P.L.K. Vicwood K.T. Chong Sixth Form College

Hong Kong Examinations and Assessment Authority
Hong Kong Advanced Level Examination 2010
Computer Studies Paper 3
Project Assignment
Title 1: Sports Day System

Name: Lo Yu Ho
Class: 7C2
Class Number: 15

Content

Attachment:
One CD and one copy of source code

Objective of the coursework

Most Hong Kong schools hold sports day annually. It is a good tradition, but organizing a sports day is not easy. The preparation of a sports day always takes a long time, about three to six months before the holding day.

In these three to six months, the planner has to collect the information, arrange the things, and finally generate a program directory to show the sports day details. In this process, the data that the planner facing is a thousand of students' profiles, a long list of participation, and also may be a mess time table. The one handling these data has to pay a lot of time and efforts. It is a tough work.

Can it be solved by a computer? Computer is good at handling massive data and processes them, which takes the most time of organizing a sports day. If using the computer in organizing a sports day, the time used in preparation will be greatly shortened.

What tasks can the computer help us to do?

The collected students' profiles are not initially sorted by the students' date of birth, which is important for categorizing students into different grades. Sorting a thousand of profiles is hard for manual operation, but only takes a while for computers. So, to categorize students into different grades according to their date of birth (age) is an easy job for a computer.

Students can participate in a few events, but always has a limit. Computer can help to restrict the number of events a student applies for. It will report to the planner if there is someone applied over the limitation.

Most of the sports grounds in Hong Kong painted with eight lanes only and the track events are always more than eight participants joined. So the events will divide into heats and take the best eight into the finals. Allocating participants into different heat events is a tough work. The planner has to calculate how many heats for each event. And also, the planner has to distribute lanes evenly to avoid only three or four participants in one heat, but full in other heats. Using computers can make it much easier and only take a short time. It can also provide a fair random allocation of the participants.

During the sports day, one of the difficult tasks is ranking the participants' results. The participants' result received from the event holding helper team is unsorted. It is hard to make these data meaningful and show who will enter the finals or who won the games. Computers can help the organizer again in sorting data. Input the result into a computer, and it will give a report based on the rank order of the results. So the qualifiers in the finals and the winners of events can be easily found out.

Near the end of the sports day, the most difficult task for the organizer will come. It is the computing of individual and house champion. One thousand students and one student can participate in three events. The worst case is three thousands results have to be computed to conclude the champion. Luckily, the results being computed is not that many, by my observation, it is around one thousand results are recorded. But it is still hard to compute and it will take a long time to do the task. Computer can help in handling this complicated task. It will base on the results inputted to calculate the scores for each individual and also the houses.

The above is showing the computers can make the organization of sports day much easier, shorten the time on preparing data and computing data greatly. But the computer does not know how to do these tasks. In other words, we need a computer program.

The followings of this report are showing the work in design and implement a sports day system performing the above tasks. I will show how to design a database schema of the system. And more important is designing the algorithms for event and participant allocation to minimize possible time clashes and reporting the results of events and merits. And also write a program to implement the algorithms. These are the aims, also the objective of this coursework.

Analysis of the situation

To develop a system, I have to know what equipment will be involved in when using the system.

In some sports events, there will be some digital devices to record the result, for example, timers in swimming gala, timers in track sports, the device for measuring distance in field sports, etc. Such devices can input data into the system automatically after recorded the results. So that there will not involve manual operations in result recording process. The biggest advantage is the result recorded can be more accurate and avoided many manual errors. They are commonly seen in a sports competition. Will they be involved in my system?

However, after I asked some sports day planner, they seldom use digital devices in recording the athletes' results. They mostly invite students to form a group of working team and assign a teacher for responsible of the team. That means results are inputted into computer after the recording person and the judger. It is reasonable since the cost of using digital equipment is not cheap and there must be a well developed system to support the devices.

So, my system will be compatible with simple, general devices in a general Hong Kong sports ground. What I need is a computer with monitor, keyboard and mouse. It is the lowest requirement for using my system. A general notebook computer can fulfill this requirement.

Computer & monitor Keyboard Mouse Notebook

But a computer can either be a Mac machine or a PC, it is a question of which platform will the system work on. In Hong Kong, Apple computers are not commonly in use, most of the computers are PC with Microsoft Windows. So, I will not consider the compatibility of the operating system other than Microsoft Windows.

After considering the sports ground equipments and the computer, the third thing I have to consider is the number of computers in use. Many schools in Hong Kong were equipped with a computer room or more. It is possible for the students to input the information into the system by there own. I think it is important to develop the system that can use this advantage. One of the usages can be letting students to register the sports events their own through computers, it is possible for them to register at the same time, since schools have computer room with a number of computers. Further more, a multi-user environment can also provide chances for different helpers use the system to input recorded results at the same time, e.g. long jump helper & track helper can input the sports result into the system at the same time in different places.

Analysis on programming language used in developing the system

After knowing what the sports day system can use, I am going to decide what I will use in developing it, the programming language and database management system (DBMS).

Programming language can be categorized into three generation, “first-generation”, “second-generation” and “third generation”. Of course, it is impossible for me to use the “first-generation” and “second-generation” language. The one I will choose is inside the “third-generation” language. However, inside the “third-generation” language, there are four commonly used languages are BASIC, C, C++, C#, Pascal and Java.

BASIC

It is a language originally aimed at providing computer access to non-science students since only scientists and mathematicians are able to write custom software.

C, C++, C#

The C family is the most popular programming language. It is originally aimed at developing system software. It is now widely used in developing many kind of software, especially C++, which is using in developing application software, device drivers, high-performance server and client applications, and entertainment software such as video games.

Pascal

Pascal is an influential imperative and procedural programming language as a small and efficient language intended to encourage good programming practices using structured programming and data structuring.

Java

The language derives much of its syntax from C and C++ but has a simpler object model and fewer low-level facilities. Java applications can run on any machine regardless of computer architecture. This programming language is a general purpose concurrent class based object oriented programming language, specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere".

The programming language I learnt in school is BASIC. It is easy to use and well supported by Microsoft Developer Network (MSDN). I can find many information, commands and source codes from http://msdn.microsoft.com/en-us/vbasic/ when I need. Since the sports day system does not require special tasks to do, BASIC can

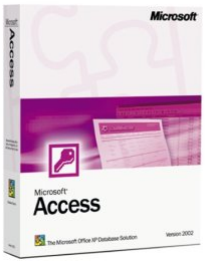
perform all the function I need. Although I know the C family (C, C++, C#) is some where more powerful than BASIC and there are many developer networks can help me, I still prefer the one that I am more familiar with. Also, the system is running on PC only, the biggest advantage of Java, "write once, run anywhere", is not important in this case. So, I will use BASIC as the programming language in developing the sports day system.

Analysis on DBMS used in developing the system

Besides the programming language, the DBMS (database management system) for the system has also to be decided.

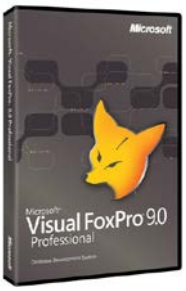
A DBMS is used for controlling the organization, storage, management, and retrieval of data in a database. DBMS accepts requests for data from an application program and instructs the operating system to transfer the appropriate data.

The database query language is almost the same along different DBMS. There is a very detailed comparison on different DBMS from wikipedia (http://en.wikipedia.org/wiki/Comparison_of_relational_database_management_systems). I get a deep understanding of different DBMS. In the market, there are many DBMS available for me to choose. However, I prefer a more popular DBMS since I can ask for help more easily. I only take some famous DBMS to discuss, namely, Access, FoxPro, MySQL and Oracle. What I want the DBMS to do in the system is very simple. I want it to perform simple database query tasks, has a basic security protection for my data, easy to set up a DBMS server and I can connect to it at anytime, free of charge is preferred.



Microsoft Access

Microsoft Access is very popular among non-programmers who can create visually pleasing and relatively advanced solutions on their own. It is mostly use in web development. Moreover, it is always use as stand alone DBMS. So, use Microsoft Access to do multi-user tasks is not preferred. It is usually for personal use rather than multi-user.



Visual FoxPro

Unlike most database management systems, Visual FoxPro is a full-featured, dynamic programming language that does not require the use of an additional general-purpose programming environment. It can be used to write not just traditional "fat client" applications, but also middleware and web applications.

MySQL

MySQL is often used in free software projects that require a full-featured database management system. MySQL is the world's most popular open source database software. It is a multi-user DBMS, allowing many users access the database concurrently.

Oracle Database

Oracle Database is powerful DBMS. The biggest selling of it is its security measures. It provides a very high security for the database, preventing data lost, preventing unauthorized access, hacking, etc. And also, the function that is can perform is the most one, advance setting of the DBMS also the most detailed one. However, it is usually for professional use.

The DBMS is the vital part of the system. As I mentioned in the objective, this system aimed at solving the massive data problem and the computing of the data. It is the work that a DBMS does. When choosing the right DBMS, I must consider the tasks of the system will do. One of these tasks is mentioned above, letting students register the sports events through computers. There are many ways to do this task. The straightest way is let the student register through computers that directly connected to the DBMS server, but it needs extra work to connect the computers to the server. The more efficient way is let the computers access the DBMS via the internet. Computers inside the schools are mostly connected to the internet, so the system can connect to the database server without extra works. But setting up a DBMS on the internet is not easy. Luckily, my school hosts a MySQL server and I can access it at anytime and anywhere via the internet. And also, when I was learning database query language, I practiced on MySQL rather than others DBMS. It will be easier for me to use it and do not need more time spending on knowing the other DBMS.

Design of the system

After analyzed the situation and decided what programming language and which DBMS I will use, I am going to design the database schema and the algorithm of the system in this phase.

In particular dimension, computer like a secretary. You give it some data and tell it what to do on those data, and then it will give you back what you want. So, when I am going to design a system, I would like to know what will be inputted into the system and what information the users expected the system to return.

Specific tasks for the system to perform

Firstly, the system is supposed to categorize students into different grades. The user input the students' profile including their date of birth which is used as the criteria to categorize students. The criteria of the categorizing process, students born on which date or before will be categorized as grade A, and also grade B and grade C. The system also required the students' profile inputted including the students' sex, house, class, class number and a unique student id for later use.

When students register the events they want to join, the organizer has to make sure no students joined the events with wrong gender or wrong sex, for example, a C grade male student cannot join a female event or an A/B grade event. So, when the user of the system is inputting the event information, the system requires the name, sex and grade of the events. The system will also require the information that shows the event is a track event or a field event, and should the system assign lanes to the participants for the event, for example, 100M is a track event needs to assign lane.

Then, the registration process can be started and the collected information can use for making some statistic to show how many participants joined a particular event. After the registration process the computer will help to distribute lanes to the participants who joined the events that need to assign lane.

To minimize possible time clashes for event and participants allocation, the system will have a time table generating tool for the user. The system will ask he/she to input the starting time and the time needed to hold the event, and then the system will generate a time table that achieve the minimum possible time clashes.

Additional tasks for the system to perform

Before the start of sports day, the system needs the calculation criteria for compute the merits, for example, score for awarding the participants entering the finals, score for awarding participants placed in position one to eight in an event, penalty for absent, etc. After the start of sports day, results of the participants will be inputted into the system continuously. And the system will compute the data inputted and show the winners of events, also the merits for individual and house champion according to the data inputted.

Form the details shown in above, the user will have to complete the following procedures when holding a sports day from start to end.

Stage 1: Input students' profiles and events information

Stage 2: Input the students' registration for events

Stage 3: Let the system to distribute lanes to participants. And user input the starting time and time length of each event. Then the computer generates an organized lane arrangement list and time table.

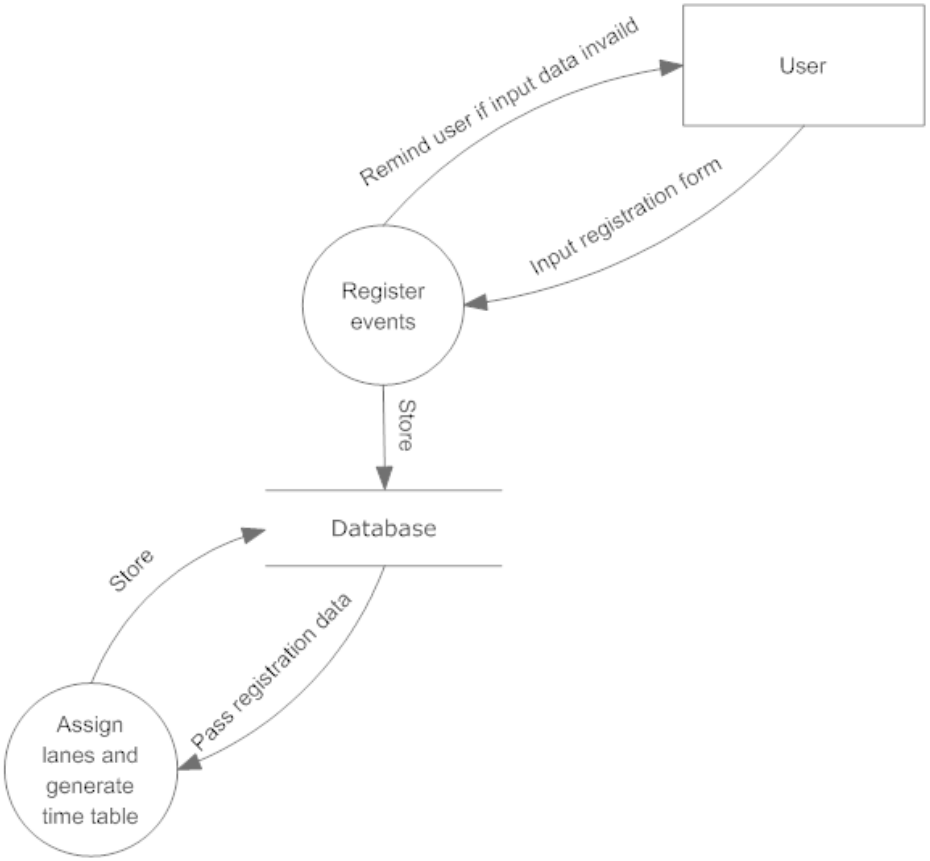
Stage 4: Input results and system will generate a report showing the winners and merits

Moreover, because of the data inputted into the system is quite a lot, I designed a management system for handling the files of the inputted data. And this system allows the user retrieves his/her work since last stopped. When doing a sports day project, the system can be interrupted to do another project. That means, when project A is in planning, it is possible to pause it. And you can start project B without losing the data in project A and recall it later to continuous your work.

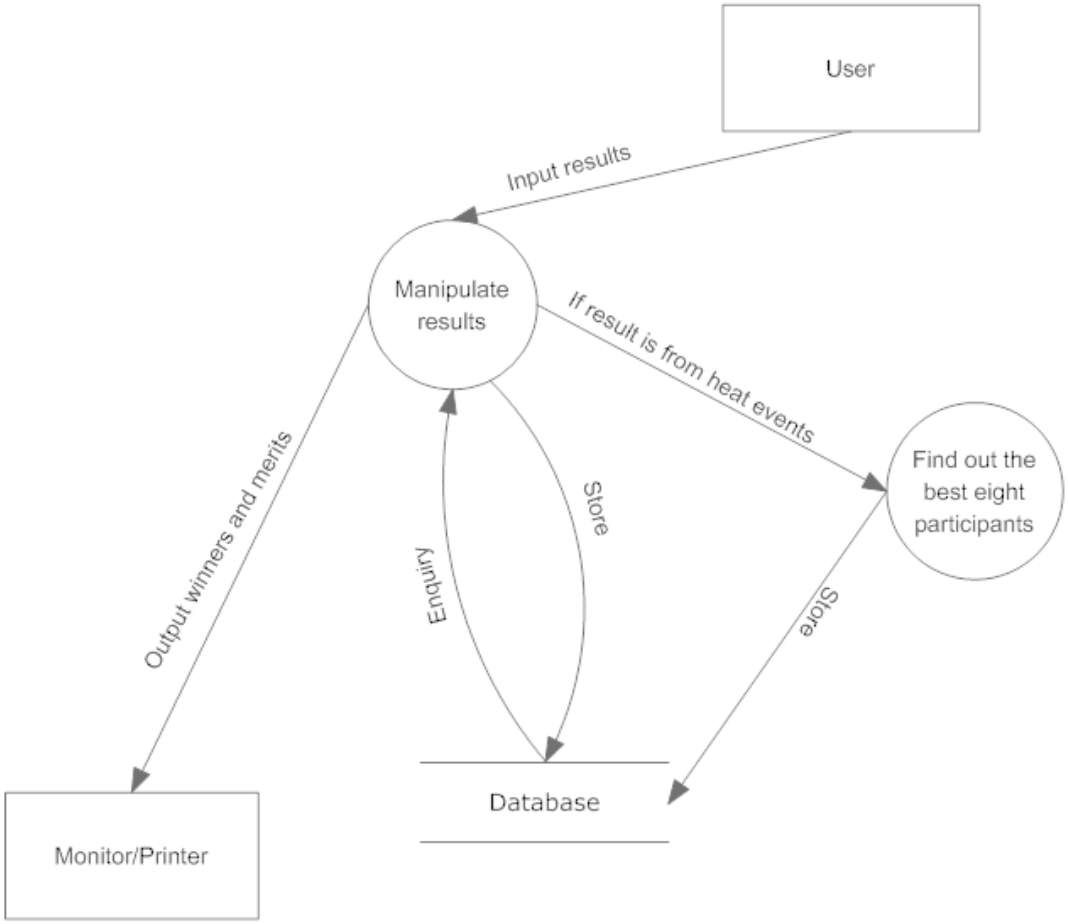
Data flow design of the system

Let me describe what the system will be. Firstly, the user can choose to start a new project or load an existing project in the first step. If the user loaded a project, he/she can continuous the work at the point that he/she stopped at last time. If the user chooses to start a new project, he/she has to input the basic information of the competition, score calculation schema, house information, students' profiles, event details and helper team details. After he/she finished this step, the computer will put the data into tables and store it inside the MySQL server. Then the user can input the registration data into the system or use the sub program to let students register events their own. Looping this registration process until all students registered the events their want to join. After the registration process, the computer will start to divide some events into a few heat events and distribute lanes to participants. The user can start planning the time table(s) after the grouping and distribution of lanes process. When finished the planning of time table(s), the organizer may has to publish some program directories to students showing what time holding what events and showing the enrollment of participants. During the sports day, the user has to input the results of events to the system, so that the system can find out who will participate in the finals and distribute lanes to the participants. After all the results of events were inputted, the computer will report the winners and merits. Finished the prize presentation, the sports day will come to the end.

Here are the data flow diagrams for the above design.



This diagram is showing what the system will do in preparing a sports day. The user input the registration form into the system, and the system store the data into the database, another program of the system will access those data and generate a table storing the lane distribution for participants and a table storing the allocation of events, that is the time table of sports day.



This diagram is showing what the system will do when a sports day is running until the end. The user input the results of events and the system will manipulate the received results. If the result is a final event result, the system will store it into the database, if it is a heat event result, the system will find out whether the participants can enter the finals or not, and then store into the database. The generated data will output to a monitor or printer through a printer.

Database schema design of the system

For a sports day project, the system requires the user to input massive data. I have done some research on the sports day held by my school and have a conclusion on what information the system requires. They are named in list form below. I will explain those items later. And this list will be used in designing the database schema of the system as well.

SDMProj *(Proj_code, Name, Author, PW, Description, Stage_1, Stage_2, Stage_3, Stage_4)*

SDMCompInfo *(proj_code, school_name, school_year, title_of_competition, venue, venue_address, no_of_days, date_of_day1, date_of_day2, bonus_for_participation, absentee_penalty, bonus_for_finals, sex, a_grade, b_grade, c_grade, a_grade_time, b_grade_time, c_grade_time, start_time_day1, end_time_day1, start_time_day2, end_time_day2, postion1_score, postion2_score, postion3_score, postion4_score, postion5_score, postion6_score, postion7_score, postion8_score)*

#Proj_student *(SID, Name, Sex, DOB, House, Class, Class_No, HKID, Emergency_Call_Phone_Number)*

Proj_event *(Event_Code, Event_Name, Grade, Sex, Field_or_Track, Assign_Lane, Holding_Area)*

Proj_helperteam *(Team_Code, Team_Name, Leading_Teacher, Working_Area, Duty_Time_Start, Duty_Time_End)*

Proj_House *(House_Code, House_Name)*

Proj_eventregister *(SID, Event1, Event2, Event3)*

Proj_helperregister *(SID, Team_Code)*

##Proj_noofgroups *(Event_Code, Event_Name, Grade, Sex, Field_or_Track, Assign_Lane, Holding_Area, No_of_Participants, No_of_Groups)*

Proj_lane *(Event_Code, Group_Code, Lane1, Lane2, Lane3, Lane4, Lane5, Lane6, Lane7, Lane8)*

Proj_eventtime *(Event_Code, Minutes_Required, Start_Time)*

###Proj_timetableday1:

###Proj_timetableday2:

Proj_results *(SID, Event_Code, Result, Absent, Score)*

Proj_finals *(Event_Code, Event_Name, Grade, Sex, Field_or_Track, Assign_Lane, Holding_Area)*

Proj_finalslane *(Event_Code, Lane1, Lane2, Lane3, Lane4, Lane5, Lane6, Lane7, Lane8)*

Proj_adjustpersonalscore *(SID, Adjustment)*

Proj_adjusthousescore *(House_Code, Adjustment)*

Proj_award *(Event_Code, Champion, 1st_Runner_Up, 2nd__Runner_Up)*

*** Primary key is the underlined attribute(s)**

 # Proj is the project code of the project, Proj_student for a project with project_code 1001 is named 1001_student

It is a view

Number of time table depends on the number of day holding the sports day, some school will hold a two day sports day while some will hold one day only. And the attributes in it is depended on the sex and the grades of the school, for example, a co-ed school holds A, B, C grade events, 1001_timetableday1*(Time, Boys_A_Grade, Boys_B_Grade, Boys_C_Grade, Girls_A_Grade, Girls_B_Grade, Girls_C_Grade)*

The table SDMProj is the table used to manage the files and the program will disable some button according to the attributes, stage_1, stage_2, stage_3 and stage_4 to avoid exceptions. PW is password. It is used to protect the project from unauthorized access.

The table SDMCompInfo is the table used to store all the general information of sports day and the information used in the scores calculation for merit computing.

The table Proj_student is the profile of students, the HKID attribute is used for a sub product program which is used for the students to register events by their own. The attribute Emergency_Call_Phone_Number is used for accident. When a student injuries during the sports day, the system offers a way for the teacher to contact the students' parent.

The table Proj_event is the information of the events.

The table Proj_helperteam is the information of the helper teams. As I mentioned in analysis, some students will form help teams to help teachers in running the sports day. This table is used to store the information of the teams.

The table Proj_house is the information of house in school.

The table Proj_eventregister is the information of students' registered events. It has three and only three attribute named as Event_1, Event_2 and Event_3. So, students are restricted to register three events at maximum.

The table Proj_eventregister is the information of students' registered helperteam. One student can register for one team only.
The table Proj_noofgroups is a view show the number of participants in an event. If the event needs to assign lane, it will calculate how many group will be formed.

The table Proj_lane is the information of lane distribution. It shows who enrolled in which lane for which group. As mentioned in analysis, most Hong Kong sports ground painted with eight lanes only, so the table will consist lane one to eight only.

The table Proj_eventtime is the information of the start time and time length of events.

The Proj_timetableday1 and Proj_timetableday2 are used to store what event holds at which time.

The table Proj_results is the information of the results of the participants. If he/she is absent in the event, the attribute will be marked as ‘abs’. The score attribute is used for computing individual and house champion.

The table Proj_finals is the information of the finals. Finals are only for the events which need to assign lane.

