your multi-platform network environment.
SecureLogin (Protocom)	Protocom SecureLogin includes legacy and internet single sign-on for an enterprise infrastructure consisting of the following: any LDAP directory, Novell eDirectory (NDS), Novell SecretStore, NT Domains, Active Directory, Solaris 2.6 or above and Linux. Protocom SecureLogin also provides advanced authentication for use with any combination of biometrics, smartcards, hardware tokens, software tokens and PKI.
Single Sign-On (Computer Associates)	Provides secure single sign-on and enterprise-wide security administration. Highly ranked by analysts and customers, Single Sign-On automates user logins to traditional systems and applications as well as to web-based applications. Single Sign-On offers extensive integration with a wide range of applications, management solutions and authentication methods, such as PKI, smart cards, tokens and biometrics.
SP Sign-On (Unisys)	Includes features to protect the enterprise at two points of vulnerability: User identification/authorization – including biometric identification – provide first-level protection at the workstation itself. Single sign-on features control access to networked resources based on individual or group authorizations while facilitating authorized access to networked resources. Administration features facilitate policy-driven administration of user accounts throughout the enterprise.
SSO Plus (PassGo)	SSO Plus is an easy to use, flexible password management product. It provides the user with the ability to access all applications on their desktop using a single user ID and password. After the user has logged onto their PC, they can access password-protected applications on their desktop without the need to enter any further credentials. SSO Plus learns the logon credentials of each application started on the user's desktop. When the application is next started, SSO Plus will automatically enter the required logon credentials. There is no need for users to write down the logon

	credentials for each application; no need for password synchronization and it encourages the use of strong passwords.
Tivoli Global Sign-On (Tivoli)	Provides a secure, single point of entry to computing resources that enables organizations to connect disparate networked systems. This results in significant benefits, such as: Increased security — Supports password, fingerprint, or smartcard authentication to confirm authorized users. Easier administration — Simplifies setting up and managing passwords and IDs for users. Increased productivity — Reduces the time required to complete logons, and simplifies system and user management.
TrustBroker Security Suite (CyberSafe)	Delivers end-to-end security, enabling secure communications inside the corporate network, as well as to key dealers, suppliers, and customers outside the network. It leverages individual trust relationships, and then brokers that trust across enterprise-wide distributed client/server networks, helping you meet both enterprise security and business objectives.
V-Go(Passlogix)	Enables comprehensive Single Sign-On through public-key infrastructure, directory services, and independently. Works out-of-the-box with virtually any Windows, Web, proprietary or host-based application. No integration or scripting required. Works with: Windows enterprise and client-server applications; Mainframe, Unix, AS/400 and other host-based applications; Web and browser-based applications; Microsoft Windows 95, Windows 98, Windows NT, Windows 2000; Microsoft Networking, Novell Netware, Novell NDS client; Entrust/PKI 4.0 and 5.0.

Table Ap 1: Summary of SSO Products.

APPENDIX B

Name:

Age:

Level of Education:

Job Description:

1. Are you aware of the technology called ‘Single Sign On’?
2. Do you know whether the solution is being used in your organization?
3. Does the smart card works are one of the signing on devices?
4. Is it widely used by all level of management?

   -if no, until what management level does this technology applied to?
5. How do you find this technology usefulness in terms of your job description?
6. In your opinion, do you find any benefits of applying this technology to other industries?
7. What are the accounts/servers that be accessed by you on the application?
8. Do you own any email account?
9. Does single sign on helps you in checking your mails and workloads?

APPENDIX C

Group A
Using their own Email IDs and passwords

Respondent	Email ID	Password
1	Wickedgurl	Kay1408
	Sweetplum82	Kmf8961
2	Lohtm3181	Sk9995
	Flyingboy	Sk1234
3	Jasonng	Nyw18181
	Siptik81	200181
4	Ahmadimran	1901817
	Ahmad_imran	191817
5	Puteri_girl	Jasmine82
	Puteri_jasmine	Js161082
6	Max_de_la_vega	Iiz9885
	Ju_united	Izzati85
7	amirhana	Upm556612
	Amir_hana	Upm556612
8	coolDaniel	200589
	Daniel_syaf	89612005
9	Nordaliza	Ros010108
	Dal2229	Ros5512
10	Hprock2000	Rockylee
	Babydotrock	Rockylee123

