

KEMI-TORNIO UNIVERSITY OF APPLIED SCIENCES

E-Attendant List

E-ATTENDANT LIST

Jiang Nan

Thesis of Information Technology Degree Program

Bachelor of Engineering

KEMI 2011

## **PREFACE**

Firstly, I would like to thank my instructor Mr. Thai Bui. I got a lot of help from him, included many encourages in the technical parts.

Then, I have to say thank you to my parents. Without their support, I could not have the chance to go abroad and study in Finland.

Lastly, I should thank all my teachers who taught me a lot at Kemi-Tornio University of Applied Sciences. In addition, I thank the many good friends I met here.

## ABSTRACT

|  |                        |
|--|------------------------|
| Kemi-Tornio University of Applied Sciences, Technology |                        |
| Degree Programme                                       | Information Technology |
| Name   | Jiang Nan              |
| Title  | E-Attendant List       |
| Type of Study  | Bachelor's Thesis      |
| Date   | 21 September 2011      |
| Pages  | 38                     |
| Instructor   | Thai Bui               |

The attendant work has become important nowadays, not only in schools, but also in companies. The original attendant work is based on paper work. That seems inefficient and waste of lots of resources in real life. To solve the problems in this phase, this project had been started. The whole system is based on the website. It makes the system work in any operating system. The request of using the system is the internet connection.

The tasks of this project included the design of the framework of this system, the database schema, the sequences of the usages, and user interface. The environment of this system is built on the cooperation of PHP and MySQL. Because this environment is one of the most popular website environments in the world, the greatest advantage of this environment is the high security.

The final product is called "E-Attendant". Through the system, the student can choose the course and sign the name on the attendant list online; teachers can admin the attendance of students, and view the attendant result; course manager can control the whole system, including admin the user (student and teacher), admin the courses, and admin the attendant.

Keywords: PHP, MySQL, Attendant List.

## TABLE OF CONTENTS

|                                      |    |
|--------------------------------------|----|
| 1. INTRODUCTION .....                | 1  |
| 2. TECHNICAL BACKGROUND .....        | 2  |
| 2.1. UML.....                        | 2  |
| 2.2. ER model & Database schema..... | 3  |
| 3. SYSTEM ANALYSIS AND DESIGN .....  | 5  |
| 3.1. Use case diagram.....           | 5  |
| 3.2. ER model.....                   | 6  |
| 3.3. Database diagram .....          | 10 |
| 3.4. Sequence diagram .....          | 14 |
| 3.5. User interface .....            | 22 |
| 3.5.1. Login page.....               | 22 |
| 3.5.2. Student .....                 | 23 |
| 3.5.3. Teacher.....                  | 26 |
| 3.5.4. Manager.....                  | 32 |
| 4. RESULTS .....                     | 39 |
| 5. CONCLUSIONS.....                  | 40 |
| 6. REFERENCES.....                   | 41 |

## **EXPLANATION OF CHARCTERS AND ABBREVIATIONS**

|       |                                       |
|-------|---------------------------------------|
| PHP   | Hypertext Pre-processor               |
| MySQL | Relational database management system |
| UML   | Unified Modelling Language            |
| ER    | Entity-Relationship                   |

## 1. INTRODUCTION

This system is used by the student, teacher and course manager. The entire users can login from the “welcome page”. The system can resolve the domain of the users. The students can choose the course and sign their name in the attendance list. The date and the time of signing would be saved in the database. The teachers can see the entire attendance situation of his or her courses. The course manager/administrator can add/Delete/Edit the course in the database. The system can count the attendance of students state and report to the manager and the teacher automatically.

The system can be used in both normal class courses and online courses.

- Student: Login, Choose the course, Sing in, Log out
- Teacher: Login, View the attendance list, Logout
- Course Manager/Administrator: Login, View result of the attendance list, Add course, Edit course, Delete course.
- All operations connected to the database.

The functions of attendance list would be designed through UML. It means that the use case diagram, activity diagram, database diagram, and system architecture diagram have to be included in this step. PHP language would be chosen to do the implementing work following. In additional, MySQL technology would be used for database.

In this project, the primary objective would complete the UML design, the basic functions of the system as design needed implemented simultaneously.

Secondly, the attendance of students can be counted by the system automatically. The results will be displayed to teachers and managers.

## 2. TECHNICAL BACKGROUND

### 2.1. UML

UML is short for Unified Modelling Language. It is used for software-intensive systems for visual modelling language. UML object-oriented development system is for the product description, visualization and preparation of documents as a standard language. UML is a non-proprietary third-generation modelling and Specification Language. UML is an open system approach used in the development phase, description, visualization, and writing to build an object-oriented software-intensive product. UML shows a series of best engineering practice. The best practices are displayed in large, complex system modelling. Especially in the software architecture level has been validated. UML is sure in data modelling, business modelling, object modelling and component modelling. /1/ /8/

Firstly, it is necessary to understand the diagram types of UML. In addition, the advantages of them should be understand. /1/

- **Class diagrams**

Class diagram is a static diagram which describes the static structure of a system. It is also known as a structural diagram. /9/ /7/

- **Package diagrams**

Package diagrams are used to organize model elements into group. It could make the complex UML diagram and the relationships to simple and clear diagram. The developers can recognize the relationship and framework of the product easily. /9/ /7/

- **Object diagrams**

Object diagrams are used to instruct a class in a pointed time. It can help clarify classes and inheritance. It seems to be the sub-diagram of class diagram. /9/ /7/

- **Use Case diagrams**

Use Case diagrams are used to model the entire system. The use case diagram includes the use case and the actors. All use cases are the activities of the actor had in the system. /9/ /7/

- **Sequence diagrams**

Sequence diagrams are used to express the sequence of message transfer between object. One sequence diagram is describing the process of one use case task. They are also used to identify and enrich the logic of a context of use. /9/ /7/

- **Collaboration diagrams**

Collaboration diagram is an interaction diagram. It emphasizes the objects of sending and receiving message between the organizational structures. Collaboration diagram shows a series of objects and the links between these objects and objects send and receive messages. /9/ /7/

- **Statechart diagrams**

Statechart diagrams describe different states of a component in a system. A component or object of a system is specified by states. /9/ /7/

- **Activity diagrams**

Activity diagram describe dynamic aspects of the system. Activity diagrams deals with all types of flow control by using different elements. /9/ /7/

- **Component diagrams**

Component diagrams are used to design the main physical or software framework of the product. It describes the access of the physical model and the software model package, including the framework and data transport process. /9/ /7/

- **Deployment diagrams**

Deployment diagrams are used to describe the static deployment view of a system. They are also used for describing the hardware components. /9/ /7/

UML becomes an important part of software design work and system design work. In other words, the design work will be easier and more logical with the help of UML. It is the mainstream of development work nowadays. UML is suitable for a variety of problem domains, even them in the construction of various sizes and complexity of the system. In addition, most programming languages, operating systems and tools, in a certain way, are object-oriented currently. UML provides a conceptual basis of object-oriented development. /1/ /2/ /9/

## 2.2. ER model & Database schema

ER is short for Entity-Relationship. It provides an entity type, properties and relationships. In additional, it is used to describe the conceptual model of the real world. Database developer uses this way to define the database structures. /4/ /10/

ER modeling is constituted of entities, attributes and links. The three elements are the basic factors which need. The details representation is as below. Rectangles are used for entities. The rectangular box indicates the entity name. Ovals are used for attributes. The



rectangular boxes are linked to the corresponding entity by undirected line. Diamond is used for relationships. The contact name is written in diamond-shaped box. The box is linked to the relevant entities by undirected line. At the same time, the linking type (1: 1, 1: n or m: n) should be mentioned next to the undirected line. Many times notations are used solely by the database team and limit the ER modeling to relational database design. /4/ /5/

One of the widely accepted languages named UML which is used to analysis and software development work. It is fit for the graphic representation of ER diagrams. The significant benefits of ER diagrams are showed as following words. The communication between team members becomes easier. Because of models, to repositories goes easier from integration. Simultaneously, from analysis to implementation is unified. /4/ /5/ /6/

A database schema of a database system is a structure table, it describes in a formal language. It refers to the organization of data to create a blueprint. It shows how a database is constructed. The database language can be realized by a database. It is showed in its logic structure. /3/ /6/

The database schema creating should be based on the ER model. In the steps of design database, first of all, the objects are declared which will be included in the system. After that, to analyze the relationship between the objects: 1 to m, 1 to n, or m to n. All relationships and objects should be displayed in the database. /4/

### 3. SYSTEM ANALYSIS AND DESIGN

#### 3.1. Use case diagram

For the function of this system, there are three kinds of users. They are Student, Teacher, and Manager. The use case is designed by their actions. All operations save the data in the database in the server.

- **Student**

Before any operations, the student must log in to this system. Student can sign his or her name in the attendant list; view the attendance history in the order of course name, and log out to exit the system. Student can edit their accounts and their personal information.

- **Teacher**

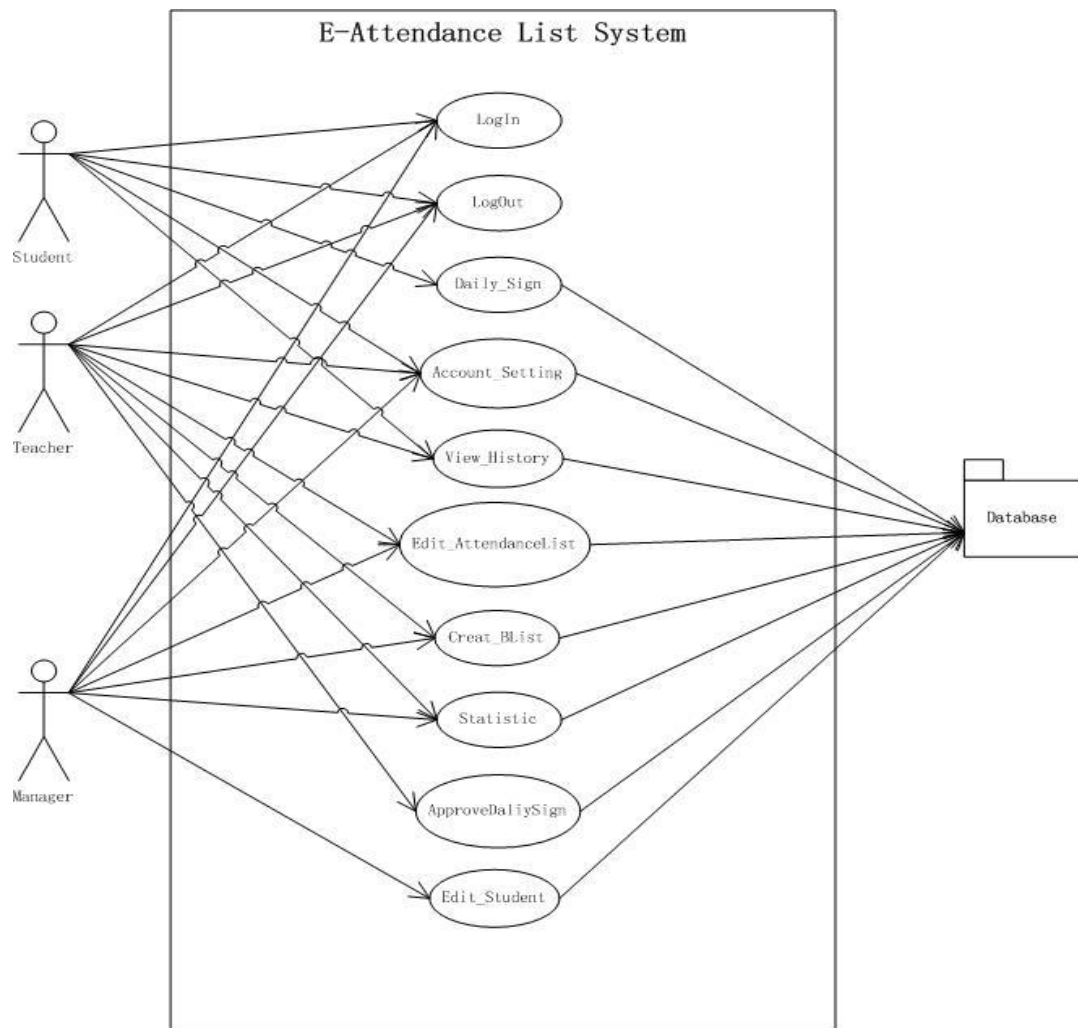
Before any operations, the teacher should log in to this system. Teacher can view the attendance history in the order of the course which the teacher teaches, open or close for the student (Approve Daily Sign), create a blank list for the new course, edit attendance list for the course, and statistics the attendance history by student. Teacher can edit their accounts and their personal information.

