

California State University, San Bernardino

CSUSB ScholarWorks

---

Theses Digitization Project

John M. Pfau Library

---

2006

## Appointment system

Ming-Tse Chen

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



Part of the [Databases and Information Systems Commons](#)

---

### Recommended Citation

Chen, Ming-Tse, "Appointment system" (2006). *Theses Digitization Project*. 3469.  
<https://scholarworks.lib.csusb.edu/etd-project/3469>

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](mailto:scholarworks@csusb.edu).

APPOINTMENT SYSTEM

---

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  
Ming-Tse Chen  
March 2006

APPOINTMENT SYSTEM


---

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

---

by  
Ming-Tse Chen  
March 2006

Approved by:

  
\_\_\_\_\_  
Dr. Keith Schubert, Chair, Computer Science

*March 13, 2006*  
\_\_\_\_\_  
Date

  
\_\_\_\_\_  
Dr. Richard Botting

  
\_\_\_\_\_  
Dr. David Turner

## ABSTRACT

Appointment system (AS) is an on-line system that allows students to make appointments through the internet. The system allows students to browse information about the faculty, including current course, available times. If the student forgets the details of the appointment, he or she can log in at any time to retrieve this information. Additionally, the system sends email to the student with the appointment information, including the date and time of the appointment.

In addition to student functions, the system also provides functions to faculty and administrators. Faculty can edit their current courses and available times or check their appointments of the site. Administrators can add new faculty or delete the faculty who is going to leave the campus.

In this system, we use the MySQL database server to maintain persistent data.

## ACKNOWLEDGMENTS

I would like to express my deep appreciation to my graduate advisor, Dr. Keith Schubert, for his important and valuable contributions to me and my project. Discussions with Dr. Schubert always give me significant help to resolve my problems and give me positive directions to do my research. Dr. Schubert offered me a lot of precious information and suggestions during all the period of doing my project. I also would like to thank my two project committee faculties, Dr. Turner and Dr. Botting, for their supportive help and important advice.

I would like to thank my parents, deeply from my heart, who provide me the chance to fulfill my every dream in life. Their love warmly supports me in every day and their encouragement gives me the power to accomplish every success in my life.

## TABLE OF CONTENTS

ABSTRACT .....	iii
ACKNOWLEDGMENTS .....	iv
LIST OF TABLES .....	viii
LIST OF FIGURES .....	ix
CHAPTER ONE: INTRODUCTION	
1.1 Purpose of this Project .....	1
1.2 Project Products .....	2
CHAPTER TWO: SYSTEM REQUIREMENTS .....	3
2.1 System Interfaces .....	4
2.2 Hardware Interfaces .....	5
2.3 Software Interfaces .....	5
2.4 Security .....	6
CHAPTER THREE: DATABASE DESIGN	
3.1 Data Analysis .....	7
3.2 Database Schema Conceptual Model - Entity Relational Schema .....	7
3.3 Database Schema Logical Model - Relational Schema .....	8
3.4 Data Type and Detail .....	9
3.5 Page Flow Models .....	12
3.5.1 Student .....	13
3.5.2 Faculty .....	21
3.5.3 Administrator .....	29
CHAPTER FOUR: PROJECT IMPLEMENTATION	
4.1 User Interface Design .....	40

4.2 Graphical User Interface and Description . . . .	40
4.2.1 Login Page . . . . .	40
4.2.2 Student Home Page . . . . .	41
4.2.3 Student Make Appointment Page . . . . .	42
4.2.4 Student Cancel Appointment Page . . . . .	43
4.2.5 Student Edit Personal Information Page . . . . .	44
4.2.6 Change Password Page . . . . .	45
4.2.7 Faculty Home Page . . . . .	46
4.2.8 Faculty Check/Cancel Appointment Page . . . . .	47
4.2.9 Faculty Edit Personal Information Page . . . . .	48
4.2.10 Faculty Edit Current Course Page . . . .	49
4.2.11 Faculty Edit Available Time Page . . . .	50
4.2.12 Administrator Home Page . . . . .	51
4.2.13 Administrator Add New Faculty Page . . . . .	52
4.2.14 Administrator Delete Faculty Page . . .	53
4.2.15 Edit Faculty Current Course Page . . . .	54
4.2.16 Edit Personal Information Page . . . . .	55
4.2.17 Add New Administrator Page . . . . .	56
4.2.18 Delete Administrator Page . . . . .	57
4.2.19 Request Password Page . . . . .	58
4.2.20 Create New User Page . . . . .	59

CHAPTER FIVE: SYSTEM VALIDATION

5.1 Unit Test . . . . .	61
-------------------------	----

5.2 Subsystem Testing .....	64
5.3 System Test Plan .....	65
CHAPTER SIX: MAINTENANCE MANUAL	
6.1 Software Installation .....	67
6.1.1 Gentoo Installation .....	67
6.1.2 Install Java 2 Platform, Standard Edition .....	68
6.1.3 Install Tomcat .....	68
6.1.4 MySQL Installation .....	69
6.2 System Variables .....	69
6.3 Appointment System Installation/Migration ...	69
6.4 Backup and Restore .....	70
6.4.1 System Backup .....	70
6.4.2 Database Backup .....	71
6.4.3 System Restore .....	71
6.4.4 Database Restore .....	72
CHAPTER SEVEN: CONCLUSION AND FUTURE DIRECTIONS	
7.1 Conclusion .....	73
7.2 Future Directions .....	73
REFERENCES .....	74



## LIST OF TABLES

Table 1. Structure of Table Studentinfo .....	10
Table 2. Structure of Table facultyinfo .....	10
Table 3. Structure of Table administratorinfo .....	11
Table 4. Structure of Table checklogin .....	11
Table 5. Structure of Table current_course .....	11
Table 6. Structure of Table availabletime .....	12
Table 7. Structure of Table app_taken .....	12
Table 8. The Unit Test Results .....	62
Table 9. Subsystem Test Results .....	65
Table 10. System Test Results .....	66

## LIST OF FIGURES

Figure 1.	System Deployment Diagram(UML 1.0) .....	4
Figure 2.	Appointment System Entity Relationship Diagram .....	8
Figure 3.	Appointment System Database Relational Schema .....	9
Figure 4.	Login Page(Student) .....	13
Figure 5.	Student Home Page .....	14
Figure 6.	Request Password Page .....	15
Figure 7.	Create New User Page .....	16
Figure 8.	Make Appointment Page .....	17
Figure 9.	Check/Cancel Appointment Page(Student) ....	18
Figure 10.	Edit Personal Information Page(Student) ...	19
Figure 11.	Change Password Page(Student) .....	20
Figure 12.	Result Page(Student) .....	21
Figure 13.	Login Page(Faculty) .....	22
Figure 14.	Faculty Page .....	23
Figure 15.	Check/Cancel Appointment Page(Faculty) ....	24
Figure 16.	Edit Personal Information Page(Faculty) ...	25
Figure 17.	Edit Current Course Page(Faculty) .....	26
Figure 18.	Edit Available Time Page .....	27
Figure 19.	Change Password Page(Faculty) .....	28
Figure 20.	Result Page(Faculty) .....	29
Figure 21.	Login Page(Administrator) .....	30
Figure 22.	Administrator Home Page .....	31
Figure 23.	Add New Faculty Page .....	32

Figure 24. Delete Faculty Page .....	33
Figure 25. Edit Current Course Page .....	34
Figure 26. Add New Administrator Page .....	35
Figure 27. Delete Administrator Page .....	36
Figure 28. Edit Personal Information Page (Administrator) .....	37
Figure 29. Change Password Page (Administrator) .....	38
Figure 30. Result Page (Administrator) .....	39
Figure 31. Login Page .....	41
Figure 32. Student Home Page .....	42
Figure 33. Student Make Appointment Page .....	43
Figure 34. Student Cancel Appointment Page .....	44
Figure 35. Student Edit Personal Information Page ....	45
Figure 36. Change Password Page .....	46
Figure 37. Faculty Home Page .....	47
Figure 38. Faculty Check/Cancel Appointment Page .....	48
Figure 39. Faculty Edit Person Information Page .....	49
Figure 40. Faculty Edit Current Course Page .....	50
Figure 41. Faculty Edit Available Time Page .....	51
Figure 42. Administrator Home Page .....	52
Figure 43. Administrator Add New Faculty Page .....	53
Figure 44. Administrator Delete Faculty Page .....	54
Figure 45. Administrator Edit Current Course Page ....	55
Figure 46. Administrator Edit Personal Information Page .....	56

Figure 47. Administrator Add New Administrator Page .....	57
Figure 48. Administrator Delete Administrator Page ...	58
Figure 49. Request Password Page .....	59
Figure 50. Create New User Page .....	60

## CHAPTER ONE

### INTRODUCTION

Appointment System (AS) is an on-line system that allows students to make appointments through the internet. In AS, information about faculty's available time is presented. Students can browse through all the information. Since the information varies with faculty, all possible information will be presented in an orderly fashion to account for all the different possibilities. Students can log on and check the information from time to time. After students make appointments, this system emails the appointment details to the students and faculty automatically.

#### 1.1 Purpose of this Project

Students who want to discuss anything with faculty usually have to make an appointment first. When we want to make an appointment, we always send an email to the faculty and then wait for the reply. It is always need to write more than one email for students and faculty to have an appointment. If there is an appointment system to let students make appointment, then it will save both faculty and student time. The Appointment System offers all the CSCI students to make appointments of any CSCI faculty.

## 1.2 Project Products

The AS project led to the following products:

- Implementation of AS: a working web-based online application system with JSP and MySQL database, which achieves the specific needs of AS.
- System documentation: a project documentation, which is available with system design, specifications, project implementation and testing reports.

## CHAPTER TWO

### SYSTEM REQUIREMENTS

The Appointment System (AS) project aims to be friendly and convenient for students. The components of AS are a web server, a database server, and a client browser. Student browsers use the Internet (TCP/IP) to make appointments through the web server under HTTP/HTTPS. The web server connects to the email server with TCP/IP in order to send or retrieve email through SMTP. The web application connects to the database via TCP, and accesses database functionality through JDBC.

In order to choose implementation components conforming to the criteria of the shareware standard, this project uses Tomcat as web server, and MySQL as database server. The other components, such as the web browsers, are dependant on which kind of browsers the students use. The email server was provided by the ISP (Internet Service Provider).

The architecture of this project is shown below in Figure 1.

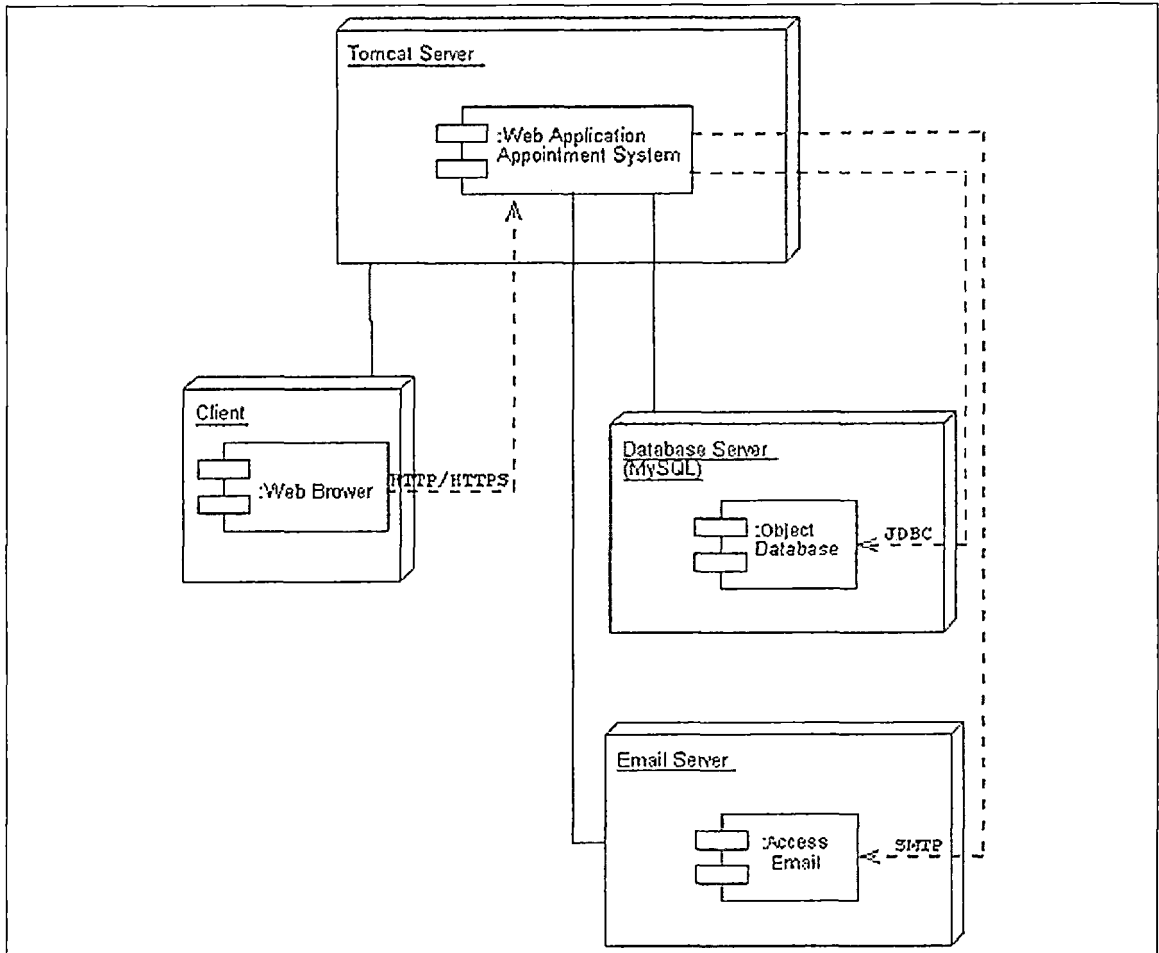


Figure 1. System Deployment Diagram (UML 1.0)

## 2.1 System Interfaces

The Appointment System (AS) is a 3-tier distributed architecture that displays the user interface in a web browser using HTML. The middle tier is a Java Servlet (automatically created from JSP pages) that handles requests from the client browser and provides access to the third tier via JDBC, which is a MySQL database. The HTTP server is provided by Apache Tomcat, which also implements JSP.



## 2.2 Hardware Interfaces

Appointment System will not implement hardware interface directly. However, it will trust the underlying operating system (Windows, NT, ME, XP, Linux, UNIX, MAC) to handle the hardware interfaces.

## 2.3 Software Interfaces

As explained above, there will be two different software interfaces depending on the type of access that the user is demanding or the function that the user wants the software to perform. The reasons that why I choose the software interfaces used in this project are summarized as following:

- Operating system (Gentoo Linux): a special flavor of Linux that can be automatically optimized and customized for just about any application or needed.
- Web Server/Container (Jakarta Tomcat Server): Tomcat server is a Java based web application container that was created to run servlets and Java Server Pages (JSP) in web applications.
- JAVA 2 Platform, Standard Edition (J2SE): A Java-based, runtime platform that provides many features for developing web-based Java

applications, including database access (JDBC API) interface technology, and security for both local network and Internet use and it's required in the Tomcat JAVA Container.

- Database Server (MySQL Server): MySQL is open source database software, and it's popular. MySQL also provides a JDBC driver to be easily connected from a JAVA program.
- Java Database Connector (JDBC): MySQL connector.
- Email package: JavaMail 1.3.
- Languages: HTML/JAVA/JSP/XML.

## 2.4 Security

In order to make sure the validity and the safety, I use the SSL (Secure Sockets Layer) between the web browser and the web server, and use a Web Service password between the web applications and Web Service. SSL is used to make sure that the messages transmitted between the users and the web applications will not be changed or be stolen. SSL will maintain the security and the completeness. The "s" in "https" stands security for http over SSL. The Web Service password is passed into the Web Service system with a configuration file (web.xml). Each time the web applications access the Web Service, it supplies the password.

## CHAPTER THREE

### DATABASE DESIGN

The database server used by Appointment System is MySQL. Once the database has been created using MySQL, all interfacing with the database will be done from Java. One must note that all interfaces as seen by the users of the system will be through pages with HTML forms generated from JSP.

#### 3.1 Data Analysis

The data for designing and implementing the schema of the database depends on seven entities: studentinfo table, facultyinfo table, administratorinfo table, checklogin table, current\_course table, availabletime table and app\_taken table. The tables of facultyinfo, current\_course, availabletime and app\_taken are connected by the relation of faculty's last name. The tables of studentinfo and app\_taken are connected by the relation of students' first name.

#### 3.2 Database Schema Conceptual Model - Entity Relational Schema

All the entities and relations used in Appointment System are described in the ER Diagram in Figure 2.

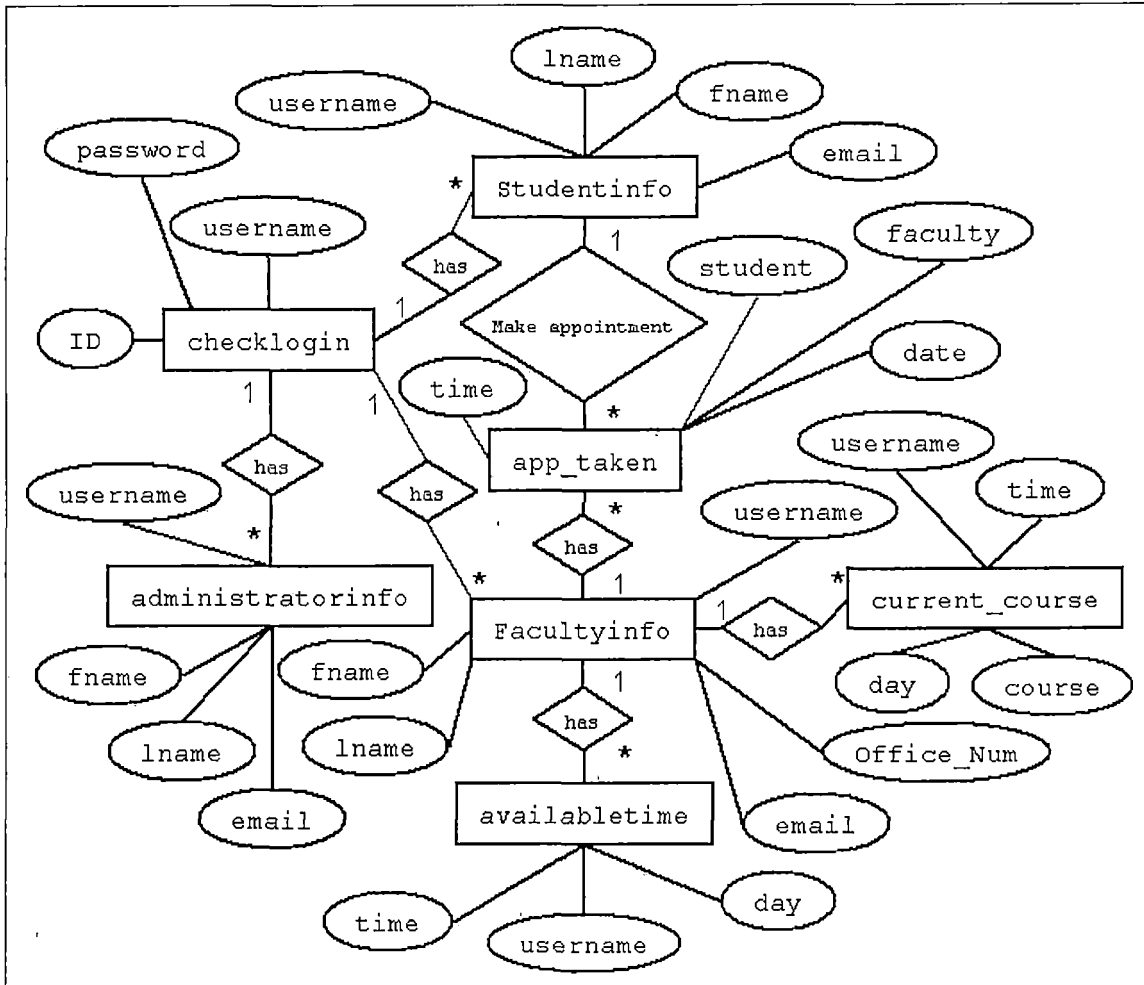


Figure 2. Appointment System Entity Relationship Diagram

### 3.3 Database Schema Logical Model - Relational Schema

The conceptual model ER diagram maps into the following relational table design. In the following tables (underlined fields indicate the primary key).