Table Ap 2: List for Group A

Group B
Using the given Master userID and Master password

Respondent	Master userID	Master password	Email ID	password
1	Resp001	Pass001		
			kimipablo	Kim2006
			lanencem	Popo221
2	Respoo2	Pass002		
			tuticu	Fai3056
			Ganesh33	Cam122
3	Resp003	Pass003		
			Anees2	Ummi9883
			faislaw	209345
4	Resp004	Pass004		
			orangegirl	Quatic14
			kenchu	Xt151
5	Resp005	Pass005		
			yenjerry	Cx0101
			kangta	070234
6	Rp006	Pass006		
			vannesswu	Jianhao0708
			kellyche	Wailin1008
7	Resp007	Pass007		
			Liz20056	Mas1945
			weilang	Jun2674
8	Resp008	Pass008		
			yaoming	China1007
			ainshahir	Kiut3546
9	Resp009	Pass009		
			nanibaba	Mann2676
			peggie	Laupk3939
10	Resp010	Pass010		
			kamal	Siha3895
			kashie	Ama4455

Table Ap 3: List for Group B

APPENDIX D

Appendix (example of interfaces)

Figure Ap 1: An HTML page showing GSS applet usage

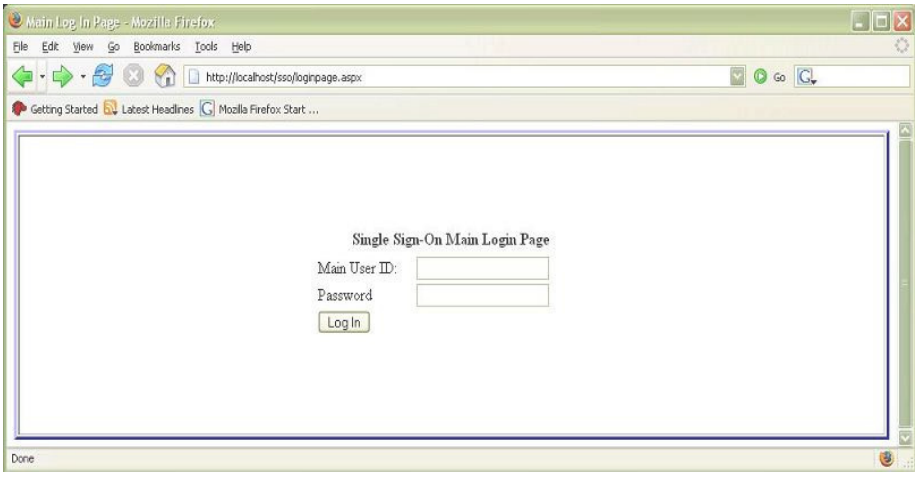

Figure Ap 2: SSO Main Login page

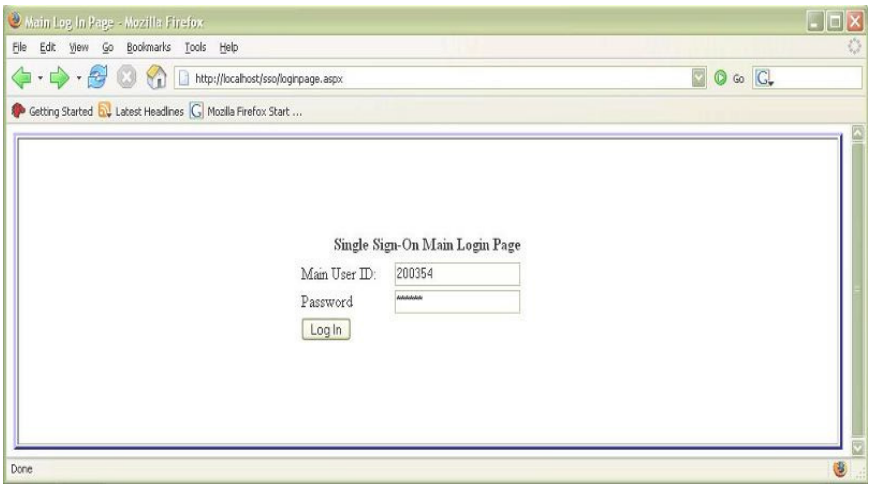

Figure Ap 3: SSO Main Login Page (with characters)