- **Manager**

Before any operations, the manager should log in to this system. Manager can edit all attendance lists in the system, create blank list for the new course, statistic the attendance history for all students, and edit (Create new, Edit, Delete) the student information. Manager can edit their accounts and their personal information.

- **Database**

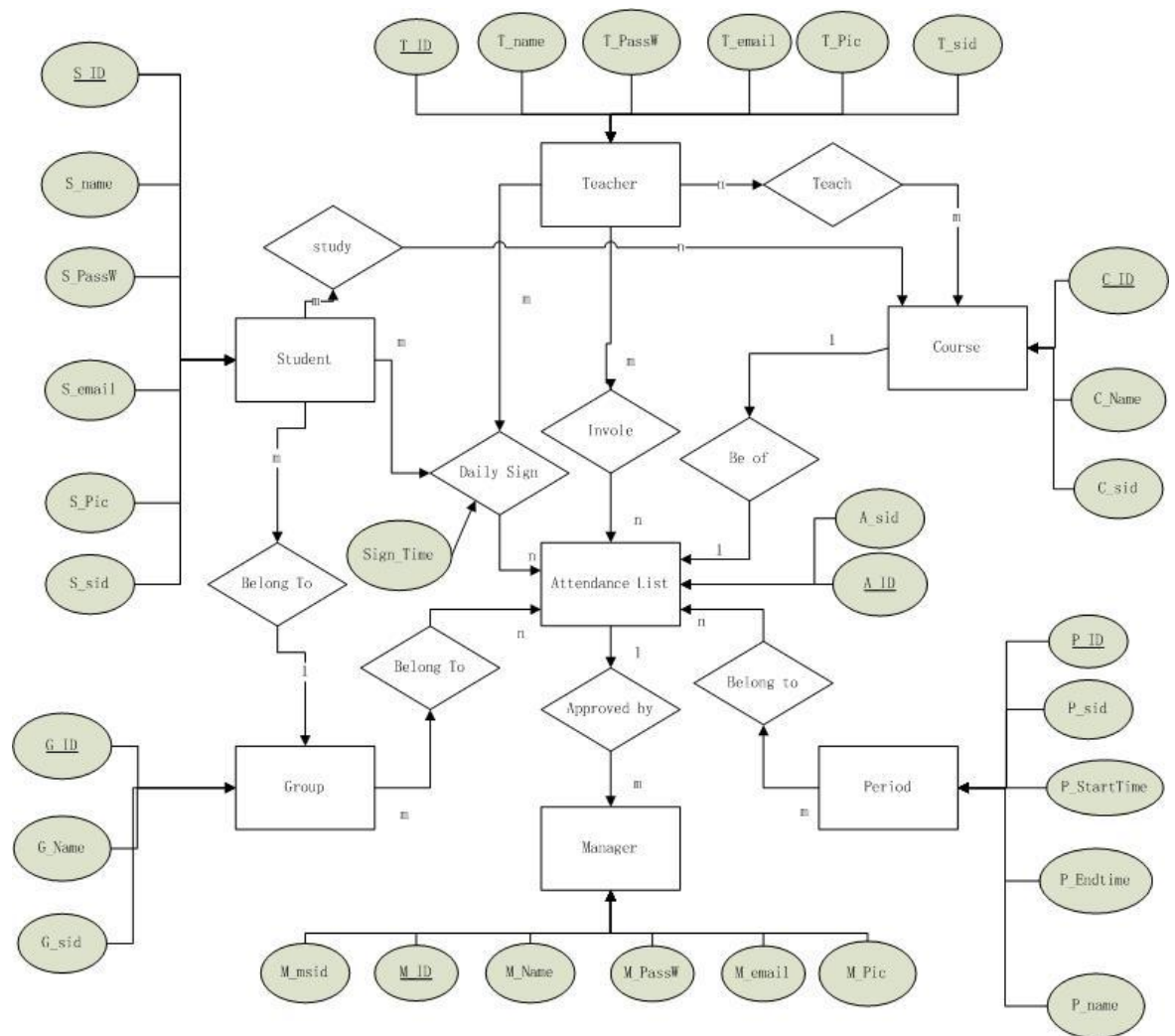
All the operations send the data query to the database. The database receives the query and processes the data, after that sends the queried data back to the server and displays to the users. The processes of the database include searching, adding, deleting, and updating to the tables.



**Fig.1. Use case diagram**

### 3.2. ER model

The objects in the system include Teacher, Student, Course, Attendance list, Manager, Group, and Period. The relationships of these objects and the properties have been shown in the picture below.



**Fig.2. ER model diagram**

- **Student:** the attributions of the students include s\_id, s\_name, s\_passW, s\_email, s\_Pic, s\_sid.  
**s\_id:** it is used to save the student number which given when the student start study in the university (example 0700679).  
**S\_name:** it is used to save the real full name of the student.  
**S\_passW:** it is used to save the password of the student.  
**S\_email:** it is used to save the email address of the student.  
**S\_Pic:** it is used to save the display image of the student.  
**S\_sid:** it is a mark for each student, for the design, it will save the group name for the student in characters (example: IT07, BIT08).

One student can belong to one group, but one group can contain several students. So the relationship between the student entity and group entity is “belong to” and the cardinality is “m to 1”.

Students can study one course, and one course can contain several students. So the relationship between the student entity and course entity is “study” and the cardinality is “m to n”.

For the design, one attendance list is used to save the attendance information for one course. So one student can sign in several attendant lists, and one attendant list can contain several students. The relationship between student entity and attendant list entity is signing and adds the sign time into the daily sign list to save the real time when the student signs in the daily sign list.

- **Teacher:** the attributions of teacher include t\_id, t\_name, t\_PassW, t\_email, t\_pic, t\_sid.
  - t\_id:** it is used to save the teacher number which given when the teacher start work in the university.
  - t\_name:** it is used to save the real full name of the teacher.
  - t\_passW:** it is used to save the password of the teacher.
  - t\_email:** it is used to save the email address of the teacher.
  - t\_Pic:** it is used to save the display image of the teacher.
  - t\_sid:** it is a mark for each teacher, for the design, it will save the group name for the teacher in characters (example IT, BIT).

One teacher can teach several courses, and one course can be taught by several teachers. So the relationship between teacher entity and course entity is “teach” and the cardinality is “m to n”.

For the relationship between course and attendant list, one teacher can involve several courses which the teacher teaches, and all teachers can be involved one course who teaches the course. So the relationship between teacher entity and attendant entity is “involve”, and the cardinality is m to n. The teacher can control the daily sign open or close for the course which the teacher teaches, so the relationship between teacher entity and attendant list entity include “daily sign”, and the cardinality is “m to n”.

- **Manager:** the attributions of manager are including m\_id, m\_name, m\_passW, m\_email, m\_Pic, m\_sid.
  - m\_id:** it is used to save the manager number which given when the manager start working in the university.
  - m\_name:** it is used to save the real full name of the manager.
  - m\_passW:** it is used to save the password of the manager.
  - m\_email:** it is used to save the email address of the manager.
  - m\_Pic:** it is used to save the display image of the manager.
  - m\_sid:** it is a mark for each manager, for the design, it will save the group name for the manager in characters (example IT, BIT).

All managers can approve one attendant list; one attendant list can be approved by all managers. The relationship between manager entity and attendant list entity is “approved by”, and the cardinality is “m to 1”.

- **Group:** the attributions of group are including g\_id, g\_name.  
**G\_id:** it is used to save the group id which is given when the group created.  
**G\_name:** it is used to save the group's full name include the years number.  
**G\_sid:** it is used to save the reorganization information for each group. For the design the main group title is saved as character type (example IT, BIT).

For the relationship between course and attendant list, one group can belong to several attendant lists, and one attendant list can contain several group students. The relationship between group entity and attendant list entity is “belong to”, and the cardinality is “m to n”.

- **Course:** the attributions of course entity are including c\_id, c\_name, c\_sid.  
**C\_id:** it is used to save the course number which given when the course created.  
**C\_name:** it is used to save the name of the course.  
**C\_sid:** it is used to save the recognize information for each course, for the design, the group name which the course belonged is saved as character type (example IT07, BIT08).

One course can belong to one attendant list, and one attendant list can be used for one course. The relationship between course entity and attendant entity is “belong to”, the cardinality is “1 to 1”.

- **Period:** the attributions of period entity are including p\_id, p\_startTime, p\_endTime, p\_name.  
**P\_id:** it is used to save the period number.  
**P\_startTime:** it is used to save the starting time of the study period (example 1/9/2011).  
**P\_endTime:** it is used to save the ending time of the study period (example 20/10/2011).  
**P\_name:** it is used to save the name of the study period; the name is including the year's number and the period numerical (example P1/2011).

In one study period, many courses are held. The time of one course can contain not only one study period. So the relationship between period entity and attendant entity is “belong to”, the cardinality is “m to n”.

- **Attendant list:** the attributions of the attendant list are including a\_sid, a\_id.  
**A\_sid:** it is used to save the recognize information for each attendant list (example: IT0701).

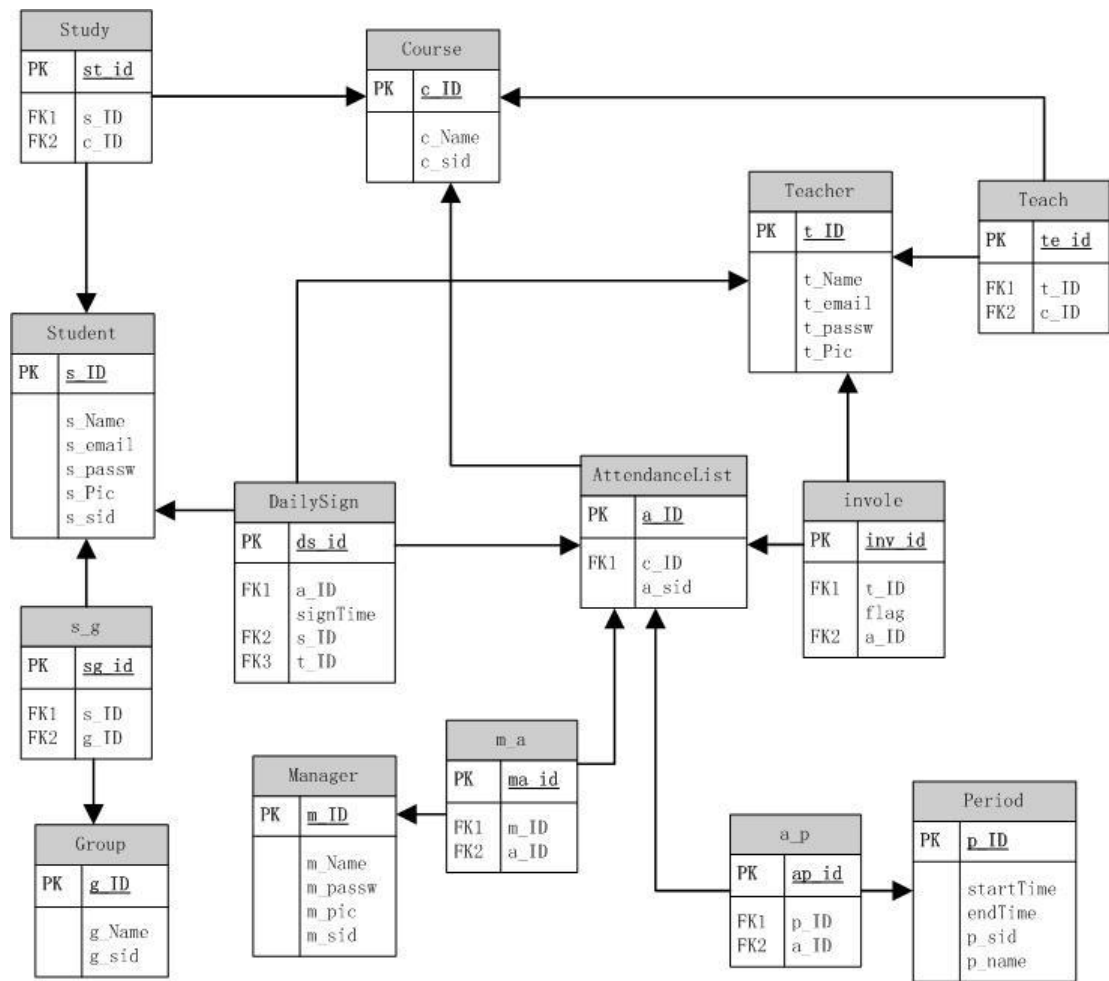
**A\_id:** it is used to save the attendant list number (example: 01).

### 3.3. Database diagram

For the design of the Entity Relationship diagram, all the entities and relations are needed to be contained in the database schema. The “m to n” relationship needs an individual table to describe. The “id” for each entity is the main reorganization flag for the data. For this reason, the “id” must be the primary key for each table.