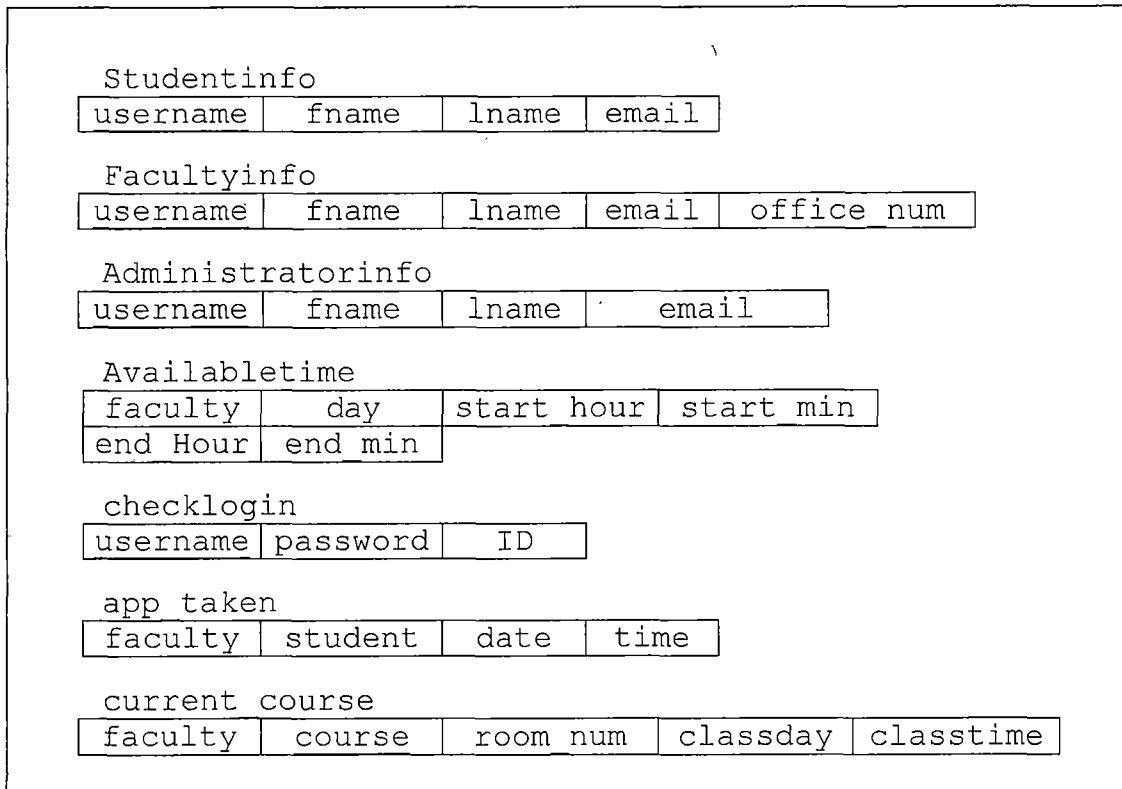


Figure 3. Appointment System Database Relational Schema

### 3.4 Data Type and Detail

The logical model establishes the following detailed design in MySQL database. The following tables describe data type, length, primary key, null or non-null keys, and extra information, such as `auto_increment`.

Table 1. Structure of Table Studentinfo

Field	type	null	key	default	extra
username	varchar(50)		PRI		
fname	varchar(20)				
lname	varchar(20)				
email	varchar(50)				

Table 2. Structure of Table facultyinfo

Field	type	null	key	default	extra
username	varchar(50)		PRI		
fname	varchar(20)				
lname	varchar(20)				
email	varchar(50)				
Office num	varchar(10)				

Table 3. Structure of Table administratorinfo

Field	type	null	key	default	extra
username	varchar(50)		PRI		
fname	varchar(20)				
lname	varchar(20)				
email	varchar(50)				
Office num	varchar(10)				

Table 4. Structure of Table checklogin

field	Type	null	key	default	extra
username	varchar(50)		PRI		
password	varchar(50)				
ID	varchar(2)				

Table 5. Structure of Table current\_course

Field	type	null	key	default	extra
faculty	varchar(20)		FOR		
course	varchar(20)		PRI		
room num	varchar(10)				
clasday	varchar(10)				
Classtime	varchar(30)				

Table 6. Structure of Table availabletime

Field	type	null	key	default	extra
faculty	varchar(20)		PRI		
day	varchar(2)				
start hour	varchar(5)				
start min	varchar(5)				
end hour	varchar(5)				
end min	varchar(5)				

Table 7. Structure of Table app\_taken

Field	type	null	key	default	extra
Faculty	varchar(20)		PRI		
student	varchar(20)		PRI		
date	varchar(20)				
time	varchar(10)				

### 3.5 Page Flow Models

Page flow model shows the process of the pages, by reading these diagrams, you can easily understand where the page goes, and where it from.

The square in the upper section of the diagram stands for the link or button to other page. The lower section stands for the content, the form field of the page.



### 3.5.1 Student

Figure 4 represents the Login Page which is used by the student, if the student successfully logs in the system, the student will be redirected to the Student Home Page; if not, the student will stay at the Login Page. If the student forgets his/her password, the student can go to the Password Page to request the password. New students can go to the New User page to create a new account.

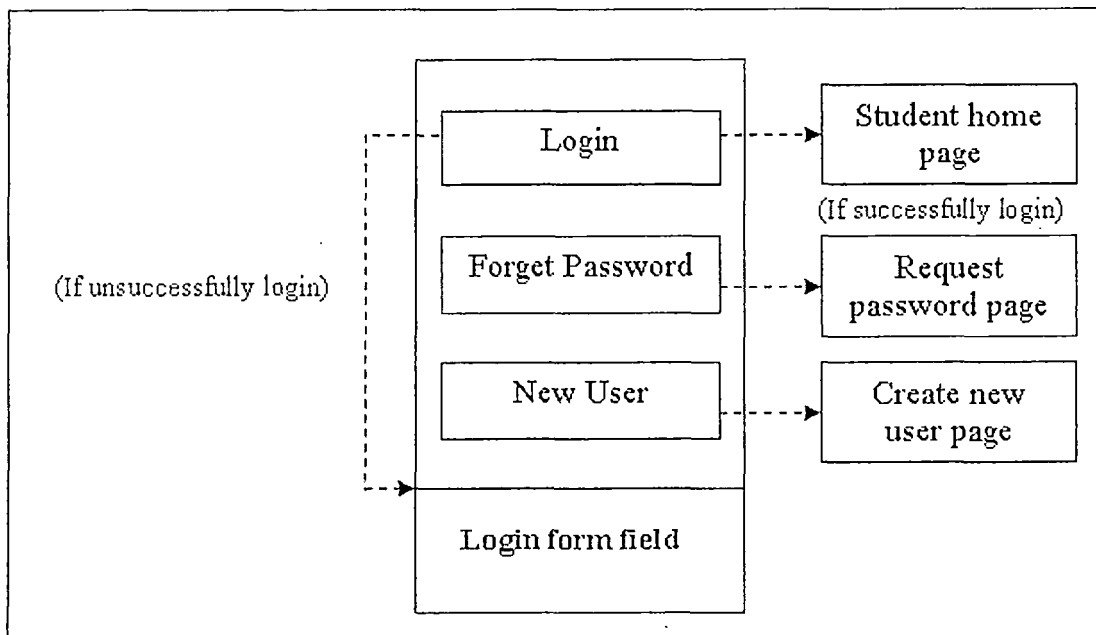


Figure 4. Login Page (Student)

Figure 5 represents the Student Home Page which is used by students. Students can choose which function they are going to use, or logout to the Login Page.

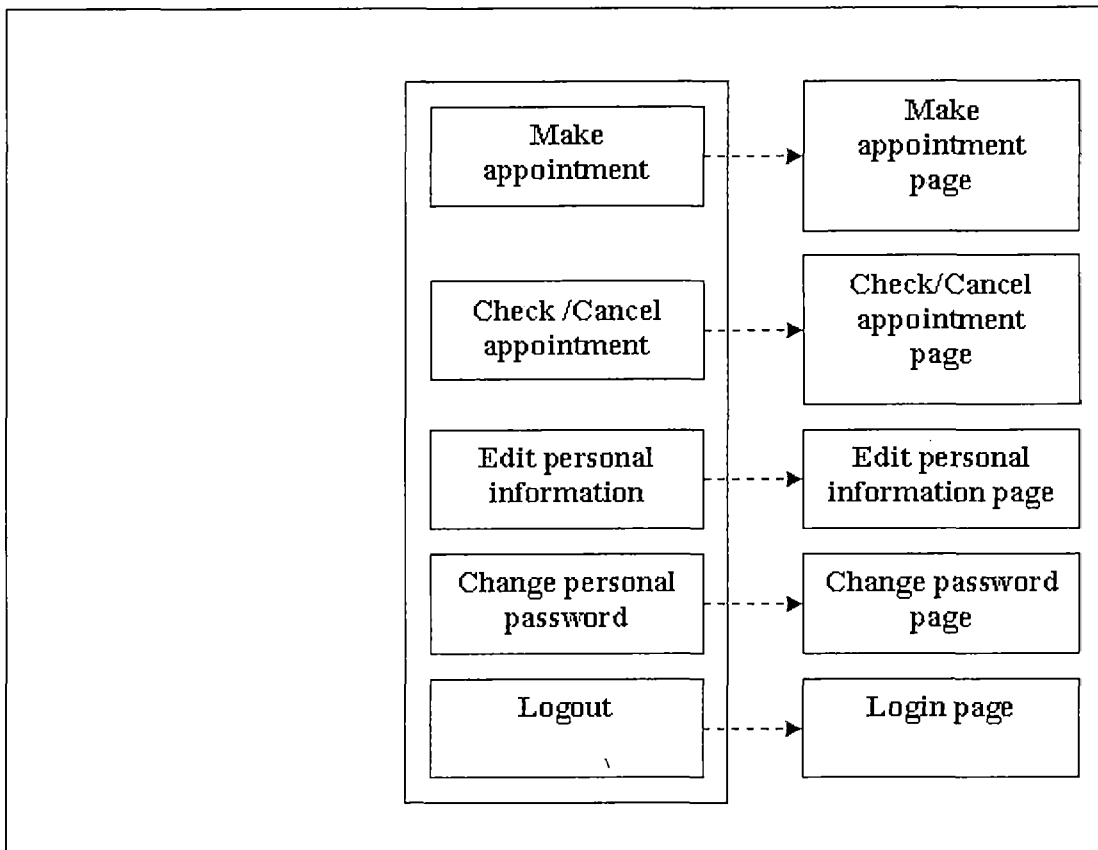


Figure 5. Student Home Page

Figure 6 represents the Request Password Page which is used by students. Students can request their password through this page, or go to other pages through other links.

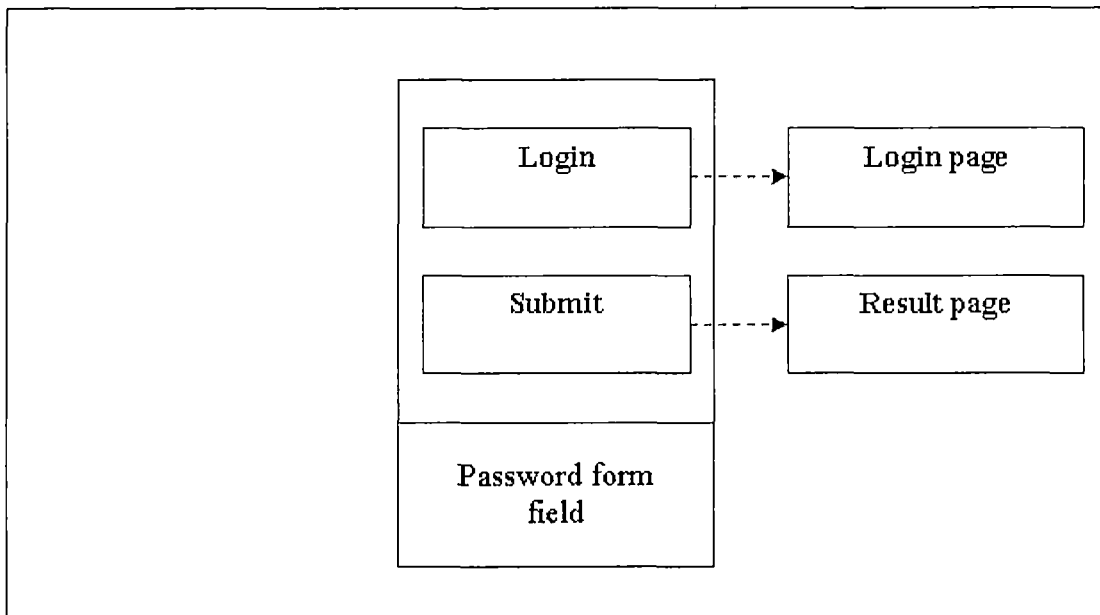


Figure 6. Request Password Page

Figure 7 represents the Create New User Page which is used by new students. New students can create new accounts through this page, or go to login page by clicking the login link.

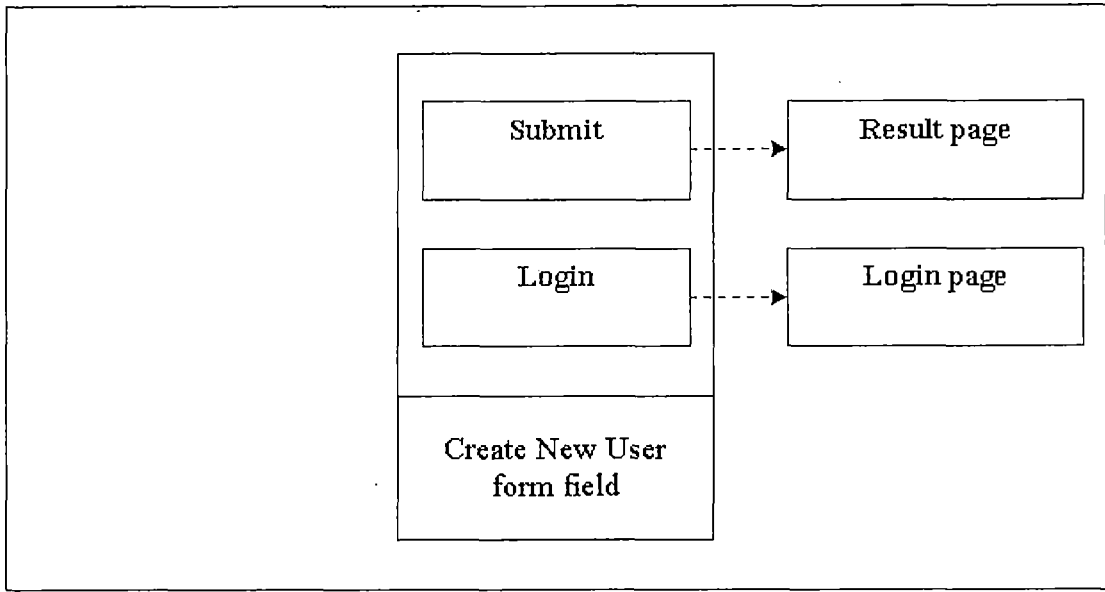


Figure 7. Create New User Page

Figure 8 represents the Make Appointment Page which is used by students. Students can make appointments through this page, or go to other pages through other links.

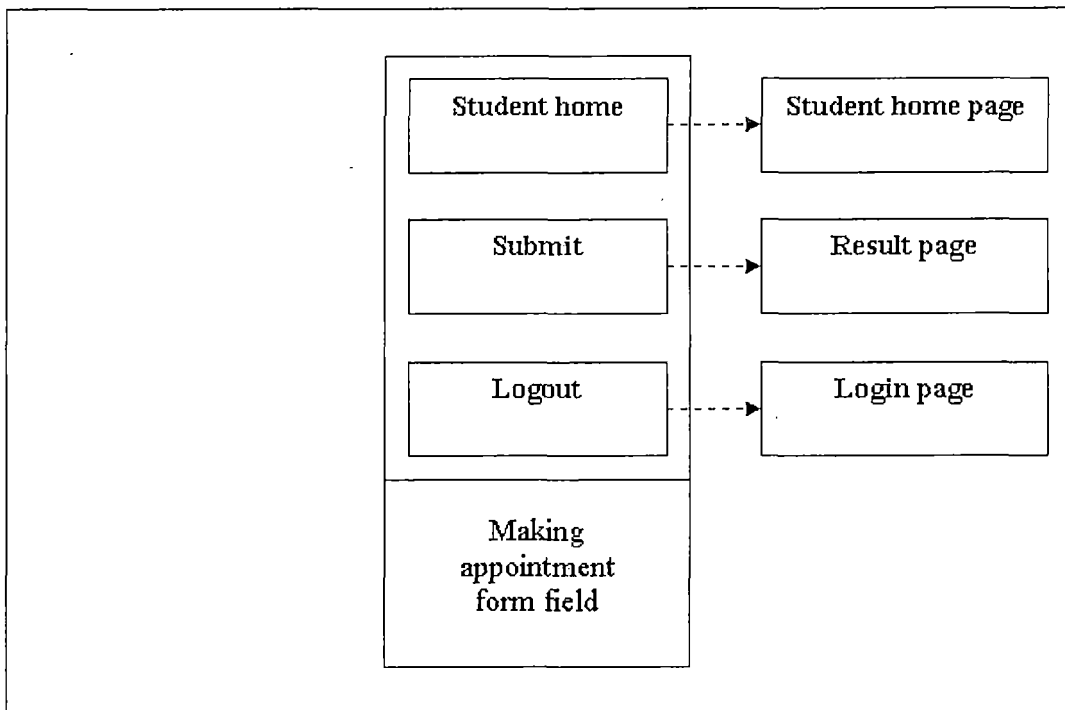


Figure 8. Make Appointment Page

Figure 9 represents the Check/Cancel Appointment Page which is used by students. Students can check and cancel their appointments through this page, or go to other pages through other links.

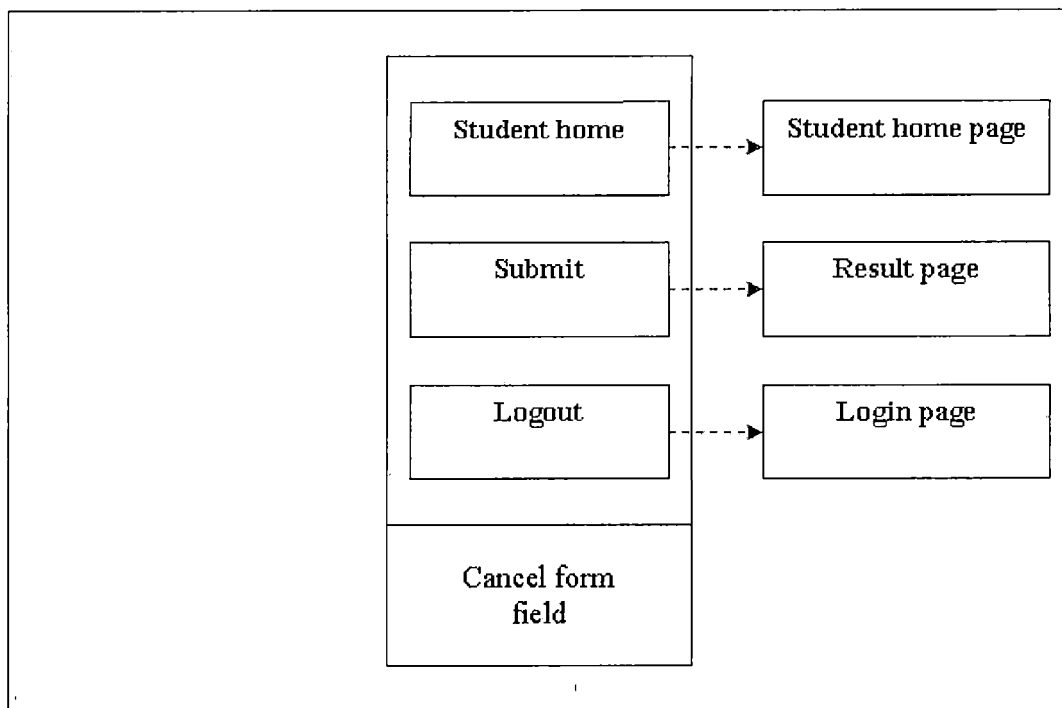


Figure 9. Check/Cancel Appointment Page (Student)

Figure 10 represents the Edit Personal Information Page which is used by students. Students can edit their personal information through this page, or go to other pages through other links.

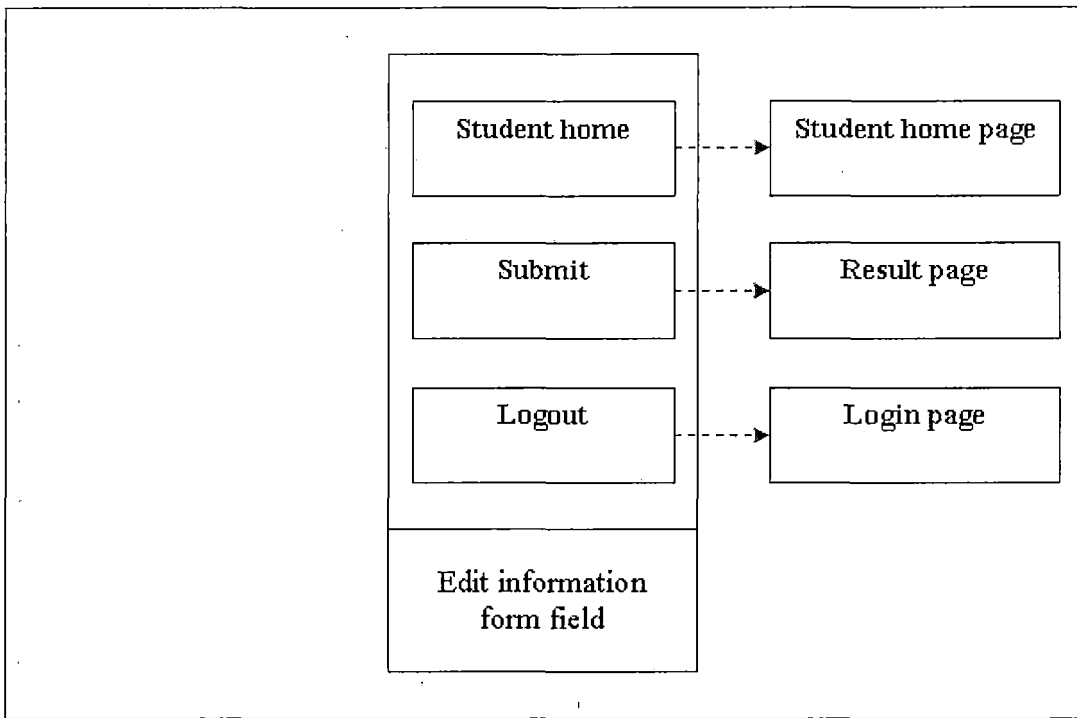


Figure 10. Edit Personal Information Page (Student)

Figure 11 represents the Change Password Page which is used by Students. Students can change their password through this page or go to other pages through other links.