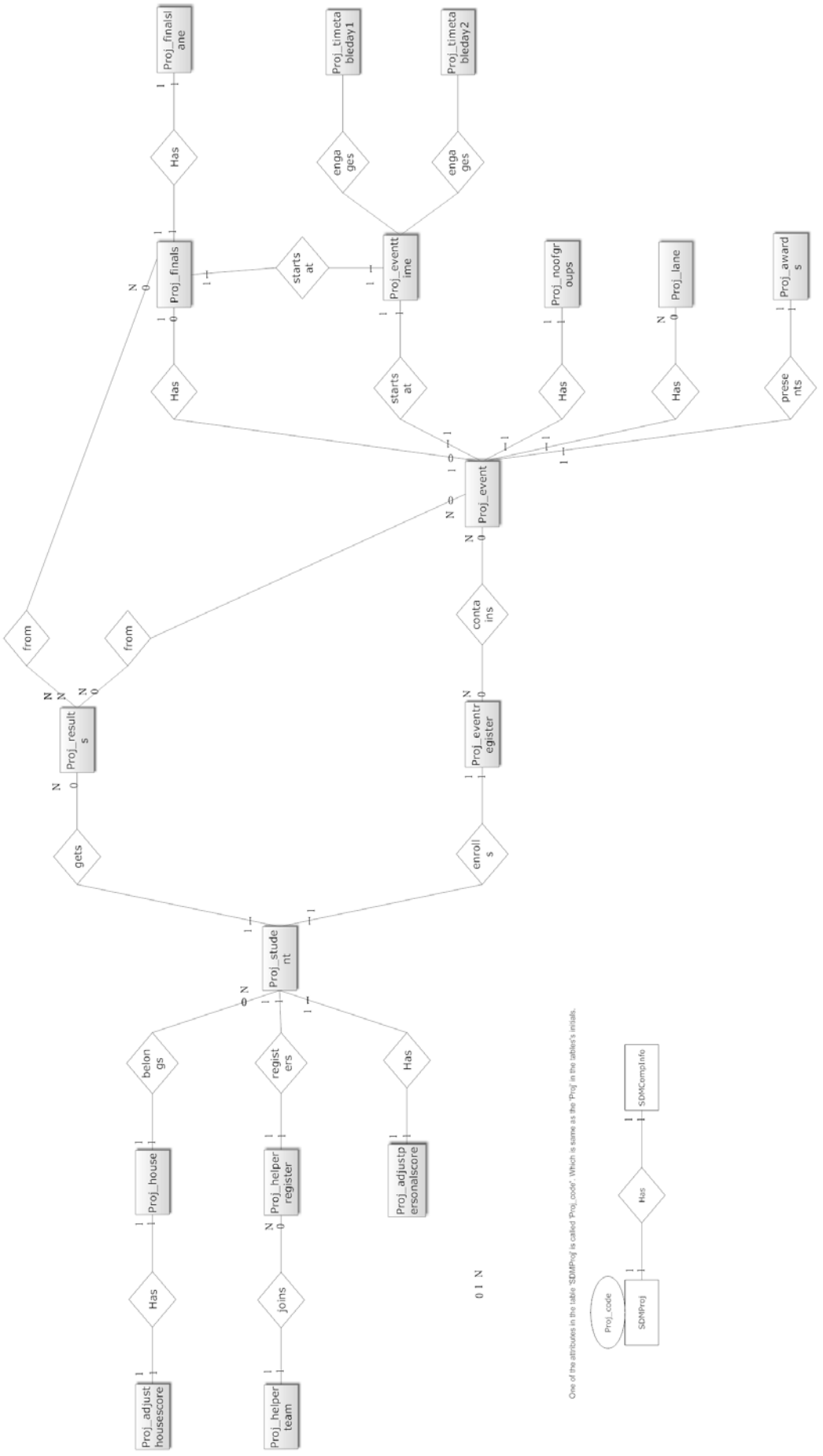
The table Proj_finalslane is same as Proj_lane, but events not long divided into groups since there are only one final for one event.

The table Proj_adjustpersonalscore is used for storing the adjustment on individual participants. While the organizer may want to award some participants for some reason, for example, breaking records. This table makes a way for the organizer to do so.

The table Proj_adjusthousescore is also used for the same reason with Proj_adjustpersonalscore. The organizer may hold some special competition not included in the system, for example, cleaning competition, cheering team competition. This table makes a way for the organizer to award the house.

The table Proj_award is the information of the winners of events. It shows the champion, 1st_Runner_Up and 2nd__Runner_Up of events.

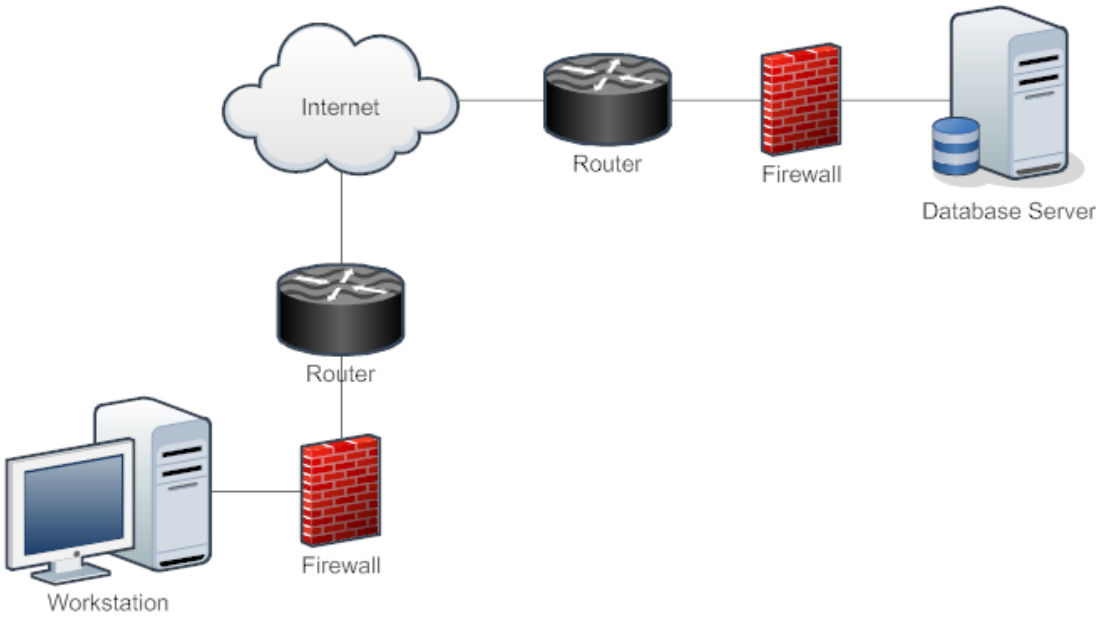
After explaining the items, here is the entity relationship diagram of these tables.


Proj_finalsl ane
Proj_timeta bleday1
Proj_timeta bleday2
Has
enga ges
enga ges
Proj_finals
starts at
Proj_eventt ime
Proj_noofgr oups
Proj_lane
Proj_award s
Has
starts at
Has
Has
prese nts
Proj_event
from
from
conta ins
Proj_result s
Proj_eventr egister
gets
enroll s
Proj_stude nt
belon gs
regist ers
Has
Proj_house
Proj_helper register
Proj_adjustp ersonalscore
Has
joins
Proj_adjust housescore
Proj_helper team
0 1 N
One of the attributes in the table 'SDMProj' is called 'Proj_code'. Which is same as the 'Proj' in the tables's initials.
Proj_code
SDMProj
Has
SDMCompInfo

The table 'SDMProj' and the table 'SDMCompInfo' are used for project management purpose. And there are totally seventeen tables including one is a view table for one project. Each project owns a project code. Although no common field in table SDMCompInfo and the other tables, the competition information of sports day in table SDMCompInfo can still link to all the tables of the same project since the tables were named with the project code as initial. That means, the tables share the same project code with the corresponding row in table SDMCompInfo.

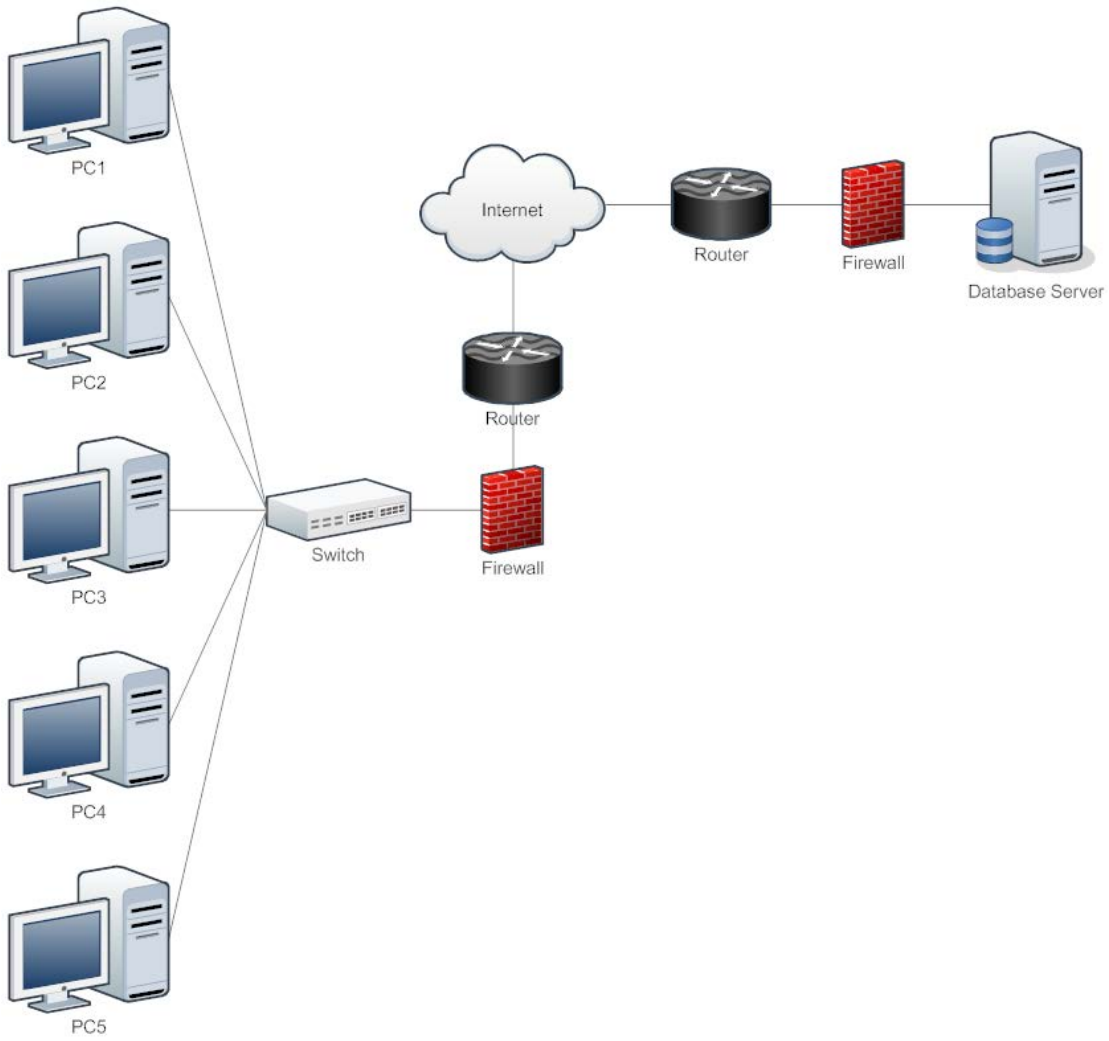
Network design of the system

As the DBMS I will use in the system is not a stand alone DBMS, it is on the internet. So, I have drawn a network diagram to show how the sports day system accesses the database server that holds by my school. The bandwidth of an ADSL or equivalent connection is enough for the communication between system and the server. The sports day system is running on a workstation.



The diagram shows the workstation connecting to the internet though a router and has a firewall between them for protecting the workstation. On the other side of the internet, the database server is connecting to a firewall and then connecting to a router and connect to the internet though the router. The firewall is a line of defense to protect the database server and keep it away from hacking or any attacks. This is also possible for the user to access the database server at anywhere with connection to the internet, for example, in the sports playground.

By the way, I discussed about letting students register sports events by their own though the internet. Here is the network diagram to show how the sub-program of the sports day system accesses the database server though the internet. The sub-program can be running on many computers at the same time.



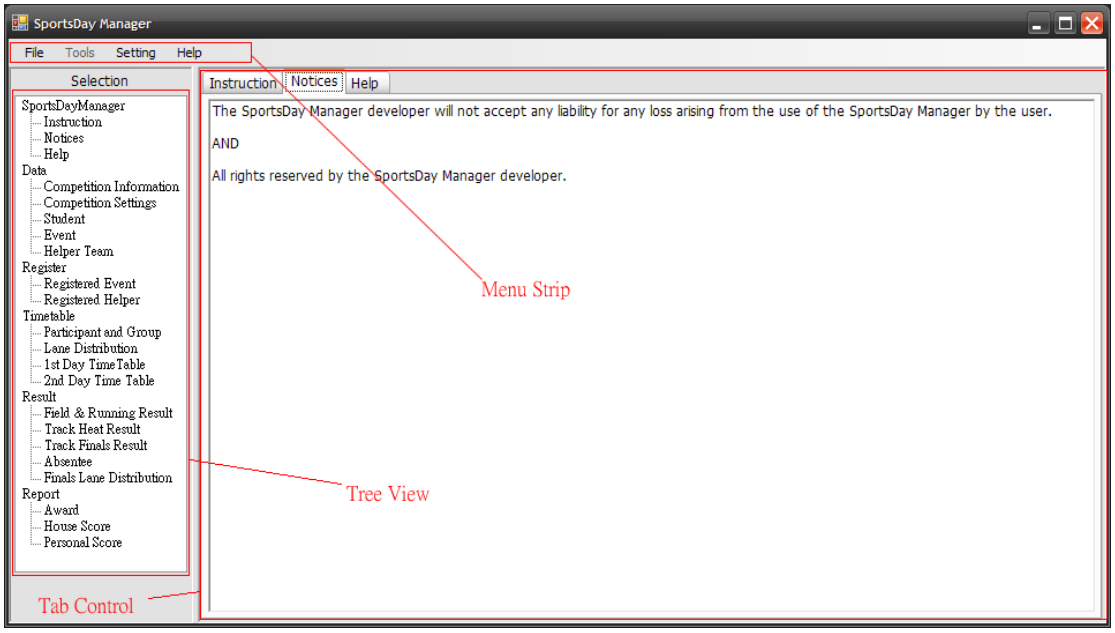
As the diagram shows, more than one computer is running the sub-program. There is a switch connecting the computers and the other side connecting to a router with a firewall between them. And then the router is connecting to the internet and can access the other side of the internet where the database server is located and connected to a router with firewall between them.

Implementation of the design

As I discussed in analysis, the programming language and the DBMS I will use in implementing my design will be BASIC and MySQL. The software I use for developing a BASIC program is Microsoft Visual Basic 2008Express Edition. And a connector of MySQL will be installed in my workstation too. It is MySQL Connector NET 5.2.6. The source code of the program can be found in the attachment.

The main form

I am going to make the main form of the system. Here is the layout design of it.



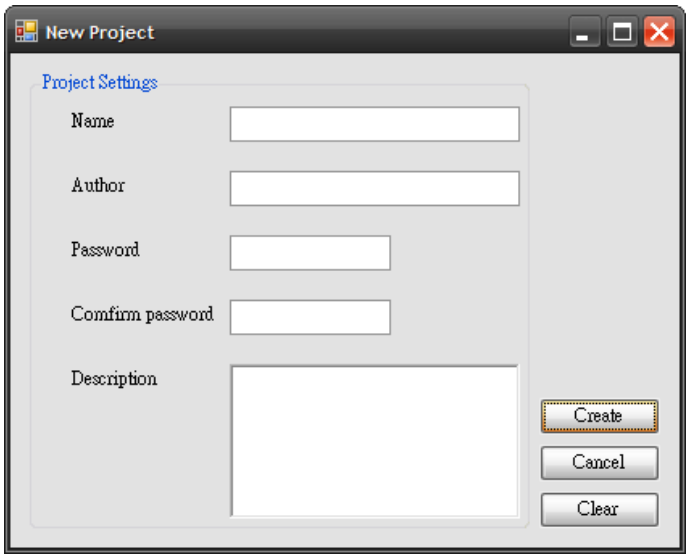
The main form contains a menu strip on the top, a tree view at the left and a tab control on the right. The menu strip is holding the items that can call out most of the function of the system. The tree view can let the user changing the content of the tab control at the right to read the information he/she needs.

That is totally six categories in the tree view, and each of the categories has a corresponding tab control. By clicking any one item inside a category, tab control at the right side will change according to which item was clicked.

Besides the tree view, there are a few functions have to be defined in the main form. It is because the system will repeat performing these functions frequently. They are dgv_show (use for load a data table to a data grid view), dgv_show2 (also use for load a data table to a data grid view), enq_exe (it is use for executing non query of the database), esint_exe (it is use for get a integer value from the database), esstr_exe (it is use for get a string from the database) and menu_strip_enable (it is use for enabling the buttons and menu strip items according to the progress of sports day planning). For the programming code of this form, please refer to form1 in the source code attached.

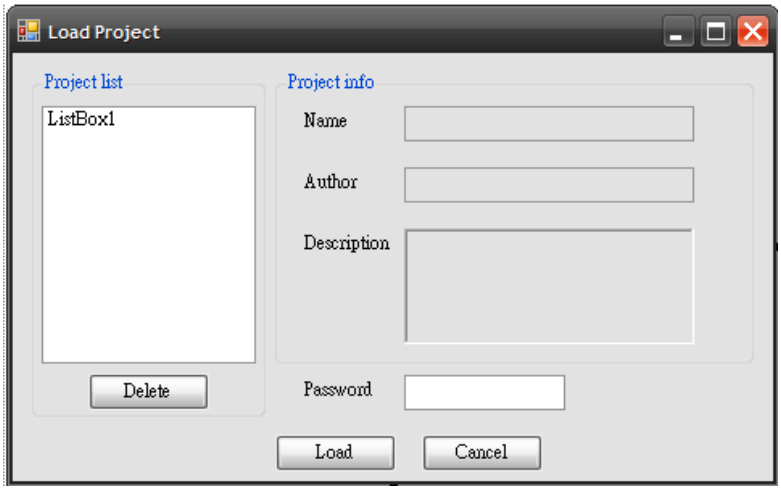
The projects management system

To start a new project, the user has to open the 'New project', which can find in the menu strip. Here is the layout design of it.



For the need of source code, please refer to form2 in the source code attached.

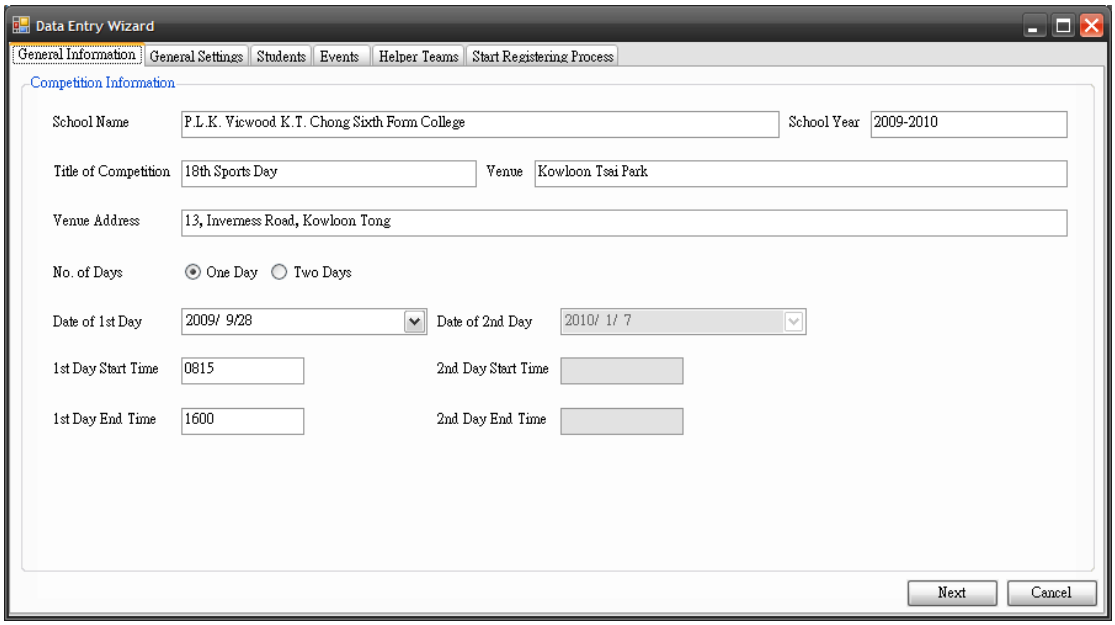
When the user want to load an existing project, he/she can call the from ‘Load project’. It can be found in the menu strip. Here is the layout design of it.



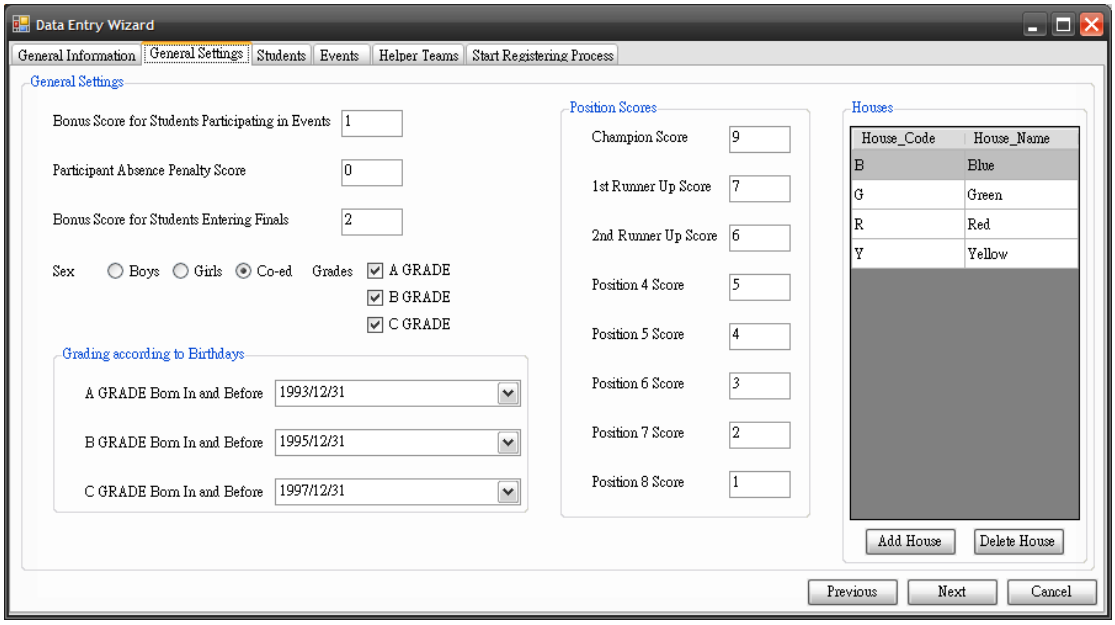
Listbox1 is used for choosing projects, button ‘delete’ is used for deleting the selected project and button ‘load’ lets the user to open a project. The action ‘delete’ and ‘load’ are projected by a password. The system will deny the user from accessing if the password is invalid. For the need of source code, please refer to form 4 of the source code.

Data entry wizard

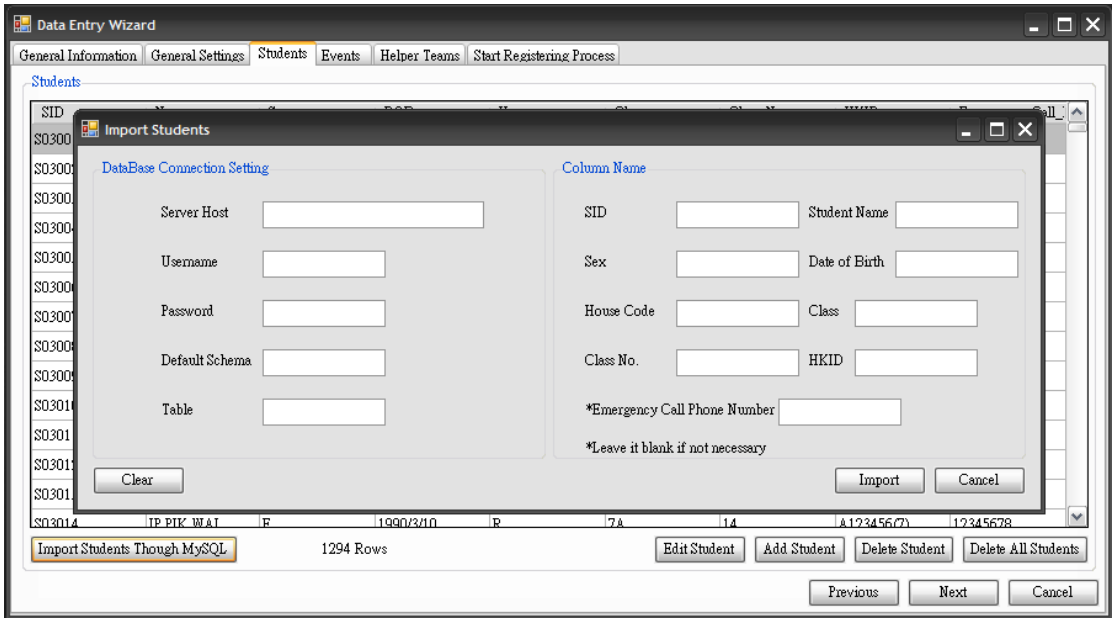
After open a project, the user has to input background data into the system, including the competition information, the scoring scheme, grades categorizing criteria, house details, students' profiles, event details and helper team details. These are the necessary data to input in the initial stage. So, I put the forms for entering these data into a single window, called 'Data Entry Wizard'. It concentrated the inputting forms together. However, these data are very massive, so I built an 'Import students through MySQL', 'Add event wizard' and 'Default Helper Team' to help the user in inputting data into the system. The layout design of these form are shown in below.



