

California State University, San Bernardino

CSUSB ScholarWorks

Theses Digitization Project

John M. Pfau Library

2005

WebSAT: Web-based systems administration tool

Juyong Jeong

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd-project>



Part of the [Digital Communications and Networking Commons](#)

Recommended Citation

Jeong, Juyong, "WebSAT: Web-based systems administration tool" (2005). *Theses Digitization Project*. 2667.

<https://scholarworks.lib.csusb.edu/etd-project/2667>

This Project is brought to you for free and open access by the John M. Pfau Library at CSUSB ScholarWorks. It has been accepted for inclusion in Theses Digitization Project by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

WebSAT: WEB-BASED SYSTEMS ADMINISTRATION TOOL

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Juyong Jeong


June 2005

WebSAT: WEB-BASED SYSTEMS ADMINISTRATION TOOL

A Project
Presented to the
Faculty of
California State University,
San Bernardino

by
Juyong Jeong
June 2005

Approved by:



Arturo I Concepcion, Chair, Computer Science

21 Mar 2005
Date



Josephine Mendoza



David Turner

© 2005 Juyong Jeong

ABSTRACT

WebSAT (Web-based Systems Administration Tool) provides the systems administrator several tools over the Internet. In this project, account management, disk space management, printer management, and security warning tools are implemented.

The systems administrator creates and deletes network accounts by using WebSAT. Also, he/she manages network accounts by using disable account and enable account functions. WebSAT shows a disk space summary in one screen for the systems administrator to manage disk space conveniently. Printer management tool shows the status of all network printers at a glance through this tool. The systems administrator can delete current printer jobs or jobs in printer queues. WebSAT can warn the systems administrator if someone is scanning the ports or the NIS and NFS servers.

PHP which is a server side embedded scripting language is used to implement WebSAT application, and MySQL is used as its database. Because a user accesses WebSAT over the Internet, security is an issue in this project. SSL (Secure Sockets Layer) is used on TCP/IP protocol to have an appropriate security level for WebSAT.

ACKNOWLEDGMENTS

I would like to express my deep gratitude to many people whose help has been crucial to my success in completing my project. First of all, I am very thankful to Professor Concepcion, my project advisor, for his direction and wise guidance of my project. Secondly, I owe a great deal to Professor Mendoza and Professor Turner, two committee members, whose supervision and advice have contributed to my study.

My heartfelt appreciation also goes out to all the people of Network Groups who have helped me during the progress of working on my project. In particular, my special gratitude is extended to Kwon Soo Han, the systems administrator of the Network Groups, who has been generous with his time and has provided helpful advice and careful guidance in working out certain problems.

The support of the National Science Foundation under the award 9810708 is gratefully acknowledged.

Finally, I would like to thank all my friends who have always supported me and have been a source of much encouragement.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER ONE: SOFTWARE REQUIREMENTS SPECIFICATION	
1.1 Introduction	1
1.1.1 Purpose of Project	1
1.1.2 Scope of Project	2
1.2 Overall Description	3
1.2.1 Product Perspective	3
1.2.2 Product Functions	10
1.2.3 User Characteristics	16
1.3 Software Specific Requirements	17
1.3.1 External Interface	17
1.3.2 Functional Requirements	45
CHAPTER TWO: DESIGN	
2.1 Architecture Design	50
2.1.1 System Design	50
2.1.2 Security Design	53
2.1.3 Database Design	56
2.2 Detailed Design	63
2.2.1 Home Class	63

2.2.2 Find Account Class	64
2.2.3 Create Account Class	65
2.2.4 Enable Account Class	67
2.2.5 Disable Account Class	68
2.2.6 Delete Account Class	69
2.2.7 Synchronize Account Class	70
2.2.8 Disk Manager Class	71
2.2.9 Alert Admin Class	72

CHAPTER THREE: SOFTWARE QUALITY ASSURANCE

3.1 Introduction	73
3.2 Unit Testing	73
3.3 System Testing	77
3.4 Scenarios	78
3.4.1 Find Account	78
3.4.2 Create Account	84
3.4.3 Enable Account	92
3.4.4 Disable Account	97
3.4.5 Delete Account	105
3.4.6 Synchronize Account	114
3.4.7 Disk Manager	124

CHAPTER FOUR: MAINTENANCE MANUAL

4.1 Files and Directories	127
4.2 Server Installation	128

4.3 Software Installation	132
4.4 Database Maintenance	132
4.4.1 Database Administration	132
4.4.2 Database Installation	133
CHAPTER FIVE: CONCLUSIONS AND FUTURE WORK	
5.1 Conclusions	136
5.2 Future Work	138
GLOSSARY OF TERMS	141
REFERENCES	144

LIST OF TABLES

Table 1.1. Hardware Specification for the Development Phase	7
Table 1.2. Hardware Specification for the Implementation Phase	8
Table 1.3. Software Specification for the Development Phase	9
Table 1.4. Software Specification for the Implementation Phase	9
Table 1.5. The Format of Class Roster	47
Table 2.1. Account Table	59
Table 2.2. Groups Table	60
Table 2.3. Status Table	61
Table 2.4. Admin Table	61
Table 3.1. Unit Testing Results	73
Table 3.2. System Testing Results	77
Table 4.1. Files and Directories	127

LIST OF FIGURES

Figure 1.1.	Deployment Diagram for WebSAT	4
Figure 1.2.	Use Case Diagram	11
Figure 1.3.	Login Page	17
Figure 1.4.	Home Page	18
Figure 1.5.	Find Account Form	19
Figure 1.6.	Create Account Menu Page	20
Figure 1.7.	Account Creation Form for Student	21
Figure 1.8.	Account Creation Form for Non-Student	22
Figure 1.9.	Enable Account Menu Page	23
Figure 1.10.	Enable Account List	24
Figure 1.11.	List of Disabled Account by Admin	25
Figure 1.12.	Disable Account Menu Page	26
Figure 1.13.	Disable Expired Account	27
Figure 1.14.	Disable Not Accessed Account	28
Figure 1.15.	Disable Account by Admin	29
Figure 1.16.	Candidate Accounts to Be Disabled	30
Figure 1.17.	Delete Account Menu Page	31
Figure 1.18.	Delete Student Group Accounts	32
Figure 1.19.	Delete Generic Group Accounts	33
Figure 1.20.	Account Deletion Form	34
Figure 1.21.	Candidate Accounts to Be Deleted	35

Figure 1.22. Synchronize System and Database Menu Page	36
Figure 1.23. Check Login Information Page 1	37
Figure 1.24. Check Login Information Page 2	38
Figure 1.25. Synchronize Login Information Page	39
Figure 1.26. Check Group Information Page 1	40
Figure 1.27. Check Group Information Page 2	41
Figure 1.28. Synchronize User Group Information Page	42
Figure 1.29. Update Last Login Information Page	43
Figure 1.30. Disk Manager Page	44
Figure 1.31. Where Clause in Standard Query Language Statement	46
Figure 1.32. Flow Chart for Account Creation	48
Figure 2.1. Architecture in Class Diagram of WebSAT	52
Figure 2.2. Class Diagram of Port Scan Alert System	54
Figure 2.3. Security Design of WebSAT	55
Figure 2.4. Extended Entity-Relationship Diagram	57
Figure 2.5. Relational Database Schema	62
Figure 2.6. Home Class	63
Figure 2.7. Find Account Class	64
Figure 2.8. Create Account Class	65
Figure 2.9. Enable Account Class.....	67

Figure 2.10. Disable Account Class.....	68
Figure 2.11. Delete Account Class	69
Figure 2.12. Synchronize Account Class	70
Figure 2.13. Disk Manager Class.....	71
Figure 2.14. Alert Admin Class	72
Figure 3.1. Demonstration of Find Account	79
Figure 3.2. Demonstration of Create Account	85
Figure 3.3. Demonstration of Enable Account	93
Figure 3.4. Demonstration of Disable Account	98
Figure 3.5. Demonstration of Delete Account	106
Figure 3.6. Demonstration of Synchronize Account	115
Figure 3.7. Demonstration of Disk Manager	125
Figure 4.1. How to Get Ready to Install Programs	128
Figure 4.2. How to Install MySQL	129
Figure 4.3. How to Install Openssl and Apache with Mod_ssl	130
Figure 4.4. How to Install PHP and Start Apache	131
Figure 4.5. Initial Database Creation	133

CHAPTER ONE
SOFTWARE REQUIREMENTS
SPECIFICATION

1.1 Introduction

1.1.1 Purpose of Project

The primary purpose of the project is to develop systems administration tools which assist the systems administrator's task of managing and controlling a computer network. The daily tasks of a systems administrator are managing user logins, monitoring systems activities, and administering file systems, devices, and network services. Among the daily tasks, managing user logins is a very repetitious and time-consuming job in a university environment. This project, WebSAT, will improve the performance of the systems administrator's job by automating several repetitious manual jobs.

The secondary purpose of the WebSAT project is to detect on the computer network any illegal access from the outside world. It is a known fact that the Department of Computer Science network has been attacked from the outside world. Thus, the high level of security is needed. In response to this security concern, WebSAT will offer an

enhanced security function which alerts the systems administrator for any symptom of the computer network is being attack.

1.1.2 Scope of Project

The WebSAT project provides tools to manage and monitor network accounts, devices, and services, and provide warning system for security attacks on the network. The functions of WebSAT project includes:

- To create a relational database schema for account management.
- To setup a network environment using NIS, NFS, Secure Web Server with MySQL and PHP.
- To provide account find interfaces.
- To provide create account interfaces.
- To provide enable account interfaces.
- To provide disable account interfaces.
- To provide delete account interfaces.
- To provide synchronize account interfaces.
- To implement PSAS (Port Scan Alert System).
- To implement DSCS (Disk Space Checking System).

1.2 Overall Description

1.2.1 Product Perspective

1.2.1.1 Systems Interfaces. When WebSAT creates a network account for a user, a user's information should be provided. A user's information can be provided from SIS+ and Account Request Form. When user information is provided from the Account Request Form, WebSAT creates a user login using a single account creation function. In this case, the required user information is student's first name, last name, major, and student ID number if the requestor is a student. When user information is provided from SIS+, the output of SIS+ will not be modified, and should satisfy the following rules.

- A single student's information must be in one line.
- Each line must have one student's information.
- Student's last name and first name must be in between column 7 and 29. Last name and first name are distinguished by a comma.
- Student ID must be in between column 30 and 40.
- Major must be in between column 52 and 55.

See Figure 1.1 for the deployment diagram for WebSAT.

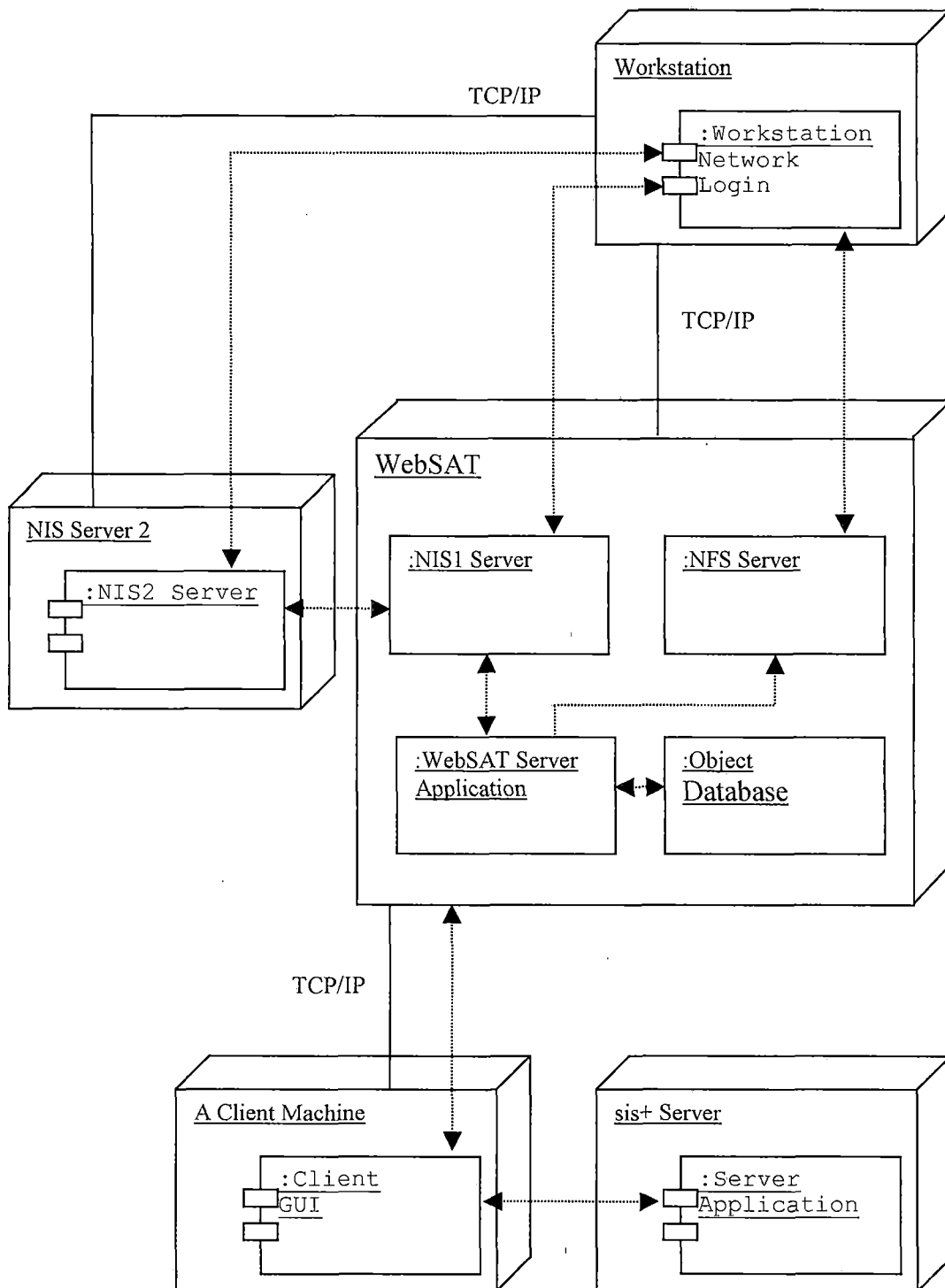


Figure 1.1. Deployment Diagram for WebSAT

The WebSAT system consists of four components: NIS server, NFS server, database system, and WebSAT application. In the current developing environment, there is secondary NIS server to backup the primary NIS. Individual workstations and a WebSAT client machine are connected over the network via TCP/IP protocol. A client machine is connected to sis+ server to get class information.

1.2.1.2 User Interfaces. The only user of WebSAT is the systems administrator. The access from the outside world must be denied for the security reason. WebSAT provides basically eight functions: Find Account, Create Account, Enable Account, Disable Account, Delete Account, Synchronize Account, Disk Management, and Port Scan Alarm System. The user of the system has a full access to these functions.

Let's imagine a scenario in the laboratory. There are three students: Jack, John, and Rose. Jack is a new student who takes CSCI 201 this quarter. Thus, he does not have an account. Unlike Jack, John is a returning student, so he has an account. John recently downloaded one movie file from the Internet, and the size of that movie file was 300MB. Rose, whose major was undeclared, previously decided to study computer science. However, her account is still

under student group. Ken plays the role of the systems administrator in this scenario. Ken decides that the maximum capacity of each account would be 50MB. Through WebSAT, Ken performs the following tasks: Ken gets the CSCI 201 student roster from SIS+ and creates the accounts for all CSCI 201 students including Jack. Jack has his new account and participates in lab activities. While monitoring disk space capacity, Ken recognizes that John's account occupies over 300MB disk space. Ken sends him a warning email. In spite of three warning emails, John did not delete his movie file which occupies 300MB. Finally, Ken disables John's account. Later, John comes and promises Ken to delete that movie file. Soon after Ken activates John's account, John deletes the movie file. Rose requests Ken for a computer science student account. So Ken will not disable her account after the current quarter ends. Ken changes Rose's account information in the system and transfers her files from the student's directory to csci directory. Finally, Ken updates WebSAT database information.

The scenario above is a simple example, but it describes how WebSAT system will provide important functionalities to the systems administrator.

1.2.1.3 Hardware Interfaces. This project is accomplished by updating current network systems to the new WebSAT embedded network system. Due to the complexity of this project, the Department of Mathematics computer Network was used for the development phase because it had the same network environment features as the Department of Computer Science, and its complexity is relatively less. Table 1.1 is the hardware specification, which was used for the development phase. After the development phase, WebSAT will be installed and run in the Department of Computer Science network. The Department of Computer Science hardware specification is show in Table 1.2.

Table 1.1. Hardware Specification for the Development Phase

Hardware	Specification
Processor	1GHz
Memory	1GB
Storage	50GB

Table 1.2. Hardware Specification for the Implementation Phase

Hardware	Specification
Processor	Two 1GHz
Memory	1GB
Storage	160GB

1.2.1.4 Software Interfaces. To implement this project, NIS server, NFS server, Secure Web server, PHP server, and MySQL server are needed in the Linux operating system. Moreover, HomeSite and Photoshop are used for creating PHP scripts and graphics. In the development phase, older versions of software were used. However, the newest versions of software will be used in the implementation phase. Table 1.3 shows the software specification for the development phase, and Table 1.4 shows the software specification for the implementation phase.

Table 1.3. Software Specification for the Development Phase

Software	Specification
Operating System	Redhat Linux 7.1
Secure Sockets Layer	Openssl 0.9.6 mod_ssl 2.8.5
Web Server	Apache 1.3.22
PHP	php 4.1.0
Database	mysql 3.23
Script Editor	allaire HomeSite v4.5.2
Graphics Editor	Adobe Photoshop 6.0

Table 1.4. Software Specification for the Implementation Phase

Software	Specification
Operating System	Fedora Core 3
Secure Sockets Layer	Openssl 0.9.7e mod_ssl 2.8.22
Web Server	Apache 1.3.33
PHP	php 4.3.9
Database	mysql 4.1.7
Script Editor	allaire HomeSite v4.5.2
Graphics Editor	Adobe Photoshop 7.0

1.2.1.5 Communications interfaces. WebSAT is implemented based on Web technologies, so a user communicates with the WebSAT server from a client over the Internet. Because some transactions between server and client include critical information, WebSAT uses SSL (Secure Sockets Layer) protocol as its communications.

1.2.2 Product Functions

In WebSAT system, all functions are developed for the systems administrator. The major functions of WebSAT are: finding account, creating account, enabling account, disabling account, deleting account, synchronizing account, managing disk space, and warning for security attacks on the network. Figure 1.2 is the use case diagram of WebSAT.

1.2.2.1 Find Account. To manage network accounts, the systems administrator frequently checks how many accounts are in the system, what account logins are available for new user, and when existing accounts were created, or disabled. The systems administrator can query those information using find account function. Logins are retrieved by login names, student ID, last names, first names, creation date, disabled date, and last login date. All logins found with this function are sorted by login,

user name, group, creation date, last login date, account status, or disabled date.

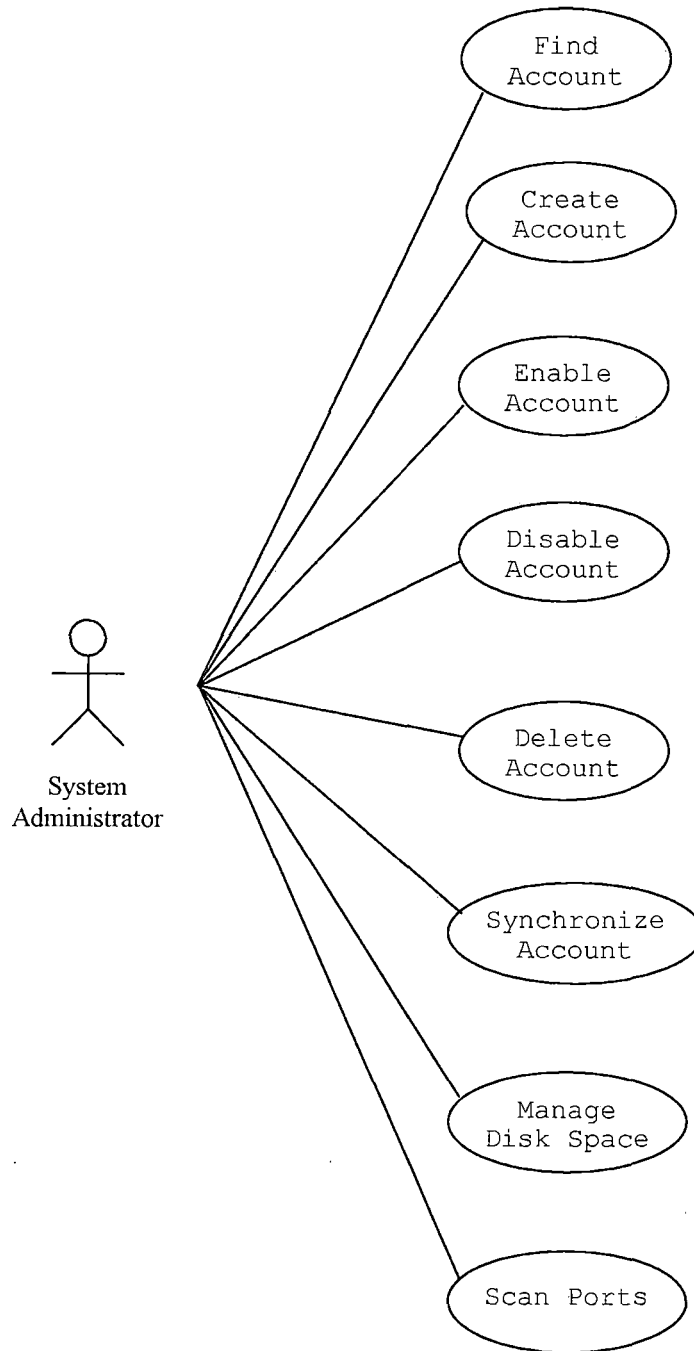


Figure 1.2. Use Case Diagram

1.2.2.2 Create Account. Create account function

divides into two sub functions: student account creation and non-student account creation. Student accounts are accounts created for students who take classes, and non-student accounts are accounts created for people who are not students. For example, faculty, staff, and others. Non-student accounts do not require students ID, whereas students account require students ID. That is, non-student account creation function is distinguished from student account creation function.

There are two forms in the student account creation interface. First form is to be used for single account creation, and the second form is to be used for multiple accounts creation. When the systems administrator creates accounts for students, he/she does not need to provide logins because WebSAT creates available logins for students. However, in terms of single mode account creation, the systems administrator can provide a login if a special login is required for the student. Passwords for new accounts are automatically set up so that the systems administrator does not worry about them. On the other hand, non-student creation interface has only one form. It

creates only one account at one time, and login and password must be provided by the systems administrator.

1.2.2.3 Disable Account. An account is disabled by the systems administrator when it has expired or when he/she decides that the account must be disabled. At the end of quarter terms, all the student group and generic group accounts become disabled because those two groups accounts are valid only for one quarter. Because there is a possibility that the user of such an account may perform illegal operation in the network or it may accumulate too many files, which exceeds a given quota limit on the network file system, their accounts are disabled.

1.2.2.4 Enable Account. Enable account function is used to activate a disabled account. As mentioned above, there are two cases for disabled accounts: expiration and the systems administrator's decision. When the systems administrator enables an account, he/she needs to know why the account was disabled. If the account was disabled because it was in the student or generic group, or it has expired, the disabled account can be enabled from the enable student and generic group account menu. If an account was disabled by the systems administrator's decision, the account can be enabled from the enable

account disabled by admin menu. Both menus display all disabled accounts. The systems administrator selects the proper accounts to be enabled in the list and clicks an enable button. Then, all the selected accounts are enabled.

1.2.2.5 Delete Account. There are many reasons why an account cannot be accessed by its owner. One of these reasons can be a student graduated from the university, or a student who does not need his/her account anymore. In this case, the systems administrator discovers that the student's account has not accessed for a long time and decides to delete that account. For some other reason, if the account has not been accessed for certain period of time, the systems administrator will decide on whether to delete this account or not.

1.2.2.6 Synchronize Account. There are three options for this function: login information synchronization, user group information synchronization, and last login information update.

When the systems administrator manages network account system, he/she may install some programs in the system, and sometimes logins for those programs are created during the installation times. In these cases, WebSAT may lose the coherency of information in the system and in the database

because the system has logins which are not found in the database. The systems administrator can solve this problem by using account synchronize option. He/she synchronizes either adding new system login information to the database or deleting new login from the system.

Some students change their majors after they obtain network accounts. When they want their login groups to be changed, the systems administrator moves their home directories and changes the group IDs to new ones. However, their group information in the database is not changed. User group check option is used when these group changes happen. By using this function, WebSAT detects all the logins whose group information is not the same as the group information in the database. By clicking on these logins, WebSAT replaces wrong group information in the database with the proper group information.

The database information for last login dates are old information, so the user of WebSAT is required to update these information by using update last login information function. This function gathers the last login information of accounts from users' home directory, and updates the WebSAT database.

1.2.2.7 Manage Disk Space. One major task of the systems administrator is to manage network disk space. WebSAT disk manager function observes total disk space, disk spaces by user group, and individual disk space. This function also detects a user's disk spaces over the disk space quota given by the systems administrator. If the systems administrator decides to take action, he/she can send them email or disable their accounts.

1.2.2.8 Scan Ports. The Port Scan Alert System (PSAS) watches if there is any port that is being scanned on the NIS server and NFS server. If somebody is scanning ports on these servers, PSAS will detect it and alerts the systems administrator to take action immediately.

1.2.3 User Characteristics

Since WebSAT is a systems administration tool, the intended user is the systems administrator of the Department of Computer Science. The access to WebSAT system by other persons who are not at the systems administrator level must be prevented for security reasons. The systems administrator must have fluent knowledge of NIS, NFS, and Secure Web Server to maintain the system. Knowledge of MySQL is extremely helpful for the user of WebSAT.

1.3 Software Specific Requirements

1.3.1 External Interface

This section describes all detailed inputs and outputs of WebSAT.

1.3.1.1 Login. This is the first page of WebSAT. See Figure 1.3. All other pages of WebSAT are accessed through this login page. A user provides his/her login ID and password to access WebSAT. When the user provides the correct login ID and password, this page authorizes the user and creates a new session.

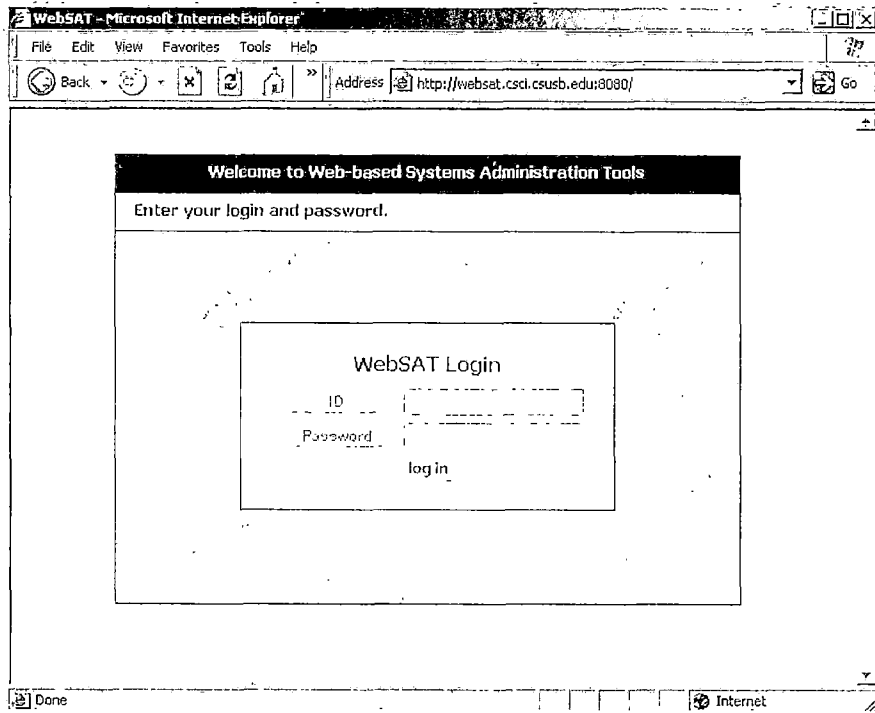


Figure 1.3. Login Page

1.3.1.2 Home. This page is the home of WebSAT (see Figure 1.4). This page provides find account, create account, enable account, disable account, delete account, synchronize account, disk manager, and logout menu.

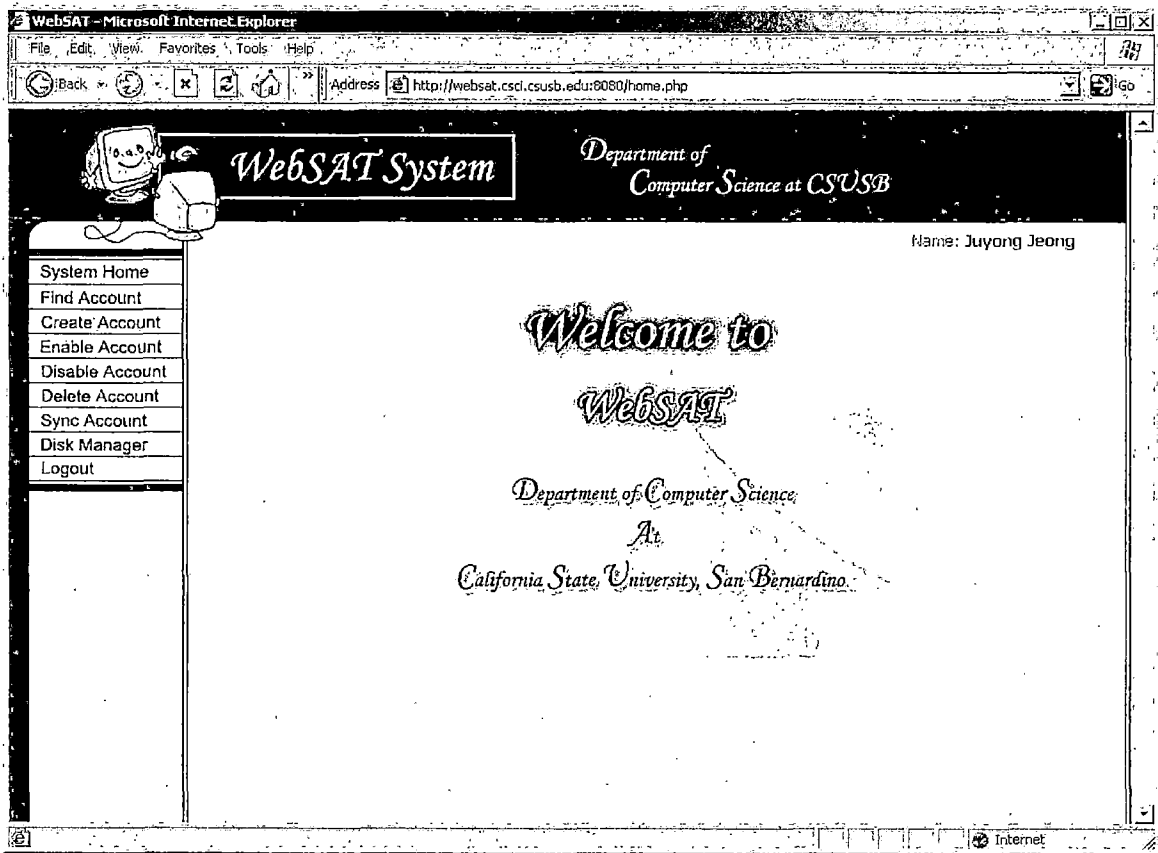


Figure 1.4. Home Page

1.3.1.3 Find Account. This page shows an information input form that the user has to fill out to find accounts (see Figure 1.5). A user can fill out logins, student ID, last names, first names, created date, disabled date, or last log-in date fields in order to search for an account information.

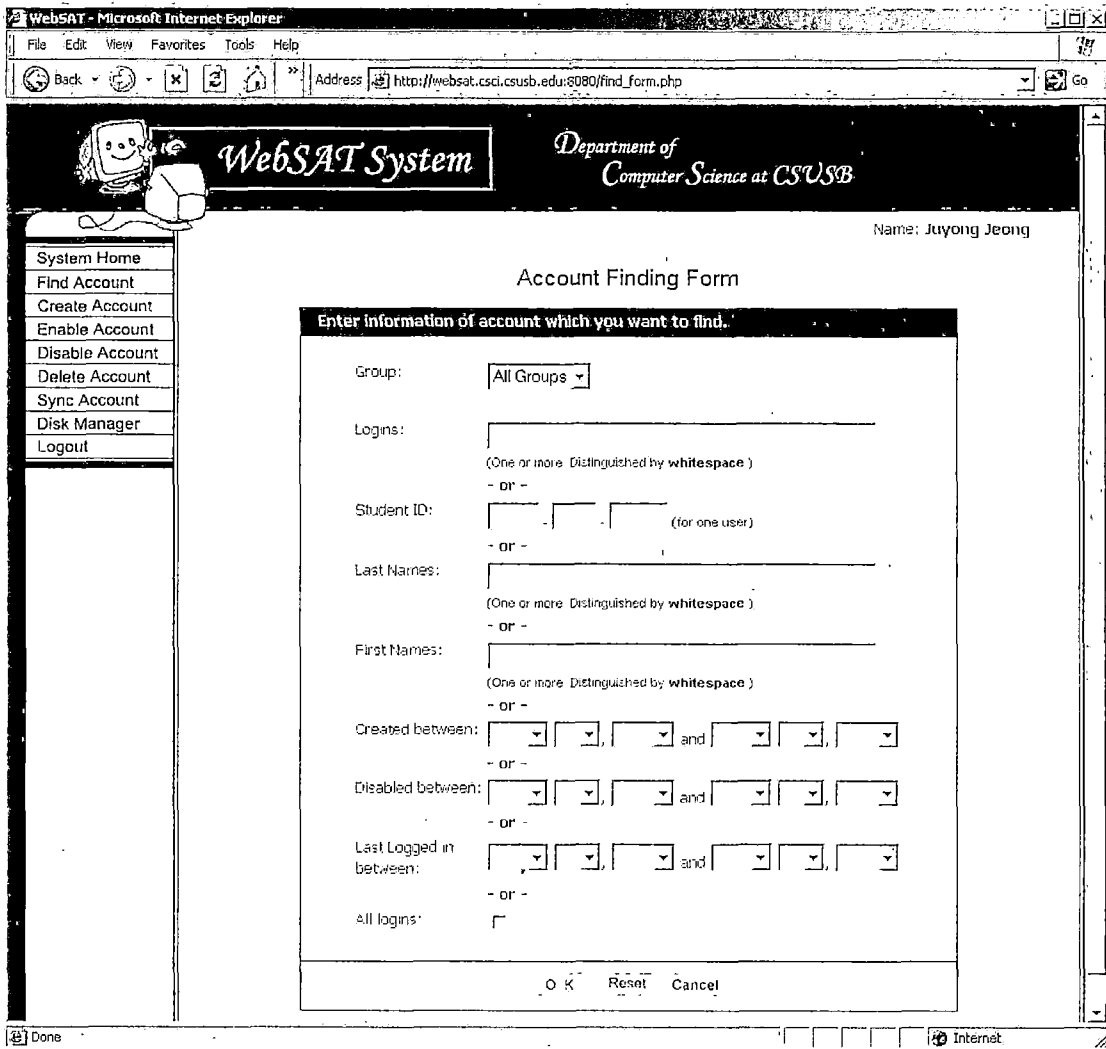


Figure 1.5. Find Account Form

1.3.1.4 Create Account Menu. This page has menus for account creation: create account for student (grad, CSCI, and student group account) and create account for non-student (faculty, staff, friends, and generic group account). See Figure 1.6.

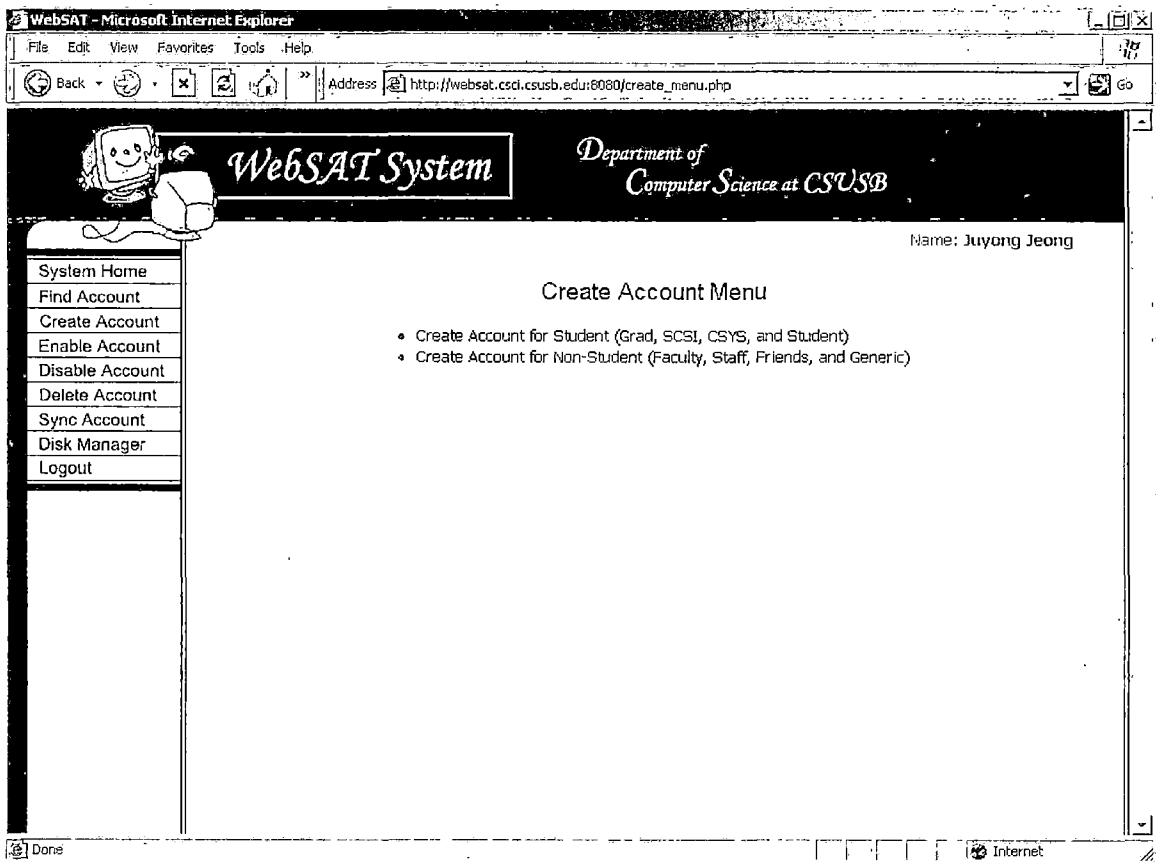


Figure 1.6. Create Account Menu Page

1.3.1.5 Account Creation Form for Student. This page provides two input forms for account creation (see Figure 1.7). First form is for single account creation. WebSAT creates an account if the user inserts all required information. Second form is for multiple accounts creation. The user copies a class roster from SIS+ and pastes it to the form to create accounts for a class.