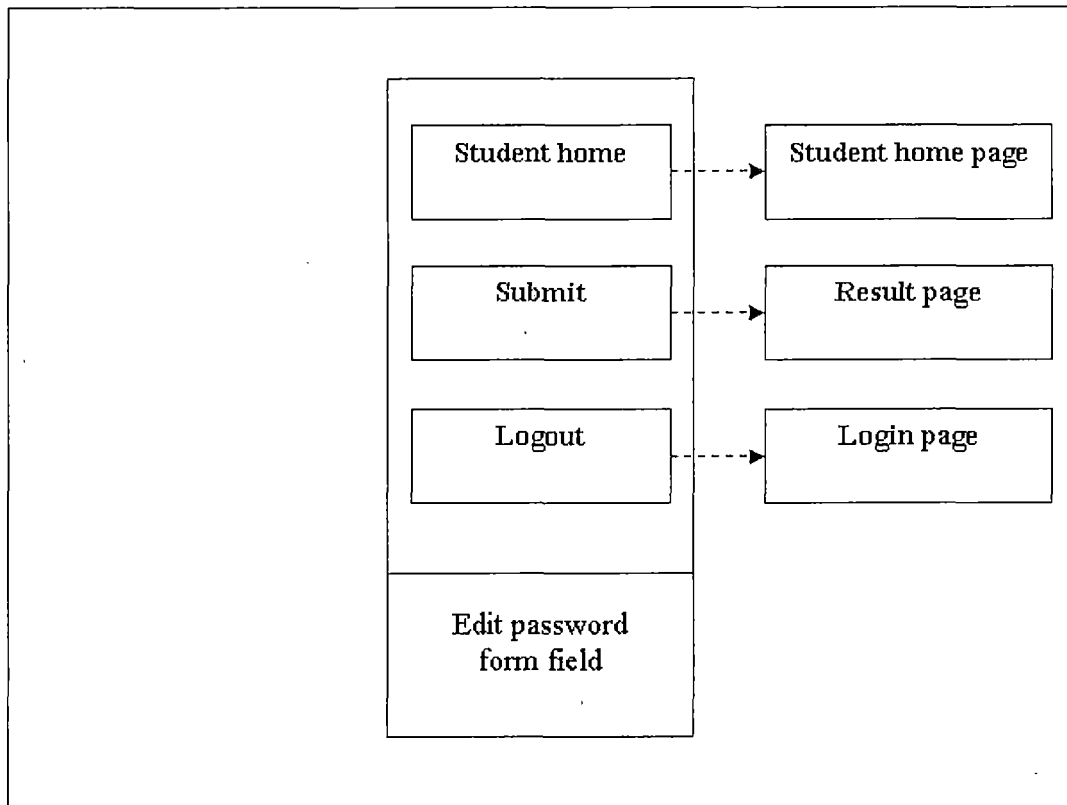


Figure 11. Change Password Page (Student)

Figure 12 represents the Result Page which is used by students. Students can see the result through this page and go to other pages through other links.



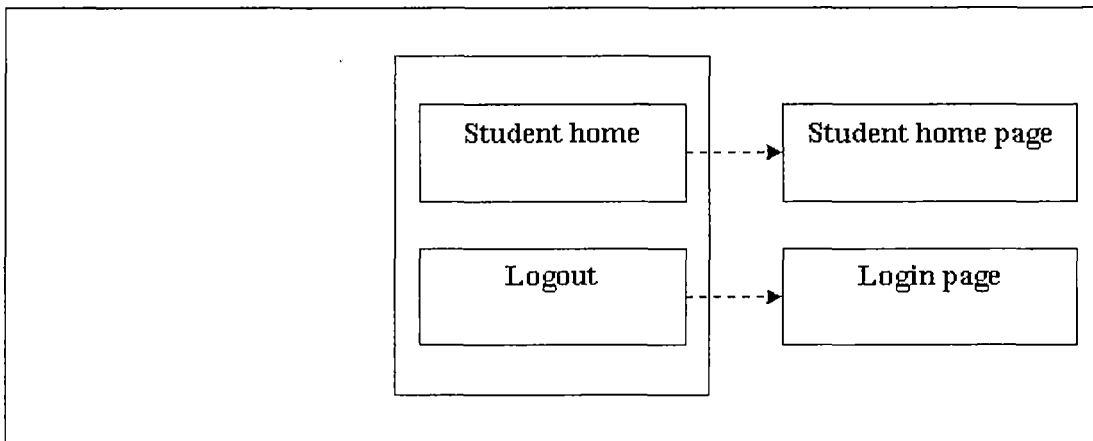


Figure 12. Result Page (Student)

### 3.5.2 Faculty

Figure 13 represents the Login Page which is used by the faculty, if the faculty successfully logs in the system, the faculty will be redirected to the Faculty Home Page; if not, the faculty will stay at the Login Page. If the faculty forgets his/her password, the faculty can go to the Request Password Page to request the password.

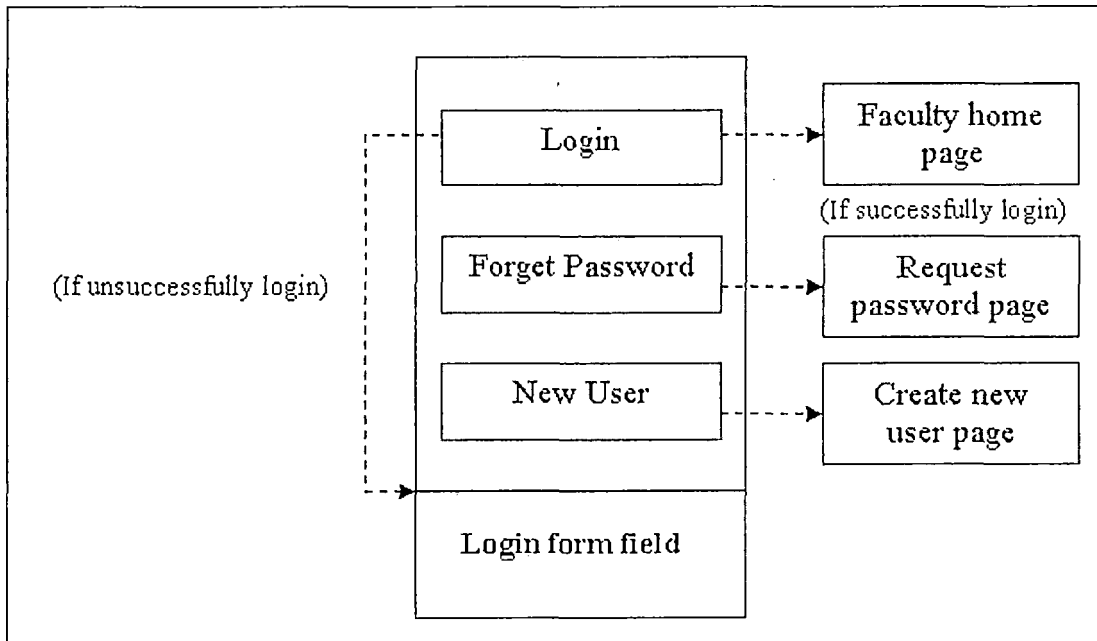


Figure 13. Login Page (Faculty)

Figure 14 represents the Faculty Home Page which is used by faculty. Faculty can choose which function they are going to use, or logout to the Login Page.

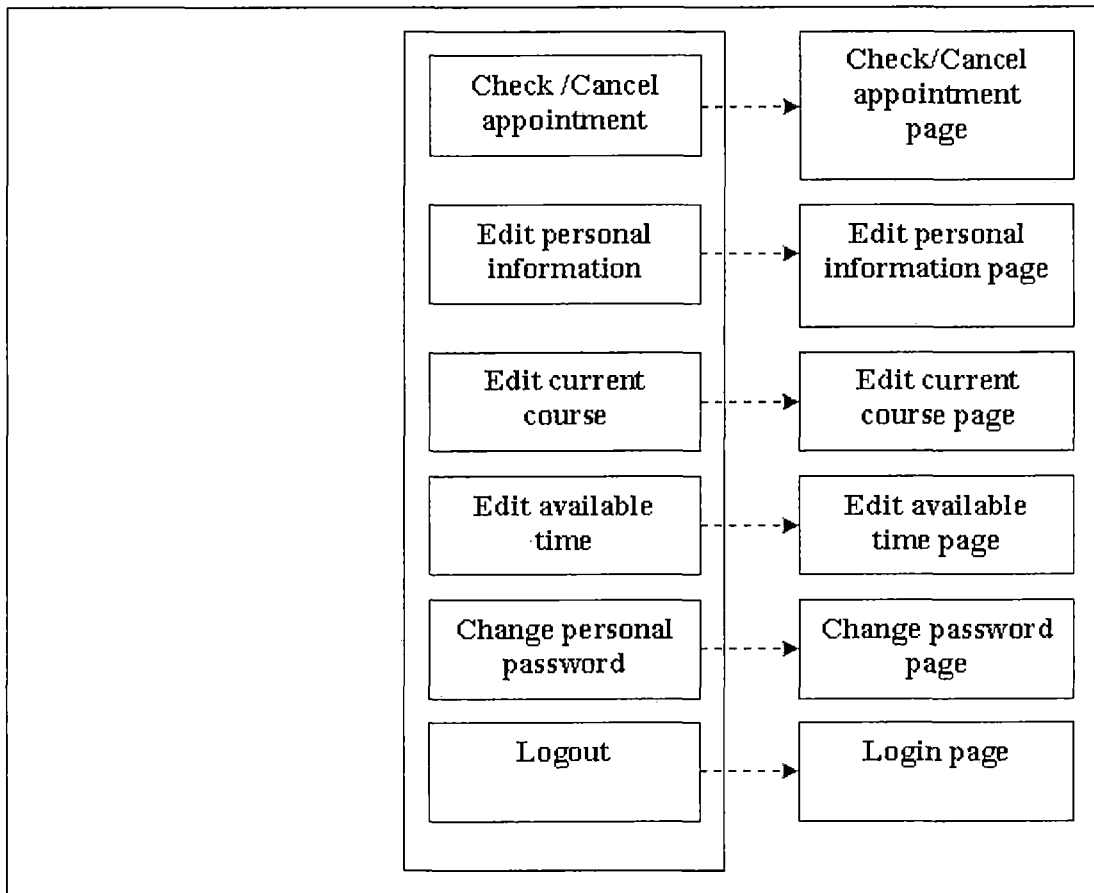


Figure 14. Faculty Page

Figure 15 represents the Check/Cancel Appointment Page which is used by faculty. Faculty can check and cancel appointment through this page, or go to other pages through other links.

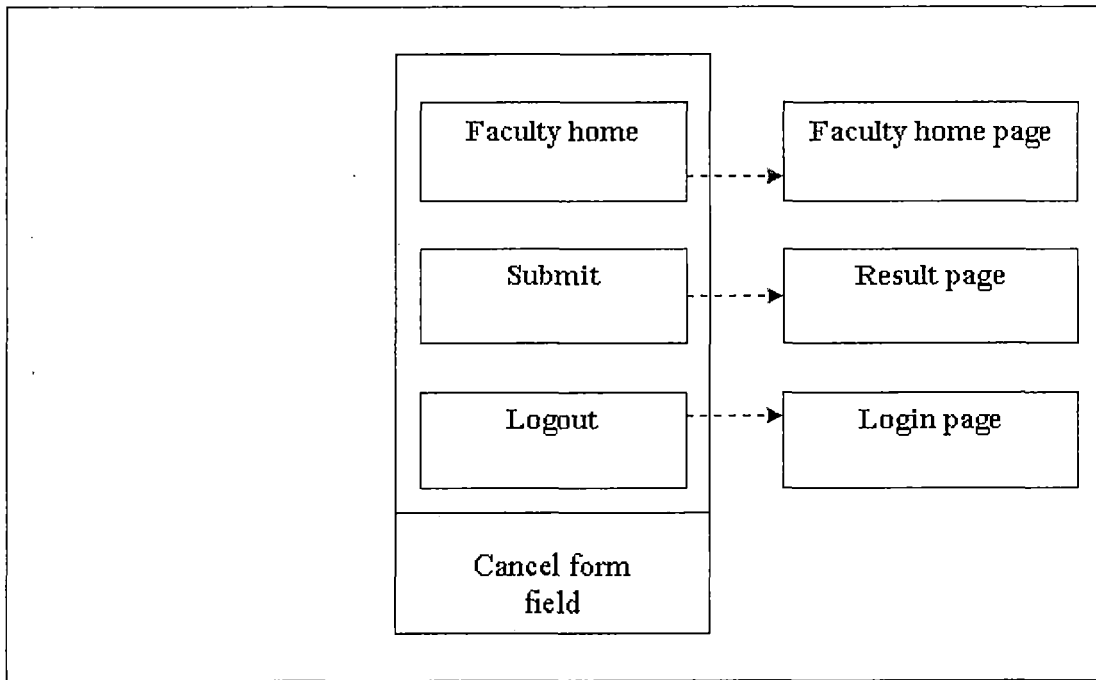


Figure 15. Check/Cancel Appointment Page (Faculty)

Figure 16 represents the Edit Personal Information Page which is used by faculty. Faculty can edit personal information through this page, or go to other pages through other links.

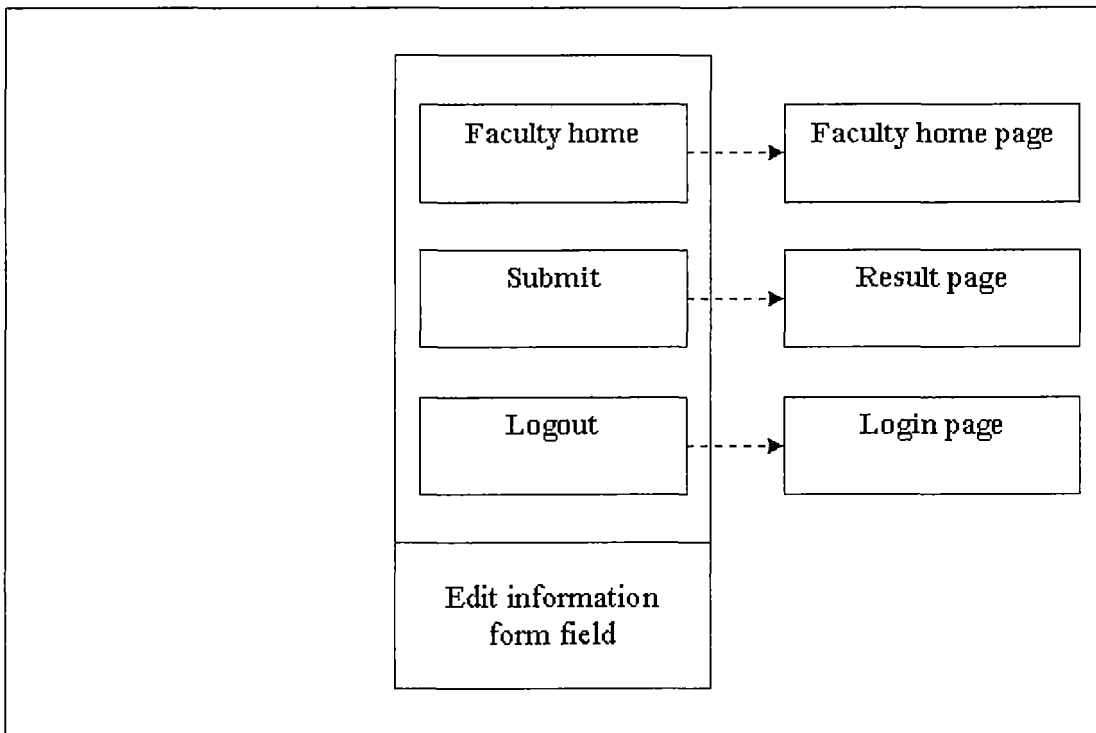


Figure 16. Edit Personal Information Page (Faculty)

Figure 17 represents the Edit Current Course Page which is used by faculty. Faculty can edit their current course through this page, or go to other pages through other links.

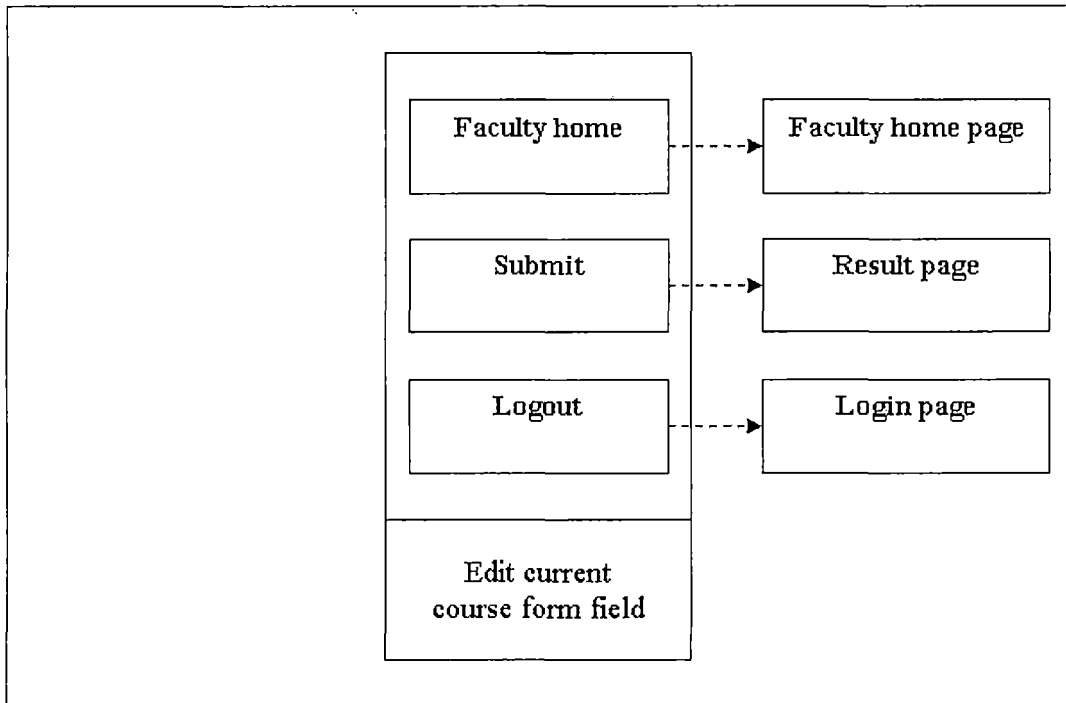


Figure 17. Edit Current Course Page (Faculty)

Figure 18 represents the Edit Available Time Page which is used by Faculty. Faculty can edit their available time through this page, or go to other pages through other links.

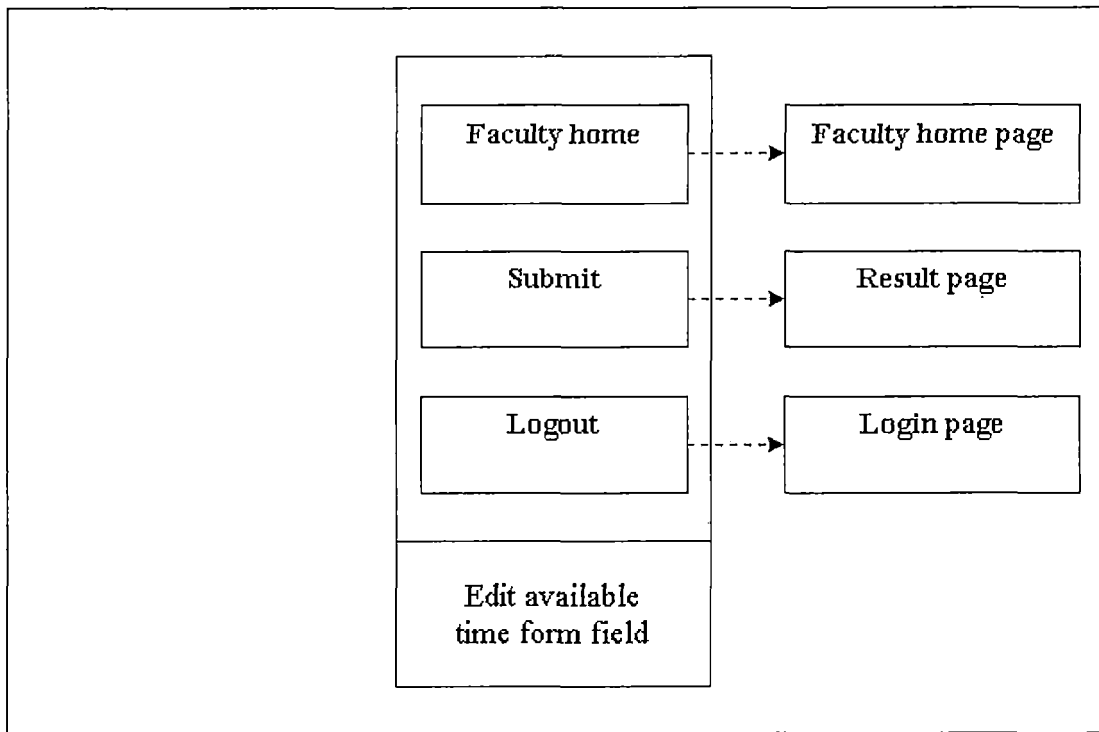


Figure 18. Edit Available Time Page

Figure 19 represents the Change Password Page which is used by faculty. Faculty can change their password through this page or go to other pages through other links.