Figure Ap 4: Welcome Page

Figure Ap 5: Main Email Account Page

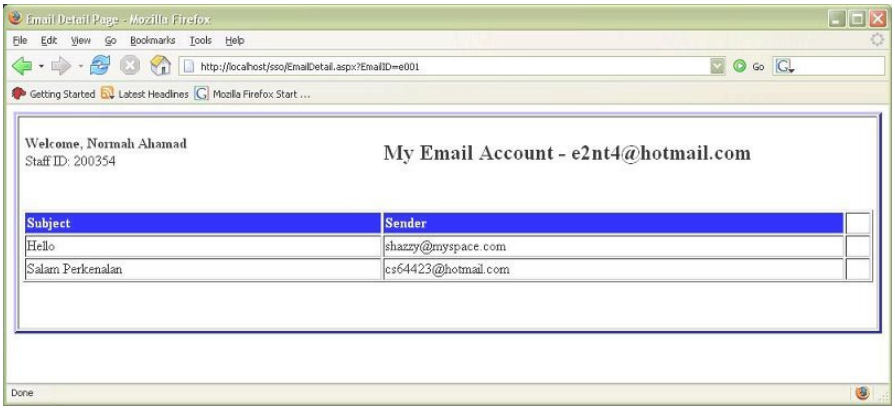

Figure Ap 6: Email Account Page (account 01)

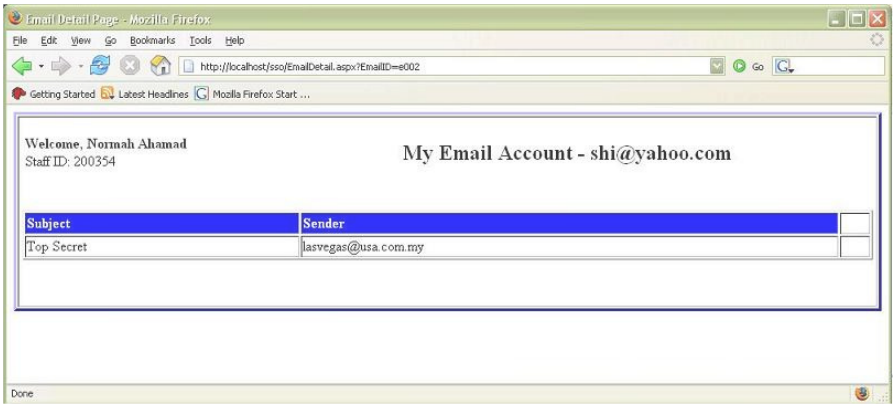

Figure Ap 7: Email Account Page (account 02)

figure Ap 8: SSO Main Login Page (wrong ID or password)

APPENDIX E

Main login page

```
<%@ Import Namespace="System.Data"%>
<%@ Import Namespace="System.Data.SqlClient"%>
<%@ Import Namespace="System.Web.Security "%>
<%@ Import Namespace="System.IO"%>
<%@ Page Language="C#" Debug="true" ContentType="text/html"
ResponseEncoding="iso-8859-1" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Main Log In Page</title>
</head>
<script language="c#" runat="server">
void BtnLogIn_Click(Object Src, EventArgs E)
{
                String cmd = "UserID='" + MainUid.Text + "'";
                SqlConnection MyConn = new SqlConnection("Data Source =
NORMAH;Integrated Security = SSPI;"+"Initial Catalog = sso;"+"uid=sa;"+"pwd=sa");
                MyConn.Open();
                SqlCommand cmdPrompt = new SqlCommand("SELECT * FROM
LogIn WHERE UserID = '"+MainUid.Text+"' AND Password =
'"+MainPwd.Text+"'",MyConn);
                SqlDataAdapter da = new SqlDataAdapter();
                da.SelectCommand = cmdPrompt;
                DataSet ds = new DataSet ( );
                da.Fill(ds,"LogIn");
                DataTable users = ds.Tables[0];
                DataRow [  ] matches = users.Select ( cmd );
                MyConn.Close();
                if ( matches != null && matches.Length > 0 )
                {
                        DataRow row = matches [ 0 ];
                        string pass = ( string ) row [ "Password" ];
                        if ( 0 != String.Compare ( pass, MainPwd.Text, false ) )
                        {
                                Message.Text="Sorry, you have entered wrong user ID or
password";
```

```
                        }
                        else
                        {
                                Session ["curUID"] = MainUid.Text;
                                Response.Redirect("home.aspx");
                        }
                }
                else
                        Message.Text="Sorry you have entered wrong user ID or
password";
}
</script>
<body>
<table width="100%" height="300" border="3" bordercolor="#0000FF">
  <form runat="server"><tr>
    <td><div align="center">
      <table width="100%" border="0">
        <tr>
          <td colspan="3"><div align="center"><strong>Single Sign-On Main Login
Page</strong></div></td>
        </tr>
        <tr>
          <td width="25%"><div align="center"></div></td>
          <td width="50%"><div align="center">
            <table width="300" border="0">
              <tr>
                <td width="36%"><div align="left">Main User ID: </div></td>
                <td width="64%"><asp:TextBox ID="MainUid" TextMode="SingleLine"
runat="server"></asp:TextBox></td>
              </tr>
              <tr>
                <td><div align="left">Password</div></td>
                <td><asp:TextBox ID="MainPwd" TextMode="Password"
runat="server"></asp:TextBox></td>
              </tr>
              <tr>
                <td colspan="2"><div align="left"><asp:Button ID="BtnLogIn"
OnClick="BtnLogIn_Click" Text="Log In" runat="server"></asp:Button></div></td>
              </tr>
            </table>
          </div></td>
          <td width="25%"><div align="center"></div></td>
        </tr>
        <tr>
```

```
        <td colspan="3"><div align="center"><asp:Label ID="Message"
ForeColor="#FF0000" runat="server"></asp:Label></div></td>
       </tr>
     </table>
   </div></td>
  </tr></form>
</table>
</body>
</html>
```