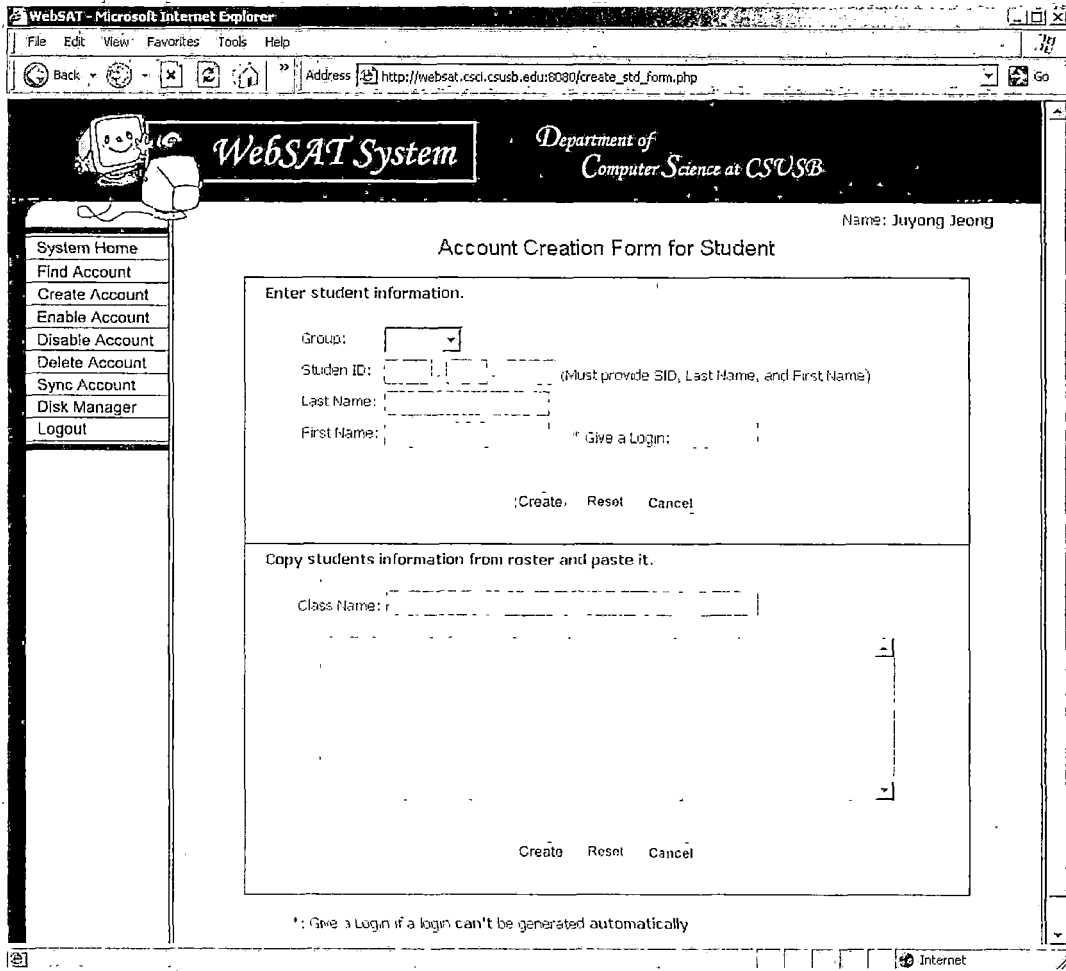


Figure 1.7. Account Creation Form for Student

1.3.1.6 Account Creation Form for Non-Student. This page provides an input form for a non-student account creation (see Figure 1.8). The user selects a group to which the account will belong, account owner's first name, last name, login, and password.

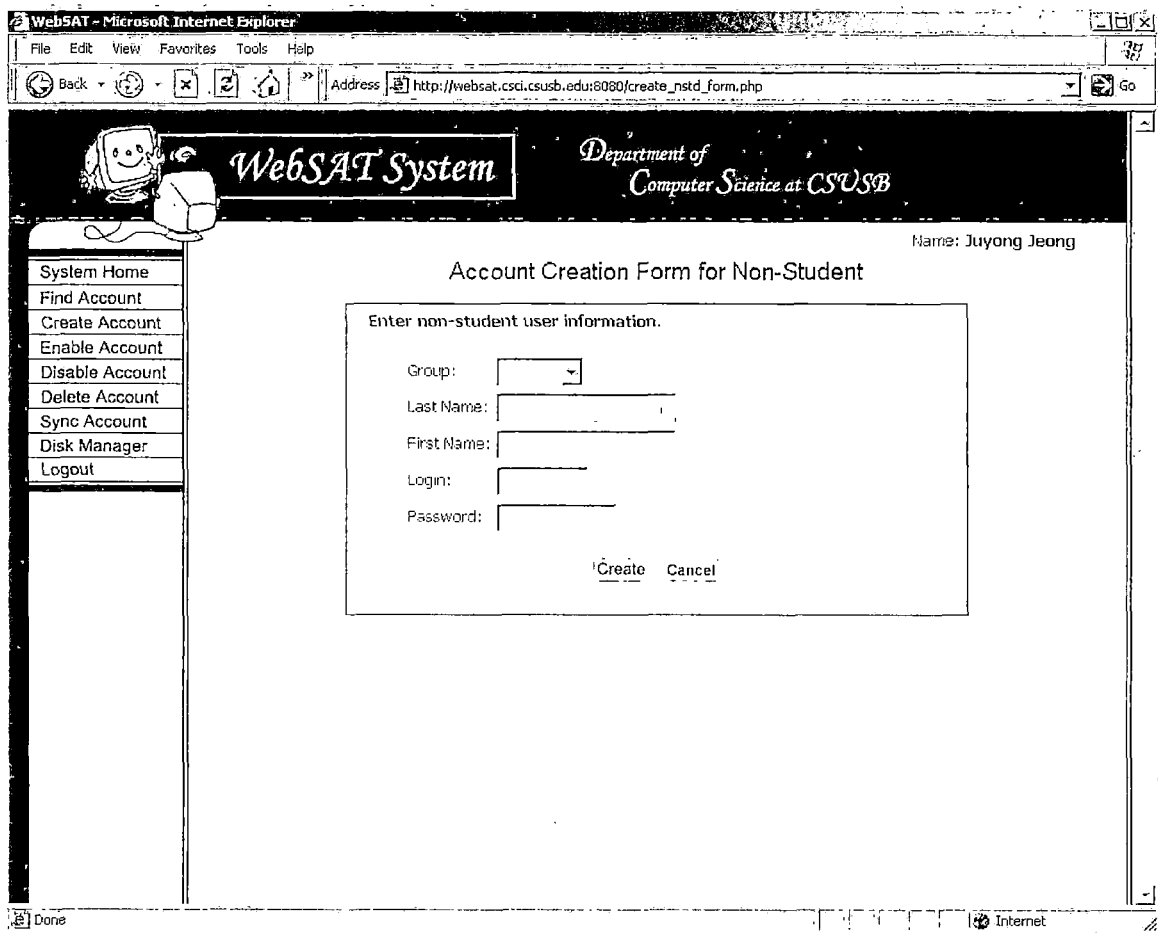


Figure 1.8. Account Creation Form for Non-Student

1.3.1.7 Enable Account Menu. This page has menus to enable accounts: enable account for student (student and generic) and enable account disabled by admin. See Figure 1.9.

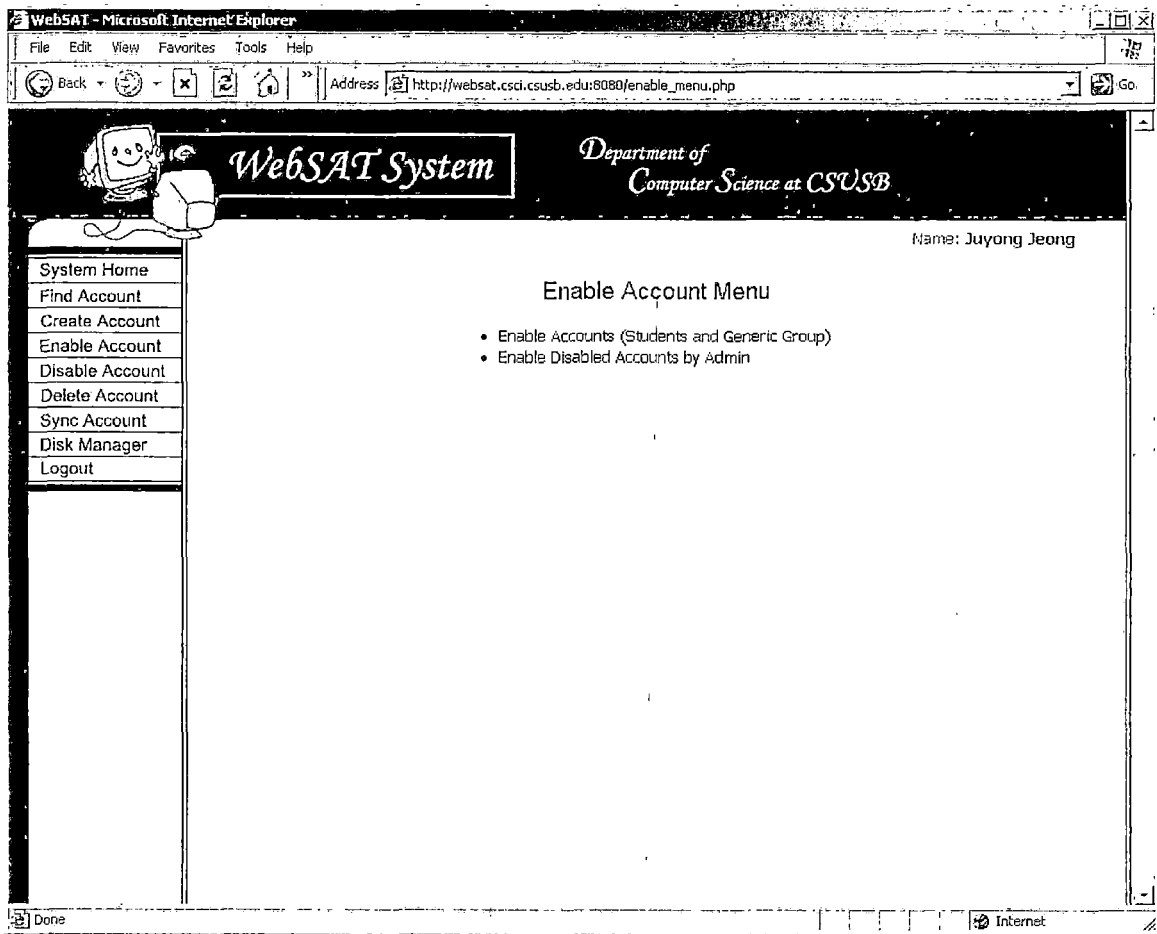


Figure 1.9. Enable Account Menu Page

1.3.1.8 Enable Account. This page shows a list of all disabled accounts which are in student group or in generic group (see Figure 1.10). The user can select accounts in the list to enable them.

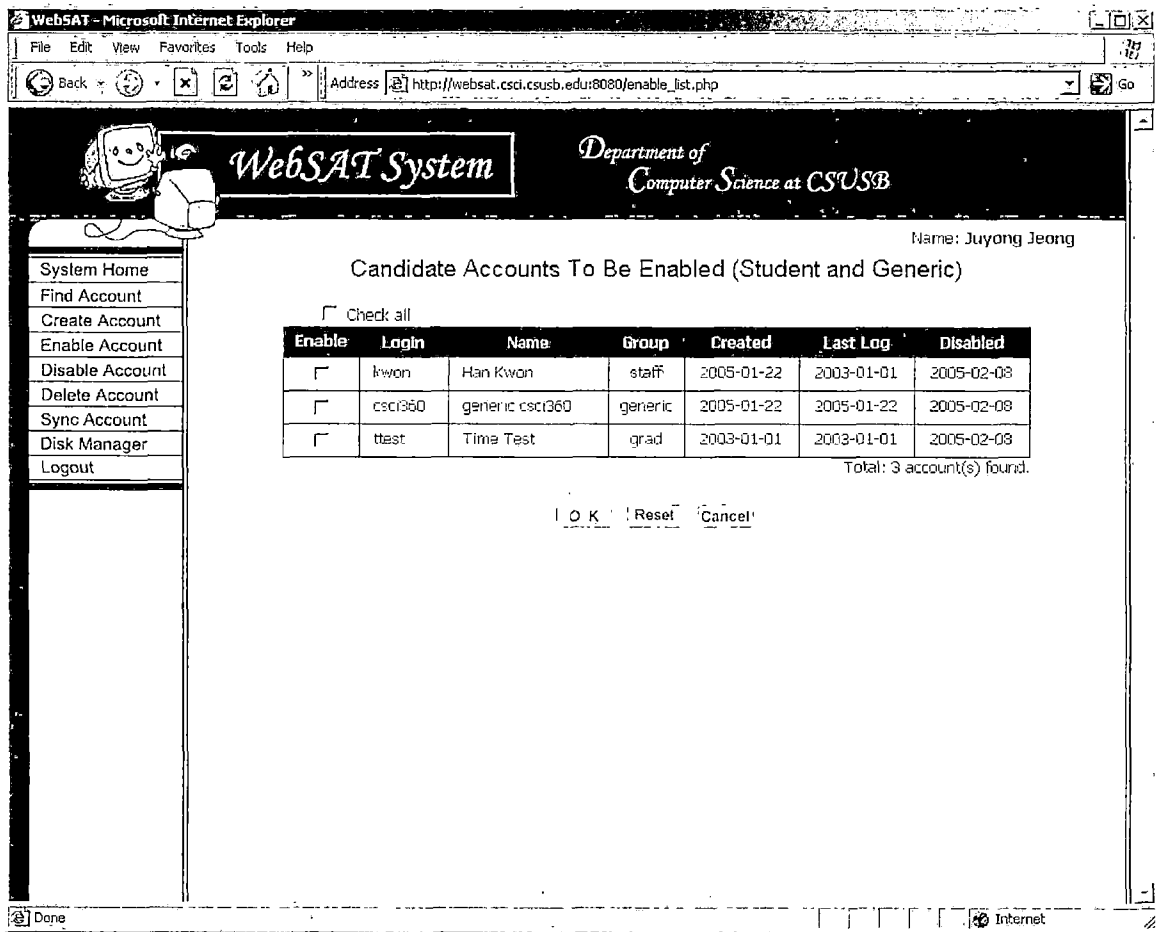


Figure 1.10. Enable Account List

1.3.1.9 Enable Account Disabled by Admin. This page shows a list of all disabled accounts by the systems administrator (see Figure 1.11). The user can select accounts in the list to enable them.

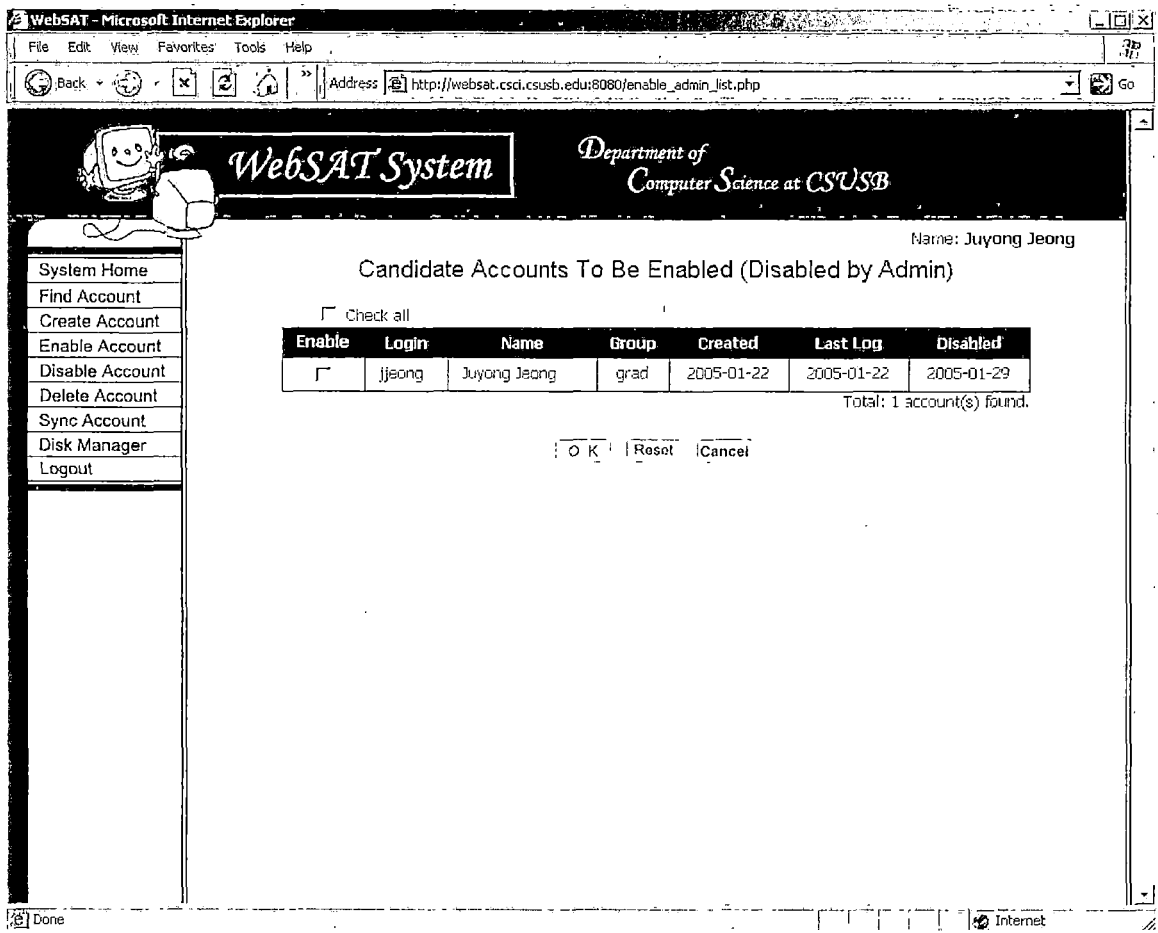


Figure 1.11. List of Disabled Account by Admin

1.3.1.10 Disable Account Menu. This page has menus for disable accounts: disable expired accounts, disable not accessed accounts, and disable accounts by admin. See Figure 1.12.

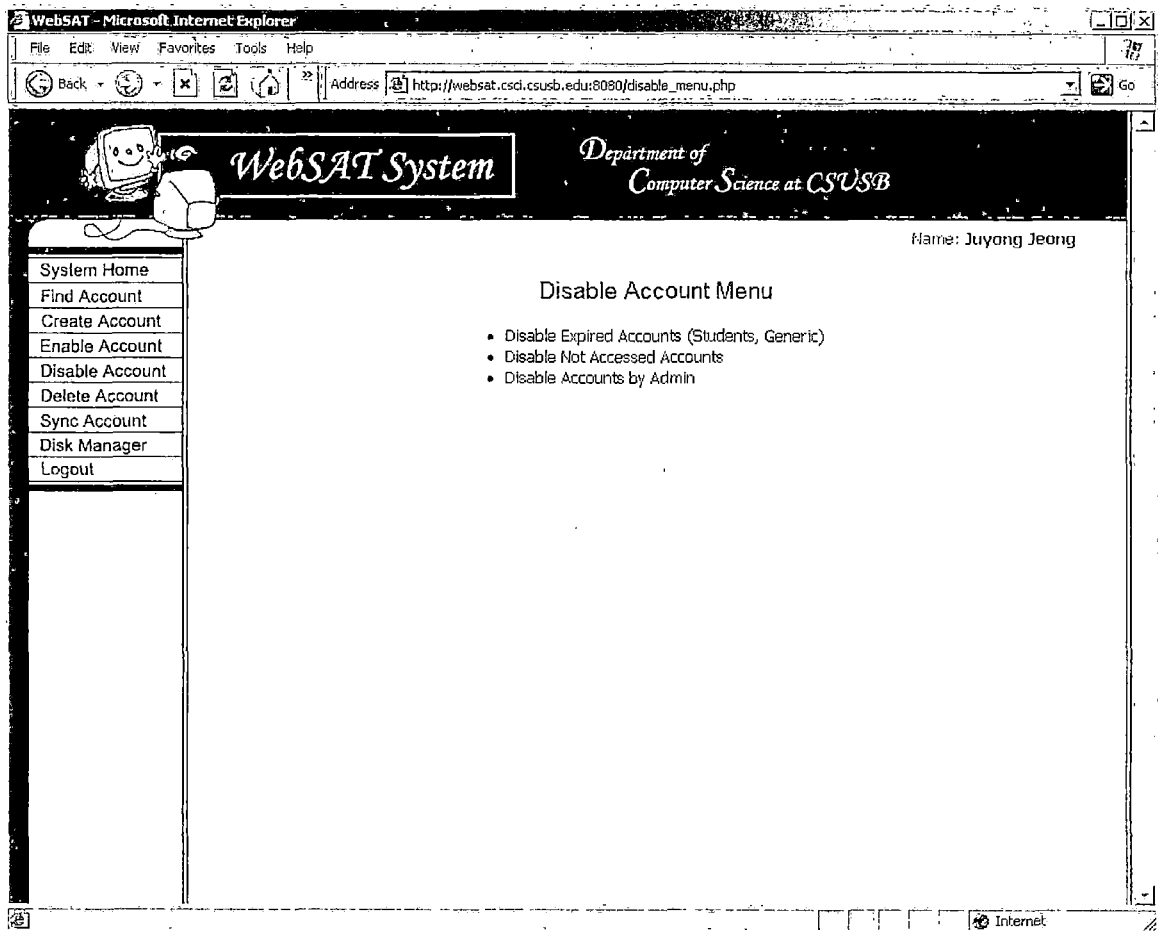


Figure 1.12. Disable Account Menu Page

1.3.1.11 Disable Expired Account. This page shows a list of student and generic group accounts (see Figure 1.13). All student and generic group accounts are expired after the current quarter ended. The user can select accounts from the list to disable them.

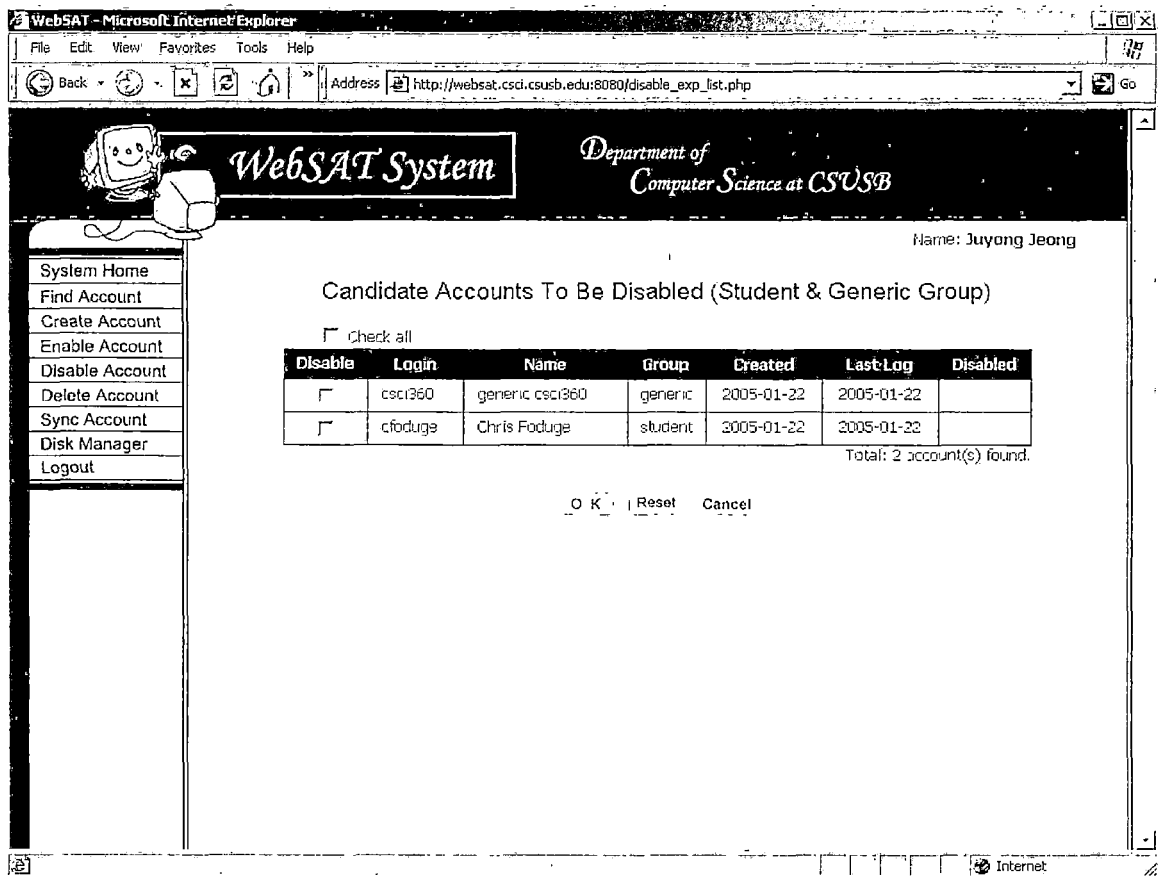


Figure 1.13. Disable Expired Account

1.3.1.12 Disable Not Accessed Account. This page shows a list of accounts which has not been accessed over one year (see Figure 1.14).

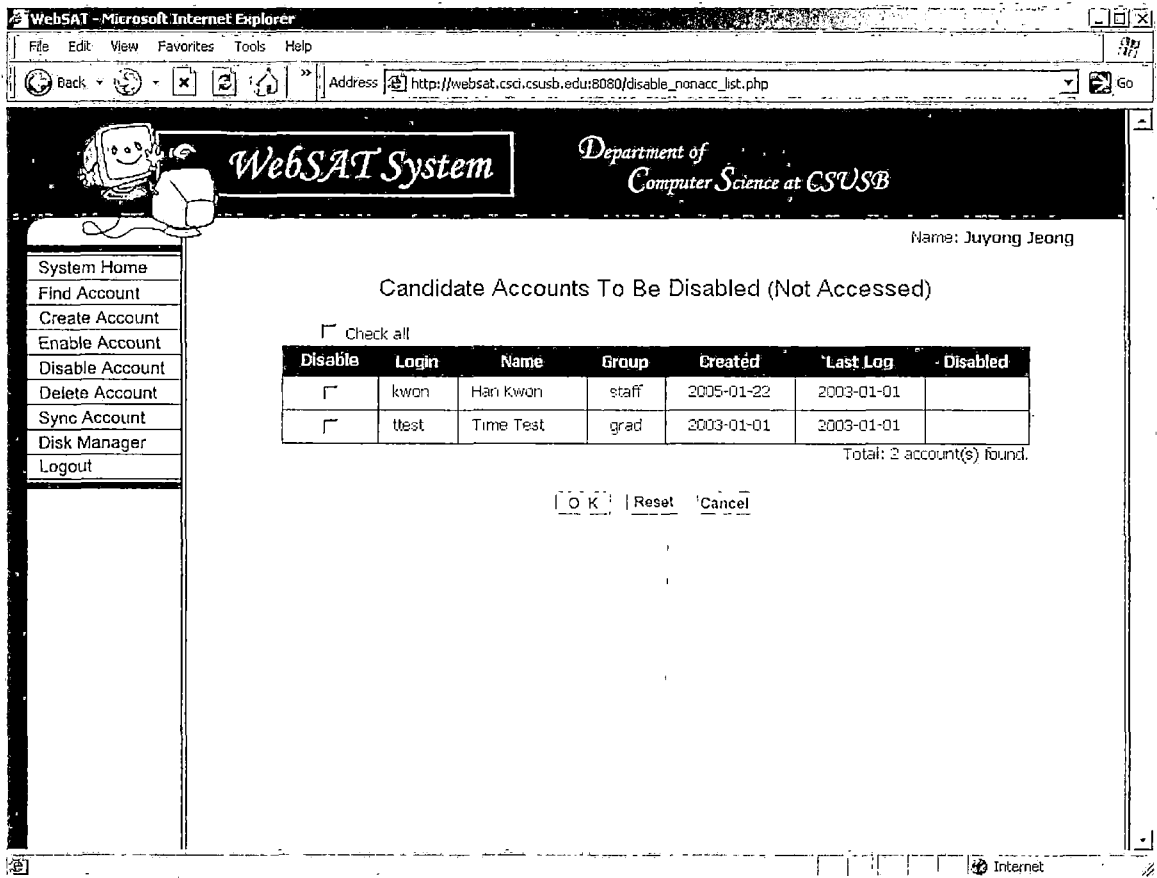


Figure 1.14. Disable Not Accessed Account

1.3.1.13 Disable Account by Admin. This page shows an information input form that the user has to fill out to find accounts which may be disabled (see Figure 1.15). The user can fill out logins, student ID, last names, first names, created date, disabled date, or last log-in date field in order to search for account information.

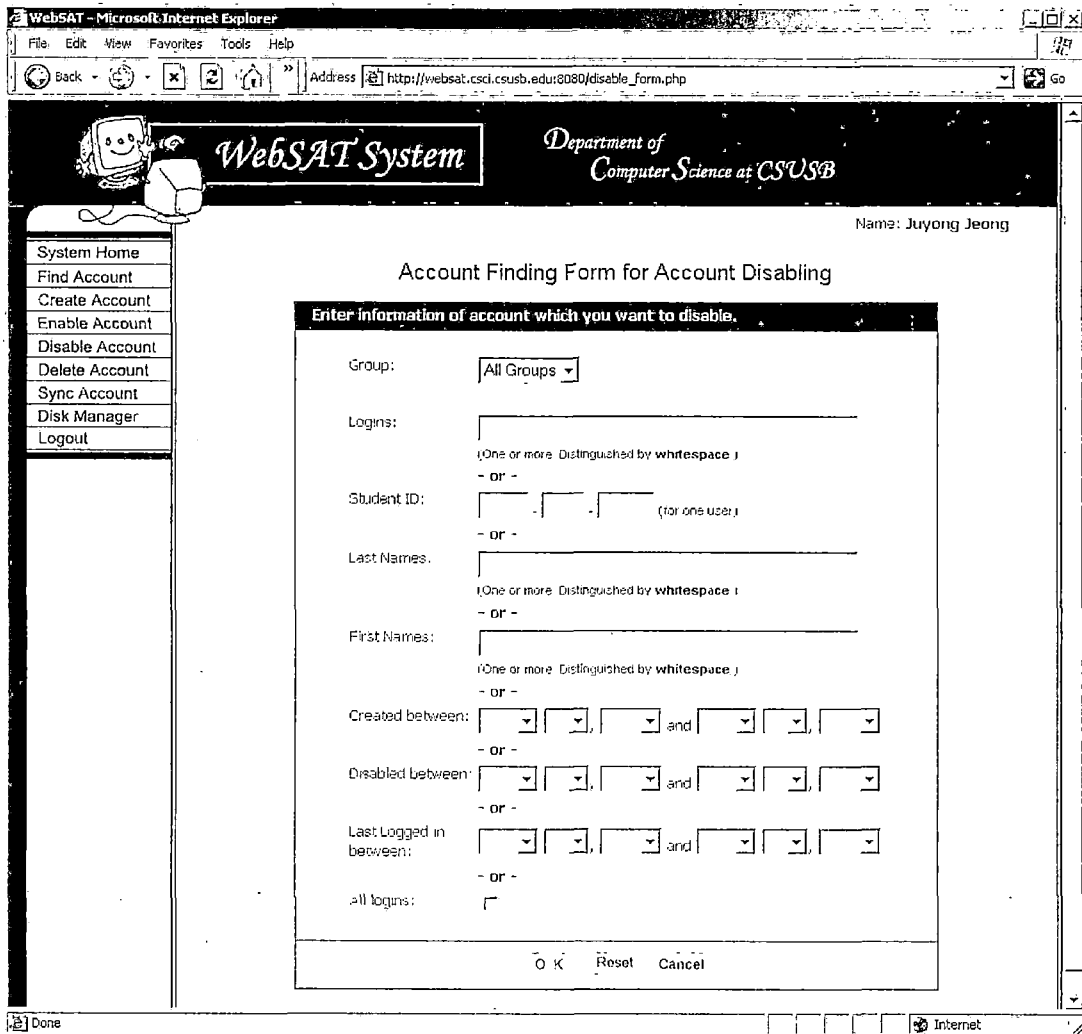


Figure 1.15. Disable Account by Admin

1.3.1.14 Candidate Accounts to Be Disabled. This page shows a list of accounts which correspond to the information the user provided (see Figure 1.16). The user can select accounts from the list to disable them. Those selected accounts will be marked as disabled by admin.

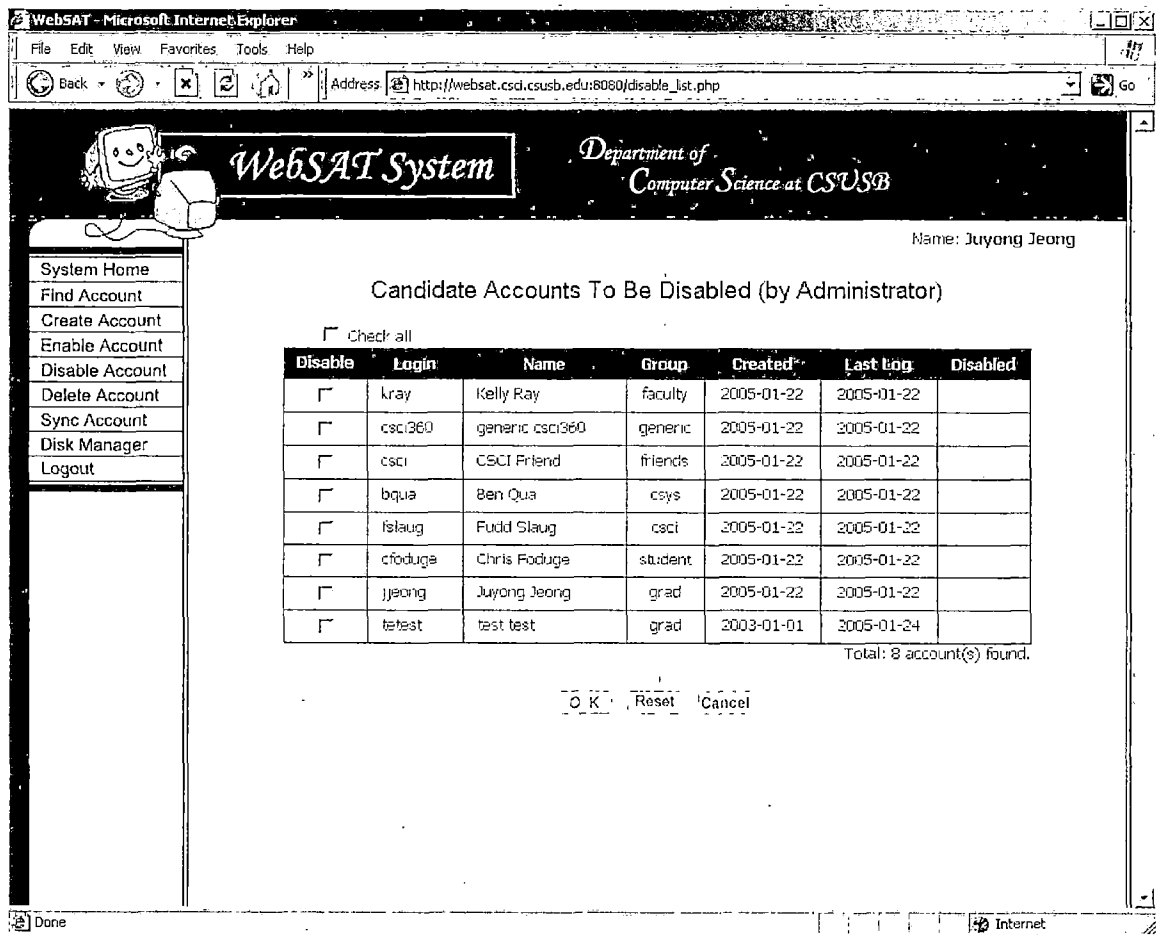


Figure 1.16. Candidate Accounts to Be Disabled

1.3.1.15 Delete Account Menu. This page has menus for delete accounts: delete student group accounts, delete generic group accounts, and delete general accounts. See Figure 1.17.

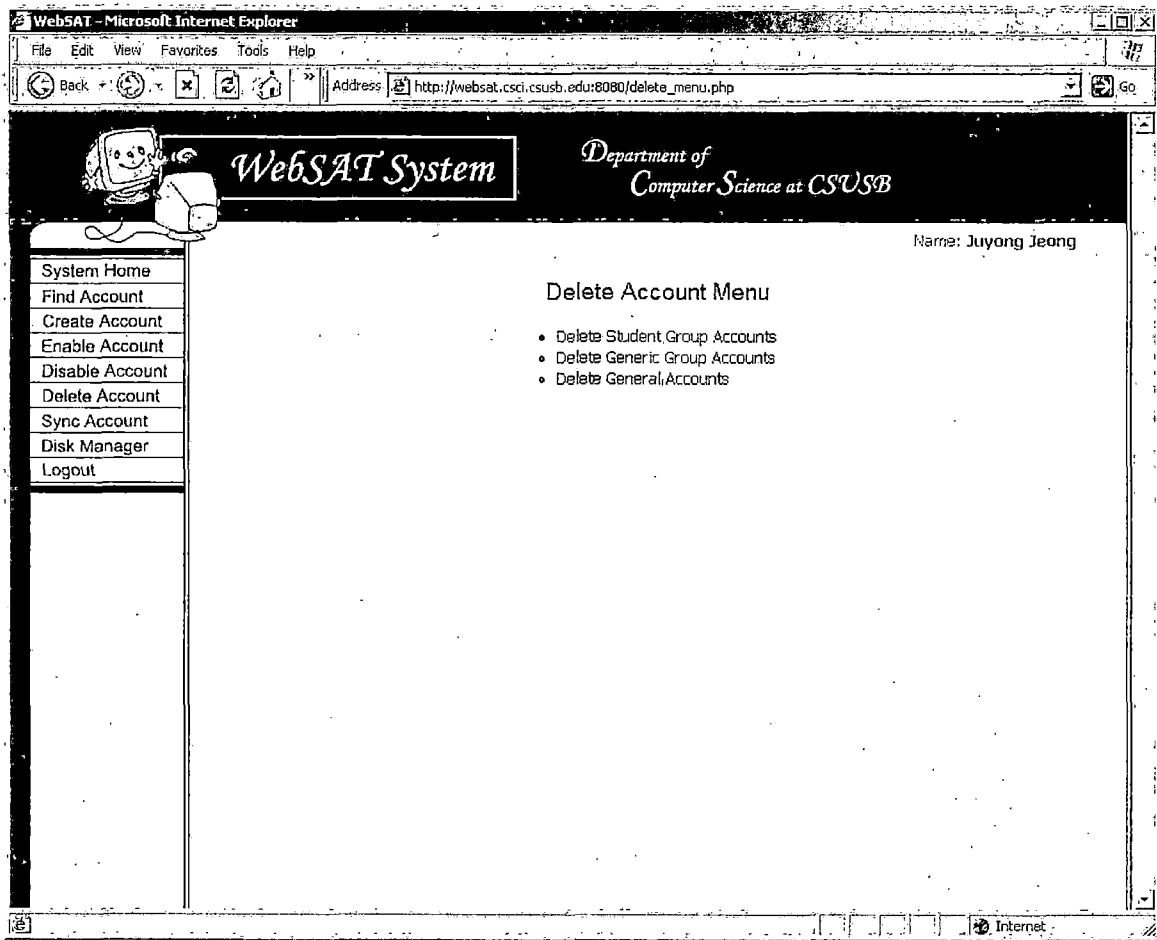


Figure 1.17: Delete Account Menu Page

1.3.1.16 Delete Student Group Accounts. This page shows a list of student group accounts (see Figure 1.18). All student group accounts are expired after the current quarter ended and needed to be deleted to save network resources.