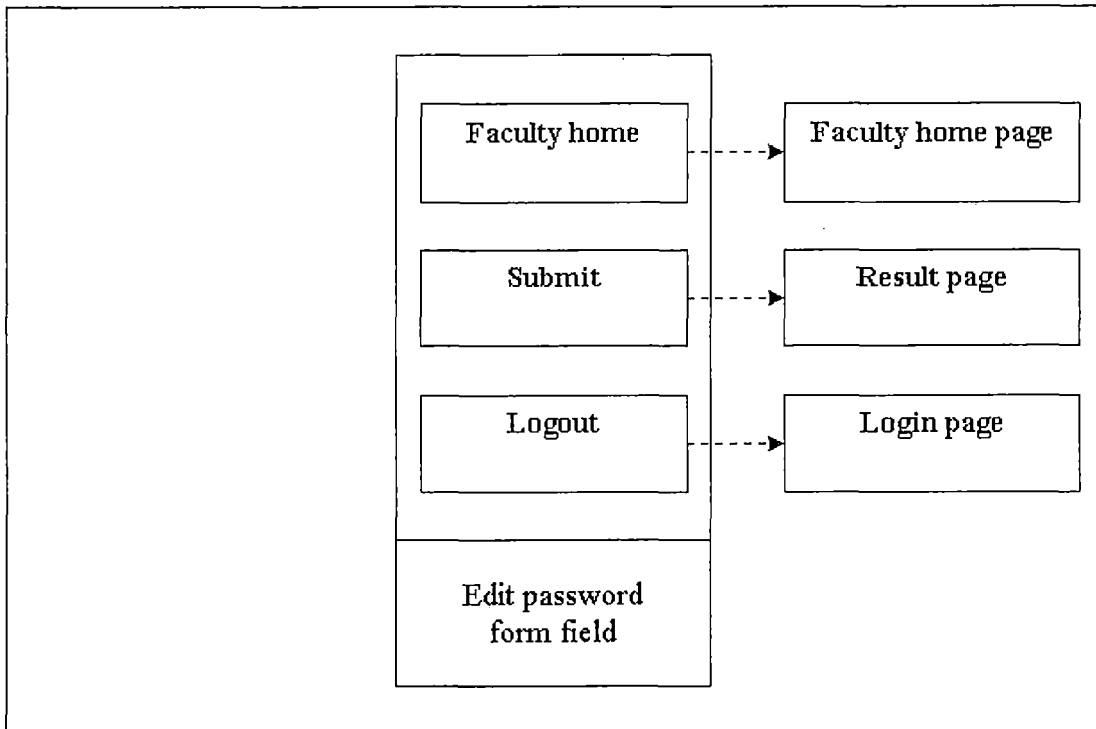


Figure 19. Change Password Page (Faculty)

Figure 20 represents the Result Page which is used by faculty. Faculty can see the result through this page and go to other pages through other links.



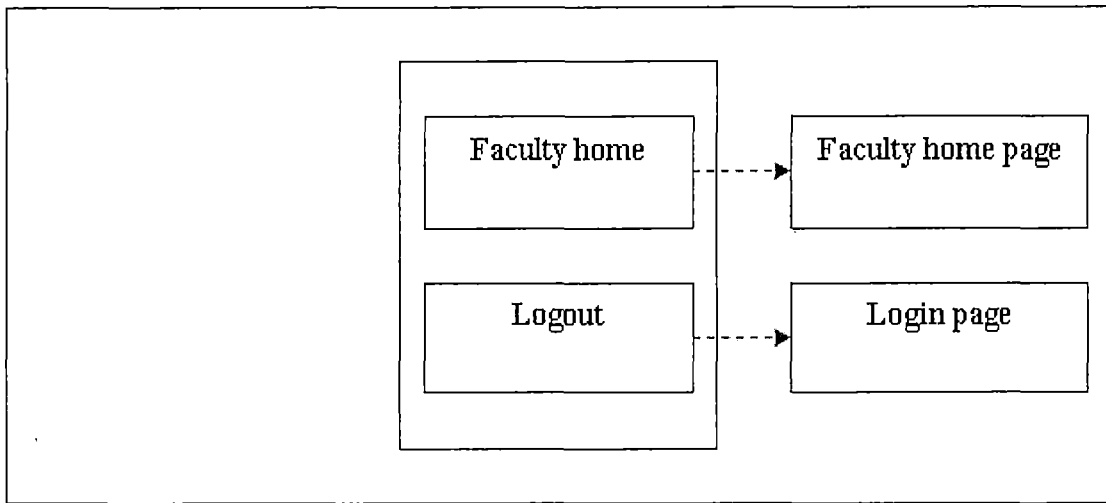


Figure 20. Result Page (Faculty)

### 3.5.3 Administrator

Figure 21 represents the Login Page which is used by the administrator, if the administrator successfully logs in the system, the administrator will be redirected to the Administrator Home Page; if not, the administrator will stay at the Login Page. If the administrator forgets his/her password, the administrator can go to the Password Page to request the password.

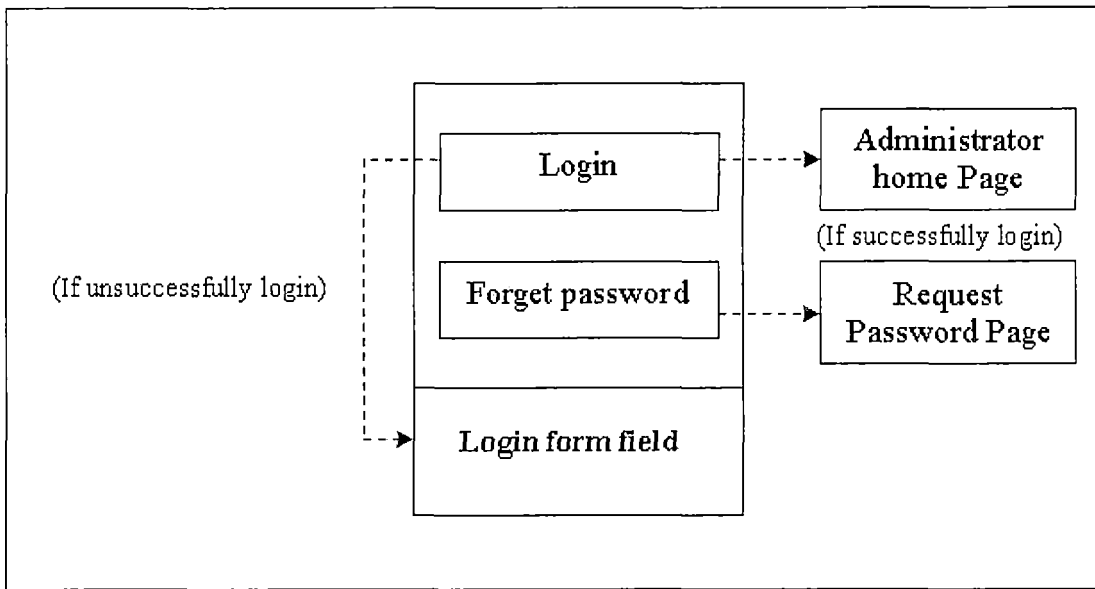


Figure 21. Login Page (Administrator)

Figure 22 represents the Administrator Home Page which is used by the administrator. The administrator can choose which function they are going to use, or logout to the Login Page.

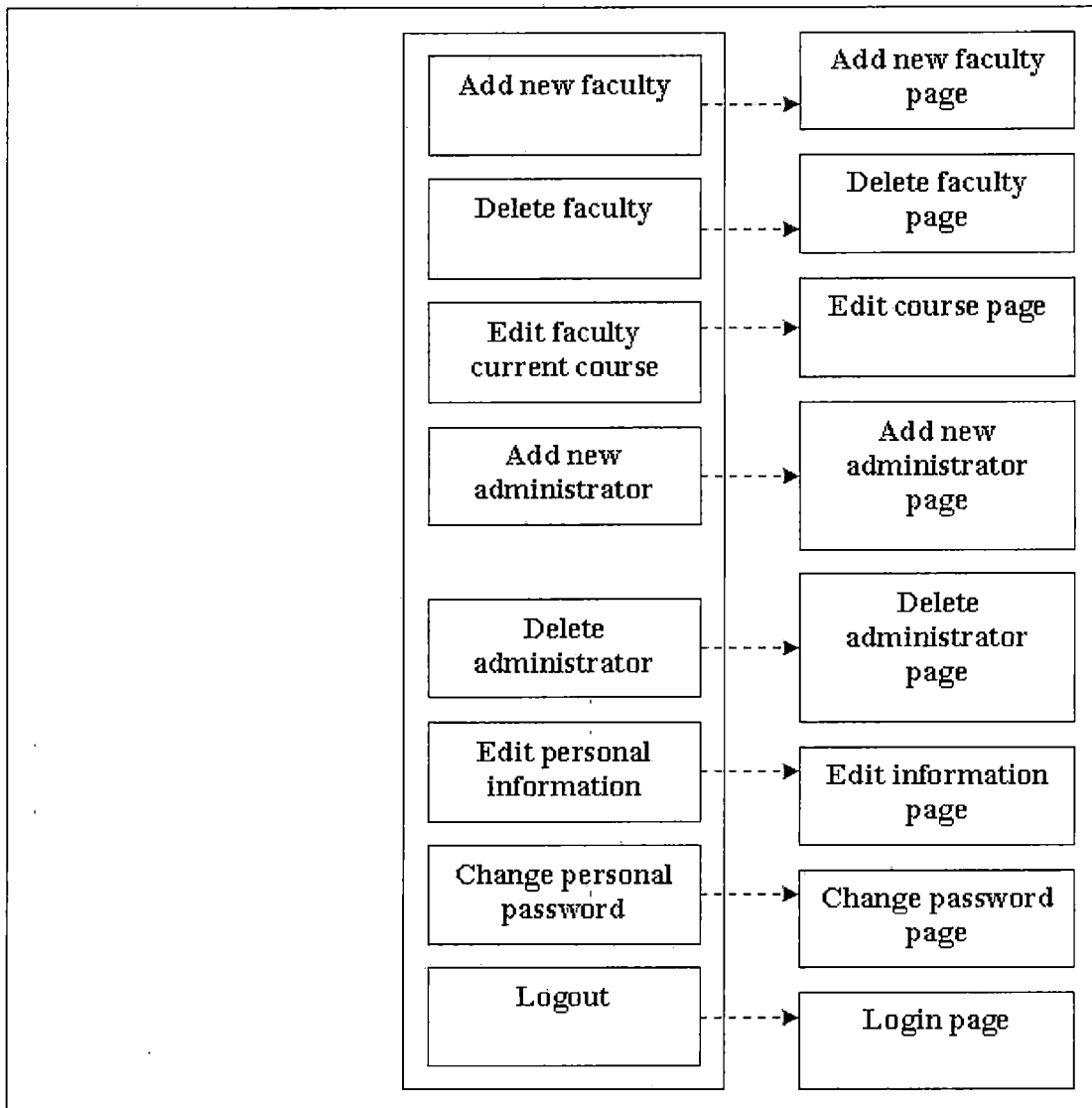


Figure 22. Administrator Home Page

Figure 23 represents the Add New Faculty Page which is used by the administrator. The administrator can add new faculty through this page, or logout to the Login Page.

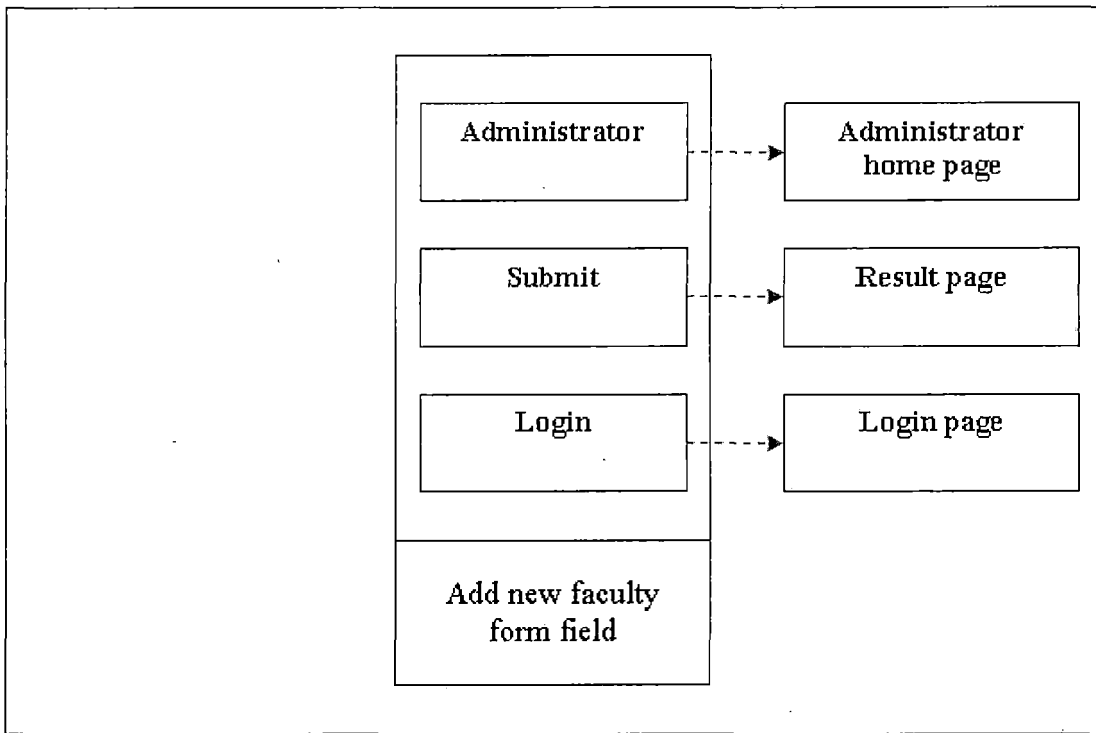


Figure 23. Add New Faculty Page

Figure 24 represents the Delete Faculty Page which is used by the administrator. The administrator can delete faculty through this page, or logout to the Login Page.

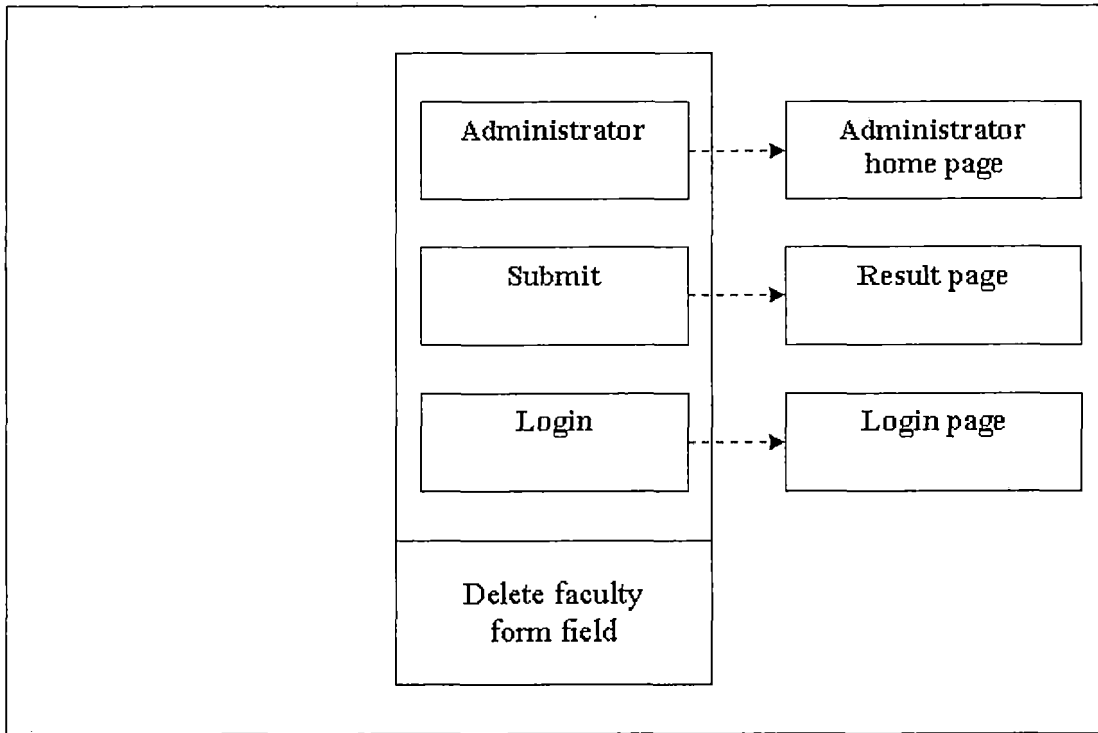


Figure 24. Delete Faculty Page

Figure 25 represents the Edit Current Course Page which is used by administrator. The administrator can edit current course for faculty through this page, or logout to the login page.

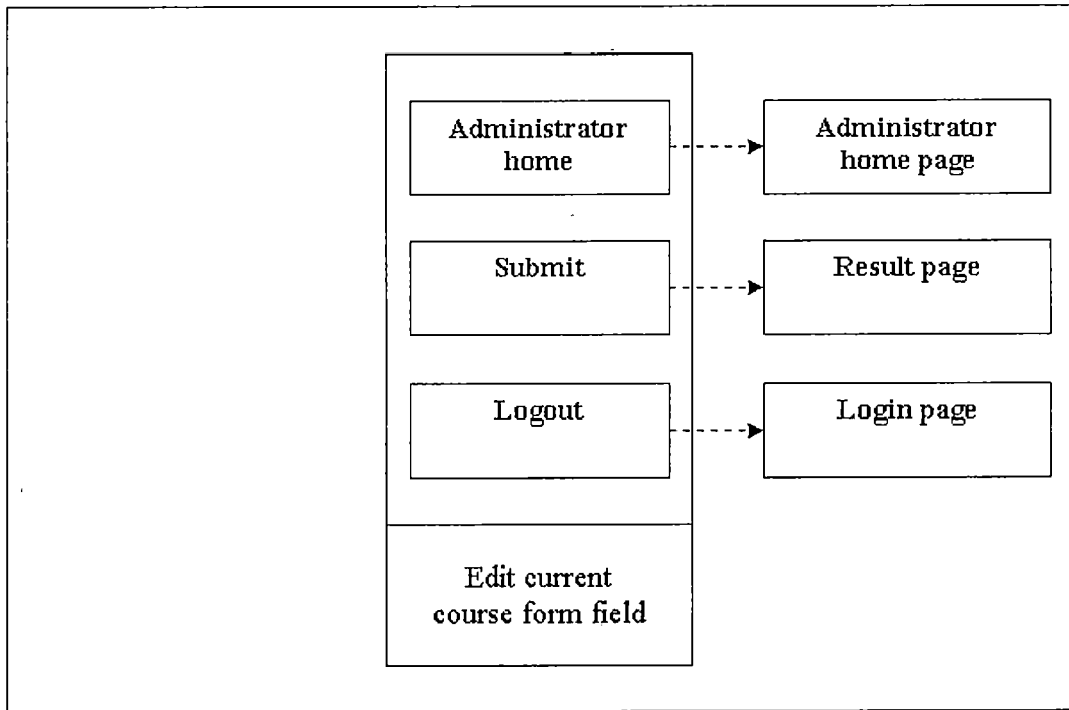


Figure 25. Edit Current Course Page

Figure 26 represents the Add New Administrator Page which is used by the administrator. The administrator can add new administrators through this page, or logout to the Login Page.

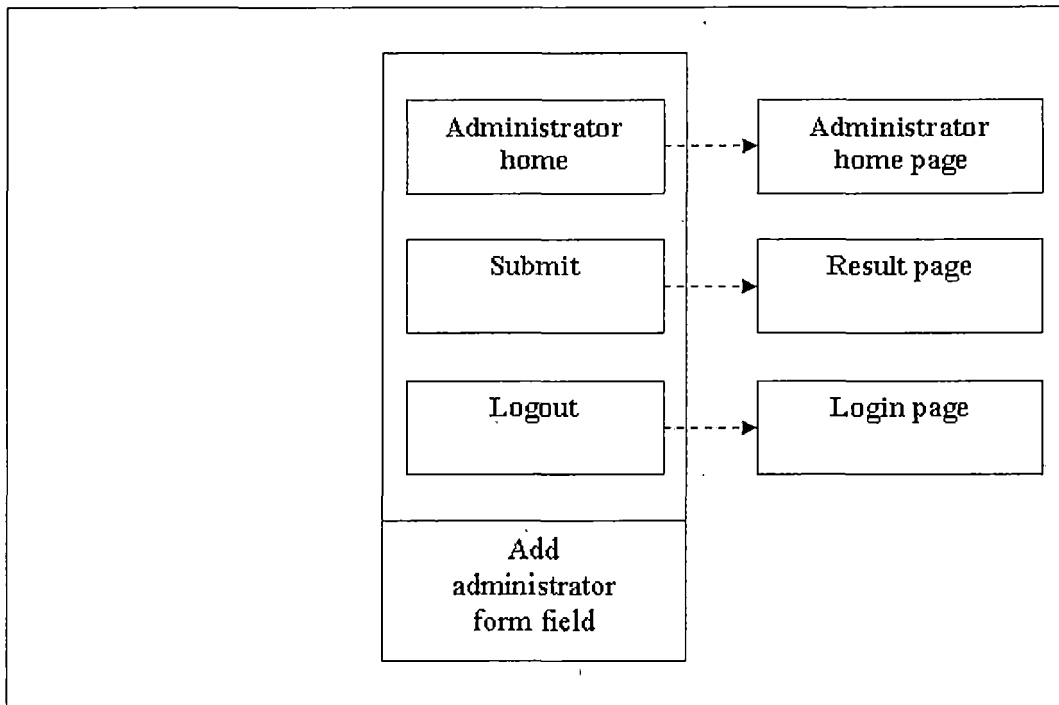


Figure 26. Add New Administrator Page

Page Figure 27 represents the Delete Administrator Page which is used by the administrator. The administrator can delete administrators through this page, or logout to the Login Page.