Main Menu Page

```
<%@ Import Namespace="System.Data"%>
<%@ Import Namespace="System.Data.SqlClient"%>
<%@ Import Namespace="System.Web.Security "%>
<%@ Import Namespace="System.IO"%>
<%@ Page Language="C#" ContentType="text/html" ResponseEncoding="iso-8859-1"
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Main Menu Page</title>
</head>
<script language="c#" runat="server">
void Page_Load(Object Src, EventArgs E)
{
        staffID.Text=Session ["curUID"].ToString();
        string id=Session ["curUID"].ToString();
        String cmd = "UserID LIKE '" + id + "'";
        SqlConnection MyConn = new SqlConnection("Data Source =
NORMAH;Integrated Security = SSPI;"+"Initial Catalog = sso;"+"uid=sa;"+"pwd=sa");
        MyConn.Open();
        SqlCommand cmdPrompt = new SqlCommand("SELECT * FROM LogIn
WHERE UserID LIKE '"+id+"'",MyConn);
        SqlDataAdapter da = new SqlDataAdapter();
        da.SelectCommand = cmdPrompt;
        DataSet dsName = new DataSet ( );
        da.Fill(dsName,"LogIn");
        DataTable usersName = dsName.Tables[0];
        DataRow [  ] matches = usersName.Select ( cmd );
        MyConn.Close();
        if ( matches != null && matches.Length > 0 )
        {
                DataRow row = matches [ 0 ];
                string name = ( string ) row [ "StaffName" ];
                curUser.Text=name;
        }
        else
                curUser.Text="Guest";
}

void btnEmail_Click(Object Src, EventArgs E)
{
```

```
        Response.Redirect("EmailMain.aspx");
}

void btnAccount_Click(Object Src, EventArgs E)
{
        Response.Redirect("BankAccMain.aspx");
}
</script>
<body>
<table width="100%" border="3" bordercolor="#0000FF">
  <tr>
    <td>
          <table width="100%" border="0">
        <tr>
          <td width="28%"><strong>Welcome, <asp:Label ID="curUser"
runat="server"></asp:Label></strong><br />
                    Staff ID: <asp:Label ID="staffID" runat="server"/></td>
          <td width="72%">
                        <div align="center"><h2>Single Sign-On Application
Interface</h2></div></td>
        </tr>
        <tr>
          <td colspan="2"> </td>
        </tr>
        <tr>
          <td colspan="2">This interface enables the user to sign in to different types of
programs using only 1 &quot;Log In&quot; interface<br />
                        The following are the programs currently compatible with this
interface:                        </td>
        </tr>
        <tr>
          <td colspan="2"> </td>
        </tr>
        <tr>
          <td colspan="2"><div align="center">
            <table width="100%" border="0">
              <tr><form runat="server">
                <td width="50%"><div align="right"><asp:Button ID="btnEmail"
OnClick="btnEmail_Click" Text="Email Account Management Program"
ForeColor="#FFFFFF" BackColor="#3366FF" Font-Bold="true" runat="server"/>
                </div></td>
                <td width="50%"><div align="left"><asp:Button ID="btnAccount"
OnClick="btnAccount_Click" Text="Bank Account Management Program"
ForeColor="#FFFFFF" BackColor="#3366FF" Font-Bold="true"
runat="server"/></div></td>
```

```
            </form>
          </tr>
        </table>
      </div></td>
    </tr>
    <tr>
      <td colspan="2"> </td>
    </tr>
  </table>  </td>
 </tr>
</table>
</body>
</html>
```