If there are some relationships in tables, the foreign key is needed. The main task of the foreign key is to connect two tables in the database. For example, the student table and the group table, the relationship between them is “belong to”. From the ER diagram design, the cardinality is “m to 1”. That means, one student should belong to one group and one group contains several students. For this reason, a table is needed to connect these two tables; the name is “s\_g”. In this table, the “s\_id”, and “g\_id” are foreign keys, they are the primary keys in the tables of students and groups. That makes the data transfer possible.

The database schema is designed as below.



**Fig.3. Database Diagram**

The data types for each table are described in following forms.

**Table 1. Student's table data**

| Items   | Type      |    |
|---------|-----------|----|
| s_id    | INT (20)  | PK |
| s_name  | CHAR (50) |    |
| s_passw | CHAR (20) |    |
| s_pic   | CHAR (50) |    |
| s_email | CHAR (50) |    |
| s_sid   | CHAR (20) |    |

**Table 2. Teacher's table data**

| Items | Type     |    |
|-------|----------|----|
| t_id  | INT (20) | PK |



|         |           |  |
|---------|-----------|--|
| t_name  | CHAR (50) |  |
| t_passw | CHAR (20) |  |
| t_pic   | CHAR (50) |  |
| t_email | CHAR (50) |  |
| t_sid   | CHAR (20) |  |

**Table 3. Manager's table data**

|         |           |    |
|---------|-----------|----|
| Items   | Type      |    |
| m_id    | INT (20)  | PK |
| m_name  | CHAR (50) |    |
| m_passw | CHAR (20) |    |
| m_pic   | CHAR (50) |    |
| m_email | CHAR (50) |    |
| m_sid   | CHAR (20) |    |

**Table 4. Course's table data**

|        |           |    |
|--------|-----------|----|
| Items  | Type      |    |
| c_id   | INT(10)   | PK |
| c_sid  | CHAR (20) |    |
| c_name | CHAR(50)  |    |

**Table 5. Attendance-list table data**

|       |           |    |
|-------|-----------|----|
| Items | Type      |    |
| a_id  | INT(10)   | PK |
| a_sid | CHAR (20) |    |
| c_id  | INT(10)   | FK |

**Table 6. Daily-Sign table data**

|          |         |    |
|----------|---------|----|
| Items    | Type    |    |
| Ds_id    | INT(20) | PK |
| a_id     | INT(10) | FK |
| s_id     | INT(20) | FK |
| t_id     | INT(20) | FK |
| signTime | DATE    |    |

**Table 7.Group's table data**

| Items  | Type      |    |
|--------|-----------|----|
| g_id   | INT(20)   | PK |
| g_name | INT(20)   |    |
| g_sid  | CHAR (20) |    |

**Table 8.Study table data**

| Items | Type    |    |
|-------|---------|----|
| St_id | INT(20) | PK |
| s_id  | INT(20) |    |
| c_id  | INT(20) | FK |

**Table 9.Teach table data**

| Items | Type    |    |
|-------|---------|----|
| Te_id | INT(20) | PK |
| t_id  | INT(20) | FK |
| c_id  | INT(20) | FK |

**Table 10.Involve table data**

| Items  | Type    |    |
|--------|---------|----|
| Inv_id | INT(20) | PK |
| t_id   | INT(20) | FK |
| Flag   | INT(10) |    |
| a_id   | INT(10) | FK |

**Table 11.Period table data**

| Items       | Type     |    |
|-------------|----------|----|
| p_id        | INT(20)  | PK |
| P_startTime | DATE     |    |
| P_endTime   | DATE     |    |
| P_name      | CHAR(20) |    |
| P_sid       | INT(20)  |    |

**Table 12.M\_a table data**

| Items | Type    |    |
|-------|---------|----|
| Ma_id | INT(20) | PK |
| M_id  | INT(20) | FK |
| A_id  | INT(20) | FK |

**Table 13.A\_p table data**

| Items | Type    |    |
|-------|---------|----|
| pa_id | INT(20) | PK |
| a_id  | INT(20) | FK |
| p_id  | INT(20) | FK |

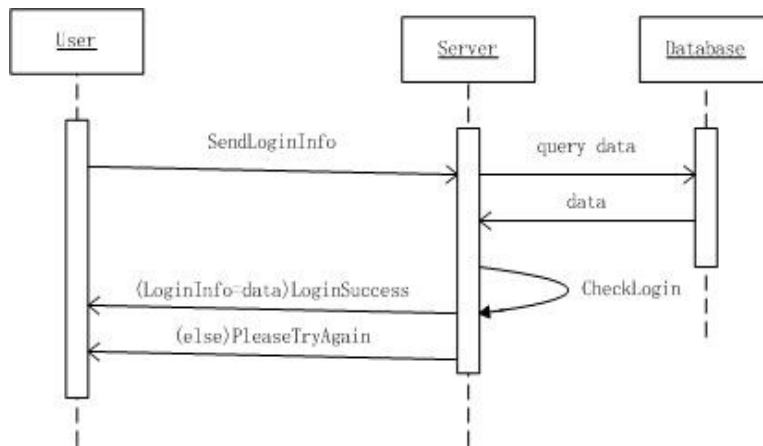
**Table 14.S\_g table data**

| Items | Type    |    |
|-------|---------|----|
| sg_id | INT(20) | PK |
| s_id  | INT(20) | FK |
| g_id  | INT(20) | FK |

### 3.4. Sequence diagram

#### Sequence diagram: Log in

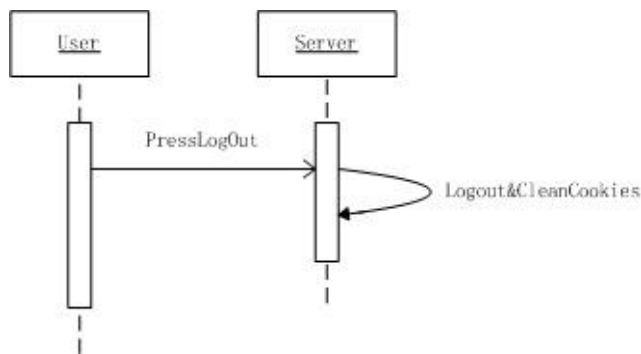
This use-case used for all users login. All actions for the user can be acted after login. User inputs the login information; includes user id, password, and user type. The server checks the data through the database, and returns the login result to the user.



**Fig.4. Login Sequence Diagram**

**Sequence diagram: Log out**

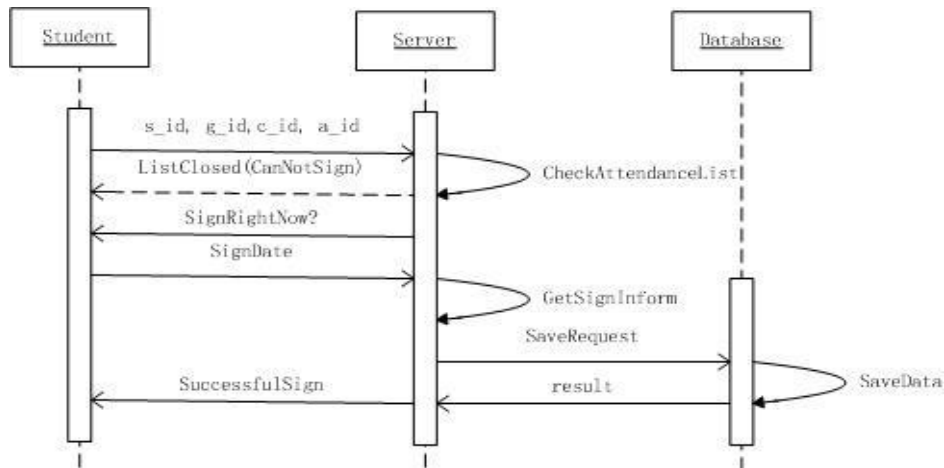
After user login, when the logout button is pressed, the server breaks the cookie and lets the user log out of the system.



**Fig.5. Logout Sequence Diagram**

**Sequence diagram: Daily Sign**

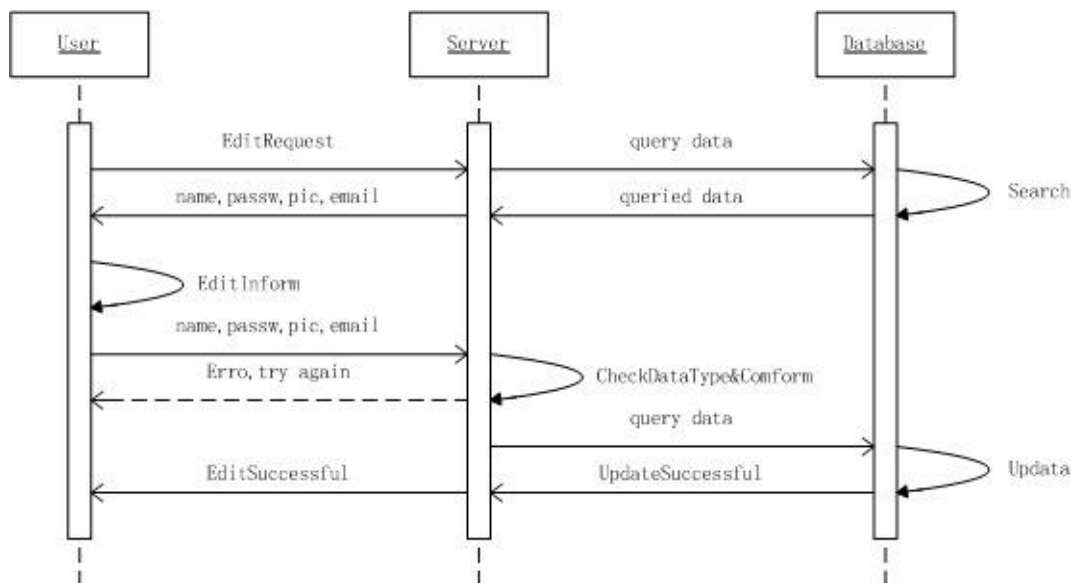
This sequence diagram of use case is used for daily sign of the student. The server checks the course daily sign open or not. Before this sequence diagram of use case, the student has to log in as student successfully.



**Fig.6. Daily Sign Sequence Diagram**

**Sequence diagram: Account setting**

This sequence diagram of use case is for all users (student, teacher, manager). The user has to log in into the system.

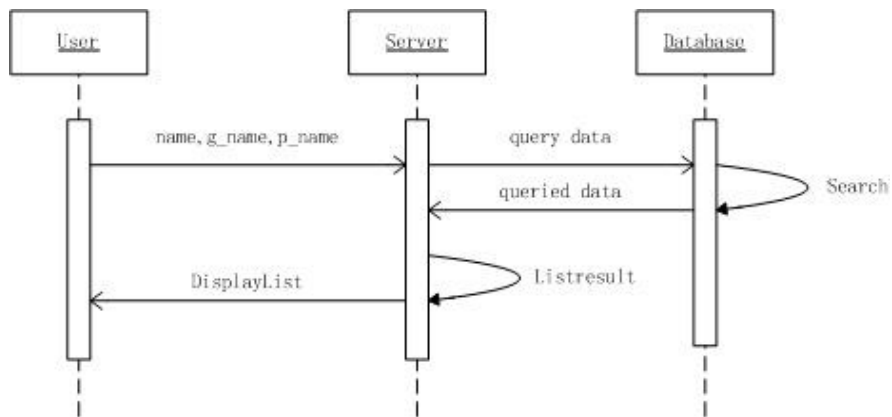


**Fig.7. Account setting Sequence Diagram**

### Sequence diagram: View History

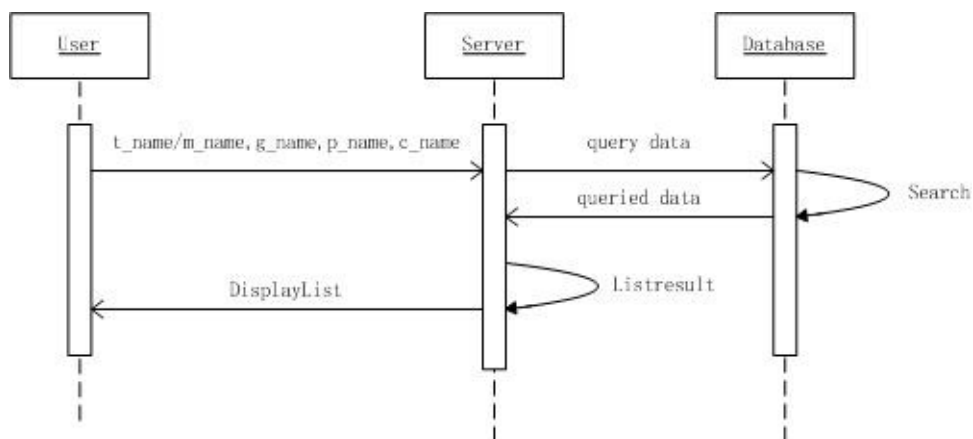
This sequence diagram of use case is for all users (student, teacher, manager). The user has to log in into the system. The difference is that the student views his or her attendance history. On the other hand, the teacher can view all the students in the course, and the manager can view all courses and all students.