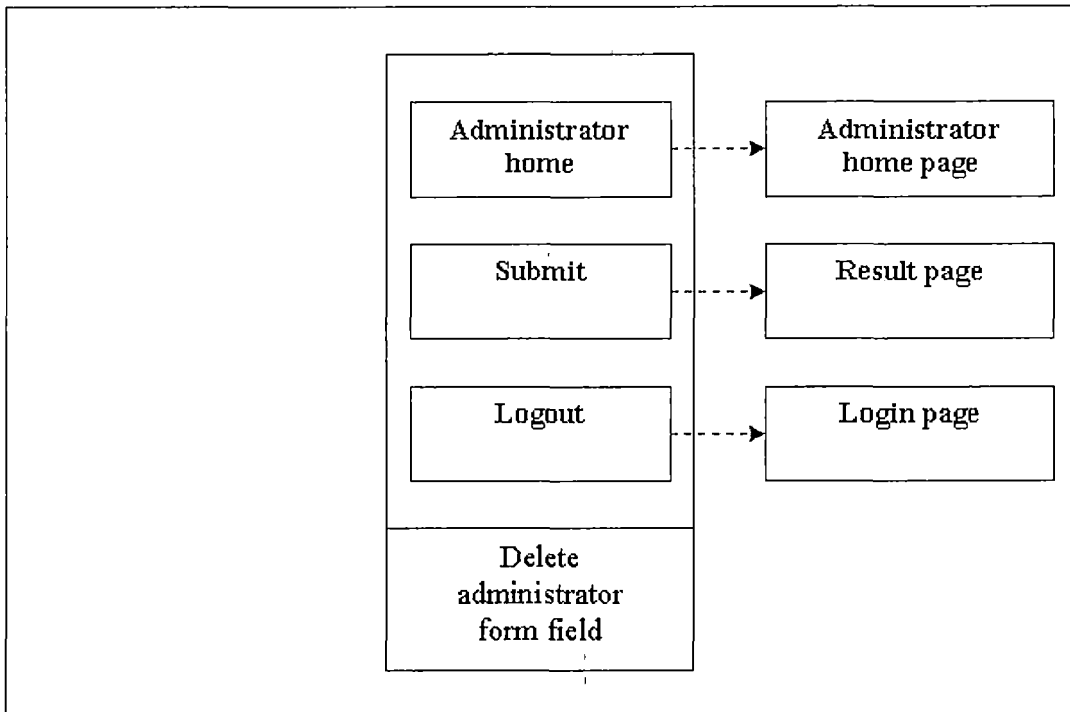


Figure 27. Delete Administrator Page

Figure 28 represents the Edit Personal Information Page which is used by administrator. The administrator can edit personal information through this page, or logout to the login page.



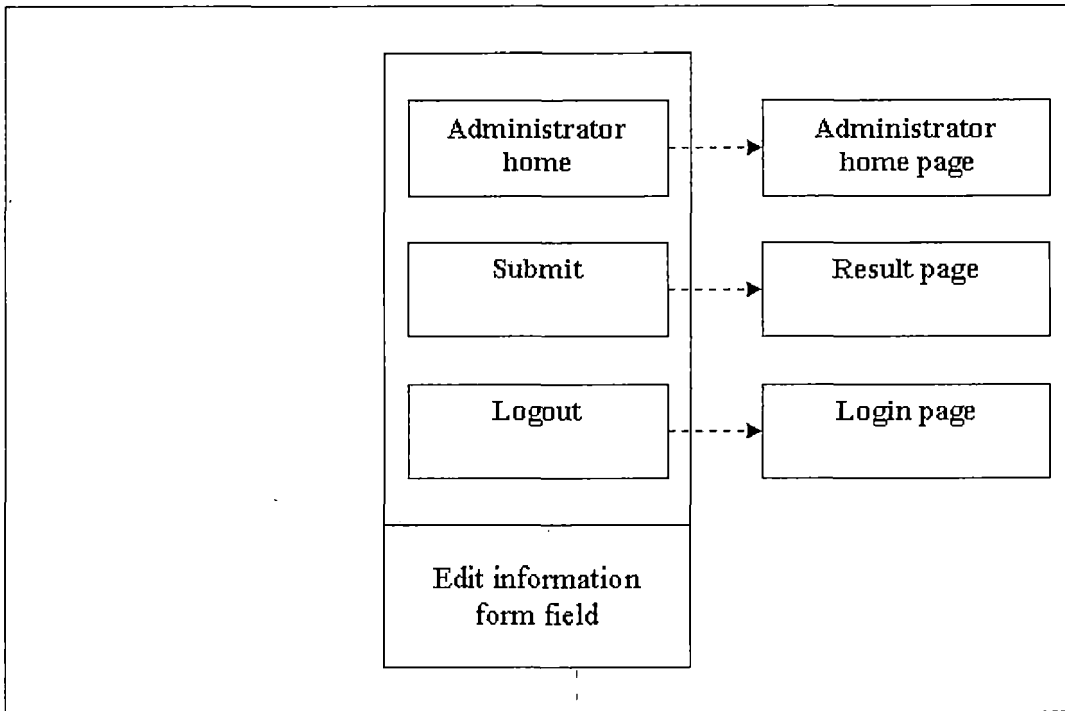


Figure 28. Edit Personal Information Page (Administrator)

Figure 29 represents the Change Password Page which is used by the administrator. The administrator can change his/her password through this page, or logout to the Login Page.

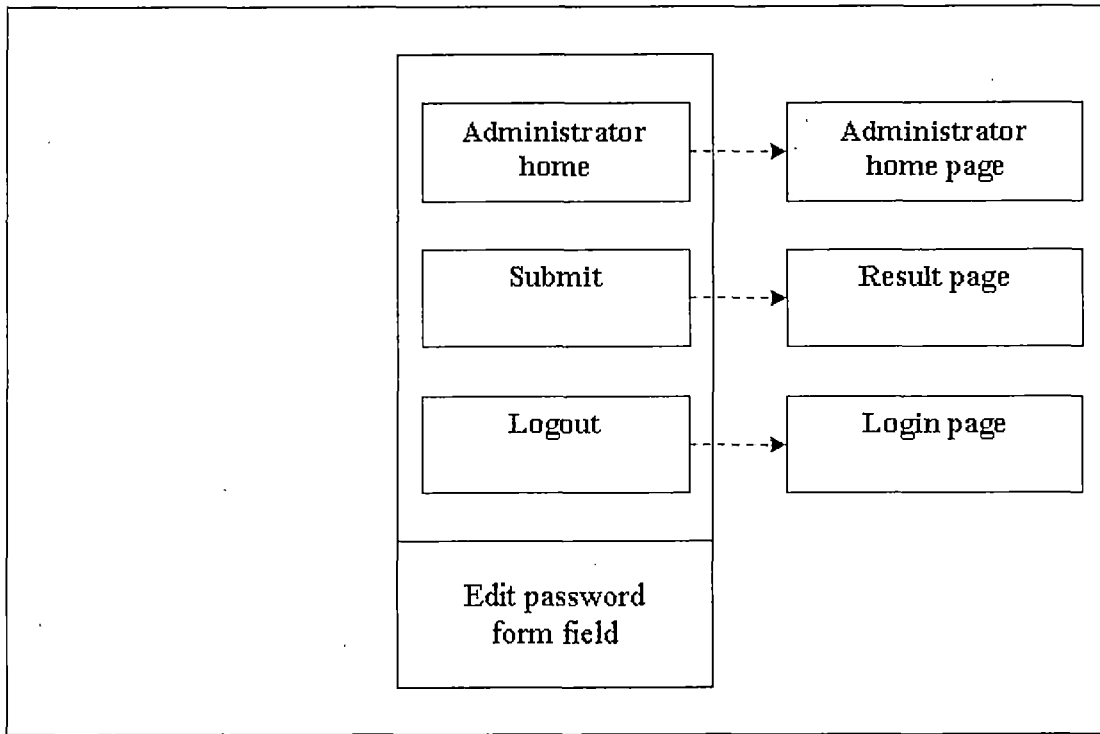


Figure 29. Change Password Page (Administrator)

Figure 30 represents the Result Page which is used by the administrator. The administrator can see the result through this page, or logout to the Login Page.

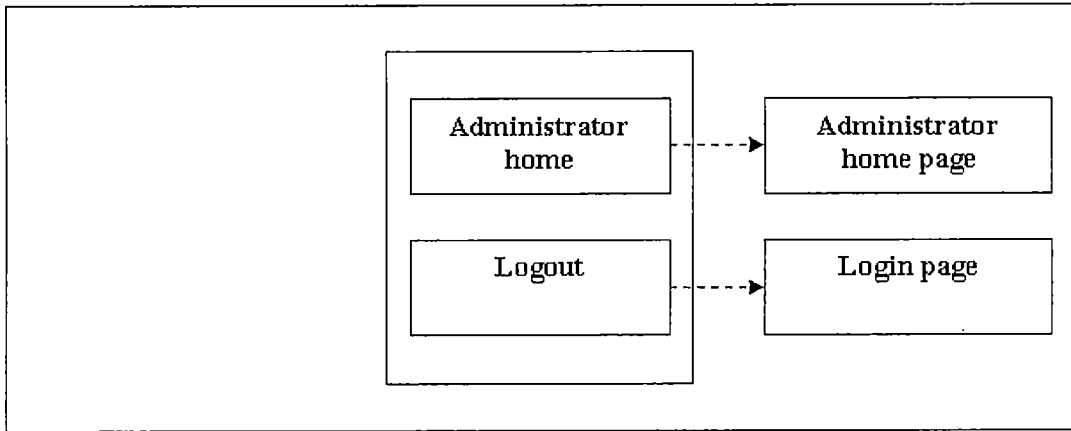


Figure 30. Result Page (Administrator)

## CHAPTER FOUR

### PROJECT IMPLEMENTATION

#### 4.1 User Interface Design

User interfaces are established on the web and therefore the system can use all user interface features provided by the web browser, including plug-ins and any added functionality that the browser may possess. Any standard web browser, such as Microsoft Internet Explorer or Netscape Navigator/Communicator can provide the user interface to the system. The user interfaces will be generated dynamically on the server side using JSP. These user interfaces will provide information tailored to the user who is logged in.

#### 4.2 Graphical User Interface and Description

##### 4.2.1 Login Page

This page is the starting page for all people (Student, Faculty and System Administrator) who are going to use this software product. Faculty and Administrator must be created by System Administrator. Students can click on the New User link in this page to create an account. All users can log in by providing a username and a password on this page. The login jsp will verify the username and password. If it is

correct, it forwards to different page base on the "ID" of each user.

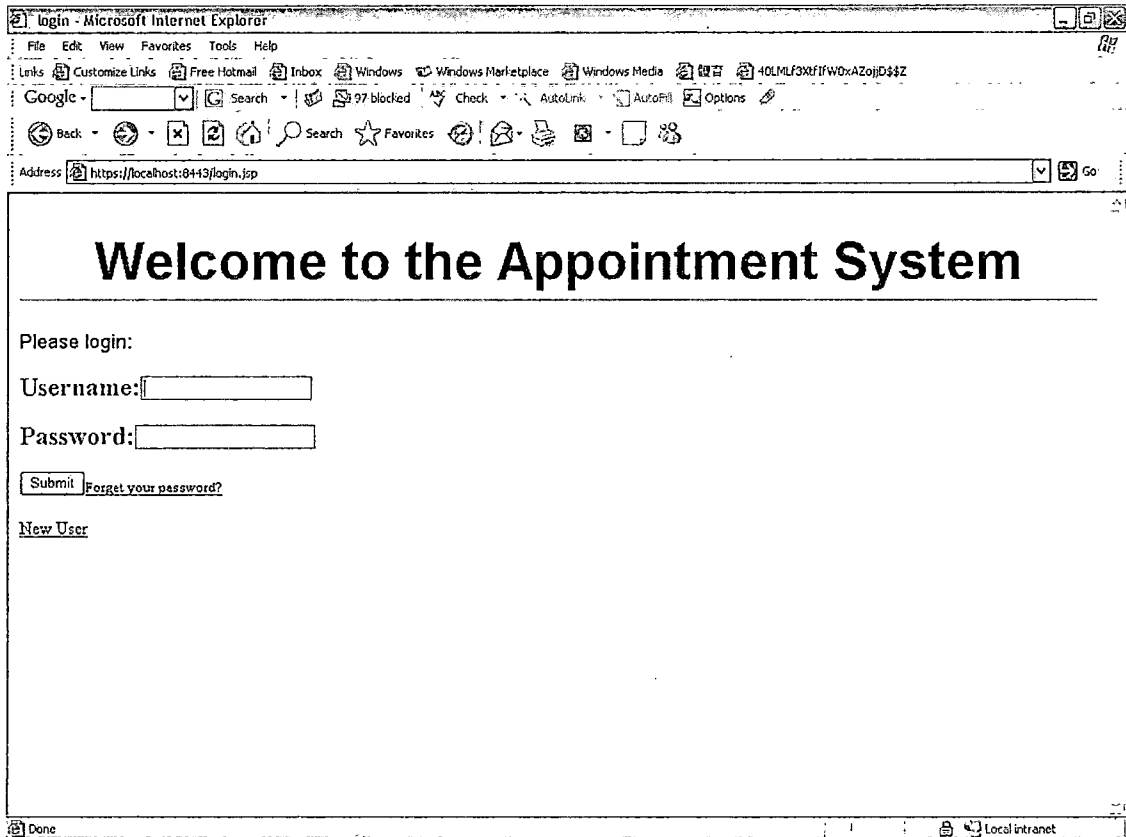


Figure 31. Login Page

#### 4.2.2 Student Home Page

On this page, there are four pages that students can choose.

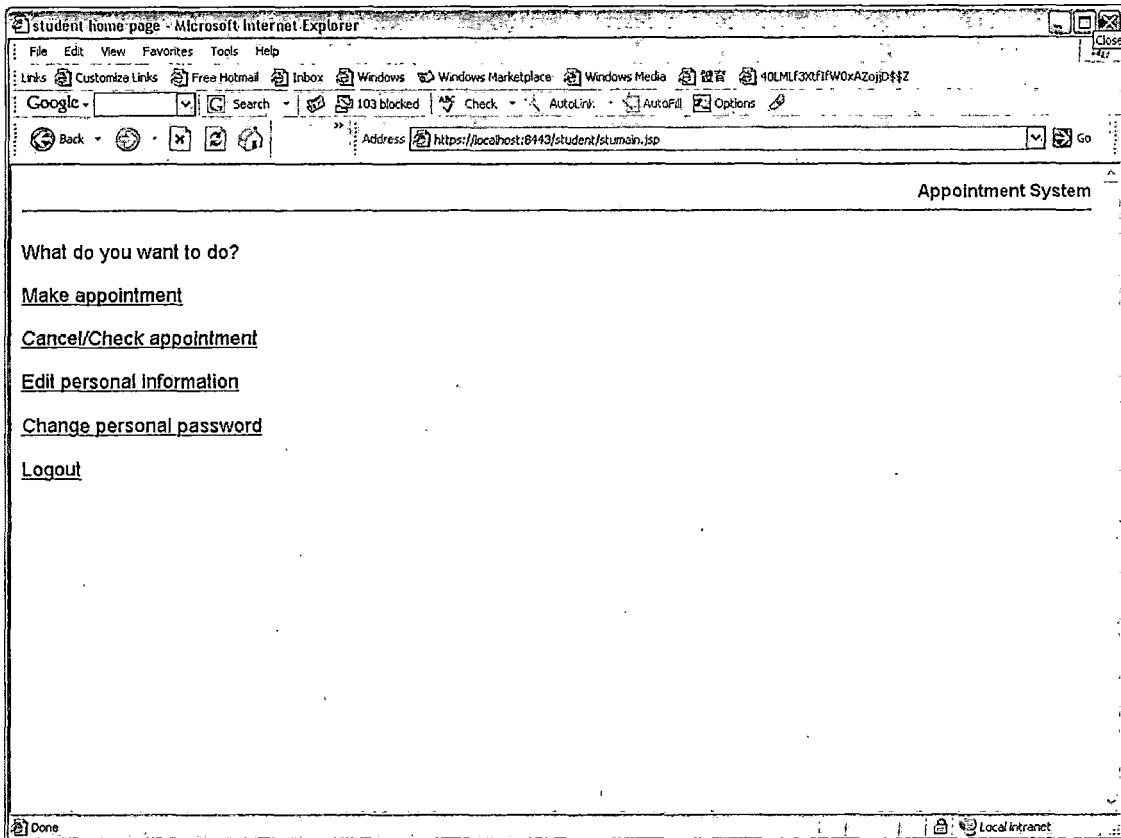


Figure 32. Student Home Page

#### 4.2.3 Student Make Appointment Page

On this page, students can pick one faculty to see the information of his/her available time and then make the appointment. After the student making the appointment, the system will send emails to the student and the faculty.

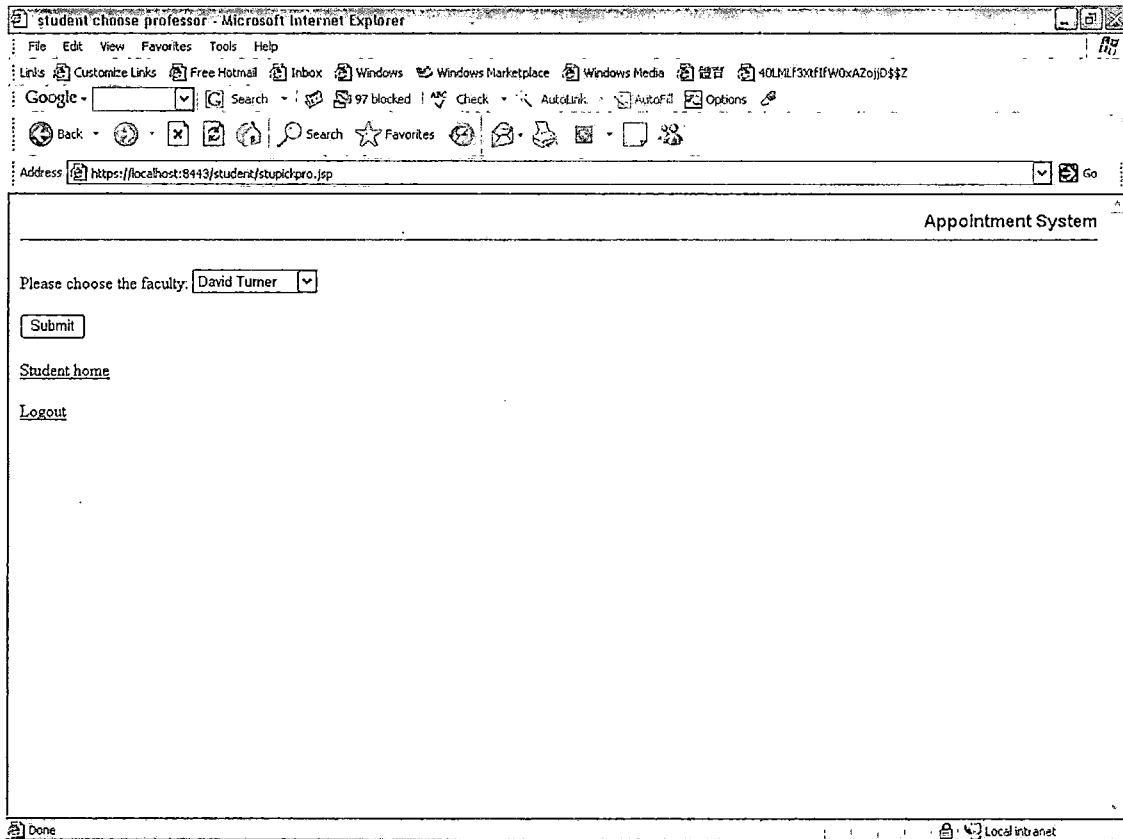


Figure 33. Student Make Appointment Page

#### 4.2.4 Student Cancel Appointment Page

This page will list the details of the appointments for the students. Students can click the "cancel" to cancel appointments.