Email Main Page

```
<%@ Import Namespace="System.Data"%>
<%@ Import Namespace="System.Data.SqlClient"%>
<%@ Import Namespace="System.Web.Security "%>
<%@ Import Namespace="System.IO"%>
<%@ Page Language="C#" ContentType="text/html" ResponseEncoding="iso-8859-1"
%>
<%@ Register TagPrefix="MM" Namespace="DreamweaverCtrls"
Assembly="DreamweaverCtrls,version=1.0.0.0,publicKeyToken=836f606ede05d46a,cu
lture=neutral" %>
<MM:DataSet
id="DataSet1"
runat="Server"
IsStoredProcedure="false"
ConnectionString='<%#
System.Configuration.ConfigurationSettings.AppSettings["MM_CONNECTION_STRI
NG_MyConn"] %>'
DatabaseType='<%#
System.Configuration.ConfigurationSettings.AppSettings["MM_CONNECTION_DAT
ABASETYPE_MyConn"] %>'
CommandText='<%# "SELECT * FROM dbo.EmailAccount WHERE UserID = ?" %>'
Debug="true"
>
  <Parameters>
    <Parameter  Name="@UserID"  Value='<%# (Session["curUID"] != null) ?
Session["curUID"] : "" %>'  Type="VarChar"   />
  </Parameters>
</MM:DataSet>
<MM:DataSet
id="DataSet2"
runat="Server"
IsStoredProcedure="false"
ConnectionString='<%#
System.Configuration.ConfigurationSettings.AppSettings["MM_CONNECTION_STRI
NG_MyConn"] %>'
DatabaseType='<%#
System.Configuration.ConfigurationSettings.AppSettings["MM_CONNECTION_DAT
ABASETYPE_MyConn"] %>'
CommandText='<%# "SELECT * FROM dbo.LogIn WHERE UserID = ?" %>'
Debug="true"
>
  <Parameters>
    <Parameter  Name="@UserID"  Value='<%# (Session["curUID"] != null) ?
Session["curUID"] : "" %>'  Type="VarChar"   />
```

```
  </Parameters>
</MM:DataSet>
<MM:PageBind runat="server" PostBackBind="true" />
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Email Main Page</title>
</head>
<script language="c#" runat="server">
//void email_link(Object Src, EventArgs E)
//{
//Session ["whichEmail"]=
//}
</script>
<body>
<table width="100%" border="3" bordercolor="#0000FF">
  <tr>
    <td><div align="center">
      <table width="100%" border="0">
        <tr>
          <td height="46"><div align="center">
            <table width="100%" border="0">
              <tr>
                <td height="28%"><div align="left"><strong>Welcome, <asp:Label
ID="curUser" runat="server" Text='<%# DataSet2.FieldValue("StaffName", Container)
%>'></asp:Label>
</strong><br />
                    Staff ID: <asp:Label ID="staffID" runat="server" Text='<%#
DataSet2.FieldValue("UserID", Container) %>'/></div></td>
                <td height="72%"><div align="center"><h2>Email Account Management
Program</h2></div></td>
              </tr>
            </table>
          </div></td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td></td>
        </tr>
        <tr>
          <td><form runat="server">
```

```
          <table width="100%" border="0">
            <tr>
              <td>Email ID </td>
              <td>Email Address </td>
              <td>Email UserID </td>
              <td> </td>
            </tr>
              <ASP:Repeater runat="server" DataSource='<%# DataSet1.DefaultView %>'>
                <ItemTemplate>
                                        <tr>
                  <td><a href="EmailDetail.aspx?EmailID=<%#
DataSet1.FieldValue("EmailID", Container) %>"><%#
DataSet1.FieldValue("EmailID", Container) %></a></td>
                  <td><%# DataSet1.FieldValue("EmailAddress", Container) %></td>
                  <td><%# DataSet1.FieldValue("EmailUid", Container) %></td>
                  <td> </td>
                                              </tr>
                </ItemTemplate>
              </ASP:Repeater>

          </table>
                </form></td>
        </tr>
        <tr>
          <td> </td>
        </tr>
      </table>
    </div></td>
  </tr>
</table>
</body>
</html>
```