For students:



**Fig.8. View history Sequence Diagram**

For teachers and managers:



**Fig.9. View history Sequence Diagram**

### Sequence diagram: Statistics

This sequence diagram of use case is for teacher and manager. The difference is that the teacher counts the course attendance; the manager can count all courses. Before this action, the user has to log in into the system.

For manager:

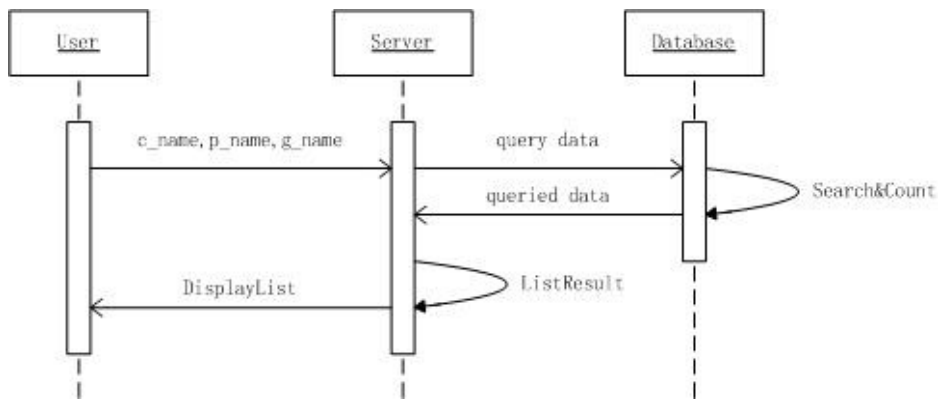


Fig.10. Statistics Sequence Diagram

For teacher:

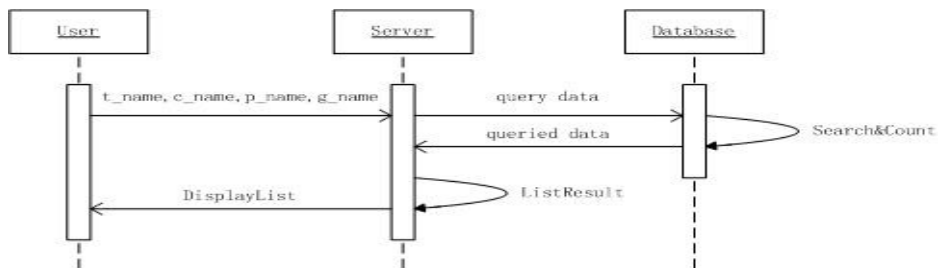
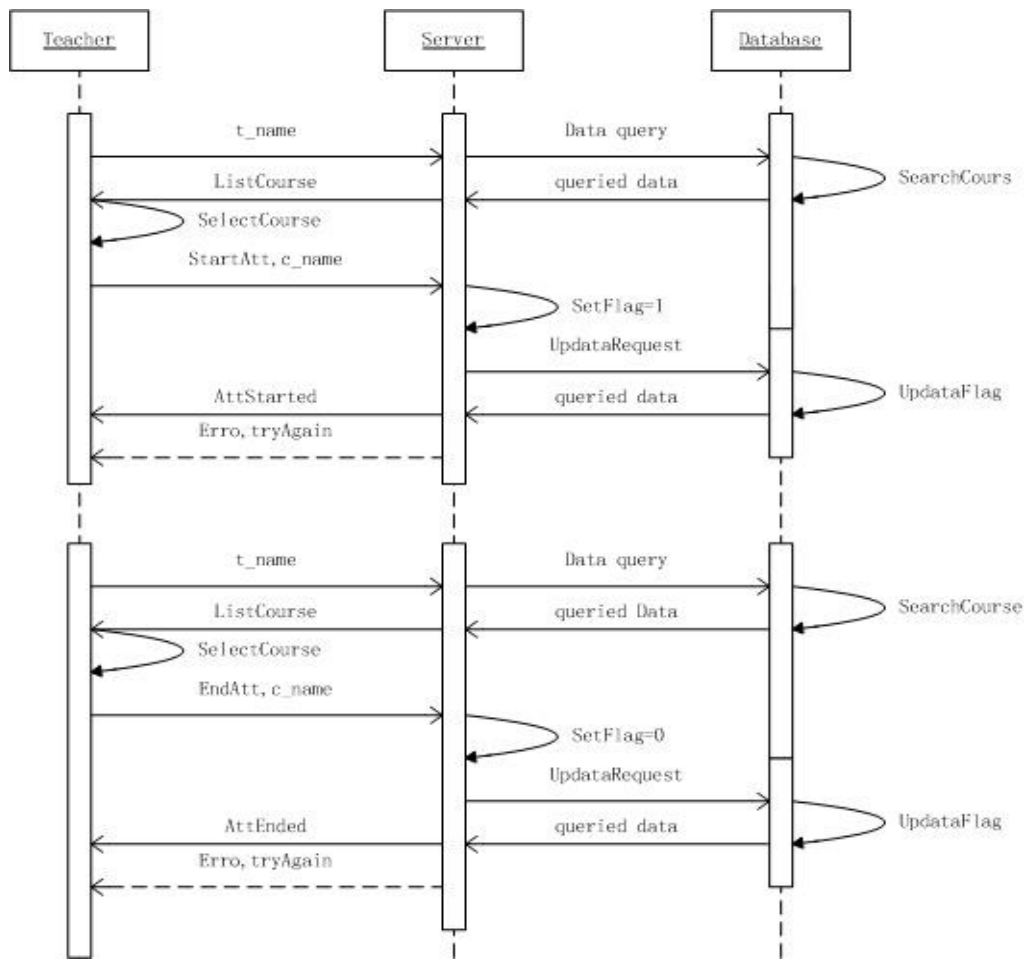


Fig.11. Statistics Sequence Diagram

### Sequence diagram: Approve daily sign

This sequence diagram of use case is for the teacher. The teacher controls the attendance list open and close to select course through this use case. The user has to log in into this system as teacher's user type.

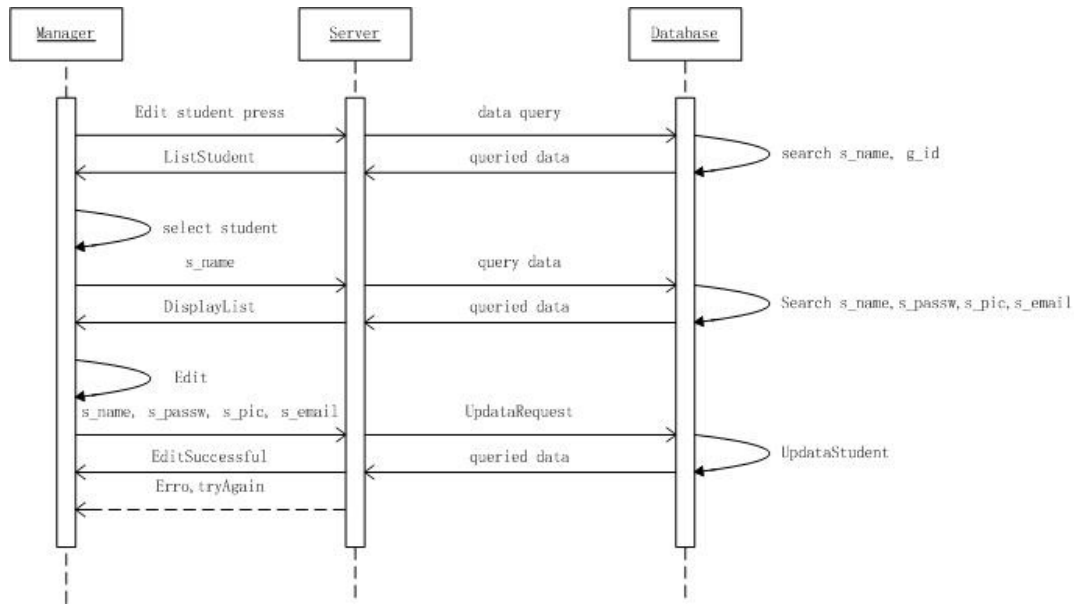


**Fig.12. Approve Daily Sign Sequence Diagram**

**Sequence diagram: Edit Student**

The sequence diagram of use case is for the manager only. This use case is used to create the new student into the database. The manager should input the basic information to create the new student, and the student can edit his or her personal information later. The user must log in into this system as manager's user type before this action.

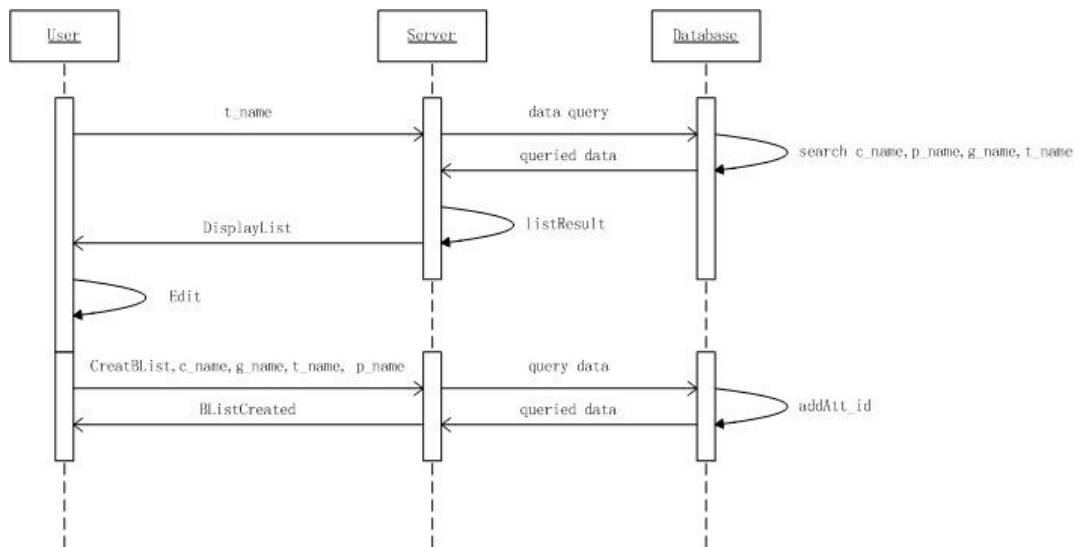




**Fig.13. Edit Student Sequence diagram**

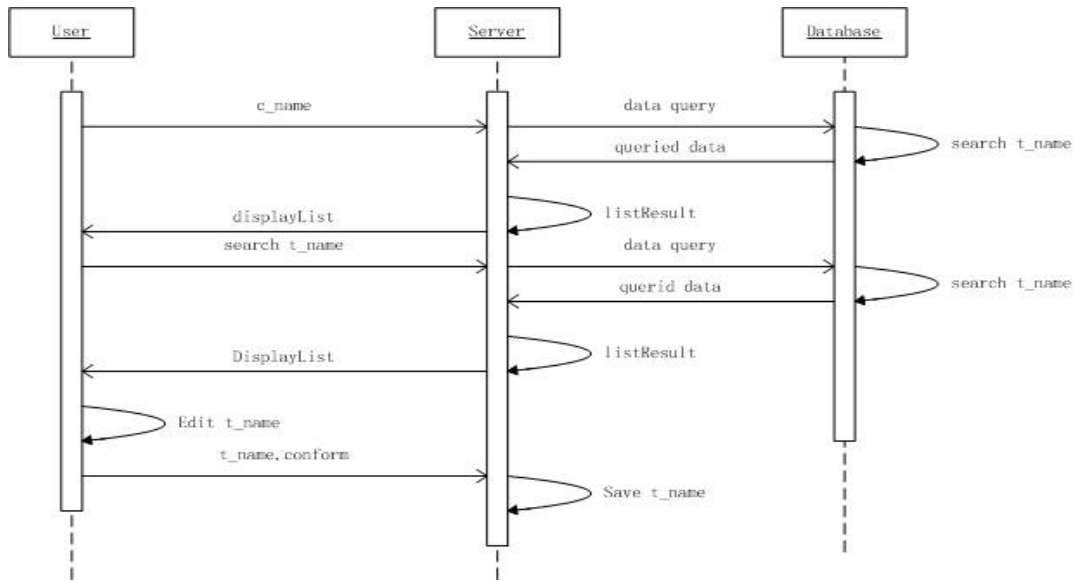
**Sequence diagram: Create Blank List**

This sequence diagram of use case is for Manager and Teacher to create the blank attendant list of new courses. The user must input the default attendance information; include course name, group, and period. Before this action, the user has to log in into the system as manager's user type.