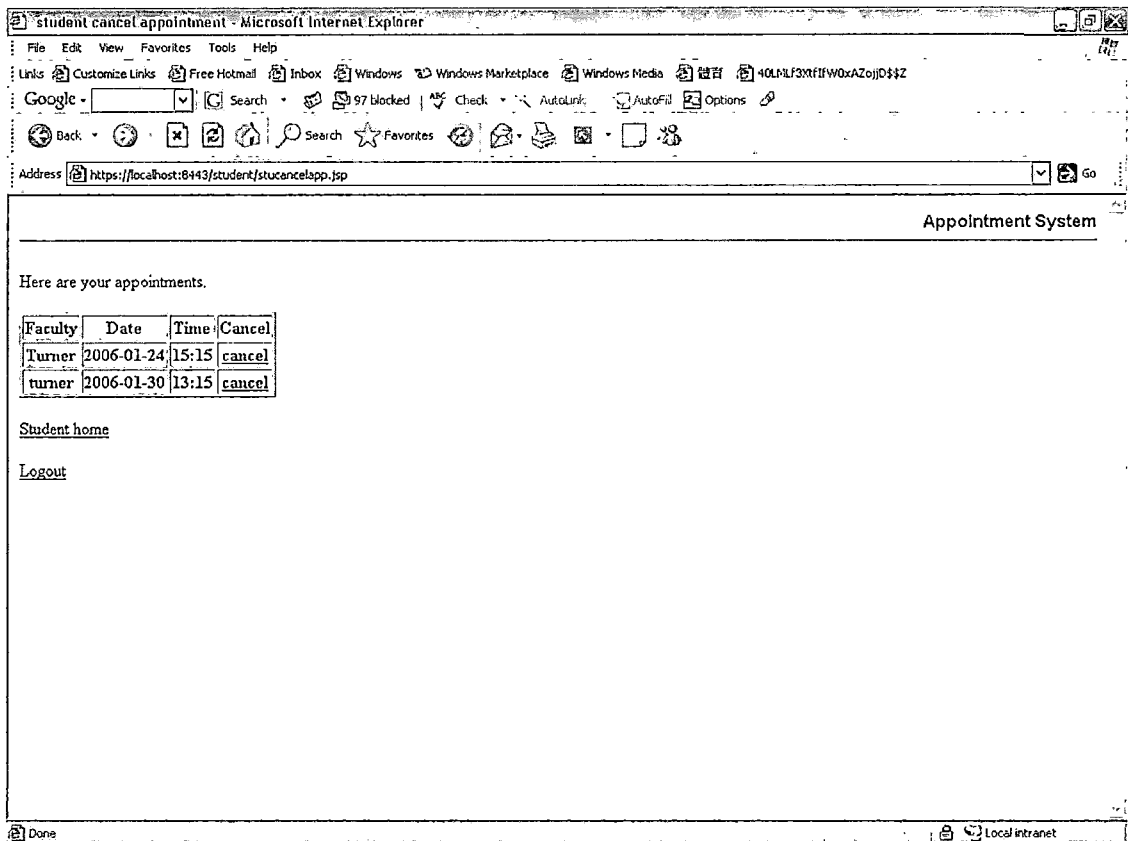


Figure 34. Student Cancel Appointment Page

#### 4.2.5 Student Edit Personal Information Page

On this page, students can edit personal information.



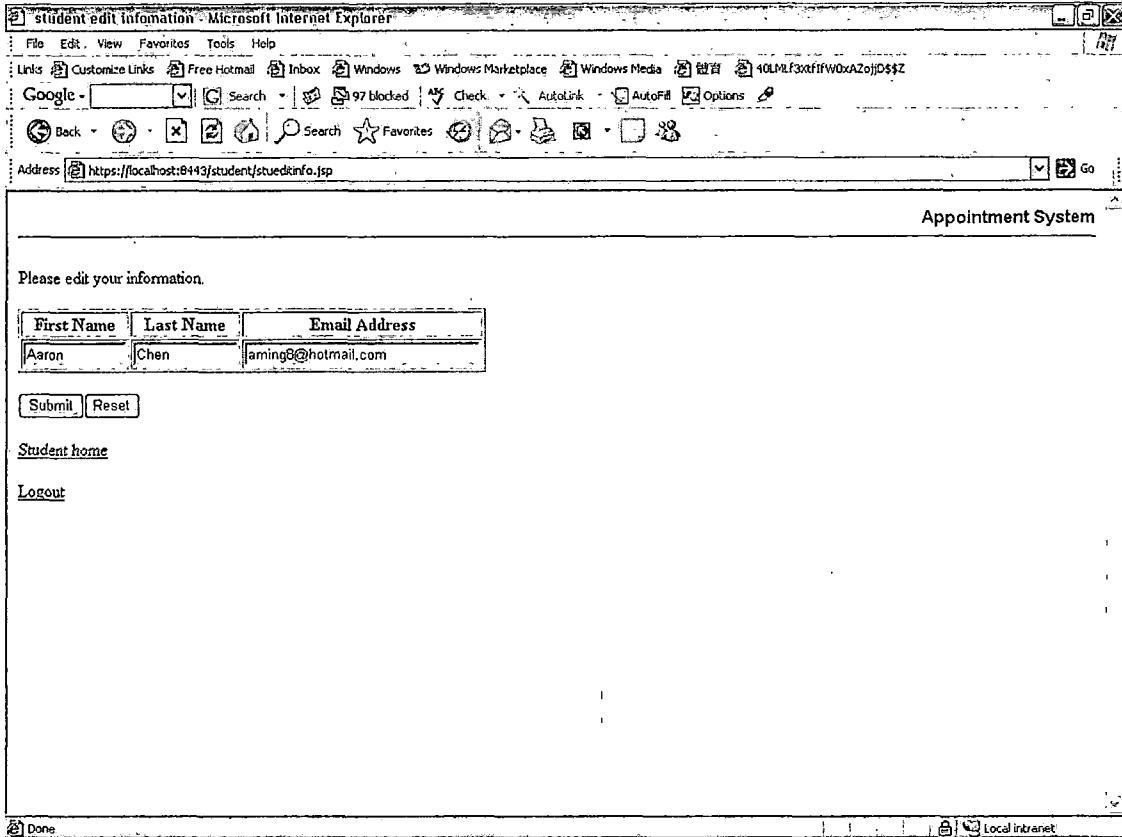


Figure 35. Student Edit Personal Information Page

#### 4.2.6 Change Password Page

This page can let users to change personal password. After users changing the password, the system will send the email to users with the new password.

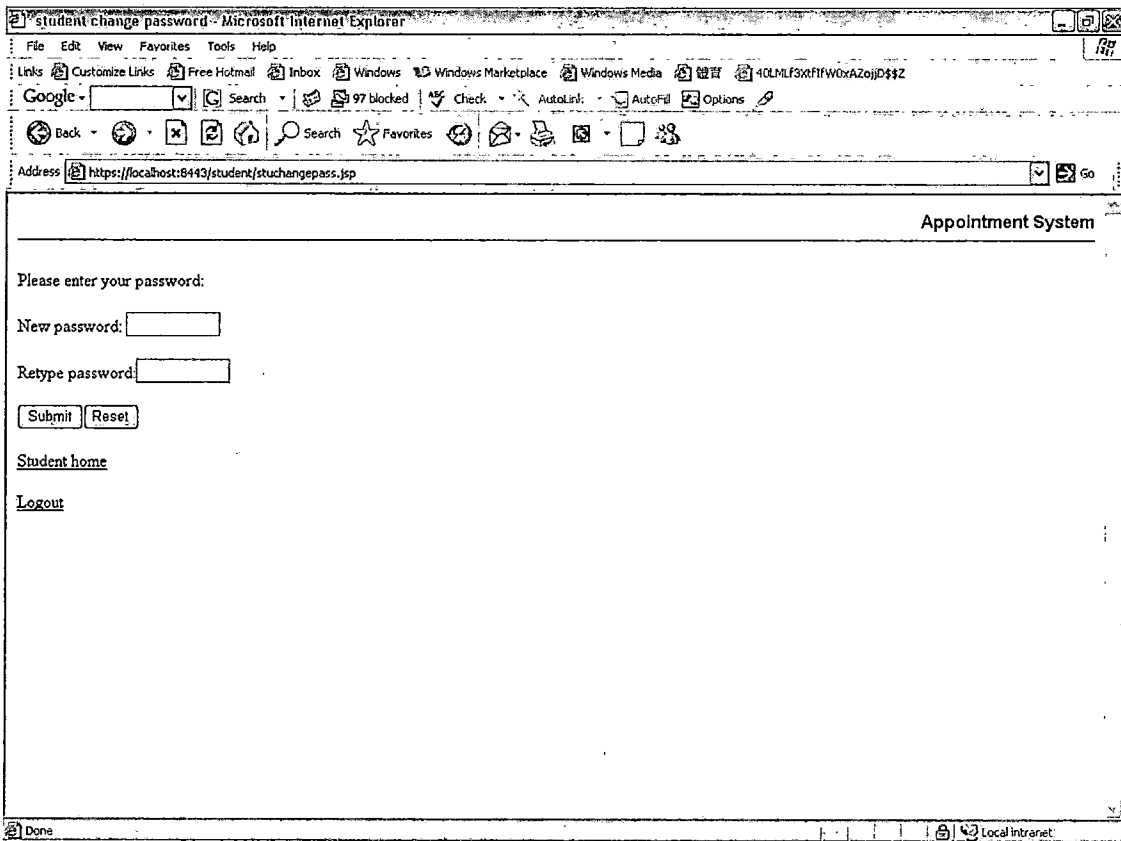


Figure 36. Change Password Page

#### 4.2.7 Faculty Home Page

On this page, there are five pages faculty can choose.

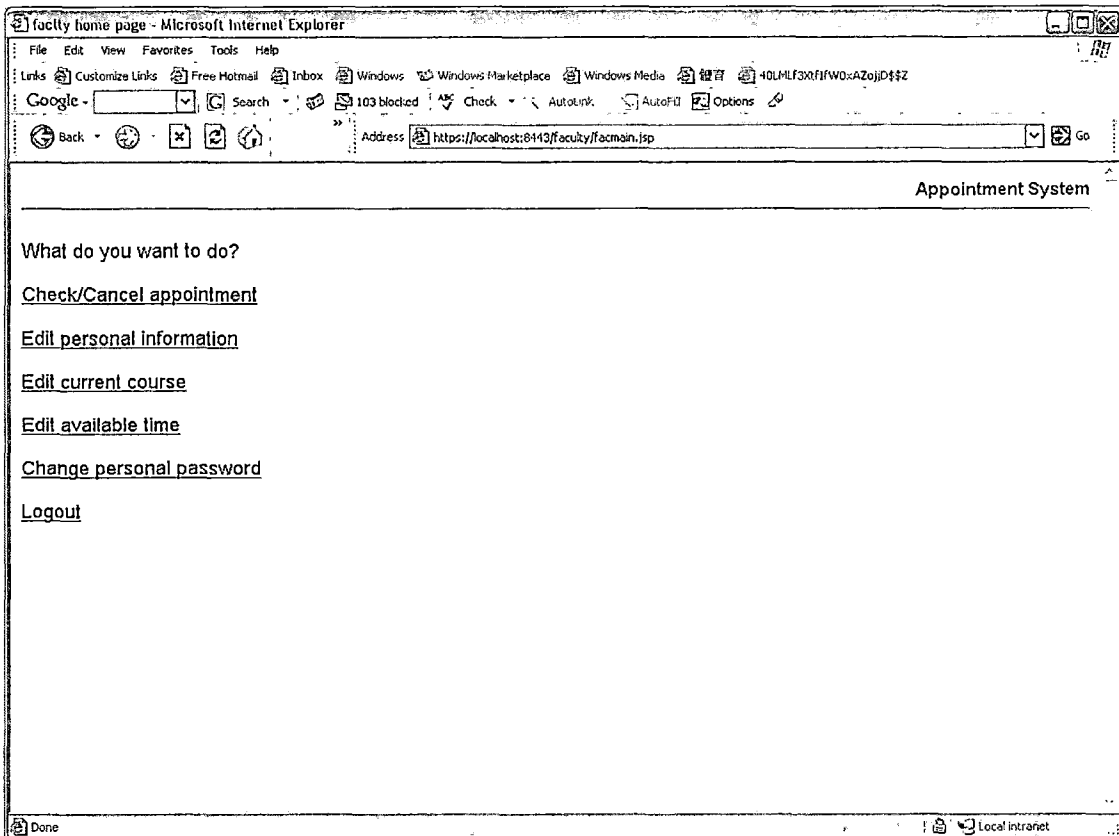


Figure 37. Faculty Home Page

#### 4.2.8 Faculty Check/Cancel Appointment Page

This page lets faculty to choose showing all appointments, appointments in a specific range or the appointments on one specific day. Once the page shows the appointments, faculty can cancel any appointment they want.

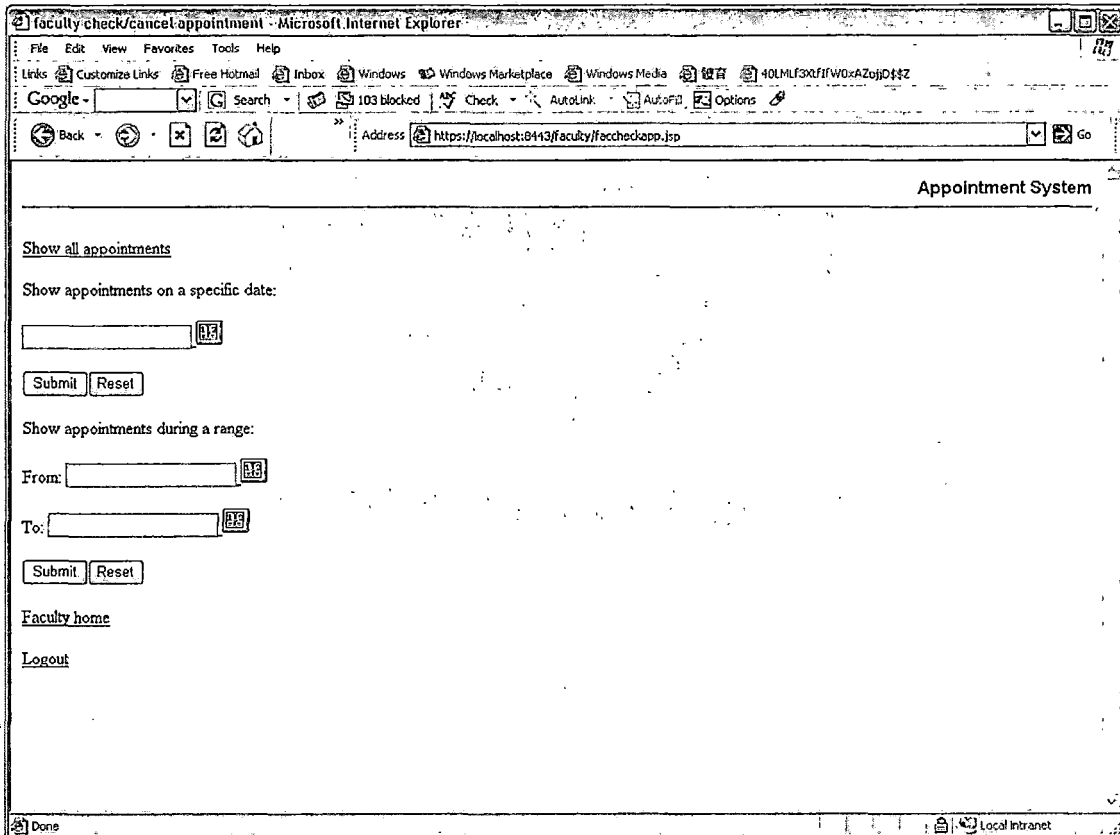


Figure 38. Faculty Check/Cancel Appointment Page

#### 4.2.9 Faculty Edit Personal Information Page

On this page, faculty can edit personal information.

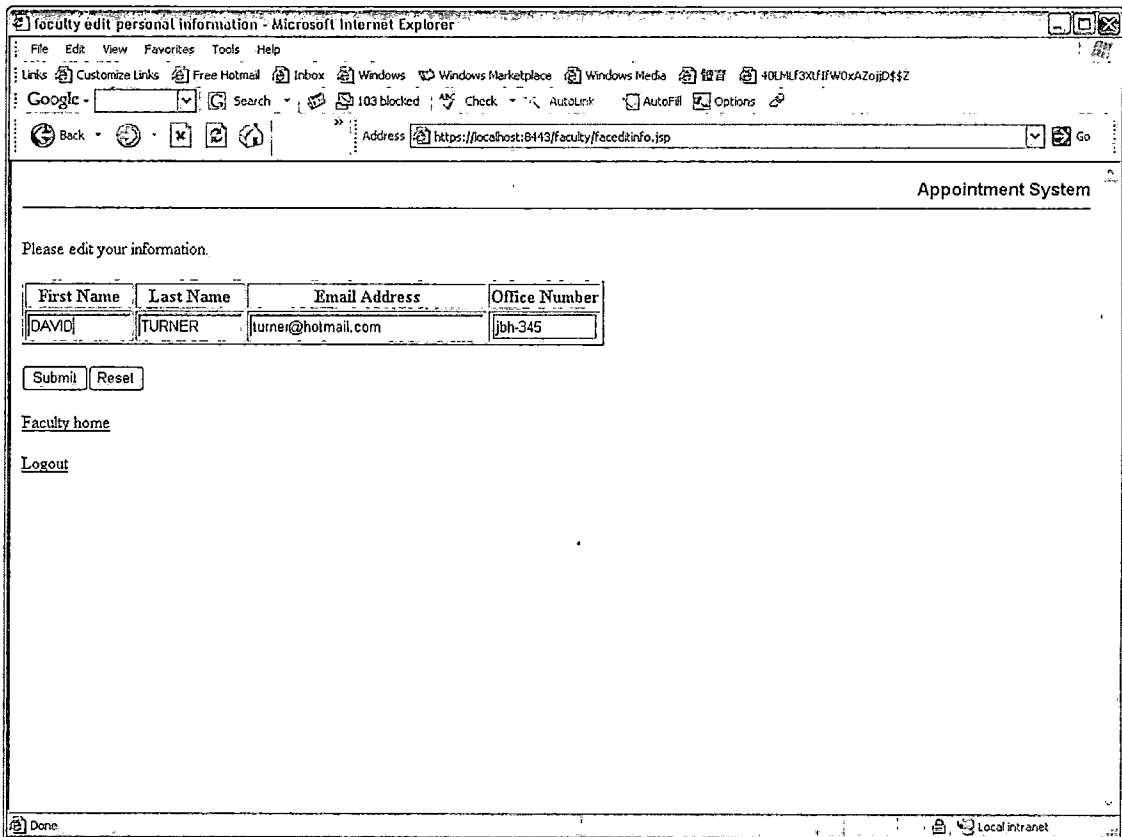


Figure 39. Faculty Edit Person Information Page

#### 4.2.10 Faculty Edit Current Course Page

On this page, faculty can edit or delete their current courses. Faculty also can add new current courses by clicking the link.

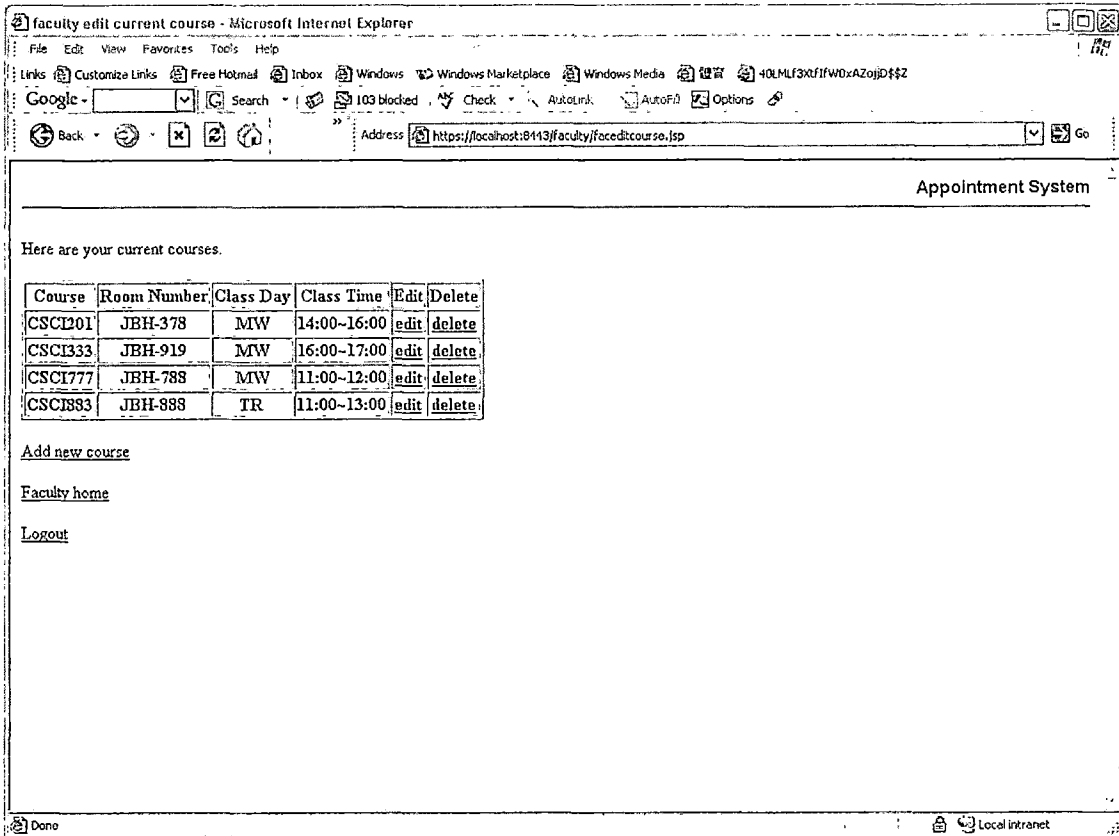


Figure 40. Faculty Edit Current Course Page

#### 4.2.11 Faculty Edit Available Time Page

On this page, faculty can edit or delete their available times. Faculty also can add new available times by clicking the link.

