Hong Kong Community College
2013-2014
Semester 1
CCN2240 Databases Systems
Final Project report

GP101B-GP01

Student Name	Student Number	Class
Chan Kwan Wing (Ken)	12607532A	101B
Chan Him Hei (Eric)	12313032A	101B
Law Chiu Kwan (Andy)	12650300A	101B
So Mau Chiu (Gary)	12660032A	101B

Topic : College Management System

Table of Contents

1. Introduction

This is a management system for some big colleges to use , such as HKCC. We design this system as we see there are several problems caused by manual process of booking room.

Problems:

- School administrators has so much works to do
- Booking process cannot be done in a 24-hours basis
- Club and Lecturers cannot book the room easily

Our system reduces the workload by allocate different access right to different parties to enhance the efficiency of school management.

Different functions can be used by different parties:

- School administrators
    - The greatest access right
    - Maintain daily school operation
    - Create, modify and delete new teachers, students, lessons and club etc.
- Club administrators
    - Enroll new member of each club
    - Book some room for activities
- Teachers
    - Add Grade to Student
    - Book some room for activities ,like talks or workshop, for different students.
    - View lessons' information
- Students
    - Find the information about the details of the class and activities
    - Register the lessons

2. Description of business rules

1. Each Lecturer can book zero to many bookings.
   Each booking belongs to only one Lecturer.

2. Each Room has zero to many Bookings.
   Each Booking belongs to only one room.

3. Each Club can book zero to many bookings.
   Each booking belongs to only one club.

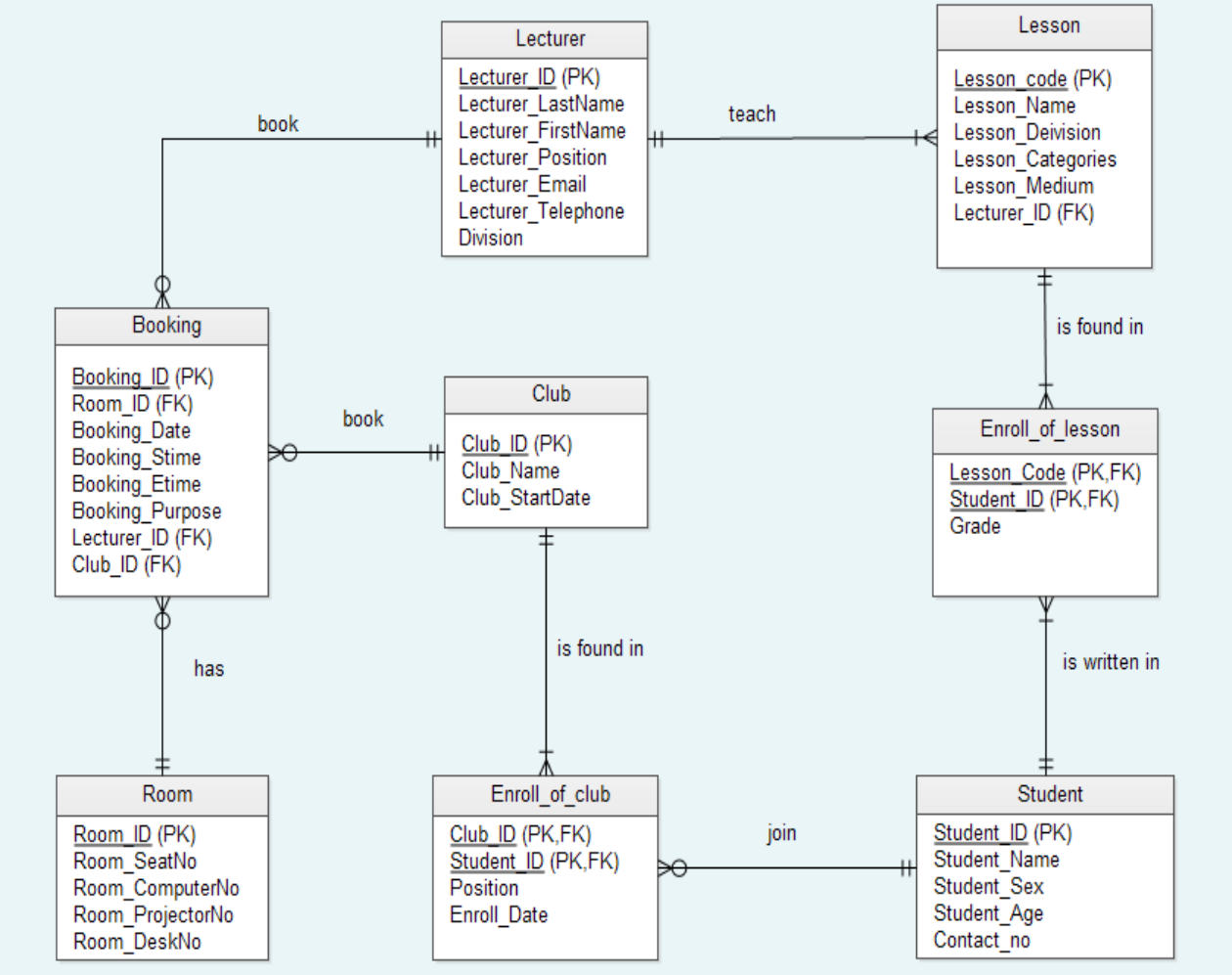
4. Each Lecturer teach one lesson only.
   Each lesson is taught by only one Lecturer.

5. Each club has at least one student.
   Each student can join one to many clubs, or not join.

6. Each student enrolls one to many lessons.
   Each lesson has one to many students

3. Conceptual Model

3.1 Entity Relationship Diagram



4.Logical design

4.1 Relations

Entity (Strong Entity)

1. Booking (**Booking_ID** , Room_ID , Booking_Date , Booking_Stime , Booking_etime , Booking_Purpose , Lecture_ID , Club_ID)
2. Lecturer (**Lecture_ID**, Lecturer_LastName, Lecturer_FirstName, Lecturer_Position, Lecturer_Email, Lecturer_Telephone, Division)
3. Student (**Student_ID**, Student_Name, Student_Sex, Student_Age, Contact_no)
4. Room(**Room_ID**, Room_SeatNo, Room_ComputerNo, Room_ProjectorNo, Room_DeskNo)
5. Club (**Club_ID**, Club_Name, Club_StartDate)
6. Lesson (**Lesson_code**, Lesson_Name, Lesson_Division, Lesson_Categories, Lesson_Medium, Lecterer_ID)

Composite Entity (Weak Entity)

7. Enroll_of_club(**Club_ID**, **Student_ID** , Position, Enroll_Date)
8. Enroll_of_lesson (**Lesson_Code**, **Student_ID**, Grade)

4.2 Relationship

1:M Relationship

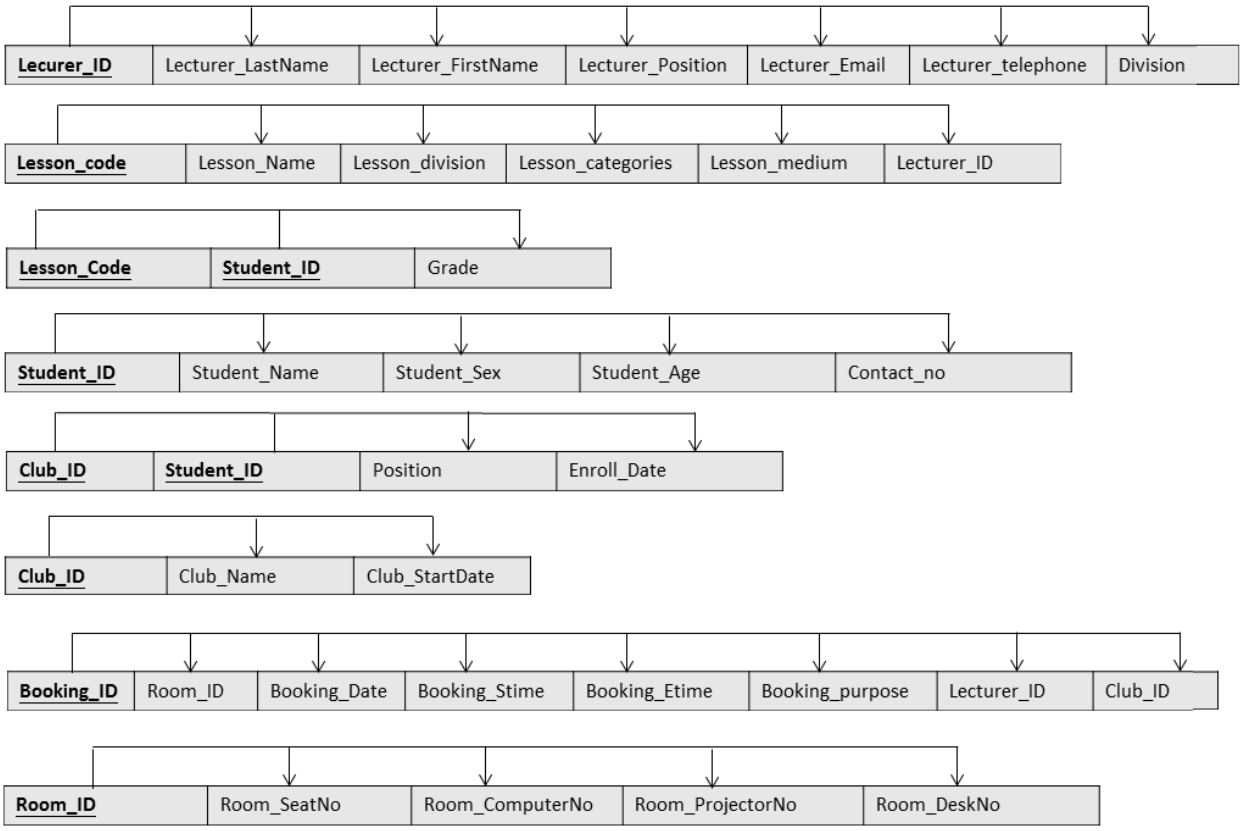
1. The relationship between LECTURER and BOOKING is 1:M.
2. The relationship between ROOM and BOOKING is 1:M.
3. The relationship between CLUB and BOOKING is 1:M.
4. The relationship between LECTURER and LESSON is 1:M

M:N Relationship

5. The relationship between LESSON and STUDENT is M:N.
   (Break into two 1:M relationship and add a composite entity)
   The relationship between ENROLL_OF_LESSON and STUDENT is M:1.
   The relationship between LESSON and ENROLL_OF_LESSON is 1:N.

6. The relationship between CLUB and STUDENT is M:N
   (Break into two 1:M relationship and add a composite entity)
   The relationship between ENROLL_OF_CLUB and STUDENT is M:1.
   The relationship between LESSON and ENROLL_OF_CLUB is 1:N.

4.3 Schema


Lecurer_ID
Lecturer_LastName
Lecturer_FirstName
Lecturer_Position
Lecturer_Email
Lecturer_telephone
Division
Lesson_code
Lesson_Name
Lesson_division
Lesson_categories
Lesson_medium
Lecturer_ID
Lesson_Code
Student_ID
Grade
Student_ID
Student_Name
Student_Sex
Student_Age
Contact_no
Club_ID
Student_ID
Position
Enroll_Date
Club_ID
Club_Name
Club_StartDate
Booking_ID
Room_ID
Booking_Date
Booking_Stime
Booking_Etime
Booking_purpose
Lecturer_ID
Club_ID
Room_ID
Room_SeatNo
Room_ComputerNo
Room_ProjectorNo
Room_DeskNo

4.4 Query and Application

As we want our interface more beautiful, we will print out the title by ourselves. Therefore, the SQL flied name may not be as same as in the website.

Also, the highlighted word is variable that we may ask user to input in the websites.

1. General

```
CREATE TABLE booking (
  Booking_ID        int(6)          NOT NULL        auto_increment,
  Room_ID           varchar(4)      NOT NULL,
  Booking_Date      date            NOT NULL,
  Booking_STime     time            NOT NULL,
  Booking_ETime     time            NOT NULL,
  Lecturer_ID       int(6)          default NULL,
  Club_ID           int(6)          default NULL,
  Booking_Purpose   varchar(30)     NOT NULL,
  PRIMARY KEY (Booking_ID),
  FOREGIN KEY (Room_ID)
        REFERENCE Room (Room_ID)
        ON DELETE CASCADE ON UPDATE CASCADE,
  FOREGIN KEY (Lecturer_ID)
        REFERENCE Lecturer (Lecturer_ID)
        ON DELETE CASCADE ON UPDATE CASCADE,
  FOREGIN KEY (Club_ID)
        REFERENCE Room (Room_ID)
        ON DELETE CASCADE ON UPDATE CASCADE
);
```

Description	Create table booking

```
CREATE TABLE club (
  Club_ID           int(6)          NOT NULL        auto_increment,
  Club_Name         varchar(30)     NOT NULL,
  Club_StartDate    date            NOT NULL,
  PRIMARY KEY  (Club_ID),
);
```

Description	Create table Club

```
CREATE TABLE enroll_of_club (
  Club_ID          int(6)          NOT NULL        auto_increment,
  Student_ID       char(9)         NOT NULL,
  Position         varchar(30)     NOT NULL,
  Enroll_date      date            NOT NULL,
  PRIMARY KEY (Club_ID ,Student_ID),
  FOREGIN KEY (Club_ID)
        REFERENCE club (Club_ID)
        ON DELETE CASCADE ON UPDATE CASCADE,
  FOREGIN KEY (Student_ID)
        REFERENCE student (Student_ID)
              ON DELETE CASCADE ON UPDATE CASCADE
);
```

Description	Create table enroll_of_club

```
CREATE TABLE enroll_of_lesson (
  Lesson_Code          char(7)          NOT NULL,
  Student_ID           char(9)          NOT NULL,
  Grade                varchar(2)       default NULL,
  PRIMARY KEY  (Lesson_Code, Student_ID)
  FOREGIN KEY (Lesson_Code)
        REFERENCE lesson (Lesson_Code)
        ON DELETE CASCADE ON UPDATE CASCADE,
  FOREGIN KEY (Student_ID)
        REFERENCE student (Student_ID)
        ON DELETE CASCADE ON UPDATE CASCADE
 )
```