**Fig.14. Create Blank List Sequence Diagram**

Add teacher:

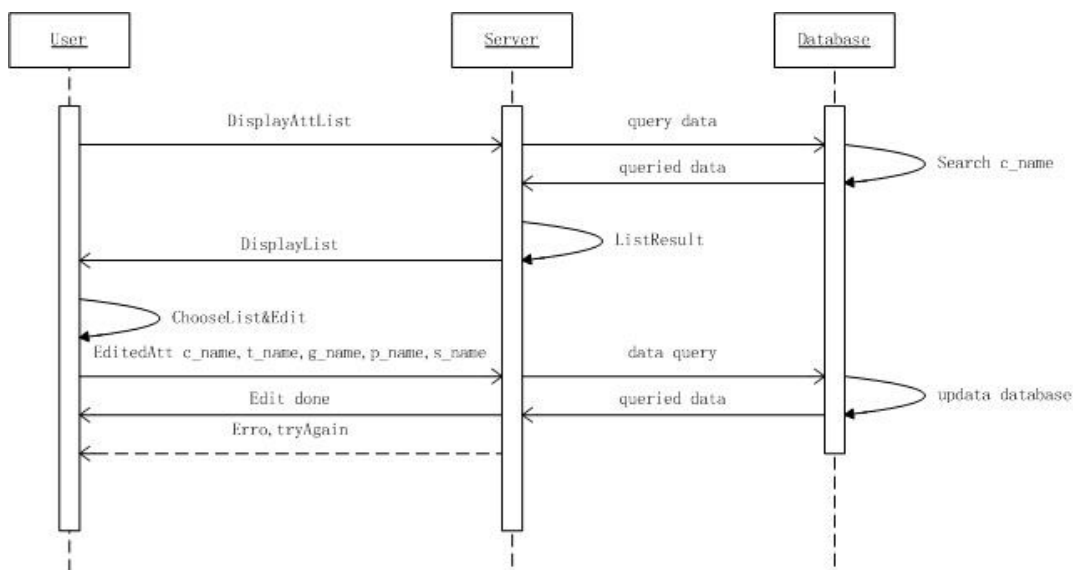


**Fig.15. Add teacher Sequence Diagram**

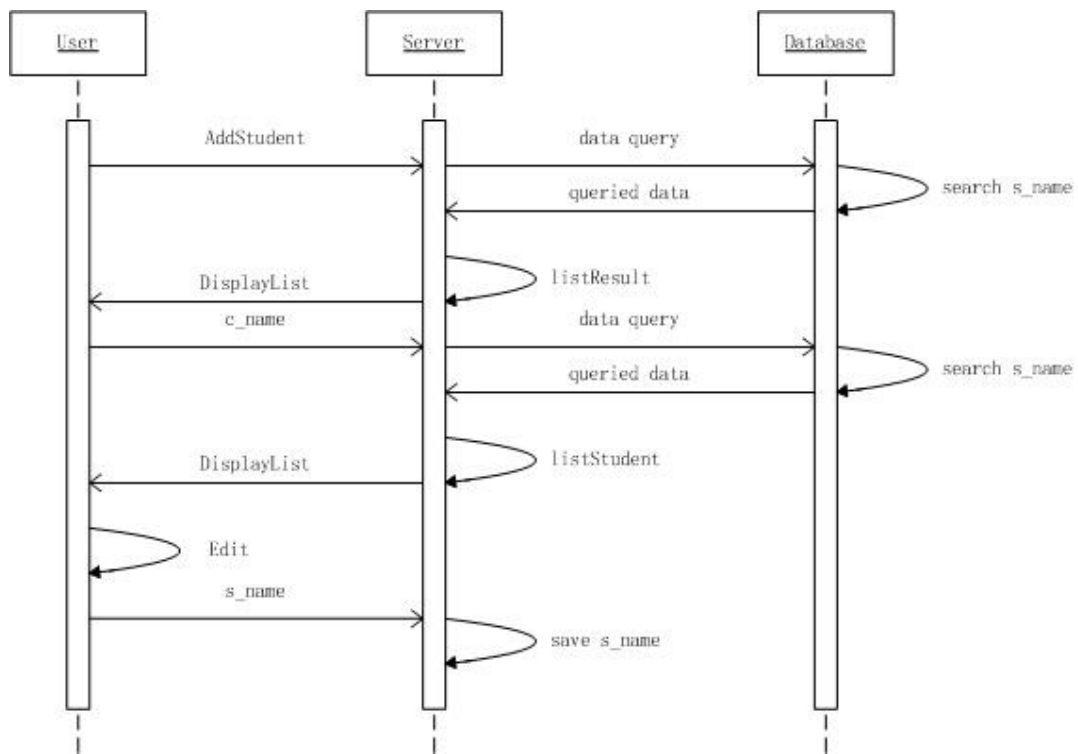
The teacher's name after edit is saved in the stuck/buffer of the server, when the user presses conformation, all data is sent to the server and saved in the database.

**Sequence diagram: edit attendance list**

This sequence diagram of use case is for teacher and manager, and edits the attendance information. The user must login in into this system before this action.



**Fig.16. Edit Attendance Sequence Diagram**



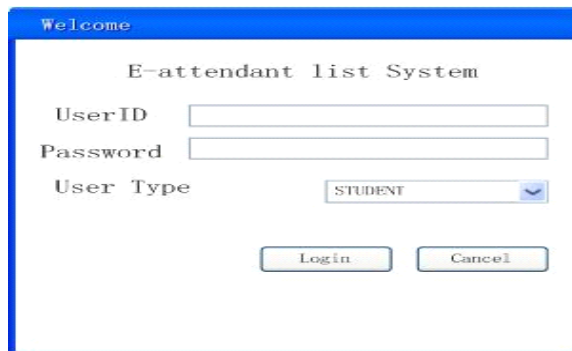
**Fig.17. Edit Attendance List Sequence Diagram**

The student's name after editing is saved in the stuck/buffer in the server. When the user finishes editing all information, all data is sent to the server and saved in the database.

### 3.5. User interface

#### 3.5.1. Login page

Before using this system, the user must login in into the system through this page. The type of the users include: Student, Teacher and Manager. User ID and Password is checked through the database.



The screenshot shows a web browser window titled "Welcome". The main heading is "E-attendant list System". Below the heading, there are three input fields: "UserID" with a text box, "Password" with a text box, and "User Type" with a dropdown menu showing "STUDENT". At the bottom, there are two buttons: "Login" and "Cancel".

**Fig.18. Login Site**

After the user has logged into this system, the system turns to the user own main page automatically.

### 3.5.2. Student

After student login in into this system, the main page of student is displayed to the user.



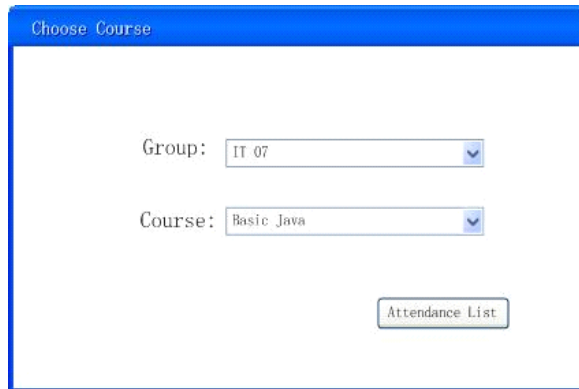
The screenshot shows a web browser window titled "Student". The main heading is "Student". Below the heading, there is a greeting: "Hello Jiang Nan" and "Please choose your operation". To the right of the text is a small icon of three people. At the bottom, there are four buttons: "DailySign", "ViewHistory", "EditPersonal", and "LogOut".

**Fig.19. Student Main page**

The user's name is displayed on the welcoming text. The students can choose their operations through this page. The buttons link to the operation pages and "LogOut" button is pressed to break the cookie, the student's picture is displayed on this page, too.

- Daily sign

The “Daily Sign” button pressed the daily page is displayed. This page is used by the students to sign their names on the attendance list. Before the students’ attendance, the student should choose Group and Course first.



The screenshot shows a window titled "Choose Course". Inside, there are two dropdown menus. The first is labeled "Group:" and has "IT 07" selected. The second is labeled "Course:" and has "Basic Java" selected. Below these menus is a button labeled "Attendance List".

**Fig.20. Daily Sign**

Pressing the “Attendance list” button will turn to the daily sign page, the course name, date and times, group and period will be displayed on this page.



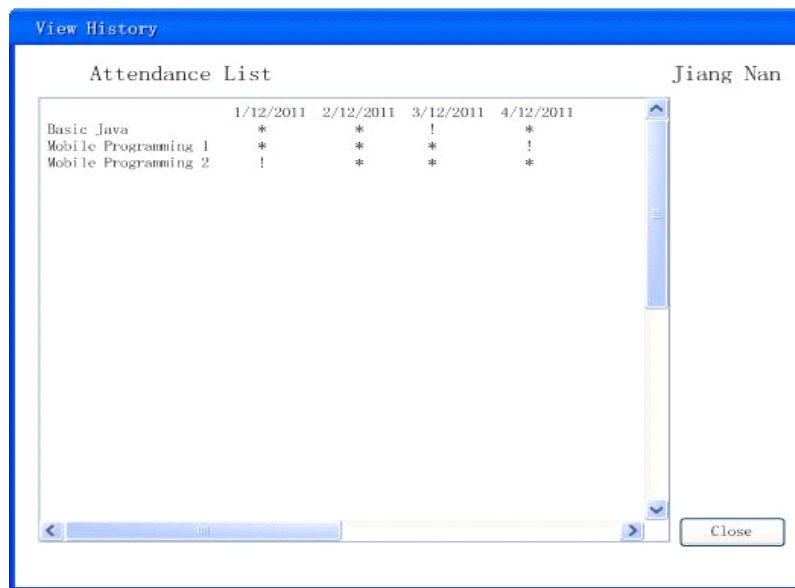
The screenshot shows a window titled "Daily Sign". At the top right, it says "Name: Jiang Nan". Below that, it displays "Teacher: Thai Bui", "Date & Time: 15:00 5/12/2011", "Group: IT07", and "Period: 4th 2011". A message in the center says "Course: Basic Java is available to sign name!". At the bottom, there are two buttons: "Sign Now" and "Later".

**Fig.21. Daily Sign**

Through pressing “Sign now” button to sign student’s name directly, or “later” button back to the student’s main page without any changes.

- View history

The “View History” button pressed, the view history page will be displayed. The attendance history will be listed and ordered by the courses’ name; in addition, the list will be displayed in the file. The close button is used to go back to the student’s main page.



The screenshot shows a window titled "View History" with a sub-header "Attendance List" and the name "Jiang Nan" in the top right corner. The main content is a table with columns for dates and attendance status.

|                      | 1/12/2011 | 2/12/2011 | 3/12/2011 | 4/12/2011 |
|----------------------|-----------|-----------|-----------|-----------|
| Basic Java           | *         | *         | !         | *         |
| Mobile Programming 1 | *         | *         | *         | !         |
| Mobile Programming 2 | !         | *         | *         | *         |

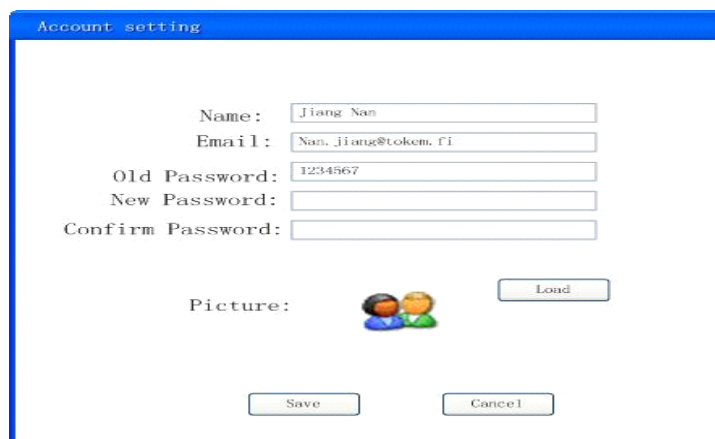
At the bottom right of the window is a "Close" button.

**Fig.22. View History**


- Edit account

The “Edit personal” button pressed, the edit account page will be displayed to the user. The default data of the user will be displayed on this page in the named field, the data can be modified by the user. The “Load” button is used to change the picture of the user. The student should type the old password which has been made before in “Old password” field. The student needs to give the new password which he or she prefers in the “New password” field. “Confirm password” is used to type the new password again to make sure the password.