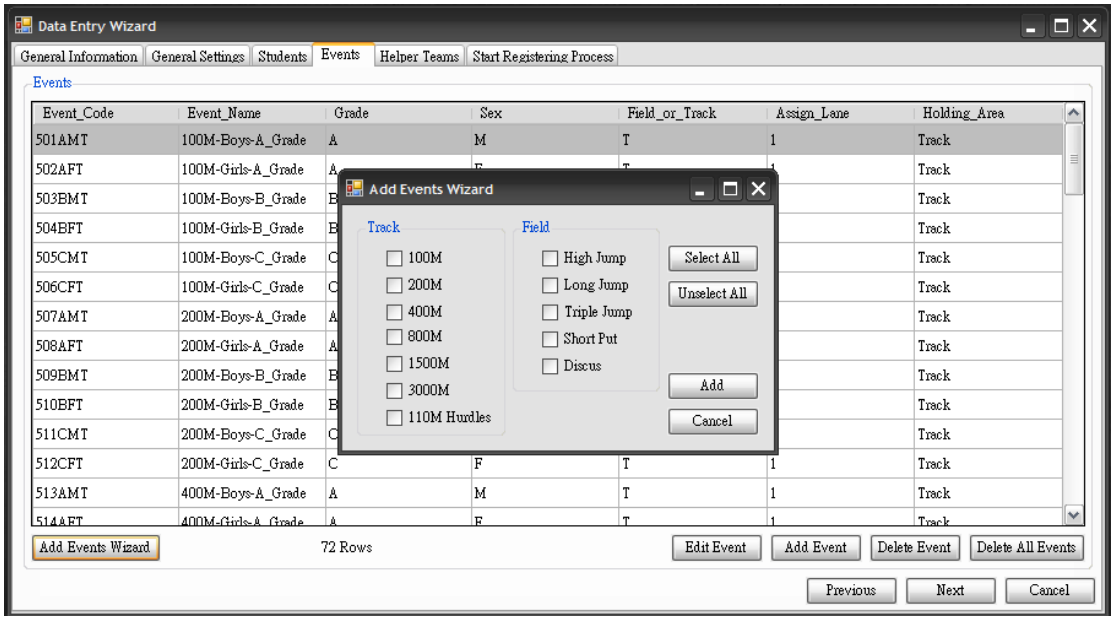
This is the page for entering competition information. For the need of source code, please refer to form6 in the source code attached.



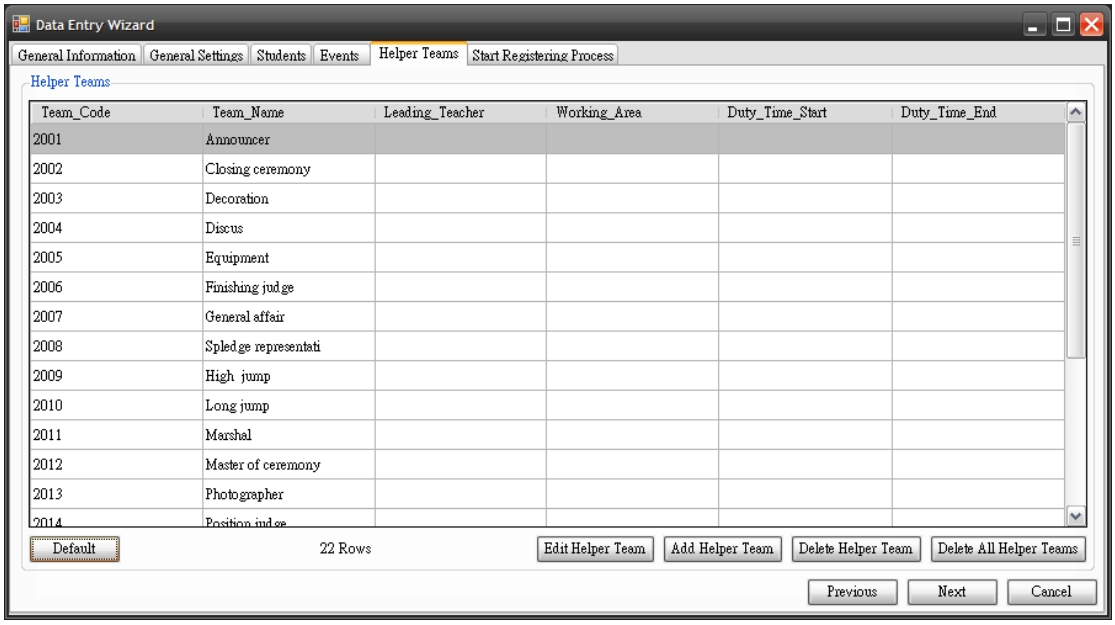
This is the page for entering the scoring scheme, grades categorizing criteria and house details. For the need of source code, please refer to form6 and form7 in the source code attached.



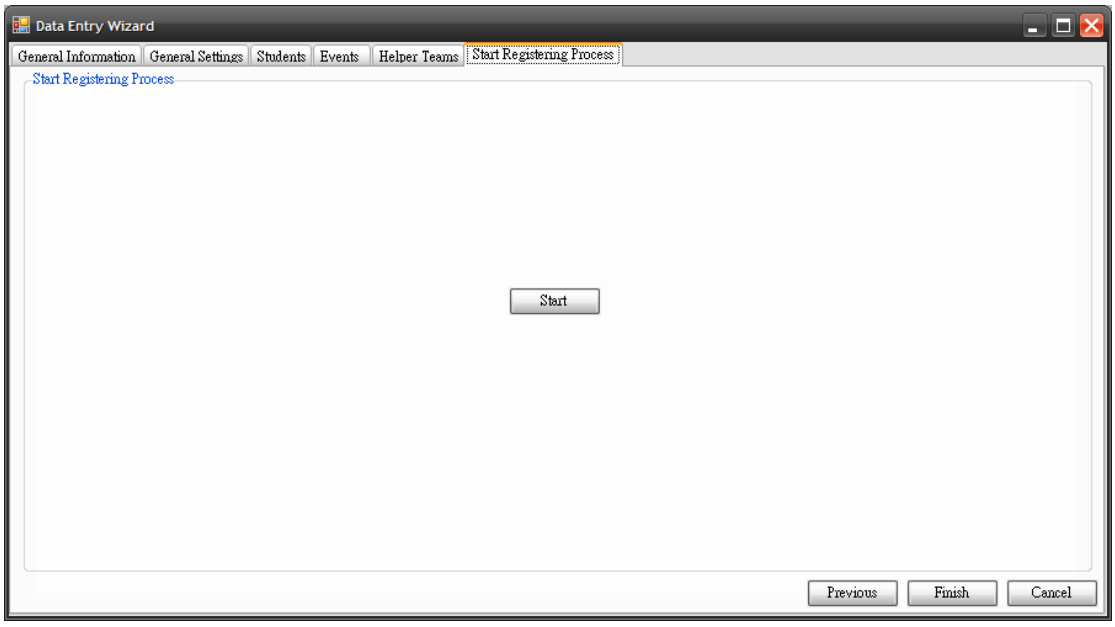
This is the ‘Import student through MySQL’ window. The user can set the connection of the other MySQL server and the name of importing table, also the field name of the corresponding items. The students’ profiles are shown at the back. And some buttons for calling out the windows of add student, edit student, delete student and delete all students. The delete function will pump up a confirmation box to ask the user if sure to delete. For the need of source code, please refer to form6, form9, form12 and form15 in the source code attached.



This is the ‘Add Events Wizard’ window. The wizard prepared the most commonly seen events hold in a sports day. The user can choose the events that will hold on the sports day. This can help the user save the time using on adding events one by one. The details of the events are shown at the back. There are buttons for calling out the windows of add event, edit event, delete event and delete all events. The delete function will pump up a confirmation box to ask the user if sure to delete. For the need of source code, please refer to form6, form10, form13 and form16 in the source code attached.



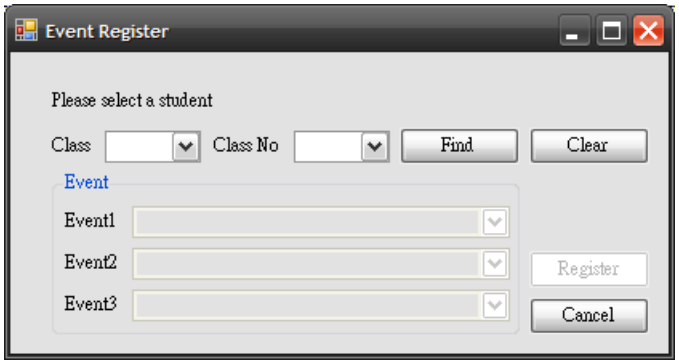
This is the page for helper team details. There is a 'Default' button at the left bottom corner. It is use to provide a default setting of helper teams. There are also buttons for add helper team, edit helper team, delete helper team and delete all helper teams. The delete function will pump up a confirmation box to ask the user if sure to delete. For the need of source code, please refer to form6, form11 and form14 in the source code attached.



This is the page for the user confirming to start the registration process. By clicking the button 'start', a few tables will be created. For the need of source code, please refer to form6 'button21' in the source code attached.

Event Register

After the 'Data Entry Wizard', the registration process will be carried out. The event register is the form for performing such task. Here is the layout design of it.



The user has to choose the class and class number of the student first, and then find out the student. The event register will categorize the selected student into different grades, A, B, or C and filter out the events that the student can join. There are only three selections for one student. The button 'register' will be enabled when a student is picked out and the user can finish the registration of the selected student by clicking it. Here is the programming code of button 'find'.

```
        If ComboBox1.SelectedIndex <> -1 And ComboBox5.SelectedIndex <> -1 Then
            Dim dt As New DataTable
            Dim cmdstr As String = "select * from " + Form1.Proj + "_student s, (select SID, 'C' as Grade
from " + Form1.Proj + "_student where DOB between (select subdate((b_grade_time),-1) from SDMCompInfo where
proj_code ='" + Form1.Proj + "') and (select c_grade_time from SDMCompInfo where proj_code ='" + Form1.Proj
+ "') union select SID, 'B' as Grade from " + Form1.Proj + "_student where DOB between (select
subdate((a_grade_time),-1) from SDMCompInfo where proj_code ='" + Form1.Proj + "') and (select b_grade_time
from SDMCompInfo where proj_code ='" + Form1.Proj + "') union select SID, 'A' as Grade from " + Form1.Proj
+ "_student where DOB between '1980/1/1' and (select a_grade_time from SDMCompInfo where proj_code ='" +
Form1.Proj + "')) g where s.SID=g.SID and Class ='" + ComboBox1.Text + "' and Class_No='" + ComboBox5.Text
+ "'"
            Form1.fill_table(cmdstr, dt)
            If dt.Rows.Count <> 1 Then
                MsgBox("student profile is invalid")
            Else
                sid = dt.Rows(0)("SID")
                grade = dt.Rows(0)("Grade")
                sex = dt.Rows(0)("Sex")
                Label3.Text = "Sex: " + sex + " Grade: " + grade + " " + dt.Rows(0)("Name")
                cmdstr = "select * from " + Form1.Proj + "_eventregister where SID ='" + sid + "'"
                dt.Clear()
                Form1.fill_table(cmdstr, dt)
                ComboBox2.Text = Form1.esstr_exe("select Event_Name from " + Form1.Proj + "_event where
Event_Code ='" + dt.Rows(0)("Event1") + "'")
                ComboBox3.Text = Form1.esstr_exe("select Event_Name from " + Form1.Proj + "_event where
Event_Code ='" + dt.Rows(0)("Event2") + "'")
                ComboBox4.Text = Form1.esstr_exe("select Event_Name from " + Form1.Proj + "_event where
Event_Code ='" + dt.Rows(0)("Event3") + "'")
                ReDim e_code(1000)
                read_event(e_code, ComboBox2)
                ReDim e_code2(1000)
                read_event(e_code2, ComboBox3)
                ReDim e_code3(1000)
                read_event(e_code3, ComboBox4)
```

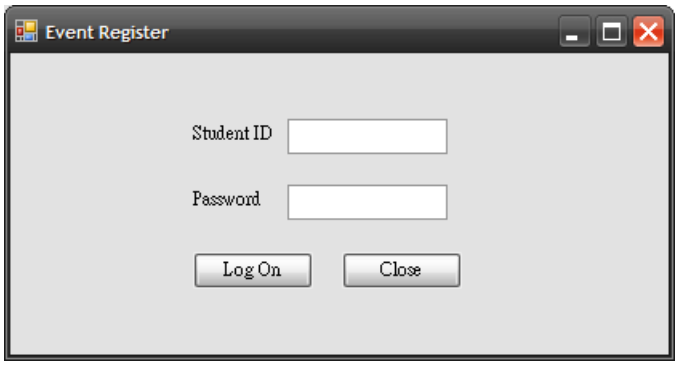
```
            ComboBox2.Enabled = True
            ComboBox3.Enabled = True
            ComboBox4.Enabled = True
            Button1.Enabled = True
        End If
    Else
        MsgBox("Invalid entry")
    End If
```

Combobox1 it the combo box of class and combobox5 is the combo box of class number. The system will generate a database query sentence for finding out the corresponding student's name, sex and categorized grade. And then the system will filter out the events that the selected student can join and insert into combobox2, 3 and 4. The programming code of function 'fill_table' and 'read event' can be found in the source code attached, form1 and form 17. If any inappropriate entry, the system will show a message box to notice the user. Here is the programming code of button 'register'.

```
    If ComboBox2.SelectedIndex <> -1 And ComboBox3.SelectedIndex <> -1 And ComboBox4.SelectedIndex <> -1 And ComboBox2.SelectedIndex <> ComboBox3.SelectedIndex And ComboBox3.SelectedIndex <> ComboBox4.SelectedIndex And ComboBox2.SelectedIndex <> ComboBox4.SelectedIndex Then
        Dim cmdstr As String = "update " + Form1.Proj + "_eventregister set Event1='" + e_code(ComboBox2.SelectedIndex) + "', Event2='" + e_code2(ComboBox3.SelectedIndex) + "', Event3='" + e_code3(ComboBox4.SelectedIndex) + "' where SID='" + sid + "'"
        Form1.enq_exe(cmdstr)
        MsgBox("registered events successfully ")
        Label13.Text = "Please select a student"
        End If
    Else
        MsgBox("Invalid entry")
    End If
```

The initial 'if' statement is for preventing invalid entry and if the condition is not fulfilled, the system will return a message 'invalid entry' to the user. The system will generate a SQL statement if no invalid entry and pass it to the database server. The programming code of function 'enq_exe' can be found in the source code attached, form 1. And then, the system will show a message box to the user for successfully registered.

Moreover, the user can let students to register the events their own. I have designed a sub-program for it. Here is the layout design of it.

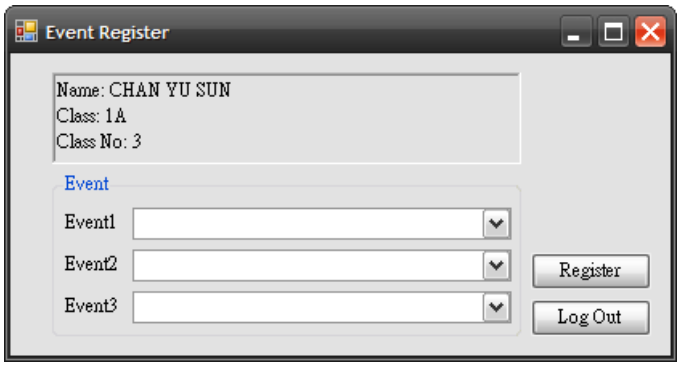


It is the student log on screen. Students are required to enter their student id and their own HKID as password to log on. Here is the programming code of button 'log on'.

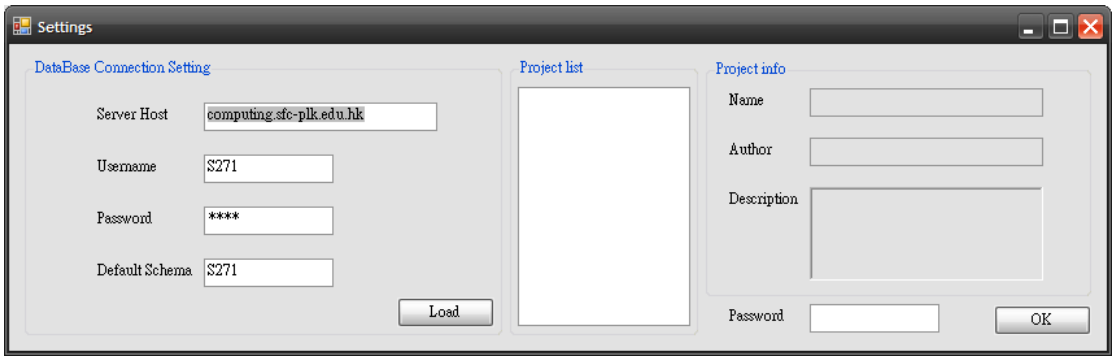
```
        If TextBox1.Text.ToUpper = "SETTINGS" Then
            If TextBox2.Text = "sportsdaymanager" Then
                Form2.Show()
            Else
                MsgBox("User Not Founnd")
            End If
        Else
            Try
                esstr_exe("select 'True' from " + Me.Proj + "_student")
            Catch ex As Exception
                MsgBox(ex.ToString)
                Exit Try
            End Try
            If esstr_exe("select 'True' from " + Me.Proj + "_student where '" + TextBox1.Text + "' in (select
SID from " + Me.Proj + "_student)") = "True" Then
                If TextBox2.Text = esstr_exe("select HKID from " + Me.Proj + "_student where SID ='" +
TextBox1.Text + "'") Then
                    Panel1.Visible = False
                    Panel2.Visible = True
                    Dim dt As New DataTable
                    Dim cmdstr As String = "select s.SID, Name, Class, Class_No, Sex, DOB, House_Name, Grade
from " + Me.Proj + "_student s, " + Me.Proj + "_house h, (select SID, 'C' as Grade from " + Me.Proj + "_student
where DOB between (select subdate((b_grade_time),-1) from SDMCompInfo where proj_code ='" + Me.Proj + "')
and (select c_grade_time from SDMCompInfo where proj_code ='" + Me.Proj + "') union select SID, 'B' as Grade
from " + Me.Proj + "_student where DOB between (select subdate((a_grade_time),-1) from SDMCompInfo where
proj_code ='" + Me.Proj + "') and (select b_grade_time from SDMCompInfo where proj_code ='" + Me.Proj + "')
union select SID, 'A' as Grade from " + Me.Proj + "_student where DOB between '1980/1/1' and (select
a_grade_time from SDMCompInfo where proj_code ='" + Me.Proj + "')) g where s.SID=g.SID and
s.House=h.House_Code and s.SID ='" + TextBox1.Text + "'"
                    Me.fill_table(cmdstr, dt)
                    sid = dt.Rows(0)("SID")
                    grade = dt.Rows(0)("Grade")
                    sex = dt.Rows(0)("Sex")
                    RichTextBox1.Text = "Name: " + dt.Rows(0)("Name") + vbNewLine + "Class: " +
dt.Rows(0)("Class") + vbNewLine + "Class No: " + dt.Rows(0)("Class_No")
                    cmdstr = "select * from " + Me.Proj + "_eventregister where SID ='" + sid + "'"
                    dt.Clear()
                    Me.fill_table(cmdstr, dt)
                    ComboBox2.Text = Me.esstr_exe("select Event_Name from " + Me.Proj + "_event where
Event_Code ='" + dt.Rows(0)("Event1") + "'")
                    ComboBox3.Text = Me.esstr_exe("select Event_Name from " + Me.Proj + "_event where
Event_Code ='" + dt.Rows(0)("Event2") + "'")
```

```
                    ComboBox4.Text = Me.esstr_exe("select Event_Name from " + Me.Proj + "_event where
Event_Code ='" + dt.Rows(0)("Event3") + "'")
                    ReDim e_code(1000)
                    read_event(e_code, ComboBox2)
                    ReDim e_code2(1000)
                    read_event(e_code2, ComboBox3)
                    ReDim e_code3(1000)
                    read_event(e_code3, ComboBox4)
                Else
                    MsgBox("Incorrect Password")
                End If
            Else
                MsgBox("User Not Found")
            End If
        End If
```

The first few rows of codes are used for calling out the setting form which I will introduce below. The 'try' statement is used for testing the connection setting is valid and successfully connected to the database server. And then the system will check the student id and password whether they are valid or not. If they are valid, the system will do the same thing as in 'event register'.



Here is an example of a student. Same as 'event register' for the organizer use, it will filter out the events that the student can join and submit the registration by clicking button 'register' and the system will automatically log out after successfully registered. For the need of source code, please refer to eventregister form1 in the source code attached.

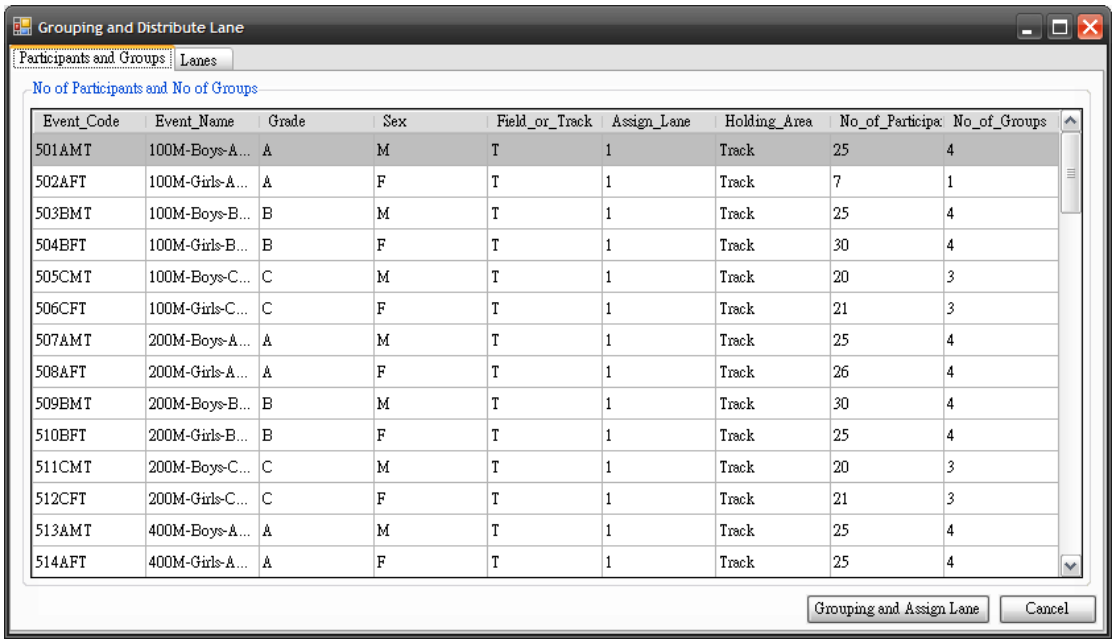


This form is used for setting up the student use event register. To call it out, the user can enter 'SETTINGS' as student id and 'sportsdaymanager' as password and click log on, the system will show this form. By setting up the connection, the project list will be shown in the list box on the right hand side. For the need of source code, please refer to eventregister form2 in the source code attached.

There is also a helper register, the use of it is same as the event register, but only on helper team for one students. And it does the similar task with 'eventregister'. For the need of source code, please refer to form18 in the source code attached.

Grouping and Distribute Lane

After the registration process, the user can start the grouping and distribution of lanes. Here is the layout design of it.



Event_Code	Event_Name	Grade	Sex	Field_or_Track	Assign_Lane	Holding_Area	No_of_Participa	No_of_Groups
501AMT	100M-Boys-A...	A	M	T	1	Track	25	4
502AFT	100M-Girls-A...	A	F	T	1	Track	7	1
503BMT	100M-Boys-B...	B	M	T	1	Track	25	4
504BFT	100M-Girls-B...	B	F	T	1	Track	30	4
505CMT	100M-Boys-C...	C	M	T	1	Track	20	3
506CFT	100M-Girls-C...	C	F	T	1	Track	21	3
507AMT	200M-Boys-A...	A	M	T	1	Track	25	4
508AFT	200M-Girls-A...	A	F	T	1	Track	26	4
509BMT	200M-Boys-B...	B	M	T	1	Track	30	4
510BFT	200M-Girls-B...	B	F	T	1	Track	25	4
511CMT	200M-Boys-C...	C	M	T	1	Track	20	3
512CFT	200M-Girls-C...	C	F	T	1	Track	21	3
513AMT	400M-Boys-A...	A	M	T	1	Track	25	4
514AFT	400M-Girls-A...	A	F	T	1	Track	25	4

It is very easy for the user to do such task because there is only one button to click and the system will generate a list of lane assignment automatically. Here is the programming code of button ‘grouping and assign lane’.

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        distribute_lanes()
    End Sub
```

This function will call out another function ‘distribute lanes’. Here is the programming code of function ‘distribute lanes’.