Description	Create table enroll_of_lesson

```
CREATE TABLE lecturer (
  Lecturer_ID              int(6)          NOT NULL        auto_increment,
  Lecturer_LastName        varchar(30)     NOT NULL,
  Lecturer_FirstName       varchar(30)     NOT NULL,
  Lecturer_Position        varchar(30)     NOT NULL,
  Lecturer_Email           varchar(30)     NOT NULL,
  Lecturer_Telephone       char(8)         NOT NULL,
  Division                 varchar(30)     NOT NULL,
 PRIMARY KEY  (`Lecturer_ID`),
 UNIQUE (`Lecturer_Email`,`Lecturer_Telephone`)
);
```

Description	Create table Lecturer

```
CREATE TABLE lesson (
 Lesson_Code          char(7)          NOT NULL,
 Lesson_Name          varchar(50)      NOT NULL,
 Lesson_Division      varchar(15)      NOT NULL,
 Lesson_Categories    char(2)          NOT NULL,
 Lesson_Medium        varchar(10)      NOT NULL,
 Lecturer_ID          int(6)           NOT NULL,
 PRIMARY KEY  (Lesson_Code),
 FOREGIN KEY (Lecturer_ID)
        REFERENCE Lecturer (Lecturer_ID)
       ON DELETE CASCADE ON UPDATE CASCADE,
);
```

Description	Create table lesson

```
CREATE TABLE room (
 Room_ID              varchar(4)       NOT NULL,
 Room_SeatNo          int(3)           NOT NULL,
 Room_ComputerNo      int(3)           NOT NULL,
 Room_ProjectorNo     int(3)           NOT NULL,
 Room_DeskNo          int(3)           NOT NULL,
 PRIMARY KEY  (Room_ID)
)
```

Description	Create table Club

```
CREATE TABLE student (
 Student_ID           char(9)          NOT NULL,
 Student_name         varchar(30)      NOT NULL,
 Student_Sex          char(1)          NOT NULL,
 Student_Age          int(2)           NOT NULL,
 Contact_No           char(8)          NOT NULL,
 PRIMARY KEY  (Student_ID),
 CHECK(Student_Age > 0)
)
```

Description	Create table Club

2. School Admin

Query:

```
SELECT      d.Student_ID, d.Student_Name
FROM        lecturer a, lesson b, enroll_of_lesson c, student d
WHERE       a.Lecturer_ID= b.Lecturer_ID
AND         b.Lesson_code= c.Lesson_code
AND         c.Student_ID= d.Student_ID
AND         a.Lecturer_ID= [ Lecturer_ID]
ORDER BY    Student_ID;
```

Description	Find out the student that is taught by a specific lecturer.

```
SELECT      a.Student_ID, a.Student_Name
FROM        student a, enroll_of_club b, club c
WHERE       a.Student_ID= b.Student_ID
AND         b.Club_ID= c.Club_ID
AND         c.Club_ID= [Club_ID]
ORDER BY    Student_ID;
```

Description	Find out the student that is in a specific club.

```
SELECT      *
FROM        lecturer where Lecturer_Position =
(SELECT     Lecturer_Position
FROM        lecturer
WHERE       Lecturer_ID=[Lecturer_ID]);
```

Description	Find out the lecturers' information who have the same position as another specific lecturer.

Form:

```
INSERT INTO student
(Student_ID, Sudent_Name, Student_Sex, Student_Age, Contact_no)
VALUE
([Student_ID], [Student_Name], [Student_Sex], [Student_Age],
[Contact_no]);
```

Description	This is a query to add student information when student start studying in the college.

	INSERT INTO lecturer (Lecturer_LastName, Lecturer_FirstName,Lecturer_Position, Lecturer_Email, Lecturer_Telephone, Division) VALUES ([Lecturer_LastName], [Lecturer_FirstName], [Lecturer_Position] , [Lecturer_Email], [Lecturer_Telephone], [Division]);
Description	Create a data about new lecturer.

	INSERT INTO lesson (Lesson_Code ,Lesson_Name, Lesson_Division, Lesson_Categories, Lesson_Medium, Lecturer_ID) VALUES ([Lesson_Code], [Lesson_Name], [Lesson_Division], [Lesson_Categories] , [Lesson_Medium], [Lecturere_ID]);
Description	CREATE a new record about an information of a new lesson

	INSERT INTO club (Club_Name, Club_StartDate) VALUES ([Club_Name], [Club_StartDate]);
Description	CREATE a new record about an information of a new club.

	INSERT INTO room (Room_ID , Room_SeatNo, Room_ComputerNo, Room_ProjectorNo, Room_DeskNo) VALUE ([Room_ID], [Room_SeatNo], [Room_ComputerNo], [Room_ProjectorNo] , [Room_DeskNo]);
Description	CREATE a new record about an information of a new room.

	INSERT INTO enroll_of_lesson (Lesson_Code ,Student_ID) VALUES ([Lesson_Code], [Student_ID]);
Description	CREATE a new record about an enrollment of a new lesson when student has enrolled a lesson.

	UPDATE lecturer SET Lecturer_LastName= [LastName], Lecturer_FirstName = [FirstName], Lecturer_Position = [Position], Lecturer_Email = [Email], Lecturer_Telephone = [Telephone], Division = [Division] WHERE Lecturer_ID= [Lecturer_ID];
Description	Update the lecturer information.

	UPDATE student SET Student_Name = [Name], Student_Sex = [Student_Sex], Student_Age = [Student_Age], Contact_no = [Contact_no] WHERE Student_ID = [Student_ID];
Description	Update the student information.

	UPDATE lesson SET Lesson_Name = [Name], Lesson_Division= [Lesson_Division], Lesson_Categories = [Lesson_Categories], Lesson_Medium= [Lesson_Medium], Lecturer_ID= [Lecturer_ID] WHERE Lesson_code = [Lesson_code];
Description	Update the lesson information.

	DELETE FORM student WHERE Student_ID = [Student_ID];
Description	This is a query to delete student information when student quit the college.

	DELETE FORM room WHERE Room_ID = [Room_ID];
Description	This is a query to delete a specific room information.

	DELETE FORM club WHERE Club_ID = [Club_ID];
Description	This is a query to delete a specific club information.

```
DELETE FORM     enroll_of_lesson
WHERE               Student_ID = [Student_ID]
  AND               Lesson_code = [Lesson_code];
```

Description	This is a query to delete the lesson taken by student when the student drops that course.

```
DELETE FORM     lecturer
WHERE               Lecturer_ID = [Lecturer_ID];
```

Description	Delete a specific lecturer from database.

```
DELETE FORM    booking
  WHERE            Booking_ID = [Booking_ID];
```

Description	Delete the booking if user presses the button.

```
  DELETE FORM    lesson
  WHERE              Lesson_code = [Lesson_code];
```

Description	Delete a specific lesson from database.

Report:

Report 1:

This is a report to show how many lesson that lecturer teach.

Lecturer ID	Lecturer Last Name	Lecturer First Name	Number of Lesson
1	LEUNG	Wing nin	1
4	CHAU	Chun Pong	1
8	WONG	Yiu Tong	1
11	CHAN	Wai Heung	1
16	CHAN	Pik Wah	2
19	CHEUNG	Hing Keung	1
21	CHIU	Hon Sun	1
23	CHOW	Kin Keung	1
24	CHU	Yiu Chung	1
26	HO	Wai Tung	1

```
SELECT          a.Lecturer_ID, Lecturer_LastName,
                Lecturer_FirstName,count(*) as  Number_of_lesson
FROM            lecturer a, lesson b
WHERE           a.Lecturer_ID = b.Lecturer_ID
GROUP BY        a.Lecturer_ID, Lecturer_LastName, Lecturer_FirstName  ;
```

Description	Find out how much lesson a lecturer teach.

Report 2:

Lesson Code	Lesson Name	Number of Student
CCN3140	Programming Project	2
CCN2273	Operating Systems	2
CCN2242	Object Oriented Programming	1
CCN2241	Discrete Structures	1
CCN2240	Database Systems	1
CCN2239	Data Structures	1
CCN2231	General Biochemistry	3
CCN2132	Principles of Investments	1
CCN2131	Marketing Research Fundamentals	1
CCN2130	Marketing in China	1
CCN2129	Introduction to Macroeconomics	1

```
SELECT      a.Lesson_code, a.Lesson_Name ,
            count(*) as Number_of_student
FROM        lesson a, enroll_of_lesson b, student c
WHERE       a.Lesson_Code = b.Lesson_Code
AND         b.Student_ID = c.Student_ID
GROUP BY    a.Lesson_code, a.Lesson_Name
ORDER BY    a.Lesson_code desc
```

Description: Find out how many students in the lesson.

Report 3:

Find out the number of the booking booked by lecturer.

Lecturer ID	Lecturer Last Name	Lecturer First Name	Number of booking
3	CHAN	Po Yin	1
5	HEUNG	Ying Yee	3
6	LO	Chi Hang	1
7	SO	Chi Ho	1
13	CHAN	Hoi Huen	1
16	CHAN	Pik Wah	9
18	CHAN	Fung Ming	1
19	CHEUNG	Hing Keung	1
24	CHU	Yiu Chung	1
26	HO	Wai Tung	1
27	KWAN	Ka Ying	3

```
SELECT      a.Lecturer_ID , b.Lecturer_LastName, b.Lecturer_FirstName,
            count(*) as No_booking_by_lecturer
FROM        booking a, lecturer b
WHERE       a.Lecturer_ID=b.Lecturer_ID
AND         a.Lecturer_ID is not null
GROUP BY    a.Lecturer_ID
```

Description: Find out the number of the booking booked by lecturer.

Report 4:

Find out the number of the booking booked by Club.

Club ID	Club Name	Number of booking
1	Accounting and Finance Society	1
2	Badminton Society	1
3	Band Society	1
4	Christian Fellowship	2
5	Computer Society	2
6	Dance Society	2
7	Drama Society	1
8	English Society	1
9	Film Appreciation Club	1
10	Hiking Society	1
11	Hong Kong Award for Young Peop	1
12	Music Society	1
13	Photography Society	1

```
SELECT      a.Club_ID, b.Club_Name , count(*) as No_booking_by_club
FROM        booking a, club b
WHERE       a.Club_ID=b.Club_ID
AND         a.Club_ID is not null
GROUP BY    a.Club_ID  ;
```

Description Find out the number of the booking booked by Club.

3. Lecturer

a. Query

```
SELECT      Room_ID
FROM        room
WHERE       Room_ID = [Room_ID];
```

Description	This is a query to check whether user input room ID is vaild or not.

```
SELECT      *
FROM        room
WHERE       Room_ID = [Room_ID];
```

Description	This is a query to show the room information which you want.

```
SELECT      Booking_ID
FROM        booking
WHERE       Booking_ID = [Booking_ID];
```

Description	This is a query to check whether user input booking ID is valid or not.

```
SELECT      *
FROM        booking
WHERE       Booking_ID = [Booking_ID];
```

Description	This is a query to show the room booking information which you want.

```
SELECT      Room_ID, Booking_Date,
            Booking_STime, Booking_ETime
FROM        booking
WHERE       Room_ID = [Room_ID]
AND         Booking_Date = [Booking_Date]
AND         Booking_STime
BETWEEN     [Booking_Start_Time]
AND         [Booking_End_Time];
```

Description	This is a query to check the room avabilablity.

```
SELECT      *
FROM        booking
WHERE       Lecture_ID <> 'NULL'
ORDER BY    Booking_Date, Room_ID,
            Booking_STime;
```

Description	This is a query to show all the booking booked by lecturer.

```
SELECT      Lecturer_ID
FROM        lecturer
WHERE       Lecturer_ID = [Lecturer_ID];
```

Description	This is a query to check whether user input lecturer ID is valid or not.

```
SELECT      Lesson_Code, Lesson_Name,
```

	Lesson_Division FROM lesson WHERE Lesson_Division = [Lesson_Division];
Description	This is a query to show a lesson division of a subject.

	SELECT Lesson_Code, Lesson_Name, Lesson_Medium FROM lesson WHERE Lesson_Medium = [Lesson_Medium];
Description	This is a query to show a teaching language of a subject.

	SELECT Lesson_Code, Lesson_Name, Lesson_Categories FROM lesson WHERE Lesson_ Categories = [Lesson_Categories];
Description	This is a query to show a lesson category of a subject.

b. Form

```
INSERT INTO booking
(Room_ID ,Booking_Date ,Booking_STime,
Booking_ETime,Lecturer_ID ,Booking_Purpose)
VALUE
([Room_ID], [Booking_Date], [Booking_Start_Time],
[Booking_End_Time] , [Booking_Lecturer_ID],
[Booking_Purpose]);
```

Description	CREATE a new row for Booking to save the booking into database.

```
UPDATE    booking
SET       Room_ID = [Room_ID]
AND       Booking_Date = [Booking_Date]
AND       Booking_STime = [Booking_Start_Time]
AND       Booking_ETime = [Booking_End_Time]
AND       Lecturer_ID = [Lecturer_ID]
AND       Booking_Purpose = [Booking_Purpose]
WHERE     Booking_ID = [Booking_ID];
```

Description	Update the information of the booking if the user input the change.

```
DELETE FROM    booking
WHERE          Booking_ID = [Booking_ID];
```

Description	Delete the booking .

```
UPDATE    enroll_of_club
SET       Grade = [Grade]
WHERE     Student_ID = [Student_ID]
AND       Lesson_Code = [Lesson_Code];
```

Description	Update the academic grade of lesson if student has finished the lesson.

c. Report

This is a report to show the frequency of using each lecturer room.

Room ID	Frequency of Room ID
101	7
103	3
105	1
106	1
107	9
1202	8
1203	1
1207	3
201	4
203	6
205	1
206	1
209	3

```
SELECT      Room_ID,
            COUNT(*) AS 'Frequency of Room ID'
FROM        booking
GROUP BY    Room_ID;
```

Description: A report to show the frequency of using each lecturer room.

This is a report to show the most highest frequency of using the lecturer room

Room ID	Frequency
101	7

```
SELECT      Room_ID,
            COUNT(*) AS Frequency
FROM        Booking
GROUP BY    Room_ID
HAVING      Room_ID <= ALL (
                SELECT  Room_ID
                FROM    booking);
```

Description: A report to show the most highest frequency of using the lecturer room.

This is a report to show the frequency of using the lecturer room in each day.