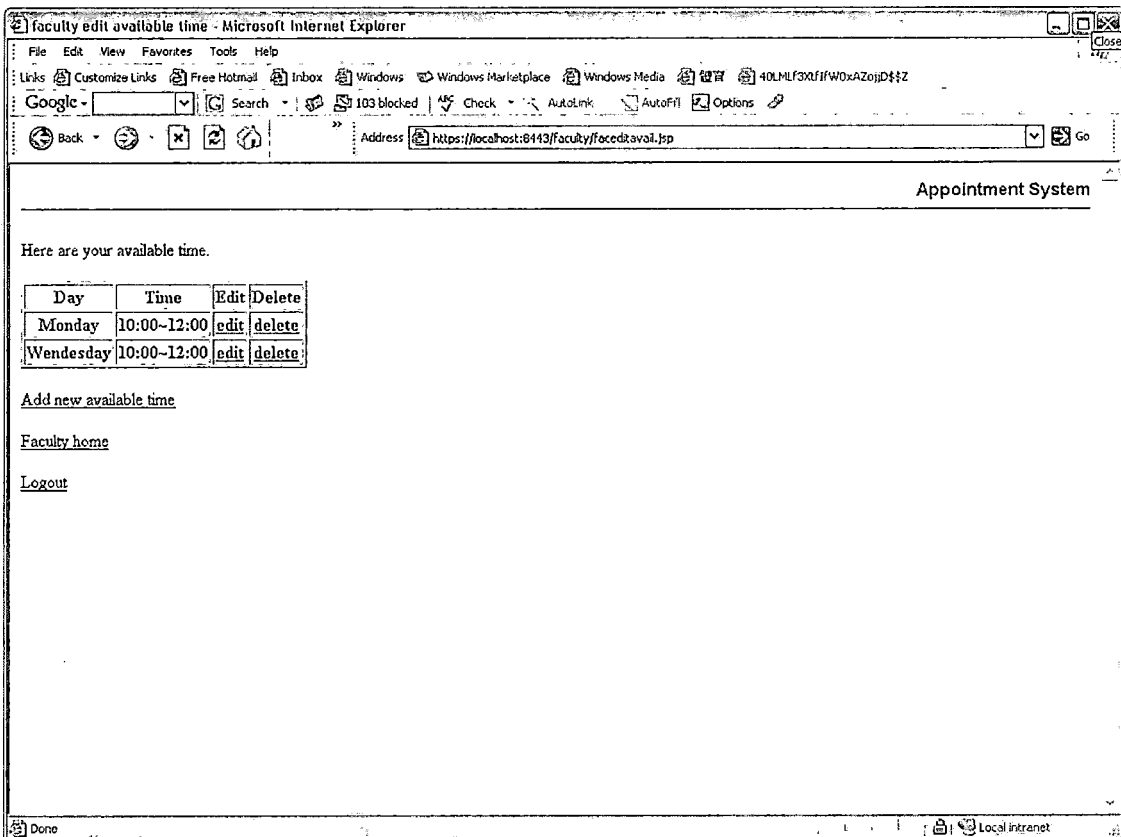


Figure 41. Faculty Edit Available Time Page

#### 4.2.12 Administrator Home Page

On this page, the system administrator can click the links to add and delete faculty, edit faculty current courses, add and delete new administrators, edit personal information or change personal password.

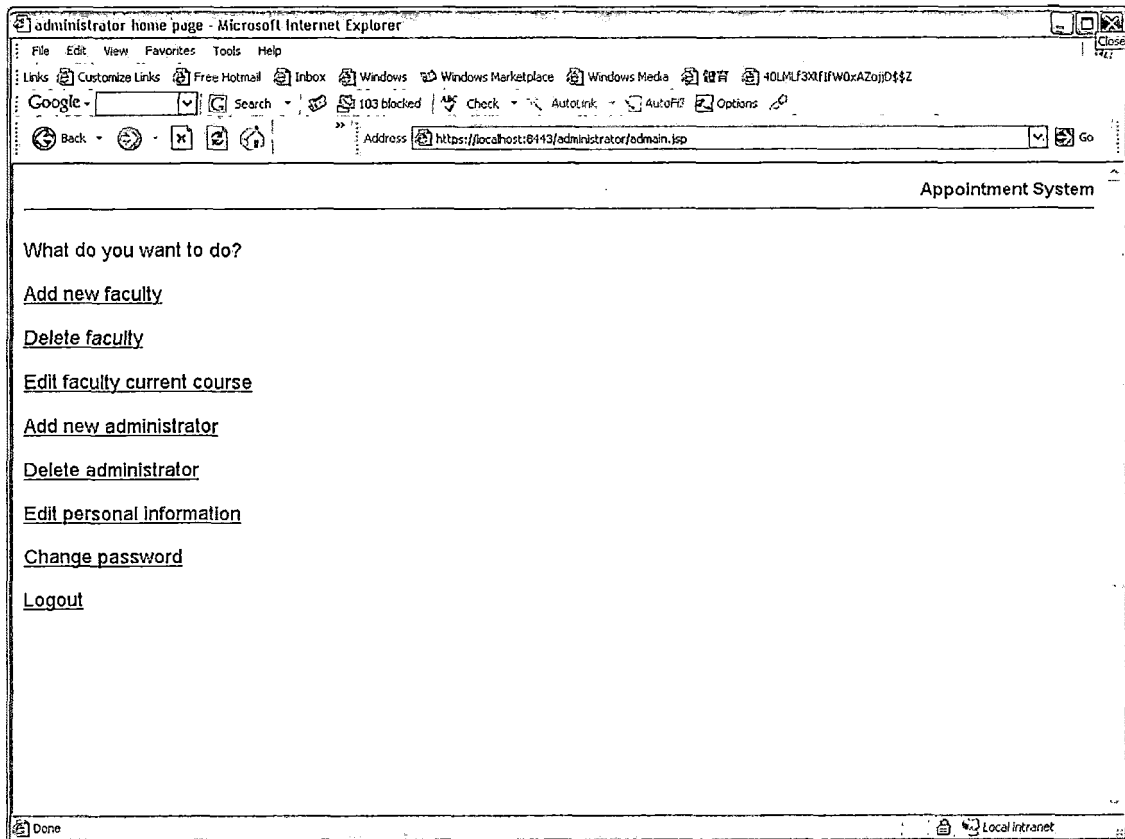


Figure 42. Administrator Home Page

#### 4.2.13 Administrator Add New Faculty Page

On this page, the system administrator can edit information to add new faculty.



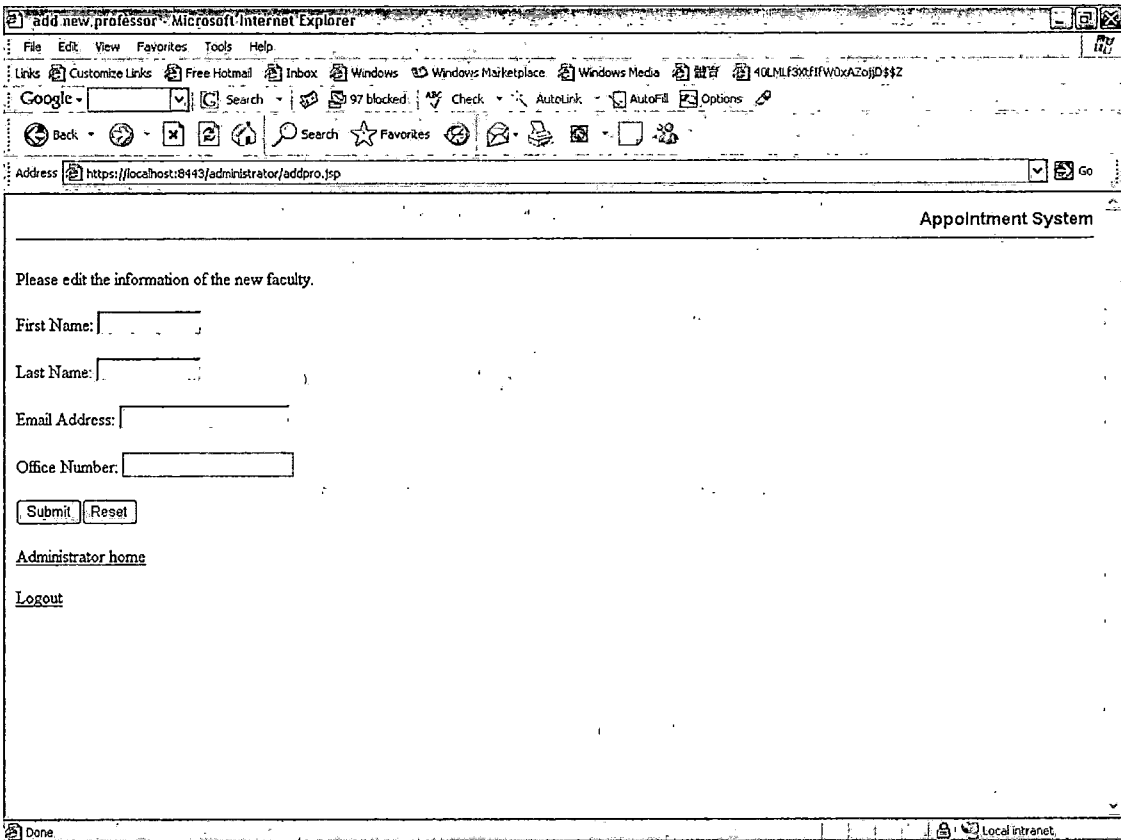


Figure 43. Administrator Add New Faculty Page

#### 4.2.14 Administrator Delete Faculty Page

On this page, the system administrator can delete the faculty who is going to leave the campus.

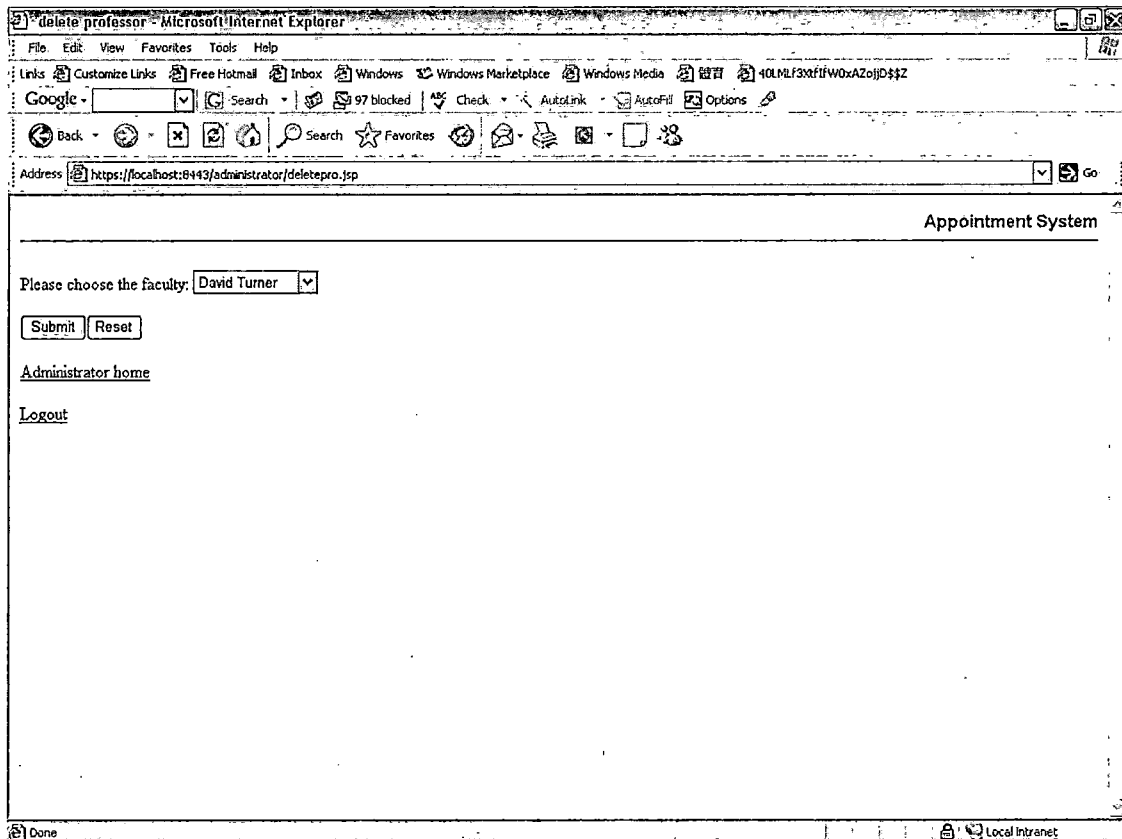


Figure 44. Administrator Delete Faculty Page

#### 4.2.15 Edit Faculty Current Course Page

On this page, the system administrator can pick a faculty to add and edit current courses.

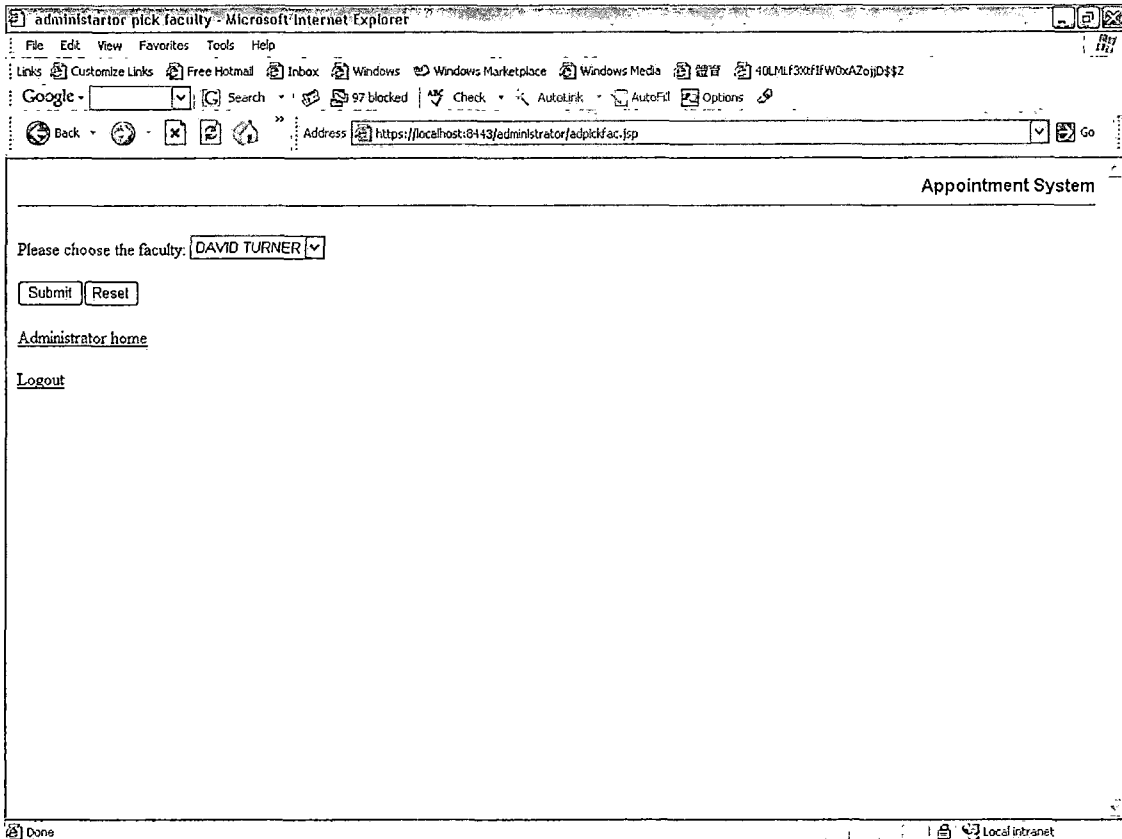


Figure 45. Administrator Edit Current Course Page

#### 4.2.16 Edit Personal Information Page

On this page, the system administrator can edit his/her personal information.

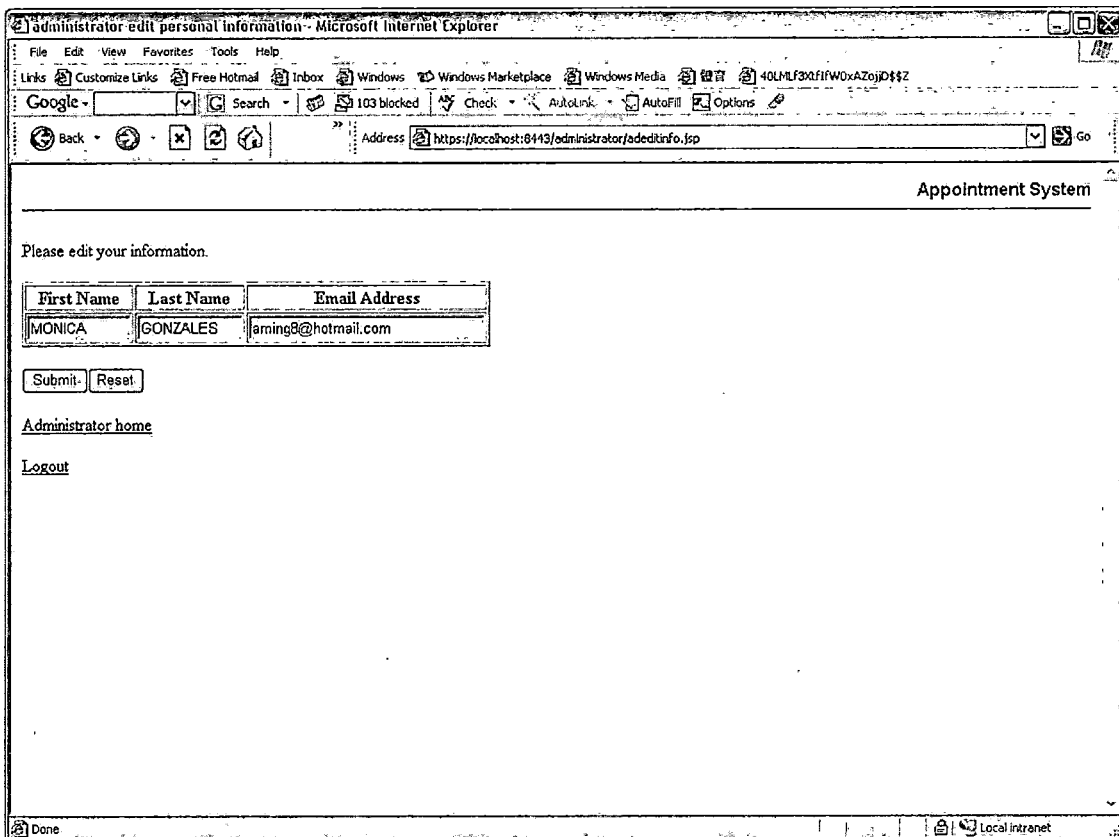


Figure 46. Administrator Edit Personal Information Page

#### 4.2.17 Add New Administrator Page

On this page, the system administrator can add new administrators.

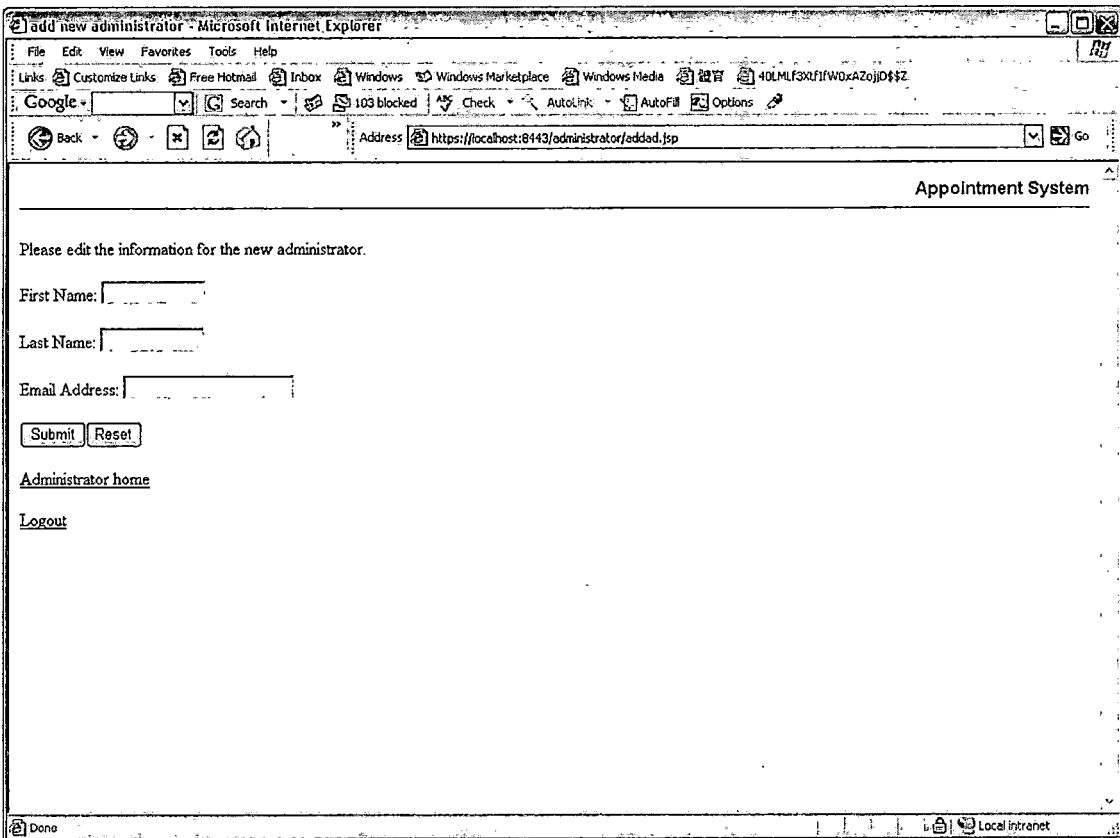


Figure 47. Administrator Add New Administrator Page

#### 4.2.18 Delete Administrator Page

On this page, the system administrator can delete administrators.