```
    Sub distribute_lanes()
        Dim cmd As String = "drop table if exists " + Form1.Proj + "_lane"
        Form1.enq_exe(cmd)
        Dim cmdstr As String = "create table " + Form1.Proj + "_lane (Event_Code char(6), Group_Code
char(20), Lane1 char(10) , Lane2 char(10), Lane3 char(10), Lane4 char(10), Lane5 char(10), Lane6 char(10),
Lane7 char(10), Lane8 char(10))"
        Form1.enq_exe(cmdstr)
        Dim dt As New DataTable
        cmdstr = "select Event_Code from " + Form1.Proj + "_noofgroups where Assign_Lane ='1' order by
Event_Code"
        Form1.fill_table(cmdstr, dt)
        Dim e_code(dt.Rows.Count) As String
        For i = 0 To dt.Rows.Count - 1
            e_code(i) = dt.Rows(i)("Event_Code")
        Next
        Dim n_groups As Integer
        Dim g_code As String
        For f = 0 To dt.Rows.Count - 1
            cmdstr = "select No_of_Groups from " + Form1.Proj + "_noofgroups where Event_Code ='" + e_code(f)
+ "'"
            n_groups = Form1.esint_exe(cmdstr)
```

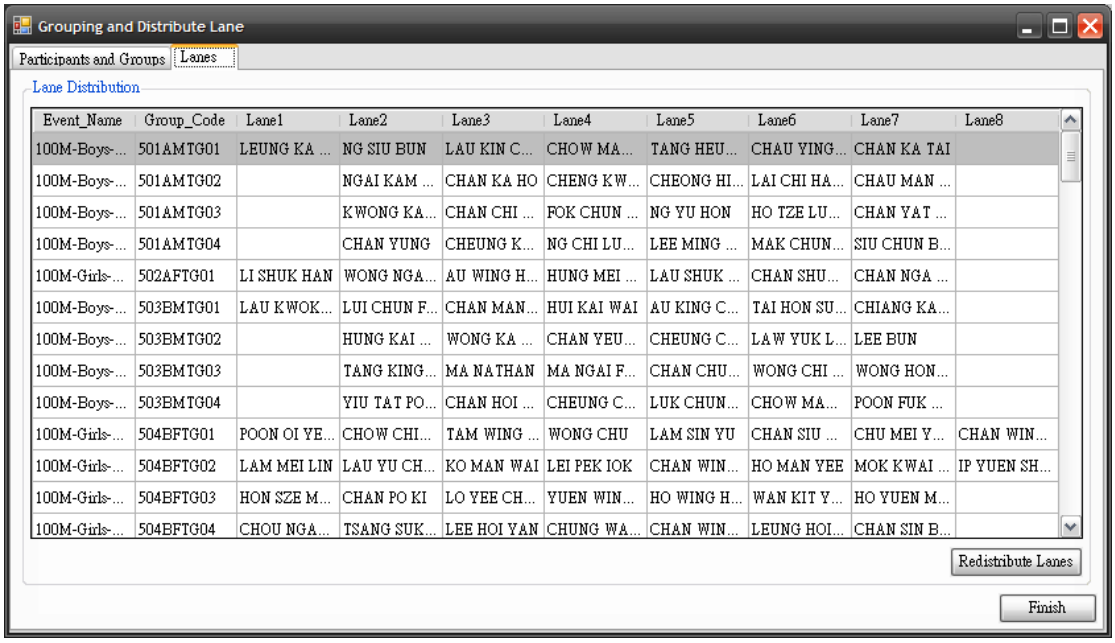
```
            cmdstr = "insert into " + Form1.Proj + "_lane (Event_Code, Group_Code) value "
            For g = 1 To n_groups
                g_code = e_code(f) + "G" + Trim(g.ToString("00"))
                cmdstr = cmdstr + "('" + e_code(f) + "','" + g_code + "'), "
            Next
            Form1.enq_exe(cmdstr.Substring(0, cmdstr.Length - 2))
        Next


        Dim h, ng, s1, s2 As Integer
        Dim tmp As String
        Dim dtsid As New DataTable
        Dim lane() As String = {"Lane4", "Lane5", "Lane3", "Lane6", "Lane2", "Lane7", "Lane1", "Lane8"}
        For f = 0 To dt.Rows.Count - 1
            cmdstr = "select SID from " + Form1.Proj + "_eventregister where Event1 ='" + e_code(f) + "'
or Event2 ='" + e_code(f) + "' or Event3 ='" + e_code(f) + "'"
            dtsid.Reset()
            Form1.fill_table(cmdstr, dtsid)
            Dim sid(dtsid.Rows.Count) As String
            For j = 0 To dtsid.Rows.Count - 1
                sid(j) = dtsid.Rows(j)("SID")
            Next
            Randomize()
            For j = 1 To 100
                Do
                    s1 = Int(Rnd() * (sid.Length - 1))
                    s2 = Int(Rnd() * (sid.Length - 1))
                Loop While s1 = s2
                tmp = sid(s1)
                sid(s1) = sid(s2)
                sid(s2) = tmp
            Next
            h = 0
            cmdstr = "select No_of_Groups from " + Form1.Proj + "_noofgroups where Event_Code ='" + e_code(f)
+ "'"
            n_groups = Form1.esint_exe(cmdstr)
            For i = 0 To 7
                ng = 1
                For ng = 1 To n_groups
                    If h > dtsid.Rows.Count - 1 Then
                        Exit For
                    Else
                        cmdstr = "update " + Form1.Proj + "_lane set " + lane(i) + "= '" + sid(h) + "' where
Group_Code ='" + Trim(e_code(f)) + "G" + Trim(ng.ToString("00")) + "'"
                        Form1.enq_exe(cmdstr)
                        h = h + 1
                    End If
                Next
            Next
        Next
    End Sub
```

The function will first create a table ‘lane’. And then insert the event code and group code into the table, where the number of groups for each heat event is calculated and stored in a view table called ‘noofgroups’. And then select the participant out from the table ‘eventregister’ and sort them randomly. The sequence of lane distribution is put in the array ‘lane()’. And then the system will update the table according to the random list of participants. And the program will make the number of participants of each heat event to be even. For example, 50 participants in 100M run, the distribution

of lane assignment will be 8 participants in group 1, 7 participants in group 2, 7 participants in group 3, 7 participants in group 4, 7 participants in group 5, 7 participants in group 6, 7 participants in group 7.

And the screen of the lane distribution is here.

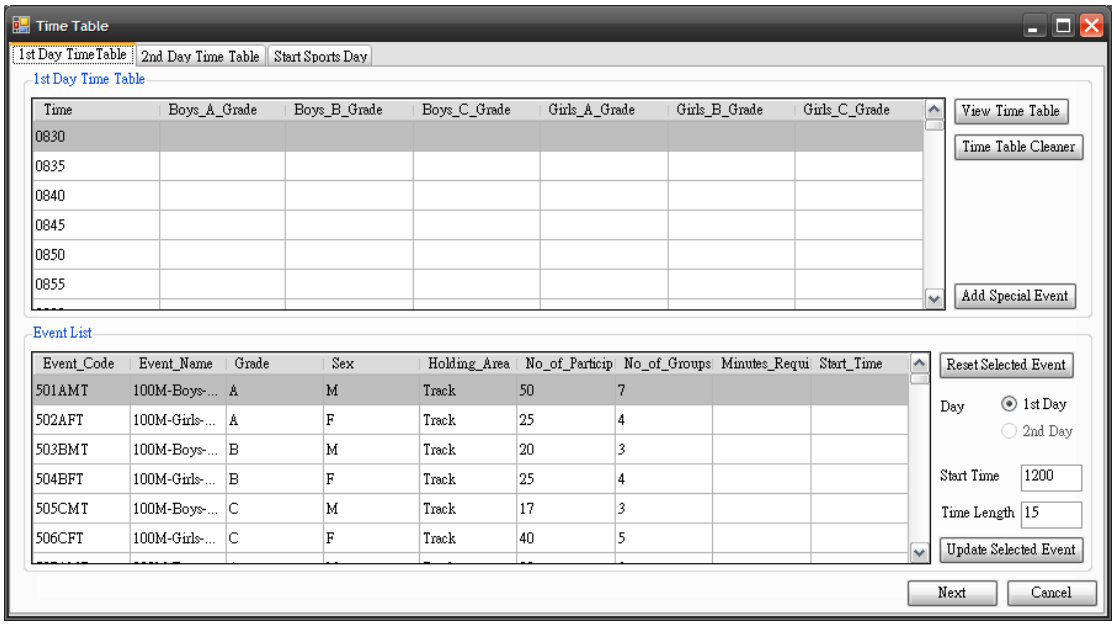


Event_Name	Group_Code	Lane1	Lane2	Lane3	Lane4	Lane5	Lane6	Lane7	Lane8
100M-Boys-...	501AMTG01	LEUNG KA ...	NG SIU BUN	LAU KIN C...	CHOW MA...	TANG HEU...	CHAU YING...	CHAN KA TAI	
100M-Boys-...	501AMTG02		NGAI KAM ...	CHAN KA HO	CHENG KW...	CHEONG HI...	LAI CHI HA...	CHAU MAN ...	
100M-Boys-...	501AMTG03		KWONG KA...	CHAN CHI ...	FOK CHUN ...	NG YU HON	HO TZE LU...	CHAN YAT ...	
100M-Boys-...	501AMTG04		CHAN YUNG	CHEUNG K...	NG CHI LU...	LEE MING ...	MAK CHUN...	SIU CHUN B...	
100M-Girls-...	502AFTG01	LI SHUK HAN	WONG NGA...	AU WING H...	HUNG MEI ...	LAU SHUK ...	CHAN SHU...	CHAN NGA ...	
100M-Boys-...	503BMTG01	LAU KWOK...	LUI CHUN F...	CHAN MAN...	HUI KAI WAI	AU KING C...	TAI HON SU...	CHIANG KA...	
100M-Boys-...	503BMTG02		HUNG KAI ...	WONG KA ...	CHAN YEU...	CHEUNG C...	LAW YUK L...	LEE BUN	
100M-Boys-...	503BMTG03		TANG KING...	MA NATHAN	MA NGAI F...	CHAN CHU...	WONG CHI ...	WONG HON...	
100M-Boys-...	503BMTG04		YIU TAT PO...	CHAN HOI ...	CHEUNG C...	LUK CHUN...	CHOW MA...	POON FUK ...	
100M-Girls-...	504BFTG01	POON OI YE...	CHOW CHI...	TAM WING ...	WONG CHU	LAM SIN YU	CHAN SIU ...	CHU MEI Y...	CHAN WIN...
100M-Girls-...	504BFTG02	LAM MEI LIN	LAU YU CH...	KO MAN WAI	LEI PEK IOK	CHAN WIN...	HO MAN YEE	MOK KWAI ...	IP YUEN SH...
100M-Girls-...	504BFTG03	HON SZE M...	CHAN PO KI	LO YEE CH...	YUEN WIN...	HO WING H...	WAN KIT Y...	HO YUEN M...	
100M-Girls-...	504BFTG04	CHOU NGA...	TSANG SUK...	LEE HOI YAN	CHUNG WA...	CHAN WIN...	LEUNG HOI...	CHAN SIN B...	

It is possible for the user to redistribute the lane assignment by clicking the button ‘redistribute lanes’. And the system will do the distribution shown in the above again. For the need of source code, please refer to form19 in the source code attached.

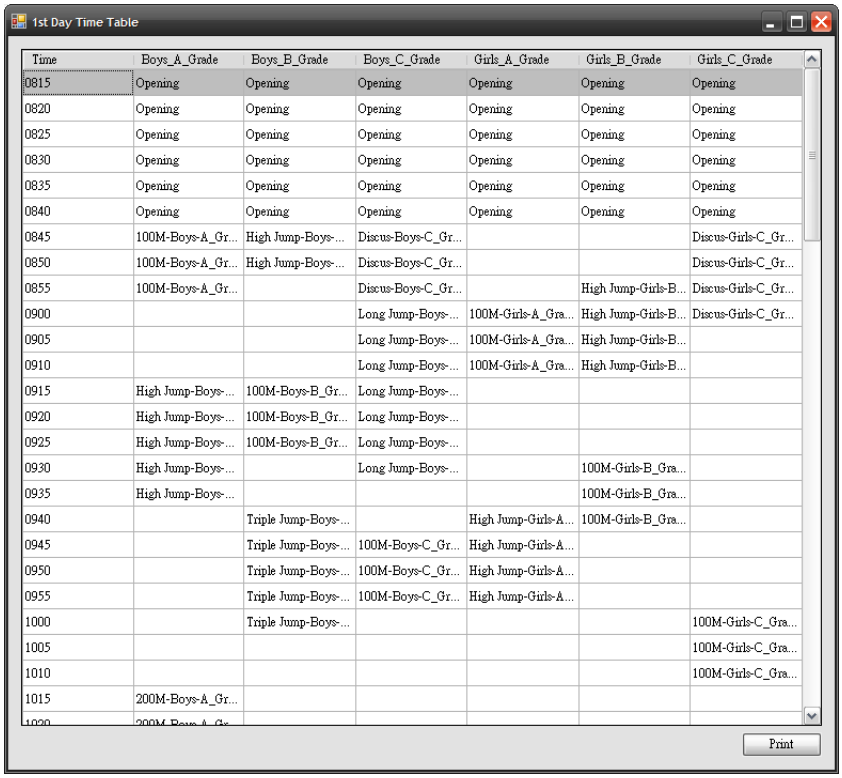
Time Table Planning

After grouping and distribution of lanes, the user can plan the time table by calling out the form ‘time table’, which can be found in ‘tools’ in the main form. Here is the layout design of it.



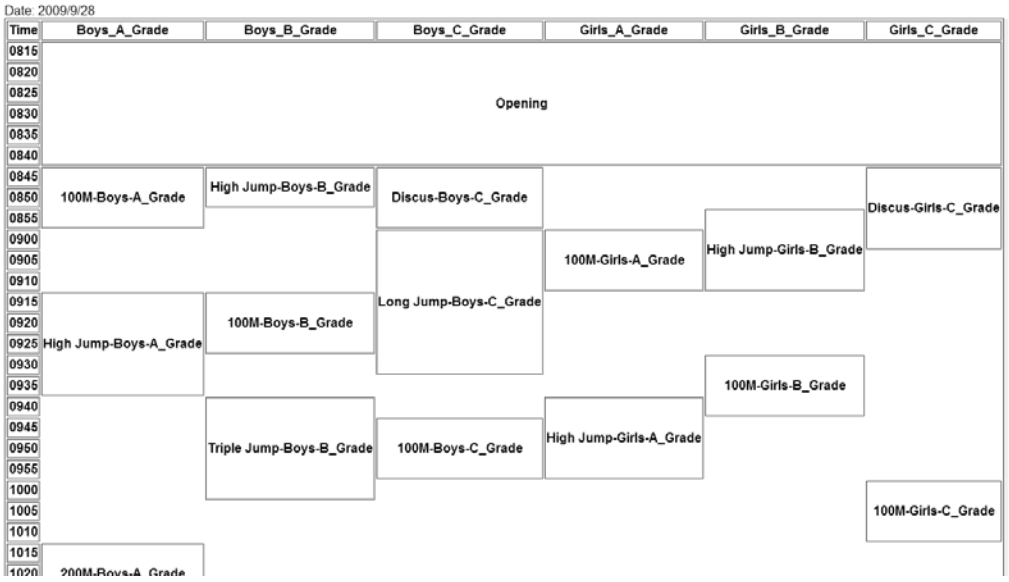
The user can view the planning time table on the upper part of the form and the planned start time and time length of events on the lower part. Users can input a start time and the time length of the selected event in the textbox provided. By clicking the button ‘update selected event’, the system will update both table event list and time table. The user can add some special event, for example, opening ceremony or lunch time, to the time table by using the function ‘add special event’. The user also can delete some items on the time table by using ‘time table cleaner’. In this time table planning, it is impossible for events with same grade and same sex holding at the same time. This means, it is impossible to have a clash of event and participant allocation. The user can have a bigger view of the time table by clicking ‘view time table’.

Here is the layout of 'view time table'.



Time	Boys_A_Grade	Boys_B_Grade	Boys_C_Grade	Girls_A_Grade	Girls_B_Grade	Girls_C_Grade
0815	Opening	Opening	Opening	Opening	Opening	Opening
0820	Opening	Opening	Opening	Opening	Opening	Opening
0825	Opening	Opening	Opening	Opening	Opening	Opening
0830	Opening	Opening	Opening	Opening	Opening	Opening
0835	Opening	Opening	Opening	Opening	Opening	Opening
0840	Opening	Opening	Opening	Opening	Opening	Opening
0845	100M-Boys-A_Gr...	High Jump-Boys-...	Discus-Boys-C_Gr...			Discus-Girls-C_Gr...
0850	100M-Boys-A_Gr...	High Jump-Boys-...	Discus-Boys-C_Gr...			Discus-Girls-C_Gr...
0855	100M-Boys-A_Gr...		Discus-Boys-C_Gr...		High Jump-Girls-B...	Discus-Girls-C_Gr...
0900			Long Jump-Boys-...	100M-Girls-A_Gra...	High Jump-Girls-B...	Discus-Girls-C_Gr...
0905			Long Jump-Boys-...	100M-Girls-A_Gra...	High Jump-Girls-B...	
0910			Long Jump-Boys-...	100M-Girls-A_Gra...	High Jump-Girls-B...	
0915	High Jump-Boys-...	100M-Boys-B_Gr...	Long Jump-Boys-...			
0920	High Jump-Boys-...	100M-Boys-B_Gr...	Long Jump-Boys-...			
0925	High Jump-Boys-...	100M-Boys-B_Gr...	Long Jump-Boys-...			
0930	High Jump-Boys-...		Long Jump-Boys-...		100M-Girls-B_Gra...	
0935	High Jump-Boys-...				100M-Girls-B_Gra...	
0940		Triple Jump-Boys-...		High Jump-Girls-A...	100M-Girls-B_Gra...	
0945		Triple Jump-Boys-...	100M-Boys-C_Gr...	High Jump-Girls-A...		
0950		Triple Jump-Boys-...	100M-Boys-C_Gr...	High Jump-Girls-A...		
0955		Triple Jump-Boys-...	100M-Boys-C_Gr...	High Jump-Girls-A...		
1000		Triple Jump-Boys-...				100M-Girls-C_Gra...
1005						100M-Girls-C_Gra...
1010						100M-Girls-C_Gra...
1015	200M-Boys-A_Gr...					

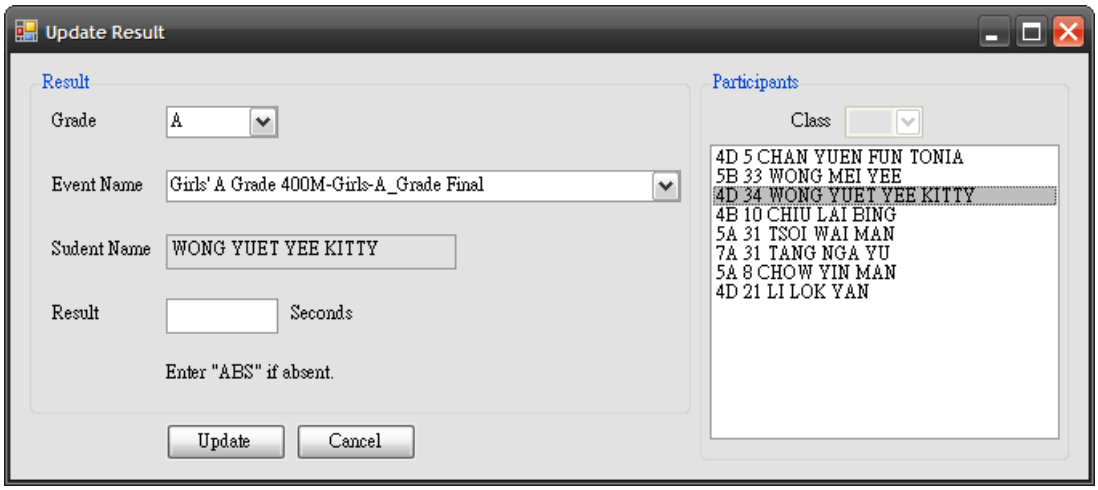
The user can click the button at right bottom corner to generate an HTML file and open it in a browser. Here is an example.



The user can use the print function of the browser to print out the time table and publish to students. For the need of source code, please refer to form20, form21, form24 and form25 in the source code attached. For the function of generating the html file, please refer to form1 'print_html' in the source code attached.

Update Result

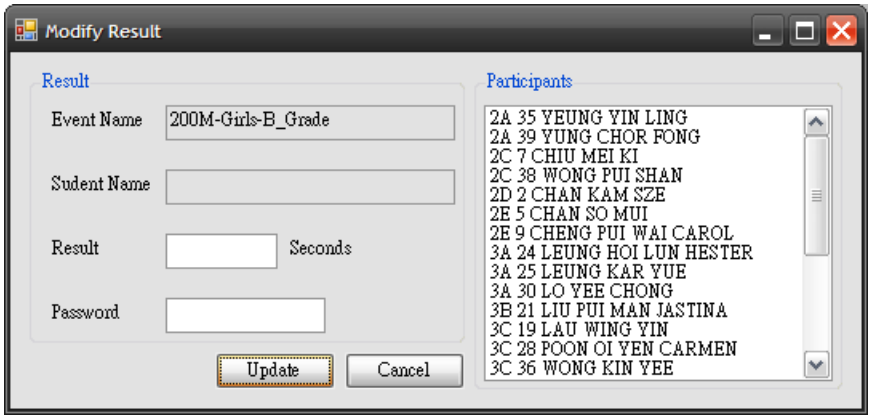
After planned the time table, the user can wait for the day holding the sports day. During the running of the sports day, once an event finished, the user can input the results into the system by using the function 'update result'. Here is the layout design of it.