Booking Date	Frequency of Booking Date
2013-11-20	3
2013-11-21	1
2013-11-25	4
2013-11-26	4
2013-11-27	12
2013-11-28	3
2013-11-29	5
2013-11-30	1
2013-12-02	4
2013-12-03	1
2013-12-04	2

```
SELECT      Booking_Date,
            COUNT(*) AS 'Frequency of Booking Date'
FROM        booking
GROUP BY    Booking_Date;
```

Description A report to show the frequency of using the lecturer room in each day.

This is a report to show the number of count of lesson division.

Lesson Division	Number of lesson
Business	11
Engineering	7
IT	11
Language	2
Lesson_Division	1
Science	8

```
SELECT      Lesson_Division,
            COUNT(*) AS 'Number of lesson'
FROM        Lesson
GROUP BY    Lesson_Division;
```

Description A report to show the number of count of lesson division.

This is a report to show the number of count of lesson categories

Lesson Categories	Number of lesson
DS	32
GE	7
Le	1

```
SELECT      Lesson_Categories,
            COUNT(*) AS 'Number of lesson'
FROM        lesson
GROUP BY    Lesson_Categories;
```

Description A report to show the number of count of lesson categories.

This is a report to show the number of count of lesson medium.

Lesson Medium	Number of lesson
Chinese	1
English	38
Lesson_Med	1

```
SELECT      Lesson_Medium,
            COUNT(*) AS 'Number of lesson'
FROM        lesson
GROUP BY    Lesson_Medium;
```

Description A report to show the number of count of lesson medium.

4. Student

a. Query

```
SELECT    student_ID
FROM      student
WHERE     student_ID = [student_ID];
```

Description	This is a query to check whether user input student ID is valid or not.

```
SELECT    a.Student_Name, b.Club_Name, c.Position
FROM      student a, club b, enroll_of_club c
WHERE     a.Student_ID = c.Student_ID
AND       b.Club_ID = c.Club_ID
AND       a.Student_ID = [Student_ID];
```

Description	This is a query to check which club the student belongs to and their position.

```
SELECT    a.Student_Name, b.Lesson_Code, Lesson_Name, Grade
FROM      student a, lesson b, enroll_of_lesson c
WHERE     a.Student_ID = c.Student_ID
AND       b.Lesson_code = c.Lesson_code
AND       Student_ID = [Student_ID];
```

Description	This is a query to check which lessons the student has taken and their grade.

```
SELECT    Booking_Date, Booking_Stime, Booking_Etime,
Booking_Purpose
FROM      student a, enroll_of_club b, club c, booking d
WHERE     a.Student_ID = b.Student_ID
AND       b.Club_ID = c.Club_ID
AND       c.Club_ID = d.Club_ID
AND       a.Student_ID = [Student_ID]
ORDER BY  Booking_Date;
```

Description	This is a query to check if there is activities the student's club is going to hold.

```
SELECT    Booking_Date, Booking_Stime, Booking_Etime,
Booking_Purpose
FROM      student a, enroll_of_lesson b, lesson c, lecturer d, booking e
WHERE     a.Student_ID = b.Student_ID
AND       b.Lesson_Code = c.Lesson_Code
AND       c.Lecturer_ID = d.Lecturer_ID
AND       d.Lecturer_ID = e.Lecturer_ID
AND       a.Student_ID = [Student_ID]
ORDER BY  Booking_Date;
```

Description	This is a query to check if there is lessons the student have to attend.

```
SELECT    c.Lesson_Code, c.Lesson_Name, d. Lecturer_LastName,
          d.Lecturer_FirstName,
```

```
                    d.Lecturer_Email, d.Lecturer_Telephone
FROM                student a, enroll_of_lesson b, lesson c, lecturer d
WHERE               a.Student_ID = b.Student_ID
AND                 b.Lesson_Code = c.Lesson_Code
AND                 c.Lecturer_ID = d.Lecturer_ID
AND                 a.Student_ID = [Student_ID]
ORDER BY            Lesson_Code;
```

Description	To help student check their lecturers' contact methods.

```
SELECT      Lesson_code, mid(Lesson_code,4,1) as Lesson_Level,
            Lesson_Categories,Lesson_Medium
FROM        lesson
WHERE       Lesson_code=[Lesson_code]
ORDER BY    Lesson_Code;
```

Description	Find out the lesson information.

```
SELECT      mid(Lesson_code,4,1) as Lesson_Level, Count(*) as
Number_of_Lesson
FROM        student a, enroll_of_lesson b
WHERE       a.Student_ID = b. Student_ID
AND         a.Student_ID = [Student_ID]
GROUP BY    mid(Lesson_code,4,1);
```

Description	To count the number of lessons of different levels the student has taken.

```
SELECT      a.Lesson_Code, a.Lesson_Name, b.Lecturer_LastName,
            b.Lecturer_FirstName
FROM lesson a, lecturer b
WHERE       a.Lecturer_ID = b.Lecturer_ID
AND         a.Lesson_Code = [Lesson_Code];
```

Description	This is a query to show that which lesson is being taught by lecturer.

b. Form

```
INSERT INTO enroll_of_lesson
(Lesson_Code, Student_ID)
VALUE
([Lesson_Code], [Student_ID]);
```

Description	This is a query for the student to enroll specific lesson.

c.Report

It is a report to show the clubs that the students have enrolled and their positions.

Student ID	Student Name	Club Name	Position
12333333A	Xu Jian Liang	Band Society	Chairperson
12333444A	Li Ki Shun	Computer Society	Secretary
12333589A	Lin Yu Yun	Hiking Society	Chairperson
12345200A	Chan Tai Man	Dance Society	Secretary
12345678A	Bao Jia Hong	Band Society	Member
12356666A	Lo Zhao Lian	Accounting and Finance Society	Member
12385468A	Chan Bai Ting	Christian Fellowship	Chairperson
12400000A	Zhang Yi Jun	English Society	Chairperson
12443399A	Zhu Shu Ting	Badminton Society	Member
12548486A	Lin Pei Xuan	Film Appreciation Club	Chairperson
12555555A	Lin Zong Lin	Hong Kong Award for Young Peop	Chairperson
12584893A	Kwok Shing Ping	Drama Society	Chairperson
12584893A	Kwok Shing Ping	Music Society	Chairperson
12587469A	Chan Wan Qang	Film Appreciation Club	Member
12587469A	Chan Wan Qang	Photography Society	Chairperson
12600001A	Ma Ya Ju	Film Appreciation Club	Treasurer
12600001A	Ma Ya Ju	Practical Entrepreneur Society	Chairperson

```
SELECT      a.Student_ID, a.Student_Name, c.Club_Name, b.Position
FROM        student a, enroll_of_club b, club c
WHERE       a.Student_ID = b.Student_ID
AND         b.Club_ID = c.Club_ID
GROUP BY    a.Student_ID, Student_Name, Club_Name, Position
ORDER BY    a.Student_ID;
```

Description	To show the clubs that the students have enrolled and their positions.

It is a report to show the courses that the students are studying

Student ID	Student Name	Lesson Code	Lesson Name
12313032A	Chan Hin Hei	CCN1002	Practical English for College Students
12313032A	Chan Hin Hei	CCN1111	General Chemistry II
12333333A	Xu Jian Liang	CCN1003	Chinese Communication for College Students
12333333A	Xu Jian Liang	CCN3140	Programming Project
12333444A	Li Ki Shun	CCN1003	Chinese Communication for College Students
12333444A	Li Ki Shun	CCN1007	Information Technology for Business
12333589A	Lin Yu Yun	CCN1049	Physics I
12345200A	Chan Tai Man	CCN1001	Elementary Chinese
12345200A	Chan Tai Man	CCN1051	Physics II
12345678A	Bao Jia Hong	CCN1001	Elementary Chinese
12345678A	Bao Jia Hong	CCN1109	General Biology
12356666A	Lo Zhao Lian	CCN1110	General Chemistry I
12385468A	Chan Bai Ting	CCN1111	General Chemistry II
12400000A	Zhang Yi Jun	CCN1003	Chinese Communication for College Students
12400000A	Zhang Yi Jun	CCN2002	Introduction to Economics
12443399A	Zhu Shu Ting	CCN1003	Chinese Communication for College Students
12443399A	Zhu Shu Ting	CCN2003	Introduction to Marketing
12548486A	Lin Pei Xuan	CCN2004	Managing Organisations
12548932A	Pan Hong Han	CCN1007	Information Technology for Business
12548932A	Pan Hong Han	CCN2006	Understanding Globalisation
12555555A	Lin Zong Lin	CCN1049	Physics I
12555555A	Lin Zong Lin	CCN2101	Financial Accounting

```
SELECT      a.Student_ID, a.Student_Name, b.Lesson_Code,
            c.Lesson_Name
FROM        student a, enroll_of_lesson b, lesson c
WHERE       a.Student_ID = b.Student_ID
AND         b.Lesson_Code = c.Lesson_Code
GROUP BY    a.Student_ID, Student_Name, b.Lesson_Code,
            Lesson_Name, Grade
ORDER BY    a.Student_ID;
```

Description	To show the courses that the students are studying.

5. Club Admin

a. Query

```
SELECT      Room_ID
FROM        room
WHERE       Room_ID = [Room_ID]
```

Description	This is a query to check whether user input a valid room id or not.

```
SELECT      Club_ID
FROM        club
WHERE       Club_ID = [Club_id];
```

Description	This is a query to check whether user input a valid club_id or not.

```
SELECT      Room_ID , Booking_Date ,
            Booking_STime, Booking_ETime
FROM        booking
WHERE       Room_ID = [Room_ID]
AND         Booking_Date = [Booking_Date]
AND         Booking_STime
BETWEEN     [Booking_Start_Time]
AND         [Booking_End_Time];
```

Description	This is a query to check the room avabiablity of the user input

```
SELECT      *
FROM        booking
WHERE       Club_ID IS NOT NULL
ORDER BY    Booking_Date , Room_ID ,
            Booking_STime
```

Description	This is a query to show all the booking booked by club.

```
SELECT      Student_ID
FROM        student
WHERE       Student_ID = [Student_No];
```

Description	This is a query to check whether user input a valid Student id or not.

```
SELECT      Club_ID
FROM        club
WHERE       Club_ID = [Club_ID];
```

Description	This is a query to check whether user input a valid Club id or not.

```
SELECT      *
FROM        enroll_of_club
WHERE       Club_ID = [Club_ID]
ORDER BY    Enroll_date, Student_ID
```

Description	This is a query to show the information of member of that club

b. Form

```
INSERT INTO booking
(Room_ID , Booking_Date , Booking_STime, Booking_ETime,
Club_ID , Booking_Purpose`)
VALUE
([Room_ID],[Booking_Date],[Booking_Start_Time],[Booking_
End_Time],[Club_ID], [Booking_Purpose]);
```

Description	CREATE a new row for Booking to save the booking into database

```
UPDATE booking
SET Room_ID = [Room_ID]
AND Booking_Date = [Booking_Date]
AND Booking_STime = [Booking_Start_Time]
AND Booking_ETime = [Booking_End_Time]
AND Club_ID = [Club_ID]
AND Booking_Purpose = [Booking_Purpose]
WHERE Booking_ID = [Booking_ID];
```

Description	Update the information of the booking if the user input the change

```
DELETE FROM booking
WHERE Booking_ID = [Booking_ID];
```

Description	Delete the booking

```
INSERT INTO enroll_of_club
(Club_ID, Student_ID, Position, Enroll_date)
VALUE
([Club_ID],[Student_no],'Member',date('Y-m-d'));
```

Description	Insert a new row for a new member of the club. date() is a function to get today date.

```
UPDATE enroll_of_club
SET Position = [Position]
WHERE Student_ID = [Student_ID]
AND Club_ID = [Club_ID];
```

Description	Update the position of the member if the user input the change

```
DELETE FROM enroll_of_club
WHERE Club_ID = [Club_ID]
AND Student_ID = [Student_ID];
```

Description	Delete the member of the club.

c. Report

This is a report to show the personal and contact information of each club's chairperson.

Club Name	Student ID	Student Name	Student Sex	Student Contact No	Student Position
Accounting and Finance Society	12660032A	So Mau Chiu	M	91109234	Chairperson
Badminton Society	12670000A	Lin Zuo Ci	F	65412879	Chairperson
Band Society	12333333A	Xu Jian Liang	F	94512036	Chairperson
Christian Fellowship	12385468A	Chan Bai Ting	M	54654131	Chairperson
Computer Society	12607532A	Chan Kwan Wing	M	93759493	Chairperson
Dance Society	12648456A	Yang Wen Xin	M	95468888	Chairperson
Drama Society	12584893A	Kwok Shing Ping	M	99944433	Chairperson
English Society	12400000A	Zhang Yi Jun	M	96321458	Chairperson
Film Appreciation Club	12548486A	Lin Pei Xuan	F	56983255	Chairperson
Hiking Society	12333589A	Lin Yu Yun	F	55662233	Chairperson

```
SELECT      c.Student_ID AS SID,
            b.student_name AS Sname,
            b.student_sex AS Ssex,
            b.student_age AS Sage,
            b.Contact_no AS Scno,
            c.Position AS SP,
            c.Enroll_Date AS SED,
            a.Club_Name AS CName
FROM        club a, student b, enroll_of_club c
WHERE       c.Student_ID=b.Student_ID
AND         b.club_ID=a.club_ID
AND         c.position = 'Chairperson'
ORDER BY    club.Club_ID
```

Description A report to show all the personal information of the club's chairperson

This is a report to show the number of members in each club.

Club Name	Numer of member
Volleyball Society	1
Ving Tsun Society	1
Taekwondo Society	1
Social Service Society	1
Rotaract Club	1
Putonghua Society	1
Psychology Society	1
Practical Entrepreneur Society	1
Photography Society	1
Music Society	1
Hong Kong Award for Young Peop	1
English Society	1

```
SELECT      club.club_name AS club_name ,
            COUNT(enroll_of_club.club_id) AS num
FROM        club, enroll_of_club
WHERE       club.club_ID=enroll_of_club.club_id
GROUP BY    club.club_name
ORDER BY    COUNT(enroll_of_club.club_id),
            club.club_name
```

Description a report to show the number of members in each club

5. Physical design

5.1 Software used

As our user need to access the database in anywhere and at any time, we design to use a website as our interface part of our database.

In the interface part, we mainly use two language, which is Hyper Text Markup Language(HTML) and Hypertext Preprocessor (PHP). We also use some JavaScript and CSS(Cascading Style Sheets) to coordinate our design of webpage. Our code editor is Dreamweaver. We use it to edit our website.