Email Account Page

```
<%@ Import Namespace="System.Data"%>
<%@ Import Namespace="System.Data.SqlClient"%>
<%@ Import Namespace="System.Web.Security "%>
<%@ Import Namespace="System.IO"%>
<%@ Page Language="C#" ContentType="text/html" ResponseEncoding="iso-8859-1"
%>
<%@ Register TagPrefix="MM" Namespace="DreamweaverCtrls"
Assembly="DreamweaverCtrls,version=1.0.0.0,publicKeyToken=836f606ede05d46a,cu
lture=neutral" %>
<MM:DataSet
id="DataSet1"
runat="Server"
IsStoredProcedure="false"
ConnectionString='<%#
System.Configuration.ConfigurationSettings.AppSettings["MM_CONNECTION_STRI
NG_MyConn"] %>'
DatabaseType='<%#
System.Configuration.ConfigurationSettings.AppSettings["MM_CONNECTION_DAT
ABASETYPE_MyConn"] %>'
CommandText='<%# "SELECT * FROM dbo.EmailAccount WHERE UserID = ?" %>'
Debug="true"
>
  <Parameters>
    <Parameter  Name="@UserID"  Value='<%# (Session["curUID"] != null) ?
Session["curUID"] : "" %>'  Type="VarChar"   />
  </Parameters>
</MM:DataSet>
<MM:DataSet
id="DataSet2"
runat="Server"
IsStoredProcedure="false"
ConnectionString='<%#
System.Configuration.ConfigurationSettings.AppSettings["MM_CONNECTION_STRI
NG_MyConn"] %>'
DatabaseType='<%#
System.Configuration.ConfigurationSettings.AppSettings["MM_CONNECTION_DAT
ABASETYPE_MyConn"] %>'
CommandText='<%# "SELECT * FROM dbo.LogIn WHERE UserID = ?" %>'
Debug="true"
>
  <Parameters>
    <Parameter  Name="@UserID"  Value='<%# (Session["curUID"] != null) ?
Session["curUID"] : "" %>'  Type="VarChar"   />
```

```
  </Parameters>
</MM:DataSet>
<MM:PageBind runat="server" PostBackBind="true" />
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Email Main Page</title>
</head>
<script language="c#" runat="server">
//void email_link(Object Src, EventArgs E)
//{
//Session ["whichEmail"]=
//}
</script>
<body>
<table width="100%" border="3" bordercolor="#0000FF">
  <tr>
    <td><div align="center">
      <table width="100%" border="0">
        <tr>
          <td height="46"><div align="center">
            <table width="100%" border="0">
              <tr>
                <td height="28%"><div align="left"><strong>Welcome, <asp:Label
ID="curUser" runat="server" Text='<%# DataSet2.FieldValue("StaffName", Container)
%>'></asp:Label>
</strong><br />
                    Staff ID: <asp:Label ID="staffID" runat="server" Text='<%#
DataSet2.FieldValue("UserID", Container) %>'/></div></td>
                <td height="72%"><div align="center"><h2>Email Account Management
Program</h2></div></td>
              </tr>
            </table>
          </div></td>
        </tr>
        <tr>
          <td> </td>
        </tr>
        <tr>
          <td></td>
        </tr>
        <tr>
          <td><form runat="server">
```

```
          <table width="100%" border="0">
            <tr>
              <td>Email ID </td>
              <td>Email Address </td>
              <td>Email UserID </td>
              <td> </td>
            </tr>
              <ASP:Repeater runat="server" DataSource='<%# DataSet1.DefaultView %>'>
                <ItemTemplate>
                                        <tr>
                  <td><a href="EmailDetail.aspx?EmailID=<%#
DataSet1.FieldValue("EmailID", Container) %>"><%#
DataSet1.FieldValue("EmailID", Container) %></a></td>
                  <td><%# DataSet1.FieldValue("EmailAddress", Container) %></td>
                  <td><%# DataSet1.FieldValue("EmailUid", Container) %></td>
                  <td> </td>
                                              </tr>
                </ItemTemplate>
              </ASP:Repeater>

          </table>
                </form></td>
        </tr>
        <tr>
          <td> </td>
        </tr>
      </table>
    </div></td>
  </tr>
</table>
</body>
</html>
```