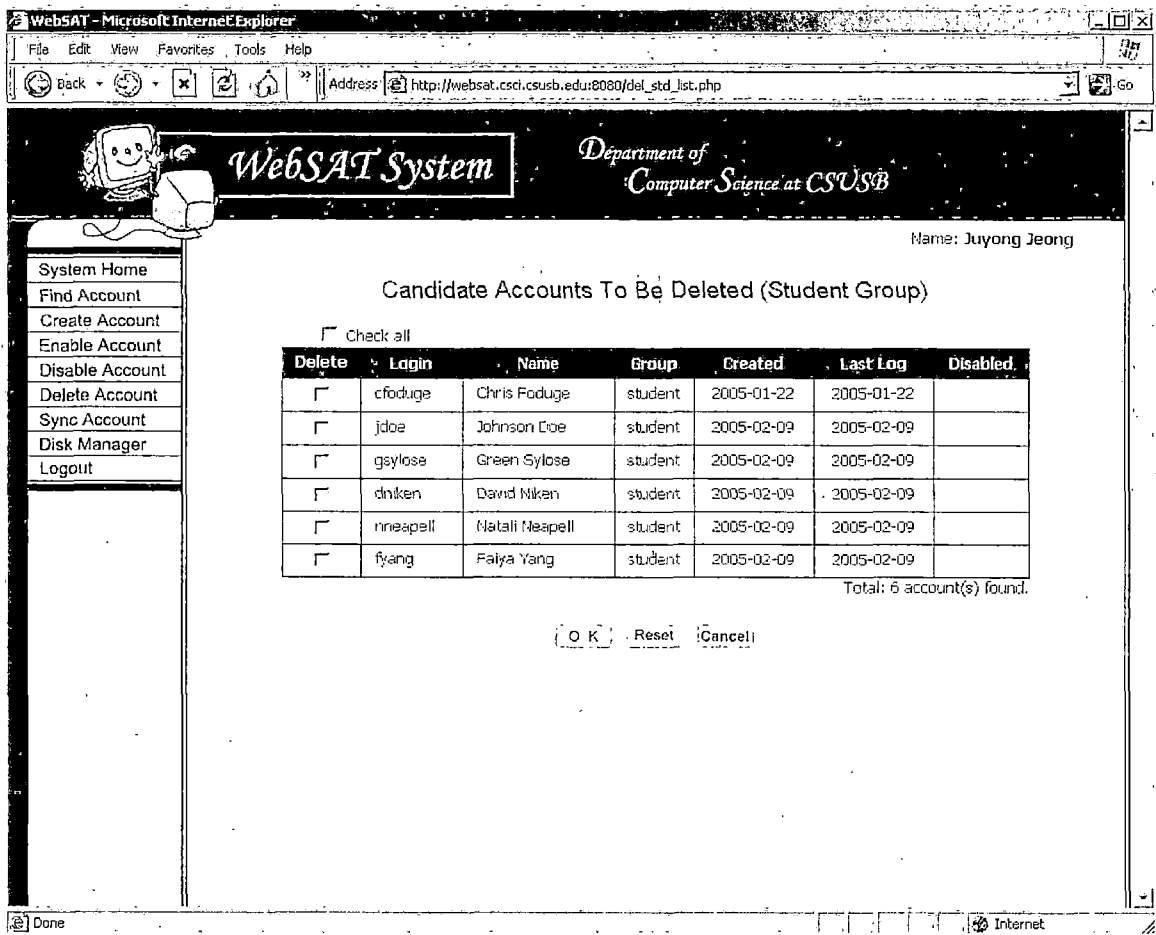


Figure 1.18. Delete Student Group Accounts

1.3.1.17 Delete Generic Group Accounts. This page shows a list of generic group accounts (see Figure 1.19). All generic group accounts are expired after the current quarter ended and needed to be deleted to save network resources.

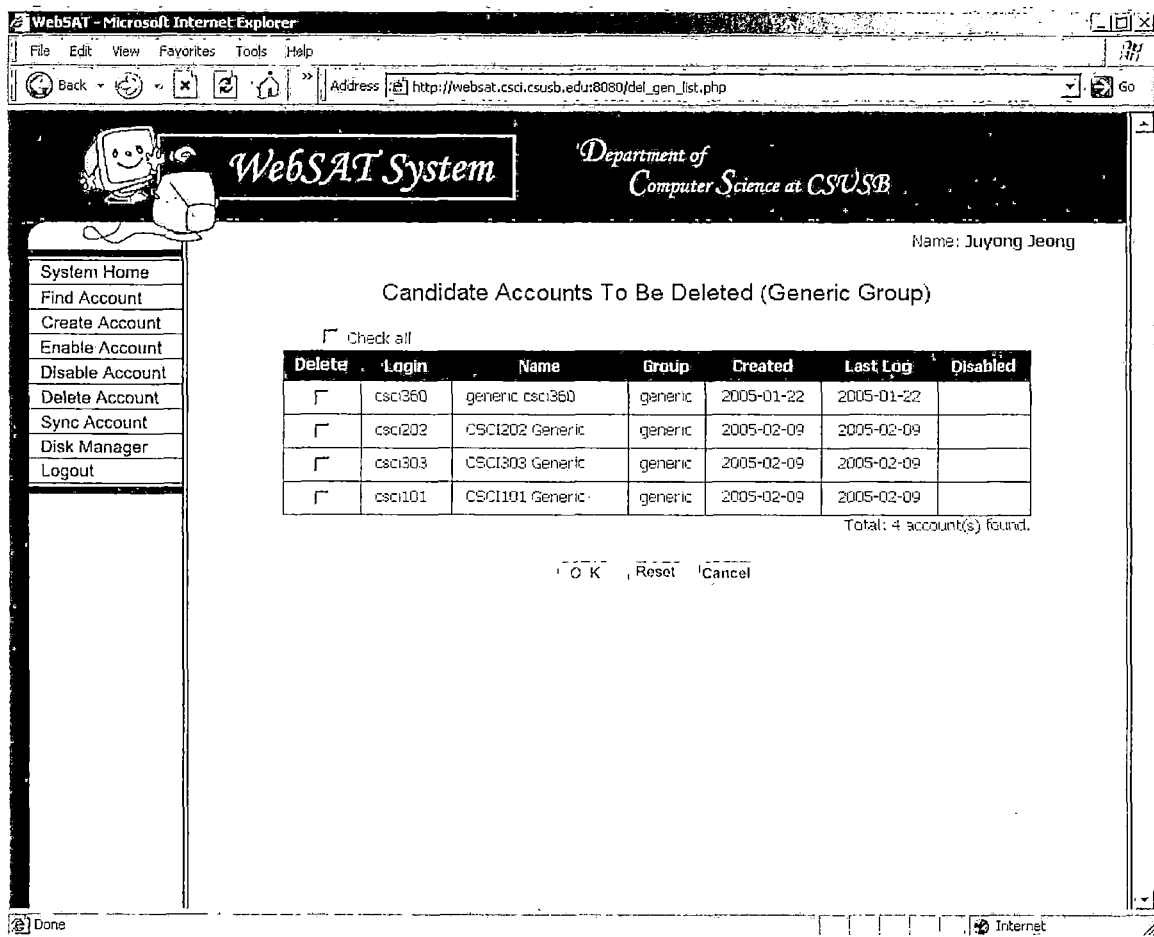


Figure 1.19. Delete Generic Group Accounts

1.3.1.18 Account Deletion Form. This page shows an information input form that the user has to fill out to find accounts which may be deleted (see Figure 1.20). The user can fill out logins, student ID, last names, first names, created date, disabled date, or last log-in date field in order to search for account information.

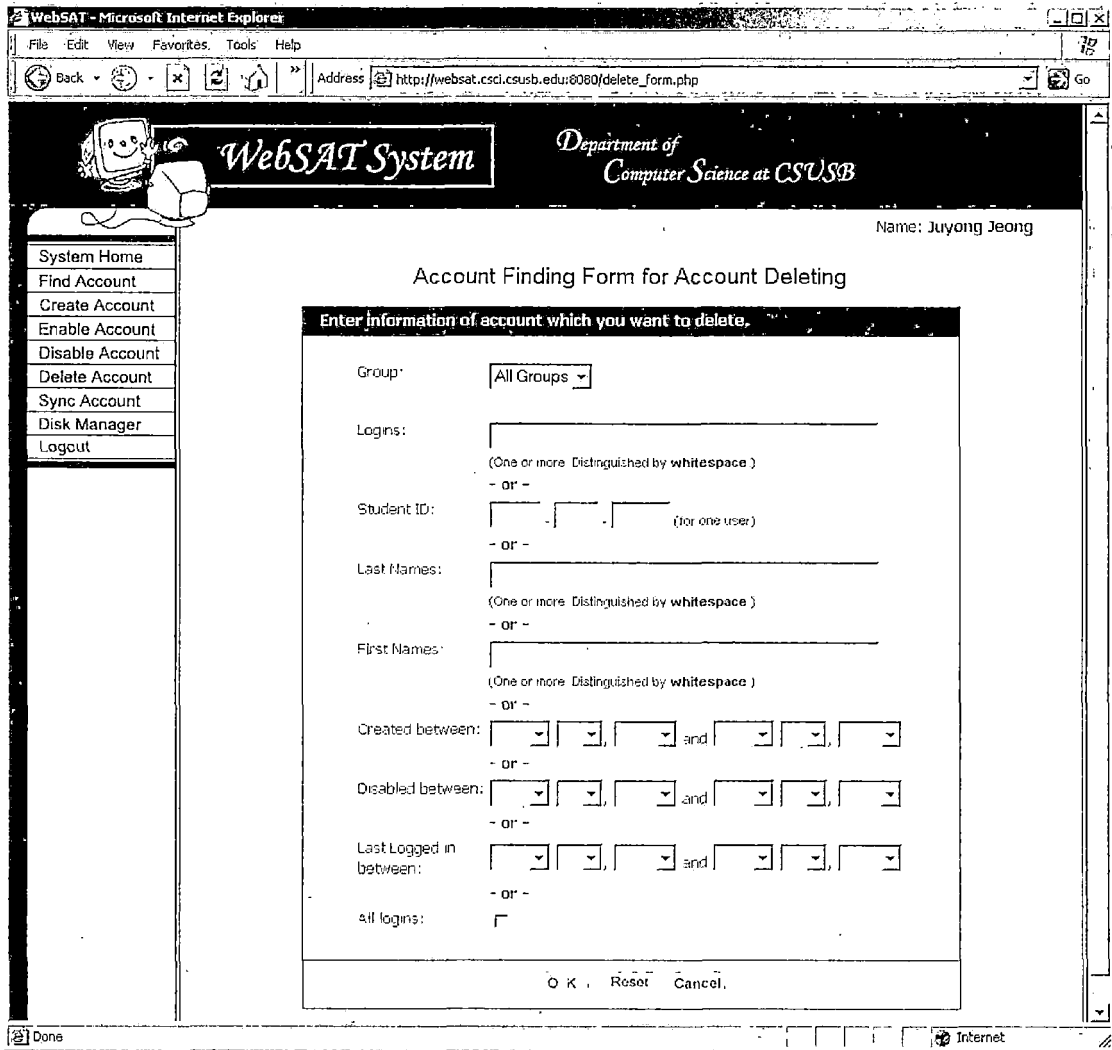


Figure 1.20. Account Deletion Form

1.3.1.19 Candidate Accounts to Be Deleted. This page shows a list of accounts which correspond to the information the user provided (see Figure 1.21). The user can select accounts from the list to delete them. Selected account information will be deleted from the system and database.

WebSAT System Department of Computer Science at CSUSB

Name: Juyong Jeong

Candidate Accounts To Be Deleted

Check all

Delete	Login	Name	Group	Created	Last Log	Disabled
<input type="checkbox"/>	kray	Kelly Ray	faculty	2005-01-22	2005-01-22	
<input type="checkbox"/>	csci360	generic csci360	generic	2005-01-22	2005-01-22	
<input type="checkbox"/>	csci	CSCI Friend	friends	2005-01-22	2005-01-22	
<input type="checkbox"/>	bqua	Ben Qua	csys	2005-01-22	2005-01-22	
<input type="checkbox"/>	fslaug	Fudd Slaug	csci	2005-01-22	2005-01-22	
<input type="checkbox"/>	cfoduge	Chris Foduge	student	2005-01-22	2005-01-22	
<input type="checkbox"/>	jjeong	Juyong Jeong	grad	2005-01-22	2005-01-22	
<input type="checkbox"/>	testest	test test	grad	2003-01-01	2005-01-24	
<input type="checkbox"/>	jdoe	Johnson Doe	student	2005-02-09	2005-02-09	
<input type="checkbox"/>	gsylose	Green Sylose	student	2005-02-09	2005-02-09	
<input type="checkbox"/>	dniken	David Niken	student	2005-02-09	2005-02-09	
<input type="checkbox"/>	nneapell	Natali Neapell	student	2005-02-09	2005-02-09	
<input type="checkbox"/>	fyang	Faiya Yang	student	2005-02-09	2005-02-09	
<input type="checkbox"/>	csci202	CSCI202 Generic	generic	2005-02-09	2005-02-09	
<input type="checkbox"/>	csci203	CSCI203 Generic	generic	2005-02-09	2005-02-09	
<input type="checkbox"/>	csci101	CSCI101 Generic	generic	2005-02-09	2005-02-09	

Total: 16 account(s) found.

OK Reset Cancel

Figure 1.21. Candidate Accounts to Be Deleted

1.3.1.20 Synchronize System and Database Menu. This page has menus for system and database synchronization: check login information, check user group information, and update last login information. See Figure 1.22.

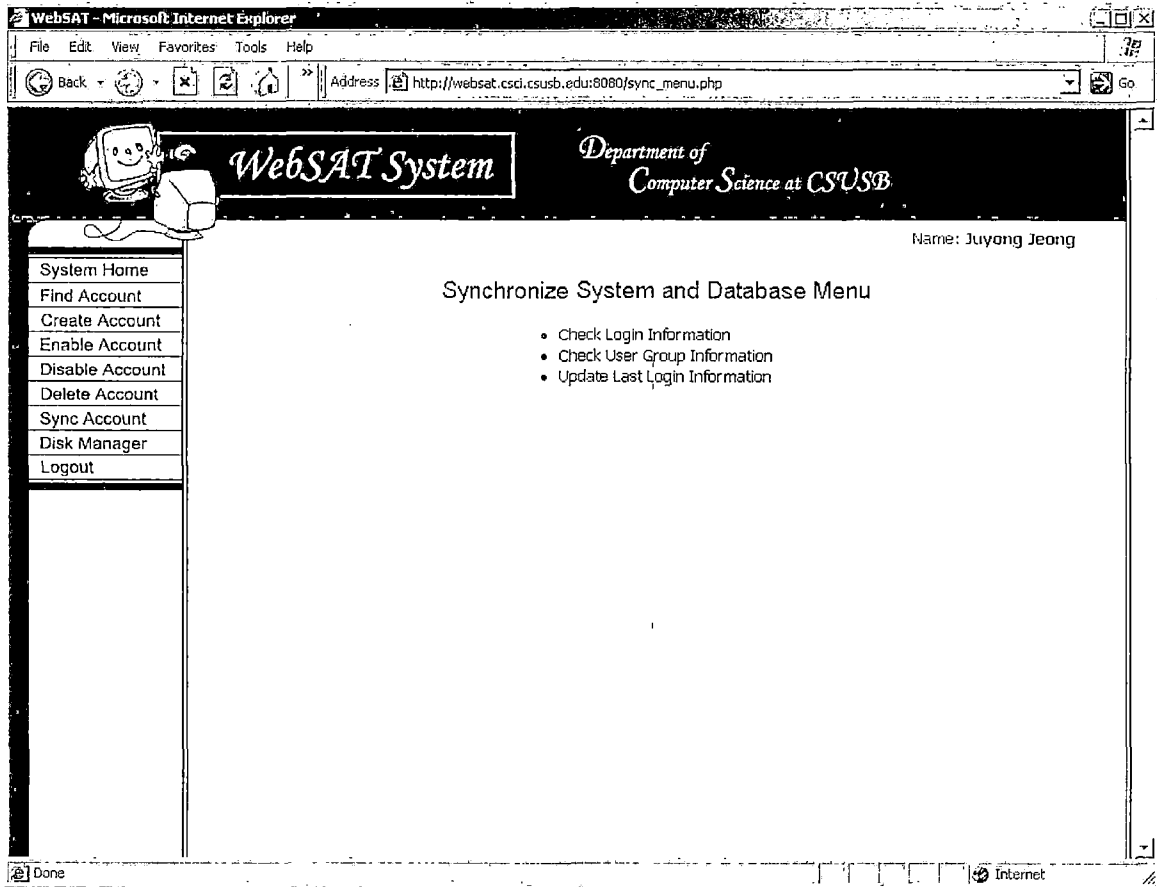


Figure 1.22. Synchronize System and Database Menu Page

1.3.1.21 Check Login Information. This page checks the login information in the system and in the database. Figure 1.23 shows the result displayed when the login information is synchronized, and Figure 1.24 shows the results displayed when the login information is not synchronized.

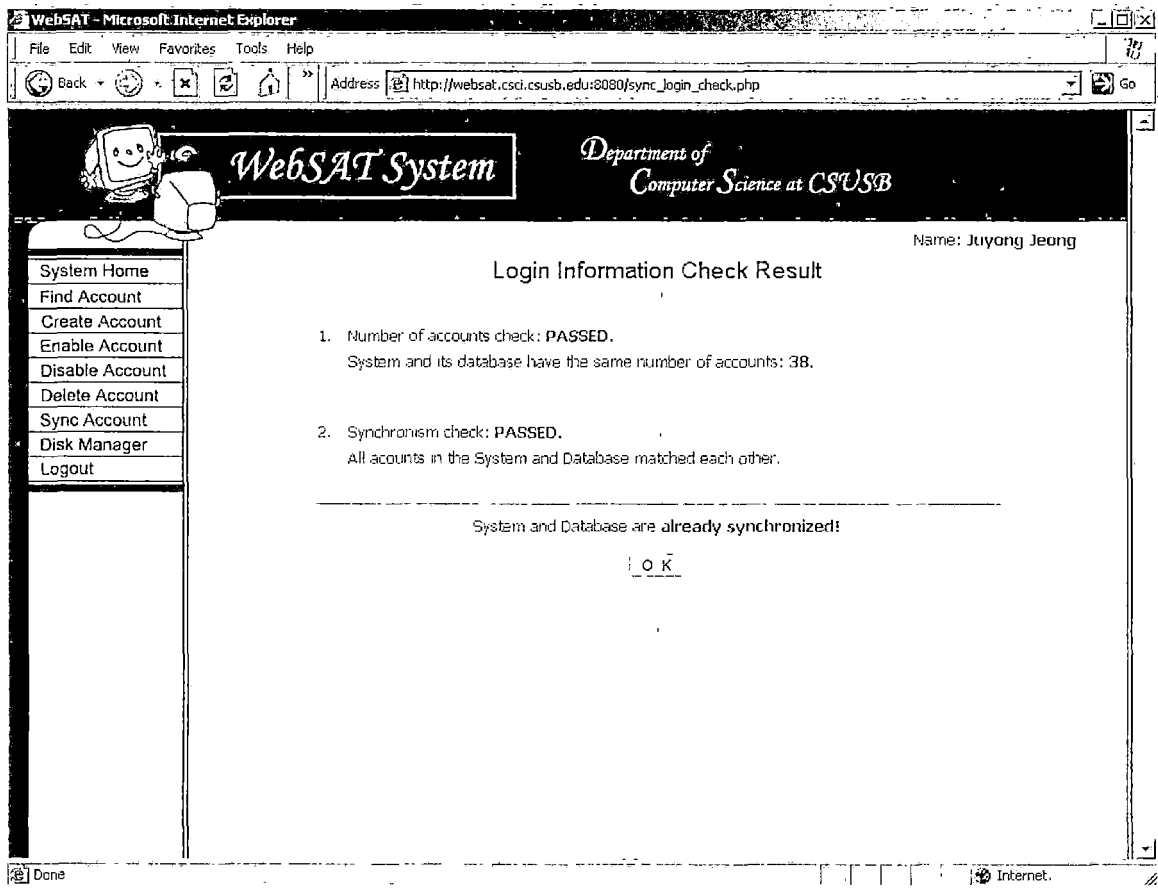


Figure 1.23. Check Login Information Page 1

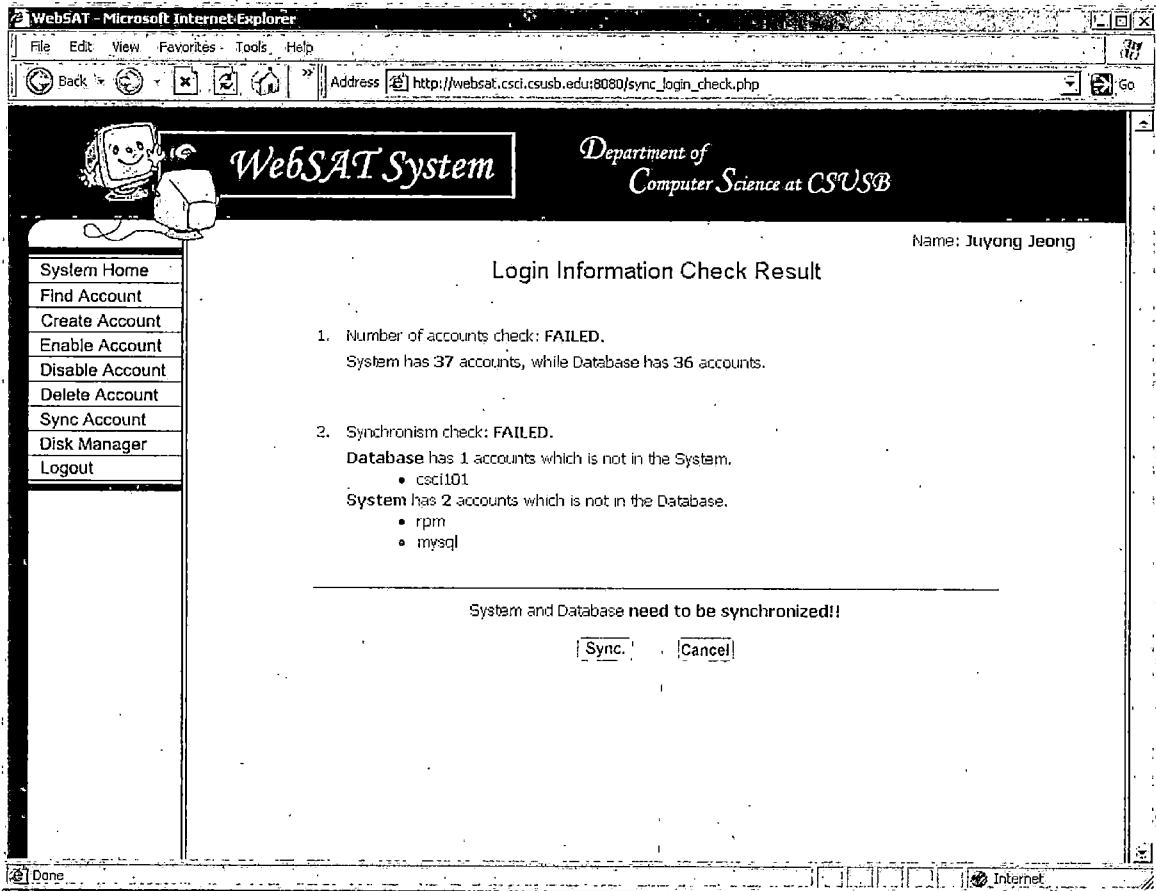


Figure 1.24. Check Login Information Page 2

1.3.1.22 Synchronize Login Information. This page shows account logins which are not in the system while these logins are in the database, or account logins which are not in the database while these logins are in the system. The user makes a decision whether to delete these accounts or insert their information in the database. See Figure 1.25.

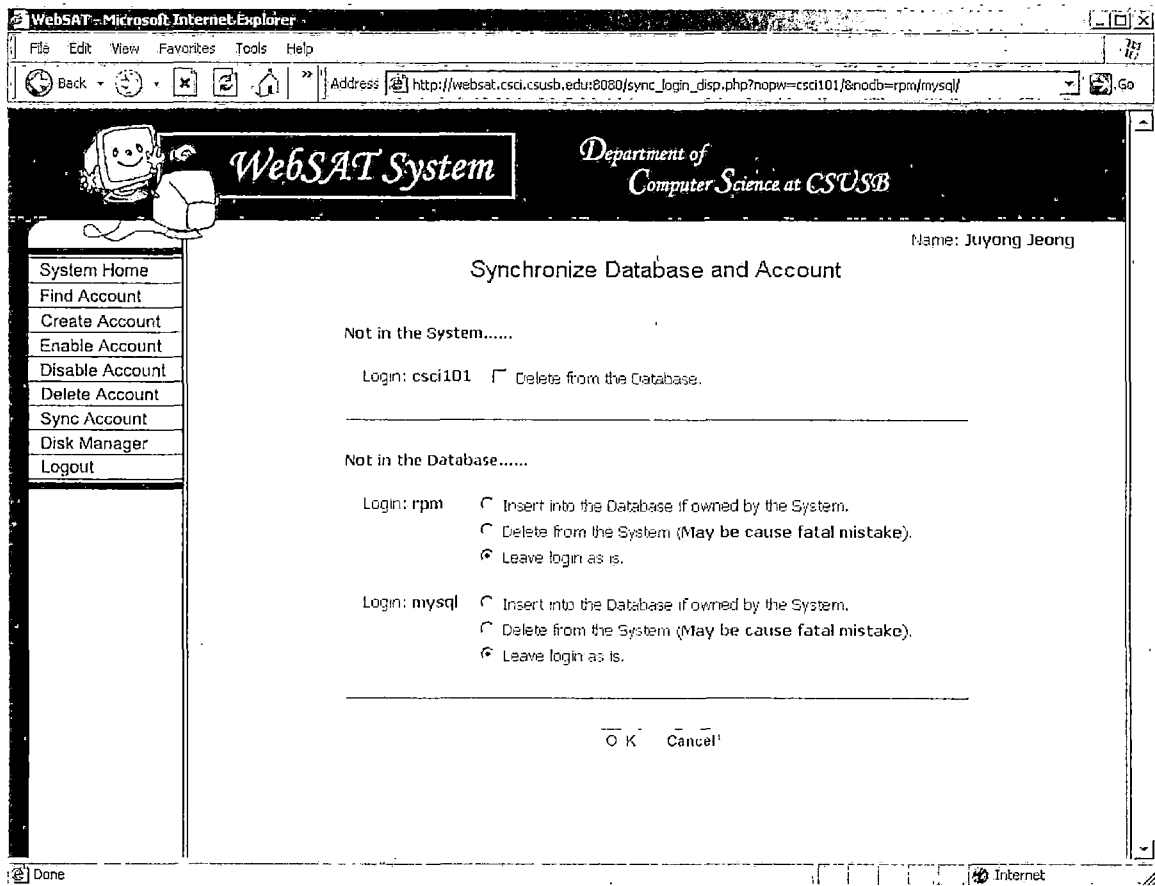


Figure 1.25. Synchronize Login Information Page

1.3.1.23 Check User Group Information. This page checks the user group information in the system and in the database. Figure 1.26 shows the result displayed when the user group information is synchronized, and Figure 1.27 shows the result displayed when the login information is not synchronized.

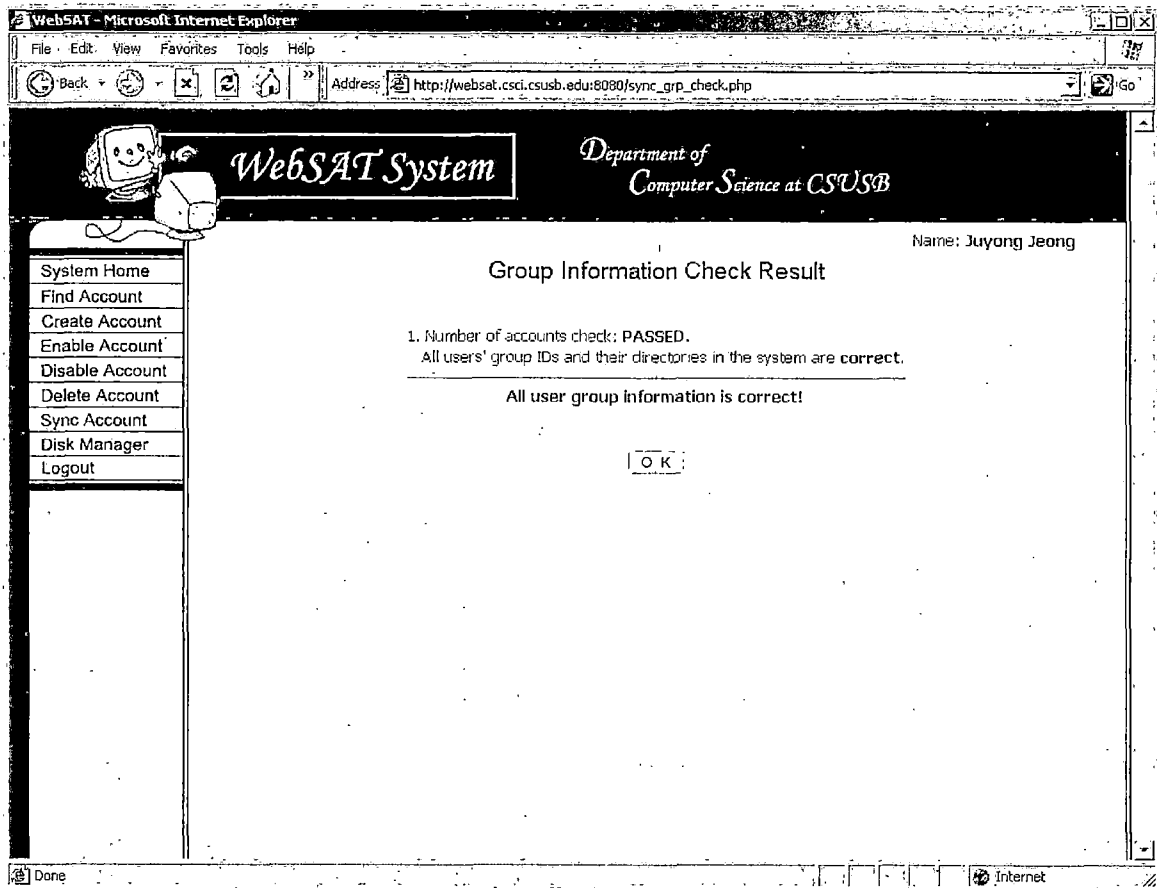


Figure 1.26. Check Group Information Page 1

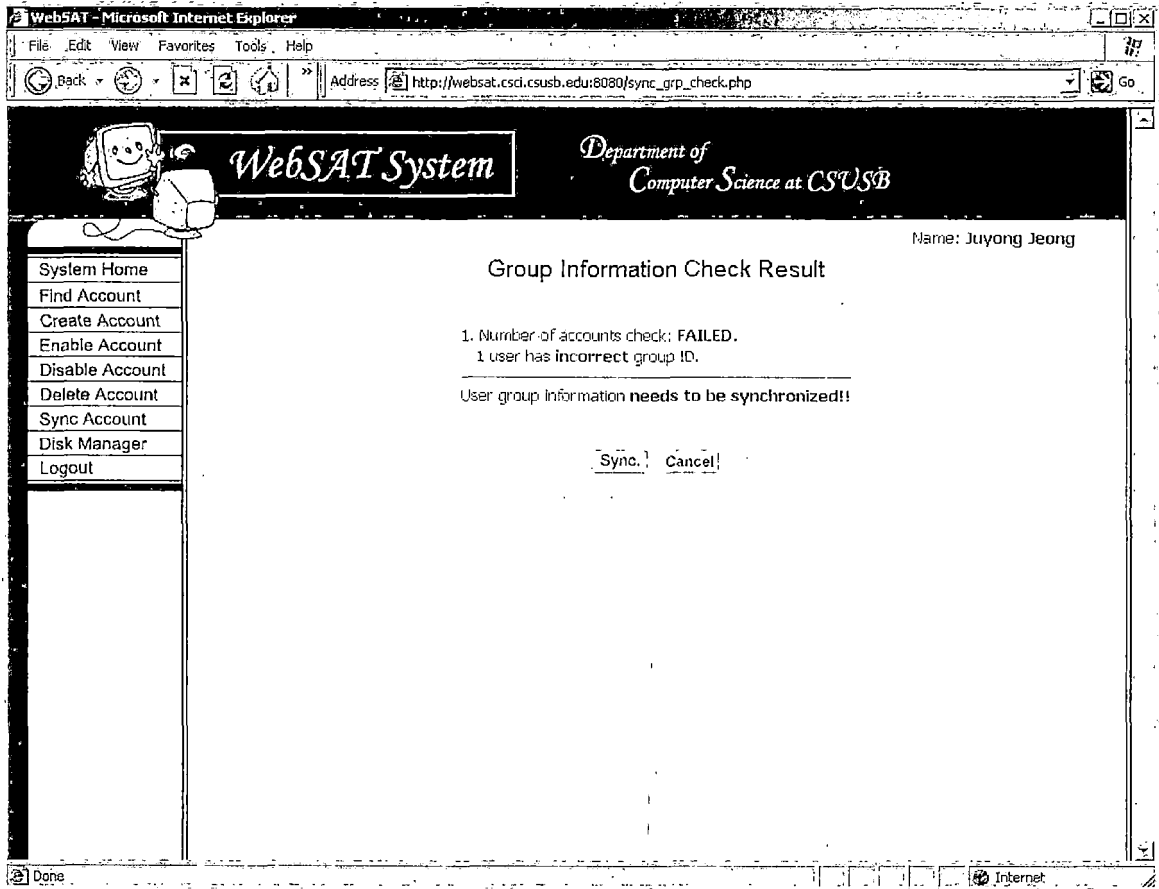


Figure 1.27. Check Group Information Page 2

1.3.1.24 Synchronize User Group Information. This page shows account logins whose information in the system is not the same as the information in the database (see Figure 1.28). The user makes a decision whether to change the group information of these accounts in the database.

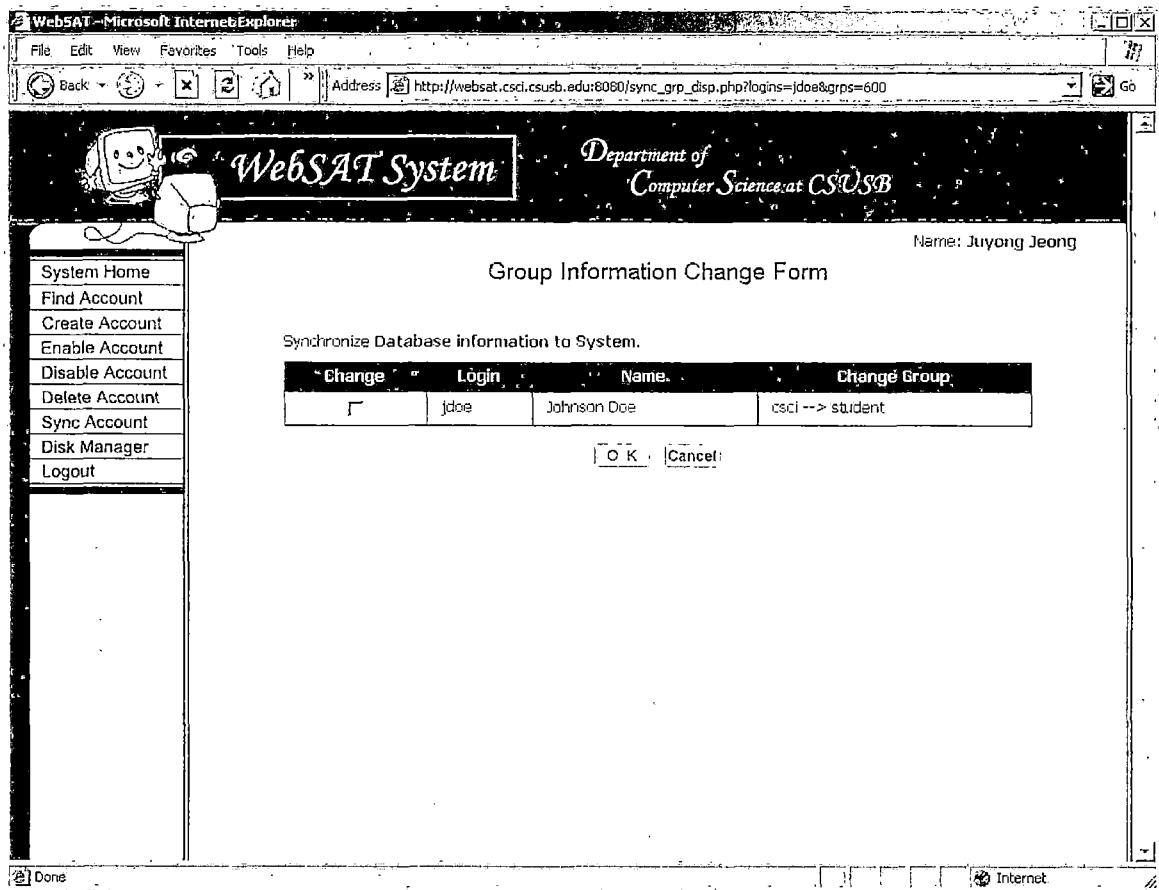


Figure 1.28. Synchronize User Group Information Page

1.3.1.25 Update Last Login Information. This page gets the last login information from the system and updates last login information in the database. Figure 1.29 displays the result.

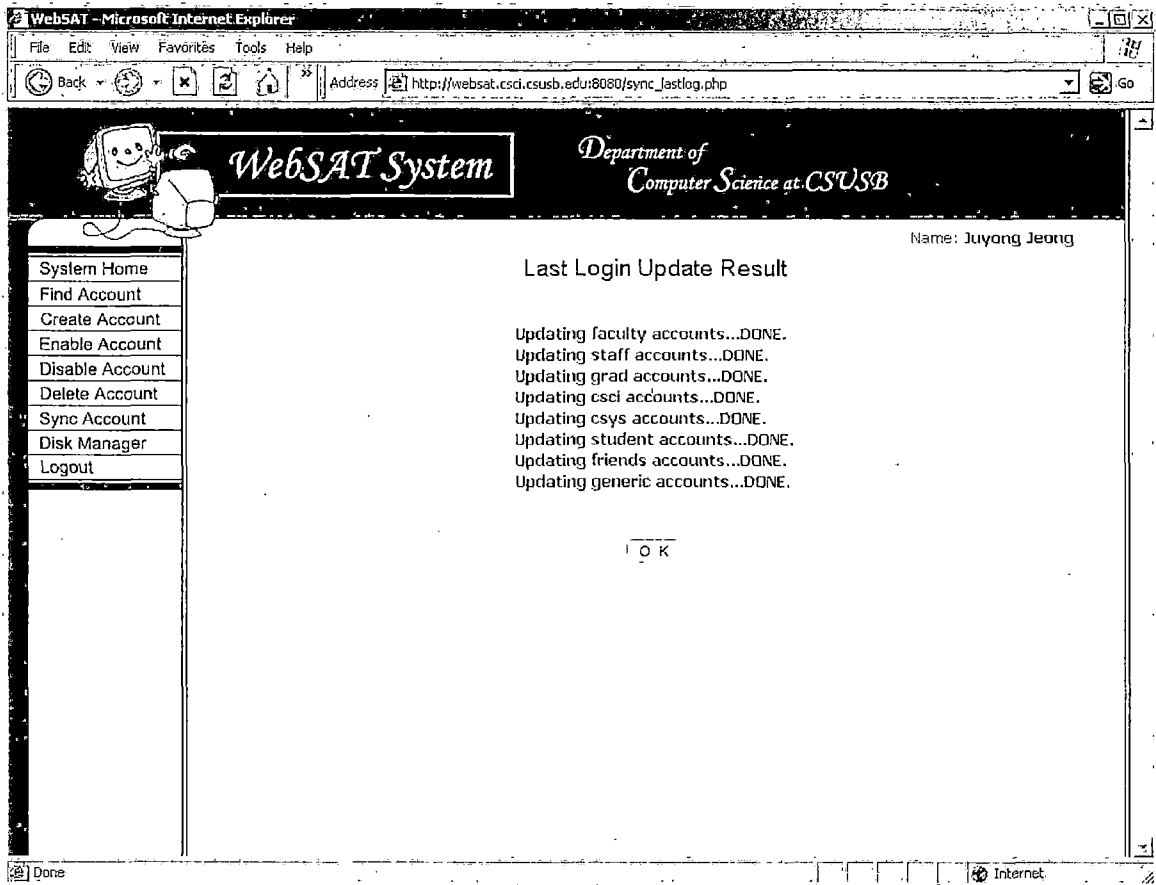


Figure 1.29. Update Last Login Information Page

1.3.1.26 Disk Manager. This page displays the total disk space of file system and draws a pie graph to indicate how much disk space is used (see Figure 1.30). This page, also, displays a list of accounts whose file sizes are over the limit. The user can select accounts from the list to disable them.