In the Back Stage, we choose to use MYSQL as our database management system because this software allows many user access in the same time and many famous website, such as Wikipedia and Google use it.

We try to publish our webpage to the web. We find out some difficulties to use a free web hosting service because it is not stable and it didn't give us a full control of the server. At the end, we find out that we can build a Apache HTTP server in the Windows. This is a free and open-source software. We also use AppServ to config all the setting automatically.

5.2 Source of data

In order to ensure the accuracy of the system, the data we used mainly come from HKCC. This makes our data more resemble to the real situation.

For the information of rooms, clubs, lessons and lecturers, we use the data from the website of HKCC.

For the information of students and clubs, since we don't have those source of data from the website of HKCC, they are originally created by us.

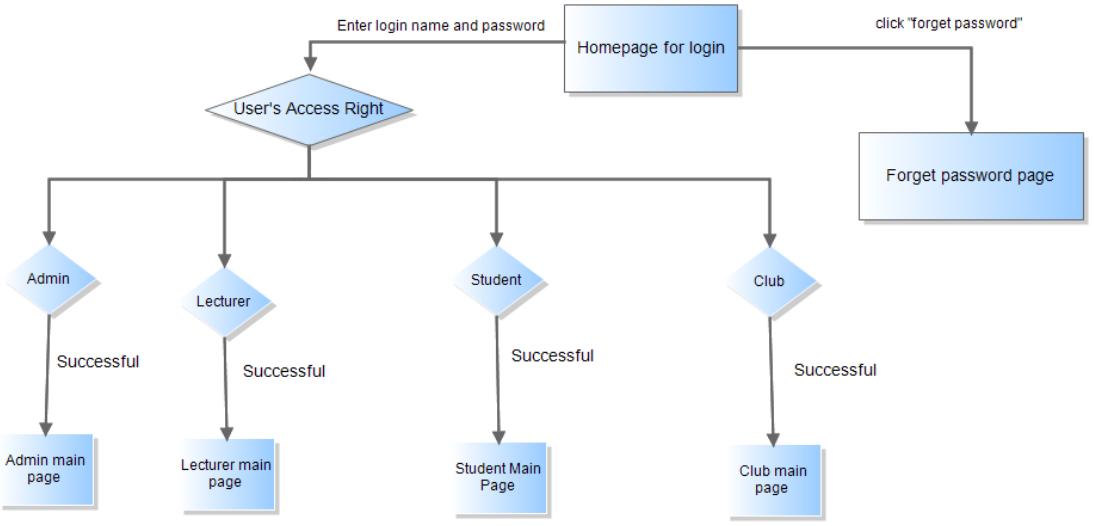
Lesson_Code	Lesson_Name	Lesson_Division	Lesson_Categories	Lesson_Medium	Lecturer_ID
CCN1001	Elementary Chinese	Language	GE	Chinese	60
CCN1002	Practical English for College Students	Language	GE	English	61
CCN1003	Chinese Communication for College Students	Language	GE	Chinese	60
CCN1007	Information Technology for Business	Business	GE	English	50
CCN1049	Physics I	Science	GE	English	30
CCN1051	Physics II	Science	DS	English	35
CCN1109	General Biology	Science	DS	English	21
CCN1110	General Chemistry I	Science	DS	English	19
CCN1111	General Chemistry II	Science	DS	English	1
CCN2002	Introduction to Economics	Business	GE	English	56
CCN2003	Introduction to Marketing	Business	GE	English	57
CCN2004	Managing Organisations	Business	DS	English	58

Student_ID	Student_name	Student_Sex	Student_Age	Contact_No
12300000A	Lin Yu Meng	M	19	66845698
12313032A	Chan Hin Hei	M	18	91552870
12333333A	Xu Jian Liang	F	18	94512036
12333444A	Li Ki Shun	M	20	92394949
12333589A	Lin Yu Yun	F	18	55662233
12345200A	Chan Tai Man	M	18	54320092
12345678A	Bao Jia Hong	M	18	54123698
12356666A	Lo Zhao Lian	F	19	99556644

Club_ID	Club_Name	Club_StartDate
1	Accounting and Finance Society	2003-01-01
2	Badminton Society	2005-01-30
3	Band Society	2005-01-25
4	Christian Fellowship	2005-02-22
5	Computer Society	2012-12-15
6	Dance Society	2009-12-25
7	Drama Society	2013-09-01
8	English Society	2013-09-01
9	Film Appreciation Club	2010-09-01
10	Hiking Society	2009-04-05

6. Implementation detail

6.1 Flow chats of the program



This is our database flowchart. User can enter their own login name and password to login ,also, if they forget password, they can go to forget password page. When they login to the system successfully, they can use query, form and report.

6.2 Sample queries and result with explanation

1.School Admin:

Select	d.Student_ID, d.Student_Name
From	lecturer a, lesson b, enroll_of_lesson c, student d
where	a.Lecturer_ID= b.Lecturer_ID
and	b.Lesson_code= c.Lesson_code
and	c.Student_ID= d.Student_ID
and	a.Lecturer_ID= 60
Order by	Student_ID;

We try to find out the students that are taught by specific Lecturer_ID=60 (Choi Wai-yuk).
The result show that 5 students are taught by lecturer (Choi Wai-yuk).

Showing rows 0 - 29 (49 total, Query took 0.0072 sec)

SQL query:

```
SELECT *
FROM lecturer
WHERE Lecturer_Position = (

  SELECT Lecturer_Position
  FROM lecturer
  WHERE Lecturer_ID =16
);

LIMIT 0 , 30
```

Profilin

Lecturer_ID	Lecturer_LastName	Lecturer_FirstName	Lecturer_Position	Lecturer_Email	Lecturer_Telephone	Division
10	CHAN	Yik Man	Lecturer	ccckm@hkcc-polyu.edu.hk	37460133	Science & Technology
11	CHAN	Wai Heung	Lecturer	cccwh@hkcc-polyu.edu.hk	37460134	Science & Technology
12	CHAN	Chun Man	Lecturer	ccccm@hkcc-polyu.edu.hk	37460135	Science & Technology
13	CHAN	Hoi Huen	Lecturer	ccchh@hkcc-polyu.edu.hk	37460136	Science & Technology
14	CHAN	Oi Wan	Lecturer	cccow@hkcc-polyu.edu.hk	37460137	Science & Technology
15	CHAN	Man Yee	Lecturer	cccmy@hkcc-polyu.edu.hk	37460138	Science & Technology
16	CHAN	Pik Wah	Lecturer	cccpw@hkcc-polyu.edu.hk	37460139	Science & Technology
17	CHAN	Kim Ming	Lecturer	ccckm@hkcc-polyu.edu.hk	37460140	Science & Technology
18	CHAN	Fung Ming	Lecturer	cccfm@hkcc-polyu.edu.hk	37460141	Science & Technology
19	CHEUNG	Hing Keung	Lecturer	ccchk@hkcc-polyu.edu.hk	37460142	Science & Technology
20	CHEUNG	Wing Yin	Lecturer	cccwy@hkcc-polyu.edu.hk	37460143	Science & Technology
21	CHIU	Hon Sun	Lecturer	ccchs@hkcc-polyu.edu.hk	37460144	Science & Technology
22	CHIU	Wang Kin	Lecturer	cccwk@hkcc-polyu.edu.hk	37460145	Science & Technology
23	CHOW	Kin Keung	Lecturer	ccckk@hkcc-polyu.edu.hk	37460146	Science & Technology
24	CHU	Yiu Chung	Lecturer	cccyc@hkcc-polyu.edu.hk	37460147	Science & Technology
25	CHUNG	Ying Ming	Lecturer	cccym@hkcc-polyu.edu.hk	37460148	Science & Technology
26	HO	Wai Tung	Lecturer	cchwt@hkcc-polyu.edu.hk	37460149	Science & Technology
27	KWAN	Ka Ying	Lecturer	cckky@hkcc-polyu.edu.hk	37460150	Science & Technology
28	KWAN	Chun Kit	Lecturer	cckck@hkcc-polyu.edu.hk	37460151	Science & Technology
29	LEE	Yik Sze	Lecturer	cclyz@hkcc-polyu.edu.hk	37460152	Science & Technology
30	LEUNG	Lai Fun	Lecturer	ccllf@hkcc-polyu.edu.hk	37460153	Science & Technology
31	LO	Ching On	Lecturer	cclco@hkcc-polyu.edu.hk	37460154	Science & Technology
32	LUI	Wing Shing	Lecturer	cclws@hkcc-polyu.edu.hk	37460155	Science & Technology
33	MAK	Kai Long	Lecturer	ccmkl@hkcc-polyu.edu.hk	37460156	Science & Technology
34	MOK	Hing Tung	Lecturer	ccmht@hkcc-polyu.edu.hk	37460157	Science & Technology
35	NG	Ka Wai	Lecturer	ccnkw@hkcc-polyu.edu.hk	37460158	Science & Technology
36	NG	Pin	Lecturer	ccnp@hkcc-polyu.edu.hk	37460159	Science & Technology
37	NG	Wing Man	Lecturer	ccnwm@hkcc-polyu.edu.hk	37460160	Science & Technology
38	NG	Sun Pui	Lecturer	ccnsp@hkcc-polyu.edu.hk	37460161	Science & Technology
39	NGAN	Chung Wai	Lecturer	ccncw@hkcc-polyu.edu.hk	37460162	Science & Technology

```
Select  *
from    lecturer
where   Lecturer_Position =
(Select         Lecturer_Position
from    lecturer
where   Lecturer_ID= 16 );
```

We try to find out the lecturers' information who have the same position as Lecturer_ID=16 (Chan Pik Wah).
As Pat Chan's position is lecturer, the results show that there are 49 lecturer where the position is same with Pat Chan.

Showing rows 0 - 29 (34 total, Query took 0.0331 sec)

SQL query:

```
SELECT a.Lesson_code, a.Lesson_Name, COUNT( * ) AS Number_of_student
FROM lesson a, enroll_of_lesson b, student c
WHERE a.Lesson_Code = b.Lesson_Code
AND b.Student_ID = c.Student_ID
GROUP BY a.Lesson_code, a.Lesson_Name
ORDER BY a.Lesson_code DESC
LIMIT 0 , 30
```

«T»

Lesson_code	Lesson_Name	Number_of_student
CCN3140	Programming Project	2
CCN2273	Operating Systems	2
CCN2242	Object Oriented Programming	1
CCN2241	Discrete Structures	1
CCN2240	Database Systems	1
CCN2239	Data Structures	1
CCN2231	General Biochemistry	3
CCN2132	Principles of Investments	1
CCN2131	Marketing Research Fundamentals	1
CCN2130	Marketing in China	1
CCN2129	Introduction to Macroeconomics	1
CCN2126	Introduction to International Business	1
CCN2123	Introduction to Auditing and Taxation	1
CCN2122	International Finance	1
CCN2121	Intermediate Accounting	1
CCN2113	Financial Management	1
CCN2111	Cost Accounting	1
CCN2108	Business Law	1
CCN2106	Business Environment in China	1
CCN2105	Business Economics	1
CCN2101	Financial Accounting	1
CCN2006	Understanding Globalisation	1
CCN2004	Managing Organisations	1
CCN2003	Introduction to Marketing	2
CCN2002	Introduction to Economics	1
CCN1111	General Chemistry II	2
CCN1110	General Chemistry I	2
CCN1109	General Biology	2
CCN1051	Physics II	2
CCN1049	Physics I	2

Select	a.Lesson_code, a.Lesson_Name ,count(*) as Number_of_student
from	lesson a, enroll_of_lesson b, student c
where	a.Lesson_Code = b.Lesson_Code
and	b.Student_ID = c.Student_ID
group by	a.Lesson_code, a.Lesson_Name
order by	a.Lesson_code desc

This is a report. We try to find out how many students in each lesson.

2.Lecturer

```
SELECT *
FROM booking
WHERE Room_ID =1203;

LIMIT 0 , 30
```

Booking_ID	Room_ID	Booking_Date	Booking_STime	Booking_ETime	Lecturer_ID	Club_ID	Booking_Purpose
42	1203	2013-11-27	18:30:00	20:00:00	0	8	Actitivies

```
SELECT      *
FROM        booking
WHERE       Room_ID = 1203;
```

We try to find out the information of room 1203.The result is listed above.

```
SELECT Lesson_Code, Lesson_Name, Lesson_Division
FROM lesson
WHERE Lesson_Division = 'IT'
LIMIT 0 , 30
```

«T»	Lesson_Code	Lesson_Name	Lesson_Division
	CCN2238	Computer Networking	IT
	CCN2239	Data Structures	IT
	CCN2240	Database Systems	IT
	CCN2241	Discrete Structures	IT
	CCN2242	Object Oriented Programming	IT
	CCN2264	Computer Organisation	IT
	CCN2265	E-Business	IT
	CCN2273	Operating Systems	IT
	CCN3133	Computer System Principles	IT
	CCN3140	Programming Project	IT
	CCN3143	Software Engineering	IT

```
SELECT      Lesson_Code, Lesson_Name,
            Lesson_Division
FROM        lesson
WHERE       Lesson_Division = 'IT';
```

This query can show the lesson name by lesson division.
For example, we can get the lesson which belongs to IT by the query.

```
SELECT Booking_Date, COUNT( * ) AS 'Frequency of Booking Date'
FROM booking
GROUP BY Booking_Date
LIMIT 0 , 30
```

Profiling [編輯]

Booking_Date	Frequency of Booking Date
2013-11-20	3
2013-11-21	1
2013-11-25	4
2013-11-26	4
2013-11-27	12
2013-11-28	3
2013-11-29	5
2013-11-30	1
2013-12-02	4
2013-12-03	1
2013-12-04	2
2013-12-05	3
2013-12-06	1
2013-12-07	1
2013-12-09	2
2013-12-10	2
2013-12-11	1
2013-12-12	2
2013-12-13	1
2013-12-16	1
2013-12-18	1
2013-12-20	2
2013-12-31	1
2014-01-31	1

```
SELECT      Booking_Date,
            COUNT(*) AS 'Frequency of Booking Date'
FROM        booking
GROUP BY    Booking_Date;
```

This is a report. We try to find out the frequency of using the lecturer room in each day.