Email Login Page

```
<%@ Import Namespace="System.Data"%>
<%@ Import Namespace="System.Data.SqlClient"%>
<%@ Import Namespace="System.Web.Security "%>
<%@ Import Namespace="System.IO"%>
<%@ Page Language="C#" Debug="true" ContentType="text/html"
ResponseEncoding="iso-8859-1" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Email Log In Page</title>
</head>
<script language="c#" runat="server">
void BtnLogIn_Click(Object Src, EventArgs E)
{
                String cmd = "EmailUid='" + MainUid.Text + "'";
                SqlConnection MyConn = new SqlConnection("Data Source =
NORMAH;Integrated Security = SSPI;"+"Initial Catalog = sso;"+"uid=sa;"+"pwd=sa");
                MyConn.Open();
                SqlCommand cmdPrompt = new SqlCommand("SELECT * FROM
EmailAccount WHERE EmailUid = '"+MainUid.Text+"' AND EmailPwd =
'"+MainPwd.Text+"'",MyConn);
                SqlDataAdapter da = new SqlDataAdapter();
                da.SelectCommand = cmdPrompt;
                DataSet ds = new DataSet ( );
                da.Fill(ds,"LogIn");
                DataTable users = ds.Tables[0];
                DataRow [  ] matches = users.Select ( cmd );
                MyConn.Close();
                if ( matches != null && matches.Length > 0 )
                {
                        DataRow row = matches [ 0 ];
                        string pass = ( string ) row [ "EmailPwd" ];
                        DataRow row1 = matches [ 0 ];
                        string id = ( string ) row1 [ "EmailID" ];
                        DataRow row2 = matches [ 0 ];
                        string user = ( string ) row2 [ "UserID" ];
                        if ( 0 != String.Compare ( pass, MainPwd.Text, false ) )
                        {
                                Message.Text="Sorry, you have entered wrong  email user
ID or password";
                        }
```

```
                    else
                    {
                        Session ["curUID"] = user;
                        Response.Redirect("EmailDetail.aspx?EmailID="+id+"");
                    }
                }
                else
                    Message.Text="Sorry you have entered wrong user ID or
password";
}
</script>
<body>
<table width="100%" height="300" border="3" bordercolor="#0000FF">
  <form runat="server"><tr>
    <td><div align="center">
      <table width="100%" border="0">
        <tr>
          <td colspan="3"><div align="center"><strong>Email Log In
Page</strong></div></td>
        </tr>
        <tr>
          <td width="25%"><div align="center"></div></td>
          <td width="50%"><div align="center">
            <table width="300" border="0">
              <tr>
                <td width="36%"><div align="left">Email User ID: </div></td>
                <td width="64%"><asp:TextBox ID="MainUid" TextMode="SingleLine"
runat="server"></asp:TextBox></td>
              </tr>
              <tr>
                <td><div align="left">Password</div></td>
                <td><asp:TextBox ID="MainPwd" TextMode="Password"
runat="server"></asp:TextBox></td>
              </tr>
              <tr>
                <td colspan="2"><div align="left"><asp:Button ID="BtnLogIn"
OnClick="BtnLogIn_Click" Text="Log In" runat="server"></asp:Button></div></td>
              </tr>
            </table>
          </div></td>
          <td width="25%"><div align="center"></div></td>
        </tr>
        <tr>
          <td colspan="3"><div align="center"><asp:Label ID="Message"
ForeColor="#FF0000" runat="server"></asp:Label></div></td>
```

```
        </tr>
      </table>
    </div></td>
  </tr></form>
</table>
</body>
</html>
```