```
        If TextBox2.Text <> "" And TextBox1.Text <> "" Then
            Dim cmdstr As String
            If TextBox1.Text.ToUpper = "ABS" Then
                cmdstr = "insert into " + Form1.Proj + "_results(SID, Event_Code, Absent) value ('" +
sid(ListBox1.SelectedIndex) + "', '" + e_code(ComboBox1.SelectedIndex) + "', '" + TextBox1.Text + "')"
            Else
                cmdstr = "insert into " + Form1.Proj + "_results(SID, Event_Code, Result) value ('" +
sid(ListBox1.SelectedIndex) + "', '" + e_code(ComboBox1.SelectedIndex) + "', '" + TextBox1.Text + "')"
            End If
            Try
                Form1.enq_exe(cmdstr)
            Catch ex As Exception
                MsgBox("Dupicate or Inappropriate Result Entered")
            End Try
            MsgBox("Updated Successfully")
            ListBox1.SelectedIndex = 0
            TextBox2.Clear()
        End If
```

The user can input the result of the selected participant into the textbox and if the participant absented in the event, user can input 'ABS' instead of his/her result. For the need of source code, please refer to form22 in the source code attached. Moreover, once the result is submitted, the user cannot modify it by this form since avoiding cheatings. I designed another program to modify the result and such program requires password before modifying the result.

Here is the layout design of 'modify result'.

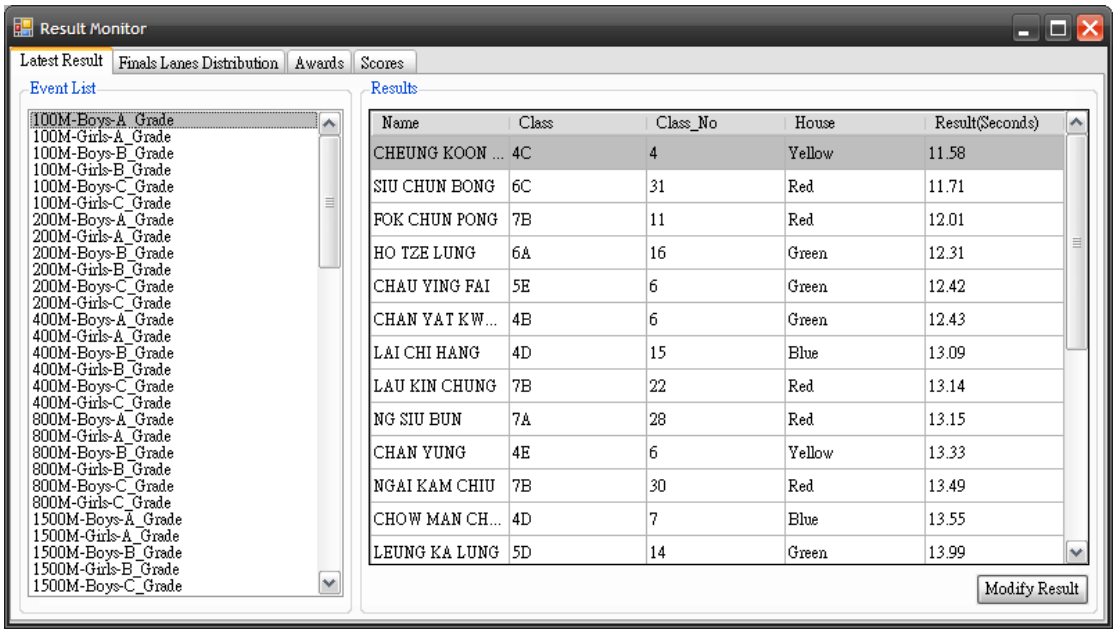



```
        If TextBox4.Text = Form1.esstr_exe("select pw from SDMProj where Proj_Code ='" + Form1.Proj + "'") Then
            Dim cmdstr As String
            If TextBox1.Text.ToUpper = "ABS" Then
                cmdstr = "update " + Form1.Proj + "_results set Result ='', absent='ABS' where SID='" + sid(ListBox1.SelectedIndex) + "' and Event_Code ='" + Form23.f23_e_code(Form23.ListBox1.SelectedIndex) + "'"
            Else
                cmdstr = "update " + Form1.Proj + "_results set Result ='" + TextBox1.Text + "', absent='' where SID='" + sid(ListBox1.SelectedIndex) + "' and Event_Code ='" + Form23.f23_e_code(Form23.ListBox1.SelectedIndex) + "'"
            End If
            Form1.enq_exe(cmdstr)
            MsgBox("Modified Successfully")
        Else
            MsgBox("Incorrect Password")
        End If
```

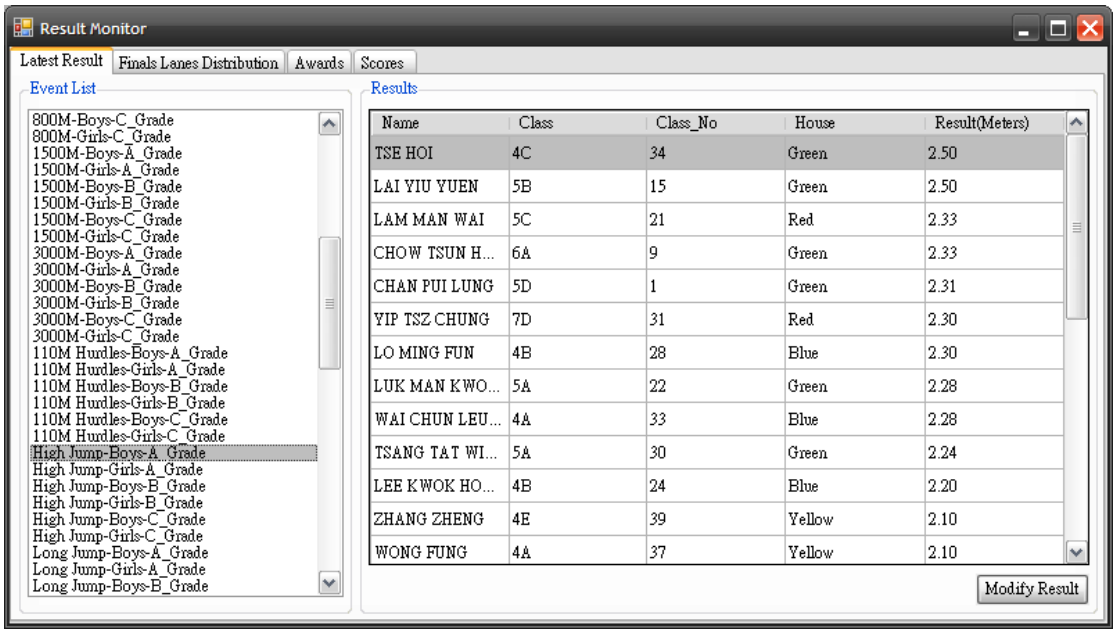
It is same as 'update result'. The user selects a student and then input a new result and password of the project. For the need of source code, please refer to form26 in the source code attached. This form can be found in the form 'result monitor', which I will introduce in below.

Result Monitor (Results Report)

After or during the process of inputting results, the user can use the form 'result monitor' to get the updated results, the lane assignment of the finals, the awards and the score of personals and houses. Here is the layout design of it.



We can find the button 'modify result', which will call out the function I talked in the above. In this screen, the results of track events are sorted by their results in ascending order.

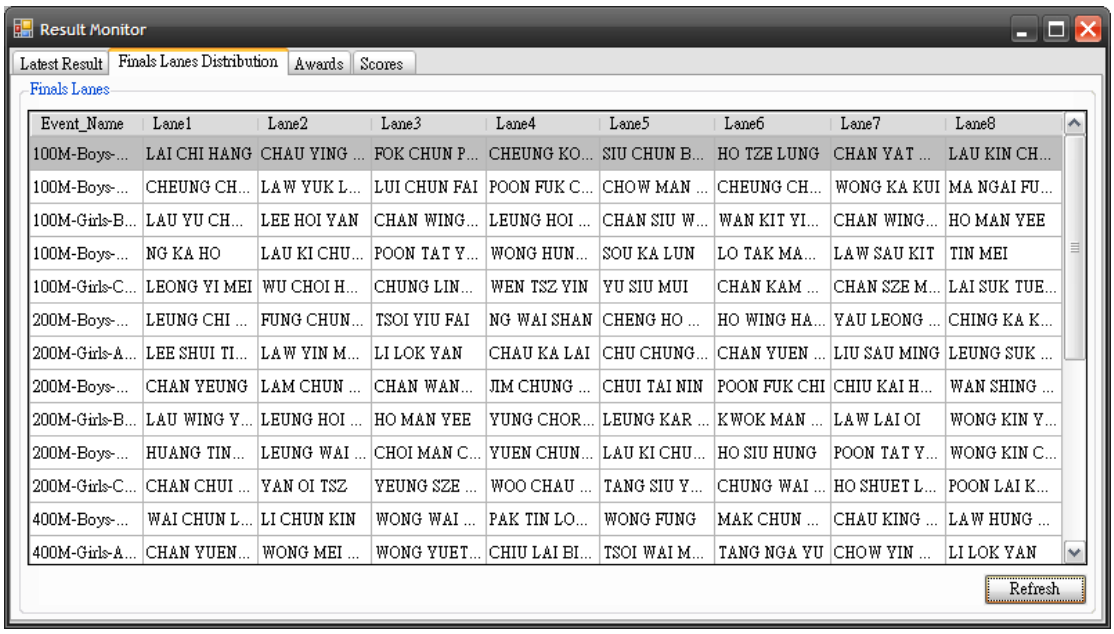


And the field events are sorted in descending order. Here is the programming code of it.

```
            If f23_e_code(ListBox1.SelectedIndex).Substring(f23_e_code(ListBox1.SelectedIndex).Length -
1, 1) = "T" Then
```

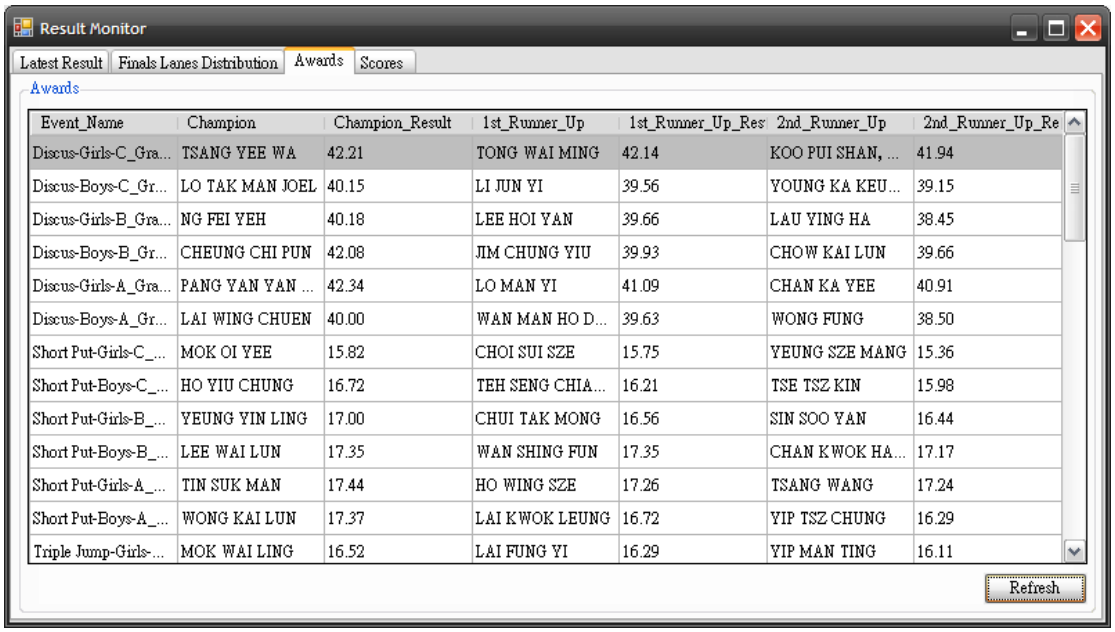
```
            Form1.dgv_show2(Me.DataGridView1, "select Name, Class, Class_No, House_Name as House,
Result as 'Result(Seconds)' from " + Form1.Proj + "_student s, " + Form1.Proj + "_house h, " + Form1.Proj
+ "_results r where s.SID=r.SID and s.House=h.House_Code and Event_Code ='" +
f23_e_code(ListBox1.SelectedIndex) + "' and Absent <>'ABS' order by Result asc")
        Else
            Form1.dgv_show2(Me.DataGridView1, "select Name, Class, Class_No, House_Name as House,
Result as 'Result(Meters)' from " + Form1.Proj + "_student s, " + Form1.Proj + "_house h, " + Form1.Proj
+ "_results r where s.SID=r.SID and s.House=h.House_Code and Event_Code ='" +
f23_e_code(ListBox1.SelectedIndex) + "' and Absent <>'ABS' order by Result desc")
        End If
```

This function generates a SQL statement according to the event code (e_code) of the selected event, which can indicate whether the selected event is a track or field event. For the need of source code, please refer to form1 ‘dgv_show2’ and form23 in the source code attached.



Event_Name	Lane1	Lane2	Lane3	Lane4	Lane5	Lane6	Lane7	Lane8
100M-Boys-...	LAI CHI HANG	CHAU YING ...	FOK CHUN P...	CHEUNG KO...	SIU CHUN B...	HO TZE LUNG	CHAN YAT ...	LAU KIN CH...
100M-Boys-...	CHEUNG CH...	LAW YUK L...	LUI CHUN FAI	POON FUK C...	CHOW MAN ...	CHEUNG CH...	WONG KA KUI	MA NGAI FU...
100M-Girls-B...	LAU YU CH...	LEE HOI YAN	CHAN WING...	LEUNG HOI ...	CHAN SIU W...	WAN KIT YI...	CHAN WING...	HO MAN YEE
100M-Boys-...	NG KA HO	LAU KI CHU...	POON TAT Y...	WONG HUN...	SOU KA LUN	LO TAK MA...	LAW SAU KIT	TIN MEI
100M-Girls-C...	LEONG YI MEI	WU CHOI H...	CHUNG LIN...	WEN TSZ YIN	YU SIU MUI	CHAN KAM ...	CHAN SZE M...	LAI SUK TUE...
200M-Boys-...	LEUNG CHI ...	FUNG CHUN...	TSOI YIU FAI	NG WAI SHAN	CHENG HO ...	HO WING HA...	YAU LEONG ...	CHING KA K...
200M-Girls-A...	LEE SHUI TI...	LAW YIN M...	LI LOK YAN	CHAU KA LAI	CHU CHUNG...	CHAN YUEN ...	LIU SAU MING	LEUNG SUK ...
200M-Boys-...	CHAN YEUNG	LAM CHUN ...	CHAN WAN...	JIM CHUNG ...	CHUI TAI NIN	POON FUK CHI	CHIU KAI H...	WAN SHING ...
200M-Girls-B...	LAU WING Y...	LEUNG HOI ...	HO MAN YEE	YUNG CHOR...	LEUNG KAR ...	KWOK MAN ...	LAW LAI OI	WONG KIN Y...
200M-Boys-...	HUANG TIN...	LEUNG WAI ...	CHOI MAN C...	YUEN CHUN...	LAU KI CHU...	HO SIU HUNG	POON TAT Y...	WONG KIN C...
200M-Girls-C...	CHAN CHUI ...	YAN OI TSZ	YEUNG SZE ...	WOO CHAU ...	TANG SIU Y...	CHUNG WAI ...	HO SHUET L...	POON LAI K...
400M-Boys-...	WAI CHUN L...	LI CHUN KIN	WONG WAI ...	PAK TIN LO...	WONG FUNG	MAK CHUN ...	CHAU KING ...	LAW HUNG ...
400M-Girls-A...	CHAN YUEN...	WONG MEI ...	WONG YUET...	CHIU LAI BI...	TSOI WAI M...	TANG NGA YU	CHOW YIN ...	LI LOK YAN

It is the screen of lane distribution for finals. It shows the participants and their assigned lane. The lane assignment is based on their position of the heat events. The first one will be assigned in lane 4, second in lane 5, third in lane3…and the last in lane 8. For the need of source code, please refer to form23 in the source code attached.



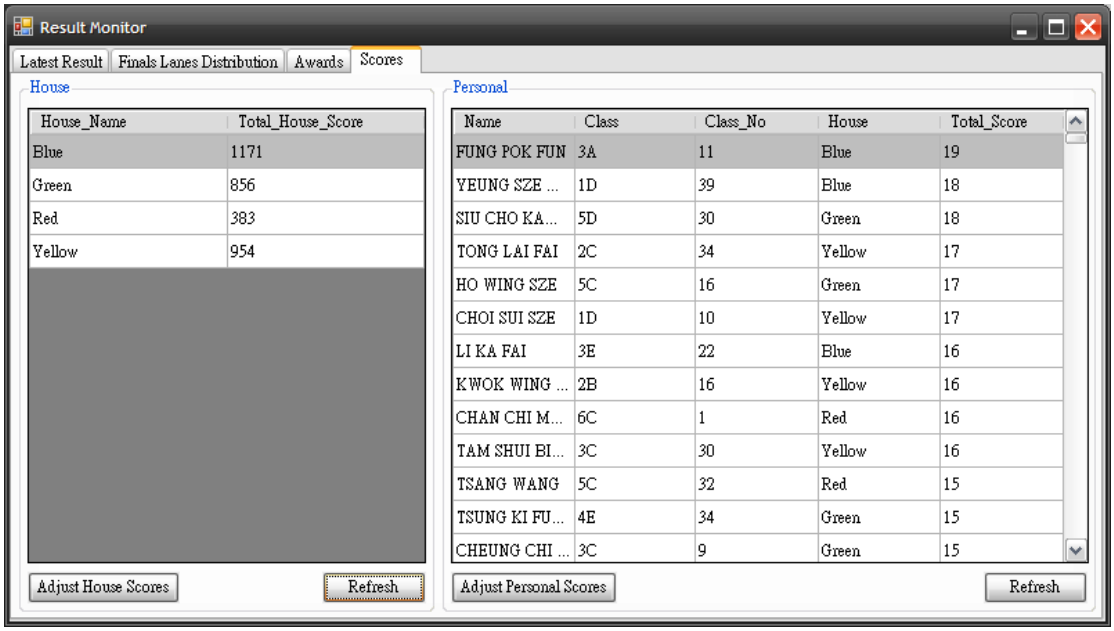
Event_Name	Champion	Champion_Result	1st_Runner_Up	1st_Runner_Up_Res	2nd_Runner_Up	2nd_Runner_Up_Re
Discus-Girls-C_Gra...	TSANG YEE WA	42.21	TONG WAI MING	42.14	KOO PUI SHAN, ...	41.94
Discus-Boys-C_Gr...	LO TAK MAN JOEL	40.15	LI JUN YI	39.56	YOUNG KA KEU...	39.15
Discus-Girls-B_Gra...	NG FEI YEH	40.18	LEE HOI YAN	39.66	LAU YING HA	38.45
Discus-Boys-B_Gr...	CHEUNG CHI PUN	42.08	JIM CHUNG YIU	39.93	CHOW KAI LUN	39.66
Discus-Girls-A_Gra...	PANG YAN YAN ...	42.34	LO MAN YI	41.09	CHAN KA YEE	40.91
Discus-Boys-A_Gr...	LAI WING CHUEN	40.00	WAN MAN HO D...	39.63	WONG FUNG	38.50
Short Put-Girls-C_...	MOK OI YEE	15.82	CHOI SUI SZE	15.75	YEUNG SZE MANG	15.36
Short Put-Boys-C_...	HO YIU CHUNG	16.72	TEH SENG CHIA...	16.21	TSE TSZ KIN	15.98
Short Put-Girls-B_...	YEUNG YIN LING	17.00	CHUI TAK MONG	16.56	SIN SOO YAN	16.44
Short Put-Boys-B_...	LEE WAI LUN	17.35	WAN SHING FUN	17.35	CHAN KWOK HA...	17.17
Short Put-Girls-A_...	TIN SUK MAN	17.44	HO WING SZE	17.26	TSANG WANG	17.24
Short Put-Boys-A_...	WONG KAI LUN	17.37	LAI KWOK LEUNG	16.72	YIP TSZ CHUNG	16.29
Triple Jump-Girls-...	MOK WAI LING	16.52	LAI FUNG YI	16.29	YIP MAN TING	16.11

It is the screen showing the winners of events. By clicking the button ‘refresh’, the system will compute the results once more and give an updated award list to the user. Here is the programming code of it.

```
        Dim cmdstr As String = "drop table if exists " + Form1.Proj + "_award"
        Form1.enq_exe(cmdstr)
        cmdstr = "create table " + Form1.Proj + "_award (Event_Code char(7), Champion char(10), 1st_Runner_Up
char(10), 2nd_Runner_Up char(10))"
        Form1.enq_exe(cmdstr)
        cmdstr = "select Event_Code from " + Form1.Proj + "_event where Assign_Lane ='0' and Event_Code
<>'100NNN' union select Event_Code from " + Form1.Proj + "_finals"
        Dim dt As New DataTable
        Form1.fill_table(cmdstr, dt)
        For i = 0 To dt.Rows.Count - 1
            If dt.Rows(i)("Event_Code").ToString.Substring(dt.Rows(i)("Event_Code").ToString.Length - 1,
1) = "T" Then
                cmdstr = "select SID from " + Form1.Proj + "_results where Absent not like 'abs' and Event_Code
='" + dt.Rows(i)("Event_Code") + "' order by Result asc limit 0,3"
            Else
                cmdstr = "select SID from " + Form1.Proj + "_results where Absent not like 'abs' and Event_Code
='" + dt.Rows(i)("Event_Code") + "' order by Result desc limit 0,3"
            End If
            Dim rank As New DataTable
            Form1.fill_table(cmdstr, rank)
            If rank.Rows.Count >= 3 Then
                cmdstr = "insert into " + Form1.Proj + "_award (Event_Code) value ('" +
dt.Rows(i)("Event_Code") + "')"
                Form1.enq_exe(cmdstr)
                cmdstr = "update " + Form1.Proj + "_award set Champion= '" + rank.Rows(0)("SID") + "',
1st_Runner_Up= '" + rank.Rows(1)("SID") + "', 2nd_Runner_Up= '" + rank.Rows(2)("SID") + "' where Event_Code
='" + dt.Rows(i)("Event_Code") + "'"
                Form1.enq_exe(cmdstr)
            End If
        Next
        Form1.dgv_show2(Me.DataGridView3, "SELECT Event_Name, c.Name as Champion, cr.result as
Champion_Result, 1st.Name as 1st_Runner_Up, 1str.result as 1st_Runner_Up_Result, 2nd.Name as 2nd_Runner_Up,
2ndr.result as 2nd_Runner_Up_Result FROM " + Form1.Proj + "_award a left join (select Event_Code, Event_Name
from " + Form1.Proj + "_event union select Event_Code, Event_Name from " + Form1.Proj + "_finals)e on
a.Event_Code=e.Event_Code left join " + Form1.Proj + "_student c on a.Champion=c.SID left join " + Form1.Proj
+ "_results cr on a.Champion=cr.SID and cr.event_code=a.event_code left join " + Form1.Proj + "_student 1st
on a.1st_Runner_Up=1st.SID left join " + Form1.Proj + "_results 1str on a.1st_Runner_Up=1str.SID and
```

```
lstr.event_code=a.event_code left join " + Form1.Proj + "_student 2nd on a.2nd_Runner_Up=2nd.SID left join " + Form1.Proj + "_results 2ndr on a.2nd_Runner_Up=2ndr.SID and 2ndr.event_code=a.event_code")
```

This function will create a table 'award'. And then select out the events do not need to assign lane, which have no heat events. And then find out the top three participants and put them into the table, champion, 1st runner up and 2nd runner up. And do it again for the final event results. For the need of source code, please refer to form23 in the source code attached.



It is the screen showing the scores of houses and personals. Here is the programming code of calculating personal scores.

```
        Dim cmdstr As String = "select * from SDMCompInfo where proj_code='" + Form1.Proj + "'"
        Dim cdt As New DataTable
        Form1.fill_table(cmdstr, cdt)
        cmdstr = "update " + Form1.Proj + "_results set Score=" + Trim(Str(cdt.Rows(0)("bonus_for_participation"))) + " where Result not like 'ABS'"
        Form1.enq_exe(cmdstr)
        cmdstr = "update " + Form1.Proj + "_results set Score=" + Trim(Str(cdt.Rows(0)("absentee_penalty"))) + " where Absent like 'ABS'"
        Form1.enq_exe(cmdstr)
        cmdstr = "select Event_Code from " + Form1.Proj + "_event where Assign_Lane='0' and Event_Code <>'100NNN' union select Event_Code from " + Form1.Proj + "_finals"
        Dim edt As New DataTable
        Form1.fill_table(cmdstr, edt)
        For i = 0 To edt.Rows.Count - 1
            cmdstr = "update " + Form1.Proj + "_results r set score=score+(select if(r.result=r1.result, " + Trim(Str(cdt.Rows(0)("position1_score"))) + ", if(r.result=r2.result, " + Trim(Str(cdt.Rows(0)("position2_score"))) + ", if(r.result=r3.result, " + Trim(Str(cdt.Rows(0)("position3_score"))) + ", if(r.result=r4.result, " + Trim(Str(cdt.Rows(0)("position4_score"))) + ", if(r.result=r5.result, " + Trim(Str(cdt.Rows(0)("position5_score"))) + ", if(r.result=r6.result, " + Trim(Str(cdt.Rows(0)("position6_score"))) + ", if(r.result=r7.result, " + Trim(Str(cdt.Rows(0)("position7_score"))) + ", if(r.result=r8.result, " + Trim(Str(cdt.Rows(0)("position8_score"))) + ", 0)))))))) from (select result from " + Form1.Proj + "_results r1 where r1.Event_Code='" + edt.Rows(i)("Event_Code") + "' and Absent<>'ABS' order by result desc limit 0,1) r1, (select result from " + Form1.Proj + "_results r1 where r1.Event_Code='" + edt.Rows(i)("Event_Code") + "' and Absent<>'ABS' order by result desc limit 1,1) r2, (select result from " + Form1.Proj + "_results r1 where r1.Event_Code='" + edt.Rows(i)("Event_Code") + "' and Absent<>'ABS' order by result desc limit 2,1) r3, (select result from " + Form1.Proj + "_results r1 where r1.Event_Code='" + edt.Rows(i)("Event_Code")
```

```
+ "' and Absent<>'ABS' order by result desc limit 3,1) r4, (select result from " + Form1.Proj + "_results r1 where r1.Event_Code='" + edt.Rows(i)("Event_Code") + "' and Absent<>'ABS' order by result desc limit 4,1) r5, (select result from " + Form1.Proj + "_results r1 where r1.Event_Code='" + edt.Rows(i)("Event_Code") + "' and Absent<>'ABS' order by result desc limit 5,1) r6, (select result from " + Form1.Proj + "_results r1 where r1.Event_Code='" + edt.Rows(i)("Event_Code") + "' and Absent<>'ABS' order by result desc limit 6,1) r7, (select result from " + Form1.Proj + "_results r1 where r1.Event_Code='" + edt.Rows(i)("Event_Code") + "' and Absent<>'ABS' order by result desc limit 7,1) r8) where r.Event_Code='" + edt.Rows(i)("Event_Code") + "'"
            Form1.enq_exe(cmdstr)
        Next
        Form1.dgv_show2(Me.DataGridView5, "select Name, Class, Class_No, h.House_Name as House, Total_Score from " + Form1.Proj + "_student s, " + Form1.Proj + "_house h, (select SID, sum(Score) as Total_Score from " + Form1.Proj + "_results group by SID) sc where s.House=h.House_code and s.SID=sc.SID order by Total_Score desc")
```

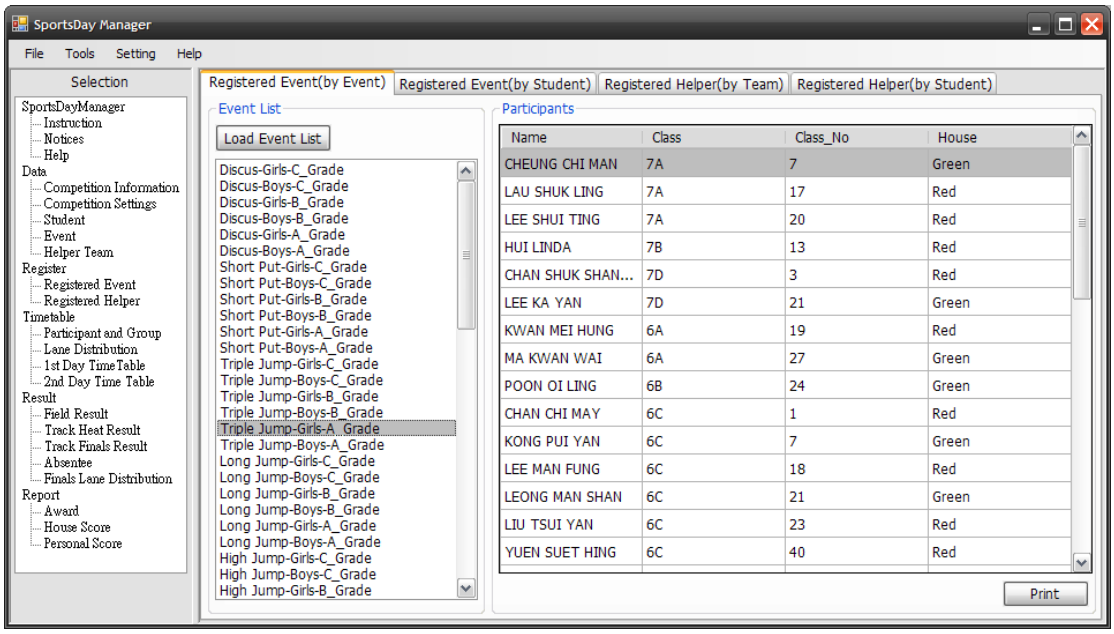
This programming code is an enhanced version, which can shorten the time for execution. The very long SQL update statement is for assigning the scores for each participant. And then the data grid view will show the accumulated score of each participant of the sports day. Here is the programming code of calculating house scores.