The “save” button pressed, the system submits the information which has been edited to the server, and saved it into the database.



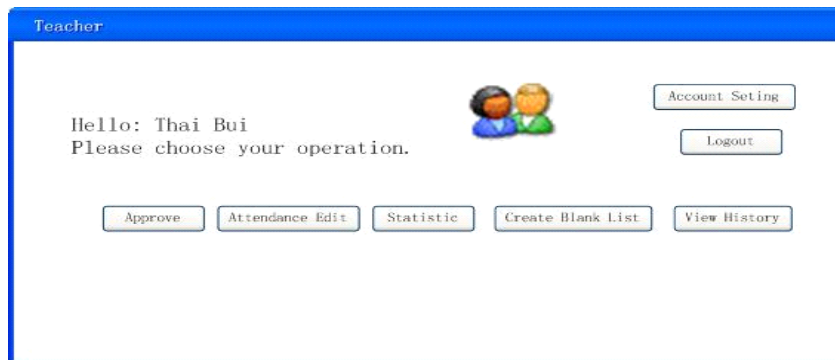
The screenshot shows a window titled "Account setting" with the following fields and buttons:

- Name:
- Email:
- Old Password:
- New Password:
- Confirm Password:
- Picture: 
- 

**Fig.23. Edit Personal Information**

### 3.5.3. Teacher

The main page of the teacher as below will be displayed to the user after teacher has logged in this system through login page.



**Fig.24. Teacher's main page**

The user's name will be displayed on the welcoming text; the teacher can choose his or her operations through this page. The buttons link to the operation pages and "LogOut" button pressed to break the cookie, the teacher's picture is displayed on this page, too.

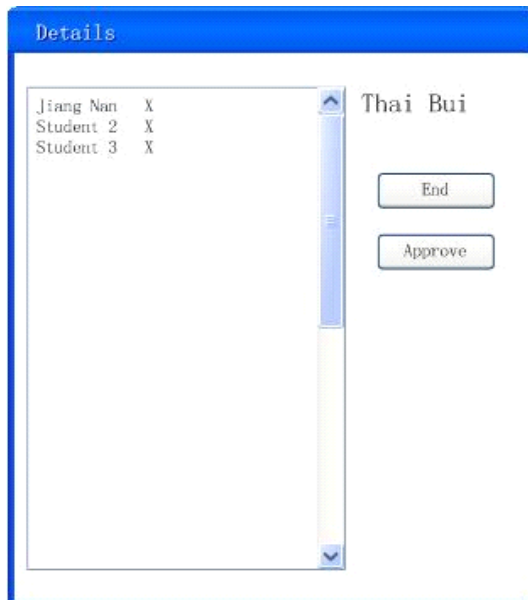
- **Approve**

Teacher can approve daily sign through this page. The courses taught by the teacher will be listed in the field. The "start signing" button is used to start the course. The close button is used to return to the teacher's main page



**Fig.25. Approve**

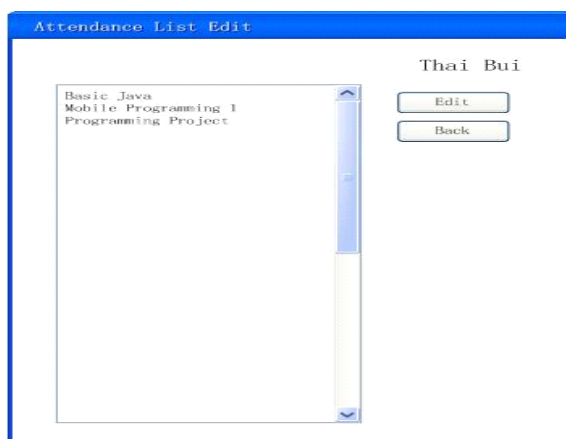
After “Start Signing” pressed, the page as below will be shown. The teacher can see which student has already signed in. The “End” button is to stop signing his or her name. The “Approve” button is used to approve the daily sign and turn back to prior page.



**Fig.26. Approve**

- Attendance list edit

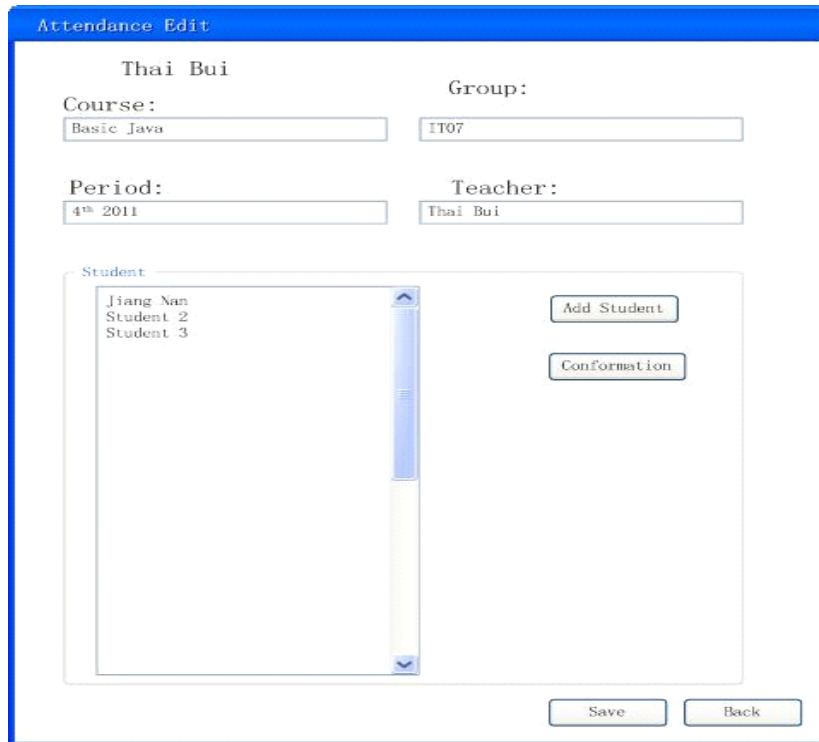
The teacher can edit the existing attendance list and the blank list through this page. The attendance list will be selected in the listing field. The “Edit” button is used to select the attendance list to edit. The “Back” button will go back to the teacher’s main page.



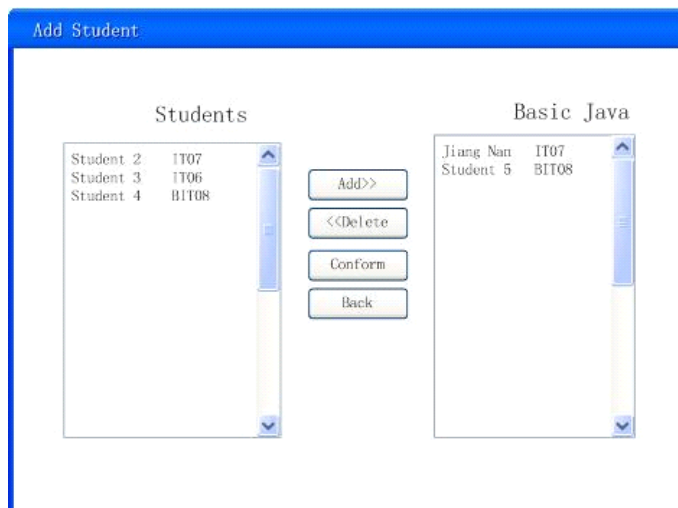
**Fig.27. Attendance List Edit**



After selecting the attendance list, the default data will be displayed in the field. It is possible to edit.



**Fig.28. Attendance List Edit**



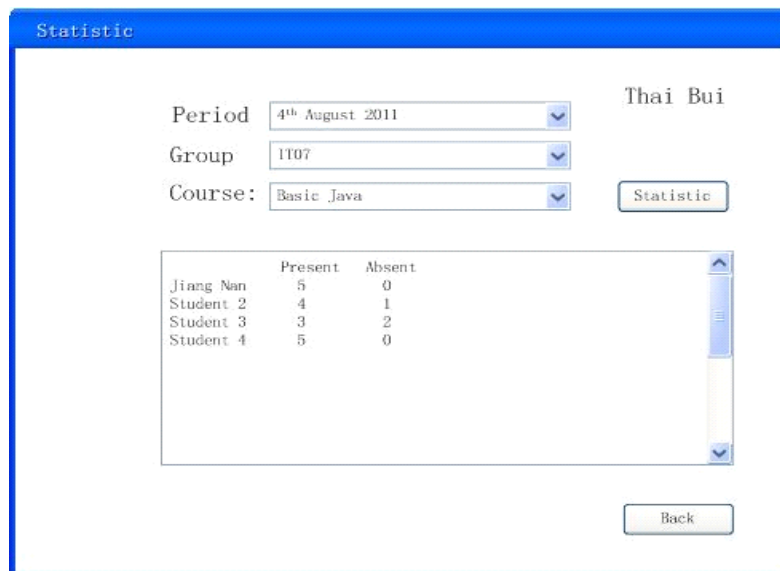
**Fig.29. Add Students**

All students are listed in the list of students list. It means the student who has in the course list on the right hand side list already. Select the student from student's list and

press the “add” button to add him or her to the course, select the student in the course list and press "Delete" button to remove the student from the course. "Confirm" button is used to confirm the edition and turn back to the prior page. The “Back” button is used to turn back to the prior page without any changes directly.

#### - Statistics

Teacher can view the count as an order of the courses. The course name should be put in the course name field. The “Statistic” button is pressed to count; the result will be listed and displayed in the listing field. “Back” button is used to go back to the teacher's main page.

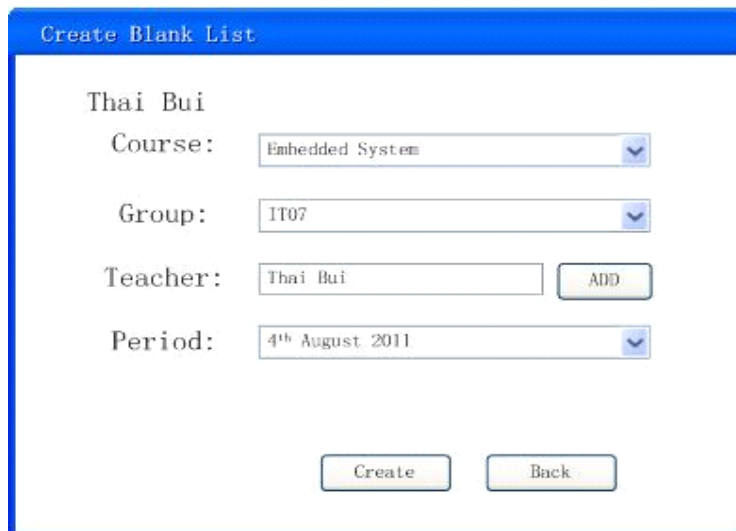


|           | Present | Absent |
|-----------|---------|--------|
| Jiang Nan | 5       | 0      |
| Student 2 | 4       | 1      |
| Student 3 | 3       | 2      |
| Student 4 | 5       | 0      |

**Fig.30. Statistics**

#### - Create Blank List

Teacher can create the new blank attendance list through this page. The course name, group, teacher, and period should be defaulted in this page. The “Create” button is used to submit the data to the server and created in the database. “Back” button is used to turn back to the teacher's main page.