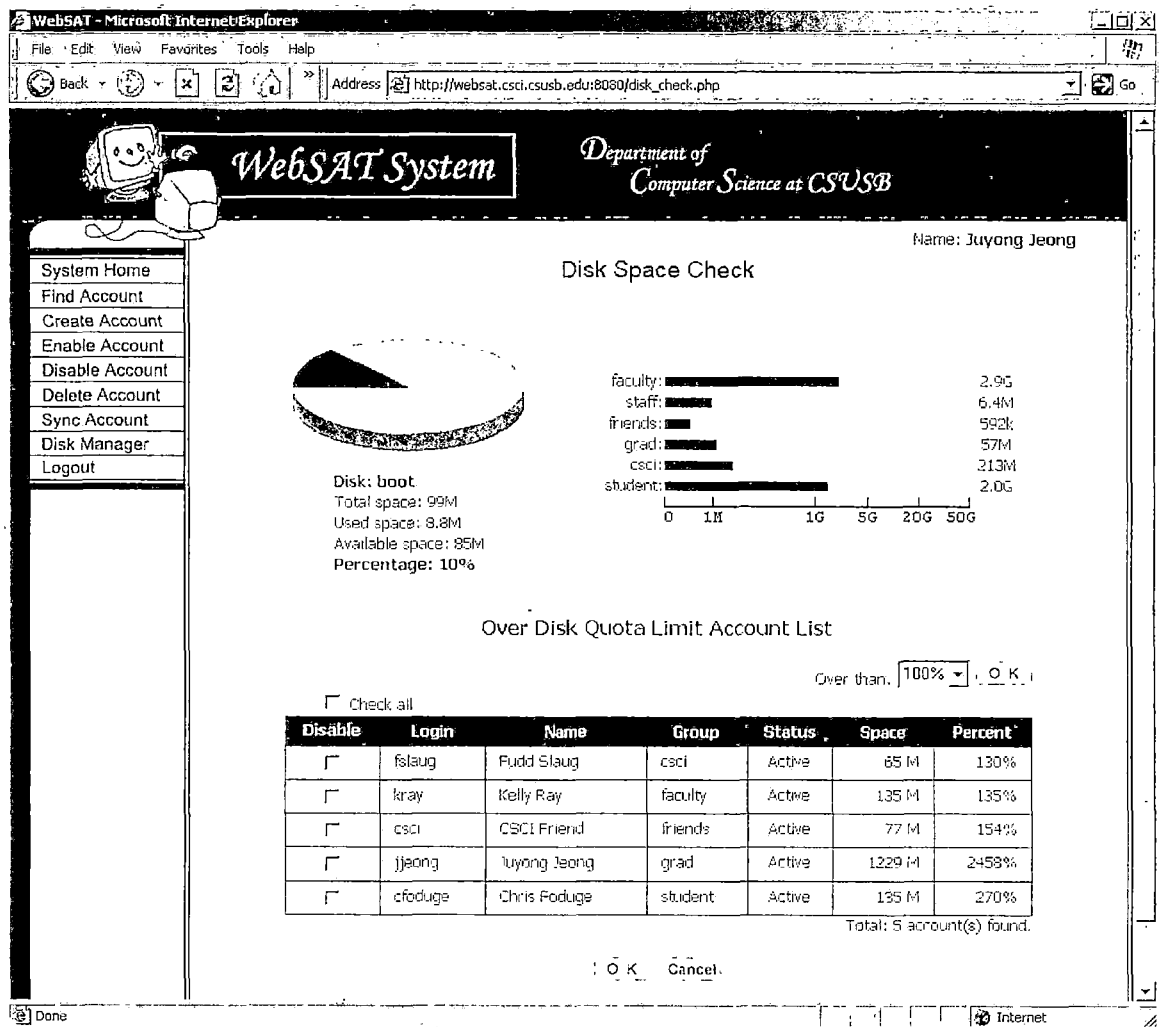


Figure 1.30. Disk Manager Page

1.3.2 Functional Requirements

This section provides a summary of the major functions that WebSAT performs.

1.3.2.1 Validity Checks on Login Information Input

Form. When a user wants WebSAT to display accounts, he/she inputs the information of accounts: logins, student ID, last names, first names, creation date, disabled date, and last login date. When the user provides one or more accounts information, WebSAT makes a query with the provided information and displays all the accounts which correspond to at least one provided information. The user can select a group or all groups. If the user selects a group when he/she inputs account information, WebSAT displays all corresponding information among the group. The user can give many values in logins, last names, and first names fields. Values in these fields are delimited by white space. Student ID consists of three numeric parts. If the user does not provide values in all fields, WebSAT ignores student ID when it makes a query. The following shows how the provided account information is used in the where clause of a SQL statement (see Figure 1.31).

```
Where Clause = user_group is $group and
(login in ($logins) or
 student_id is $student_id or
 last_name in ($last_names) or
 first_name in ($first_names) or
 create_date between ($date1 and $date2) or
 disabled_date between ($date1 and $date2) or
 last_login_date between ($date1 and $date2) or
 all_logins)
```

Figure 1.31. Where Clause in Standard Query Language Statement

1.3.2.2 Validity Checks on Create Account Form. When the user creates a single student account, he/she must provide four kinds of information of the account user: group, student ID, last name, and first name. Login is optional. If the user provides login, the account user gets that login. Otherwise, the WebSAT creates one login for that account user. When the user creates a non-student account, he/she must provide five kinds information of the account user: group, last name, first name, login, and password. When the user creates multiple student accounts, he/she needs the class roster from SIS+. The format of class roster is shown in Table 1.5.

Table 1.5. The Format of Class Roster

Column	Value	Example	Description
7 - 29	Last name and first name	Jeong, Juyong	Last name comes first and then first name. Last name and first name are separated by comma.
30 - 40	Student id	123-12-1234	Format of student id is three numbers, dash, two numbers, dash, and four numbers.
52 - 55	Major	CSCI	Major is four characters in length.

1.3.2.3 Confirmation Check. When the user deletes accounts, WebSAT confirms the deletion. It displays a warning message box to confirm the action with the user. If the user clicks O.K. on the box, WebSAT deletes these accounts. Otherwise, it does not delete accounts.

1.3.2.4 Account Creation Function. When the user creates an account, WebSAT follows the procedure shown in Figure 1.32.

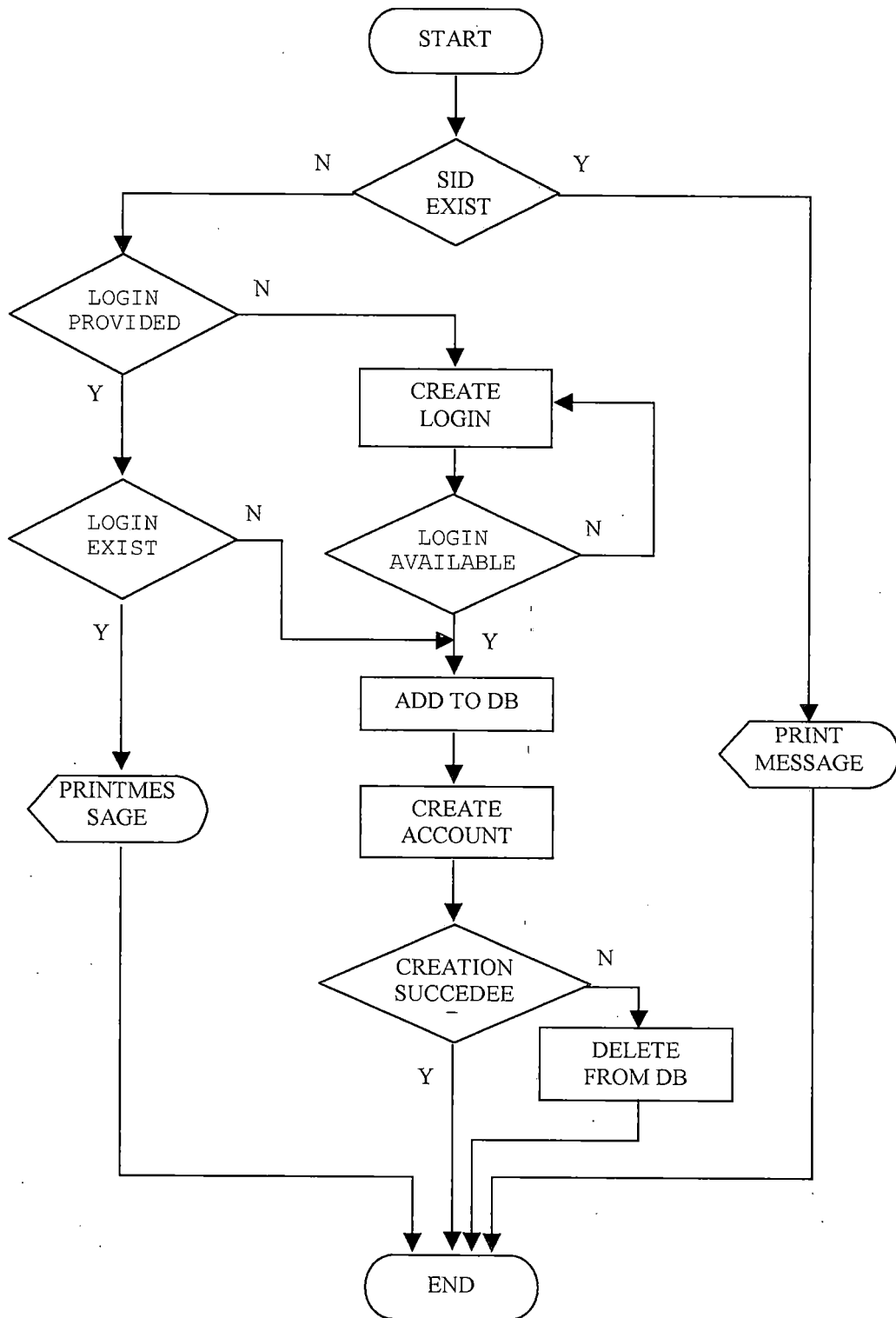


Figure 1.32. Flow Chart for Account Creation

1.3.2.5 Enable Account Function. The user enables only disabled accounts. Therefore, when the user executes this function, WebSAT shows a list of all disabled accounts first. Then, the user selects accounts from the list to disable.

CHAPTER TWO

DESIGN

2.1 Architecture Design

2.1.1 System Design

WebSAT system mainly consists of four components: WebSAT server, Database, NIS, and NFS. The core part of this system is WebSAT server. WebSAT provides its services over the Internet, so Apache Web server is used to provide its services. To implement WebSAT services, a server side embedded scripting language is needed. Among many server side scripting languages, PHP was selected for this project because of its powerful functions and fast loading time of a browser. WebSAT uses a database to manage account information. Because PHP and MySQL are a perfect combination to implement Web-based database applications, MySQL is selected as a WebSAT database server. The traditional network account system uses NIS to save network accounts information and NFS to save user files. However, the mechanism of a traditional network account system is not designed to support enough information to distinguish one of its accounts from other accounts in the system. By building database, WebSAT solves that problem, and that is

the primary purpose of implementing this project. WebSAT system is put together by adding WebSAT server and its database of the traditional network account system.

A client of WebSAT is connected to WebSAT server by TCP/IP protocol. When a WebSAT user creates a network account, WebSAT saves the account user information into its database and updates NIS information, and then it creates the home directory for the user on NFS. When a user logs in his/her account from a workstation, the workstation compares the provided login information with NIS server. If the information is correct, the workstation finds his/her home directory from NFS server, and finally he/she logs in his/her account. While WebSAT server updates the primary NIS information, the secondary NIS synchronizes its information with the information of the primary NIS. A WebSAT user gets student information from sis+ server.

Figure 2.1 shows the architecture in class diagram of WebSAT. Class diagrams show the attributes and operations of a class and the constraints that apply to the way objects are connected. The WebSAT system has nine classes, and each class is connected to Home.

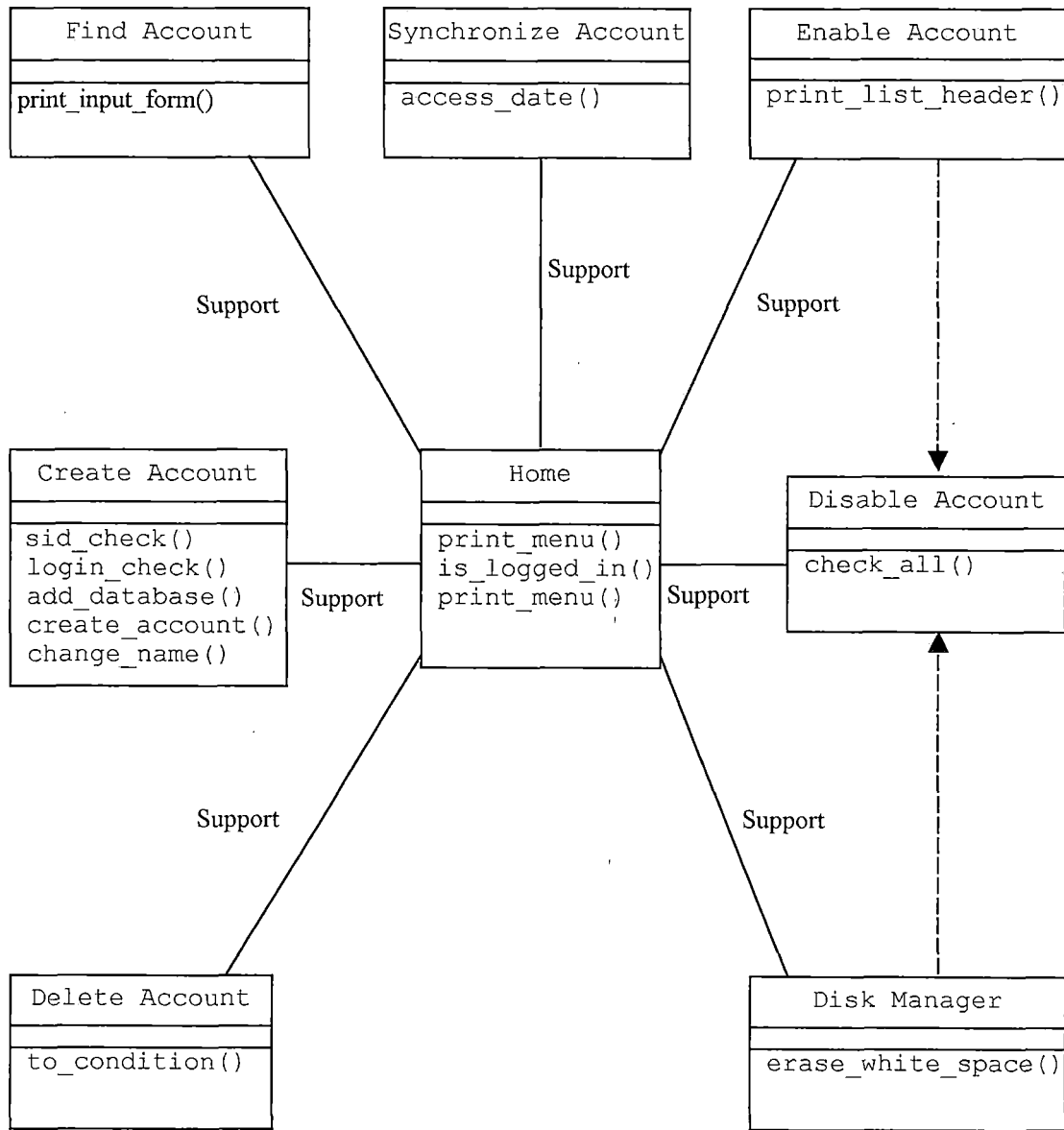


Figure 2.1. Architecture in Class Diagram of WebSAT

2.1.2 Security Design

The main issue in WebSAT project is security because it deals with network account information. Once an outsider obtains network account information or account user information for illegal purposes, the outsider could use this information for breaking the network system or selling user information to a third party. To protect account and user information, WebSAT is shielded with several security layers. WebSAT hides itself from the world. First, WebSAT server allows access from a limited number of authenticated clients. All accesses from other clients except those authorized clients are denied. Second, three firewalls hide WebSAT. First firewall hides the whole campus networks from outside. Second firewall is the Department of Computer Science which hide computer science network. And the third firewall hides WebSAT itself from the outside world. WebSAT uses SSL (Secure Sockets Layer) on TCP/IP. Each transaction between server and client is encrypted, so even if somebody hijacks transactions, the transactions are very hard to be decrypted. WebSAT encrypts critical user information when it saves information in the database. That information can not be decrypted, so even if somebody gets that information, it is useless for him/her. When a hacker plans to break a

computer network, he/she does port scans to find access methods to break into the machine. When somebody scans ports on the WebSAT machine, WebSAT detects this action and reports it to the systems administrator so that the systems administrator takes an adequate action against it. WebSAT functions are controlled by session. If a user does not have any activities for a certain amount of time, the current session is terminated automatically. The user must establish a new session to access WebSAT server again. Each WebSAT page performs session check to prevent abnormal access to the page. If somebody accesses one of WebSAT page without establishing a session, WebSAT rejects the access. WebSAT has a dense security layer which protects itself from the world. Figure 2.2 is the class diagram of PSAS. Figure 2.3 shows the security design of WebSAT.

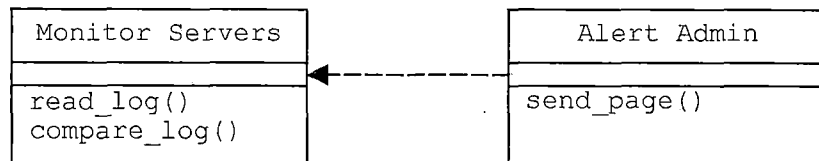
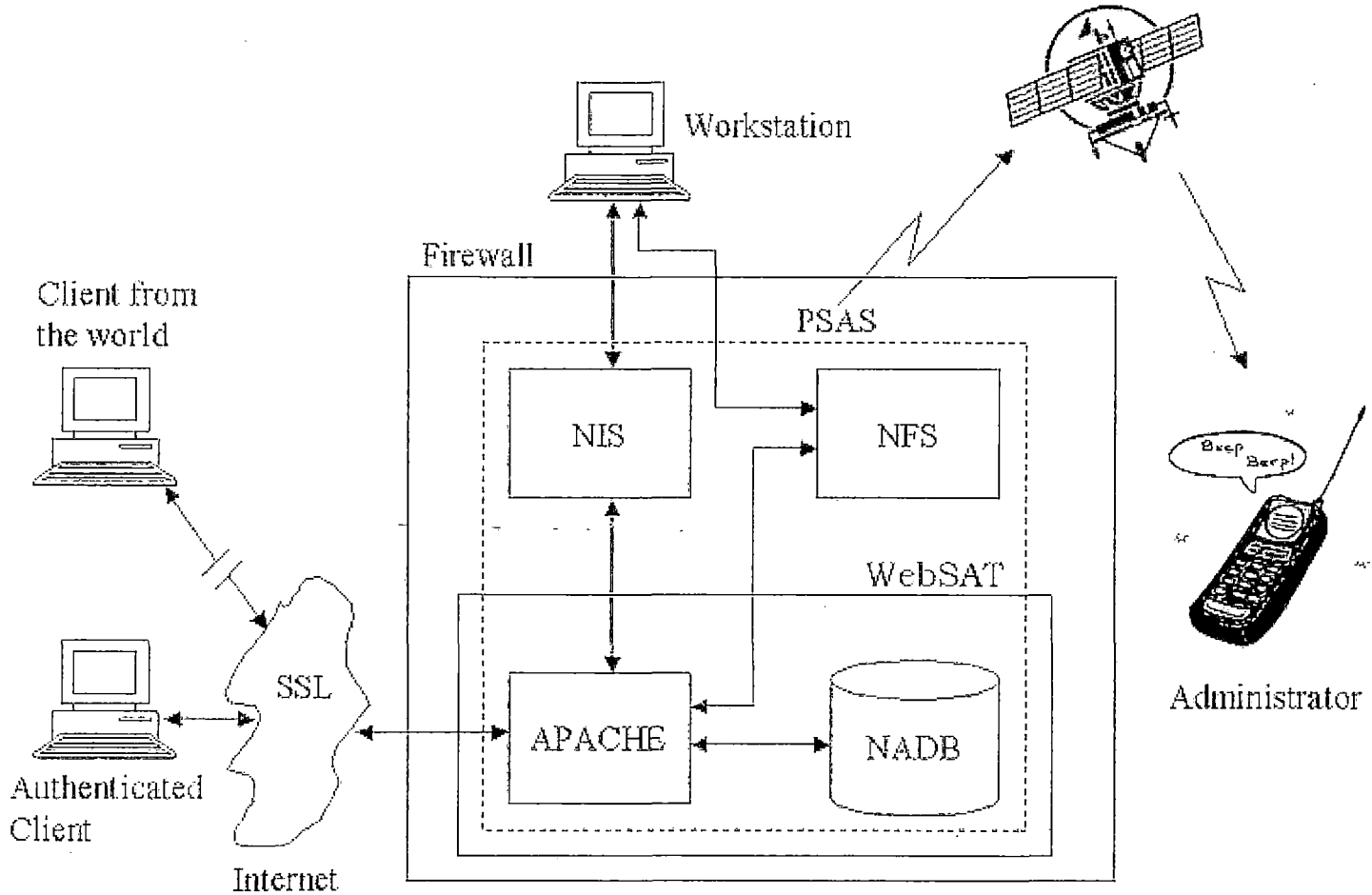


Figure 2.2. Class Diagram of Port Scan Alert System

Figure 2.3. Security Design of WebSAT



2.1.3 Database Design

Based on the database needs of WebSAT, Extended Entity-Relationship Diagram is used to show all objects. Then, this EER Diagram is converted to a relational database schema. A relational database schema presents all objects in EER Diagram as a series of related tables. WebSAT database has been normalized for performance and logical error reduction.

2.1.3.1 Extended Entity-Relationship Diagram. WebSAT has six entities: ACCOUNT, STATUS,, GROUPS, STUDENT, NON-STUDENT, and ADMIN. The EER diagram shows all entities and their attributes. Also, it shows constraints and the relationships between entities.

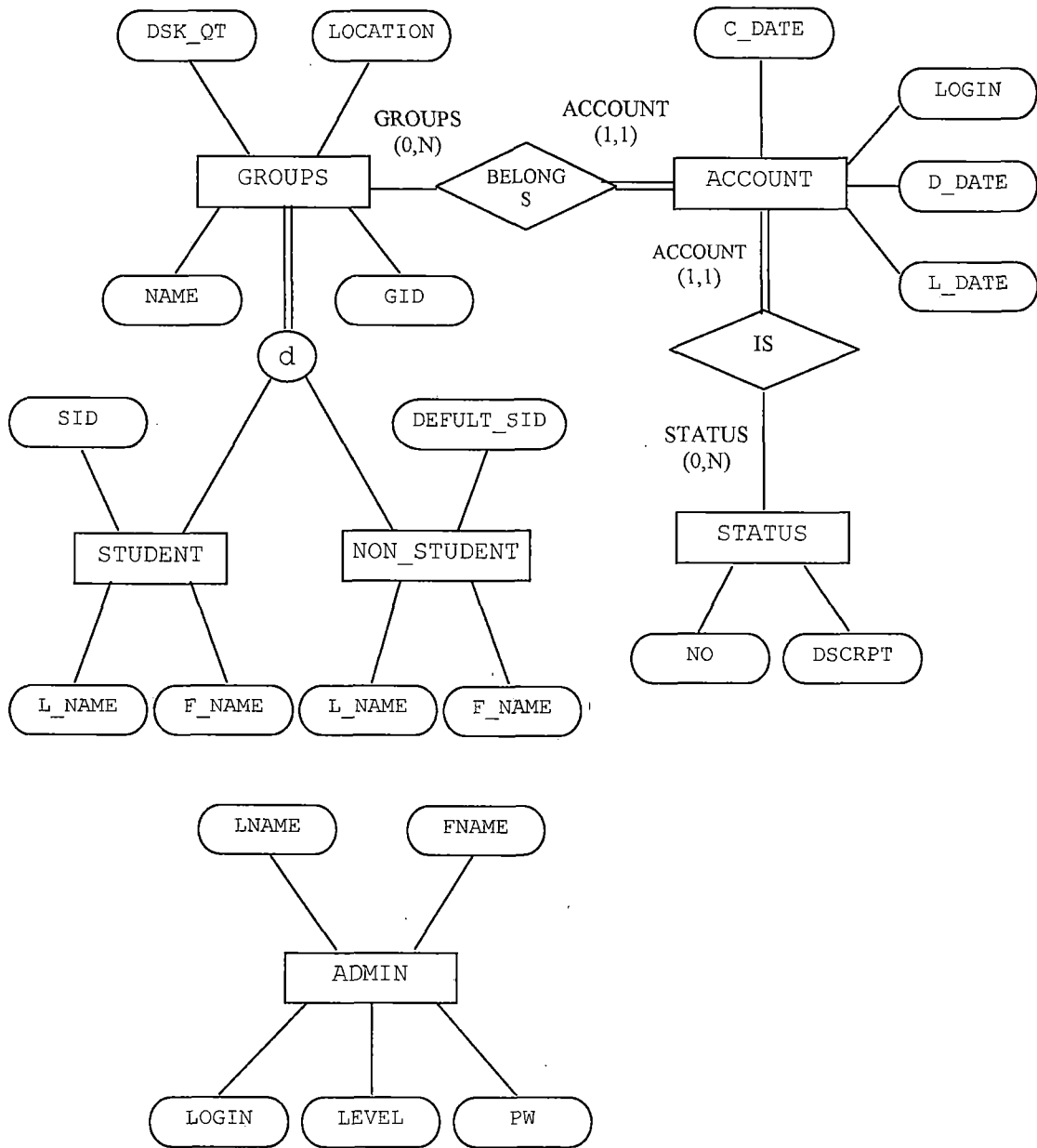


Figure 2.4. Extended Entity-Relationship Diagram

2.1.3.2 Relational Database Tables. Relational

database schema of WebSAT consists of four relation schemas: ACCOUNT, STATUS, GROUPS, and ADMIN. The characteristics of each schema are described on Table 2.1 - 2.4. Account table is used to save account information. User name, student ID number, login name, group ID, account status, created date, last logged in date, and disabled date are saved in this table. Groups table has group type which is one of non-student, student, generic, or system. Also, group ID, group name, directory location, disk quota, and default SID are saved in groups table. Status table has information about account status. Account status is known by referring to this status table. Admin table consists of user name, login name, password, and access level. In these tables student ID in account table and password in admin table are encrypted.

Table 2.1. Account Table

FIELD NAME	DATA TYPE	DESCRIPTION
L_NAME	VARCHAR(20)	Last name of account owner Not null
F_NAME	VARCHAR(20)	First name of account owner Not null
SID	CHAR(16)	Student id of account owner Not null This value is encrypted
LOGIN	VARCHAR(10)	Login name Primary key Not null
GRP	INTEGER	User group id Foreign Key: NO of GROUPS Table
STATUS	INTEGER	Status of account activation Foreign Key: NO of STATUS Table
C_DATE	DATE	Date when account is created
L_DATE	DATE	Date when account is last accessed
D_DATE	DATE	Date when account is disabled by the systems administrator

Table 2.2. Groups Table

FIELD NAME	DATA TYPE	DESCRIPTION
TYPE	INTEGER	Account group type Not null 1: non-student type (faculty, staff, and friends group) 2: student type (student, csci, and grad group) 3: generic type (generic group) 4: system type (system group)
GID	INTEGER	Group ID number Primary key Not null 100: faculty 300: staff 400: generic 500: student 550: csys 600: csci 700: friends 800: grad
NAME	VARCHAR(20)	Group name
DEFAULT_SID	CHAR(11)	Default student id for non-student, generic, and system type
MAJOR	VARCHAR(20)	Abbreviation for major
LOCATION	VARCHAR(50)	Directory path for groups
DSK_QT	INTEGER	Allowed Disk Space in megabyte

Table 2.3. Status Table

FIELD NAME	DATA TYPE	DESCRIPTION
NO	INTEGER	Primary key Not null 0: disabled account 1: active account 2: disabled by administrator
DSCRIPT	VARCHAR(40)	Description of the status of account Not null Disabled account (0) Active account (1) Disable by administrator (2)

Table 2.4. Admin Table

FIELD NAME	DATA TYPE	DESCRIPTION
LNAME	VARCHAR(20)	Last name of administrator
FNAME	VARCHAR(20)	First name of administrator
LOGIN	CHAR(16)	Login of administrator Primary key Value is encrypted
PW	CHAR(16)	Password of administrator Value is encrypted
LEVEL	INTEGER	Access rights to the WebSAT This is for later purpose 0: full access

Table ACCOUNT

<u>LOGIN</u>	L_NAME	F_NAME	SID	GRP	STATUS	C_DATE	L_DATE	D_DATE
--------------	--------	--------	-----	-----	--------	--------	--------	--------

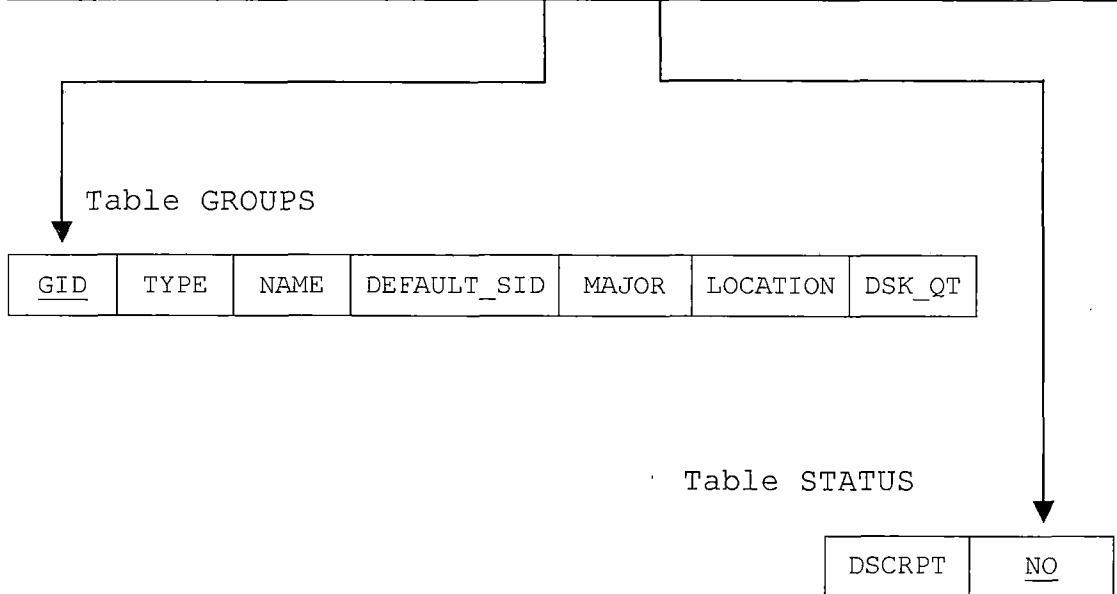


Table ADMIN

<u>LOGIN</u>	LNAME	FNAME	PW	LEVEL
--------------	-------	-------	----	-------

Figure 2.5. Relational Database Schema

Figure 2.5 shows relational database schema. ADMIN table which has LOGIN field as its primary key has no connection to other tables. However, ACCOUNT table, GROUPS table, and STATUS table are connected to each other. GRP in ACCOUNT table is used as reference to GROUPS table, and STATUS in ACCOUNT table is used as reference to STATUS table.

2.2 Detailed Design

This section describes the detailed design showing the pseudo-code of all classes shown in Figures 2.1 and 2.2.

2.2.1 Home Class

Home class displays WebSAT menus and check whether the user is registered or not. Also, this class connects WebSAT to its database.

```
Class name: home
Purpose: to display menus
Begin class

Function dbconn: no return
/* connect physical database */
Begin
  Activate driver
  Establish connection to the database
End

Function print_menu: no return
/* print menus */
Begin
  Print WebSAT menus
End

Function is_logged_in: return int
/* verify user identification */
Begin
  Execute SQL statement
  If user is registered
    Return 1
  Else
    Return 0
  End If
End

End class
```

Figure 2.6. Home Class

2.2.2 Find Account Class

Find account class extends home class. This class has print_input_form function which prints the account information input form.

```
Class name: find_account extends home
Purpose: to find and display account information
Begin class

Function dbconn: no return
Begin
  dbconn();
End

Function print_menu: no return
Begin
  print_menu();
End

Function is_logged_in: return int
Begin
  is_logged_in();
End

Function print_input_form: no return
/* print input form */
Begin
  Print account information input form
End

End class
```

Figure 2.7. Find Account Class

2.2.3 Create Account Class

Create account class extends home class. This class has sid_check, login_check, add_database, create_account, and change_name functions. sid_check function check whether a SID exists in the database. login_check verifies whether a login is in the database. add_database function inserts a user information into the database. create_account function creates an account in the system. Change_name function gets rid of all special characters from a user name.

```
Class name: create_account extends home
Purpose: to create network accounts
Begin class

Function dbconn: no return
Begin
  dbconn();
End

Function print_menu: no return
Begin
  print_menu();
End

Function is_logged_in: return int
Begin
  is_logged_in();
End

Function print_input_form: no return
Begin
  Print account information input form
```

Figure 2.8. Create Account Class

```

End

Function sid_check: return int
/* prevent a SID conflict */
Begin
    Check if there is the same SID in the database
    Return the result
End

Function login_check: return int
/* prevent a login conflict */
Begin
    Make a login for a new account
    Check if there is the same login in the database
    Return the result
End

Function add_database: return int
/* update database */
Begin
    Insert account user information into the database
    Check if the information inserted successfully
    Return the result
End

Function create_account: return int
/* create account */
Begin
    Create an account in the system
    Check if the account created successfully
    Return the result
End

Function change_name: return string
/* get ready to be used as a login */
Begin
    Remove all special characters from the input string
    Return a new string
End

End class

```

Figure 2.8. Create Account Class (Continued)

2.2.4 Enable Account Class

Enable account class extends home class. This class has print_list_header function which print the header of a result table.

```
Class name: enable_account extends home
Purpose: to enable an account
Begin class

Function dbconn: no return
Begin
  dbconn();
End

Function print_menu: no return
Begin
  print_menu();
End

Function is_logged_in: return int
Begin
  is_logged_in();
End

Function print_list_header: no return
/* print table header */
Begin
  Print the header of account list table
End

End Class
```

Figure 2.9. Enable Account Class

2.2.5 Disable Account Class

Disable account class extends home class. This class has check_all function which gives a user an easy way to select all account information.

```
Class name: disable_account extends home
Purpose: to disable accounts
Begin class

Function dbconn: no return
Begin
    dbconn();
End

Function print_menu: no return
Begin
    print_menu();
End

Function is_logged_in: return int
Begin
    is_logged_in();
End

Function check_all: no return
/* Give a user an easy way to select all buttons */
Begin
    Check if check_all button is clicked
    If clicked
        Check all login buttons
    Elseif
        Uncheck all login buttons
    Endif
End

End Class
```

Figure 2.10. Disable Account Class

2.2.6 Delete Account Class

Delete account class extends home class. This class has to_condition function. to_condition function converts input string to where clause of SQL statement.

```
Class name: delete_account extends home
Purpose: to delete an account
Begin class

Function dbconn: no return
Begin
  dbconn();
End

Function print_menu: no return
Begin
  print_menu();
End

Function is_logged_in: return int
Begin
  is_logged_in();
End

Function to_condition: return string
/* make a where clause of SQL with string input */
Begin
  Split input with white space delimiter
  Make where clause with each value
  Return where clause
End

End Class
```

Figure 2.11. Delete Account Class

2.2.7 Synchronize Account Class

Synchronize account class extends home class. This class has access_date function. This function gets last access information from the system, converts the information to date type string, and returns the string.

```
Class name: sync_account extends home
Purpose: to synchronize the information in the system and in
the database
Begin class

Function dbconn: no return
Begin
    dbconn();
End

Function print_menu: no return
Begin
    print_menu();
End

Function is_logged_in: return int
Begin
    is_logged_in();
End

Function access_date: return string
/* make a date type string with a Linux output */
Begin
    Check if login date is before six months from now
    If yes
        year = current_year - 1
    Else
        year = current_year
    Endif
    Return 'year-current_month-current_date'
End

End Class
```

Figure 2.12. Synchronize Account Class

2.2.8 Disk Manager Class

Disk manager class extends home class. This class has erase_white_space function which removes all white space in the input string, and returns the result.

```
Class name: disk_manager extends home
Purpose: to manage disk spaces
Begin class

Function dbconn: no return
Begin
  dbconn();
End

Function print_menu: no return
Begin
  print_menu();
End

Function is_logged_in: return int
Begin
  is_logged_in();
End

Function erase_white_space: return array
/* remove white space in a string */
Begin
  Split a string with white space delimiter
  Get rid of empty strings
  Return strings
End

End Class
```

Figure 2.13. Disk Manager Class

2.2.9 Alert Admin Class

Alert admin class extends home class. This class has send_page function. This function sends a summary of new log to the systems administrator.

```
Class name: alert_admin extends home
Purpose: to alert the system administrator the security
attack

Function dbconn: no return
Begin
  dbconn();
End

Function print_menu: no return
Begin
  print_menu();
End

Function is_logged_in: return int
Begin
  is_logged_in();
End

Function send_page: return string
/* send a page to the systems administrator */
Begin
  Send a page to the system administrator
End

End Class
```

Figure 2.14. Alert Admin Class

CHAPTER THREE

SOFTWARE QUALITY ASSURANCE

3.1 Introduction

This chapter documents software validation testing processes for WebSAT. The purpose of software validation test is to guaranty the quality of software and its functionalities. Two testing processes are used to assure WebSAT software quality: unit test and system test.

3.2 Unit Testing

Unit testing greatly improved the quality of WebSAT. It also accelerated the development of this project, since unit testing allowed individual modules to be tested before the entire program was completed. Table 3.1 shows the result of unit testing for WebSAT.