```
        Form1.dgv_show2(Me.DataGridView4, "select House_Name, Total_House_Score from " + Form1.Proj + "_house h, (select House, (House_Score+Adjustment) as Total_House_Score from " + Form1.Proj + "_adjusthousescore a, (select House, sum(Score) as House_Score from " + Form1.Proj + "_student s, " + Form1.Proj + "_results r where s.SID=r.SID group by House) hs where hs.House=a.House_code) t where t.House=h.House_code")
```

It is showing the accumulated score of each house. Moreover, the user can adjust the score by using the function ‘adjust house scores’ and ‘adjust personal scores’. Both adjustments require password to avoid cheating. Please refer to form23, form27 and form28 in the source code attached.

Print out data

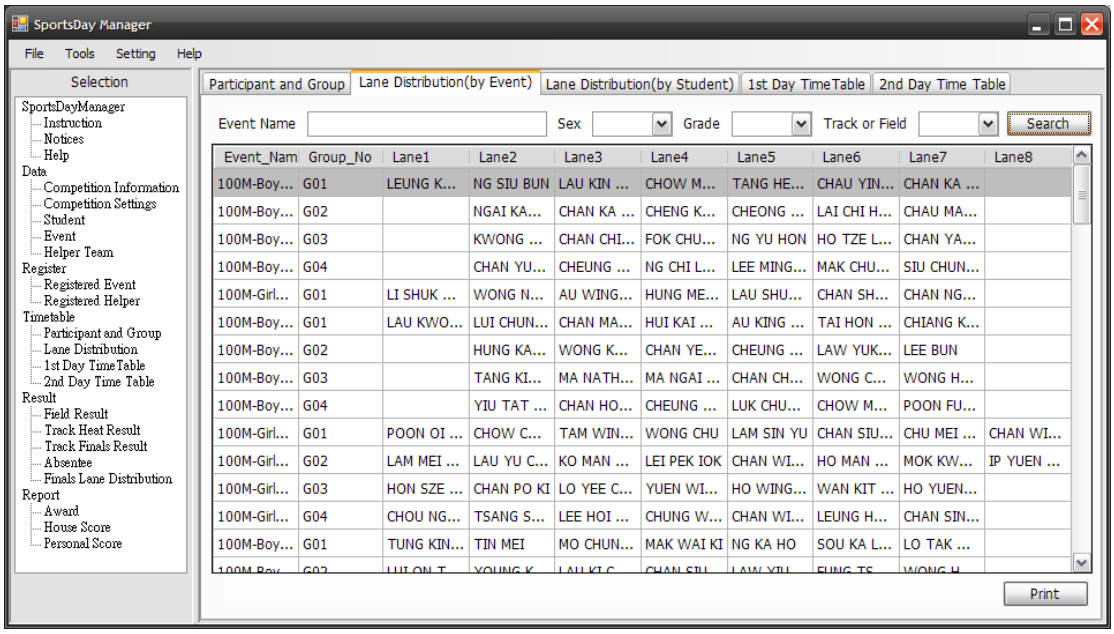
After all the events finished, the user can present prizes according to the report generated by the system. And they can print out such information by using the functions in the main form. There is always a button at the right bottom corner of the tab control of the main form. Here are some examples of them.



Triple Jump-Girls-A_Grade

Name	Class	Class_No	House
CHEUNG CHI MAN	7A	7	Green
LAU SHUK LING	7A	17	Red
LEE SHUI TING	7A	20	Red
HUI LINDA	7B	13	Red
CHAN SHUK SHAN ANNA	7D	3	Red
LEE KA YAN	7D	21	Green
KWAN MEI HUNG	6A	19	Red
MA KWAN WAI	6A	27	Green
POON OI LING	6B	24	Green
CHAN CHI MAY	6C	1	Red
KONG PUI YAN	6C	7	Green
LEE MAN FUNG	6C	18	Red
LEONG MAN SHAN	6C	21	Green
LIU TSUI YAN	6C	23	Red
YUEN SUET HING	6C	40	Red
LEE TSZ YAN	6D	17	Red
WONG SIU LING	6D	27	Red
TIN SUK MAN	5A	29	Yellow
NG SHUK FAN YONNIE	5B	26	Yellow
TSANG TSUI SHAN	5B	30	Green
TSANG YEE KI	5B	31	Green
HO WING SZE	5C	16	Green
HUNG MEI KAM	5C	17	Yellow
LEUNG SUK WAI	5C	24	Green
NG LAI KEI	5C	26	Green
TSANG WANG	5C	32	Red

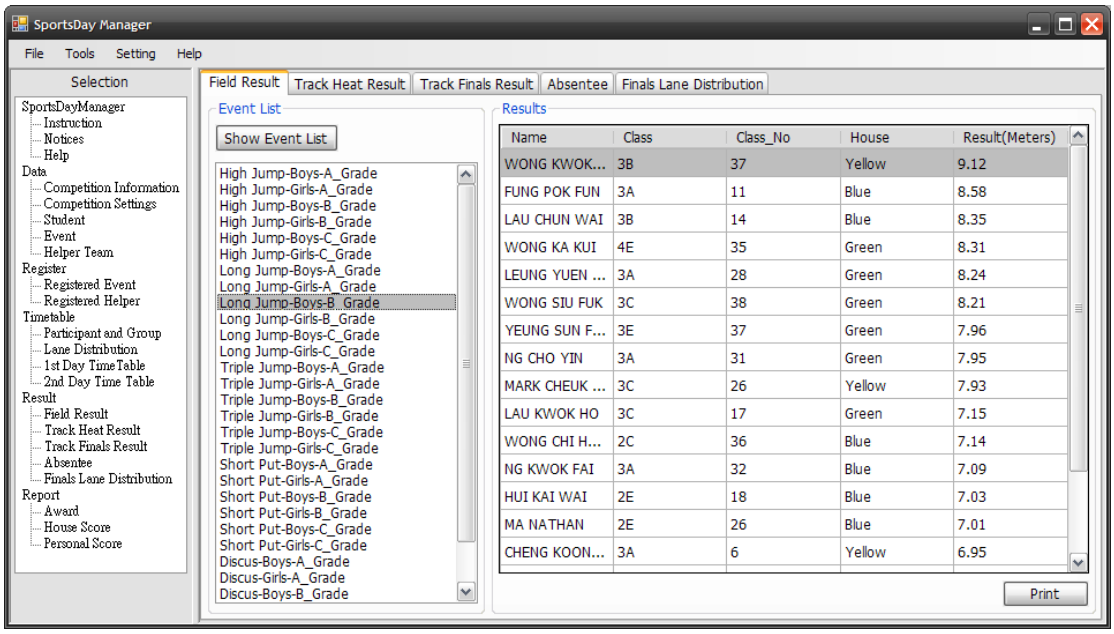
It is the participant list. It can be use to generate the number clothes for athletes by copying the list to Microsoft Office Excel or Word and using the function of mail merge. And then print out the document.



Event_Nam	Group_No	Lane1	Lane2	Lane3	Lane4	Lane5	Lane6	Lane7	Lane8
100M-Boy...	G01	LEUNG K...	NG SIU BUN	LAU KIN ...	CHOW M...	TANG HE...	CHAU YIN...	CHAN KA ...	
100M-Boy...	G02		NGAI KA...	CHAN KA ...	CHENG K...	CHEONG ...	LAI CHI H...	CHAU MA...	
100M-Boy...	G03		KWONG ...	CHAN CHI...	FOK CHU...	NG YU HON	HO TZE L...	CHAN YA...	
100M-Boy...	G04		CHAN YU...	CHEUNG ...	NG CHI L...	LEE MING...	MAK CHU...	SIU CHUN...	
100M-Girl...	G01	LI SHUK ...	WONG N...	AU WING...	HUNG ME...	LAU SHU...	CHAN SH...	CHAN NG...	
100M-Boy...	G01	LAU KWO...	LUI CHUN...	CHAN MA...	HUI KAI ...	AU KING ...	TAI HON ...	CHIANG K...	
100M-Boy...	G02		HUNG KA...	WONG K...	CHAN YE...	CHEUNG ...	LAW YUK...	LEE BUN	
100M-Boy...	G03		TANG KI...	MA NATH...	MA NGAI ...	CHAN CH...	WONG C...	WONG H...	
100M-Boy...	G04		YIU TAT ...	CHAN HO...	CHEUNG ...	LUK CHU...	CHOW M...	POON FU...	
100M-Girl...	G01	POON OI ...	CHOW C...	TAM WIN...	WONG CHU	LAM SIN YU	CHAN SIU...	CHU MEI ...	CHAN WI...
100M-Girl...	G02	LAM MEI ...	LAU YU C...	KO MAN ...	LEI PEK IOK	CHAN WI...	HO MAN ...	MOK KW...	IP YUEN ...
100M-Girl...	G03	HON SZE ...	CHAN PO KI	LO YEE C...	YUEN WI...	HO WING...	WAN KIT ...	HO YUEN...	
100M-Girl...	G04	CHOU NG...	TSANG S...	LEE HOI ...	CHUNG W...	CHAN WI...	LEUNG H...	CHAN SIN...	
100M-Boy...	G01	TUNG KIN...	TIN MEI	MO CHUN...	MAK WAI KI	NG KA HO	SOU KA L...	LO TAK ...	

Event_Name	Group_No	Lane1	Lane2	Lane3	Lane4	Lane5	Lane6	Lane7	Lane8
100M-Boys-A_Grade	G01	LEUNG KA LUNG	NG SIU BUN	LAU KIN CHUNG	CHOW MAN CHEONG	TANG HEUNG MAN	CHAU YING FAI	CHAN KA TAI	
100M-Boys-A_Grade	G02		NGAI KAM CHIU	CHAN KA HO	CHENG KWOK FUNG	CHEONG HING PANG	LAI CHI HANG	CHAU MAN FUNG	
100M-Boys-A_Grade	G03		KWONG KAM WING	CHAN CHI CHUEN	FOK CHUN PONG	NG YU HON	HO TZE LUNG	CHAN YAT KWAN EDWARD	
100M-Boys-A_Grade	G04		CHAN YUNG	CHEUNG KOON LUN,ALAN	NG CHI LUNG	LEE MING TAT	MAK CHUN PONG	SIU CHUN BONG	
100M-Girls-A_Grade	G01	LI SHUK HAN	WONG NGAI MUN	AU WING HANG	HUNG MEI KAM	LAU SHUK LING	CHAN SHUK SHAN ANNA	CHAN NGA MEI	
100M-Boys-B_Grade	G01	LAU KWOK HO	LUI CHUN FAI	CHAN MAN KO	HUI KAI WAI	AU KING CHEUNG	TAI HON SUM,VINCENT	CHIANG KA CHI	
100M-Boys-B_Grade	G02		HUNG KAI MING	WONG KA KUI	CHAN YEUNG	CHEUNG CHIN HUNG	LAW YUK LUN	LEE BUN	
100M-Boys-B_Grade	G03		TANG KING PONG	MA NATHAN	MA NGAI FUNG	CHAN CHUN PONG	WONG CHI CHEONG	WONG HON FAI	
100M-Boys-B_Grade	G04		YIU TAT PONG	CHAN HOI YIN	CHEUNG CHUNG MAN	LUK CHUNG HO	CHOW MAN KIT	POON FUK CHI	
100M-Girls-B_Grade	G01	POON OI YEN CARMEN	CHOW CHING LAN	TAM WING YAN SMANDA	WONG CHU	LAM SIN YU	CHAN SIU WAN	CHU MEI YAN,MARIAN	CHAN WING SZE
100M-Girls-B_Grade	G02	LAM MEI LIN	LAU YU CHUNG	KO MAN WAI	LEI PEK IOK	CHAN WING YAN, CHELSIA	HO MAN YEE	MOK KWAI HA	IP YUEN SHAN
100M-Girls-B_Grade	G03	HON SZE MAN	CHAN PO KI	LO YEE CHONG	YUEN WING SUM	HO WING HA CHRISTINE	WAN KIT YING	HO YUEN MEI	
100M-Girls-B_Grade	G04	CHOU NGAN PING	TSANG SUK HAN	LEE HOI YAN	CHUNG WAI MAN	CHAN WING YEE	LEUNG HOI LUN HESTER	CHAN SIN BUN	
100M-Boys-C_Grade	G01	TUNG KIN HO	TIN MEI	MO CHUNG CHI	MAK WAI KI	NG KA HO	SOU KA LUN	LO TAK MAN JOEL	

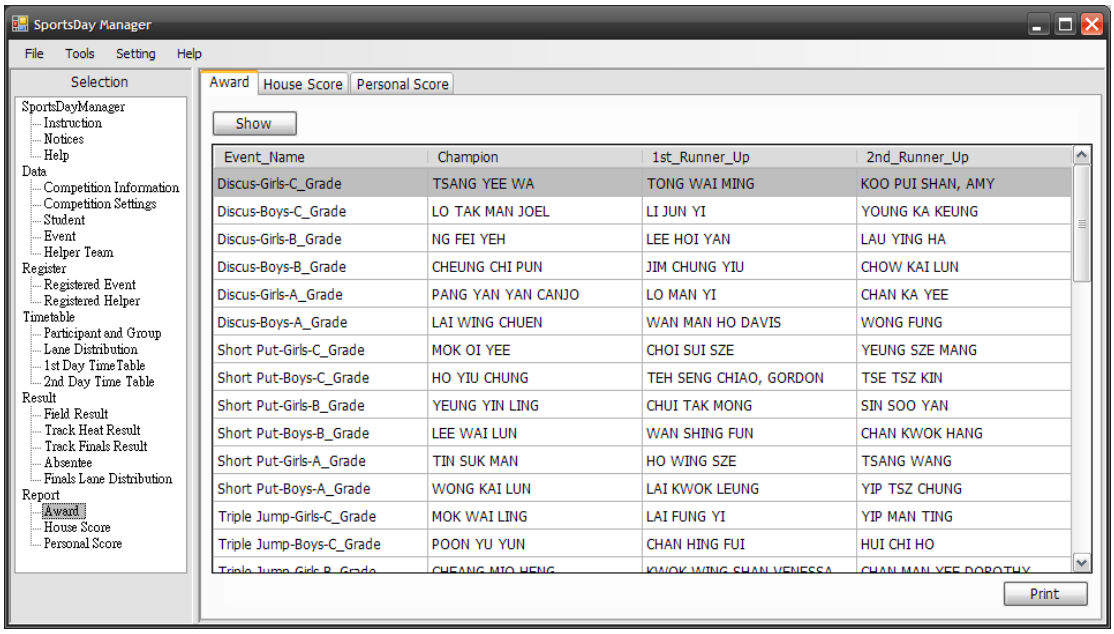
It is the lane assignment list.



Long Jump-Boys-B_Grade

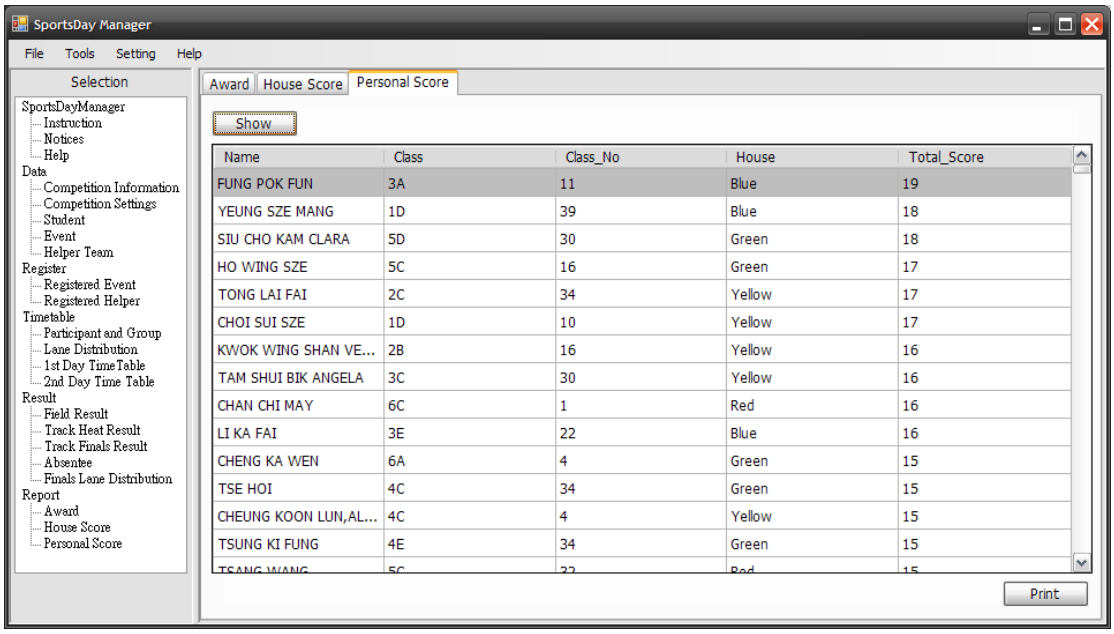
Name	Class	Class_No	House	Result(Meters)
WONG KWOK KIT	3B	37	Yellow	9.12
FUNG POK FUN	3A	11	Blue	8.58
LAU CHUN WAI	3B	14	Blue	8.35
WONG KA KUI	4E	35	Green	8.31
LEUNG YUEN KWON	3A	28	Green	8.24
WONG SIU FUK	3C	38	Green	8.21
YEUNG SUN FUK	3E	37	Green	7.96
NG CHO YIN	3A	31	Green	7.95
MARK CHEUK YIU	3C	26	Yellow	7.93
LAU KWOK HO	3C	17	Green	7.15
WONG CHI HANG	2C	36	Blue	7.14
NG KWOK FAI	3A	32	Blue	7.09
HUI KAI WAI	2E	18	Blue	7.03
MA NATHAN	2E	26	Blue	7.01
CHENG KOON KAU	3A	6	Yellow	6.95
KWOK CHI WAI	3A	14	Blue	6.66
LIU YING HON	3E	24	Yellow	6.64
LAU SZE FUNG	3E	15	Green	6.40
CHEUNG CHEUK HANG	3A	7	Green	6.36

It is the event results list.



Event_Name	Champion	1st_Runner_Up	2nd_Runner_Up
Discus-Girls-C_Grade	TSANG YEE WA	TONG WAI MING	KOO PUI SHAN, AMY
Discus-Boys-C_Grade	LO TAK MAN JOEL	LI JUN YI	YOUNG KA KEUNG
Discus-Girls-B_Grade	NG FEI YEH	LEE HOI YAN	LAU YING HA
Discus-Boys-B_Grade	CHEUNG CHI PUN	JIM CHUNG YIU	CHOW KAI LUN
Discus-Girls-A_Grade	PANG YAN YAN CANJO	LO MAN YI	CHAN KA YEE
Discus-Boys-A_Grade	LAI WING CHUEN	WAN MAN HO DAVIS	WONG FUNG
Short Put-Girls-C_Grade	MOK OI YEE	CHOI SUI SZE	YEUNG SZE MANG
Short Put-Boys-C_Grade	HO YIU CHUNG	TEH SENG CHIAO, GORDON	TSE TSZ KIN
Short Put-Girls-B_Grade	YEUNG YIN LING	CHUI TAK MONG	SIN SOO YAN
Short Put-Boys-B_Grade	LEE WAI LUN	WAN SHING FUN	CHAN KWOK HANG
Short Put-Girls-A_Grade	TIN SUK MAN	HO WING SZE	TSANG WANG
Short Put-Boys-A_Grade	WONG KAI LUN	LAI KWOK LEUNG	YIP TSZ CHUNG
Triple Jump-Girls-C_Grade	MOK WAI LING	LAI FUNG YI	YIP MAN TING
Triple Jump-Boys-C_Grade	POON YU YUN	CHAN HING FUI	HUI CHI HO
Triple Jump-Girls-B_Grade	CHEANG MIO HENG	KWOK WING SHAN VENESSA	CHAN MAN YEE DOROTHY
Triple Jump-Boys-B_Grade	LI KA FAI	LI CHI WA EMMANUEL	CHEUNG CHIN HUNG
Triple Jump-Girls-A_Grade	CHAU KA LAI	HO WING SZE	TSANG WANG
Triple Jump-Boys-A_Grade	CHEONG HING PANG	NG WAI SHAN	LAW HUNG YUE
Long Jump-Girls-C_Grade	LAM MEI YIN	YAN OI TSZ	LAU YUK YUNG
Long Jump-Boys-C_Grade	IP CHUN CHOI	LAI WING HANG	LEUNG HING LUN
Long Jump-Girls-B_Grade	HO WING HA CHRISTINE	HO WING YAN PAMELA	IP YUEN SHAN
Long Jump-Boys-B_Grade	WONG KWOK KIT	FUNG POK FUN	LAU CHUN WAI
Long Jump-Girls-A_Grade	SIU CHO KAM CLARA	TAM WING SEE CHELSIA	WONG SIU LING
Long Jump-Boys-A_Grade	HO KA MING	CHENG KA WEN	YAU LEONG MING
High Jump-Girls-C_Grade	YEUNG SZE MANG	CHOI SUI SZE	WOO SUK YIN

It is the winners list.



Name	Class	Class_No	House	Total_Score
FUNG POK FUN	3A	11	Blue	19
YEUNG SZE MANG	1D	39	Blue	18
SIU CHO KAM CLARA	5D	30	Green	18
HO WING SZE	5C	16	Green	17
TONG LAI FAI	2C	34	Yellow	17
CHOI SUI SZE	1D	10	Yellow	17
KWOK WING SHAN VENESSA	2B	16	Yellow	16
TAM SHUI BIK ANGELA	3C	30	Yellow	16
CHAN CHI MAY	6C	1	Red	16
LI KA FAI	3E	22	Blue	16
CHENG KA WEN	6A	4	Green	15
TSE HOI	4C	34	Green	15
CHEUNG KOON LUN,ALAN	4C	4	Yellow	15
TSUNG KI FUNG	4E	34	Green	15
TSANG WANG	5C	32	Red	15
CHOW TSUN HUNG	6A	9	Green	15
CHEUNG CHI PUN	3C	9	Green	15
HUI CHI HO	2C	12	Blue	14
CHAN KWOK HANG	3C	1	Green	14
YIP MAN TING	1B	40	Blue	14
LONG TAK YEE	6C	27	Green	14
WOO WAI MAN	2B	40	Blue	14
IP CHUN CHOI	2E	19	Blue	14
TAM WING SEE CHELSIA	4C	31	Green	14
POON YU YUN	2C	33	Green	13

It is the personal scores list. For the need of source code, please refer to form1 'print_html' in the source code attached.

Additionally, there are some forms for other use, but not be introduced, form3, 5, 8, 29, 30. Form3 is used for setting the connection of the system to the database server. Form5 is used for debugging and building basic tables, it can be called out by inputting 'DebugToolsBox' into the password textbox in Load Project. Form8 is used for editing project, changing password, changing description, etc. Form29 is used for noticing the user that the programming is running but not 'no responds'. Form30 will be introduced later since it is enhancing the program to be more users friendly. For the need of source code, please refer to the source code.

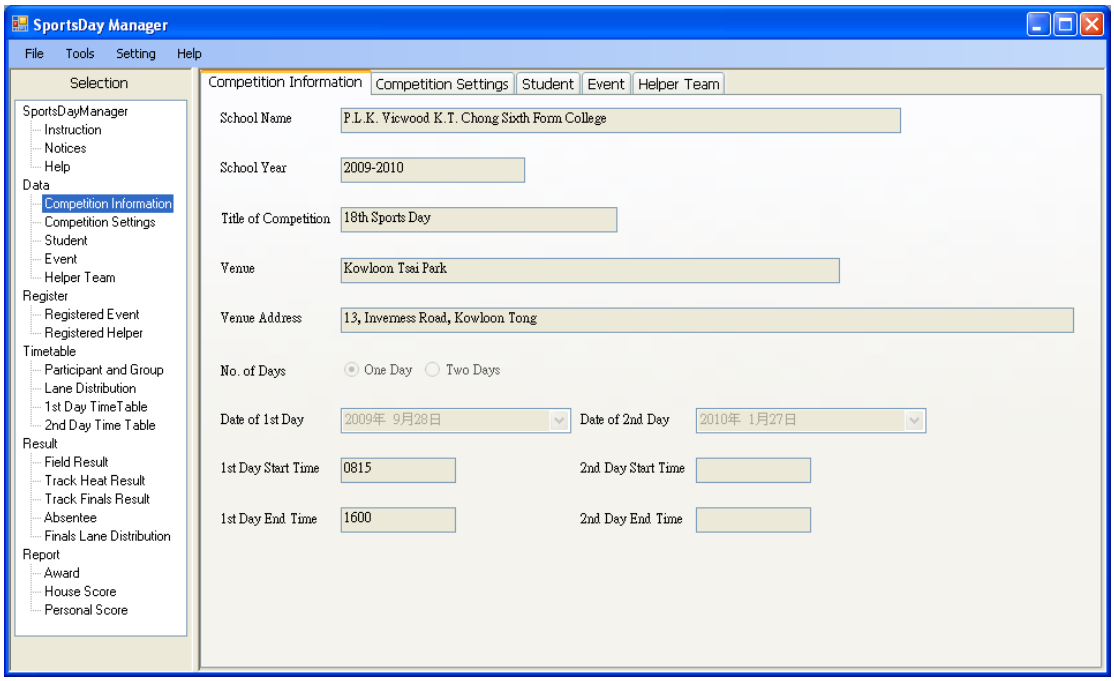
Testing and Evaluation of the system

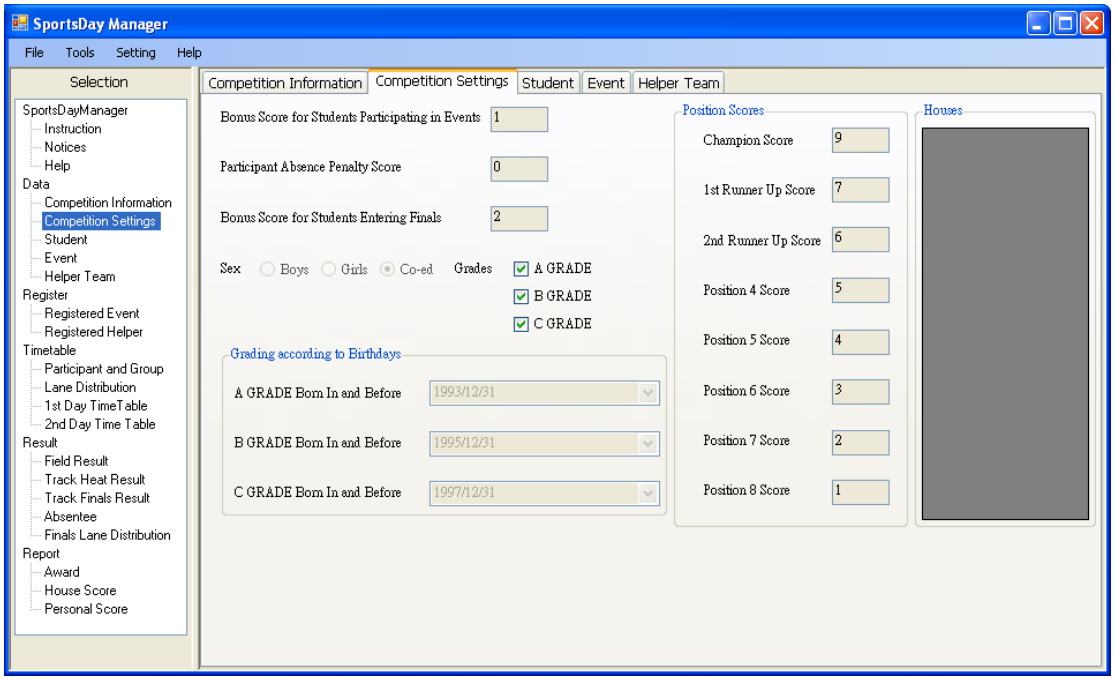
After completed the programming codes, I will try to dig out some bugs and fix them. And also try to do some enhancement on the system. I am going to do some tests by myself first and then invite someone to test it.

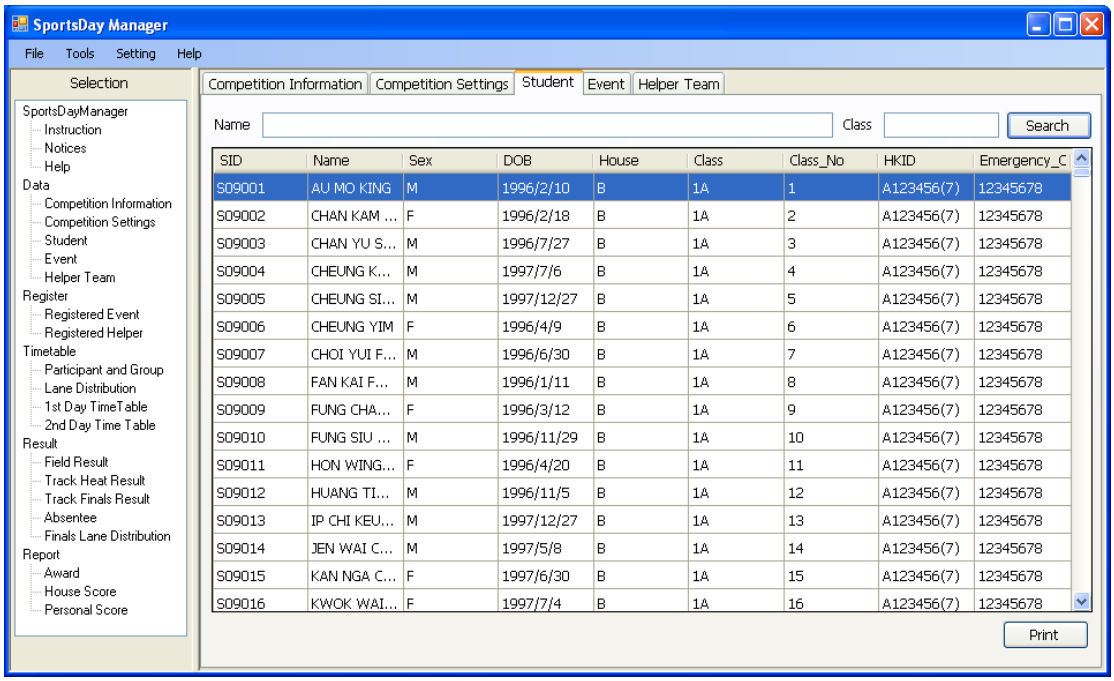
Self testing

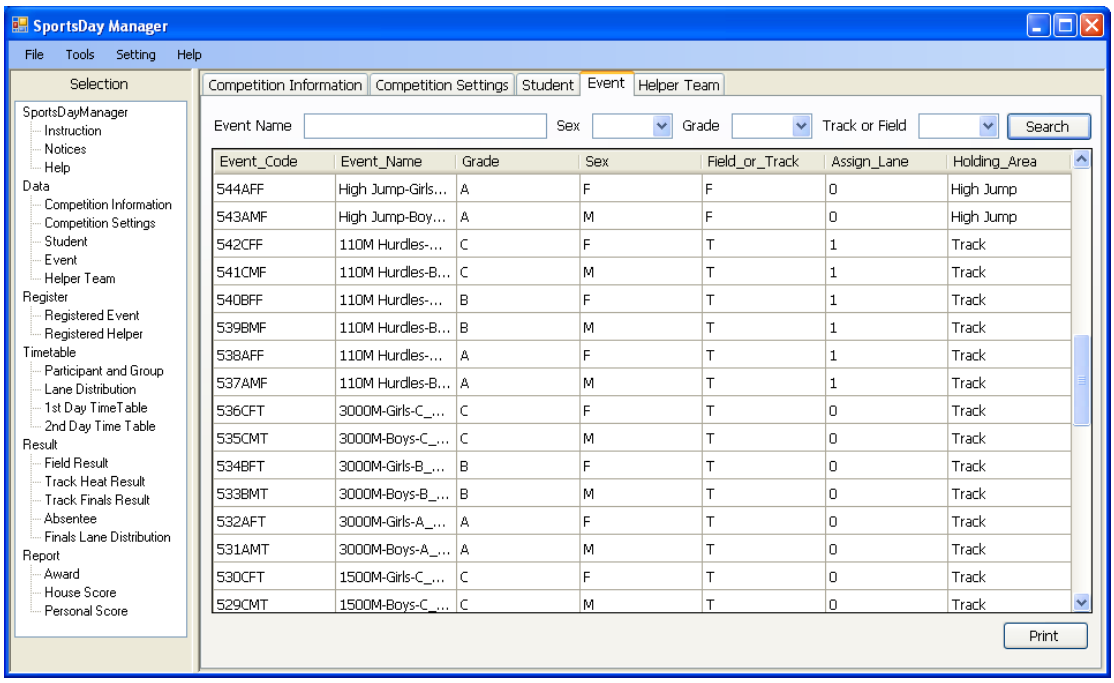
I will create a project and input some data into the system for making a scenario. I take some basic information of the sports day held by my school and some hypothetical data. And then use the function 'Import students through MySQL' to import around one thousand students' profiles, also use the event wizard to create some events and set default helper teams. After that, I use some of computers in my school to test the sub-program 'event register', which aim at providing a platform for concurrent registration for students. Moreover, I will write a computer program simulating the registration process since I tried not to input thousands of data manually. And then follow the use of the system, grouping and assign lanes, planning a time table, and printing out the athletes' number cloth. After these, I will input some results of participants by using 'update result'. Again, I will write another computer program to simulate the results of participants since I cannot input thousands of data manually, same as I wrote another program for event registration. Finally, I will use the system to compute the merits and print out the report.

I successfully use 'data entry wizard' to input all of the data and start the registration process. (The HKID and emergency call phone number is not important, so I input the same HKID and emergency call phone number for all students)

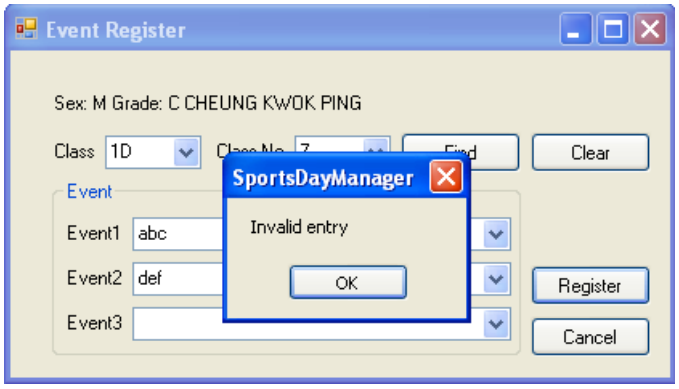





SportsDay Manager
File
Tools
Setting
Help
Selection
SportsDayManager
Instruction
Notices
Help
Data
Competition Information
Competition Settings
Student
Event
Helper Team
Register
Registered Event
Registered Helper
Timetable
Participant and Group
Lane Distribution
1st Day TimeTable
2nd Day Time Table
Result
Field Result
Track Heat Result
Track Finals Result
Absentee
Finals Lane Distribution
Report
Award
House Score
Personal Score
Competition Information
Competition Settings
Student
Event
Helper Team
Name
Class
Search
SID
Name
Sex
DOB
House
Class
Class_No
HKID
Emergency_C
S09001 AU MO KING M 1996/2/10 B 1A 1 A123456(7) 12345678
S09002 CHAN KAM ... F 1996/2/18 B 1A 2 A123456(7) 12345678
S09003 CHAN YU S... M 1996/7/27 B 1A 3 A123456(7) 12345678
S09004 CHEUNG K... M 1997/7/6 B 1A 4 A123456(7) 12345678
S09005 CHEUNG SI... M 1997/12/27 B 1A 5 A123456(7) 12345678
S09006 CHEUNG YIM F 1996/4/9 B 1A 6 A123456(7) 12345678
S09007 CHOI YUI F... M 1996/6/30 B 1A 7 A123456(7) 12345678
S09008 FAN KAI F... M 1996/1/11 B 1A 8 A123456(7) 12345678
S09009 FUNG CHA... F 1996/3/12 B 1A 9 A123456(7) 12345678
S09010 FUNG SIU ... M 1996/11/29 B 1A 10 A123456(7) 12345678
S09011 HON WING... F 1996/4/20 B 1A 11 A123456(7) 12345678
S09012 HUANG TI... M 1996/11/5 B 1A 12 A123456(7) 12345678
S09013 IP CHI KEU... M 1997/12/27 B 1A 13 A123456(7) 12345678
S09014 JEN WAI C... M 1997/5/8 B 1A 14 A123456(7) 12345678
S09015 KAN NGA C... F 1997/6/30 B 1A 15 A123456(7) 12345678
S09016 KWOK WAI... F 1997/7/4 B 1A 16 A123456(7) 12345678
Print


SportsDay Manager
File
Tools
Setting
Help
Selection
SportsDayManager
Instruction
Notices
Help
Data
Competition Information
Competition Settings
Student
Event
Helper Team
Register
Registered Event
Registered Helper
Timetable
Participant and Group
Lane Distribution
1st Day TimeTable
2nd Day Time Table
Result
Field Result
Track Heat Result
Track Finals Result
Absentee
Finals Lane Distribution
Report
Award
House Score
Personal Score
Competition Information
Competition Settings
Student
Event
Helper Team
Event Name
Sex
Grade
Track or Field
Search
Event_Code
Event_Name
Grade
Sex
Field_or_Track
Assign_Lane
Holding_Area
544AFF High Jump-Girls... A F F 0 High Jump
543AMF High Jump-Boy... A M F 0 High Jump
542CFF 110M Hurdles-... C F T 1 Track
541CMF 110M Hurdles-B... C M T 1 Track
540BFF 110M Hurdles-... B F T 1 Track
539BMF 110M Hurdles-B... B M T 1 Track
538AFF 110M Hurdles-... A F T 1 Track
537AMF 110M Hurdles-B... A M T 1 Track
536CFT 3000M-Girls-C_... C F T 0 Track
535CMT 3000M-Boys-C_... C M T 0 Track
534BFT 3000M-Girls-B_... B F T 0 Track
533BMT 3000M-Boys-B_... B M T 0 Track
532AFT 3000M-Girls-A_... A F T 0 Track
531AMT 3000M-Boys-A_... A M T 0 Track
530CFT 1500M-Girls-C_... C F T 0 Track
529CMT 1500M-Boys-C_... C M T 0 Track
Print

And then I inputted some registration form into the system, I find that it is possible to type some words to the combo boxes, which may cause error. So, I modified the codes and restricted the user to choose the items provided. (The code mentioned in implementation is the modified version, and the old version can be found in SportsDayManager_Ver1.0_Beta_20091027, which is in the CD of attachment.)

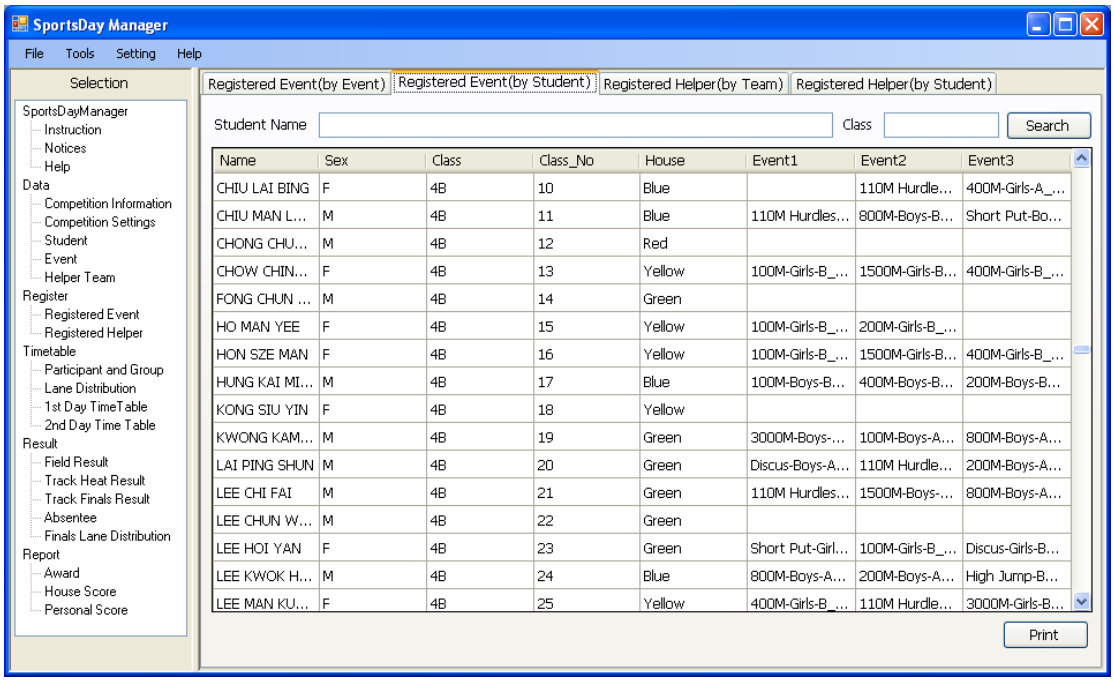


After this, I write a program to simulate the registration process. Here is the programming code of it.

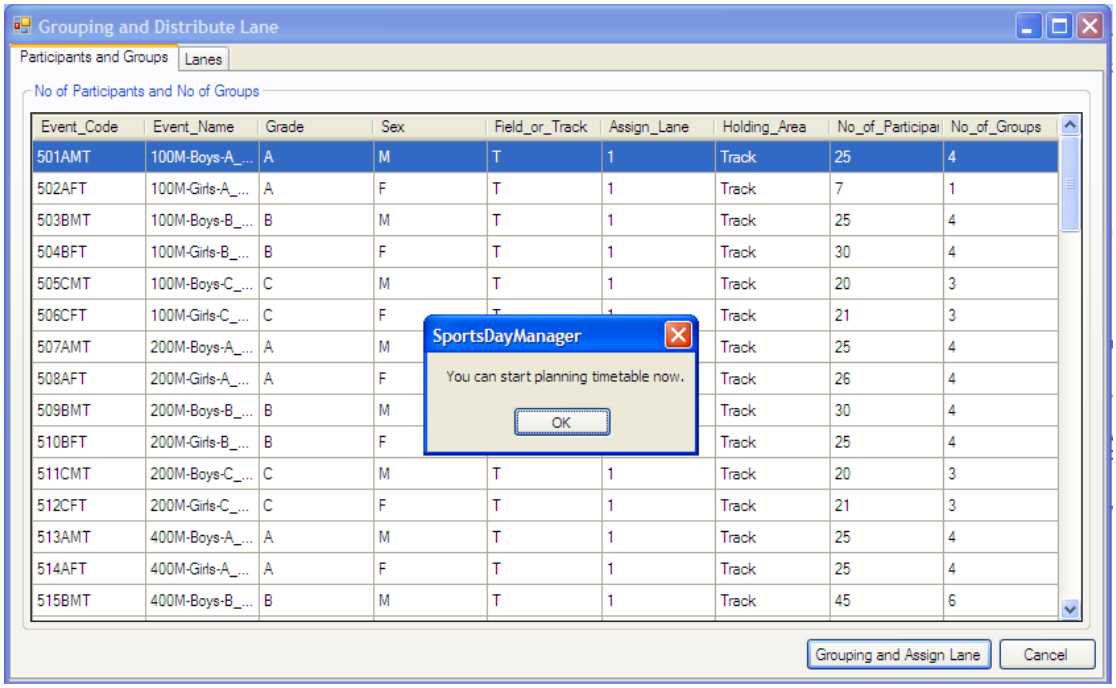
```
        Dim sx As String
        Dim gd As String
        Dim p_code As String = TextBox2.Text
        For sex = 0 To 1
            If sex = 0 Then
                sx = "M"
            Else
                sx = "F"
            End If
            For grade = 0 To 2
                If grade = 0 Then
                    gd = "A"
                ElseIf grade = 1 Then
                    gd = "B"
                Else
                    gd = "C"
                End If
                Dim cmdstr As String = "select * from " + p_code + "_student s, (select SID, 'C' as Grade from " + p_code + "_student where DOB between (select subdate((b_grade_time),-1) from SDMCompInfo where proj_code ='" + p_code + "') and (select c_grade_time from SDMCompInfo where proj_code ='" + p_code + "') union select SID, 'B' as Grade from "+ p_code + "_student where DOB between (select subdate((a_grade_time),-1) from SDMCompInfo where proj_code ='"+ p_code + "') and (select b_grade_time from SDMCompInfo where proj_code ='" + p_code + "') union select SID, 'A' as Grade from " + p_code + "_student where DOB between '1980/1/1' and (select a_grade_time from SDMCompInfo where proj_code ='" + p_code + "')) g where s.SID=g.SID and Sex ='" + sx + "' and Grade ='" + gd + "'"
                Dim std As New DataTable
                fill_table(cmdstr, std)
                cmdstr = "select * from " + p_code + "_event where Event_Code <>'100NNN' and Sex ='" + sx + "' and Grade ='" + gd + "'"
                Dim etd As New DataTable
                fill_table(cmdstr, etd)
                Dim e_code(etd.Rows.Count) As String
                Dim sid(std.Rows.Count) As String
                For j = 0 To etd.Rows.Count - 1
                    e_code(j) = etd.Rows(j)("Event_Code")
                Next
                For j = 0 To etd.Rows.Count - 1
                    Debug.Print(e_code(j))
```