3.Student

```
SELECT c.Lesson_Code, c.Lesson_Name, d.Lecturer_LastName, d.Lecturer_FirstName, d.Lecturer_Email, d.Lecturer_Telephone
FROM student a, enroll_of_lesson b, lesson c, lecturer d
WHERE a.Student_ID = b.Student_ID
AND b.Lesson_Code = c.Lesson_Code
AND c.Lecturer_ID = d.Lecturer_ID
AND a.Student_ID = '12660032A'
ORDER BY Lesson_Code;

LIMIT 0 , 30
```

«T»

Lesson_Code	Lesson_Name	Lecturer_LastName	Lecturer_FirstName	Lecturer_Email	Lecturer_Telephone
CCN1003	Chinese Communication for College Students	CHOI	Wai-yuk	wychoi@hkcc-polyu.edu.hk	37460645
CCN2003	Introduction to Marketing	LAU	Kin-wai	ccdanny@hkcc-polyu.edu.hk	37460248
CCN2129	Introduction to Macroeconomics	WOO, Arison	Suk-ching	ccarison@hkcc-polyu.edu.hk	37460618
CCN2231	General Biochemistry	WONG	Yiu Tong	ccwyt@hkcc-polyu.edu.hk	37460131
CCN2239	Data Structures	WOO	Kin Sang	ccwks@hkcc-polyu.edu.hk	37460132
CCN2240	Database Systems	CHAN	Pik Wah	cccpw@hkcc-polyu.edu.hk	37460139
CCN2241	Discrete Structures	WONG	Kwong Cheong	ccwkc@hkcc-polyu.edu.hk	37460166
CCN2242	Object Oriented Programming	NG	Pin	ccnp@hkcc-polyu.edu.hk	37460159

```
SELECT      c.Lesson_Code, c.Lesson_Name, d. Lecturer_LastName,
            d.Lecturer_FirstName,
            d.Lecturer_Email, d.Lecturer_Telephone
FROM        student a, enroll_of_lesson b, lesson c, lecturer d
WHERE       a.Student_ID = b.Student_ID
AND         b.Lesson_Code = c.Lesson_Code
AND         c.Lecturer_ID = d.Lecturer_ID
AND         a.Student_ID = 12660032A;
ORDER BY    Lesson_Code;
```

We try to find out the lessons enrolled by specific Student_ID = 12660032A (So Man Chiu).
This result shows that 8 lessons is enrolled by the student (So Man Chiu).

```
SELECT a.Student_Name, b.Club_Name, c.Position
FROM student a, club b, enroll_of_club c
WHERE a.Student_ID = c.Student_ID
AND b.Club_ID = c.Club_ID
AND a.Student_ID = '12660032A';

LIMIT 0 , 30
```

«T»

Student_Name	Club_Name	Position
So Mau Chiu	Accounting and Finance Society	Chairperson
So Mau Chiu	Volleyball Society	Chairperson

```
SELECT   a.Student_Name, b.Club_Name, c.Position
FROM     student a, club b, enroll_of_club c
WHERE    a.Student_ID = c.Student_ID
AND      b.Club_ID = c.Club_ID
AND      a.Student_ID = 12660032A;
```

We try to find out the clubs enrolled by specific Student_ID = 12660032A (So Man Chiu).
This result shows that 2 clubs is enrolled by the student (So Man Chiu).

```
SELECT a.Student_ID, a.Student_Name, c.Club_Name, b.Position
FROM student a, enroll_of_club b, club c
WHERE a.Student_ID = b.Student_ID
AND b.Club_ID = c.Club_ID
GROUP BY a.Student_ID, Student_Name, Club_Name, Position
ORDER BY a.Student_ID;

LIMIT 0 , 30
```

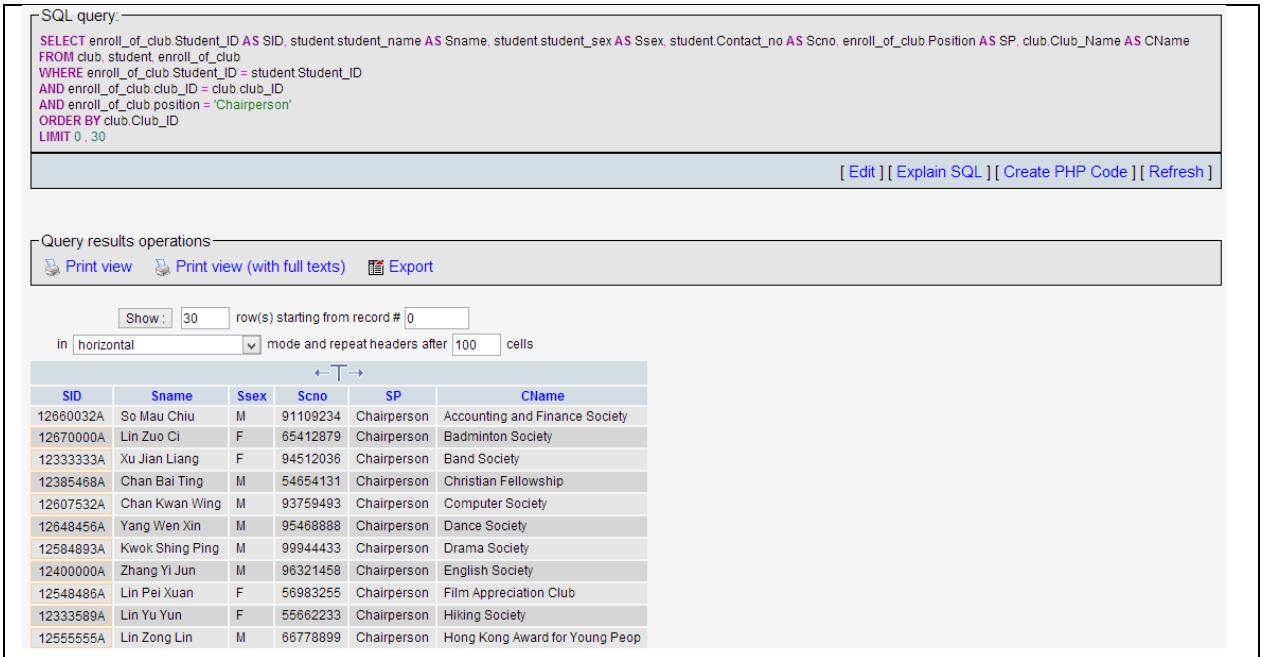
«T»

Student_ID	Student_Name	Club_Name	Position
12300000A	Lin Yu Meng	Drama Society	Member
12333333A	Xu Jian Liang	Band Society	Chairperson
12333444A	Li Ki Shun	Computer Society	Secretarys
12333589A	Lin Yu Yun	Hiking Society	Chairperson
12345200A	Chan Tai Man	Dance Society	Secretary
12345678A	Bao Jia Hong	Band Society	Member
12356666A	Lo Zhao Lian	Accounting and Finance Society	Member
12385468A	Chan Bai Ting	Christian Fellowship	Chairperson
12400000A	Zhang Yi Jun	English Society	Chairperson
12443399A	Zhu Shu Ting	Badminton Society	Member

```
SELECT   a.Student_ID, a.Student_Name, c.Club_Name, b.Position
FROM     student a, enroll_of_club b, club c
WHERE    a.Student_ID = b.Student_ID
AND      b.Club_ID = c.Club_ID
GROUP BY a.Student_ID, Student_Name, Club_Name, Position
ORDER BY a.Student_ID;
```

This is a report. We find out how which clubs are enrolled by each student.

4.Club admin



SQL query:

```
SELECT enroll_of_club.Student_ID AS SID, student.student_name AS Sname, student.student_sex AS Ssex, student.Contact_no AS Scno, enroll_of_club.Position AS SP, club.Club_Name AS CName
FROM club, student, enroll_of_club
WHERE enroll_of_club.Student_ID = student.Student_ID
AND enroll_of_club.club_ID = club.club_ID
AND enroll_of_club.position = 'Chairperson'
ORDER BY club.Club_ID
LIMIT 0 , 30
```

[Edit] [Explain SQL] [Create PHP Code] [Refresh]

Query results operations

Print view | Print view (with full texts) | Export

Show : 30 row(s) starting from record # 0

in horizontal mode and repeat headers after 100 cells

SID	Sname	Ssex	Scno	SP	CName
12660032A	So Mau Chiu	M	91109234	Chairperson	Accounting and Finance Society
12670000A	Lin Zuo Ci	F	65412879	Chairperson	Badminton Society
12333333A	Xu Jian Liang	F	94512036	Chairperson	Band Society
12385468A	Chan Bai Ting	M	54654131	Chairperson	Christian Fellowship
12607532A	Chan Kwan Wing	M	93759493	Chairperson	Computer Society
12648456A	Yang Wen Xin	M	95468888	Chairperson	Dance Society
12584893A	Kwok Shing Ping	M	99944433	Chairperson	Drama Society
12400000A	Zhang Yi Jun	M	96321458	Chairperson	English Society
12548486A	Lin Pei Xuan	F	56983255	Chairperson	Film Appreciation Club
12333589A	Lin Yu Yun	F	55662233	Chairperson	Hiking Society
12555555A	Lin Zong Lin	M	66778899	Chairperson	Hong Kong Award for Young Peop

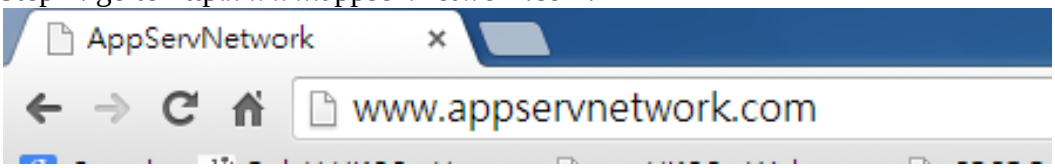
SELECT	enroll_of_club.Student_ID AS SID, student.student_name AS Sname, student.student_sex AS Ssex, student.Contact_no AS Scno, enroll_of_club.Position AS SP, club.Club_Name AS CName
FROM	club, student, enroll_of_club
WHERE	enroll_of_club.Student_ID=student.Student_ID
AND	enroll_of_club.club_ID=club.club_ID
AND	enroll_of_club.position = 'Chairperson'
ORDER BY	club.Club_ID

This is a report to show all the club's chairperson's information.

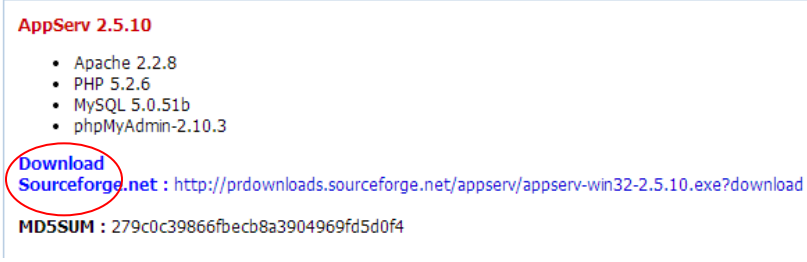
7. Usage

7.1 Setting up the database

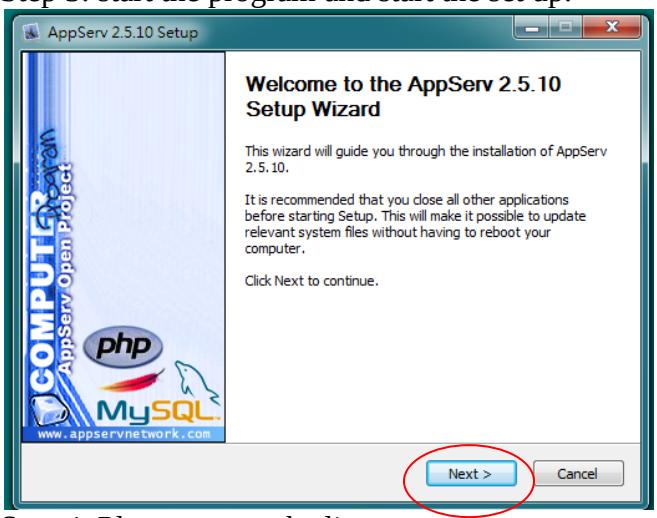
Step 1: go to http://www.appservnetwork.com.



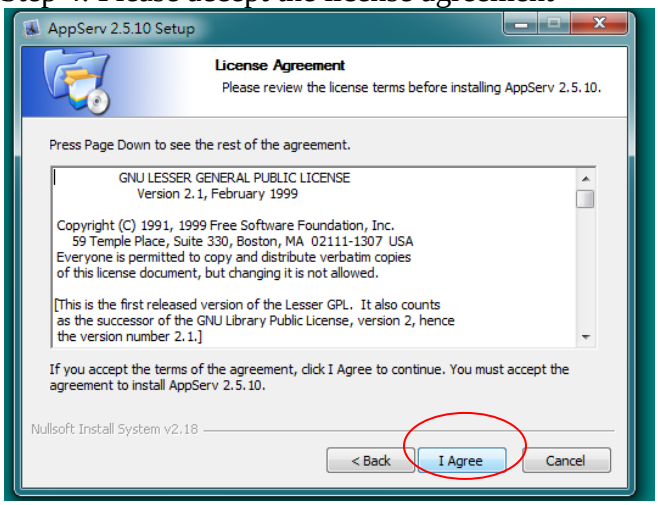
Step 2: download AppServ 2.5.10 (the Setup program has been include in the CD/DVD).



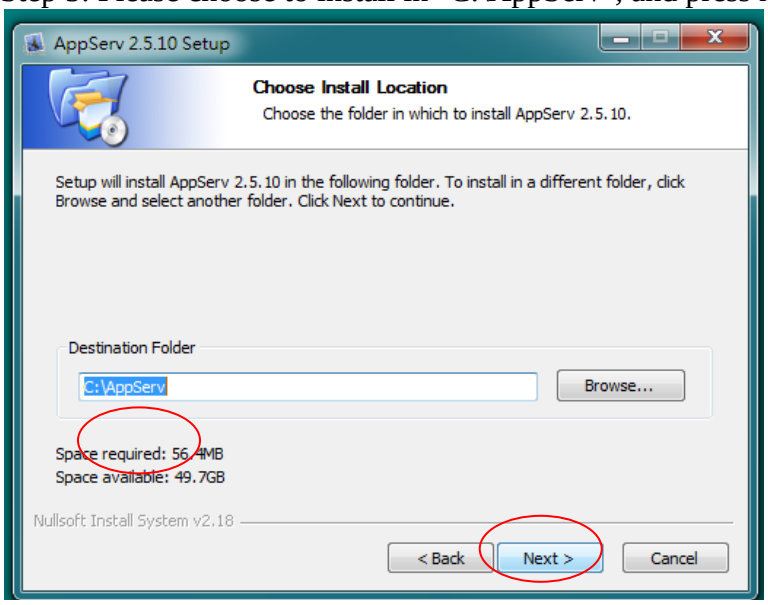
Step 3: start the program and start the set up.