Thai Bui

Course: Embedded System

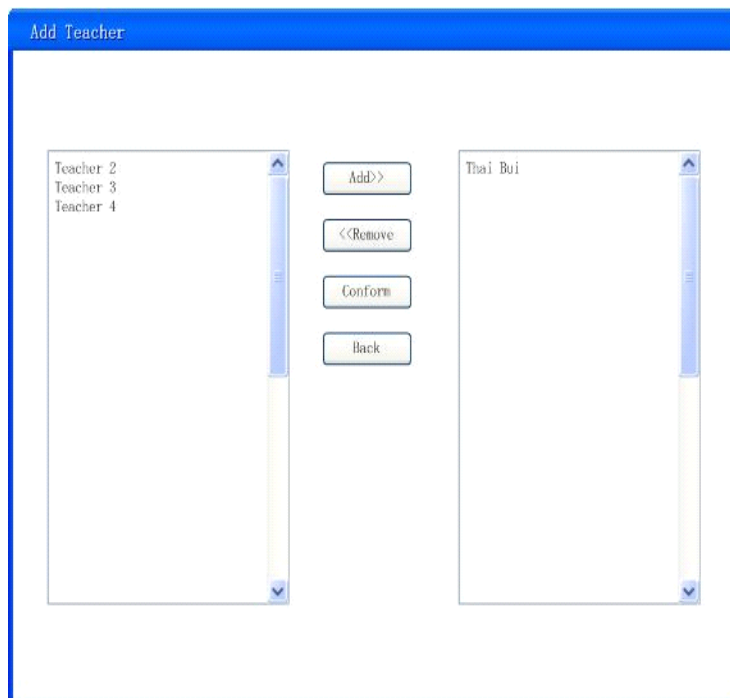
Group: IT07

Teacher: Thai Bui ADD

Period: 4<sup>th</sup> August, 2011

Create Back

**Fig.31. Create Blank List**



Add Teacher

Teacher 2  
Teacher 3  
Teacher 4

Add>>

<<Remove

Conform

Back

Thai Bui

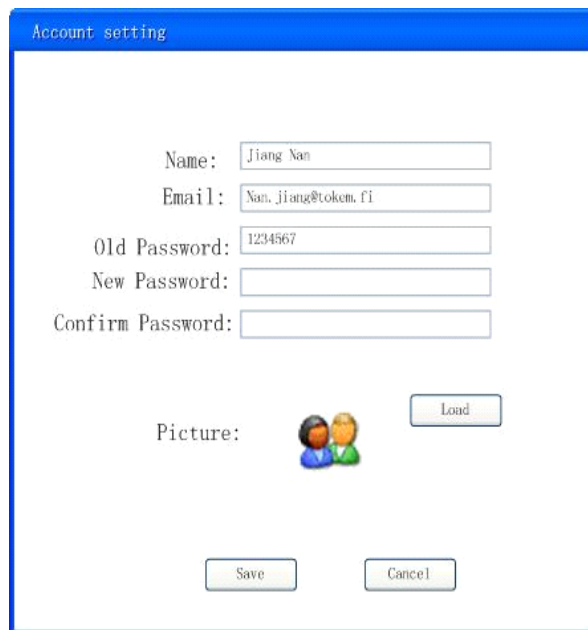
**Fig.32. Add Teacher**

- Edit account

The default data for the user will display on this page in the named field and can be modified. The “Load” button uses to change the picture of the user. The teacher should type the old password which has been made before in “Old password” field. The teacher has to give the new password which he or she prefers in the “New password” field.

“Confirm password” is used for student to type the new password again to confirm the password.

The “save” button pressed, the system submits the information after editing to the server, and saving it in the database.

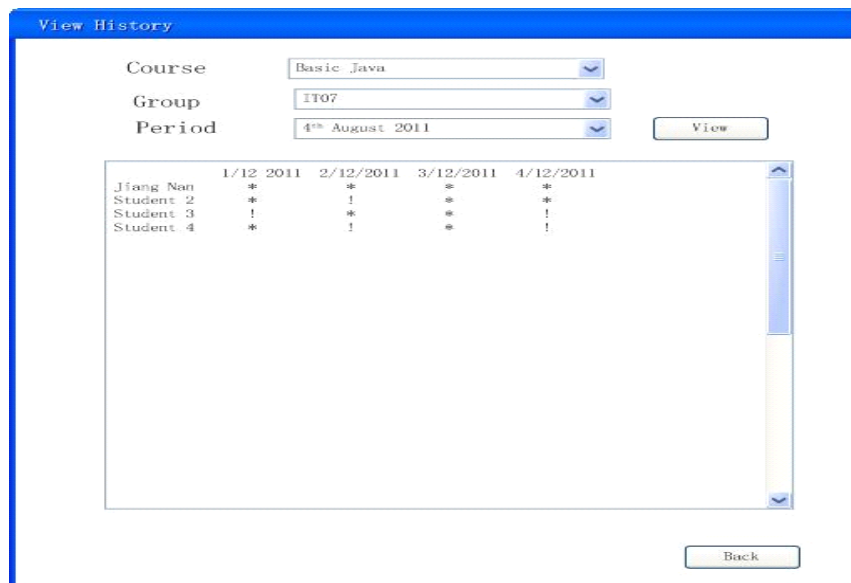


The screenshot shows a window titled "Account setting" with a blue border. Inside, there are five text input fields arranged vertically. The first field is labeled "Name:" and contains "Jiang Nan". The second is labeled "Email:" and contains "Nan.jiang@tokem.fi". The third is labeled "Old Password:" and contains "1234567". The fourth is labeled "New Password:" and is empty. The fifth is labeled "Confirm Password:" and is empty. Below these fields is a "Picture:" label next to a small icon of three people. To the right of the icon is a "Load" button. At the bottom of the window are two buttons: "Save" on the left and "Cancel" on the right.

**Fig.33. Edit Personal information**

- View History

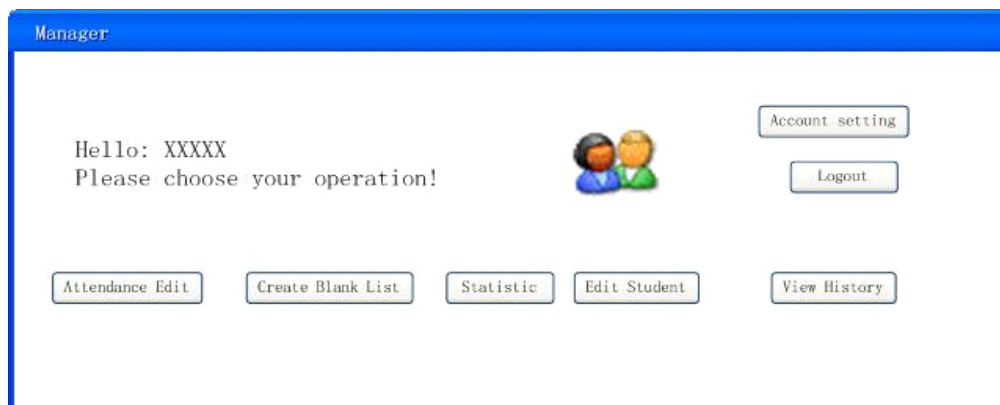
The teacher chooses the correct sections from “Course”, “Group” and “Period” of this page. Press “View” button in this page. All the attendance details of every student will be showed in the field. The student name and signing date can be found in this page. “Back” button is used to go back to the main page of the teacher.



**Fig.34. View History**

### 3.5.4. Manager

After manager login to the system, the manager's main page will be displayed to the user.

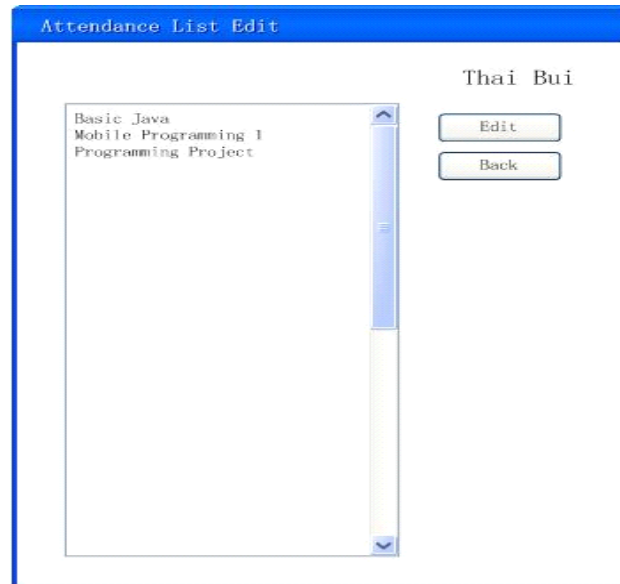


**Fig.35. Manager Main Page**

The user's name will be displayed on the welcoming text; the managers can choose their operations through this page. The buttons link to the operation pages, and "LogOut" button is pressed to break the cookie, the manager's picture will be displayed on this page, too.

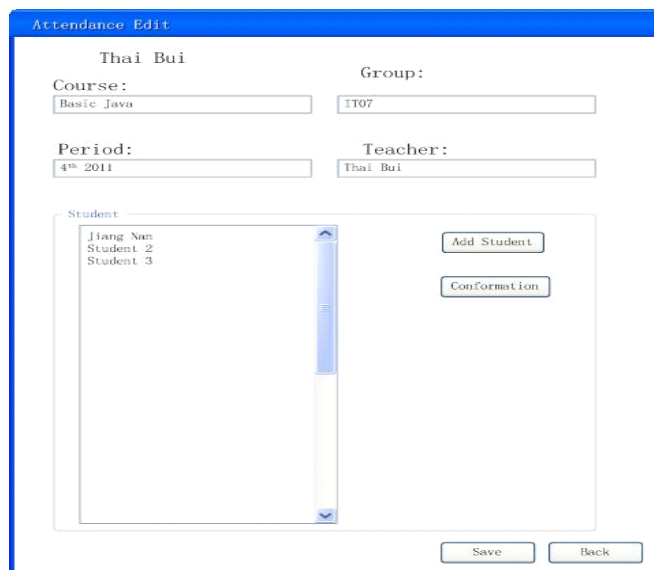
### - Attendance Edit

The manager can edit the existing attendance list and the blank list through this page. The attendance list is selected in the listing field. The “Edit” button is used to select the attendance list to edit. The “Back” button will go back to the teacher’s main page.

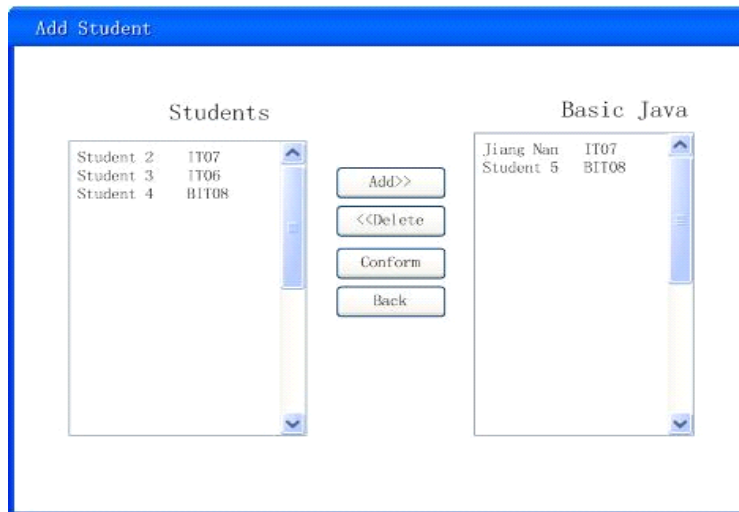


**Fig.36. Attendance List Edit**

After selecting the attendance list, the default data is displayed in this field. The user can edit it.



**Fig.37. Attendance List Edit**



**Fig.38. Edit Student**

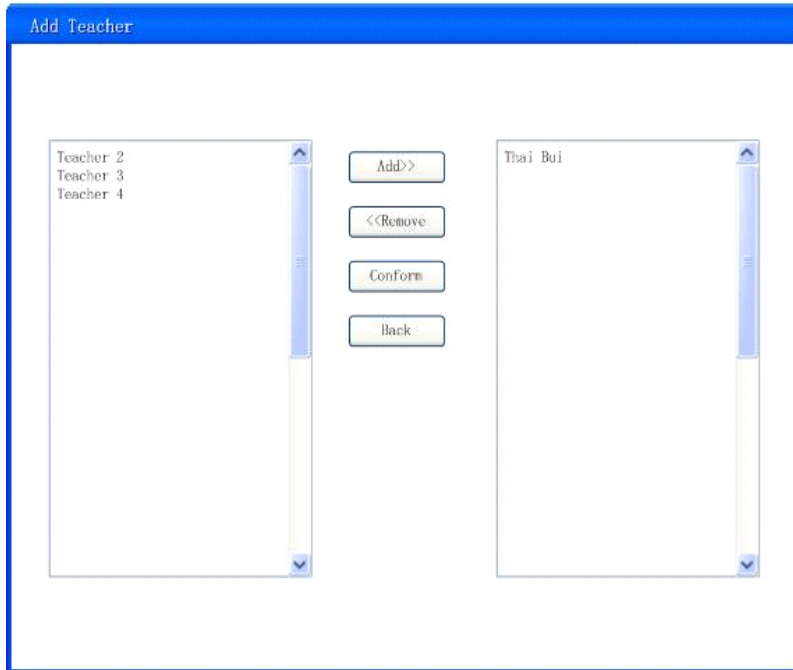
The student part can add the new student to the attendance list and remove the student from the attendance list. The “save” button is pressed to submit the information after editing to the server and to update the database. The “Back” button is pressed to display the prior page.

#### - Create Blank List

Manager can create the new blank attendance list through this page. The course name, group, teacher, and period should be defaulted in this page. The “Create” button is used to submit the data to the server and created a new course in the database. “Back” button is used to turn back to the manager’s main page.