```
            Next
            For j = 0 To std.Rows.Count - 1
                sid(j) = std.Rows(j)("SID")
            Next
            Randomize()
            Dim r1, r2 As Integer
            Dim tmp As String
            For j = 0 To 200
                Do
                    r1 = Int(Rnd() * sid.Length)
                    r2 = Int(Rnd() * sid.Length)
                Loop Until (r1 <> r2)
                tmp = sid(r1)
                sid(r1) = sid(r2)
                sid(r2) = tmp
            Next
            For j = 0 To 100 Step 5
                reg(j, j + 4, e_code, sid, p_code)
            Next
        Next
    Next
```

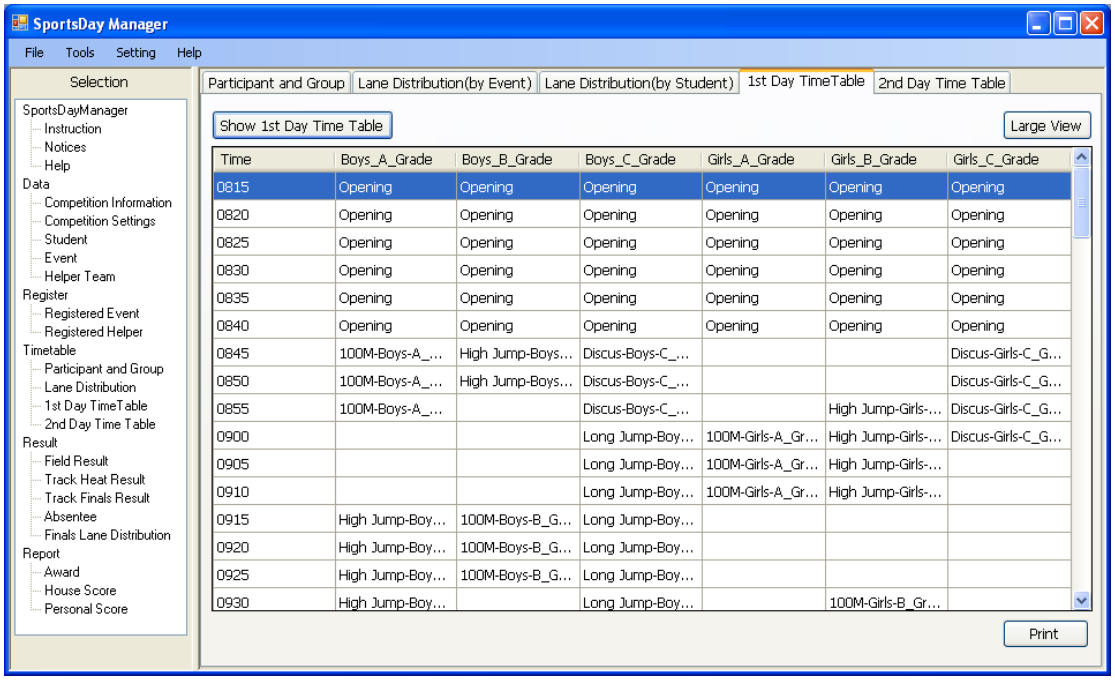
These codes are similar to the codes in event register, which is form 17, but adding some codes for generating random data for simulating registration process. Using these codes, I successfully inputted the registration information. I also recorded the random data and checked the data stored in the database server, all data are correctly passed to the database server.



And then the process of printing number clothes, and grouping and assign lanes are completed successfully.

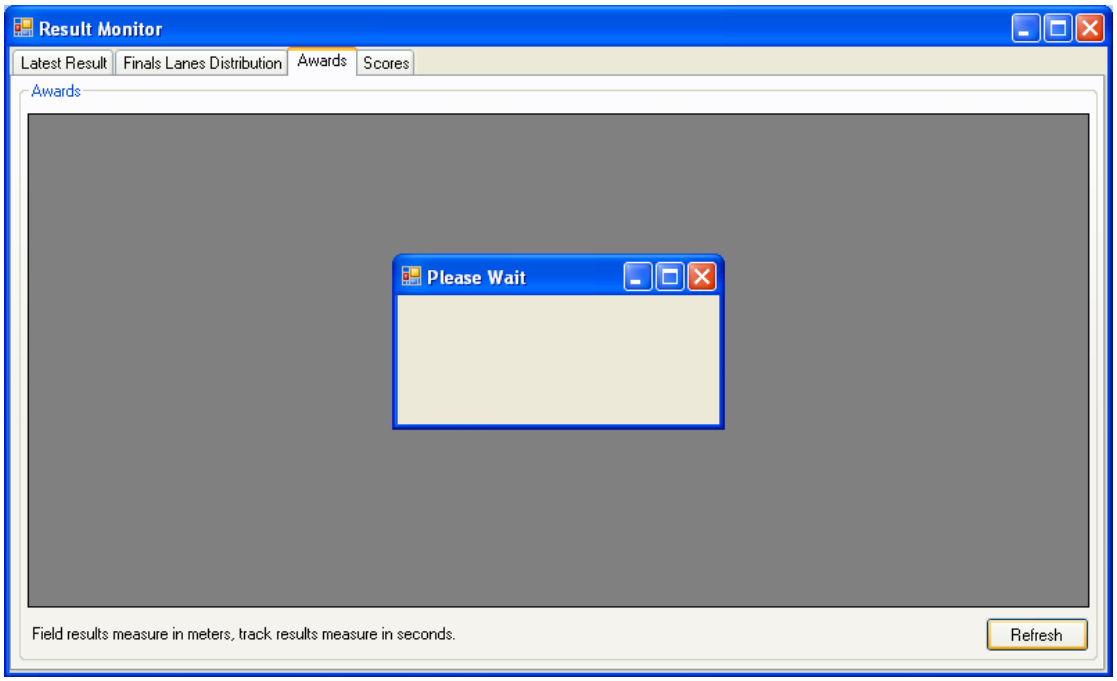


In the planning of time table, I successfully generated a time table.



In the process of running the sports day, I inputted some events result data into the system.

And then try to use result monitor to see the inputted result and calculate the merits and awards. I found that the system requires a long time for computing such data (around one minute).

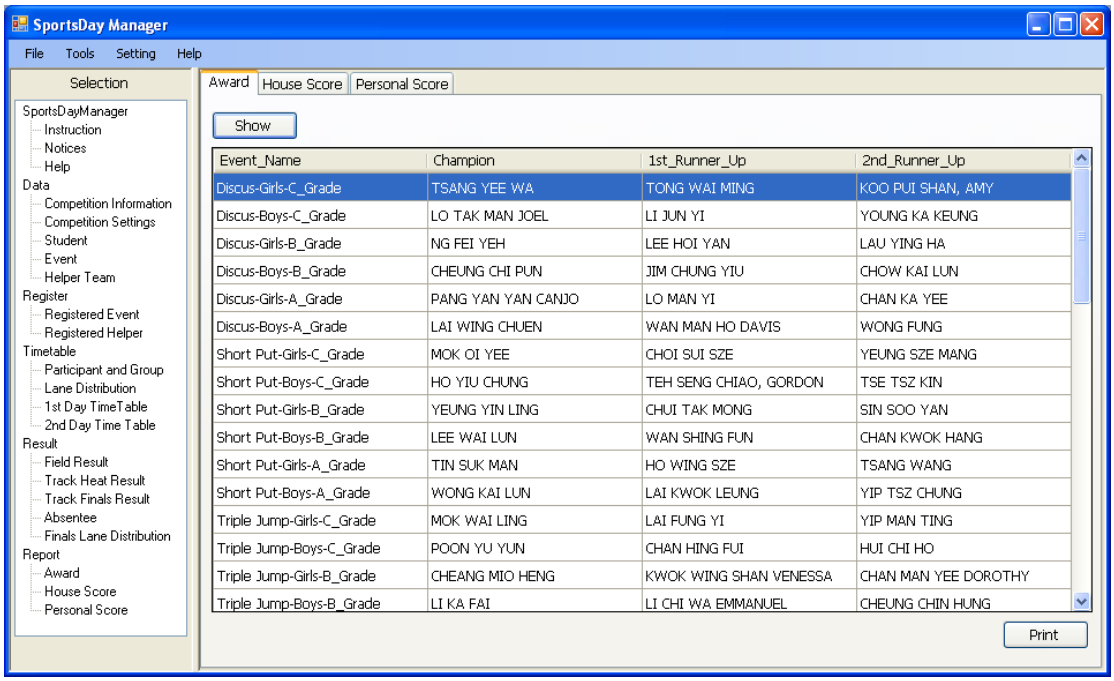


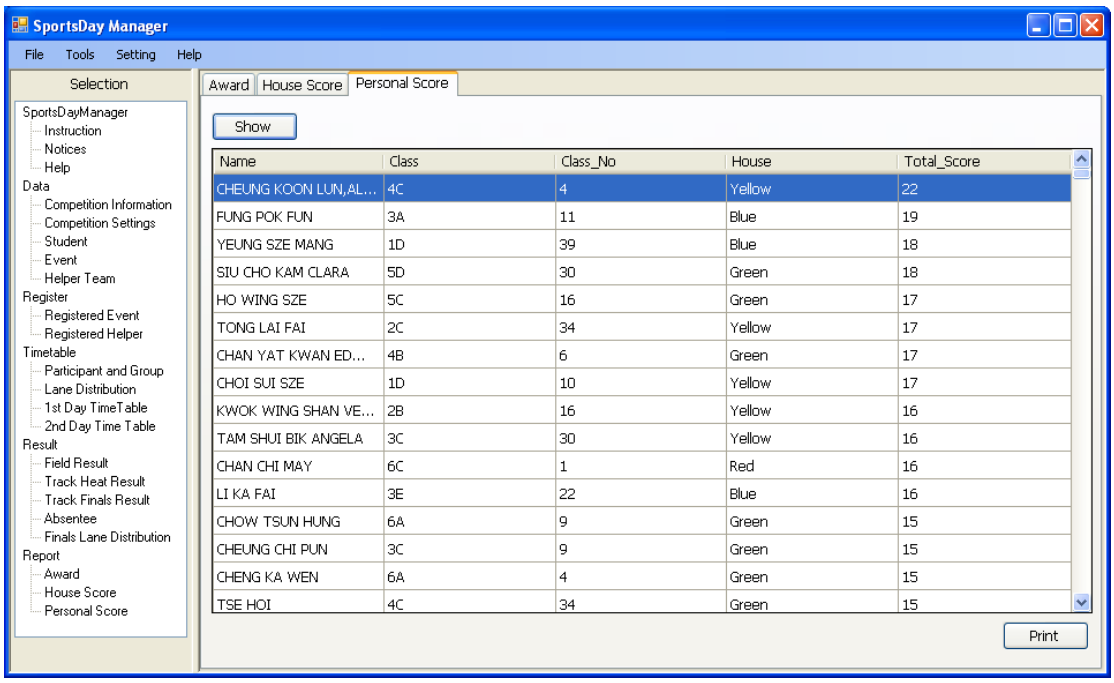
And because of this, I study the program and try to dig out the reason. I found that the reason is the system generates a number of short queries and the database server cannot respond quickly. I try to modify the program and make a few short query statements into one long query statement. The code mentioned in implementation is the modified version, and the old version can be found in SportsDayManager_Ver1.0_Beta_20091027, which is in the CD of attachment. And after that, I try to input more result by writing another program. Here is the programming code of it.

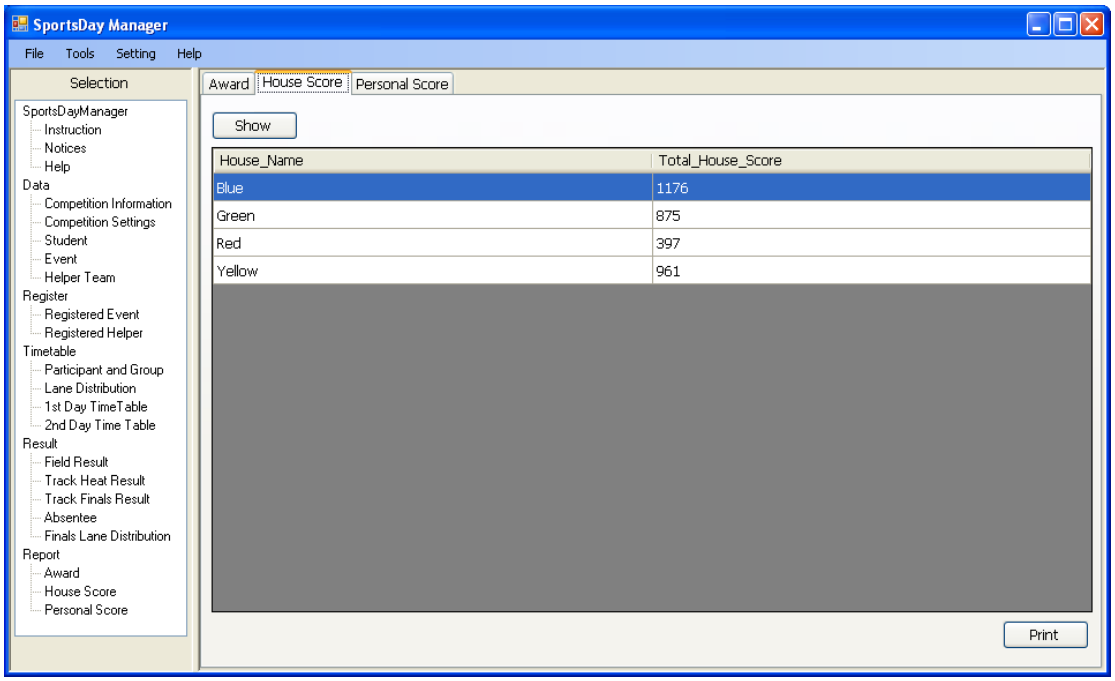
```
        Dim cmdstr As String = "select Event_Code, Event_Name from 1002_event where Event_Code<>'100NNN' order by Event_Code"
        Dim edt As New DataTable
        fill_table(cmdstr, edt)
        Dim e_code(edt.Rows.Count), e_name(edt.Rows.Count) As String
        Randomize()
        Dim randombase As Double
        Dim sid(0) As String
        For j = 0 To edt.Rows.Count - 1
            e_code(j) = edt.Rows(j)("Event_Code")
            e_name(j) = edt.Rows(j)("Event_Name")
        Next
        For j = 0 To edt.Rows.Count - 1
            If e_code(j).Contains("T") Then
                If e_name(j).Contains("100M") Then
                    randombase = 11.27
                ElseIf e_name(j).Contains("200M") Then
                    randombase = 24.19
                ElseIf e_name(j).Contains("400M") Then
                    randombase = 54.74
                ElseIf e_name(j).Contains("800M") Then
                    randombase = 133.87
```