Email Account Detail Page

```
<%@ Page Language="C#" ContentType="text/html" ResponseEncoding="iso-8859-1"
%>
<%@ Register TagPrefix="MM" Namespace="DreamweaverCtrls"
Assembly="DreamweaverCtrls,version=1.0.0.0,publicKeyToken=836f606ede05d46a,cu
lture=neutral" %>
<MM:DataSet
id="DataSet2"
runat="Server"
IsStoredProcedure="false"
ConnectionString='<%#
System.Configuration.ConfigurationSettings.AppSettings["MM_CONNECTION_STRI
NG_MyConn"] %>'
DatabaseType='<%#
System.Configuration.ConfigurationSettings.AppSettings["MM_CONNECTION_DAT
ABASETYPE_MyConn"] %>'
CommandText='<%# "SELECT * FROM dbo.LogIn WHERE UserID = ?" %>'
Debug="true" PageSize="10" CurrentPage='<%#
((Request.QueryString["DataSet2_CurrentPage"] != null) &&
(Request.QueryString["DataSet2_CurrentPage"].Length > 0)) ?
Int32.Parse(Request.QueryString["DataSet2_CurrentPage"]) : 0  %>'
>
  <Parameters>
    <Parameter  Name="@UserID"   Value='<%# (Session["curUID"] != null) ?
Session["curUID"] : "" %>'  Type="VarChar"   />
  </Parameters>
</MM:DataSet>
<MM:DataSet
id="DataSet1"
runat="Server"
IsStoredProcedure="false"
ConnectionString='<%#
System.Configuration.ConfigurationSettings.AppSettings["MM_CONNECTION_STRI
NG_MyConn"] %>'
DatabaseType='<%#
System.Configuration.ConfigurationSettings.AppSettings["MM_CONNECTION_DAT
ABASETYPE_MyConn"] %>'
CommandText='<%# "SELECT dbo.LogIn.UserID, dbo.LogIn.StaffName,
dbo.Mail.EmailID, dbo.Mail.EmailSubject, dbo.Mail.EmailBody, dbo.Mail.Sender,
dbo.EmailAccount.EmailAddress  FROM dbo.Mail, dbo.EmailAccount, dbo.LogIn
WHERE dbo.LogIn.UserID = ? AND dbo.EmailAccount.EmailID = dbo.Mail.EmailID
AND dbo.Mail.EmailID=?" %>'
Debug="true"
>
```

```
  <Parameters>
    <Parameter  Name="@UserUID"  Value='<%# (Session["curUID"] != null) ?
Session["curUID"] : "" %>'  Type="VarChar"   />
    <Parameter  Name="@EmailID"  Value='<%# ((Request.QueryString["EmailID"] !=
null) && (Request.QueryString["EmailID"].Length > 0)) ?
Request.QueryString["EmailID"] : "" %>'  Type="VarChar"   />
  </Parameters>
</MM:DataSet>
<MM:PageBind runat="server" PostBackBind="true" />
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Email Detail Page</title>
<style type="text/css">
<!--
.style1 {
          color: #FFFFFF;
          font-weight: bold;
}
.style2 {color: #FFFFFF}
-->
</style>
</head>
<body>
<table width="100%" border="3" bordercolor="#0000FF">
  <tr>

  <td><table width="100%" border="0">
      <tr>
        <td><table width="100%" border="0">
          <tr>
            <td width="28%"><div align="left"><strong>Welcome,
              <asp:Label ID="curUser" runat="server" Text='<%#
DataSet1.FieldValue("StaffName", Container) %>'></asp:Label>
                </strong><br />
              Staff ID:
              <asp:Label ID="staffID" runat="server" Text='<%#
DataSet1.FieldValue("UserID", Container) %>'/>
            </div></td>
            <td width="72%"><div align="center">
              <h2>My Email Account - <%# DataSet1.FieldValue("EmailAddress",
Container) %></h2>
            </div></td>
```

```
        </tr>
      </table></td>
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr>
      <td><table width="100%" border="1" bordercolor="#999999">
        <tr>
          <td bgcolor="#0000FF"><span class="style1">Subject</span></td>
          <td bgcolor="#0000FF"><span class="style1">Sender</span></td>
          <td> </td>
        </tr>

          <ASP:Repeater runat="server" DataSource='<%# DataSet1.DefaultView %>'>
            <ItemTemplate>
                              <tr>
              <td><%# DataSet1.FieldValue("EmailSubject", Container) %></td>
              <td><%# DataSet1.FieldValue("Sender", Container) %></td>
                                    <td> </td>
                    </tr>
            </ItemTemplate>
          </ASP:Repeater>
      </table></td>
    </tr>
    <tr>
      <td> </td>
    </tr>
    <tr>
      <td> </td>
    </tr>
  </table></td>
  </tr>

</table>
</body>

</html>
```