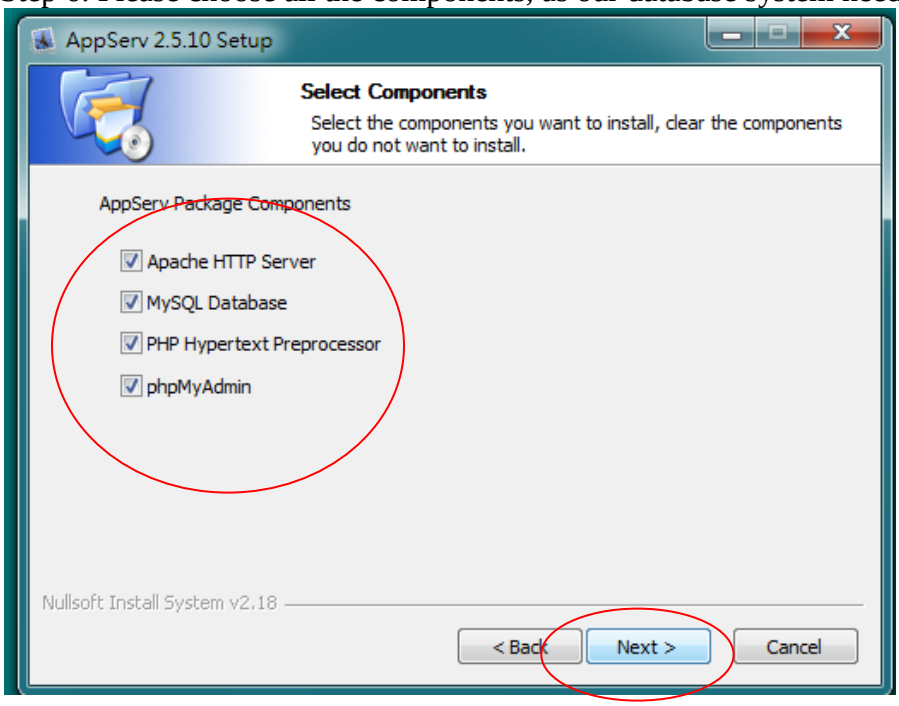
Step 4: Please accept the license agreement



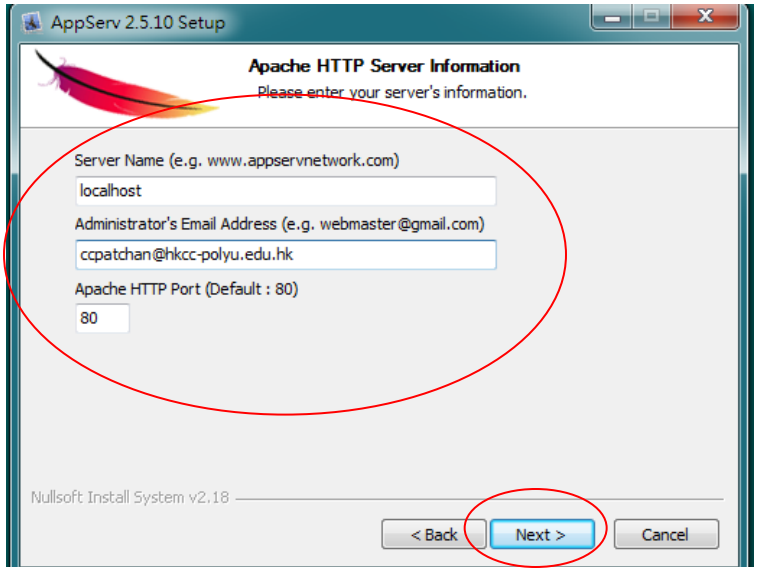
Step 5: Please choose to install in "C:\AppServ", and press next



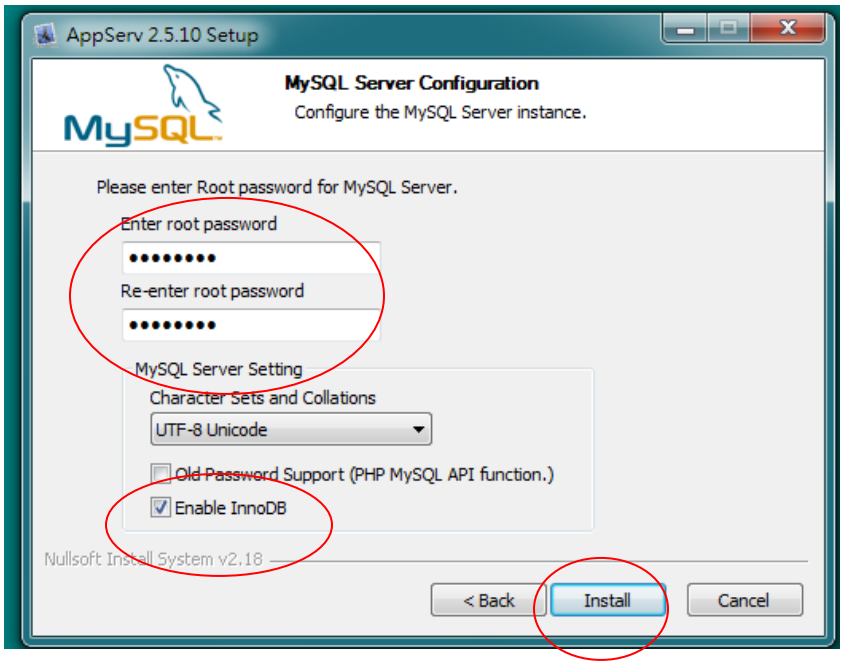
Step 6: Please choose all the components, as our database system needs to use all, and press next.



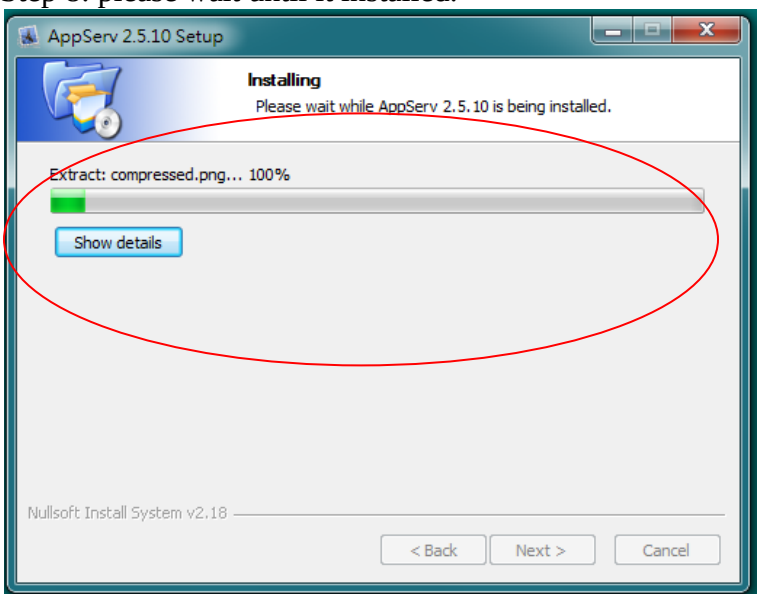
Step 7: if you use it as a local server, please enter "localhost" in Server name filed, otherwise please enter your server domain name. Also, please enter your own email address in Administrator's email Address filed. Please check the Apache HTTP Port is 80, the default one, if there are no special reason, please don't change it. Press "next".



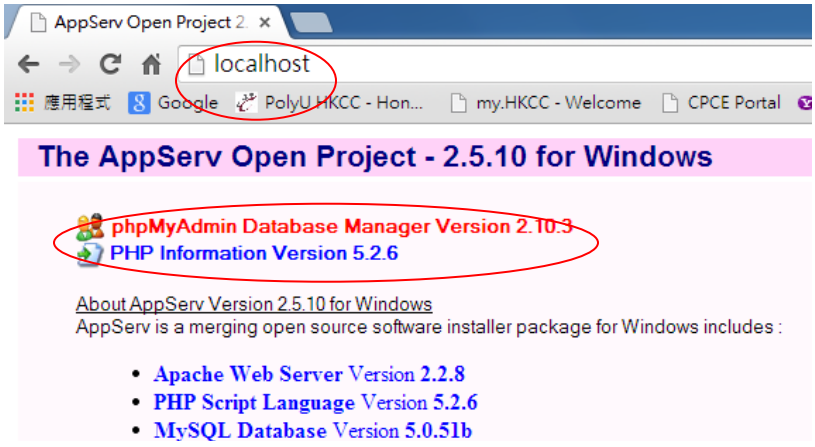
Step 7: please enter the password to protect the database. Also, please click enable "InnoDB" . Press "install".



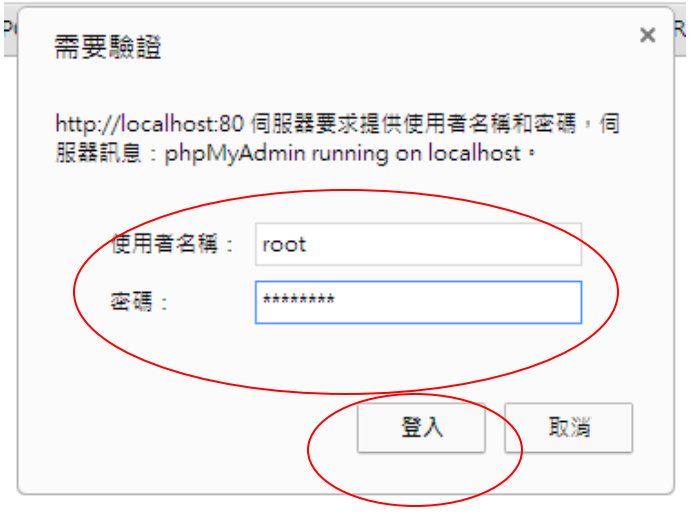
Step 8: please wait until it installed.



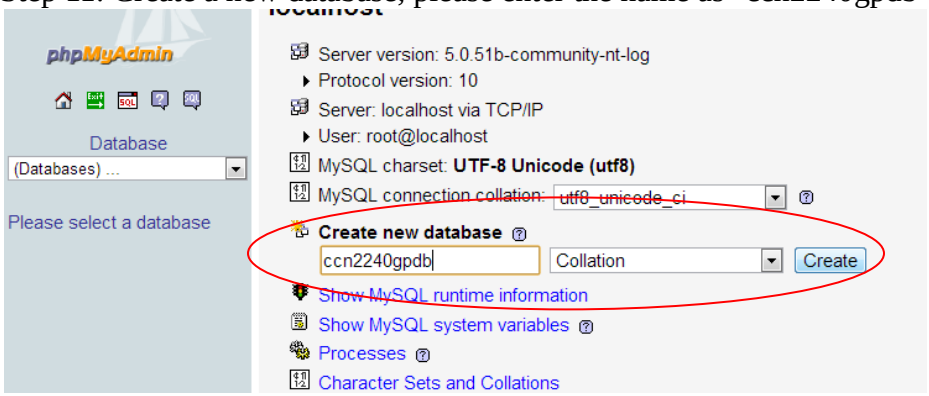
Step 9: Open a browser and go to http://localhost , and choose "phpMyAdmin Database Manager version 2.10.3".



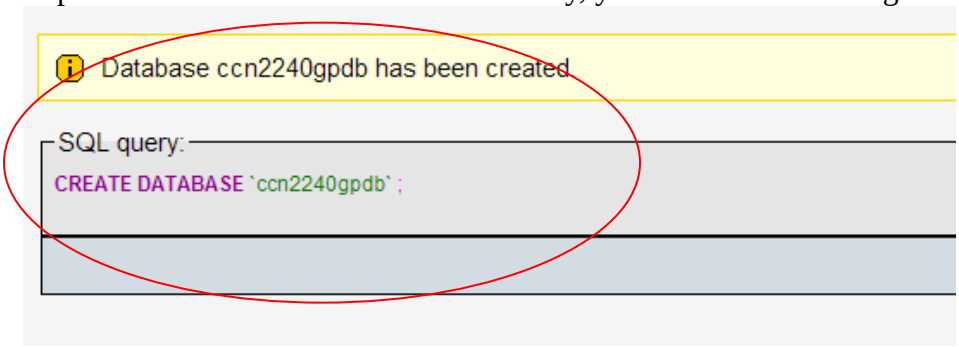
Step 10: Please enter the login name as "root" and the password you set in step 7. After that, press login



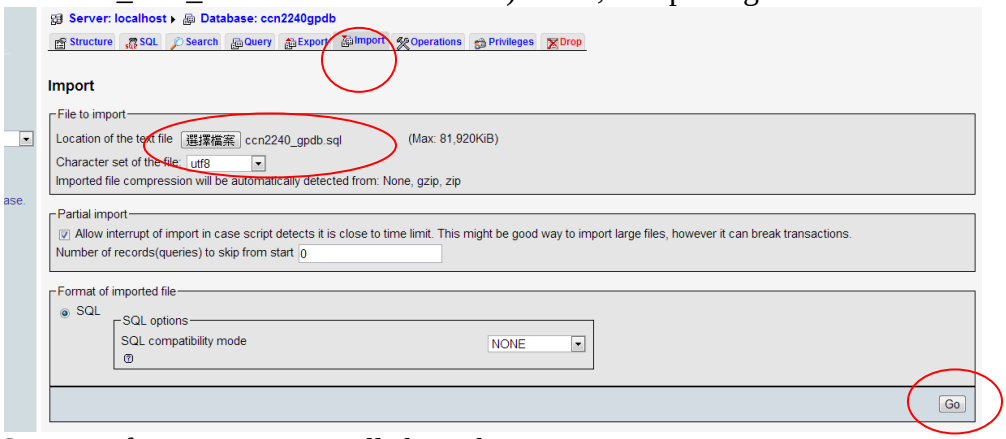
Step 11: Create a new database, please enter the name as "ccn2240gpdb" and press create.