```
                ElseIf e_name(j).Contains("1500") Then
                    randombase = 282.64
                ElseIf e_name(j).Contains("3000M") Then
                    randombase = 630
                ElseIf e_name(j).Contains("110M Hurdles") Then
                    randombase = 15.3
                End If
            Else
                If e_name(j).Contains("High Jump") Then
                    randombase = 1.71
                ElseIf e_name(j).Contains("Long Jump") Then
                    randombase = 6.29
                ElseIf e_name(j).Contains("Triple Jump") Then
                    randombase = 11.37
                ElseIf e_name(j).Contains("Short Put") Then
                    randombase = 11.84
                ElseIf e_name(j).Contains("Discus") Then
                    randombase = 28.75
                End If
            End If
            cmdstr = "select SID from 1002_eventregister where Event1='" + e_code(j) + "' or Event2='" +
e_code(j) + "' or Event3='" + e_code(j) + "'"
            Dim sdt As New DataTable
            fill_table(cmdstr, sdt)
            ReDim sid(sdt.Rows.Count)
            For k = 0 To sdt.Rows.Count - 1
                sid(k) = sdt.Rows(k)("SID")
            Next
            Dim results As Double
            Dim rand As Double
            For k = 0 To sdt.Rows.Count - 1
                rand = Rnd()
                If rand > 0.95 Then
                    results = 0
                Else
                    results = randombase * (rand * 0.5 + 1)
                End If
                If results = 0 Then
                    cmdstr = "insert into 1002_results(SID, Event_Code, Absent) values ('" + sid(k) + "',
'" + e_code(j) + "', 'ABS')"
                Else
                    cmdstr = "insert into 1002_results(SID, Event_Code, Result) values ('" + sid(k) + "',
'" + e_code(j) + "', '" + Str(results) + "')"
                End If
                enq_exe(cmdstr)
            Next
        Next
```

I used the records of events of my school's sports day to generate some hypothytical data. The program will add a random number to the record to make the results of athletes to be different and also make some to be absent for the events. And then, I use the enhanced 'result monitor' to compute the results and it reported the merits and results accurately.


SportsDay Manager
File
Tools
Setting
Help
Selection
SportsDayManager
Instruction
Notices
Help
Data
Competition Information
Competition Settings
Student
Event
Helper Team
Register
Registered Event
Registered Helper
Timetable
Participant and Group
Lane Distribution
1st Day TimeTable
2nd Day Time Table
Result
Field Result
Track Heat Result
Track Finals Result
Absentee
Finals Lane Distribution
Report
Award
House Score
Personal Score
Award
House Score
Personal Score
Show
Event_Name | Champion | 1st_Runner_Up | 2nd_Runner_Up
Discus-Girls-C_Grade | TSANG YEE WA | TONG WAI MING | KOO PUI SHAN, AMY
Discus-Boys-C_Grade | LO TAK MAN JOEL | LI JUN YI | YOUNG KA KEUNG
Discus-Girls-B_Grade | NG FEI YEH | LEE HOI YAN | LAU YING HA
Discus-Boys-B_Grade | CHEUNG CHI PUN | JIM CHUNG YIU | CHOW KAI LUN
Discus-Girls-A_Grade | PANG YAN YAN CANJO | LO MAN YI | CHAN KA YEE
Discus-Boys-A_Grade | LAI WING CHUEN | WAN MAN HO DAVIS | WONG FUNG
Short Put-Girls-C_Grade | MOK OI YEE | CHOI SUI SZE | YEUNG SZE MANG
Short Put-Boys-C_Grade | HO YIU CHUNG | TEH SENG CHIAO, GORDON | TSE TSZ KIN
Short Put-Girls-B_Grade | YEUNG YIN LING | CHUI TAK MONG | SIN SOO YAN
Short Put-Boys-B_Grade | LEE WAI LUN | WAN SHING FUN | CHAN KWOK HANG
Short Put-Girls-A_Grade | TIN SUK MAN | HO WING SZE | TSANG WANG
Short Put-Boys-A_Grade | WONG KAI LUN | LAI KWOK LEUNG | YIP TSZ CHUNG
Triple Jump-Girls-C_Grade | MOK WAI LING | LAI FUNG YI | YIP MAN TING
Triple Jump-Boys-C_Grade | POON YU YUN | CHAN HING FUI | HUI CHI HO
Triple Jump-Girls-B_Grade | CHEANG MIO HENG | KWOK WING SHAN VENESSA | CHAN MAN YEE DOROTHY
Triple Jump-Boys-B_Grade | LI KA FAI | LI CHI WA EMMANUEL | CHEUNG CHIN HUNG
Print


SportsDay Manager
File
Tools
Setting
Help
Selection
SportsDayManager
Instruction
Notices
Help
Data
Competition Information
Competition Settings
Student
Event
Helper Team
Register
Registered Event
Registered Helper
Timetable
Participant and Group
Lane Distribution
1st Day TimeTable
2nd Day Time Table
Result
Field Result
Track Heat Result
Track Finals Result
Absentee
Finals Lane Distribution
Report
Award
House Score
Personal Score
Award
House Score
Personal Score
Show
Name | Class | Class_No | House | Total_Score
CHEUNG KOON LUN,AL... | 4C | 4 | Yellow | 22
FUNG POK FUN | 3A | 11 | Blue | 19
YEUNG SZE MANG | 1D | 39 | Blue | 18
SIU CHO KAM CLARA | 5D | 30 | Green | 18
HO WING SZE | 5C | 16 | Green | 17
TONG LAI FAI | 2C | 34 | Yellow | 17
CHAN YAT KWAN ED... | 4B | 6 | Green | 17
CHOI SUI SZE | 1D | 10 | Yellow | 17
KWOK WING SHAN VE... | 2B | 16 | Yellow | 16
TAM SHUI BIK ANGELA | 3C | 30 | Yellow | 16
CHAN CHI MAY | 6C | 1 | Red | 16
LI KA FAI | 3E | 22 | Blue | 16
CHOW TSUN HUNG | 6A | 9 | Green | 15
CHEUNG CHI PUN | 3C | 9 | Green | 15
CHENG KA WEN | 6A | 4 | Green | 15
TSE HOI | 4C | 34 | Green | 15
Print

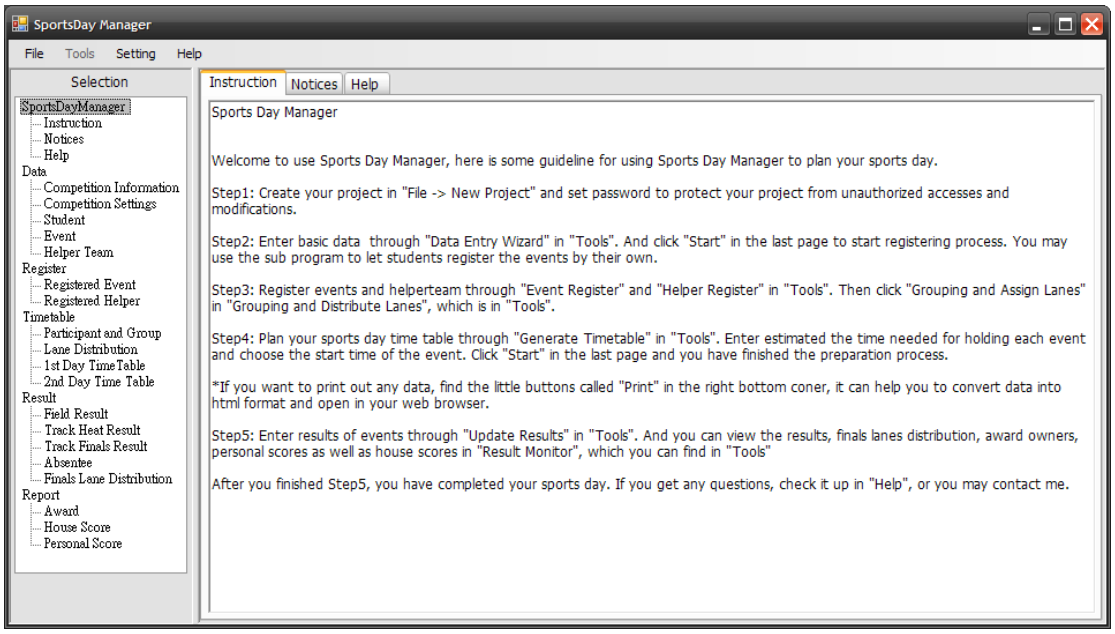


User testing

After the test of myself, I invited some people to use my system. Here are the comments from them.

One of them told me that he does not have an idea of how to use the program once he starts it. He is able to test since he asked me.

It is always a problem for developers. Developers know the program very clear, by when the program is passed to the others, they do not know the program anything even the first step of using it. So, I made an instruction page and placed it at the front page of the program. The user can read it once they open the program.



It shows all the step of using the system. And also, the buttons inside SportsDay Manager are initially disabled to click for preventing the users clicking them unintentionally. The buttons and menu strip items will be enabled according to the progress of sports day planning.

Another user told me that he faced some difficulties when using the system, especially he does not know there is some quick way to input data and how to call out the settings form to modify.

I designed a help function of the system, and it can be called out by clicking ‘help’ in the menu strip of the main form.

Minor amendment: Added an item into the tree view(Student Registration) to let the user can easily find the page for printing out number clothes.

Conclusion and Discussion

The system can correctly and accurately perform the tasks listed in the objective of the coursework. Correctly categorized students into different grades according to their date of birth (age). And also successfully arranged heats for events and allocated lanes to the athletes with eight participants for each heat. After the system receive the results of events, it can accurately report the ranking of every athletes and computed the individual champion and house champion based on the overall results.

Reflection

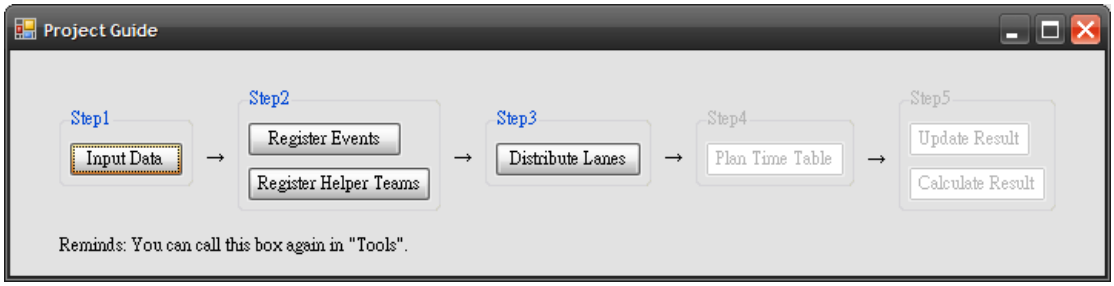
During the development process, there are some problems I have to tackle. For example, I do not know how to write a visual basic program that can access a MySQL server. I tried to find out something can help from searching on the internet and asked my teachers. I finally find out there is a connector for .NET and it is available to download in the MySQL official website. In the process of thinking the algorithms of performing the tasks, I failed a lot of times, but I finally go through them. Other than the tasks required in the objective, there are two difficult tasks, one is the project management, another one is the generation of time table. The project management system is important because it makes this sports day system not working for only one project and enhanced the security of the data inputted. The time table planning system is very complicated. It takes quite a long time for me to make it runs successfully. To generate an html file for the time tables, I have to find out some html tutorial to learn how of do the task. It combines the html coding skills and also some recursive functions. It is satisfactory to finish the whole development of the sports day system ‘Sports Day Manager’.

Improvement suggestions

However, there is still a room of improvement for the system. I have done some improvements after the testing from the users. But, I have some new ideas for the system to make it be more users friendly.

1. Make a user guide to help the users and let them know their progress of organizing the sports day.
2. The filters in the main form are not powerful enough. It is possible to make to filtering options be more detailed.
3. Support importing students' profiles by Microsoft Excel. Some schools do not hold their students' information in a MySQL server, supporting Microsoft Excel will be more convenience for the users.

I have tried to make the above improvement. Here is the layout design of the user guide.



I use a project progressed to step3 (stage2) as a demonstration. The button in the right hand side is unable to use because the project is not finished the 'grouping and distribution of lanes' process yet. For the need of source code, please refer to form30 in the source code attached.

However, suggestion2 and suggestion3 is not yet completed because both of them need some time to think out the algorithms. I can foresee that these two suggestions will help the users a lot if they can be included in the system. The users can print out the exact information that he/she wants by using the more detailed filters. And also can help the users to save a lot of time in inputting students' profiles. Many database software support exporting as Microsoft Excel but not transferring to MySQL. The idea of modifying the system to become compatible with Microsoft Excel will be a big improvement. Hope that I can do the improvement in the near future.

Acknowledgement

Thanks to my computer teachers, LWI, TCY and my school computer teacher, TCK, they helped me a lot in developing the system and doing the coursework. Also thanks to my two friends for testing my system.

References

http://en.wikipedia.org/wiki/Wiki
http://www.microsoft.com/express/vb/
http://www.microsoft.com/express/vc/
http://www.sun.com/
http://www.oracle.com/index.html
http://office.microsoft.com/en-us/default.aspx
http://msdn.microsoft.com/en-us/vfoxpro/default.aspx
http://www.mysql.com/
http://www.smartdraw.com/
http://msdn.microsoft.com/zh-tw/vbasic/default.aspx
http://www.w3schools.com/
http://www.kaner.com/pdfs/GoodTest.pdf

Visual Basic 2008 程式設計經典，碁峰資訊
Visual Basic 2005 功能索引式參考手冊

Software for referencing

Hong Kong Schools Sports Day, Swimming Gala and Games Day Computer Assisted Processing System, *Designed and programmed by Mr. CHIU Kam-wa, website: http://www.useit.us/software/pe/*

Software

Macromedia Fireworks 8
Microsoft Office Excel 2003
Microsoft Office Word 2003
Microsoft Visual Basic 2008 Express Edition
MySQL Query Browser
SmartDraw 2010