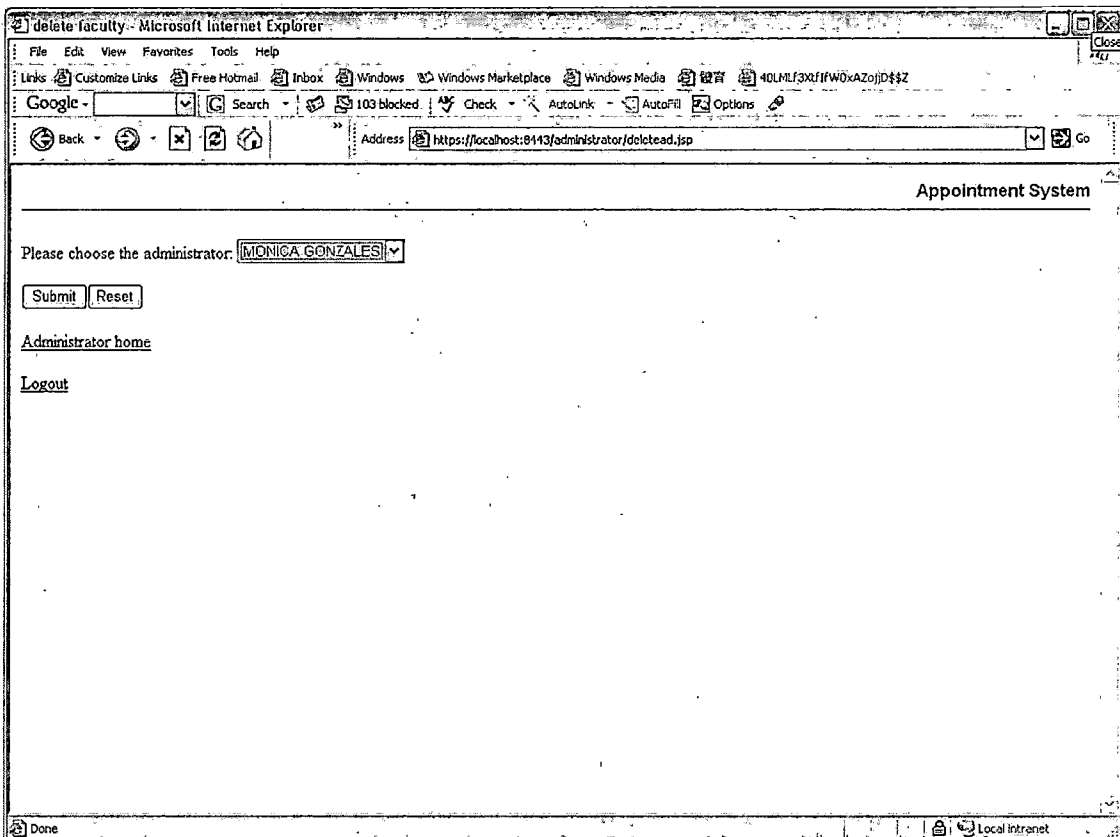


Figure 48. Administrator Delete Administrator Page

#### 4.2.19 Request Password Page

On this page, users provide their ID and email to ask their password. The Appointment System will send an email with new password to the email address which the user entered.

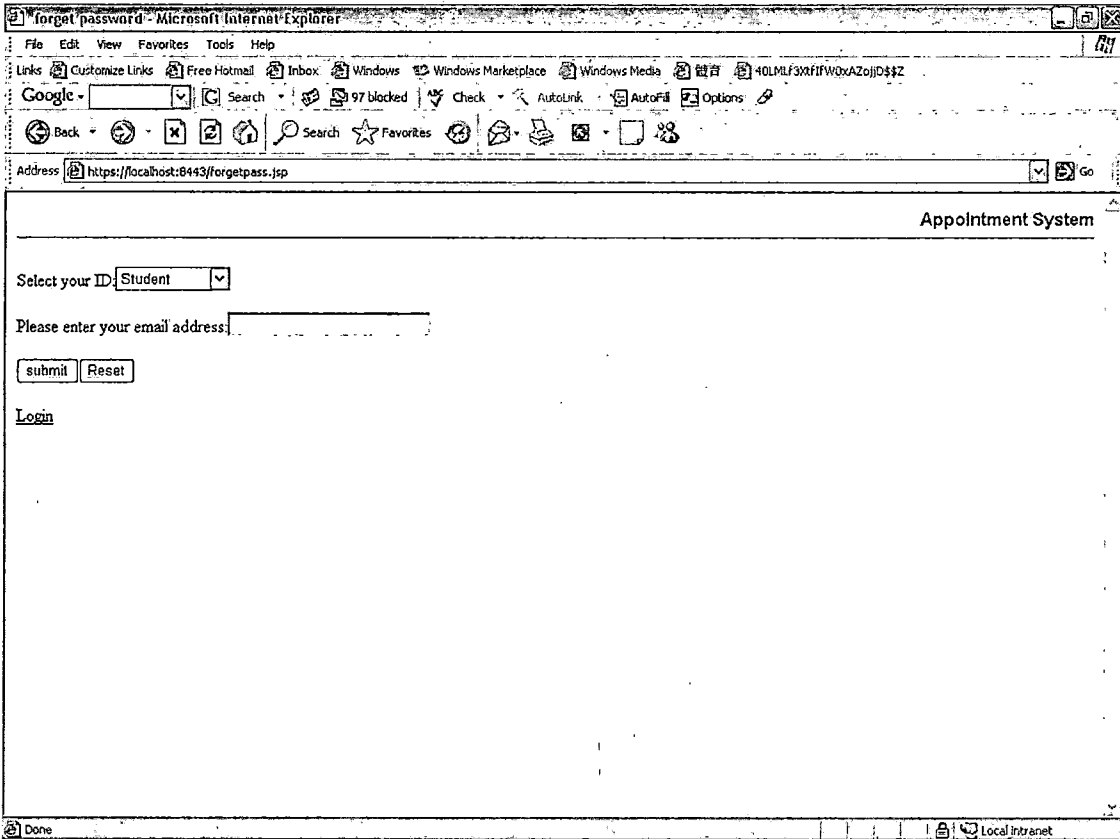


Figure 49. Request Password Page

#### 4.2.20 Create New User Page

New students can create new accounts through this page.

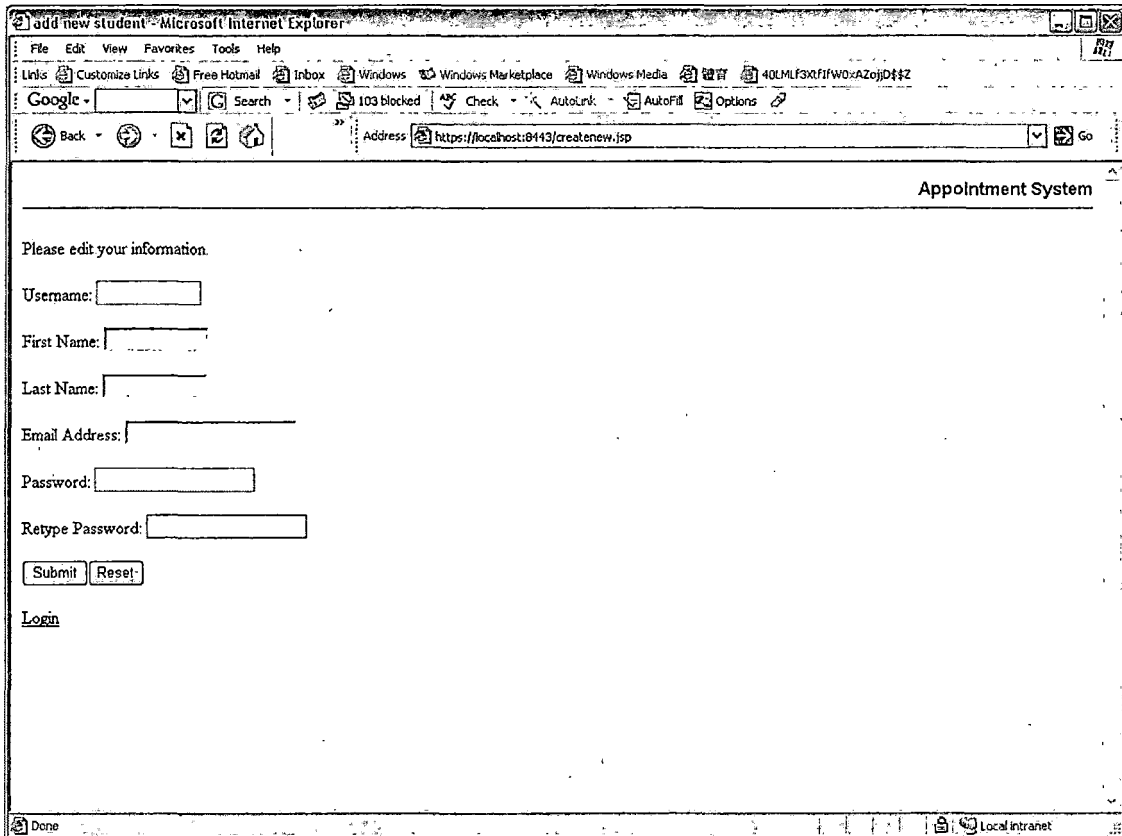


Figure 50. Create New User Page



CHAPTER FIVE  
SYSTEM VALIDATION

To validate a system, just having an understanding of computers and software systems is not adequate. It is essential to fully understand the process and the equipment that is being validated. Validation testing is a concern that overlaps with integration testing. Ensuring that the application fulfils its specification is a major criterion for the construction of an integration test. Validation testing also overlaps to a large extent with system testing, where the application is tested with respect to its typical working environment.

5.1 Unit Test

The Unit test presents the basic level of testing on all the individual components. The individual components include the object, the class, and the program in the system. The following table shows the results of the unit test for the Appointment System.

Table 8. The Unit Test Results

Unit Test	Tests Performed	Results
Login page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data in JSP included pages components.</li> <li>• Verify the login function working properly and get the error message properly.</li> </ul>	OK
Student home page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Check the links working properly.</li> </ul>	OK
Student make appointment page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Check the entire selection list, buttons, and links working properly.</li> <li>• Check JavaScript function working properly.</li> </ul>	OK
Student cancel appointment page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Verify all links working as expected.</li> <li>• Verify the page can get the error message and work properly by the message.</li> </ul>	OK
Student edit personal information page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Verify all links and buttons working as expected.</li> <li>• Test the correctness of the validation for all input data.</li> <li>• Verify the page can get the error message and work properly by the message.</li> </ul>	OK
change password page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Verify all links and buttons working as expected.</li> <li>• Test the correctness of the validation for all input data.</li> <li>• Verify the page can get the error message and work properly by the message.</li> </ul>	OK
Faculty home page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Verify all links working as expected.</li> <li>• Verify the page can get the error message and work properly by the message.</li> </ul>	OK
Faculty check/cancel appointment page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Verify all links and buttons working as expected.</li> <li>• Test the correctness of the validation for all input data.</li> <li>• Verify the page can get the error message</li> </ul>	OK

Unit Test	Tests Performed	Results
Faculty edit personal information page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Verify all links and buttons working as expected.</li> <li>• Test the correctness of the validation for all input data.</li> <li>• Verify the page can get the error message and working properly by the message.</li> <li>• Verify the form is generated correctly.</li> </ul>	OK
Faculty edit current course	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Verify all links working as expected.</li> <li>• Verify the page can get the error message and working properly by the message.</li> </ul>	OK
Faculty edit available time page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Verify all links working as expected.</li> <li>• Verify the page can get the error message and working properly by the message.</li> </ul>	OK
Administrator home page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Verify all links working as expected.</li> </ul>	OK
Administrator add faculty page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Check the entire buttons and links working properly.</li> <li>• Verify all links and buttons working as expected.</li> <li>• Verify the form is generated correctly.</li> </ul>	OK
Administrator delete faculty page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Check the entire buttons and links working properly.</li> <li>• Verify all links and buttons working as expected.</li> </ul>	OK
Administrator edit faculty current course	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Verify all links working as expected.</li> <li>• Verify the page can get the error message and working properly by the message.</li> </ul>	OK
Administrator edit personal information page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Verify all links and buttons working as expected.</li> <li>• Test the correctness of the validation for all input data.</li> <li>• Verify the page can get the error message and working properly by the message.</li> </ul>	OK

Unit Test	Tests Performed	Results
	<ul style="list-style-type: none"> <li>• Verify the form is generated correctly.</li> </ul>	
Administrator add administrator page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Check the entire buttons and links working properly.</li> <li>• Verify all links and buttons working as expected.</li> <li>• Verify the form is generated correctly.</li> </ul>	OK
Administrator delete administrator page	<ul style="list-style-type: none"> <li>• Check the correctness of the displayed data.</li> <li>• Check the entire buttons and links working properly.</li> <li>• Verify all links and buttons working as expected.</li> </ul>	OK
Create new user page	<ul style="list-style-type: none"> <li>• Check the entire buttons and links working properly.</li> <li>• Check the entire selection list, buttons, and links working properly.</li> <li>• Check JavaScript function working properly.</li> <li>• Test the correctness of the validation for all input data.</li> <li>• Verify the page can get the error message and working properly by the message</li> </ul>	OK

## 5.2 Subsystem Testing

Subsystem testing is the next step up in the testing process where all related units from a subsystem does a certain task. Thus, the subsystem test process is useful for detecting interface errors and specific functions. Table 9 show subsystem test results in detail.

Table 9. Subsystem Test Results

Subsystem	Tests Performed	Results
Make appointment Subsystem	<ul style="list-style-type: none"> <li>• Make sure the time that the student chooses has not taken by other students.</li> <li>• Make sure students choose the time which is faculty's available time.</li> </ul>	OK
Check/ Cancel Subsystem	<ul style="list-style-type: none"> <li>• Make sure the subsystem list all the appointments.</li> <li>• Make sure all the columns are printed correctly.</li> <li>• Make sure after canceling the appointment, the time will be released for other students.</li> </ul>	OK
Approve Subsystem	<ul style="list-style-type: none"> <li>• Make sure each column is listed correctly.</li> <li>• Make sure students and faculties will receive the confirmation letter after appointments being made.</li> </ul>	OK
Create User Subsystem	<ul style="list-style-type: none"> <li>• Check if the subsystem can detect the error of creating of the user that exists in the subsystem.</li> <li>• Make sure the subsystem can automatically assign the substitute.</li> </ul>	OK
Modify User Subsystem	<ul style="list-style-type: none"> <li>• Check if users can update their personal information properly.</li> <li>• Verify if the subsystem shows the page properties correctly when the users are students, faculties or the administrator.</li> </ul>	OK
Delete User Subsystem	<ul style="list-style-type: none"> <li>• Verify the subsystem can delete a user account properly.</li> </ul>	OK
Security	<ul style="list-style-type: none"> <li>• Verify the page will be redirected to Login page when the user is not authenticated.</li> </ul>	OK

### 5.3 System Test Plan

System test plan is a test process that uses real data, which the system is intended to manipulate, to test the system. First of all the subsystem will be integrated into

one system. Then test the system by using a variety of data to see the overall results.

The steps for the system test plan are showed in the following table:

Table 10. System Test Results

System Test	Results
1. Install the Appointment System into server.	OK
2. Start up all services such as JSP engine, MySQL database engine.	OK
3. Running testing by using real data on all forms and reports.	OK

## CHAPTER SIX

### MAINTENANCE MANUAL

The Maintenance Manual provides maintenance personnel with the information necessary to maintain the system effectively. The manual provides the definition of the software support environment, the roles and responsibilities of maintenance personnel, and the regular activities essential to the support and maintenance of program modules, job streams, and database structures. In MTTOS, there are 3 major issues: Software Installation, Variable Installation, and AS Installation.

#### 6.1 Software Installation

In AS, it requires Gentoo, MySQL, JSDK, and TOMCAT to run the programs. Following will detail the installation of those five software systems.

##### 6.1.1 Gentoo Installation

Gentoo is a very good distribution for Linux. Following are the steps to install Gentoo Linux onto your machine.

1. Download the latest version of the Gentoo Universal LiveCD operating systems from <http://gentoo.osuosl.org/releases/x86/2004.3/livecd/> and burn the iso files into CDs.
2. Install the operating system by inserting CD into

the CD-ROM and make sure the bios is set to boot from CD.

3. The machine will startup via CD-ROM and start to install Gentoo.
4. Check the date and time, and correct if needed. Set the root password. Configure networking. Start sshd. Exit shell running on local console.
5. Setup the environment and install the components, then Gentoo Linux is installed.

#### 6.1.2 Install Java 2 Platform, Standard Edition

To install J2SDK in your machine under the Gentoo Linux System, you only have to type the command:

```
emerge jdk
```

Then the system will automatically finish the installation.

#### 6.1.3 Install Tomcat

Following are the steps to install Tomcat onto your machine.

1. Run `emerge tomcat`.
2. Run `rc-update add tomcat5 default`.
3. Run `chmod 755 /opt/tomcat5`
4. Restart Tomcat and type `http://localhost:8080` into the browser to test that your system works OK.



#### 6.1.4 MySQL Installation

The database system in Vacation System is MySQL. To install MySQL, follow the following steps:

1. Run emerge mysql.
2. Initialize the data directory and create the MySQL grant tables.

```
mysql_install_db --user=mysql
```

3. Add to default run-level, and start the server.

```
rc-update add mysql default
```

```
/etc/init.d/mysql start
```

4. Remove accounts without passwords.

```
mysql -u root -p
```

```
mysql> use mysql;
```

```
mysql> delete from user where password = '';
```

```
mysql> flush privileges;
```

#### 6.2 System Variables

In this system, we don't have to change any environment variables in the Linux system and server.xml in Tomcat server configuration directory.

The default configuration for the MySQL database is sufficient.

#### 6.3 Appointment System Installation/Migration

1. All the JSP programs and HTML programs are stored in  
/tomcat5/webapps/ROOT  
/tomcat5/webapps/as
2. All the \*.java are stored in  
/tomcat5/webapps/ROOT/WEB-INF/classes
3. All the classes are stored in  
/tomcat5/webapps/ROOT/WEB-INF/classes  
/tomcat5/webapps/vs/WEB-INF/classes  
/tomcat5/webapps/ers/WEB-INF/classes

#### 6.4 Backup and Restore

Protecting system information is one of the system administrator's most important tasks. Backups allow the administrator to restore a file system to the condition it was in at the time of the last backup. Backups must be done carefully and on a strict schedule. The backup system and backup media must also be tested regularly to verify that they are working correctly. There are two steps to back up Appointment System (AS). One is to backup the system files. The other step is to backup the database which is used by AS.

##### 6.4.1 System Backup

All the system files are stored in the directory  
"/tomcat5/webapps/ROOT," "/opt/tomcat5/webapps/as," and the

subdirectory of its. Thus, in order to backup the system files, we can compress this directory by using the compress program "tar" to backup the system files:

```
tar -cf root.tar /tomcat5/webapps/ROOT
```

```
tar -cf vs.tar /tomcat5/webapps/as
```

#### 6.4.2 Database Backup

To backup the database system, we use mysqldump command to backup all database used by the system. The following command is used to backup the database:

1. Create a subdirectory named backup under the directory /var/lib/mysql/appointment

2. Using command:

```
mysqldump --tab=/var/lib/mysql/appointment/backup  
--opt appointment
```

After executing the backup command above, we will have 14 files: studentinfo.sql, studentinfo.txt, facultyinfo.sql, studentinfo.txt, administratorinfo.sql, administratorinfo.txt, checklogin.sql, checklogin.txt, availabletime.sql, availabletime.txt, app\_taken.sql, app\_taken.txt, current\_course.sql, current\_course.txt.

#### 6.4.3 System Restore

To restore the system file, simply extract the backup file by using the following command:

```
tar -xzvf root.tar /
```

```
tar -xzvf as.tar /
```

#### 6.4.4 Database Restore

To restore the database needed for the system, go to the directory `/var/lib/mysql/appointment/backup`, and execute the following commands:

1. `create database appointment;`
2. `restore table users from`  
`/var/lib/mysql/appointment/backup`

## CHAPTER SEVEN

### CONCLUSION AND FUTURE DIRECTIONS

#### 7.1 Conclusion

All the functions are described as follows: 1) Students can make appointments with faculty in this system. 2) Students can make appointments online without sending several emails, which is more efficient. 3) Users get confirmation letter through the email submitted by server. 4) Students can log on and get the information of faculty.

By using the internet to make appointment, students and faculty can save so much time. Students do not have to write email to faculty to ask the available time and wait for the reply. Faculty does not have to check their schedule and write emails to students. Students can log on the Appointment System to check the available time and make appointments anytime.

#### 7.2 Future Directions

The next step is to make this application as part of the CSCI Department website. This implies analyzing the connect site and integrate the new function with it.

## REFERENCES

- [1] "Beginning JSP Web Development." Jayson Falkner. August 2001.
- [2] IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1993).
- [3] "JavaScript: The Definitive Guide 4th edition." David Flanagan. O'Reilly & Associates, December 2001.
- [4] "JavaServer Pages." Larne Pekowsky. April 2000.
- [5] "Java Servlet Programming Second Edition." Jason Hunter and William Crawford. O'Reilly and Associates, 2002.
- [6] "JavaSpaces Principles, Patterns, and Practice." Eric Freeman. November 1999.
- [7] SSL 3.0 Specification.  
<http://wp.netscape.com/eng/ssl3/>, November 1996.
- [8] "The Java Programming Language Second Edition." Ken Arnold and James Gosling. February 2000.
- [9] "UML Distilled, A Brief Guide to Standard Object Modeling Language", Fowler & Scott, 1999.
- [10] "A Web Services Primer." Venu Vasudevan,  
<http://webservices.xml.com/pub/a/ws/2001/04/04/webservices/index.html>, April 2001.
- [11] "What is Service-Oriented Architecture?" Hao He,  
<http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html>, September 2003.
- [12] "JAVA and XML 2nd Edition" Brett McLaughlin. O'Reilly & Associates, January 2002.
- [13] "XML Web Services Basics" Roger Wolter. Microsoft Corporation, December 2001.