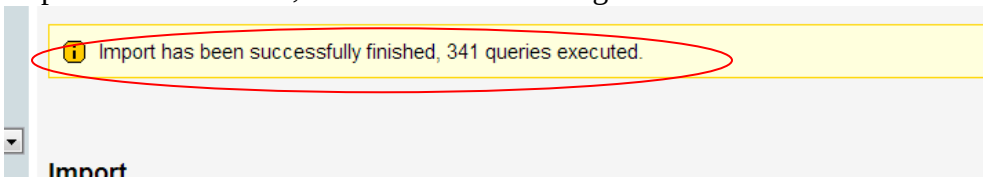
Step 12: If the database created successfully, you will see the message.



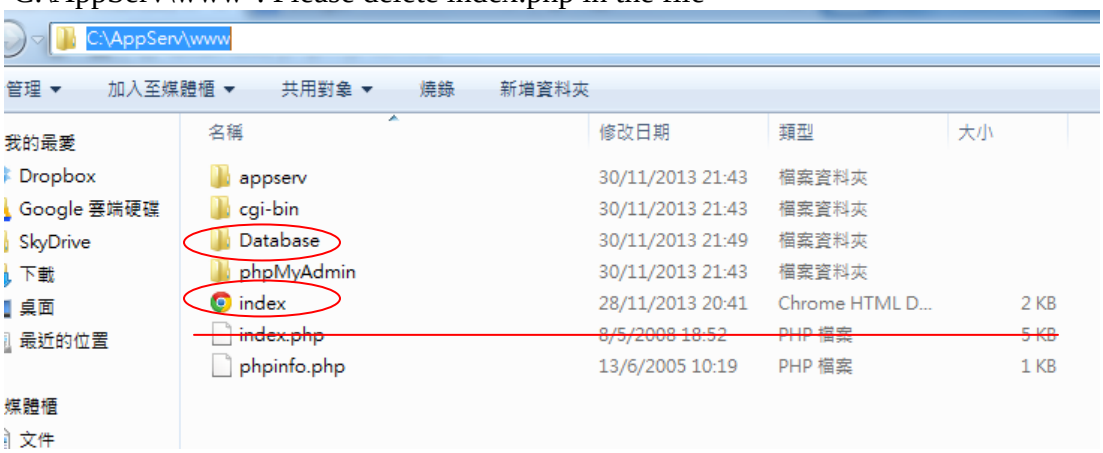
Step 13: Please go to import, and import "ccn2240_gpdb.sql" (the file is in the Database_data_file file in the CD/DVD). After, that press go.



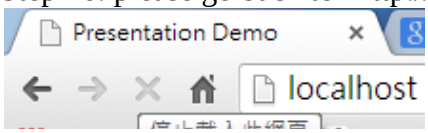
Step 14: if it is success, it will show the message



Step 15: Copy the documents and file in the Source_Code_v4.2 file in the CD/DVD to "C:\AppServ\www". Please delete index.php in the file



Step 16: please go back to "http://localhost" and click "Enter to the Database System"



Hong Kong Community College

2013-2014 Semseter 1

CCN2240 Databases Systems

Database

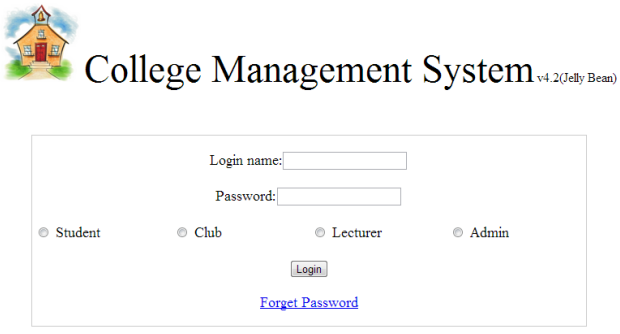
GP101B-GP01

Student Name	Student Number	Class
Chan Kwan Wing	12607532A	101B
Chan Him Hei	12313032A	101B
Lau Kwan Chiu	12650300A	101B
So Mau Chiu	12660032A	101B

Topic : College Management System

Enter to the database System

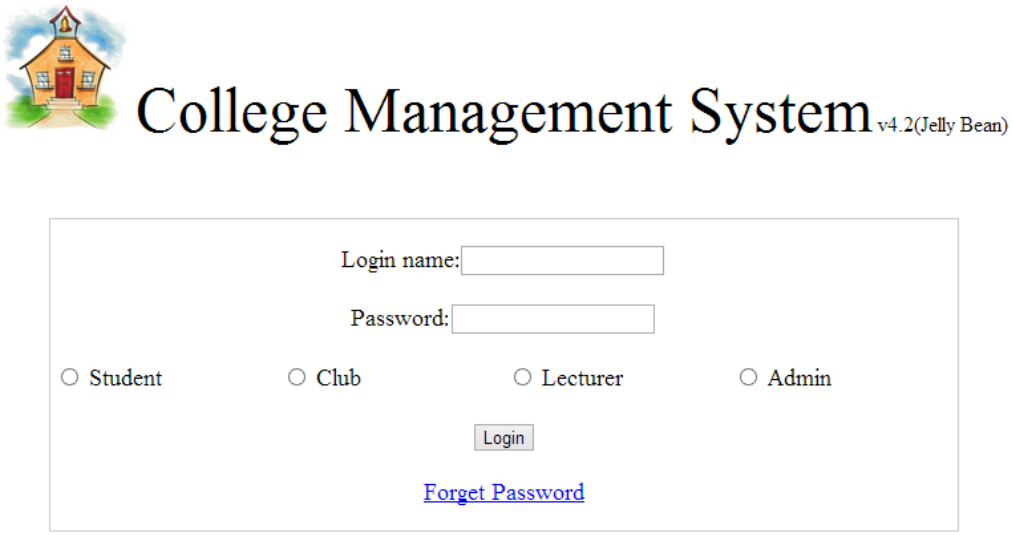
Step 17: You can use our database now.



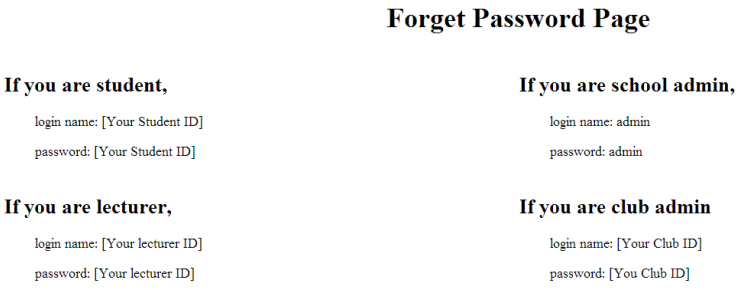
7.2 Using the database applications

1.General

For the application of our System, we expressed by graphical user interface (GUI) The GUI implement by web site format.

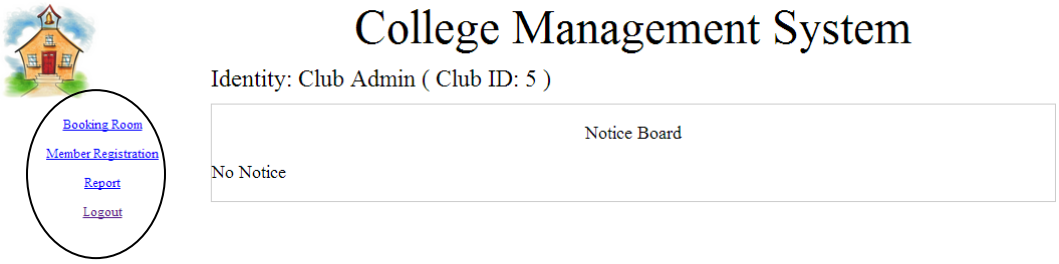


This is our main page. Each user can use their own login name and password to login their own account.
Example:
Student: Student ID -12660032A(login name:12660032A, password:12660032A)
Lecturer: Lecturer No - 16 (login name:16, password:16)
Club Admin: Club ID - 5 (login name:5, password:5)
School admin: login name: admin, password: admin

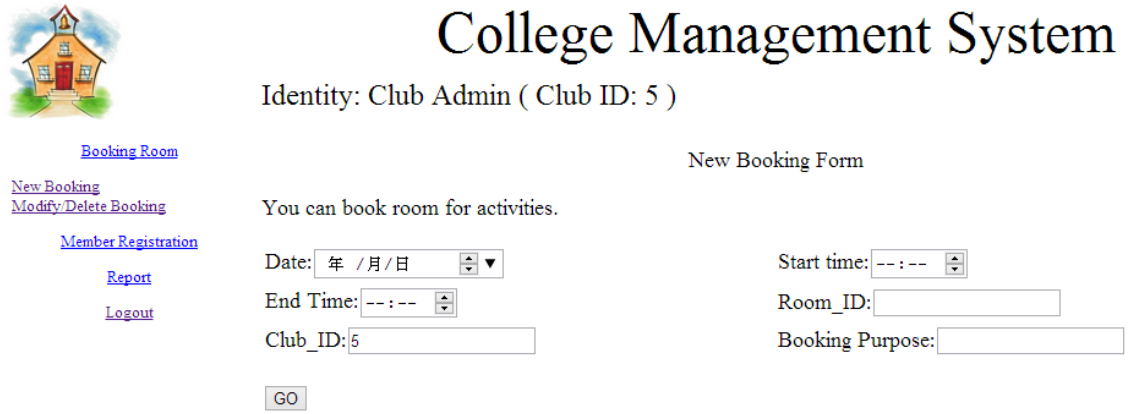


This page show the user how to login.

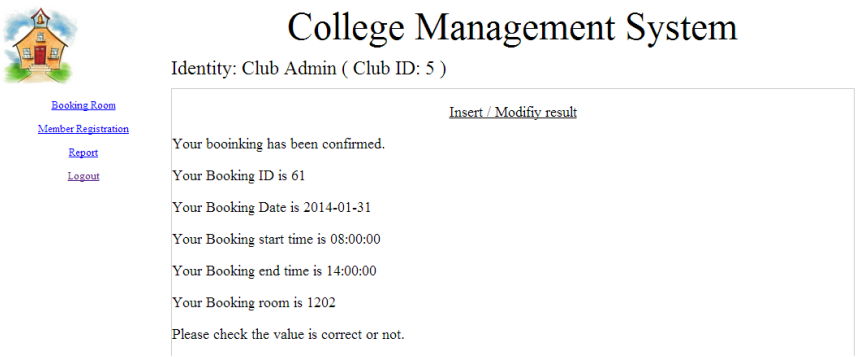
2. Club Admin



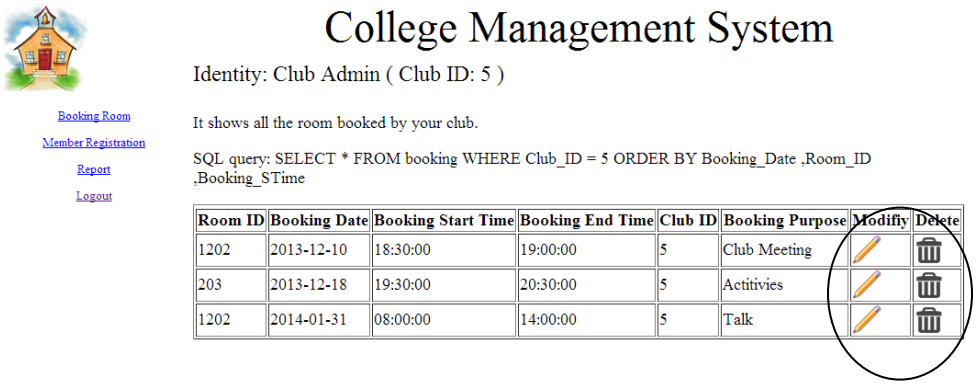
This is the index of club admin. If there are some notice, it will show in the notice board. You can find out the application in the left side.



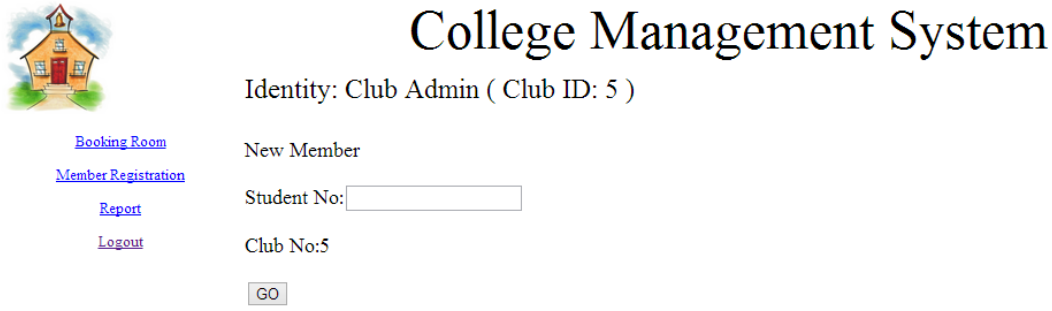
This a booking form. User can book the room for their uses.



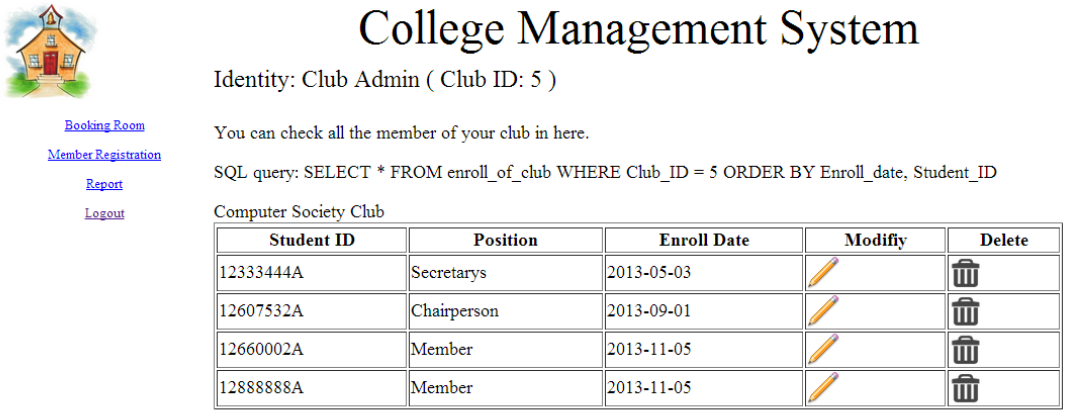
It shows that booking of the room is successful



In this page, you can find out all the booking booked by the club that user control. You can modify the booking or delete the booking by click the photo.