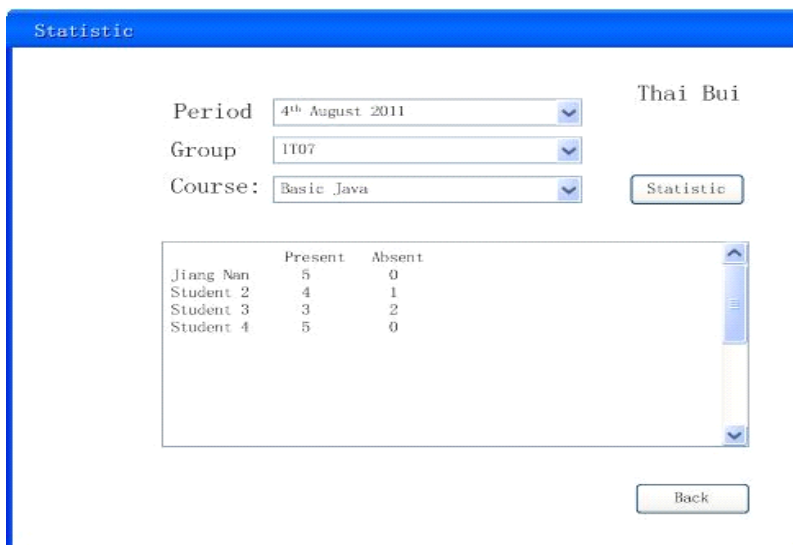
**Fig.39. Create Blank List**



**Fig.40. Add teacher**

- **Count**

Manager can count the attendance list in an order of the courses and group. The course name and group name should be default in the input field. “Statistic” button will submit the data to the server, and the server will calculate the result from the database and list the result. The list of the result is displayed in the field.

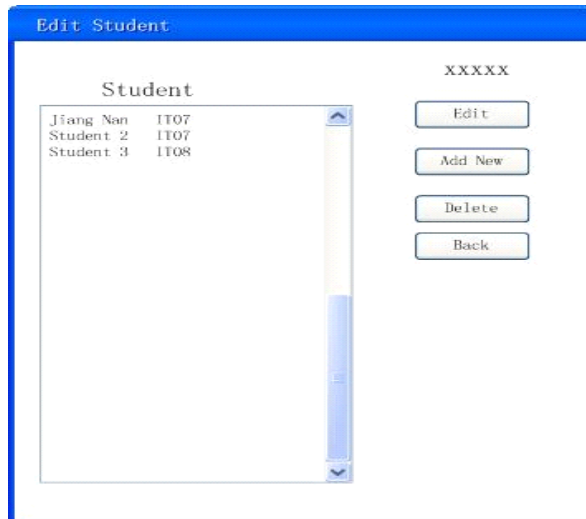


**Fig.41. Statistic**



## - Edit student

The manager can edit student from the “Edit student” button on the manager’s main page. Both editing student information and adding the new student are included in the edit part. The student in the database will be listed in the student field. Select student first before editing.



**Fig.42. Edit Student**

To add the new student, the manager should default the student’s name, email, and password. “Save” button is used to submit the student information, and create the new student in the database.

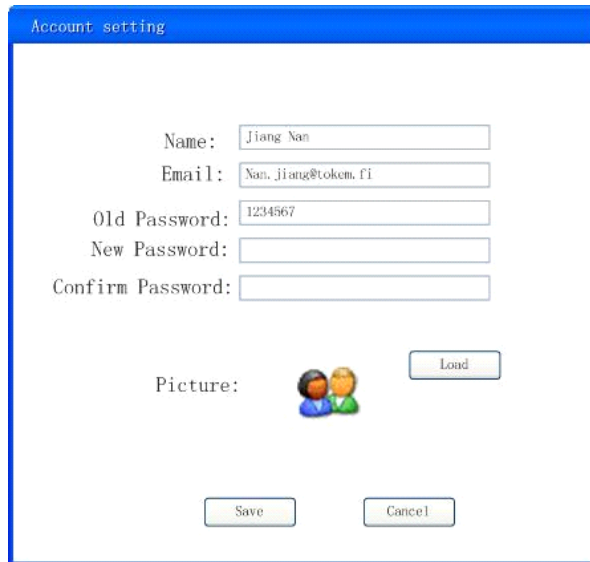


**Fig.43. Add new student information**

After selecting the student and pressing “Edit” button. The edit student page will be displayed to the user. The default data of the user will be displayed on this page in the

named field and can be modified by the user. The “Load” button is used to change the picture of the user.

The “Save” button is pressed; the system will submit the information after editing to the server, and save it in the database.



Account setting


Name:

Email:

Old Password:

New Password:

Confirm Password:

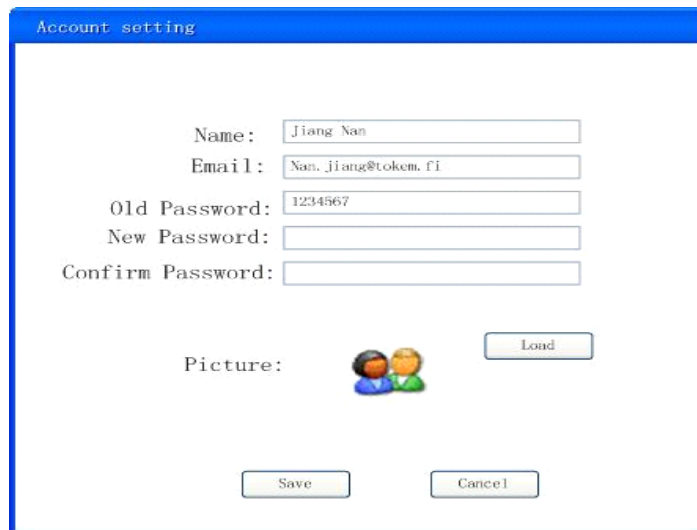
Picture: 

**Fig.44. Edit student information**

#### - **Edit account**

The default data of the user will be displayed on this page in the named place and can be modified by the user. The “Load” button is used to change the picture of the user. Manager should type the old password which was made before in the “Old password” field. The manager needs to give the new password which he or she prefers in the “New password” field. “Confirm password” is used to type the new password again to confirm the password.

The “Save” button is pressed, the system submit the information after editing to the server, and saving it in the database.



Account setting


Name:

Email:

Old Password:

New Password:

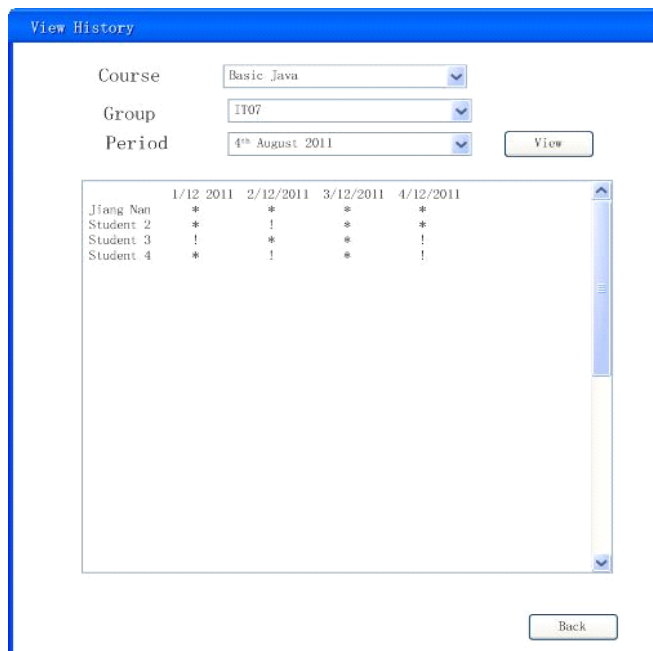
Confirm Password:

Picture: 

**Fig.45. Edit personal information**

- View History

Manager chooses the correct sections from “Course”, “Group” and “Period” of this page. Press “View” button in this page. All the attendance details of every student will be showed in the field. The student name and signing date could be found in this page. “Back” button is used to go back to the main page of the manager.



View History

Course:

Group:

Period:

|           | 1/12 2011 | 2/12/2011 | 3/12/2011 | 4/12/2011 |
|-----------|-----------|-----------|-----------|-----------|
| Jiang Nan | *         | *         | *         | *         |
| Student 2 | *         | !         | *         | *         |
| Student 3 | !         | *         | *         | !         |
| Student 4 | *         | !         | *         | !         |

**Fig.46. View History**

## 4. RESULTS

E-Attendance List is the name of this system. Students sign their names through it in the classes instead of paper-based work currently. In addition, both teachers and managers have the right to edit the course arrangements. In other words, E-Attendance List is one of several channels to assess students' commitment on a particular course at Kemi-Tornio University of Applied Science. In each contact session, the teachers and managers give out the list which attending students can mark or sign to show that they are present. An improvement of this activity is necessary so that all people feel comfortable and flexible to do the task, and then the information from this activity can be utilized in the best way.

The expected result is a completed system, including system design and system implementing. Because of lack of time, the implementing and test process cannot be done. Fortunately, with the help of the instructor, the entire design works are done. The details of the design introduced in chapter 3.

Each student should follow the system instructions after logging in. However, the sign name activity has to be done except sign name switch is opened by teachers. A period is stored in the database from opening time to ending time. For example, 40 minutes is set in database. If teachers open the switch at 9:00 am, at 9:40 am the sign name system will be closed automatically.

The times of sign name can be displayed in this system, too. The students can see the results in the history of their own main page. At the same time, students can edit their personal information at their main page, such as photos, usernames and passwords. Except editing personal information, the teachers and managers can edit the courses information, attendance of students and total quantity of students. Due to the internet, the changes of courses can be showed in real time. The students can get the latest information and follow it.

It is suggested that personal real picture upload is one of the best ways to keep the system's security. Nobody can cheat for instating of others to sign name. In teachers' field, they can see the attendance quantity of students through statistics in their main page.

## 5. CONCLUSIONS

During this project, I got a good chance to practice my technical knowledge. This is an opportunity to improve my English writing and reading.

Firstly, I used UML knowledge which I have learnt in my Bachelor study period to design E-Attendant List structure. I drew the ER diagram because the system database table is based on it. I began to build the database after the ER diagram was decided. The next step was to draw the sequence diagram of the whole system. Finally, I did the user interface. It is a pre-view of the completed system.

I spent the most time to do the ER diagram and database. They are the basic but necessary part of this system. I did not know ER diagram clearly at the beginning, especially the entity and relationship. So my instructor and I held a meeting per week to process my thesis. At the same time I found out the supporting theories and samples to prove my idea through websites. During the meeting, I discussed with my instructor. I could get the right opinions or knowledge via discussions and website searching.

Although I used almost known knowledge to complete my thesis, I got real experience of my study life. I did the design work as the design knowledge which I had learnt before. The different thing from project course is that I did each part as difference to make a complete work. In project course I could do an individual part only.

I also found I have to do everything more carefully in my future life. I need to pay enough attention for previewing. As regards my English, including speaking, writing and listening, I still should improve.

## 6. REFERENCES

- /1/ Allen I. Holub. All rights reserved, 2011, [www document],  
[ <http://www.holub.com/goodies/uml/>] 5 November 2011.
- /2/ Open Directory, [www document],  
[[http://www.dmoz.org/Computers/Programming/Methodologies/Modeling\\_Languages/Unified\\_Modeling\\_Language/Tools/](http://www.dmoz.org/Computers/Programming/Methodologies/Modeling_Languages/Unified_Modeling_Language/Tools/)] 8 November 2011.
- /3/ UML overview, Tutorials point, [www document],  
[[http://www.tutorialspoint.com/uml/uml\\_overview.htm](http://www.tutorialspoint.com/uml/uml_overview.htm)] 15 November 2011.
- /4/ Entity-Relationship Diagram, smartDraw, [www.document],  
[<http://www.smartdraw.com/resources/tutorials/entity-relationship-diagrams/>] 18 November 2011.
- /5/ Professor Sauter, Entity-Relationship Diagrams, [www.document],  
[ [http://www.umsl.edu/~sauterv/analysis/er/er\\_intro.html](http://www.umsl.edu/~sauterv/analysis/er/er_intro.html)] 22 November 2011.
- /6/ Mike Chapple, Entity-Relationship Diagram, [www document],  
[<http://databases.about.com/cs/specificproducts/g/er.htm>] 22 November 2011.
- /7/ UML Diagram, smartDraw, [www document],  
[<http://www.smartdraw.com/resources/tutorials/uml-diagrams/>] 25 November 2011.
- /8/ OMG, Catalog of OMG Modeling And Metadata Specifications, [www document],  
[[http://www.omg.org/technology/documents/modeling\\_spec\\_catalog.htm](http://www.omg.org/technology/documents/modeling_spec_catalog.htm)] 30 November 2011.
- /9/ Mario Jeckle, Unified Modeling Language, [www document],  
[[http://www.jeckle.de/uml\\_spec.htm](http://www.jeckle.de/uml_spec.htm)] 30 November 2011.
- /10/ SourceMaking, UML, [www document], [<http://sourcemaking.com/uml>] 12 December 2011.