Table 3.1. Unit Testing Results

Page	Unit	Tests Performed	Result
Login	Login	<ul style="list-style-type: none">• Test if only authenticated users access WebSAT	Pass
	Buttons/Links	<ul style="list-style-type: none">• Ensure all buttons and links work as expected	Pass

Table 3.1. Unit Testing Results (continued)

Page	Unit	Tests Performed	Result
Find Account	Account Information Input Form	<ul style="list-style-type: none"> • Verify all groups are in the select input form • Verify all input fields work as specified in Functional Requirements Section 	Pass
	Account list	<ul style="list-style-type: none"> • Check all column names and displayed information are corresponding • Verify all information is sorted by column 	Pass
	Buttons/Links	<ul style="list-style-type: none"> • Ensure all buttons and links work as expected 	Pass
Create Account	Menu	<ul style="list-style-type: none"> • Ensure all menus are displayed and linked to proper pages 	Pass
	Student Account Creation Form	<ul style="list-style-type: none"> • Verify all student type groups are in the select input form • Verify student id input fields validate proper data types • Verify give a login field works correct • Verify input field for multiple account creation works correct 	Pass
	Buttons/Links	<ul style="list-style-type: none"> • Ensure all buttons and links work as expected 	Pass
Enable Account	Menu	<ul style="list-style-type: none"> • Ensure all menus are displayed and linked to proper pages 	Pass
	List of Disabled Accounts	<ul style="list-style-type: none"> • Verify the list show all disabled accounts • Verify all check boxes are associated with correspond account information • Verify all information is sorted by column 	Pass

Table 3.1. Unit Testing Results (continued)

Page	Unit	Tests Performed	Result
Enable Account	Buttons/Links	<ul style="list-style-type: none"> • Ensure all buttons and links work as expected 	Pass
Disable Account	Account Information Input Form	<ul style="list-style-type: none"> • Verify all groups are in the select input form • Verify all input fields work as specified in Functional Requirements Section 	Pass
	List of Disabled Accounts	<ul style="list-style-type: none"> • Verify check_all function works • Verify all check boxes are associated with corresponding account information • Verify all information is sorted by column 	Pass
	Buttons/Links	<ul style="list-style-type: none"> • Ensure all buttons and links work as expected 	Pass
Delete Account	Account Information Input Form	<ul style="list-style-type: none"> • Verify all groups are in the select input form • Verify all input fields work as specified in Functional Requirements Section 	Pass
	List of Disabled Accounts	<ul style="list-style-type: none"> • Verify check_all function works • Verify all check boxes are associated with correspond account information • Verify all information is sorted by column 	Pass
	Buttons/Links	<ul style="list-style-type: none"> • Ensure all buttons and links work as expected 	Pass

Table 3.1. Unit Testing Results (continued)

Page	Unit	Tests Performed	Result
Synchronize Account	Menu	<ul style="list-style-type: none"> • Ensure all menus are displayed and linked to proper pages 	Pass
	Synchronize Login Information	<ul style="list-style-type: none"> • Verify the result of synchronization check returns correct information • Verify WebSAT provides proper options on each login according to the result of synchronization check. 	Pass
	Group Information Check	<ul style="list-style-type: none"> • Verify the result of group information check returns correct information • Verify WebSAT provides proper group for logins which have wrong group information • Verify all check boxes are associated with correspond account information 	Pass
	Last Login Information Update	<ul style="list-style-type: none"> • Check WebSAT performs this function for all groups in the system 	Pass
	Buttons/Links	<ul style="list-style-type: none"> • Ensure all buttons and links work as expected 	Pass
Disk Manager	Visual Aids	<ul style="list-style-type: none"> • Check pie graph and bar graphs indicate correct values 	Pass
	Account List	<ul style="list-style-type: none"> • Check all column names and displayed information are corresponding • Verify check_all functions work • Verify all check boxes are associated with corresponding account information 	Pass

Table 3.1. Unit Testing Results (continued)

Page	Unit	Tests Performed	Result
Disk Manager	Account List	<ul style="list-style-type: none"> • Verify all information is sorted by column • Verify account information is displayed by given percentage 	Pass

3.3 System Testing

System testing involves examination of the whole WebSAT system. All the software components, all the hardware components and any interfaces are tested in this process. Table 3.2 shows the result of system testing for WebSAT.

Table 3.2. System Testing Results

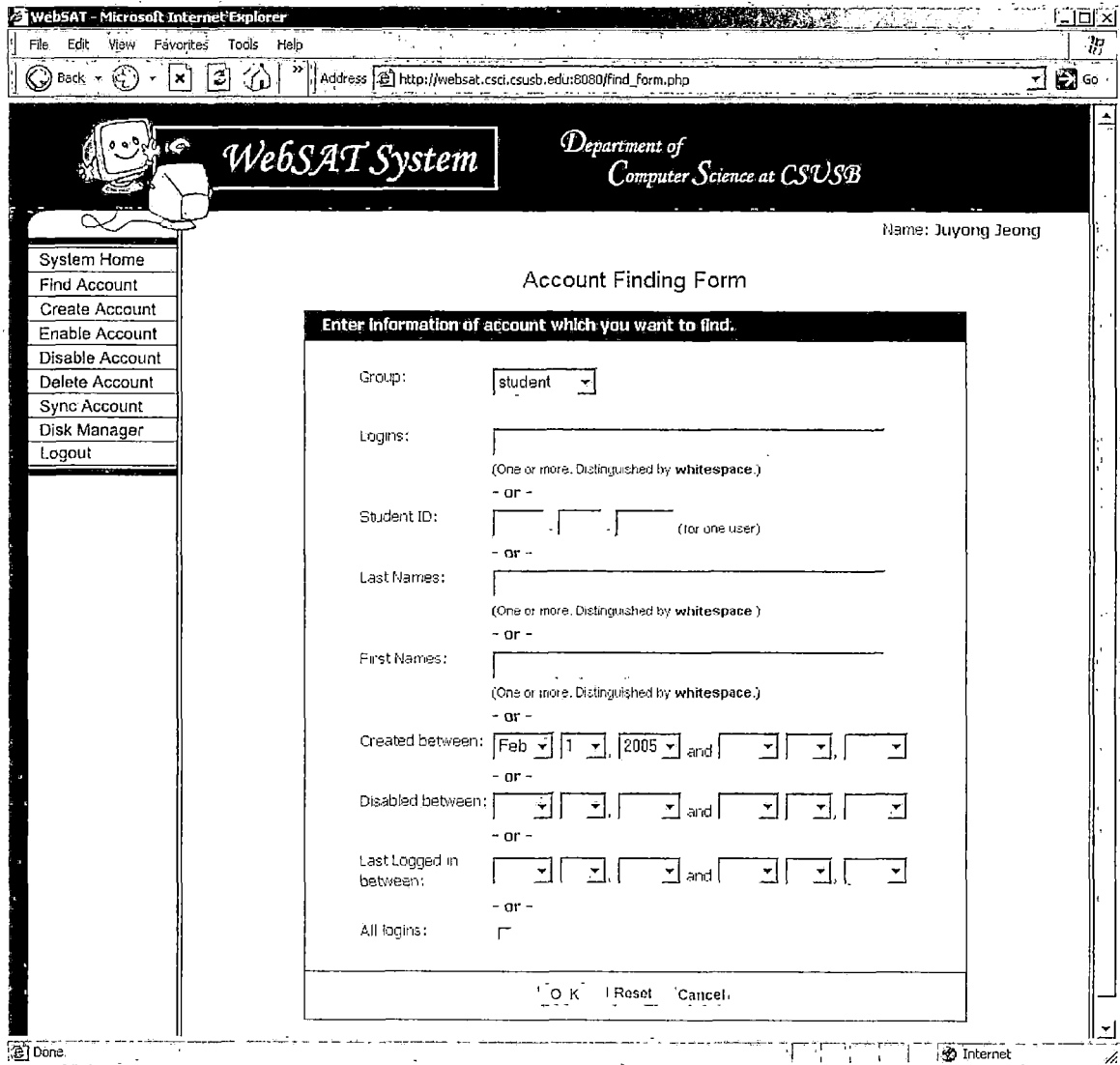
System Testing	Result
• Install MySQL and start mysqld	Pass
• Install apache secure web server and start httpd	Pass
• Install WebSAT program on secure web server	Pass
• Verify connections from unauthorized clients are not established	Pass
• Verify Port Scan Alert System works	Pass
• Verify all functions of WebSAT work with various real data	Pass

3.4 Scenarios

This section shows sample runs of WebSAT showing all the functions of WebSAT.

3.4.1 Find Account

Figure 3.1 demonstrates find account function. The user wants to find all student group accounts which are created after February 1, 2005 (see Figure 3.1 (1)). WebSAT shows the information of five accounts which are created after that day (see Figure 3.1 (2)). Also, Figure 3.1 (3) shows the same results, but the results are ascending ordered by Login. If the user puts invalid values in SID fields or provides invalid dates, WebSAT ignores those conditions and just queries with other conditions. Figure 3.1 (4) and Figure 3.1 (5) show this scenario.



(1)

Figure 3.1. Demonstration of Find Account

WebSAT - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://websat.csci.csusb.edu:8080/find.php

WebSAT System Department of Computer Science at CSUSB

Name: Juyong Jeong

List of Found Accounts

Login	Name	Group	Created	Last Log	Status	Disabled
jdob	Johnson Doe	student	2005-02-09	2005-02-09	Active	
gsylose	Green Sylose	student	2005-02-09	2005-02-09	Active	
dniken	David Niken	student	2005-02-09	2005-02-09	Active	
nneapell	Natali Neapell	student	2005-02-09	2005-02-09	Active	
fyang	Faiya Yang	student	2005-02-09	2005-02-09	Active	

Total: 5 account(s) found

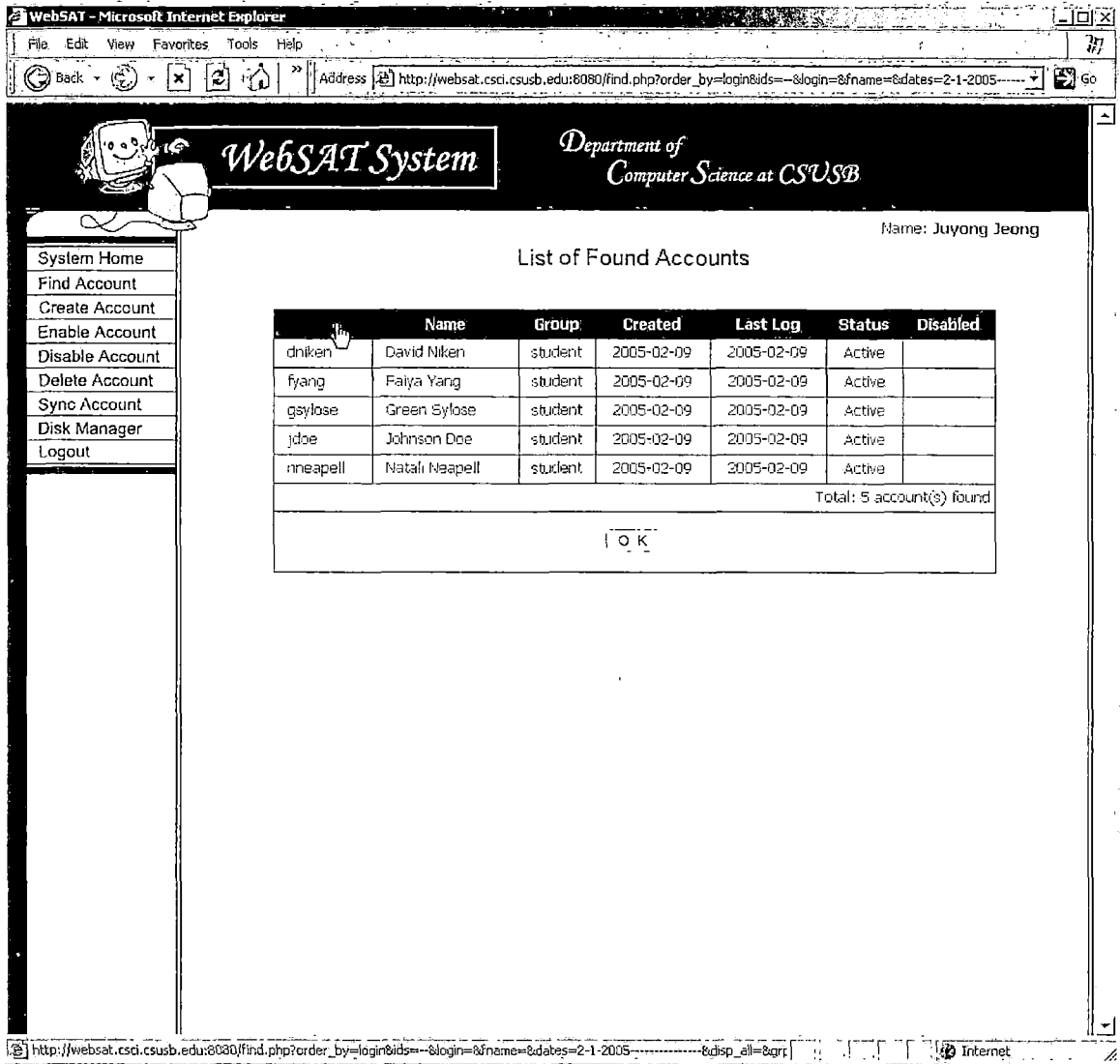
System Home
Find Account
Create Account
Enable Account
Disable Account
Delete Account
Sync Account
Disk Manager
Logout

http://websat.csci.csusb.edu:8080/find_form.php

Internet

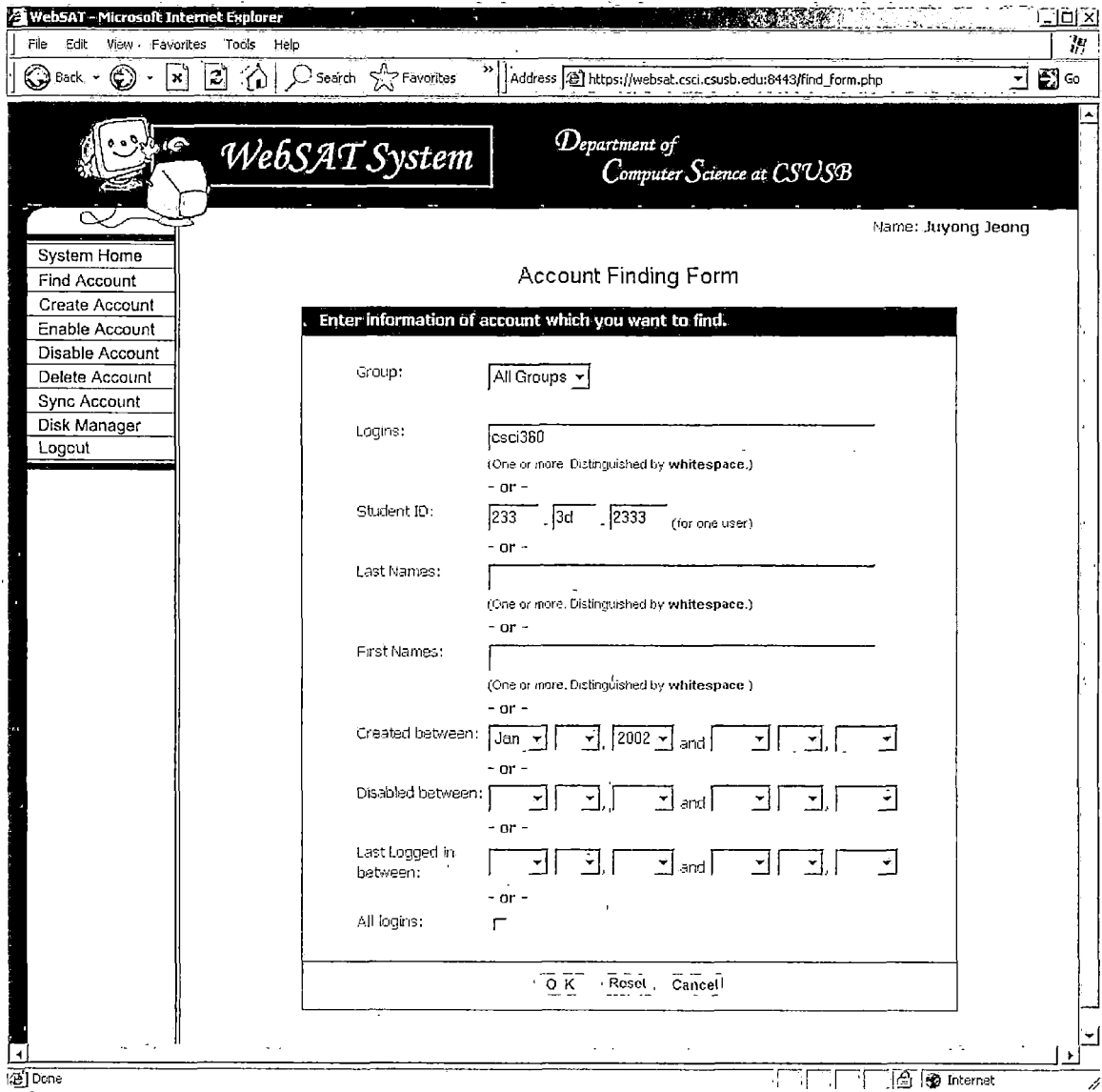
(2)

Figure 3.1. Demonstration of Find Account (Continued)



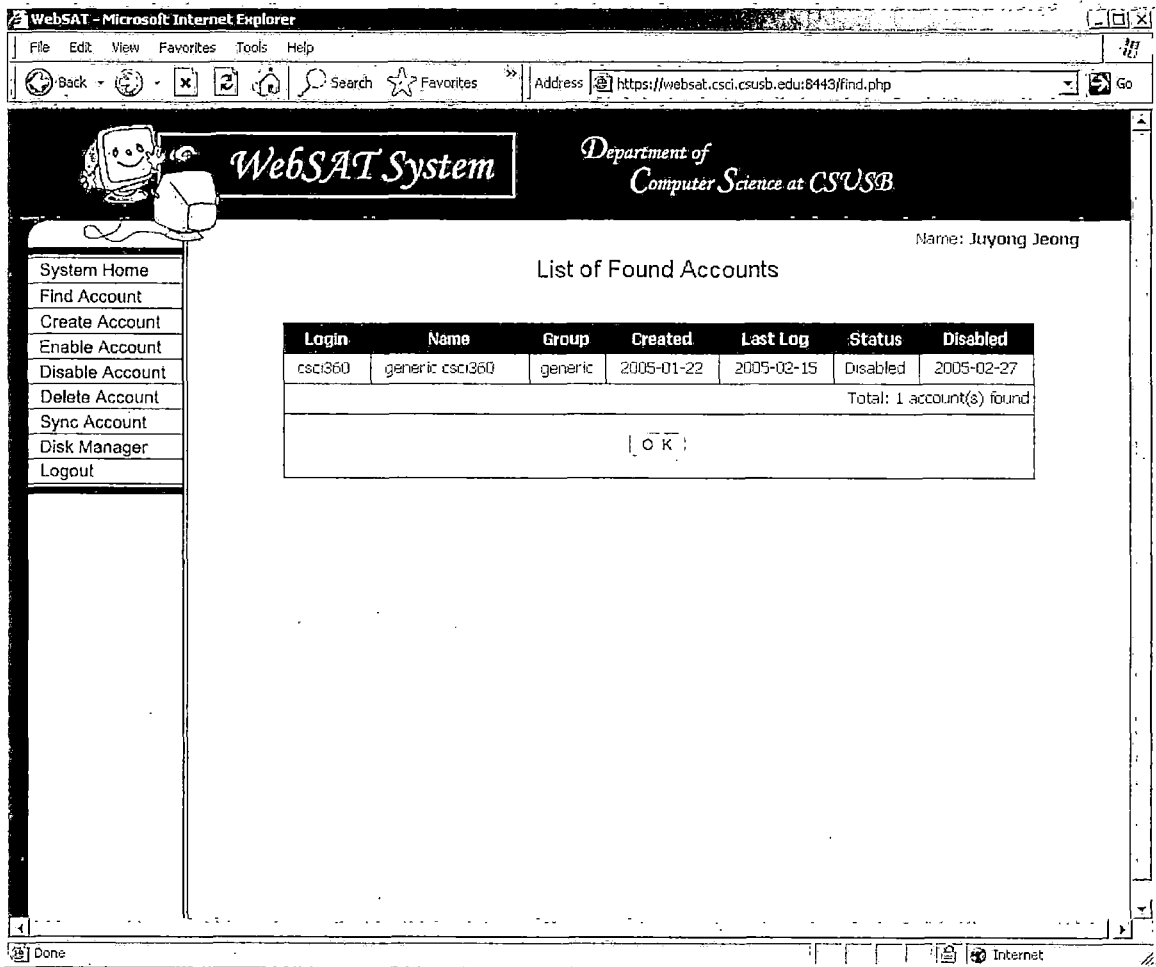
(3)

Figure 3.1. Demonstration of Find Account (Continued)



(4)

Figure 3.1. Demonstration of Find Account (Continued)

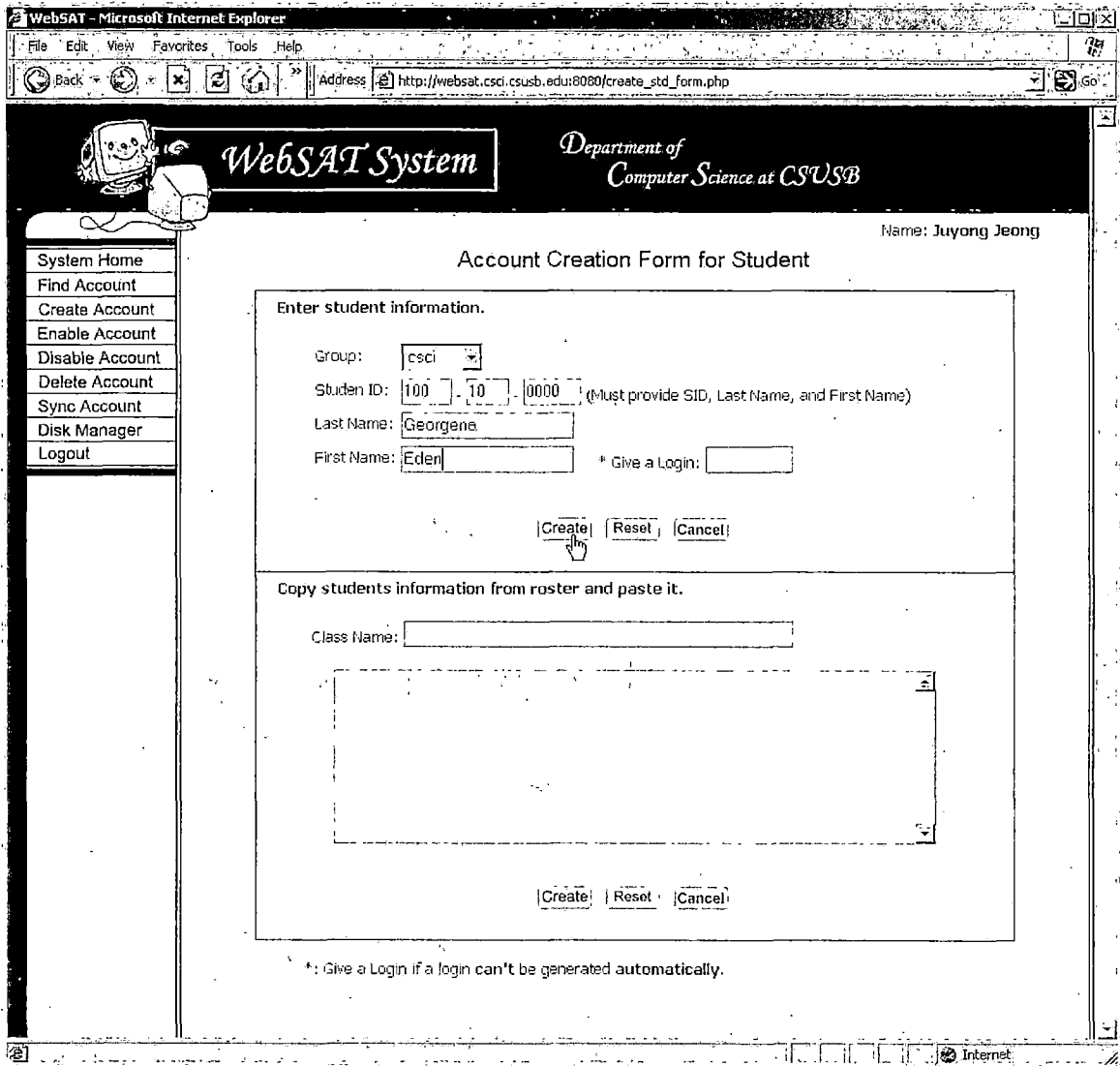


(5)

Figure 3.1. Demonstration of Find Account (Continued)

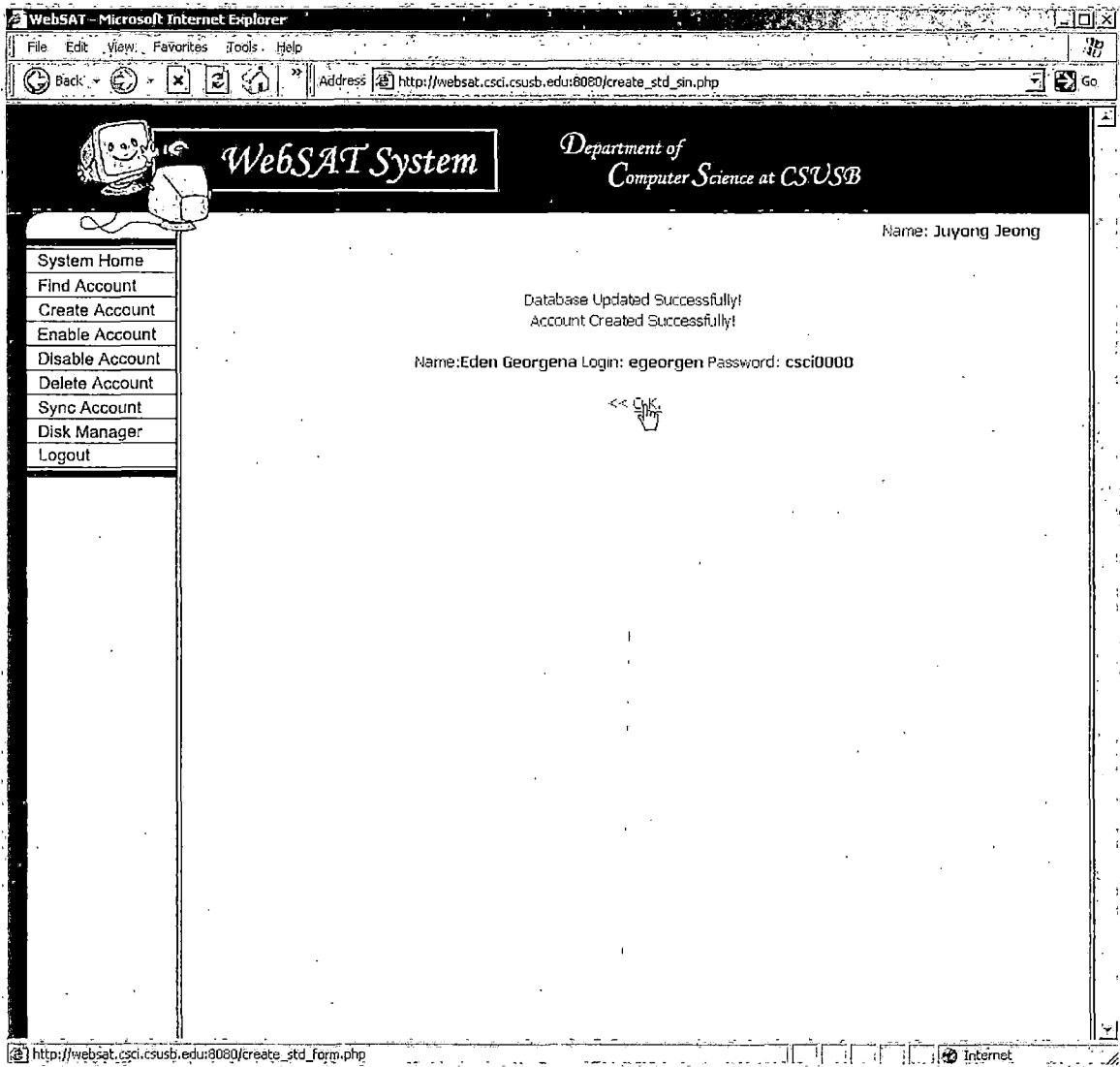
3.4.2 Create Account

Figure 3.2 demonstrates create account function. Figure 3.2 (1) and Figure 3.2 (2) show single account creation. The user created an account for Eden Georgena, and WebSAT assigned login egeorgen to Eden Georgena. Figure 3.2 (3) and Figure 3.2 (4) show multiple accounts creation. The user inserted seven students' information, and WebSAT created only six accounts for the students because Mirta Chandravadia already has an account. The results show that one login is not made with first character of first name and last name but first two characters of first name and last name because somebody already took the login name. If the user forgets to enter all information on single account creation form, WebSAT pops up a alert window and asks the user to enter all required information (see Figure 3.2 (5)). Figure 3.2 (6) and Figure 3.2 (7) show how WebSAT detects errors when students' information provided to multiple account creation form does not follow rules. If WebSAT detects one or more errors in students' information, WebSAT does not create accounts for all students. It just shows error message and asks the user to check students' information and to create accounts again.



(1)

Figure 3.2. Demonstration of Create Account



(2)

Figure 3.2. Demonstration of Create Account (Continued)

WebSAT - Microsoft Internet Explorer
 Address: http://websat.csd.csusb.edu:8080/create_std_form.php

WebSAT System Department of Computer Science at CSUSB
 Name: Juyong Jeong

Account Creation Form for Student

Enter student information.

Group:

Student ID: (Must provide SID, Last Name, and First Name)

Last Name:

First Name: * Give a Login:

Copy students information from roster and paste it.

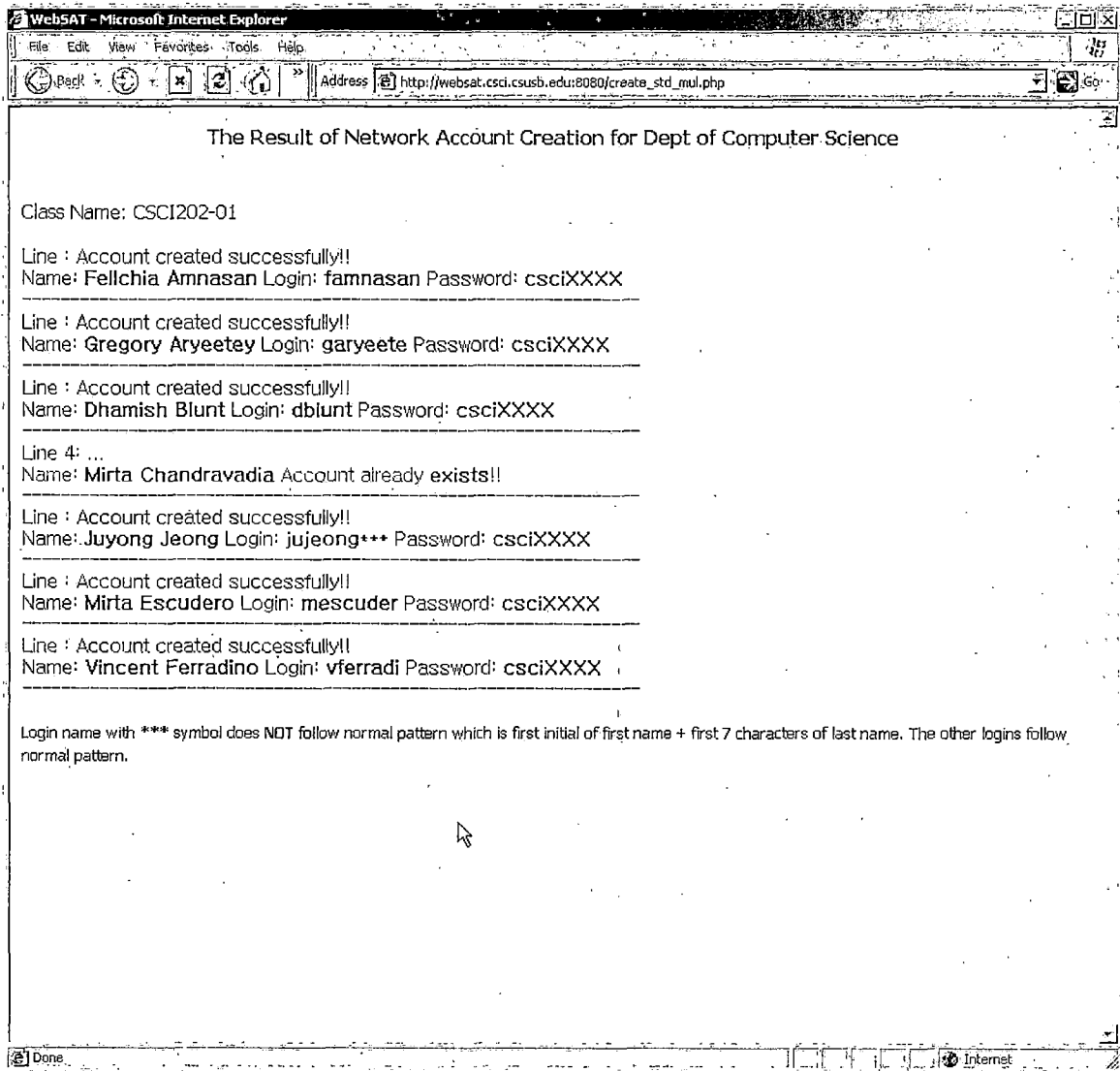
Class Name:

1	Anasan, Felichia Delo	583-55-8876	NS	4	FNTR	Enrolled
2	Aryeetey, Gregory	641-57-3635	SBS	2	MATH	Enrolled
3	Blunt, Dhanish	832-35-4632	HUN	4	CMAS	Enrolled
4	Chandravadia, Mirta	741-22-2222	BPA	3	INFO	Enrolled
5	Jeong, Juyong	387-32-2222	NS	3	NTHN	Enrolled
6	Escudero, Mirta Lidia	234-45-5678	HUN	4	CMAS	Enrolled
7	Ferradino, Vincent Ant	942-52-2568	NS	4	PRNR	Enrolled

*: Give a Login if a login can't be generated automatically.

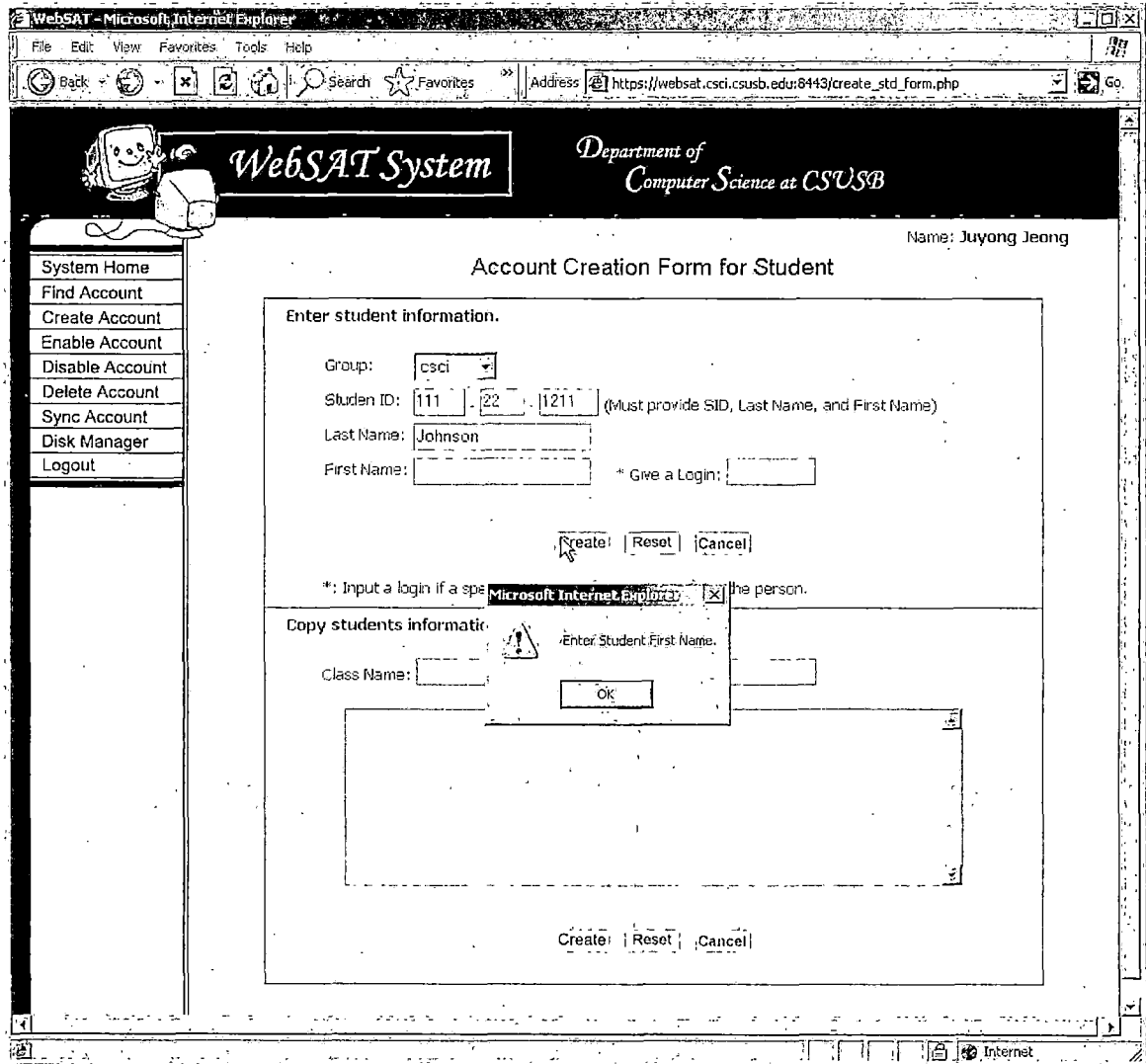
(3)

Figure 3.2. Demonstration of Create Account (Continued)



(4)

Figure 3.2. Demonstration of Create Account (Continued)



(5)

Figure 3.2. Demonstration of Create Account (Continued)

WebSAT - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address: https://websat.csci.csusb.edu:8443/create_std_form.php

WebSAT System Department of Computer Science at CSUSB

Name: Juyong Jeong

Account Creation Form for Student

Enter student information.

Group:

Student ID: (Must provide SID, Last Name, and First Name)

Last Name:

First Name: * Give a Login:

*: Input a login if a special login is needed to be given to the person.