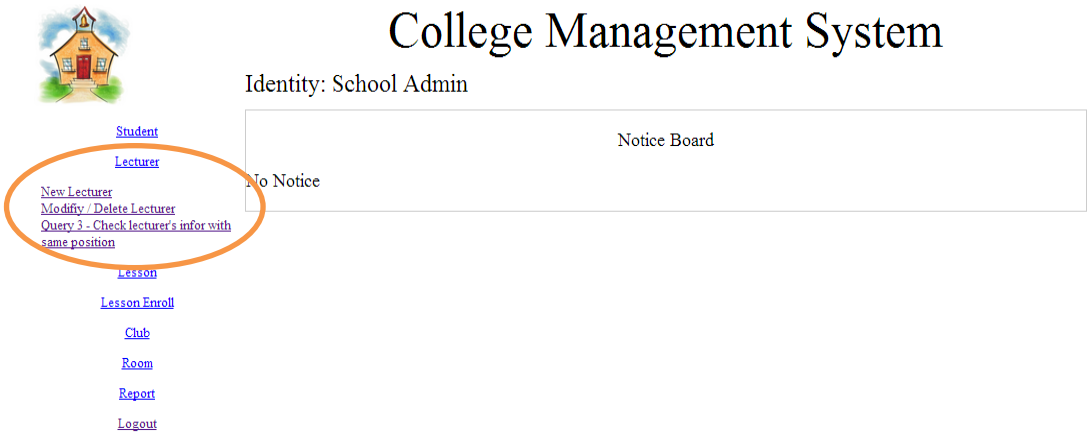


You can add a new member by entering their student number.

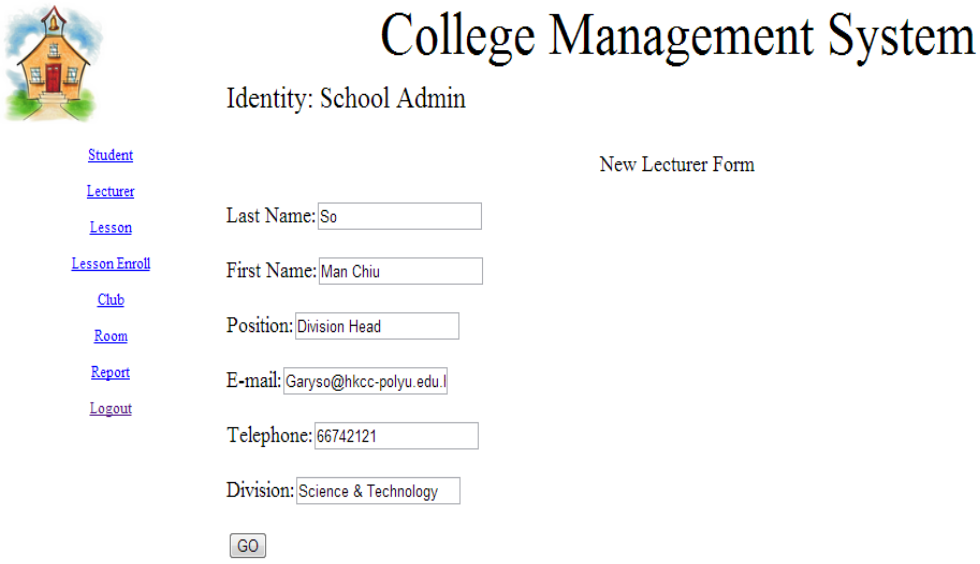


You can see all the member of your club. You can modify their position of the club. Also, you can delete the member.

3. School Admin



We try to create a new lecturer called So Man Chiu.



Click Lecturer→ New Lecturer.

Clicked “OK” to submit.

College Management System

Identity: School Admin

Student
Lecturer
Lesson
Lesson Enroll
Club
Room
Report
Logout

Insert / Modifiy result

The Lecturer's information has been saved.

Information added:

Lecturer's ID is 62

Lecturer's Name is So Man Chiu

Lecturer's Position is Division Head

Lecturer's E-mail is Garyso@hkcc-polyu.edu.hk

Lecturer's telephone is 66742121

Lecturer's Division is Science & Technology

Please check the value is correct or not.

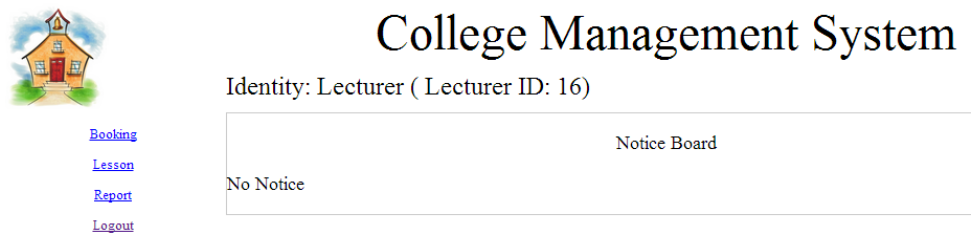
You can go back to previous page to change the value.

After submitted,

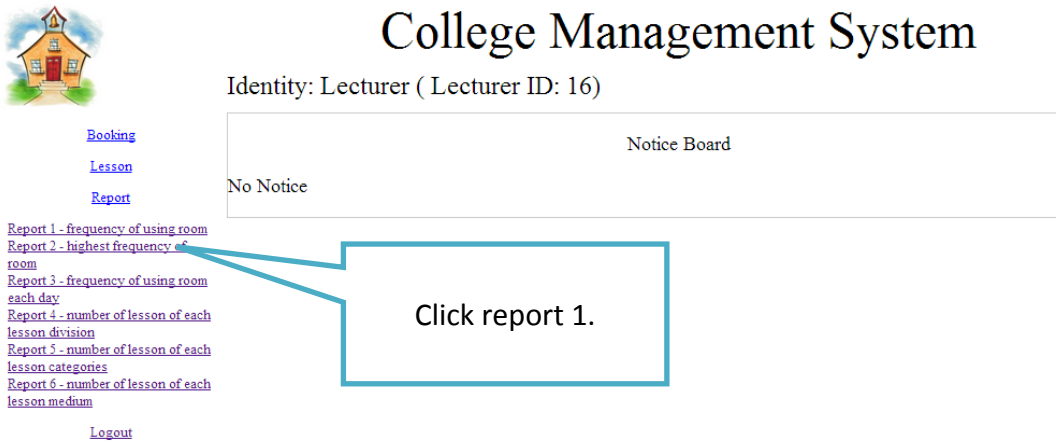
57	LAU	Kin-wai	Lecturer	ccdanny@hkcc-polyu.edu.hk	37460248	Business		
58	NGAI	Sze-ngo	Lecturer	ccbngai@hkcc-polyu.edu.hk	37460219	Business		
59	WOO, Arison	Suk-ching	Lecturer	ccarison@hkcc-polyu.edu.hk	37460618	Business		
60	CHOI	Wai-yuk	Lecturer	wychoi@hkcc-polyu.edu.hk	37460645	Language		
61	CHOW	Wen-chun	Lecturer	ccrchow@hkcc-polyu.edu.hk	37460473	Language		
62	So	Man Chiu	Division Head	Garyso@hkcc-polyu.edu.hk	66742121	Science & Technology		

➔ The new lecturer will be automatically distributed a Lecturer_ID which is not null and unique.
➔ Successfully created a lecturer.

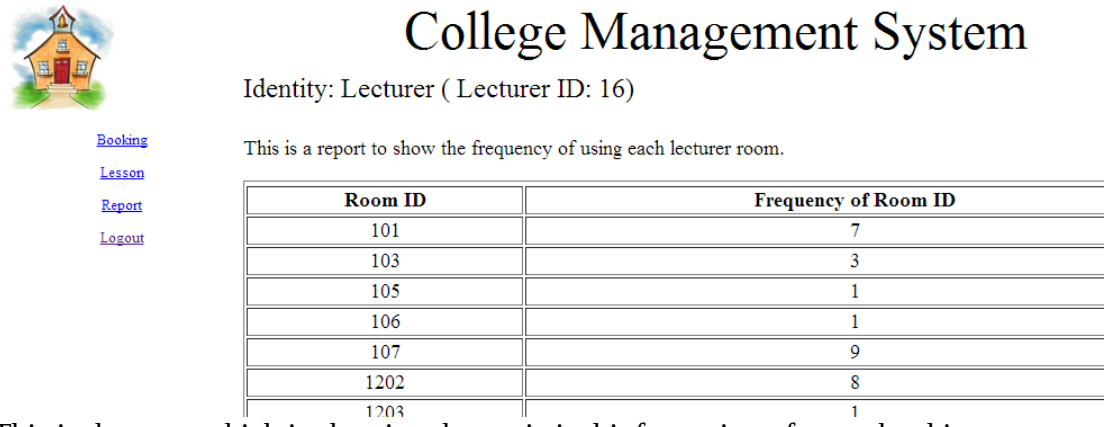
4.Lecturer



This is the index page of lecturer.



Not only lecturer can book room, but also lecturer can retrieve the statistical information of room booking.
Click Report -> Report 1 - frequency of using room

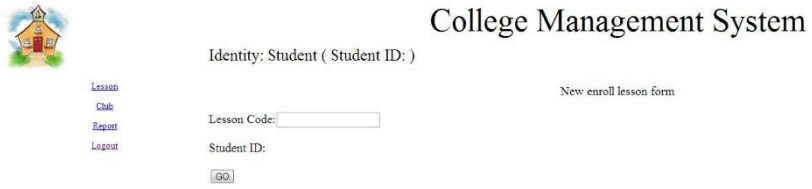


Room ID	Frequency of Room ID
101	7
103	3
105	1
106	1
107	9
1202	8
1203	1

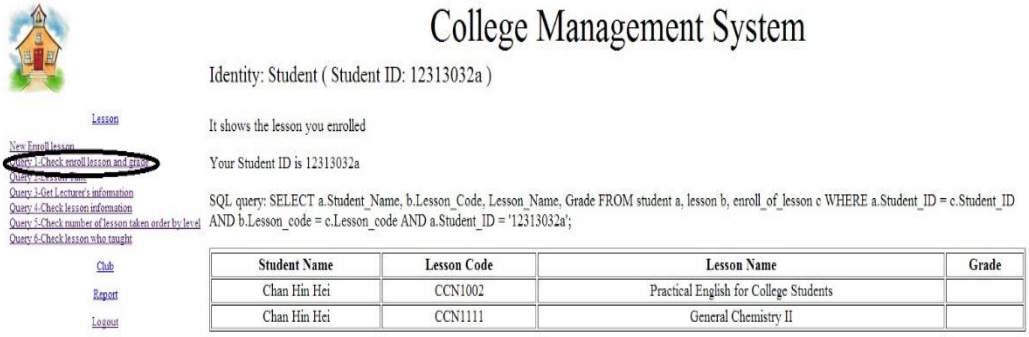
This is the page which is showing the statistical information of room booking.

5.Student

This is the index of student. If there is some notice, it will also show at the notice broad. You can find out the application in the left side.



This is the form for student to enroll lesson. Students can enroll courses by typing the lesson code.



Students can find the courses they enrolled and the grades in this page.

8. Difficulties

8.1 Design phase

After we collect the room information of HKCC, we summarize and classify the data. Moreover, we try to think about the relationship of the database. During the process of discussing the relationship of the database, we find that there are some difficulties about M:N relationship. To handle the difficulties, we learn from books and lecture notes. Finally, we break M:N into two 1:M relationship and add a composite entity to solve the problem.

Moreover, we have to design many queries because we have lots of user -- school admin, lecturer, student and club admin. We consider the query carefully because different user has different authority.

Finally, we try our best to design a PHP website interface which can make user easy to use. We do some research about login system and other PHP function which raises our searching skills.

8.2 Implementation phase

After designing the database, we are facing a big technical challenge. For example, we need to find an effective method to make a valid connection between PHP and database.

Moreover, we use some JavaScript, CSS, AppServ, MYSQL to coordinate and maintain the design of webpage.

8.3 Lessons learnt

After we finish the project, we fully understand the flow of developing database. We understand some abstract concept of the lecture. For example, we should always use Database Life Cycle (DBLC) when designing our database. We realize that the technique of DBLC is used on our project because we always step back to the previous part to better the system. This process is really strengthened our carefulness, patience and team spirit.

Moreover, we get the experience of developing database. We may face the familiar situation in the future work. The project builds up our confidence to develop database.

9.Work distribution list with time

Task Name	Start Date	Finish Date	Responsible
1. Proposal	25 September	9 October	Ken, Gray, Andy, Eric
2. Conceptual data modeling	14 October	20 October	Gary, Andy
2.1 Business Rule			Gary, Ken, Andy, Eric
2.2 ER-Diagram			Gary
3. Logical design	21 October	27 October	Eric, Gary
3.1 Relationship			Gary, Eric, Ken, Andy
3.2 Schema			Gary, Ken
4. Physical design	28 October	12 November	Andy, Eric
4.1 Source of data			Andy, Eric, Gary, Ken
4.2 SQL			Gray, Andy, Eric, Ken
5. Implementation	6 November	29 November	Ken, Gary
5.1 Source Code			Ken
5.2 Test Case			Gary, Andy, Eric
7.Presentation	11 November	27 November	Andy, Gary, Ken, Eric
8.Documentation (Report)	20 November	2 December	Andy, Ken, Gary, Eric

10. Conclusion

To be concluded, our College Management System aims to provide an online system for user to book room for lessons or club activities.

There are lots of advantages of using this system.
First of all, users can book room 24 hours a day on the internet. It can enhance the usability because user can book room at everywhere on everytime at any time. It can also reduce the workload of the administrator as the users can book room by themselves.

Secondly, the system contains specific access right for different users. It can protect the privacy of the users' data. For example, a student cannot get the personal information of another student. However, school administrator can get every student' information.

Moreover, we develop many queries which is used to:
1. Create/ Modify/ Delete the record of booking, lecturer, student, lesson, club, etc.
2. Retrieve the information of lecturer, student, lesson, club, etc.
3. Retrieve the statistical information for decision making.

11. Reference

1. Club information
   http://www.hkcc-polyu.edu.hk/collegelife/Students'_Union-38.html
2. Lecturer information
   http://www.hkcc-polyu.edu.hk/staff_directory/index.php
3. AppServ
   http://www.appservnetwork.com