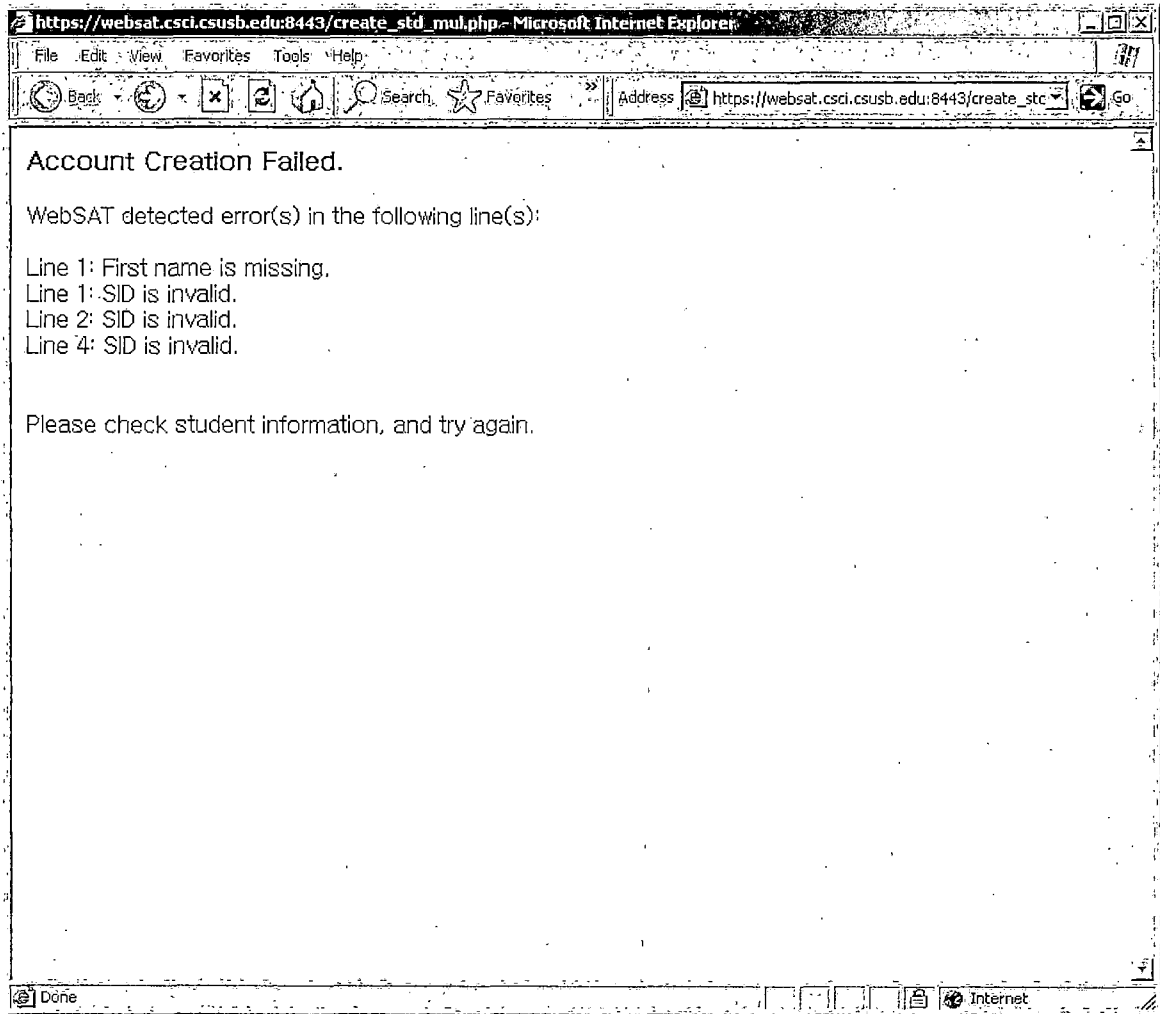
Copy students information from roster and paste it.

Class Name:

1	Doedan-Olaru Camelia	722-23-0523	NS	3	MATH	Enrolled
2	Bowles, Stephanie Anna	55A-99-7727	NS	3	CSCM	Enrolled
3	Bridges, Bryan Jason	525-59-9929	NS	3	MATH	Enrolled
4	Brooks, Angel Steven	522-72-9920	BPA	3	MNGG	Enrolled
5	Brown, Marquette Jon-J	222-93-9222	HUM	2	ARTG	Enrolled
6	Burbach, Timothy John	503-99-3999	NS	2	PENG	Enrolled
7	Carranza, Rafael	577-92-5532	BPA	2	MRTG	Enrolled
7	Denson, Tommie Hush	225-03-2299	NS	7	MATH	Enrolled
9	Fajardo, Sandra	222-32-5070	NS	3	MATH	Enrolled
10	Garcia, Rolando Antoni	227-03-7993	NS	2	CSCI	Enrolled

(6)

Figure 3.2. Demonstration of Create Account (Continued)



(7)

Figure 3.2. Demonstration of Create Account (Continued)

3.4.3 Enable Account

Figure 3.3 demonstrates enable account function. Figure 3.3 (1) and Figure 3.3 (2) is showing disabled accounts in student and generic groups. Figure 3.3 (1) shows that there are seven accounts which are disabled in student and generic groups. Figure 3.3 (2) shows the results of enable account function. Because the user selected all the seven accounts, WebSAT enabled these seven accounts. Figure 3.3 (3) shows that there are two accounts which are disabled by the systems administrator. Figure 3.3 (4) shows the result of enable account. Because the user selected only one account in the list, WebSAT enabled only that account.

WebSAT - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://websat.csci.csusb.edu:8080/enable_list.php

WebSAT System Department of Computer Science at CSUSB

Name: Juyong Jeong

Candidate Accounts To Be Enabled (Student and Generic)

Check all

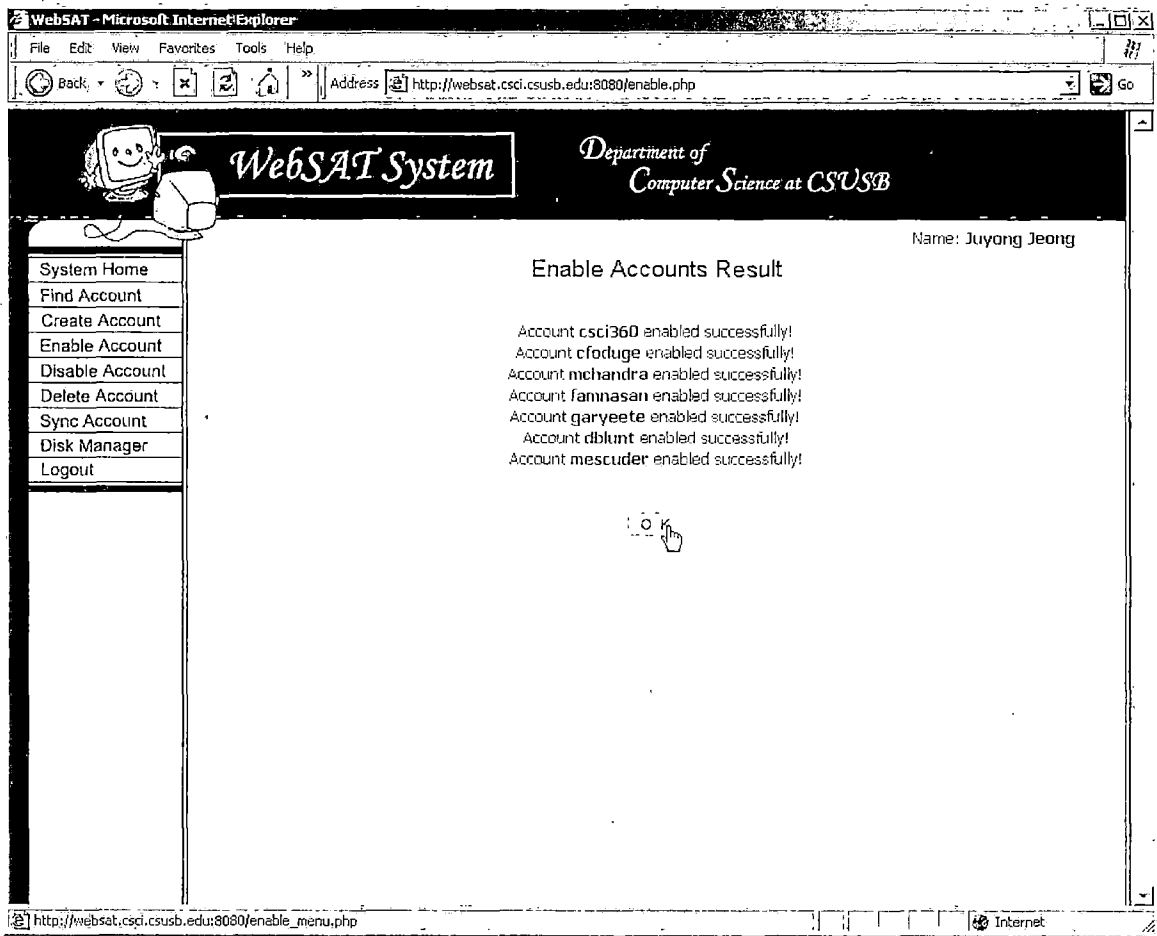
Enable	Login	Name	Group	Created	Last Log	Disabled
<input checked="" type="checkbox"/>	csci360	generic csci360	generic	2005-01-22	2005-01-22	2005-02-09
<input checked="" type="checkbox"/>	cfoduge	Chris Foduge	student	2005-01-22	2005-01-22	2005-02-09
<input checked="" type="checkbox"/>	mchandra	Mirta Chandravadia	student	2005-02-09	2005-02-09	2005-02-09
<input checked="" type="checkbox"/>	famnasan	Felicia Amnasan	student	2005-02-09	2005-02-09	2005-02-09
<input checked="" type="checkbox"/>	garyeete	Gregory Aryeete	student	2005-02-09	2005-02-09	2005-02-09
<input checked="" type="checkbox"/>	dblunt	Dhamish Blunt	student	2005-02-09	2005-02-09	2005-02-09
<input checked="" type="checkbox"/>	mescuder	Mirta Escudero	student	2005-02-09	2005-02-09	2005-02-09

Total: 7 account(s) found.

System Home
Find Account
Create Account
Enable Account
Disable Account
Delete Account
Sync Account
Disk Manager
Logout

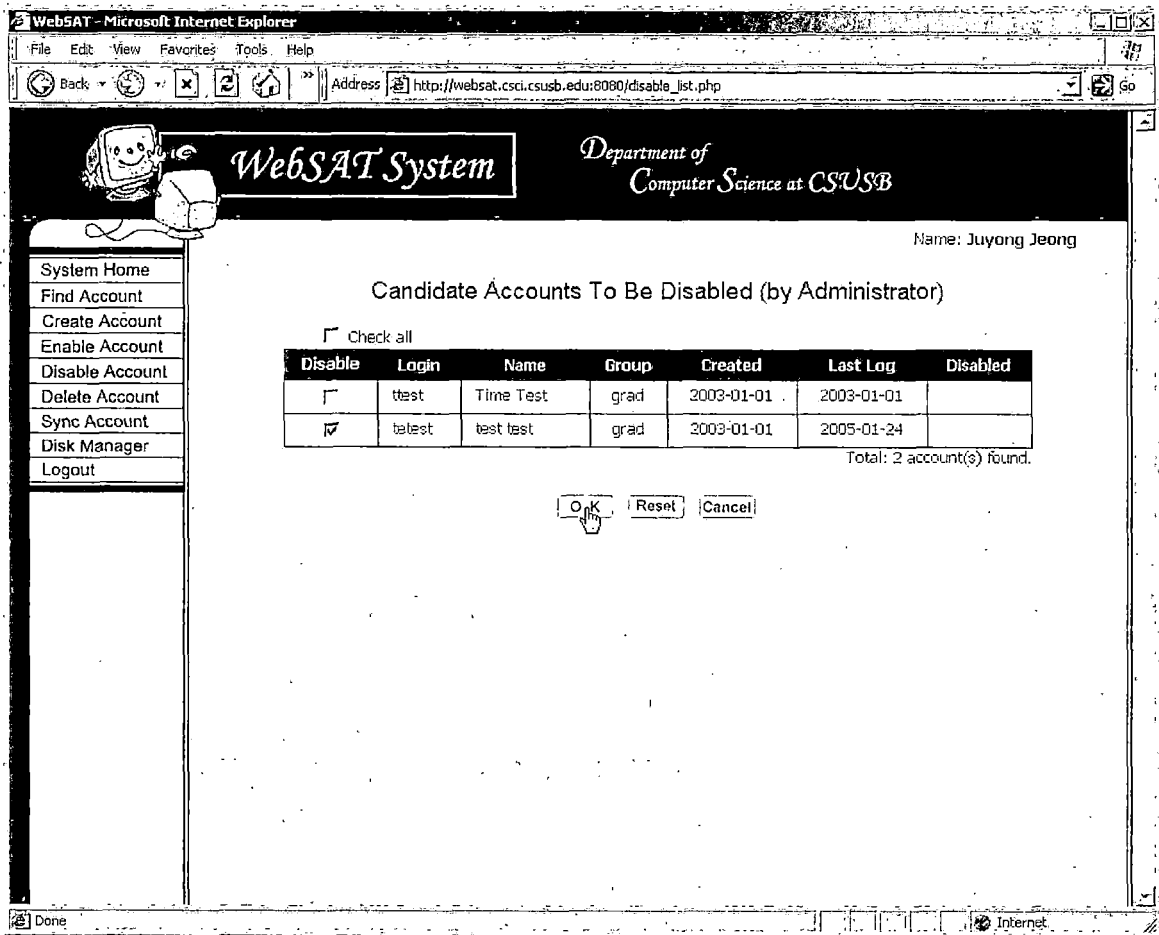
(1)

Figure 3.3. Demonstration of Enable Account



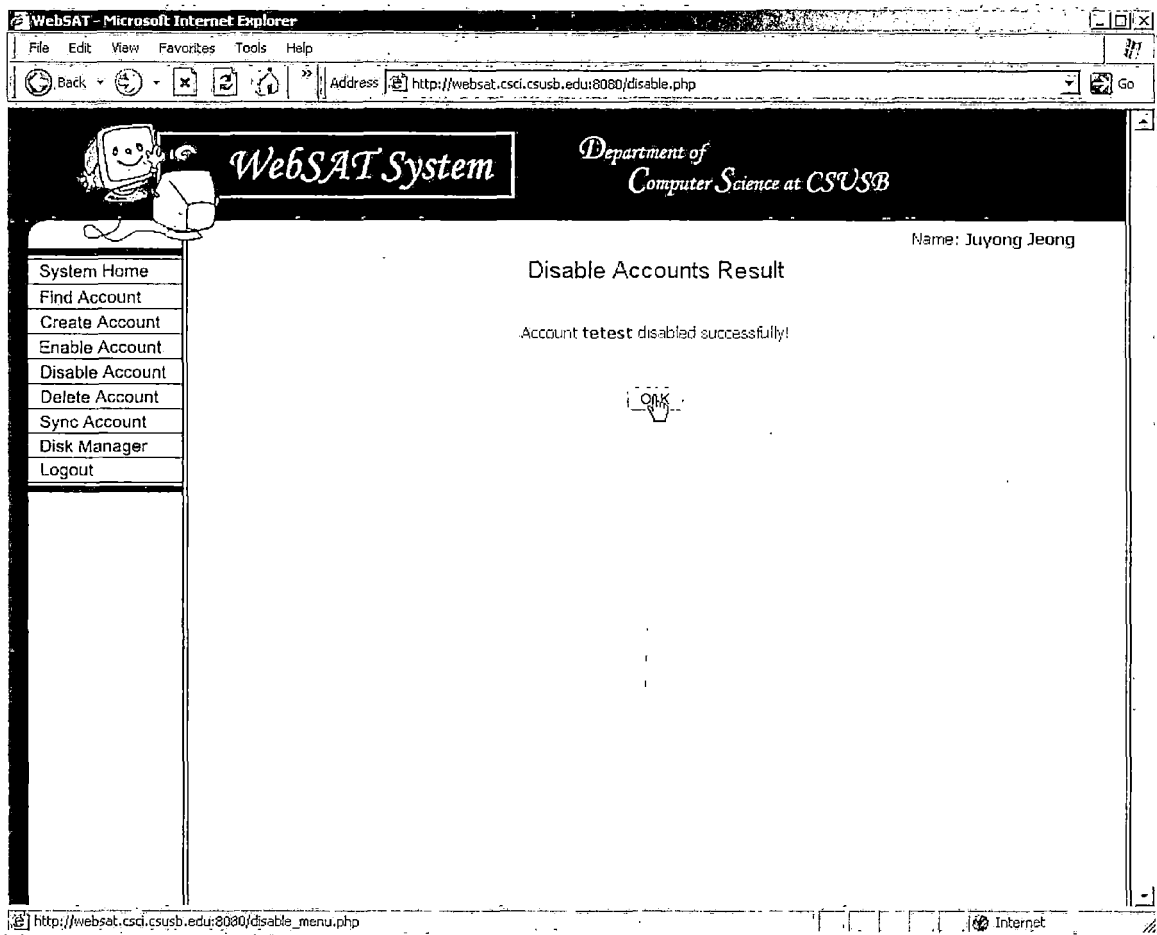
(2)

Figure 3.3. Demonstration of Enable Account (Continued)



(3)

Figure 3.3. Demonstration of Enable Account (Continued)



(4)

Figure 3.3. Demonstration of Enable Account (Continued)


3.4.4 Disable Account

Figure 3.4 demonstrates disable account function. Figure 3.4 (1) displays all accounts in the student and generic groups. The user selected 13 logins which are in student group, and WebSAT disabled these 13 accounts and displays the results in Figure 3.4 (2). Figure 3.4 (3) shows that there are two accounts which were accessed at least one year ago. The user selected one login out of two, and WebSAT disabled that account. Figure 3.4 (5) shows the disable account form. The user wants to see all accounts which were accessed January 1, 2005 or before that time for the last time, and Figure 3.4 (6), displays an account which was accessed January 1, 2003 for the last time. The user selects the account and clicks on O.K. button. Figure 3.4 (7) shows the results of disable account.

WebSAT - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://websat.csci.csusb.edu:8080/disable_exp_list.php?order_by=name&sort= Go



WebSAT System

Department of
Computer Science at CSUSB

Name: Juyong Jeong

System Home

Find Account

Create Account

Enable Account

Disable Account

Delete Account

Sync Account

Disk Manager

Logout

Candidate Accounts To Be Disabled (Student & Generic Group)

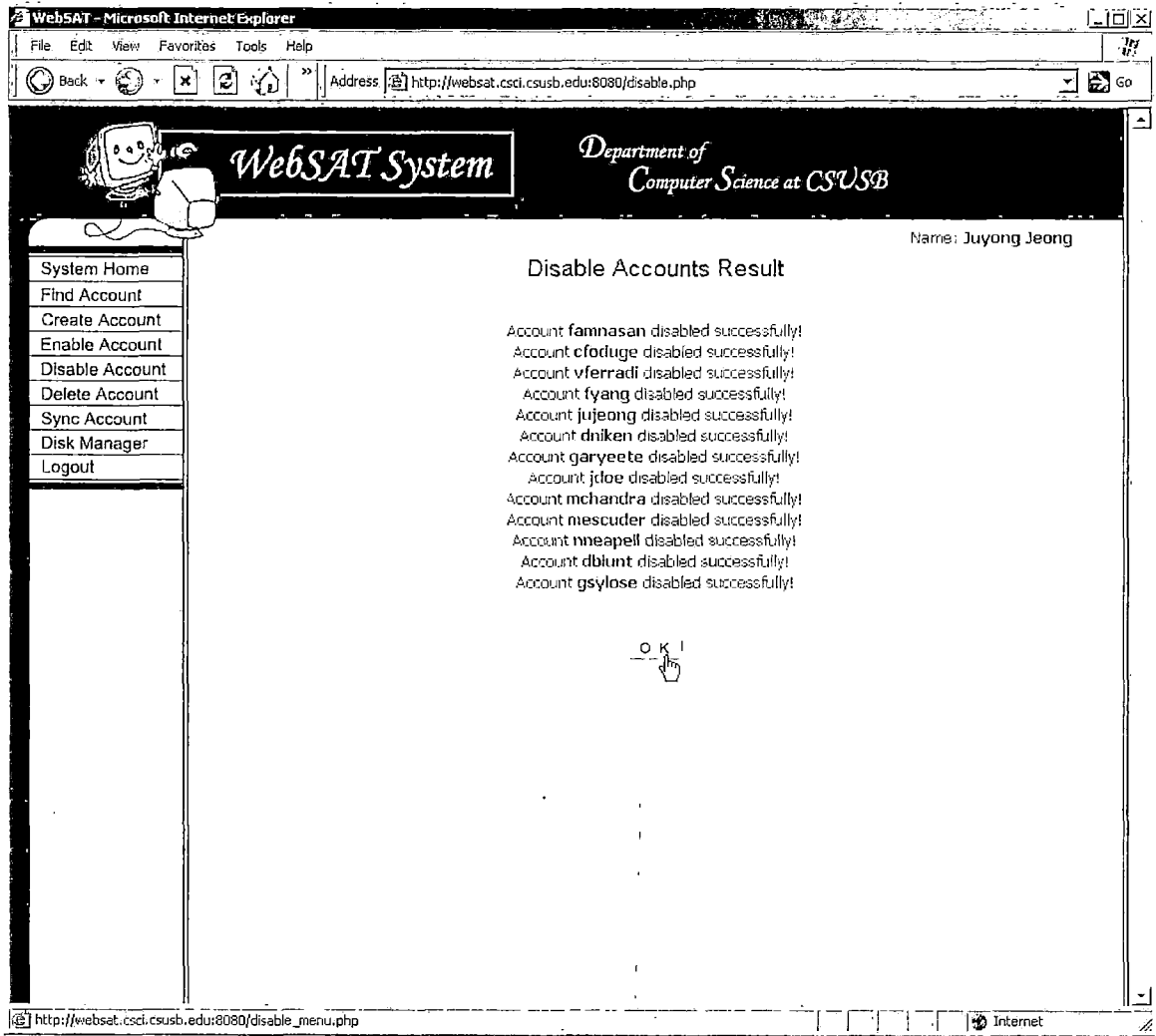
Check all

Disable	Login	Name	Group	Created	Last Log	Disabled
<input type="checkbox"/>	csci303	CSCI303 Generic	generic	2005-02-09	2005-02-09	
<input type="checkbox"/>	csci360	generic csci360	generic	2005-01-22	2005-01-22	
<input type="checkbox"/>	csci202	CSCI202 Generic	generic	2005-02-09	2005-02-09	
<input checked="" type="checkbox"/>	famnasan	Fellchia Amnasan	student	2005-02-09	2005-02-09	
<input checked="" type="checkbox"/>	cfoduge	Chris Foduge	student	2005-01-22	2005-01-22	
<input checked="" type="checkbox"/>	vferradi	Vincent Ferradino	student	2005-02-09	2005-02-09	
<input checked="" type="checkbox"/>	fyang	Faiya Yang	student	2005-02-09	2005-02-09	
<input checked="" type="checkbox"/>	jyjeong	Juyong Jeong	student	2005-02-09	2005-02-09	
<input checked="" type="checkbox"/>	dniken	David Niken	student	2005-02-09	2005-02-09	
<input checked="" type="checkbox"/>	garyeeta	Gregory Aryeetey	student	2005-02-09	2005-02-09	
<input checked="" type="checkbox"/>	jdse	Johnson Doe	student	2005-02-09	2005-02-09	
<input checked="" type="checkbox"/>	mchandra	Mirta Chandravadia	student	2005-02-09	2005-02-09	
<input checked="" type="checkbox"/>	mescuder	Mirta Escudero	student	2005-02-09	2005-02-09	
<input checked="" type="checkbox"/>	nneapell	Natali Neapell	student	2005-02-09	2005-02-09	
<input checked="" type="checkbox"/>	tblunt	Dhamish Blunt	student	2005-02-09	2005-02-09	
<input checked="" type="checkbox"/>	gsylose	Green Sylosa	student	2005-02-09	2005-02-09	

Total: 16 account(s) found.

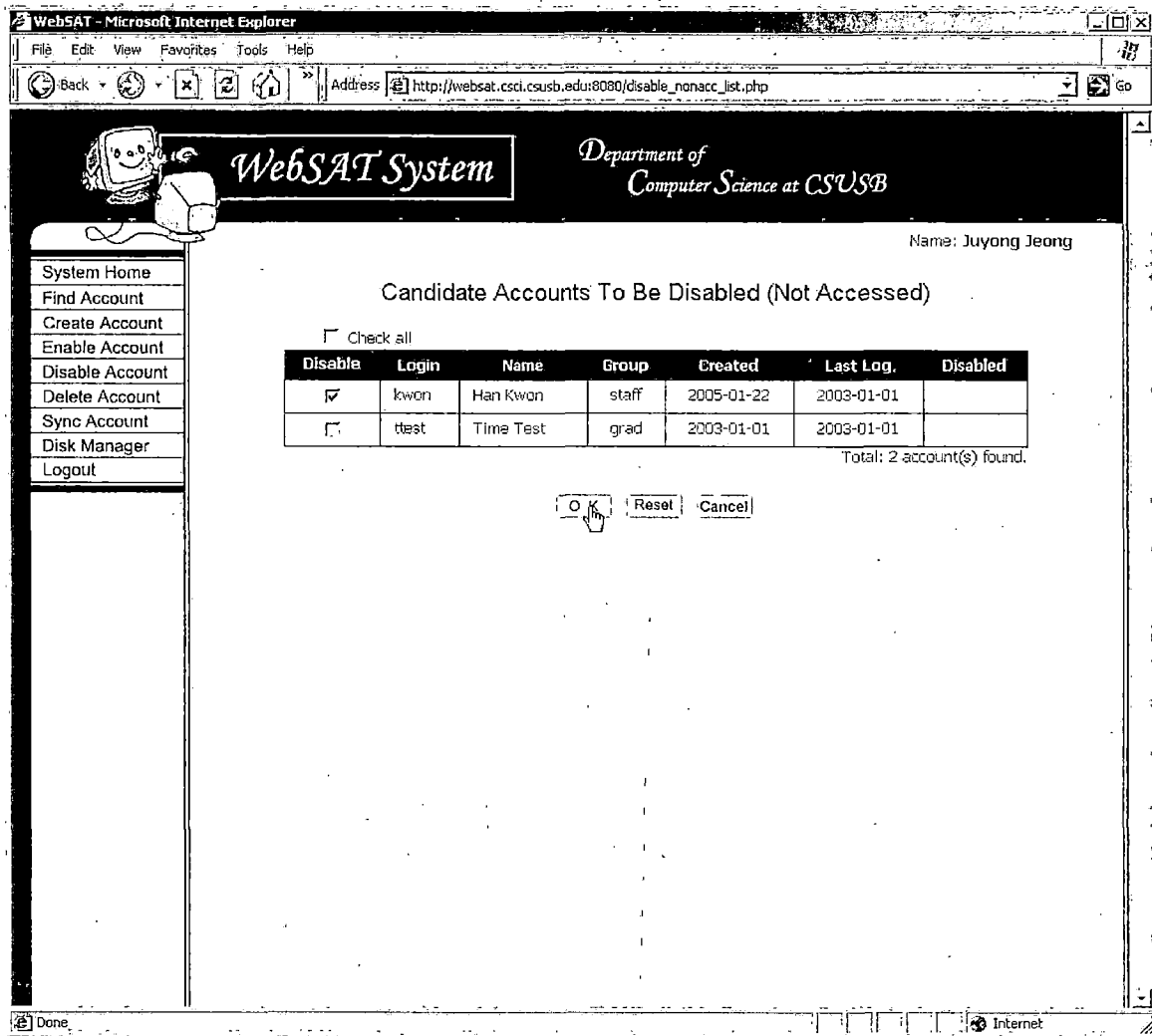
(1)

Figure 3.4. Demonstration of Disable Account



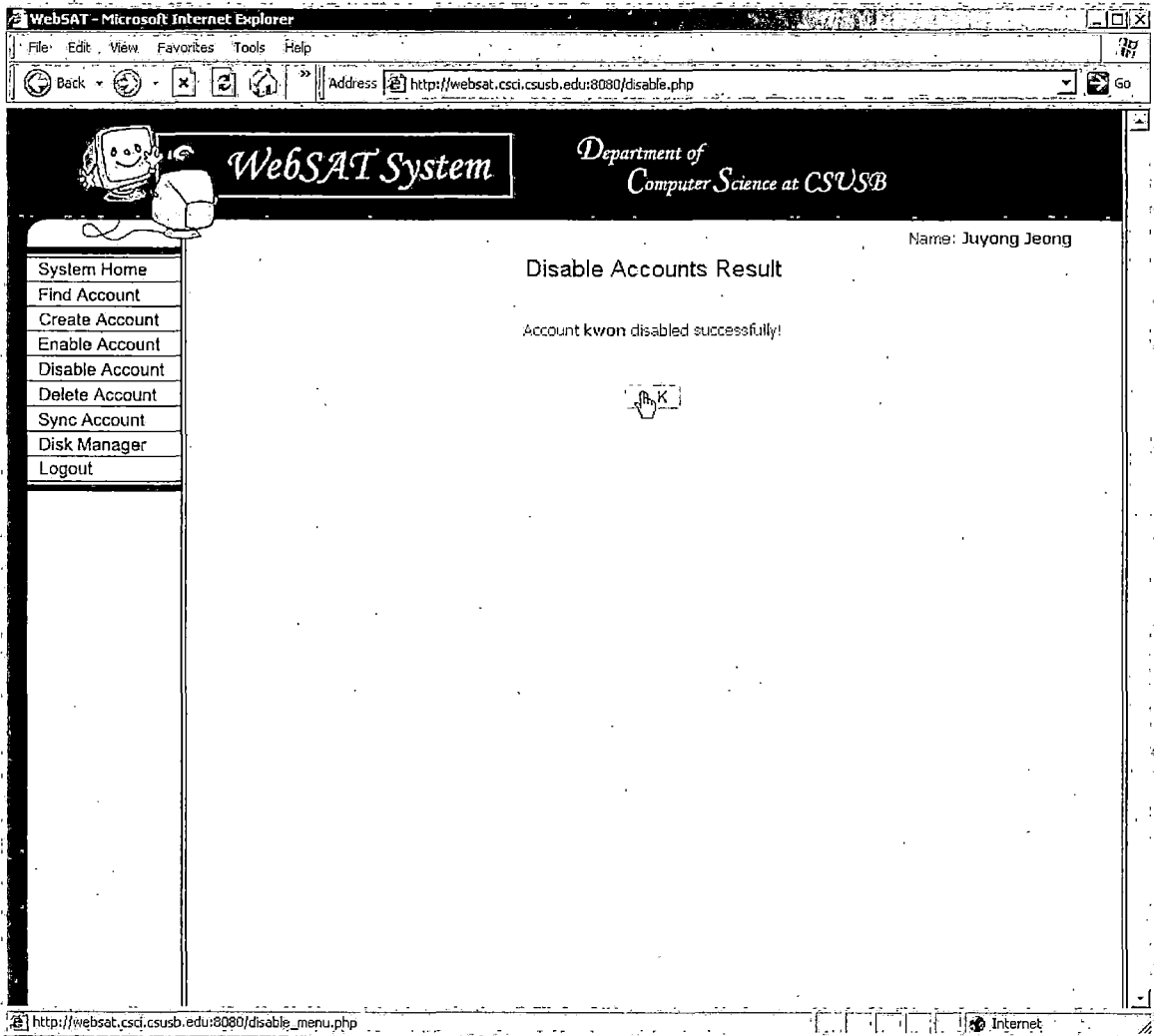
(2)

Figure 3.4. Demonstration of Disable Account (Continued)



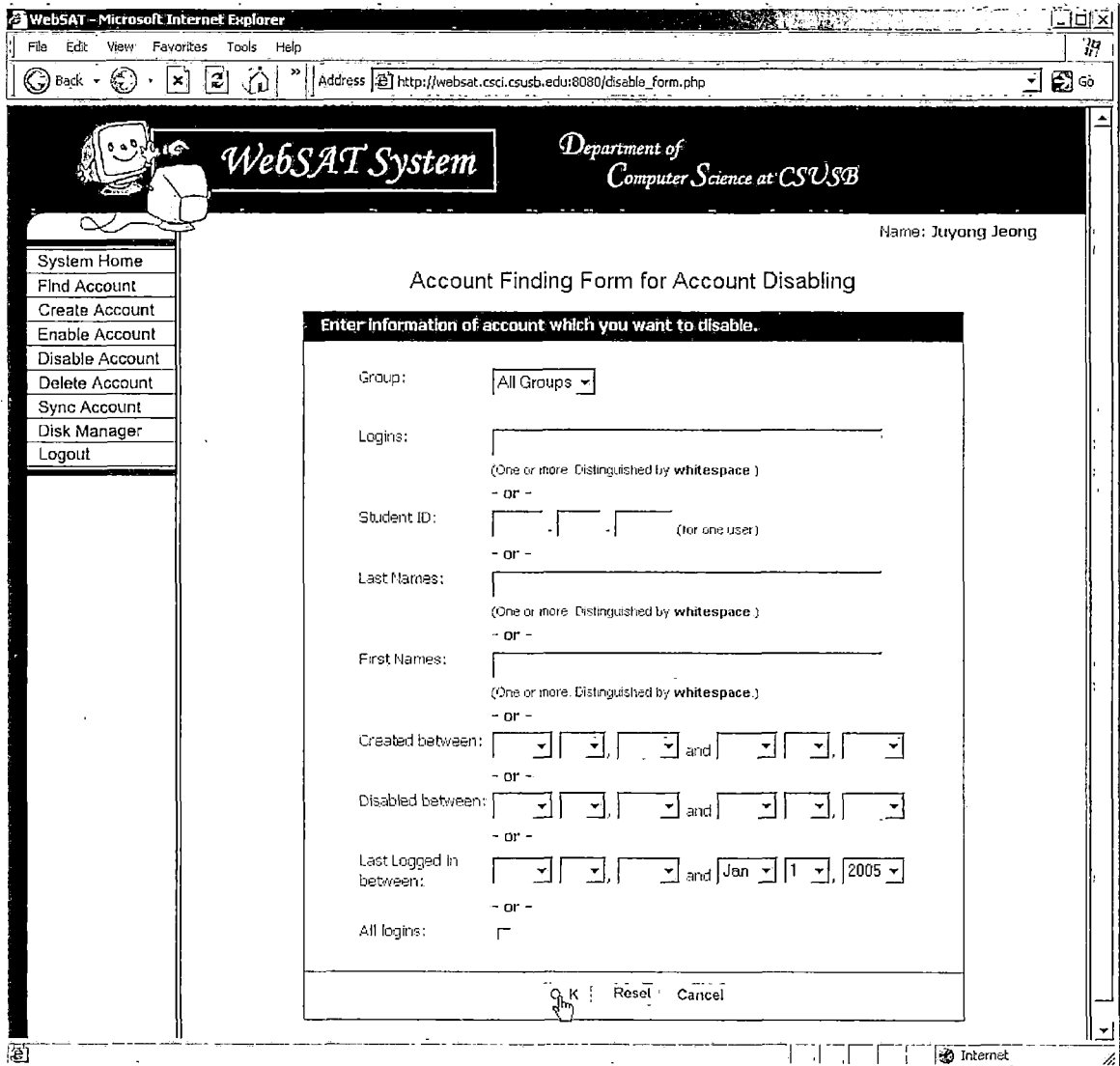
(3)

Figure 3.4. Demonstration of Disable Account (Continued)



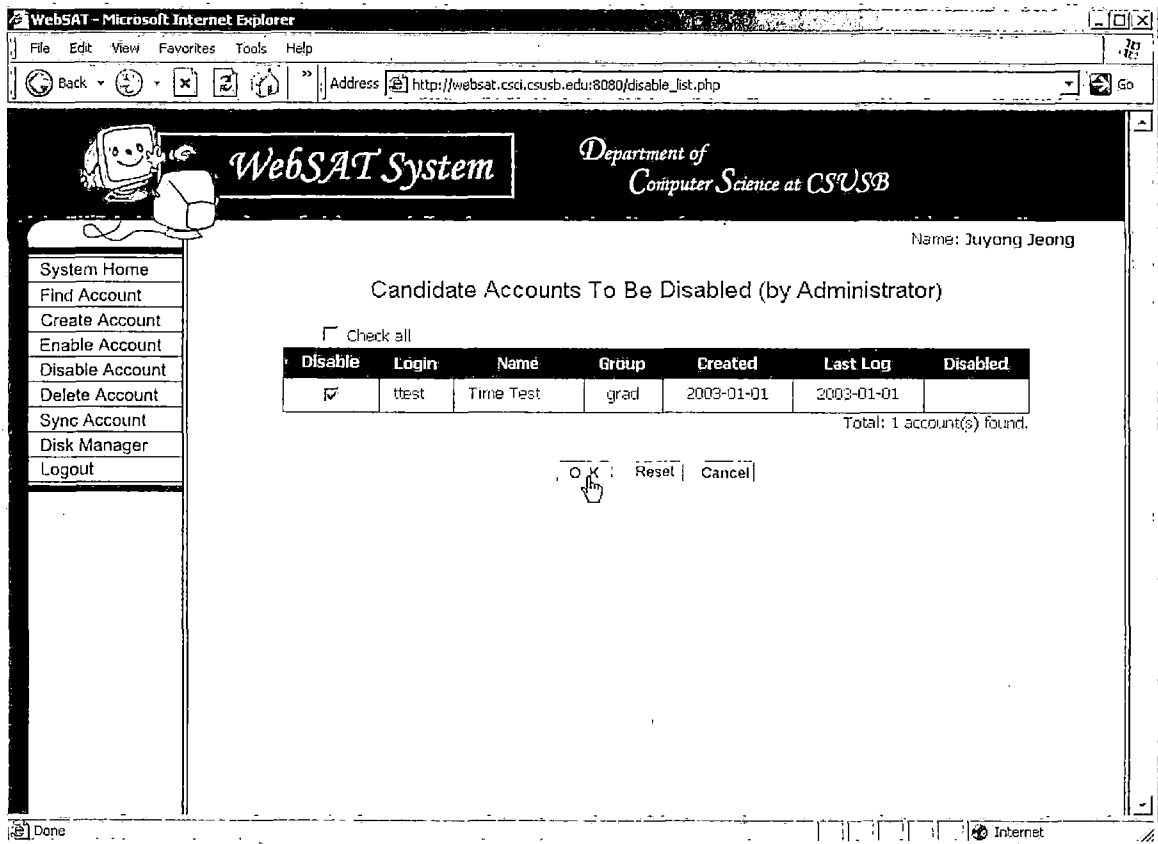
(4)

Figure 3.4. Demonstration of Disable Account (Continued)



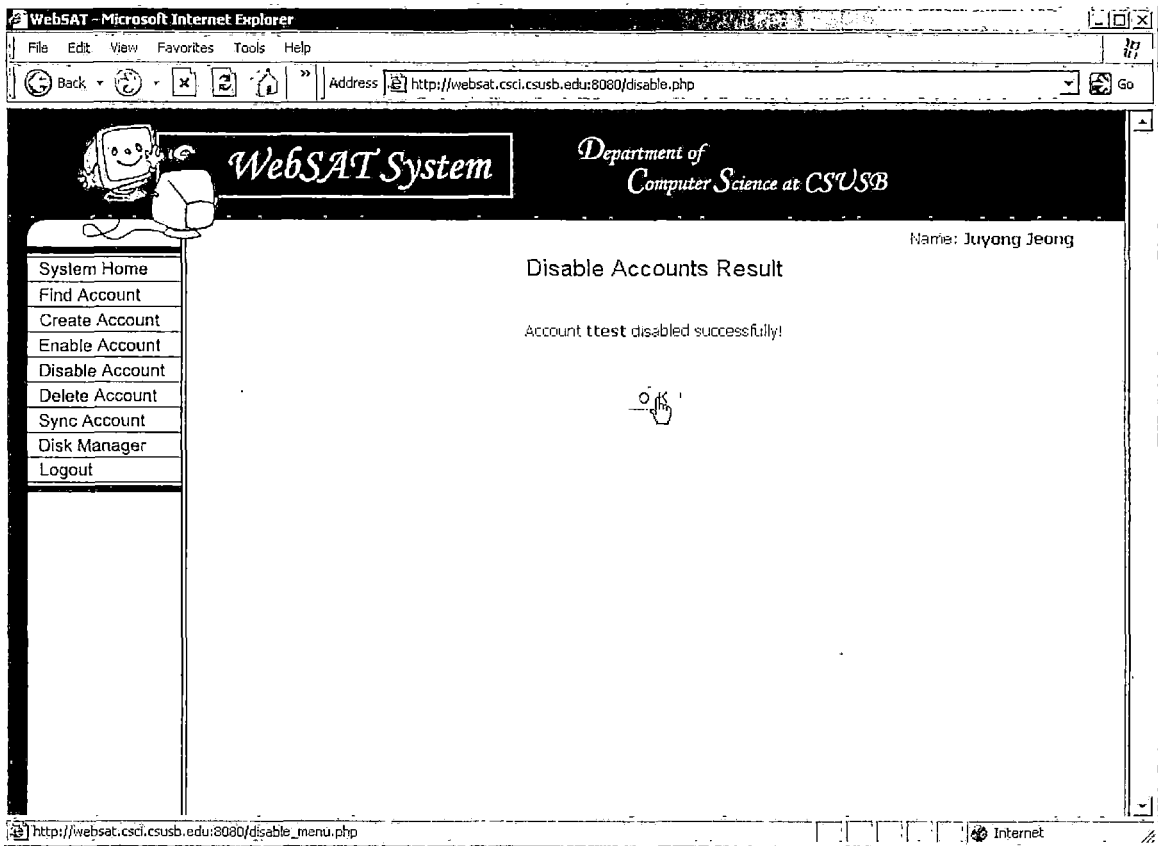
(5)

Figure 3.4. Demonstration of Disable Account (Continued)



(6)

Figure 3.4. Demonstration of Disable Account (Continued)



(7)

Figure 3.4. Demonstration of Disable Account (Continued)

3.4.5 Delete Account

Figure 3.5 demonstrates delete account function. Figure 3.5 (1) displays all student group accounts. The user selected first three accounts out of 13 accounts. When the user click on O.K. button to delete those three accounts, the system pops up a message box to make sure the action (see Figure 3.5 (2)). The user clicks on O.K. button on the message box, and WebSAT deletes selected accounts and return the result (see Figure 3.5 (3)). Figure 3.5 (4) displays all generic group accounts. The user selected one account from the list, and WebSAT deleted that account from the system and database. Figure 3.5. (5) displays the result of deletion.

Figure 3.5 (6) shows the delete account form. The user wants to see all accounts in student group, and Figure 3.5 (7) displays a list of student group accounts. The user selects one account in the list and clicks on O.K. button. Figure 3.5 (8) shows the result of delete account.

WebSAT - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://websat.csci.csusb.edu:8080/del_std_list.php?order_by=l_date&sort=0

WebSAT System Department of Computer Science at CSUSB

Name: Juyong Jeong

System Home
Find Account
Create Account
Enable Account
Disable Account
Delete Account
Sync Account
Disk Manager
Logout

Candidate Accounts To Be Deleted (Student Group)

Check all

Delete	Login	Name	Group	Created	Last Log	Disabled
<input checked="" type="checkbox"/>	cfoduge	Chris Foduge	student	2005-01-22	2005-01-22	2005-02-09
<input checked="" type="checkbox"/>	mescuador	Mirta Escudero	student	2005-02-09	2005-02-09	2005-02-09
<input checked="" type="checkbox"/>	diblunt	Dhamish Blunt	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	garyeete	Gregory Aryeete	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	famnasan	Felichia Amnasan	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	mchandra	Mirta Chandravadia	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	jujeong	Juyong Jeong	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	fyang	Faiya Yang	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	nneapell	Natali Neapell	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	dniken	David Niken	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	gsylose	Green Sylose	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	jdoe	Johnson Doe	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	vierradi	Vincent Ferradino	student	2005-02-09	2005-02-09	2005-02-09

Total: 13 account(s) found.

Internet

(1)

Figure 3.5. Demonstration of Delete Account

WebSAT - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://websat.csci.csusb.edu:8080/del_std_list.php?order_by=_date&sort=0

WebSAT System Department of Computer Science at CSUSB

Name: Juyong Jeong

System Home
Find Account
Create Account
Enable Account
Disable Account
Delete Account
Sync Account
Disk Manager
Logout

Candidate Accounts To Be Deleted (Student Group)

Check all

Delete	Login	Name	Group	Created	Last Log	Disabled
<input checked="" type="checkbox"/>	cfoduge	Chris Foduge	student	2005-01-22	2005-01-22	2005-02-09
<input checked="" type="checkbox"/>	mescurier	Mirta Escudero	student	2005-02-09	2005-02-09	2005-02-09
<input checked="" type="checkbox"/>	dblunt	Dhamish Blunt	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	garyeete	Gregory Aryeete	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	famnasan	Felchia Amnasan	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	mchandra	Mirta Chandravadia	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	jujaong	Juyong Jeong	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	lyang	Faiya Yang	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	nneapell	Natali Neapell	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	dniken	David Niken	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	gsvlose	Green Syllose	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	jdoe	Johnson Doe	student	2005-02-09	2005-02-09	2005-02-09
<input type="checkbox"/>	vferradi	Vincent Ferradho	student	2005-02-09	2005-02-09	2005-02-09

Total: 13 account(s) found.

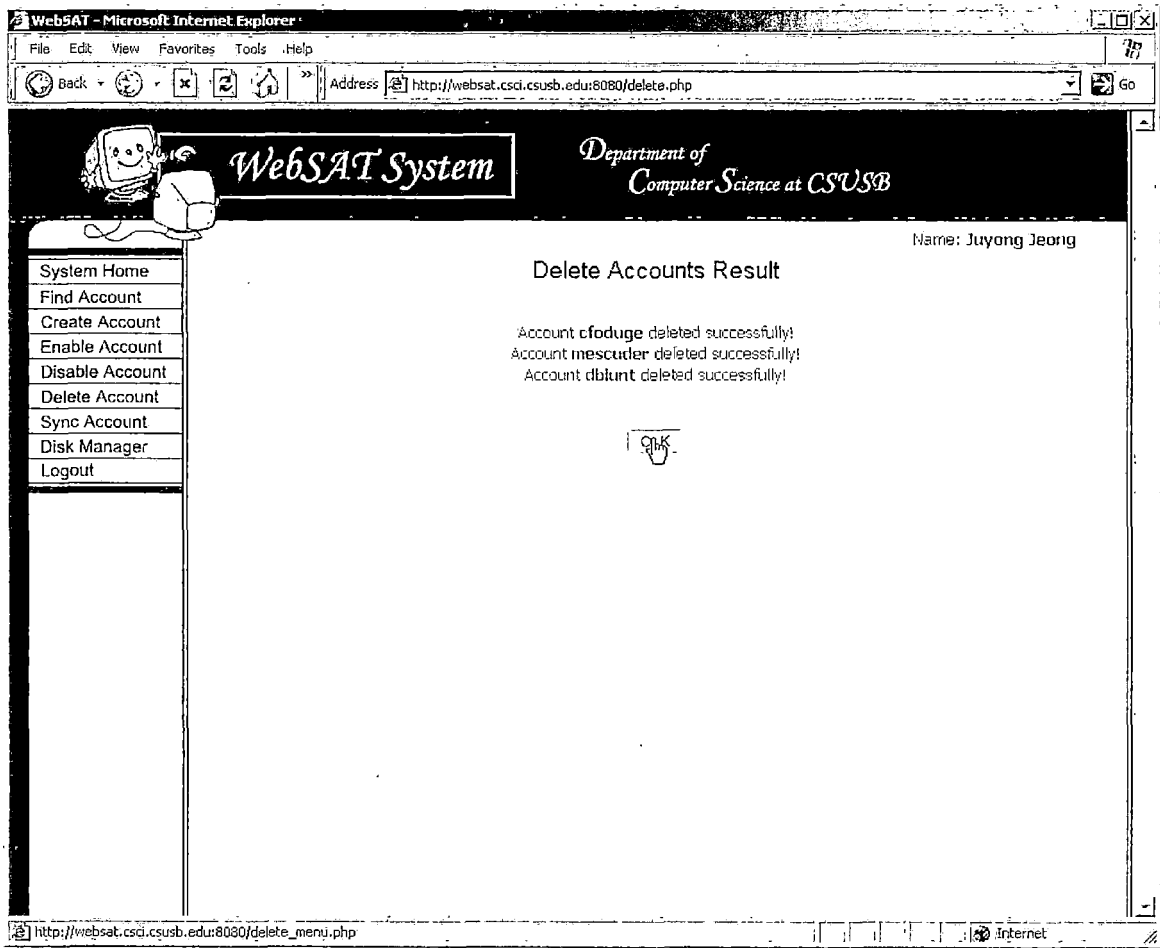
OK Reset Cancel

Are you sure you want to delete accounts?

OK Cancel

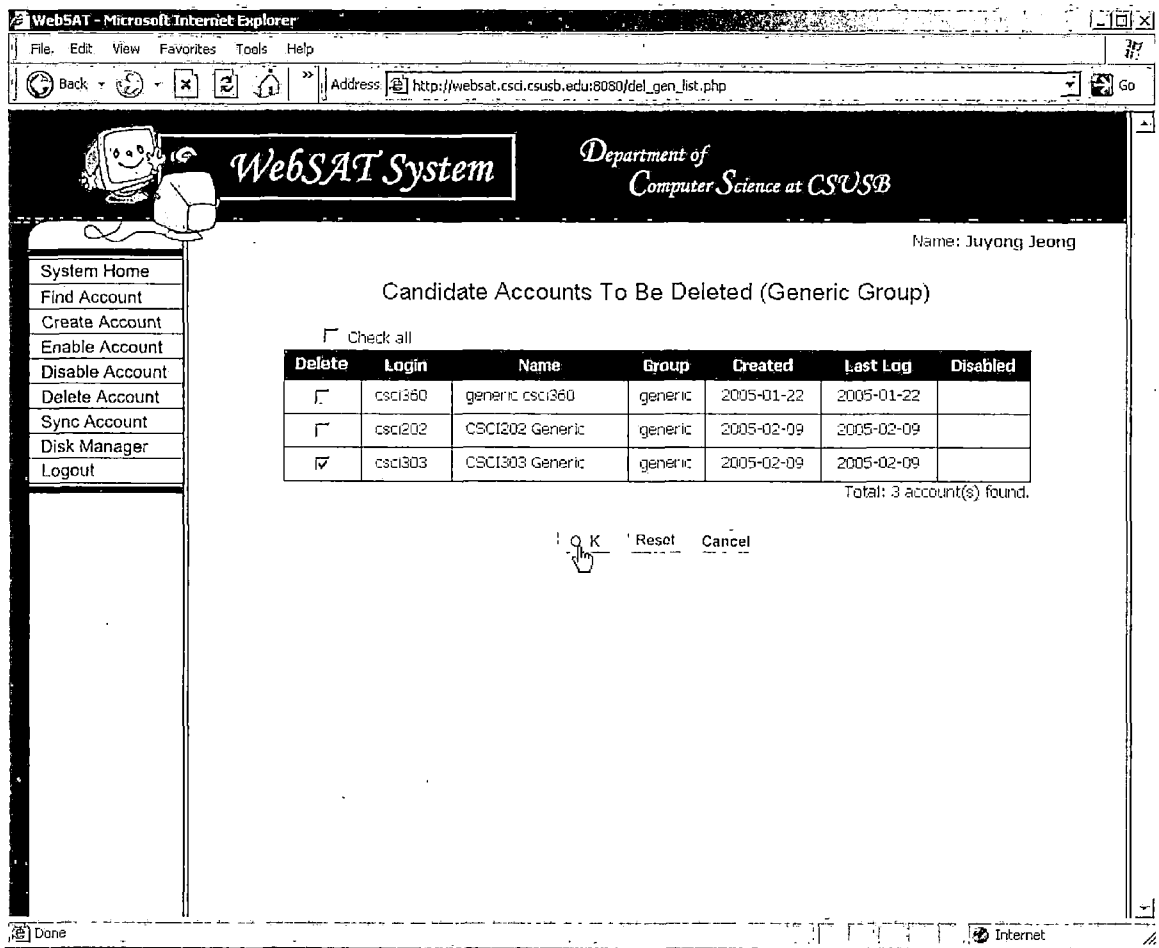
(2)

Figure 3.5. Demonstration of Delete Account (Continued)



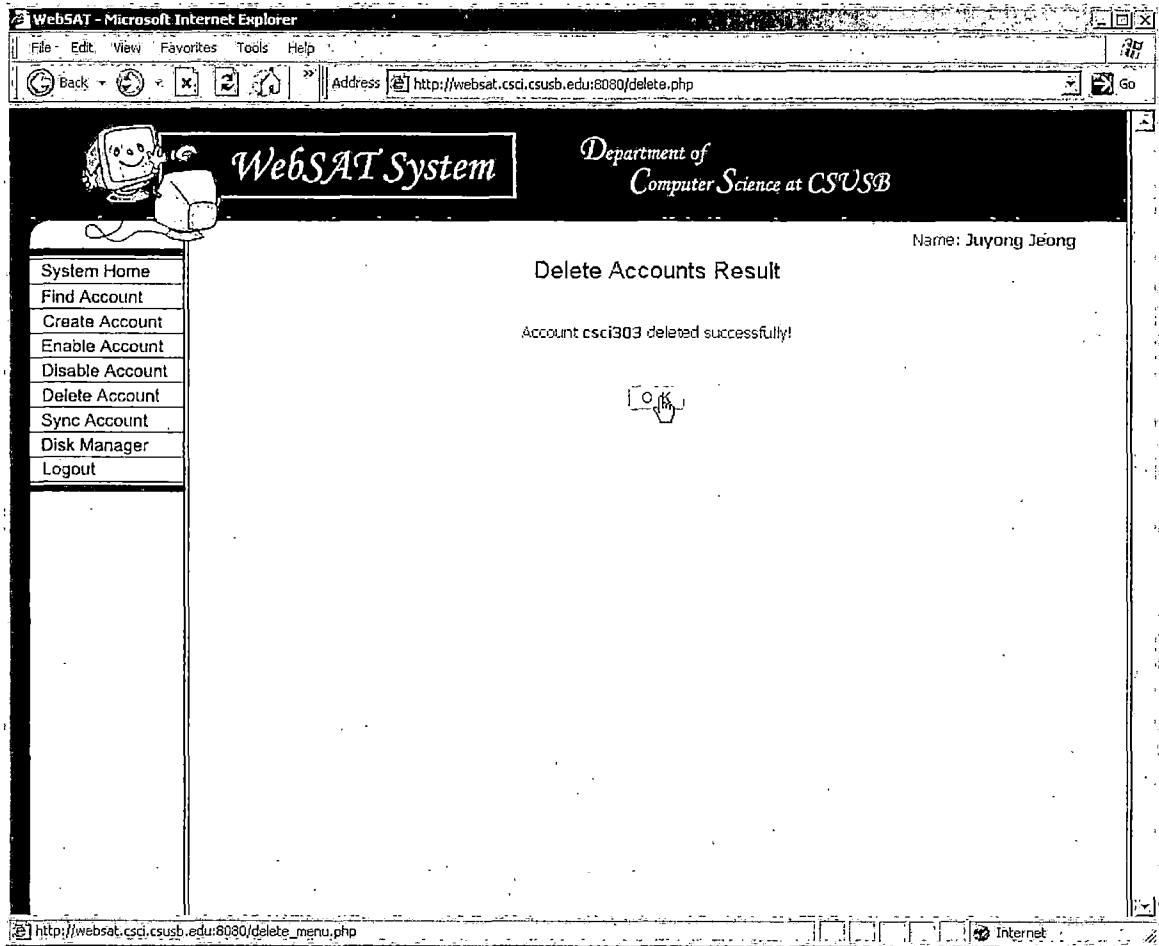
(3)

Figure 3.5. Demonstration of Delete Account (Continued)



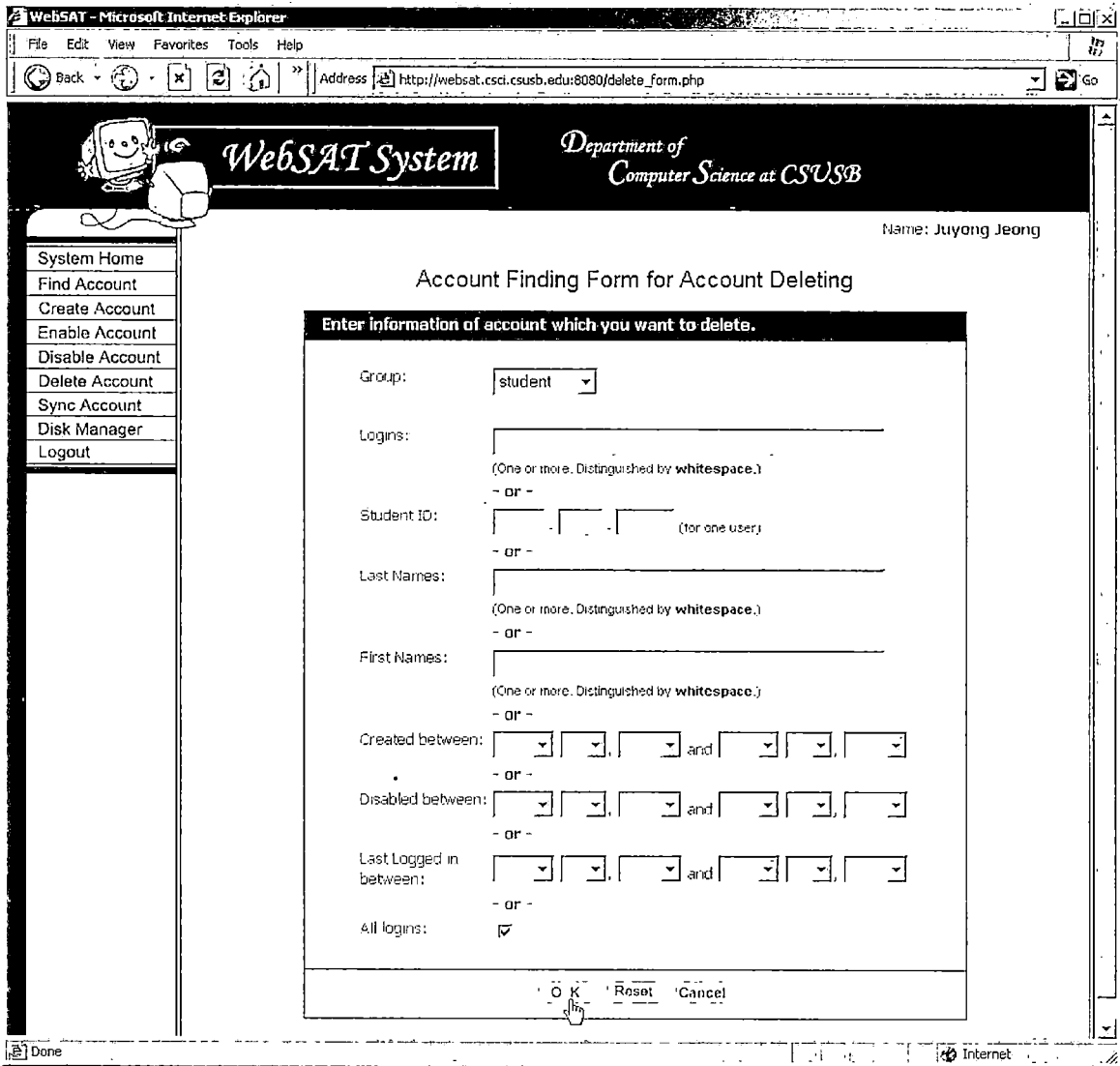
(4)

Figure 3.5. Demonstration of Delete Account (Continued)



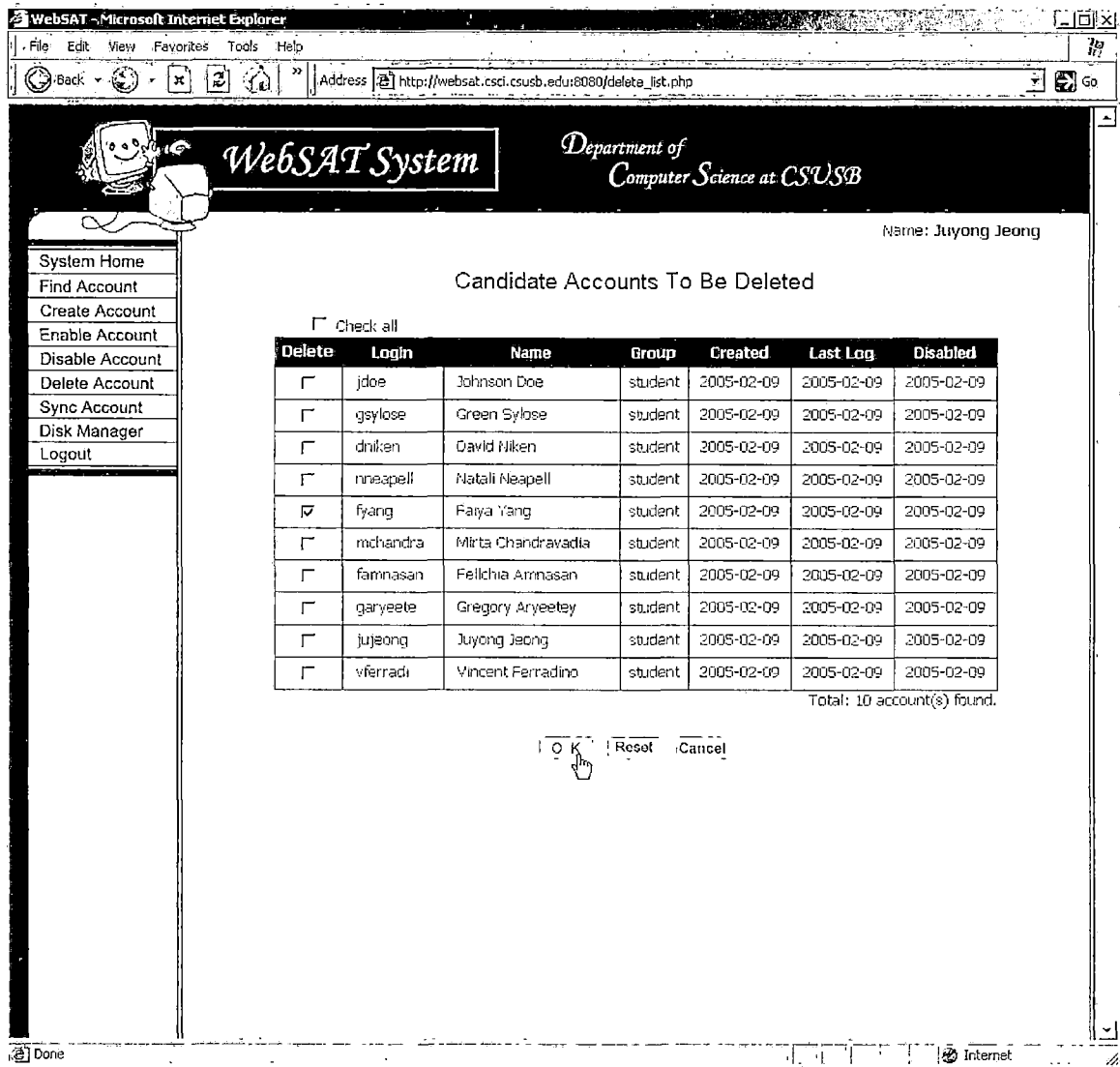
(5)

Figure 3.5. Demonstration of Delete Account (Continued)



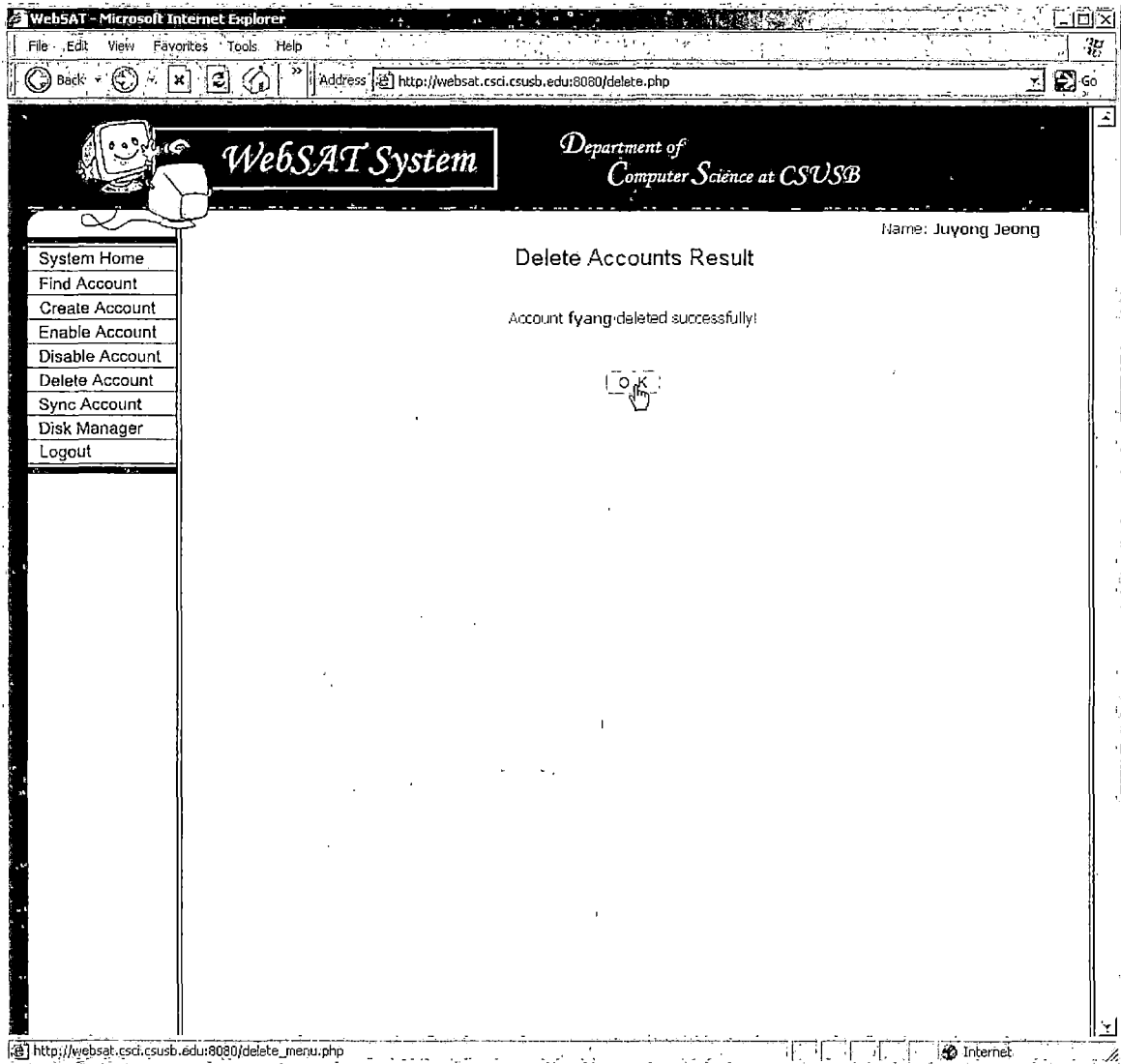
(6)

Figure 3.5. Demonstration of Delete Account (Continued)



(7)

Figure 3.5. Demonstration of Delete Account (Continued)



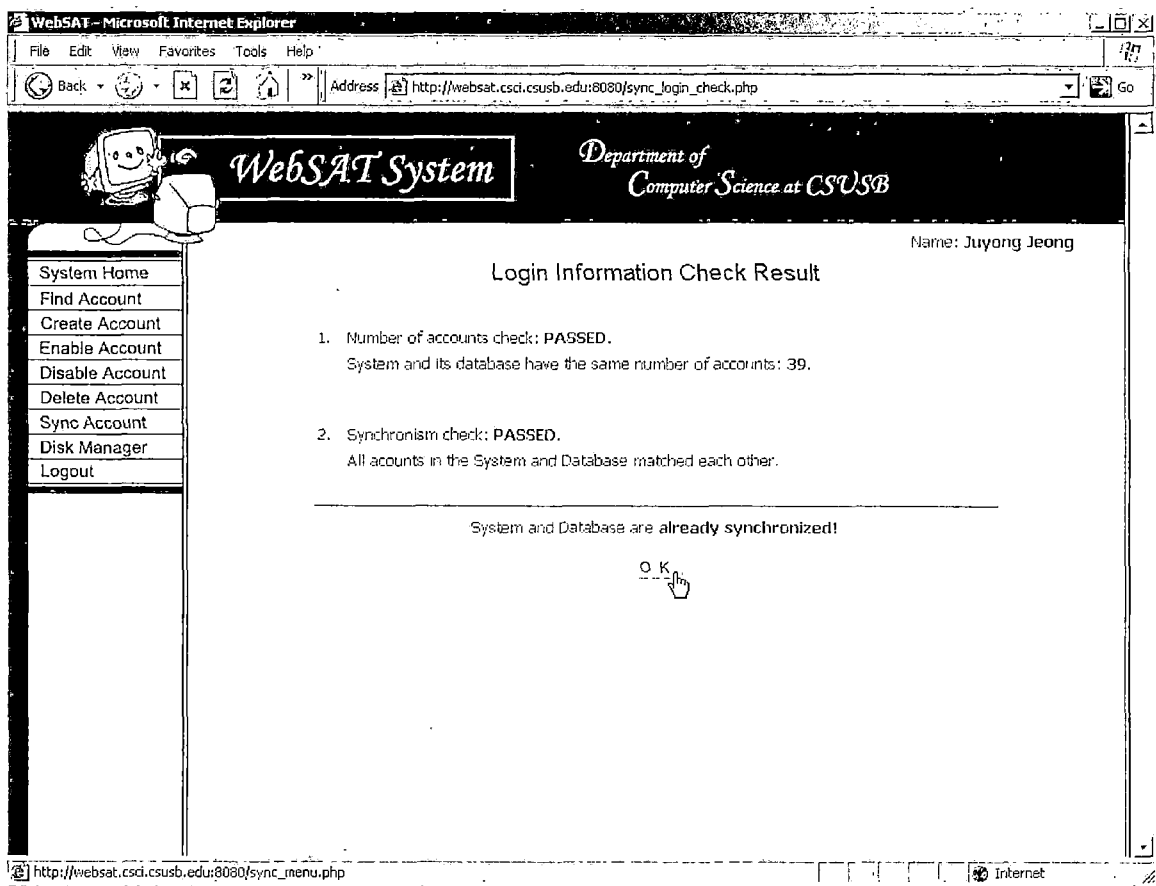
(8)

Figure 3.5. Demonstration of Delete Account (Continued)

3.4.6 Synchronize Account

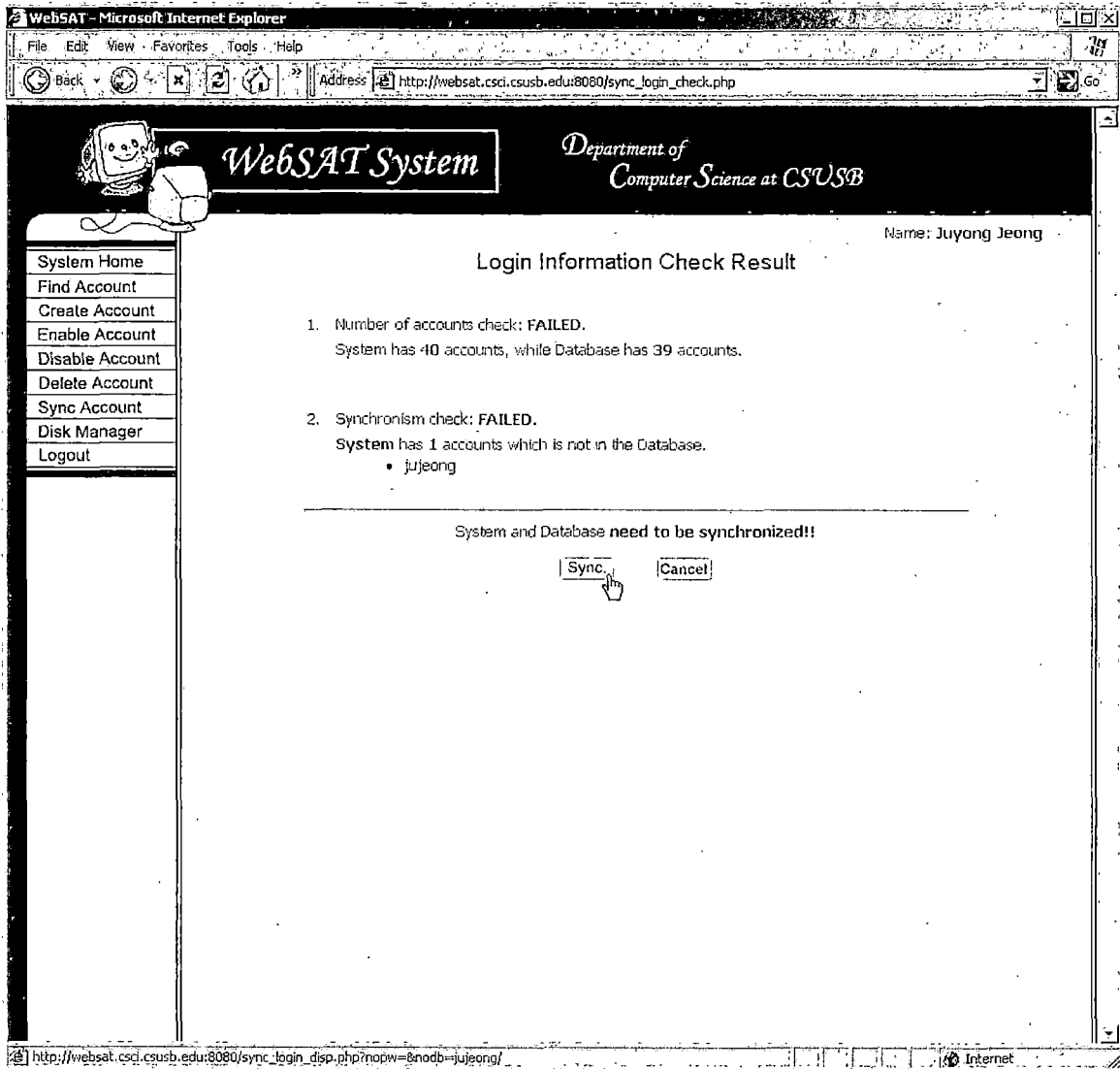
Figure 3.6 demonstrates synchronize account function. Figure 3.6 (1) - Figure 3.6 (4) show login information synchronization. The result of synchronized login information is showed in Figure 3.6 (1). On the other hand, Figure 3.6 (2) shows the result of login synchronization failure. If the user wants to synchronize login information, he/she clicks on Sync button, and WebSAT leads the user to synchronize account option page. There is one account which is not in the database. The user is prompted a message box as soon as he/she clicks on radio option button. The user wants the account to be deleted from the system, so he/she click on O.K. button on the message box, and system commits the action. This action is shown in Figure 3.6 (3), and Figure 3.6 (4) shows the result of account synchronization. Figure 3.6 (5) - Figure 3.6 (8) show group information synchronization. When group information is synchronized, WebSAT displays the result like Figure 3.6 (5). On the other hand, WebSAT displays group information failure message like Figure 3.6 (6). Figure 3.6 (7) shows group information synchronization form. All logins whose group information is not synchronized are shown in this form. Now,

WebSAT asks the user that he/she wants to change group information for Ben Qua from student group to csys group. Figure 3.6 (8) shows the result of group information change for Ben Qua. Figure 3.6 (8) shows that WebSAT updated last login date for all group accounts.



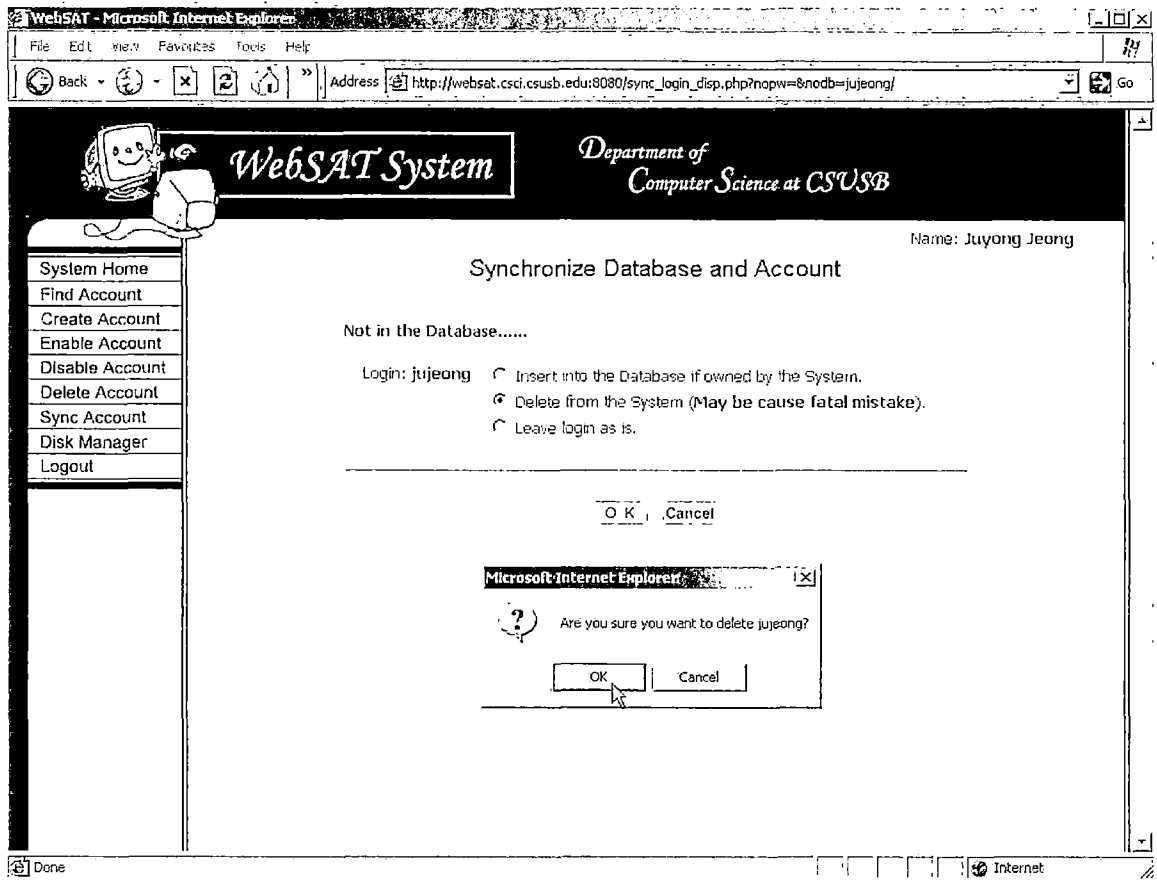
(1)

Figure 3.6. Demonstration of Synchronize Account



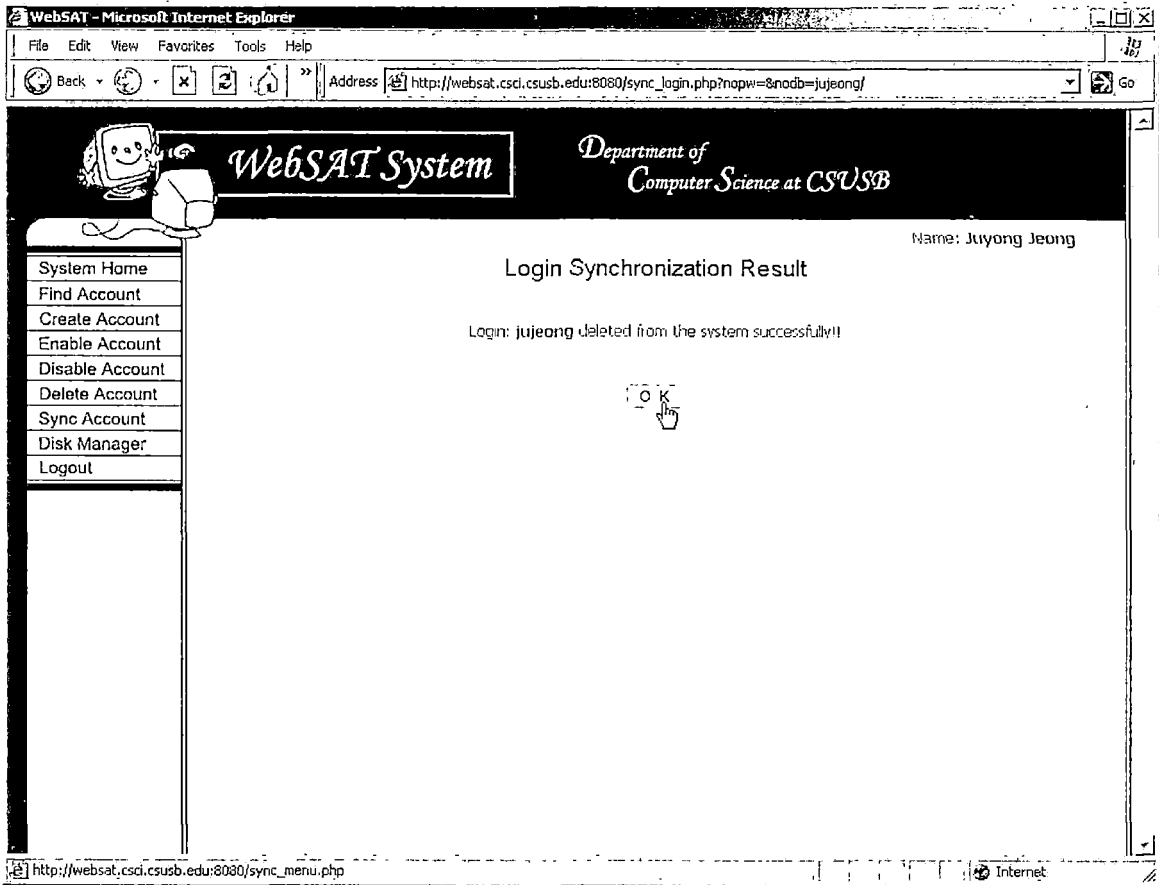
(2)

Figure 3.6. Demonstration of Synchronize Account
(Continued)



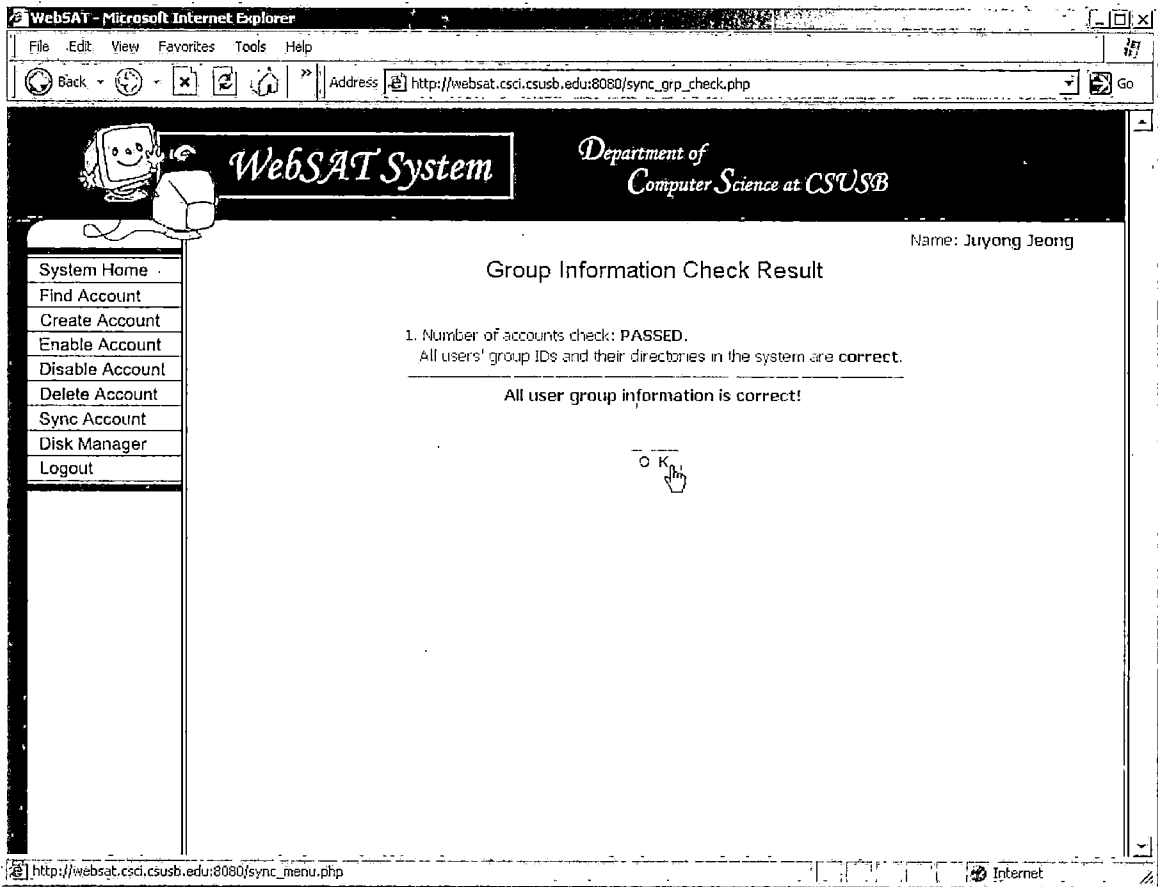
(3)

Figure 3.6. Demonstration of Synchronize Account
(Continued)



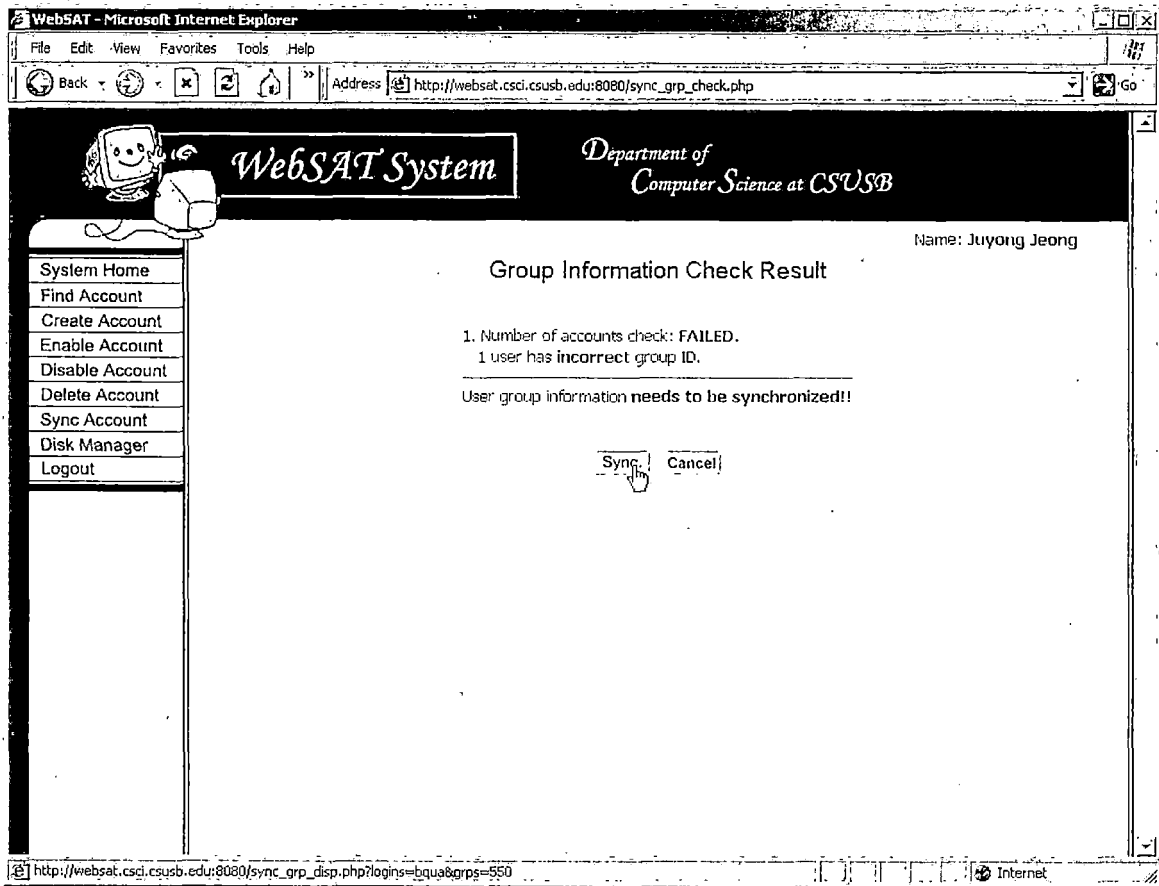
(4)

Figure 3.6. Demonstration of Synchronize Account
(Continued)



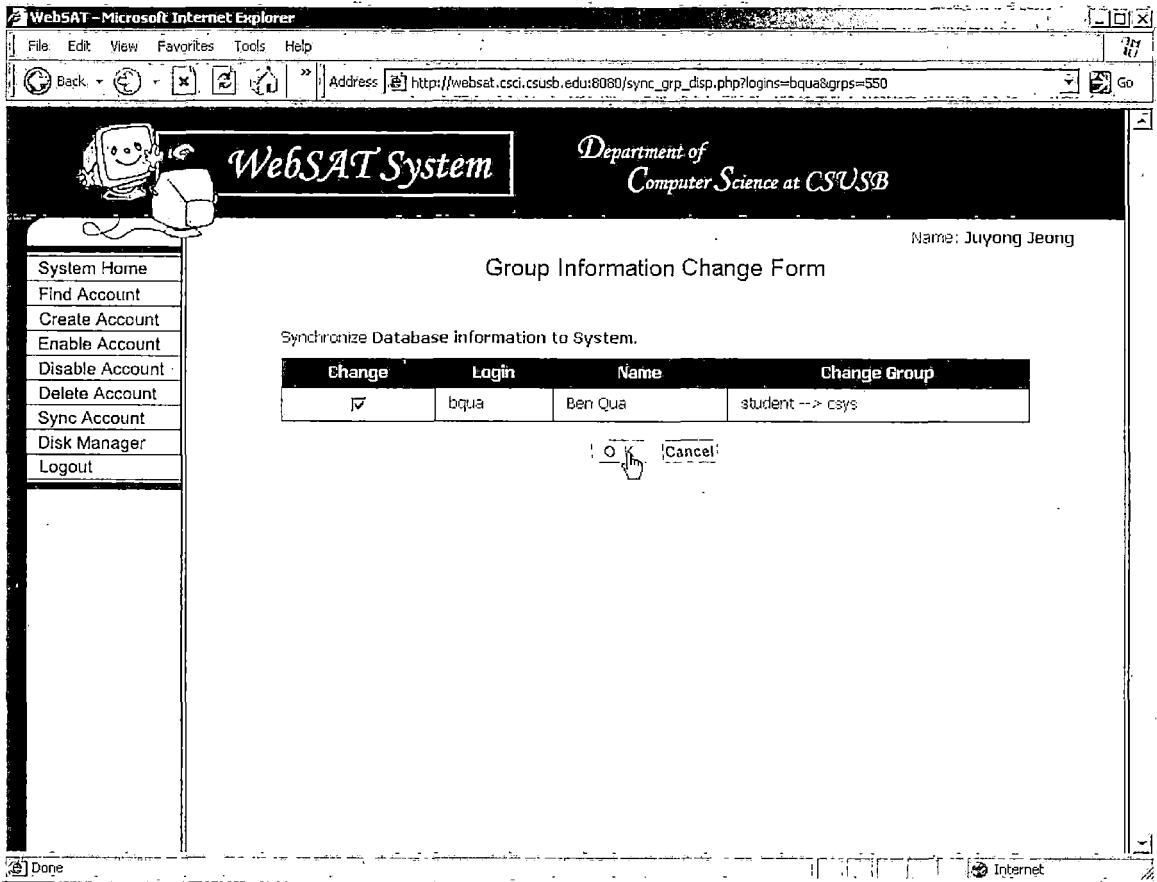
(5)

Figure 3.6. Demonstration of Synchronize Account
(Continued)



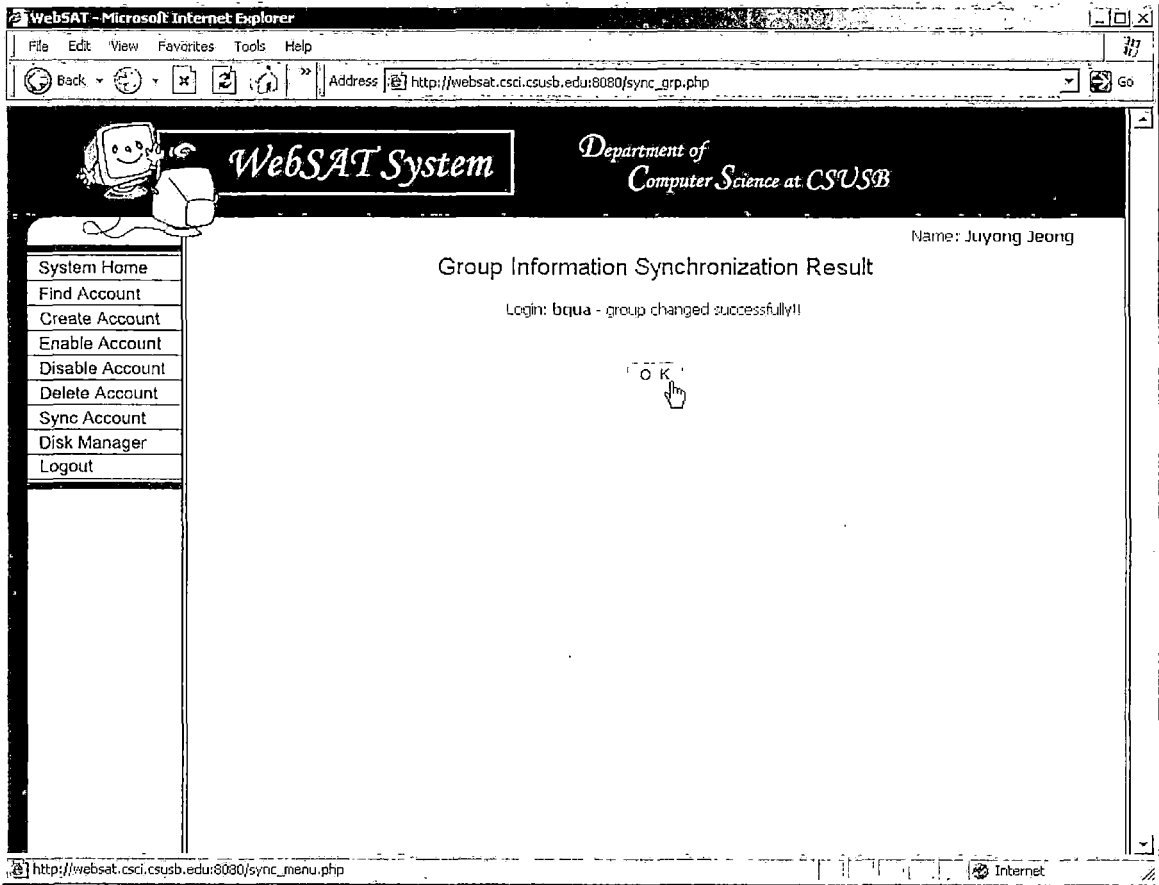
(6)

Figure 3.6. Demonstration of Synchronize Account
(Continued)



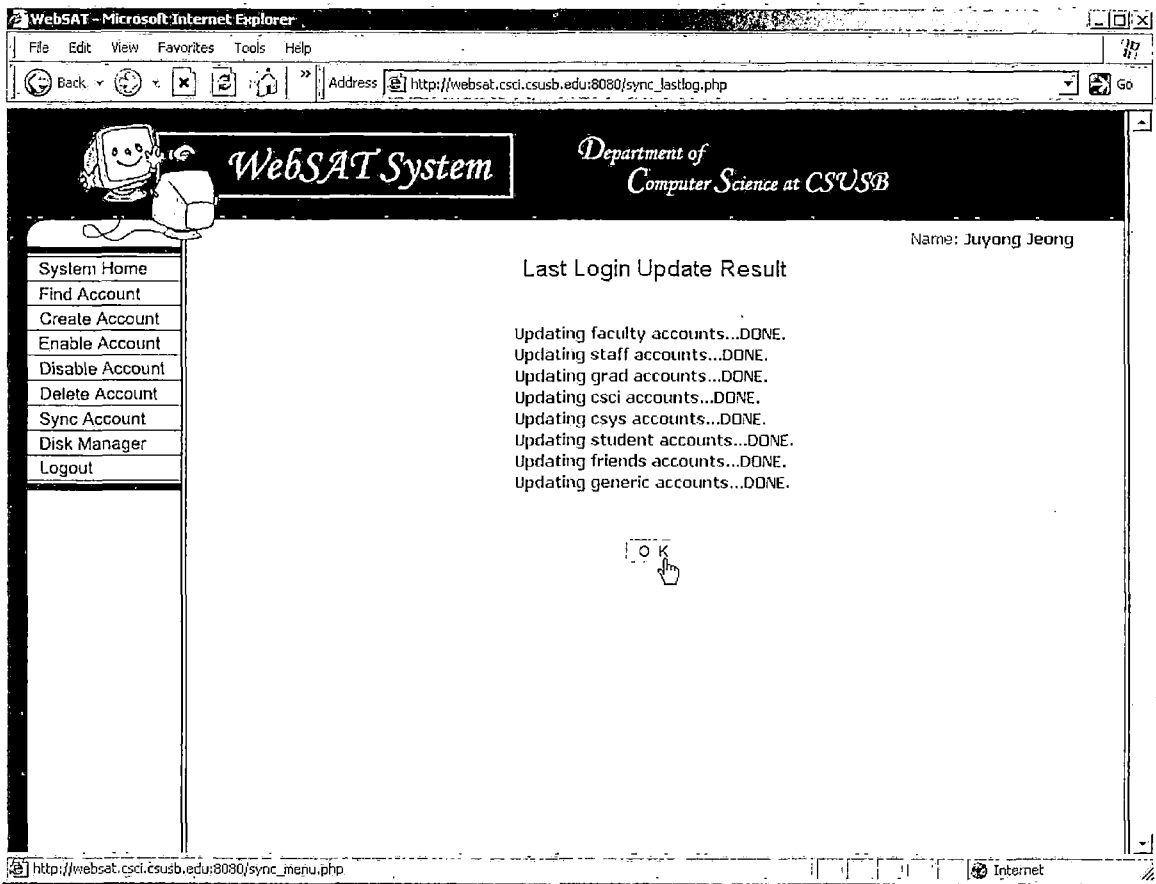
(7)

Figure 3.6. Demonstration of Synchronize Account
(Continued)



(8)

Figure 3.6. Demonstration of Synchronize Account
(Continued)



(9)

Figure 3.6. Demonstration of Synchronize Account
(Continued)

3.4.7 Disk Manager

Figure 3.7 demonstrates disk manager function. Figure 3.7 (1) shows the total disk capacity of the file system and how much disk spaces are used by each group. Also, WebSAT shows login information of users who exceeds his/her given disk quota limit. Figure 3.7 (1) shows that three users use disk spaces over the quota limits. Among them, one account is disabled already, and one account is selected to be disabled. Figure 3.7 (2) shows the result of disable account.

WebSAT - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address: https://websat.csci.csusb.edu:8443/disk_check.php

WebSAT System Department of Computer Science at CSUSB

Name: Juyong Jeong

Disk Space Check

Disk: boot
 Total space: 99M
 Used space: 8.8M
 Available space: 85M
 Percentage: 10%

faculty/:	342M
staff/:	256M
grad/:	129M
csci/:	115M
csys/:	34M
student/:	300M
friends/:	43M
generic/:	86M

Over Disk Quota Limit Account List

Over than: K

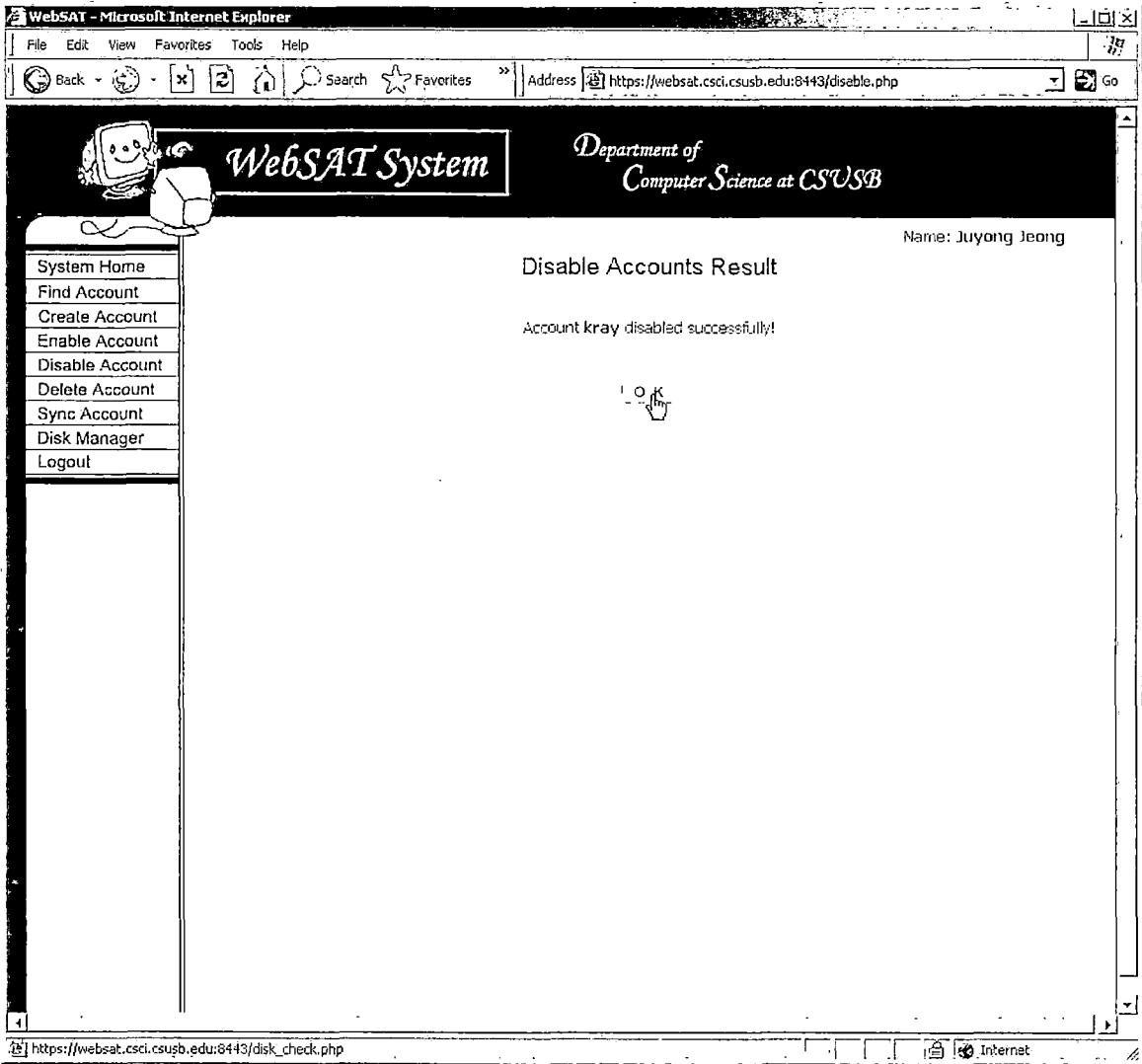
Check all

Disable	Login	Name	Group	Status	Quota	Space	Percent
<input checked="" type="checkbox"/>	kray	Kelly Ray	faculty	Active	100 M	342 M	342%
<input type="checkbox"/>	kwon	Han Kwon	staff	Disabled by Admin	100 M	256 M	256%
<input type="checkbox"/>	testest	test test	grad	Active	50 M	107 M	214%

Total: 2 account(s) found.

(1)

Figure 3.7. Demonstration of Disk Manager



(2)

Figure 3.7. Demonstration of Disk Manager (Continued)

CHAPTER FOUR

MAINTENANCE MANUAL

4.1 Files and Directories

Since this project is implemented through Apache Web Server, the document root directory of Apache is WebSAT root. There are diskuse, images, and inc directories under WebSAT root. All php scripts are in WebSAT root directories. Table 4.1 shows all related directories and describes what files in each directory.

Table 4.1. Files and Directories

Directory	Files
root	There are 39 PHP scripts and one expect script file.
diskuse	There are two data file: diskuseG and diskuseU
images	All graphics files which are used on WebSAT are in this directory.
inc	There are four environment files: Dbconnection.php, functions.php, setting.php, variables.php

4.2 Server Installation

This section shows how to install Mysql and Secure Web Server with PHP. Assume that NIS server and NFS server are installed on RedHat Linux 7.2. Figure 4.1 - 4.4 show the procedure.

First of all, the systems administrator needs to download all source files form the Internet. Figure 4.1 shows where to download files from, and how to decompress zip files.

```
1 source files to /usr/local directory.

• cd /usr/local
• ftp://ftp.openssl.org/source/openssl-0.9.7e.tar.gz
• http://www.modssl.org/source/mod_ssl-2.8.22-1.3.33.tar.gz
• http://httpd.apache.org/dist/httpd/apache_1.3.33.tar.gz
• http://www.linuxguruz.org/downloads/php-4.3.9.tar.gz
• http://prdownloads.sourceforge.net/mysql/mysql-4.1.7.tar.gz

3. Decompress source files

• # tar -zxvf openssl-0.9.7e.tar.gz
• # tar -zxvf mod_ssl-2.8.22-1.3.33.tar.gz
• # tar -zxvf apache_1.3.33.tar.gz
• # tar -zxvf php-4.3.9.tar.gz
• # tar -zxvf mysql-4.1.7.tar.gz

4. Remove tarballs

• # rm /usr/local/*.tar.gz
```

Figure 4.1. How to Get Ready to Install Programs

Figure 4.2 shows how to install MySQL. MySQL is installed through `configure`, `make`, and `make install` procedures. After installation, the systems administrator creates a database for MySQL and starts `mysqld`. Then, the systems administrator sets up the password for MySQL root.

```
How to Install MySQL

• # cd ../mysql-4.1.7
• # ./configure
• # make
• # make install
• # scripts/mysql_install_db
• # adduser mysql
• # chown -R mysql.users /usr/local/var
• # safe_mysqld --user=mysql &
• # <enter>
• # mysqladmin -u root password '<enter your root
  password>'
```

Disable mysql so that the user mysql never actually logs in by adding this line in the file `/etc/login.access`:

```
 -:mysql:ALL
```

Figure 4.2. How to Install MySQL

Now, the systems administrator installs Openssl and Apache with mod_ssl. Figure 4.3 shows the procedures.

```
How to Install Openssl and mod_ssl

Install Openssl

  • # cd ../openssl-0.9.7e
  • # sh config -fPIC
  • # make

Install Apache with mod_ssl

  • # cd ../mod_ssl-2.8.22-1.3.33
  • # ./configure --with-apache=../apache_1.3.33
  • # cd ../apache_1.3.33
  • # SSL_BASE=../openssl-0.9.7e ./configure \
    --prefix=/usr/local/apache --enable-module=most \
    --enable-shared=max --enable-module=ssl --enable-
    shared=ssl
  • # make
  • # make certificate
  • # make install
```

Figure 4.3. How to Install Openssl and Apache with Mod_ssl

Figure 4.4 show how to install PHP and start Apache with Secure Sockets Layer support. During this procedure, all flags are provided correctly.

Install PHP and Start Apache with SSL

- # cd ../php-4.3.9
- # ./configure --with-apxs=/usr/local/apache/bin/apxs \
--with-config-file-path=/usr/local/apache/conf \
--enable-versioning --with-mysql=/usr/local \
--enable-ftp --enable-bcmath --disable-debug \
--enable-memory-limit=yes --enable-track-vars
- # make
- # make install
- # cp /usr/local/php-4.3.9/php.ini-dist
/usr/local/apache/conf/php.ini
- Make sure you have, Uncomment (remove the #'s), Change
or Create the following lines in the file
/usr/local/apache/conf/httpd.conf

```
LoadModule php4_module    libexec/libphp4.so

AddModule mod_php4.c

<IfModule mod_dir.c>
DirectoryIndex index.html index.php index.php3 index.phtml
</IfModule>
# And for PHP 4.x, use:
#
AddType application/x-httpd-php .html .php .php3 .phtml
AddType application/x-httpd-php-source .phps
```

- # /usr/local/apache/bin/apachectl startssl

Figure 4.4. How to Install PHP and Start Apache

4.3 Software Installation

This section shows how to install WebSAT in the Linux operating system. Assume that NIS, NFS, Secure Web Server with PHP, and MySQL servers are running on the machine.

1. Get "WebSAT.tar.gz" file.
2. Copy the file to Apache document root directory.
3. Extract "WebSAT.tar.gz" by using -zxvf option "tar -zxvf WebSAT.tar.gz"
4. Delete the tarball.

4.4 Database Maintenance

4.4.1 Database Administration

Before WebSAT starts to run, WebSAT needs a MySQL account to create its database. The systems administrator will create a MySQL account for WebSAT. To create a user account on MySQL and to assign a database to WebSAT, the systems administrator takes the following steps:

1. #> mysql -u root -p mysql

This command will prompt root password.

2. mysql> insert into user (host, user, password)
values ('localhost', 'websat', password('websat-
password'));

This command creates a user websat.

3. mysql> create database websat;

This command create database websat.

4. mysql> insert into db (host, user, db, select_priv)
values ('localhost', 'websat', 'websat', 'Y');

This command gives select privilege on websat
database to user websat.

5. mysql> flush privileges;

This command flushes all privileges.

4.4.2 Database Installation

Before a user uses WebSAT, WebSAT needs its initial database. Figure 4.5 shows how to create initial database in MySQL.

```
mysql> create table status (  
-> no int NOT NULL,  
-> st varchar(40) NOT NULL  
-> );  
mysql> insert into status values (0, 'Disable');  
mysql> insert into status values (1, 'Active');  
mysql> insert into status values (2, 'Dis by Adm');
```

Figure 4.5. Initial Database Creation

```

mysql> create table account (
-> lname varchar(20) NOT NULL,
-> fname varchar(20) NOT NULL,
-> sid char(16) NOT NULL,
-> login varchar(10) NOT NULL,
-> grp int NOT NULL,
-> status int default 1,
-> c_date date NOT NULL,
-> l_date date,
-> d_date date,
-> primary key(login)
-> );

mysql> create table groups (
-> type int NOT NULL,
-> gid int NOT NULL,
-> name varchar(20) NOT NULL,
-> default_sid varchar(11),
-> major varchar(20),
-> location varchar(50) NOT NULL,
-> dsk_qt int,
-> primary key(gid)
-> );

mysql> insert into groups values (1, 200, 'faculty', '000-00-0001', NULL, '/u/faculty/', 50);

mysql> insert into groups values (1, 300, 'staff', '000-00-0002', NULL, '/u/staff/', 50);

mysql> insert into groups values (2, 400, 'grad', NULL, 'CSCM', '/u/grad/', 50);

mysql> insert into groups values (2, 500, 'csci', NULL, 'CSCI', '/u/csci/', 50);

mysql> insert into groups values (2, 550, 'csys', NULL, 'CSYS', '/u/csys/', 50);

mysql> insert into groups values (2, 600, 'student', NULL, NULL, '/u/student/', 50);

mysql> insert into groups values (1, 700, 'friends', '000-00-0003', NULL, '/u/friends/', 50);

```

Figure 4.5. Initial Database Creation (continued)

```
mysql> insert into groups values (3, 800, 'generic', '000-00-0004', NULL, '/u/generic/', 50);

mysql> insert into groups values (4, 0, 'system', '000-00-0000', NULL, '/u/system/', 50);

mysql> create table admin (
-> lname varchar(20) NOT NULL,
-> fname varchar(20) NOT NULL,
-> login char(16) NOT NULL,
-> pw char(16) NOT NULL,
-> level int,
-> primary key(login)
-> );
```

Figure 4.5. Initial Database Creation (continued)

CHAPTER FIVE

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

As the number of students in the Department of Computer Science at CSUSB increases, the network account system of the Department is getting larger and larger. Then, it becomes harder for the systems administrator to manually manage the computer science network system. These circumstances require the need for a new network account management system, and to meet this demand WebSAT was developed with the following functions:

- Finding account
- Creating account
- Enabling account
- Disabling account
- Delete account
- Synchronizing account
- Managing disk space
- Warning for security attacks on the network

These WebSAT functions benefit not only the systems administrator but also the Department. The following are the benefits of WebSAT:

- WebSAT reduces the systems administrator's one week job in managing students' accounts to just 30 minutes.
- WebSAT prevents the leak of systems resources.
- WebSAT gives high quality services to the users.
- WebSAT alerts the systems administrator to attacks from the outside world.

The best benefit of WebSAT is that it saves the systems administrator's time. Usually, the systems administrator of the Department creates over 300 accounts before a new quarter begins, and it takes about three minutes to create one account. That means it takes over 900 minutes (15 hours) for the systems administrator to create accounts. However, by using WebSAT create account function, the system administrator can create 300 accounts in 30 minutes. WebSAT helps the systems administrator find accounts which are no longer used. WebSAT provides the systems administrator with functions to delete or disable these accounts at one time. Systems resources are saved by

collecting the disk spaces which are used by these accounts. WebSAT also prevents access to the computer network from these accounts. WebSAT has PSAS (Port Scan Alert System), which alerts the systems administrator when an intruder scans ports on the NIS server and NFS server. WebSAT helps the systems administrator to take actions immediately against attacks from the outside world. These aspects of WebSAT benefit the systems administrator and the Department.

5.2 Future Work

Future work on WebSAT will consists of the following features:

- Automation of student group and generic group account disabling - After certain amount of time passed all student group and generic group accounts need to be disabled. This job can be done by WebSAT system itself, rather than by the user. It would save the user's valuable time.
- More detailed account status - It will help the user understand why he/she disabled accounts. It happens time to time that the user disabled one account, and never knew why he/she disabled that account. The user

can prevent being in this situation by writing a brief note of reasons.

- Internet Address Manager - It will have all IP address and MAC address information to identify each machine in the Department. The systems administrator will use this feature when he/she assigns a new IP address to a new machine. When there is an IP conflict, this feature will figure out which machine has an IP conflict.
- Workstation Manager - This component will check whether workstations and servers are reachable or not. Then, workstation manager will display the result on the screen in graphical mode. The user will know the status of all network machines in the Department.
- Print Log Manager - Print log manager will record all print logs and report who has printed how many pages to the systems administrator. This feature will be very helpful to save Department resources.
- Traffic Controller - Traffic controller will scan the traffic between workstations, make logs, and draw graphs to visualize the current traffic. Also, it

alerts the systems administrator if there is suddenly heavy traffic in the network.

- Service Manager - It will check if certain services and demons are working properly on remote systems, and display the result in one table.
- Password Reset Feature - This feature will reset account password. When a student forgets his/her account password, this function will be used.
- Automated Account Information Synchronization - This feature will check if account information need to be synchronized. If synchronization is needed, this feature will show that message when the user logins in. Then, if the user allows the system to synchronize information, the system synchronizes account information automatically.

GLOSSARY OF TERMS

Terminology	Definition
CLASS DIAGRAM	A diagram that shows a collection of declarative (static) model elements, such as classes, types, and their contents and relationships.
DEPLOYMENT DIAGRAM	A diagram that shows the configuration of run-time processing nodes and the components, processes, and objects that live on them. Components represent run-time manifestations of code units.
ENTITY-RELATIONSHIP DIAGRAM	An entity-relationship diagram is a data modeling technique that creates a graphical representation of the entities, and the relationships between entities, within an information system.
MySQL	MySQL is an open source relational database management system that uses Structured Query Language, the most popular language for adding, accessing, and processing data in a database.
NIS	NIS (Network Information System) is a network naming and administration system for smaller networks that was developed by Sun Microsystems. NIS+ is a later version that provides additional security and other facilities. Using NIS, each host client or server computer in the system has knowledge about the entire system.

Terminology	Definition
NFS	The Network File System (NFS) is a client/server application that lets a computer user view and optionally store and update file on a remote computer as though they were on the user's own computer. The user's system needs to have an NFS client and the other computer needs the NFS server.
PHP	PHP is a web-programming script language and interpreter that is freely available and used primarily on Linux Web Servers.
PORT SCAN	A port scan is a series of messages sent by someone attempting to break into a computer to learn which computer network services, each associated with a "well-known" port number, the computer provides. Port scanning, a favorite approach of computer cracker, gives the assailant an idea where to probe for weaknesses. Essentially, a port scan consists of sending a message to each port, one at a time. The kind of response received indicates whether the port is used and can therefore be probed for weakness.
PRIMARY KEY	The primary key of a relational table uniquely identifies each record in the table. It can either be a normal attribute that is guaranteed to be unique or it can be generated by the DBMS.

Terminology	Definition
PSEUDO-CODE	A high-level abstraction of code, usually used to outline the general steps in an algorithm without having to write actual code.
SESSION	In telecommunication, a session is a series of interactions between two communication end points that occur during the span of a single connection. Typically, one end point requests a connection with another specified end point and if that end point replies agreeing to the connection, the end points take turns exchanging commands and data.
SSL	The Secure Sockets Layer (SSL) is a commonly-used protocol for managing the security of a message transmission on the Internet. SSL has recently been succeeded by Transport Layer Security (TLS), which is based on SSL. SSL uses a program layer located between the Internet's Hypertext Transfer Protocol (HTTP) and Transport Control Protocol (TCP) layers. SSL is included as part of both the Microsoft and Netscape browsers and most Web server products.
TCP/IP	Short for Transmission Control Protocol/Internet Protocol, the suite of communications protocols used to connect hosts on the Internet. TCP/IP uses several protocols, the two main ones being TCP and IP. TCP/IP is built into the UNIX operating system and is used by the Internet, making it the de facto standard for transmitting data over networks.

REFERENCES

- [1] Luis Argerich, Wankyu Choi, John Coggershall, Ken Egervari, "Professional PHP 4", Wrox Press, United States, 2002.
- [2] Paul Dubois, "MySQL", New Riders, United States, 1999.
- [3] Steven Holzner, "Perl Black Book", United States, 1999.
- [4] Thorsten Kukuk, "The Linux NIS(YP)/NYS/NIS+ HOWTO", Linux.org, Feb 7, 2002
<<http://www.linux.org/docs/ldp/howto/NIS-HOWTO/index.html>>.
- [5] Nicolai Langfeldt, Tavis Barr, Seth Vidal, "Linux NFS-HOWTO", Linux.org, Feb 7, 2002
<<http://www.linux.org/docs/ldp/howto/NFS-HOWTO/index.html>>.
- [6] Ralf S. Engelschall, "mod_ssl. The Apache Interface to Open SSL", mod_ssl.org, Jan 8, 2002
<<http://www.modssl.org/docs/2.8/>>.
- [7] The OpenSSL project, "Miscellaneous OpenSSL Documents", openssl.org, Jan 8, 2002
<<http://www.openssl.org/docs/>>.
- [8] Software Engineering Standard Committee, "IEEE Recommended Practice for Software Requirements Specifications Std.", The Institute of Electrical and Electronics Engineers, United States, 1998.